حسین سیلانی
طراح، توسعه دهنده وسیستم ادمین

Book 4\100

لینوکس را فارسی یادبگیرید
www.learninghive.ir
www.seilany.ir



1001 Linux Commands

3st Edition

:معرفی کوتاه کتاب

این مرجع فارسی به توضیح دستورات مهم و رایج در سیستم عامل لینوکس و توزیع های آن میپردازد. این کتاب نزدیک به 1000 مثال و دستور در قالب معرفی،شرح،ساختار،گزینه ها و مثالهای دستورات نوشته شده است.فصلهای ابتدایی مبانی و مفاهیم لینوکسی را در برمیگیرد و سایرفصل ها بر حسب کاربرد دستورات نسبت به هم نوشته شده اند

مرجع کامل دستورات لینوکس

- شرح دستورات
- ساختار دستورات
- گزینه های دستورات
- مثال های کامل دستورات

هزار و یک دستور لینوکس
1001 Linux commands

حسین سیلانی

ویرایش سوم

۱۴۰۳

سازمان اسناد و کتابخانه ملی جمهوری اسلامی ایران

سرشناسه: لطفی، فهیمه، ۱۳۶۷-
عنوان و نام پدیدآور: هزار و یک دستور لینوکس ۱۰۰۱ /Linux commandsفهیمه لطفی، حسین سیلانی؛ ویراستاری نسیم ناصری‌مقدم.
مشخصات نشر: تهران: کیان دانش، ۱۴۰۳.
مشخصات ظاهری: ۷۳۴ ص.
شابک: ۹۷۸-۶۲۲-۴۰۰-۰۱۵-۶
وضعیت فهرست نویسی: فیپا
موضوع: سیستم عامل لینوکس Linux
موضوع: سیستم‌های عامل (کامپیوتر) Operating systems (Computers)
شناسه افزوده: سیلانی، حسین، ۱۳۶۵ -
رده بندی کنگره: QA۷۶/۷۶
رده بندی دیویی: ۰۰۵/۴۳۳
شماره کتابشناسی ملی: ۹۶۷۳۵۹۳
اطلاعات رکورد کتابشناسی: فیپا

نشر کیان دانش: ناشر کتاب‌های دانشگاهی

عنوان: هزار و یک دستور لینوکس
تالیف: فهیمه لطفی، حسین سیلانی
صفحه آرایی و ویراستاری: نسیم ناصری مقدم
چاپ و صحافی: موسسه مهراد
طراحی جلد: امین حیدری
چاپ اول: ۱۴۰۳
شمارگان: ۱۰۰ جلد
شابک: ۹۷۸-۶۲۲-۴۰۰-۰۱۵-

حق چاپ محفوظ است.

خیابان ۱۲ فروردین، خیابان شهدای ژاندارمری، شماره ۱۲۶، طبقه چهارم

فهرست مطالب

مقدمه ۳۲
تکامل لینوکس و سیستم‌عامل‌های محبوب ۳۲
اصطلاحات کلیدی ۳۲
اهمیت لینوکس ۳۲
نرم‌افزار آزاد(Free Software) ۳۲
نرم‌افزار متن‌باز(Open Source) ۳۳
"آزاد و متن‌باز" بودن لینوکس ۳۳
مفاهیم پایه ۳۳
حقوق و مجوزها ۳۴
مفاهیم مرتبط با نرم‌افزار آزاد و متن‌باز ۳۴
مجوزهای نرم‌افزار ۳۵
مجوزهای نرم‌افزاری انعطاف‌پذیر (Permissive Software Licenses) ۳۵
حق اختراع ۳۶
چرا از لینوکس استفاده کنیم یا به لینوکس مهاجرت کنیم؟ ۳۶
فصل اول ۴۰
معرفی لینوکس و ساختار لینوکس ۴۰
تاریخچه سیستم عامل لینوکس ۴۰
پذیرش محبوب و تجاری ۴۱
تکامل کامپیوتر ۴۱
تکامل یونیکس ۴۱
گسترش یونیکس ۴۱
تکامل لینوکس ۴۲
مقایسه توزیع ها ۴۲
ویژگی های توزیع ۴۲
چرا مجوزهای نرم‌افزار آزاد مهم هستند؟ ۴۳
انواع اصلی مجوزهای نرم‌افزار آزاد ۴۳
مراحل بوت در لینوکس ۴۴
عملیات BIOS (POST) ۴۴
بوت لودر(GRUB2) ۴۵
مقداردهی و ارزیابی اولیه هسته ۴۶
اجرای Systemd ۴۸

سیستم فایل _____ ۴۹
تاریخچه سیستم فایل _____ ۴۹
ساختار سیستم فایل _____ ۴۹
سازمان سیستم فایل _____ ۵۰
ساختار فهرست ها _____ ۵۱
بررسی فایل مهم سیستم فایل _____ ۵۳
فهرست ریشه _____ ۵۴
/bin : user binaries _____ ۵۴
/sbin: system binaries _____ ۵۴
/etc:configuration files _____ ۵۴
فهرست‌های موجود در /etc _____ ۵۴
/etc/apt/sources.list _____ ۵۵
/dev: device files _____ ۵۵
/proc: process information _____ ۵۵
/var : variable files _____ ۵۵
/tmp: temporary files _____ ۵۵
/usr: user programs _____ ۵۵
/home : home directories _____ ۵۵
/boot: boot loader files _____ ۵۵
/lib : system libraries _____ ۵۵
/opt: optional add-on applications _____ ۵۵
/mnt: mount directory _____ ۵۶
/media : removable media devices _____ ۵۶
/srv: service data _____ ۵۶
سایر سیستم فایل ها _____ ۵۶
/lost+found _____ ۵۶
/sys _____ ۵۶
/boot/vmlinuz _____ ۵۶
/etc/fstab _____ ۵۶
/etc/default/grub.conf _____ ۵۶
/etc/lilo.conf _____ ۵۶
/etc/init.d _____ ۵۶

/etc/hosts _____ ۵۶
/etc/passwd_____ ۵۶
/etc/printcap _____ ۵۶
/etc/x11 _____ ۵۶
/etc/resolv.conf _____ ۵۶
/usr/share _____ ۵۶
/proc/cpuinfo_____ ۵۷
/proc/filesystems _____ ۵۷
/proc/interrupts _____ ۵۷
/proc/ioports _____ ۵۷
/proc/meminfo_____ ۵۷
/proc/modules _____ ۵۷
/proc/mount _____ ۵۷
/proc/stat _____ ۵۷
/proc/swaps _____ ۵۷
/version _____ ۵۷
/var/log _____ ۵۷
مسیرهای مطلق و نسبی _____ ۵۷
توضیح مفاهیم و فایل‌های بوت _____ ۵۸
فایل‌های اصلی سیستم _____ ۵۸
تنظیمات شبکه _____ ۵۹
تنظیمات بوت و سرویس‌ها _____ ۵۹
فایل میزبان _____ ۵۹
مسیرهای مربوط بهSSH _____ ۵۹
مسیرهای مربوط به مدیریت بسته _____ ۶۰
مسیرهای دیگر _____ ۶۰
مسیرهای مربوط به شبکه _____ ۶۱
مسیرهای مربوط به سیستم فایل _____ ۶۱
مسیرهای مربوط به چاپ _____ ۶۱
مسیرهای مربوط به کرون(Cron) _____ ۶۱
مسیرهای مربوط به سیستم‌عامل _____ ۶۱
سایر مسیرهای مهم _____ ۶۱

تکمیل مسیر در لینوکس ۶۱
انواع سیستم فایل پارتیشن ۶۲
سیستم فایل:Ext2 ۶۲
سیستم فایل:Ext3 ۶۲
سیستم فایل:Ext4 ۶۲
سیستم فایل XFS ۶۲
سیستم فایل Btrfs ۶۳
سیستم فایل ZFS ۶۳
مشاهده نوع سیستم فایل ۶۳
دستورات XFS ۶۳
دستورxfs_info ۶۳
دستورxfs_check ۶۴
دستورxfs_repair ۶۴
دستور xfsdumpوxfsrestore ۶۴
دستور mke2fs ۶۵
ایجاد سیستم فایل Ext2 ۶۵
ایجاد سیستم فایل Ext3 ۶۵
ایجاد سیستم فایل Ext4 ۶۶
تبدیل سیستم فایلها ۶۶
دستور :fsck بررسی و تعمیر سیستم فایل ۶۷
مثال‌های عملی برای دستور fsck با استفاده از گزینه‌های مختلف ۶۸
دستورات مدیریت آرایه‌هایRAID ۶۹
دستورmdadm ۶۹
توضیحات و مثال‌های بیشتر برای گزینه‌های دستورmdadm ۶۹
مفهوم ژورنالینگ ۷۲
ویژگی‌های سیستم فایل ژورنالی ۷۲
سیستم فایل ReiserFS ۷۲
ویژگی‌های سیستم فایل :ReiserFS ۷۲
سیستم فایل XFS ۷۳
ویژگی‌های سیستم فایل :XFS ۷۳
سیستم فایل Btrfs ۷۳
ویژگی‌های سیستم فایل :Btrfs ۷۳

پوسته-شل چیست؟ ۷۴
ترمینال چیست؟ ۷۴
انواع ترمینالها ۷۴
ترمینال Terminator ۷۵
ترمینال Tilda ۷۶
ترمینال Guake ۷۷
ترمینال Yakuake ۷۸
ترمینال ROXTerm ۷۹
ترمینال Eterm ۸۰
ترمینال Rxvt ۸۱
ترمینال Tilix ۸۲
ترمینال LXTerminal ۸۳
ترمینال Konsole ۸۵
ترمینال Kitty ۸۶
ترمینال ST ۸۷
ترمینال Gnome-Terminal ۸۸
ترمینال xfce4-terminal ۸۹
ترمینال Terminology ۹۰
ترمینال Deepin Terminal ۹۱
ترمینال xterm ۹۲
ترمینال LilyTerm ۹۳
ترمینال sakura ۹۴
ترمینال Extraterm ۹۵
ترمینال DomTerm ۹۶
ترمینال TermKit ۹۷
منابع بیشتر درباره ترمینالها ۹۷
کار با ترمینالها ۹۹
بازکردن ترمینال ۹۹
اجزای پنجره ترمینال ۱۰۰
حساسیت دستورات در ترمینال ۱۰۲
تغییر مسیر جاری شل در ترمینال ۱۰۳
میانبرهای ترمینال ۱۰۴

۱۰۵ _____

فصل دوم _____ ۱۰۶

آناتومی دستور _____ ۱۰۶

دستورات داخلی شل(پوسته) _____ ۱۰۷

توابع درون پوسته _____ ۱۰۷

برنامه های اجرایی _____ ۱۰۷

روش تایپ دستورات در ترمینال _____ ۱۰۸

انواع پوسته _____ ۱۰۹

پوسته بورن _____ ۱۱۰

پوسته C Shel _____ ۱۱۰

پوسته Korn _____ ۱۱۱

پوسته GNU Bourne-Again _____ ۱۱۲

پوسته Z _____ ۱۱۲

پوسته ماهی-fish _____ ۱۱۲

روند رشد و تکامل شل های لینوکس _____ ۱۱۳

نمایش شل های نصب شده _____ ۱۱۳

مشاهده شل فعال جاری _____ ۱۱۵

تغییر پوسته پیش فرض _____ ۱۱۸

دستورtype _____ ۱۱۸

دستور which _____ ۱۱۹

فصل سوم _____ ۱۲۲

ترمینال های مجازی _____ ۱۲۲

تاریخچه و مفهوم ترمینال مجازی _____ ۱۲۲

راه اندازی ترمینال مجازی _____ ۱۲۳

پیمایش ترمینالهای مجازی _____ ۱۲۴

دستورchvt _____ ۱۲۵

دستور screen _____ ۱۲۶

فصل چهارم _____ ۱۲۸

تنظیمات bash _____ ۱۲۸

دستور export _____ ۱۲۸

متغییر های محیطی _____ ۱۲۹

تنظیمات محلی و گلوبال _____ ۱۲۹

ایجاد و حذف یک متغییر محیطی لوکال _____ ۱۳۰

لغو یک متغییر محیلی تنظیم شده ۱۳۰
تعریف یک متغییر سیستمی یا گلوبال ۱۳۰
انواع متغیرهای محلی ۱۳۰
دستور set ۱۳۲
دستور env ۱۳۲
دستور unset ۱۳۳
دستور env ۱۳۳
دستور printenv ۱۳۴
دستور alias ۱۳۵
ایجاد نام مستعار ۱۳۵
حذف یک نام مستعار ۱۳۶
دستور unalias ۱۳۷
ایجاد نام مستعار دائمی ۱۳۷
دستور history ۱۳۷
پاک کردن تاریخچه دستورات ۱۳۷
دستور :builtin ۱۳۸
فصل پنجم ۱۴۰
دستورات پایه ای و مقدماتی ۱۴۰
تفاوت بین su وsudo ۱۴۰
دستور visudo ۱۴۰
اعطای دسترسیsudo ۱۴۱
دستورsu ۱۴۱
دستورsudo ۱۴۲
دسترسی کابر ریشه ۱۴۴
اجازه دسترسی رابط کاربری گرافیکی ۱۴۴
دستور pwd ۱۴۴
دستور cd ۱۴۵
دستور clear ۱۴۸
میانبر دستور clear ۱۴۹
دستور sleep ۱۴۹
دستورات پیمایش ۱۴۹
دستور dir ۱۵۹

دستور dirs _____ ۱۶۵
دستور vdir _____ ۱۶۶
مشاهده راهنمای دستورات _____ ۱۶۷
دستور apropos _____ ۱۶۸
یافتن یک دستور باینری _____ ۱۷۳
دستور whatis _____ ۱۷۴
دستور compgen _____ ۱۷۴
دستور TLDR _____ ۱۷۴
دستور man _____ ۱۷۵
دستور ln _____ ۱۷۹
دستور readlink _____ ۱۸۰
دستور touch _____ ۱۸۱
سایر روش های ایجاد فایل _____ ۱۸۲
دستور truncate _____ ۱۸۳
دستور rm _____ ۱۸۳
دستور mkdir _____ ۱۸۴
دستور mktemp _____ ۱۸۴
دستور rmdir _____ ۱۸۵
دستور CP _____ ۱۸۶
دستور Advacned-Copy _____ ۱۸۶
دستور scp _____ ۱۸۷
دستور rsync _____ ۱۸۷
دستور mv _____ ۱۹۰
دستور date _____ ۱۹۰
دستور hwclock _____ ۱۹۱
دستور rename _____ ۱۹۲
دستور file _____ ۱۹۳
دستور stat _____ ۱۹۵
دستور chacl _____ ۱۹۶
دستور cksum _____ ۱۹۷
دستور md5sum _____ ۱۹۸
دستور sha256sum _____ ۲۰۰

دستور sha512sum _____ ۲۰۰
مشاهده محل اجرا دستورات _____ ۲۰۰
فصل ششم _____ ۲۰۳
مدیریت بسته ها _____ ۲۰۳
مدیربسته apt _____ ۲۰۳
پیدا کردن فایل های نصب شده با apt _____ ۲۰۴
جستجو برای بسته با apt _____ ۲۰۴
مشاهده اطلاعات مربوط به بسته با apt _____ ۲۰۵
تأیید بسته برای هر وابستگی ناقص با apt _____ ۲۰۵
نمایش بسته های از دست رفته با apt _____ ۲۰۶
نمایش نسخه بسته نصب شده با apt _____ ۲۰۶
به روز رسانی بسته های سیستم با apt _____ ۲۰۶
ارتقا دادن سیستم با apt _____ ۲۰۶
حذف بسته نصب شده با apt _____ ۲۰۷
پاک کردن مخزن قدیمی بسته های دانلود شده با apt _____ ۲۰۷
حذف بسته ها با فایل های پیکربندی آن با apt _____ ۲۰۷
نصب بسته ها با پسوند deb. با apt _____ ۲۰۸
راهنمای دستورapt _____ ۲۰۸
لیست بسته های نصب شده موجود با apt _____ ۲۰۹
مشاهده نام و جزئیات بسته با apt _____ ۲۰۹
بررسی Cache بسته ها با apt _____ ۲۰۹
ارتقا یک بسته نرم افزاری با apt _____ ۲۰۹
نصب چند بسته همزمان با apt _____ ۲۰۹
ارتقا یک بسته خاص با apt _____ ۲۱۰
نصب یک نسخه خاص با apt _____ ۲۱۰
دانلود یک بسته بدون نصب با apt _____ ۲۱۰
دانلود سورس یک بسته با apt _____ ۲۱۰
مدیربسته dnf _____ ۲۱۰
نحوه عملکردdnf _____ ۲۱۱
نصب مدیر بستهdnf _____ ۲۱۱
نصب بسته و وابستگی های با dnf _____ ۲۱۱
حذف یک بسته و وابستگی ها با dnf _____ ۲۱۱

بروزرسانی یک بسته و وابستگی ها با dnf _____ ۲۱۱
ارتقاء یک بسته و وابستگی ها با dnf _____ ۲۱۱
بررسی به روز رسانی های موجود با dnf _____ ۲۱۲
لیست کردن تمام بسته ها با dnf _____ ۲۱۲
دریافت تنها لیست بسته های نصب شده با dnf _____ ۲۱۲
جستجوی یک بسته در مخازن موجود با dnf _____ ۲۱۲
دریافت لیست بسته های خاص با dnf _____ ۲۱۳
اطلاعات بسته را با دستور dnf دریافت کنید با dnf _____ ۲۱۳
نصب مجدد بسته و وابستگی ها با dnf _____ ۲۱۴
دانلود بسته به جای نصب با dnf _____ ۲۱۴
لیست کردن مخازن فعال فعلی با dnf _____ ۲۱۴
لیست کردن مخازن غیرفعال فعلی با dnf _____ ۲۱۵
لیست کردن تمام مخازن با dnf _____ ۲۱۵
فعال کردن مخزن خاص با dnf _____ ۲۱۶
غیرفعال کردن مخزن خاص با dnf _____ ۲۱۶
پیدا کردن فایل خاص یک بسته با dnf _____ ۲۱۶
پیدا کردن وابستگی های بسته با dnf _____ ۲۱۷
لیست کردن تمام بسته های گروه با dnf _____ ۲۱۷
نصب بسته گروهی با dnf _____ ۲۱۸
حذف بسته گروهی با dnf _____ ۲۱۸
بروزرسانی بسته گروهی با dnf _____ ۲۱۸
مشاهده تاریخچه تمام دستورات dnf اجرا شده _____ ۲۱۸
لغو و انجام مجدد تغییرات با dnf _____ ۲۱۹
لغو تمام تراکنش ها پس از تراکنش مشخص شده با dnf _____ ۲۱۹
پاک کردن تمام حافظه کش مدیربسته با dnf _____ ۲۱۹
ورود به پوستهdnf _____ ۲۱۹
مدیر بسته pacman برای آرچ بیس ها نصب یک یا چند بسته با pacman _____ ۲۱۹
نصب بسته از مخزن خاص با pacman _____ ۲۱۹
حذف یک بسته نرم افزاری با pacman _____ ۲۱۹
حذف یک بسته نرم افزاری با تمام وابستگی ها با pacman _____ ۲۱۹
آپدیت سیستم از مخازن با pacman _____ ۲۱۹
جستجوی یک بسته نرم افزاری در مخازن با pacman _____ ۲۲۰

جستجوی بسته نصب شده روی سیستم عامل با pacman ۲۲۰
لیست فایل های نصب شده به همراه یک بسته با pacman ۲۲۰
مشاهده لیست تمام بسته های نصب شده با pacman ۲۲۰
مشاهده لیست برنامه های نصب شده به همراه برنامه های وابسته با pacman ۲۲۰
دانلود بسته بدون نصب شدن با pacman ۲۲۰
حذف بسته های دانلود شده با pacman ۲۲۰
مدیر بسته dpkg ۲۲۰
توضیحات گزینه‌های اضافی دستورdpkg ۲۲۱
دستور:dpkg-reconfigure ۲۲۳
نصب میزهای کارلینوکسی ۲۲۴
محیط های مختلف دسکتاپ ۲۲۵
میز MATE ۲۲۵
میز Cinnamon ۲۲۶
میز GNOME ۲۲۷
میز پلاسما ۲۲۸
میز UKUI ۲۲۹
میز XFCE ۲۳۰
میز LXDE ۲۳۱
میز LXQt ۲۳۲
فصل هفتم ۲۳۴
شخصی سازی میزهای لینوکس ۲۳۴
سفارشی‌سازی میزگنوم ۲۳۴
دستور gsettings ۲۳۴
تنظیم تصویر پس‌زمینه میز گنوم ۲۳۴
تنظیم نمایش آیکن های میزکار میز گنوم ۲۳۴
سفارشی کردن گفتگوی اعلان ۲۳۴
راه‌اندازی مجدد GNOME Shell ۲۳۵
نصب ابزارهای ضروری ۲۳۵
غیرفعال کردن لیست کاربران ۲۳۵
قفل شدن سیستم و رفتن به حالت لاگین ۲۳۵
غیرفعال‌سازی و متوقف کردن انواع hybrid sleep، sleep، hibernate ۲۳۶
فعال کردن screen saver ۲۳۶

دستور pactl ۲۳۷
غیرفعال کردن بلوتوث ۲۳۷
غیرفعال کردن وب کم ۲۳۷
غیرفعال کردن میکروفن ۲۳۷
نصب درایور تاچ پد ۲۳۷
نصب درایور پرینتر ۲۳۷
نصب درایور اسکنر ۲۳۷
فصل هشتم ۲۳۹
پردازش متون ۲۳۹
دستور nl ۲۳۹
دستور head ۲۴۰
دستور tail ۲۴۲
دستور cat ۲۴۳
مشاهده محتوای یک فایل ۲۴۳
مشاهده محتوای چندین فایل ۲۴۴
مشاهده محتویات یک پرونده قبل با شماره خط ۲۴۴
ارسال محتوای یک پرونده را در پرونده دیگر ۲۴۴
نمایش شماره خط در فایل ۲۴۶
نمایش $ در پایان فایل ۲۴۶
نمایش فاصله بین خطوط جدا شده در فایل ۲۴۶
نمایش محتوای چندین فایل ۲۴۶
استفاده از خروجی استاندارد با اپراتور هدایتگر ۲۴۶
ضمیمه خروجی استاندارد با اپراتور هدایتگر ۲۴۷
انتقال استاندارد ورودی با اپراتور هدایتگر ۲۴۷
مسیر فایل های چندگانه در یک فایل ۲۴۷
مرتب سازی محتویات فایل های چندگانه در یک فایل ۲۴۷
هدایت گرها-Redirect ۲۴۷
دستورaspell ۲۴۸
دستور column ۲۵۱
دستور column ۲۵۳
دستور rev ۲۵۴
دستورSort ۲۵۵

ذخیره خروجی دستور در فایل ۲۵۶
مرتب سازی برای شماره ستون ۲۵۷
حذف موارد تکراری ۲۵۸
مرتب سازی با استفاده از pipe در یک فرمان ۲۵۸
مرتب سازی تصادفی ۲۵۹
دستورcomm ۲۶۰
دستورed ۲۶۲
دستورgrep ۲۶۴
دستور egrep ۲۶۵
دستورexec ۲۶۶
دستور exo-open ۲۶۸
دستورexit ۲۶۹
مقایسه فایل ها ۲۷۰
دستور diff ۲۷۰
ابزار Meld ۲۷۲
دستور meld ۲۷۳
دستورcmp ۲۷۴
دستور command ۲۷۵
دستور wc ۲۷۶
دستور bzcmp ۲۷۷
دستور bzdiff ۲۷۸
دستور uniq ۲۷۹
دستور expand ۲۸۰
دستور tr ۲۸۰
دستور split ۲۸۱
دستور od ۲۸۲
دستور cut ۲۸۳
دستور sed ۲۸۴
دستور tee ۲۸۵
دستور tac ۲۸۶
دستور iconv ۲۸۷
دستور wall ۲۸۷

دستور biff _____ ۲۸۸
دستور agetty _____ ۲۹۰
دستور chroot _____ ۲۹۰
دستور ldd _____ ۲۹۱
دستور ldconfig _____ ۲۹۳
دستور hexdump _____ ۲۹۴
دستور nano _____ ۲۹۸
دستور recode _____ ۲۹۹
دستور echo _____ ۲۹۹
مقادیر environment variables _____ ۳۰۲
دستور printf _____ ۳۰۳
دستور fold _____ ۳۰۴
دستور shuf _____ ۳۰۵
فصل نهم _____ ۳۰۸
مدیریت سیستم فایل،دیسک و پارتیشن ها _____ ۳۰۸
دستور smartctl _____ ۳۰۸
دستور badblocks _____ ۳۰۸
دستور dd _____ ۳۰۹
دستور df _____ ۳۰۹
دستور pydf _____ ۳۱۳
نصب ابزار pydf _____ ۳۱۳
نحوه استفاده از دستور pydf _____ ۳۱۳
دستور du _____ ۳۱۴
دستور ncdu _____ ۳۱۷
دستور mount _____ ۳۱۸
مونت کردن cd/dvd _____ ۳۱۹
دستور dump _____ ۳۲۰
دستور dumpe2fs _____ ۳۲۳
دستور fdisk _____ ۳۲۶
دستور cfdisk _____ ۳۲۹
دستور :xfs_repair تعمیر فایل سیستم XFS _____ ۳۳۳
دستور cryptsetup _____ ۳۳۴

VeraCryptمحافظی قدرتمند برای حریم خصوصی شما ۳۳۵
نحوه نصب VeraCrypt در لینوکس ۳۳۵
دستور quota ۳۳۵
نحوه پاک کردنSwap Space ۳۳۶
دستورmkswap ۳۳۶
دستور parted ۳۳۸
دستورBaobab ۳۳۹
دستور ddrescue ۳۳۹
فصل دهم ۳۴۴
دریافت اطلاعات سخت افزاری در linux ۳۴۴
دستورDmidecode ۳۴۴
دریافت انواعDMI ۳۴۵
دریافت اطلاعات memory ۳۴۵
دریافت اطلاعاتBIOS ۳۴۶
بدست آوردن اطلاعات سیستم ۳۴۶
بدست آوردن اطلاعات سخت افزاری ۳۴۷
بدست آوردن کلاس دستگاهH / W ۳۴۷
بدست آوردن اطلاعاتCPU ۳۴۸
بدست آوردن اطلاعات مربوط به بلوک ها ۳۴۸
بدست آوردن اطلاعات پردازنده ۳۴۹
دستور cpufreq-info ۳۵۰
دستور lscpu ۳۵۲
دستور mpstat ۳۵۳
دستورInxi ۳۵۴
دستورlshw ۳۵۵
دستورhardinfo ۳۵۵
دستور hwinfo ۳۵۷
دستور lshal ۳۵۷
دستورI-Nex ۳۵۷
دستورhdparm ۳۵۸
دستور sdparm ۳۶۱
دستورftrim ۳۶۲

دستور iscsiadm ۳۶۴
دستور nproc ۳۶۶
دستور Cpustat ۳۶۷
دستور info ۳۶۹
بدست آوردن اطلاعات سخت افزاری سیستم ۳۷۱
دستور lsblk ۳۷۲
بدست آوردن دستگاه های بلوک خالی ۳۷۲
دستور blkid ۳۷۶
بدست آوردن اطلاعات سیستم عامل ۳۷۸
بدست آوردن اطلاعات USB ۳۷۸
بدست آوردن اطلاعات دستگاه هایPCI ۳۷۹
بدست آوردن اطلاعات SCSI دستگاه ۳۷۹
بدست آوردن اطلاعات در مورد دستگاه هایSATA ۳۸۰
بدست آوردن اطلاعات geometry ۳۸۰
بدست آوردن اطلاعات سیستم فایل ۳۸۰
بدست آوردن معماری سیستم ۳۸۰
دستور acpid ۳۸۱
دستور acpi_available ۳۸۴
دستور dmesg ۳۸۶
دستور pmap ۳۸۸
دستور addr2line ۳۸۸
دستور free ۳۸۹
فصل یازدهم ۳۹۳
مانیتورینگ در لینوکس ۳۹۳
دستور Vmstat ۳۹۳
دستور ss ۳۹۴
مثال‌های کاربردی برای دستور ss ۳۹۵
دستور iptraf ۳۹۶
مثال‌های کاربردی دستور iptraf ۳۹۷
دستور iostat ۳۹۸
دستور iw ۳۹۹
دستور iwlist ۴۰۰

دستور iwconfig ۴۰۲
دستور iwevent ۴۰۴
دستورchrt ۴۰۵
دستور top ۴۰۸
مرتب سازی فرآیندهای لینوکس را بر اساسPID ۴۰۸
دستورhtop ۴۱۹
نصب htop ۴۱۹
دستورmonit ۴۲۳
دستور iotop ۴۲۳
دستور nice/renice ۴۲۵
دستور ionice ۴۲۵
دستور strace ۴۲۶
دستور Wait ۴۲۷
دستور kill ۴۲۷
دستور nohup ۴۲۸
دستور pstree ۴۲۹
دستورps ۴۳۱
دستور tree ۴۳۳
دستورkillall ۴۳۵
دستور login ۴۳۶
دستورulimit ۴۳۷
مثال‌های کاربردی ۴۳۸
دستورlsof ۴۳۹
دستور fuser ۴۴۴
دستور tuptime ۴۴۵
نصب ابزار tuptime ۴۴۵
دستورWatch ۴۴۷
فصل دوازدهم ۴۵۱
مدیریت سرویس ها و اجرای سطوح در لینوکس ۴۵۱
دستورchkconfig ۴۵۱
مدیر سیستم systemd ۴۵۳
مدیریت سرویس ها ۴۵۳

شروع و توقف سرویس ۴۵۳
راه اندازی مجدد و بارگذاری مجدد سرویس ۴۵۴
فعال و غیرفعال کردن سرویس ۴۵۴
بررسی وضعیت سرویس ۴۵۵
نمای کلی وضعیت سیستم ۴۵۵
لیست واحدهای فعلی ۴۵۶
دستور systemctl ۴۵۶
فهرست کردن همه فایل های واحد ۴۵۷
مدیریت واحد ۴۵۷
نمایش فایل واحد ۴۵۷
نمایش وابستگی ها ۴۵۸
بررسی ویژگی های واحد ۴۵۸
ماسک کردن یک واحد ۴۵۹
ویرایش فایل های واحد ۴۵۹
تنظیم (Runlevel)با target ۴۶۰
دریافت و تنظیم target پیش فرض ۴۶۰
فهرست کردن target موجود ۴۶۰
جداسازی target ۴۶۱
سطوح اجرایی ۴۶۲
تغییر سطح اجرا ۴۶۲
دستور halt ۴۶۳
فصل سیزدهم ۴۶۶
مدیریت زمان و تاریخ ۴۶۶
دستور cal ۴۶۶
دستور hwclock ۴۶۶
دستور tzselect ۴۶۹
دستور locale ۴۷۰
دستور timedatectl ۴۷۱
دستور date ۴۷۳
دستور ntpd ۴۷۴
دستور fc-cache ۴۷۶
دستور fc-list ۴۷۷

دستورcron ۴۸۲
فرمت Crontab ۴۸۳
دستور anacron ۴۸۵
دستورbg ۴۸۶
دستورATQ ۴۸۷
دستور جایگزین برای:atq ۴۸۸
دستور atrm ۴۸۸
دستور جایگزین برای:atrm ۴۸۹
فصل چهاردهم ۴۹۱
مجوزها ۴۹۱
دستورchmod ۴۹۱
دستورchown ۴۹۳
دستور chattr ۴۹۵
دستورchgrp ۴۹۸
دستور getfacl ۵۰۰
دستور setfacl ۵۰۲
دسترسی به حسابroot ۵۰۳
دستورchage ۵۰۵
دستور file ۵۰۸
ماژول های تأیید هویت ۵۱۰
دستور setenforce ۵۱۱
دستور chcon ۵۱۲
۵۱۴
فصل پانزدهم ۵۱۵
مدیریت چندرسانه ای ۵۱۵
دستورamixer ۵۱۵
دستورaplay ۵۲۰
دستورaplaymidi ۵۲۲
دستورffmpeg ۵۲۳
دستور spd- say ۵۲۴
دستور cdrecord ۵۲۸
دستور رایت CD/DVD ۵۲۹
دستور brasero ۵۲۹

دستور mkisofs ۵۳۱
دستور genisoimage ۵۳۳
دستورeject ۵۳۴
فصل شانزدهم ۵۳۹
دستور find ۵۳۹
دستورlocate ۵۴۴
دستور updatedb ۵۴۵
فصل هفدهم ۵۴۸
مدیریت شبکه ۵۴۸
دستور hostname ۵۴۸
ایجاد نام هاست دائمی ۵۴۹
دستور host ۵۴۹
دستورdomainname ۵۵۱
تنظیمات شبکه ۵۵۲
دستور ifconfig ۵۵۲
مشاهده تمام تنظیمات شبکه ۵۵۳
دستور ethtool ۵۵۵
دستور addresses ۵۵۶
دستور ip ۵۵۷
دستورfail2ban ۵۶۲
دستور ufw ۵۶۳
دستور iptables ۵۶۴
دیگر ابزارهای شبکه ۵۶۷
تعریف ip ثابت ۵۶۷
دستورarp ۵۷۰
دستور arp-scan ۵۷۳
دستور traceroute ۵۷۴
دستور tracepath ۵۷۶
دستور mtr ۵۷۸
دستور nmcli ۵۸۱
دستور tcpdump ۵۸۳
دستورWget ۵۸۷

دستور ping _____ ۵۹۱
دستورaria2c _____ ۵۹۶
دستور ss _____ ۵۹۸
دستور sar _____ ۶۰۳
دستور dstat _____ ۶۰۴
دستور nmap _____ ۶۰۶
دستور snort _____ ۶۰۹
دستور nc _____ ۶۱۱
دستور iwconfig _____ ۶۱۳
دستور iwlist _____ ۶۱۴
دستور RFKILL _____ ۶۱۶
دستورdig _____ ۶۱۷
دستور drill _____ ۶۱۸
دستور xdg-open _____ ۶۲۰
دستور whois _____ ۶۲۱
دستور ssh _____ ۶۲۳
دستور sftp _____ ۶۲۵
دستور rcp _____ ۶۲۶
دستور rlogin _____ ۶۲۷
دستورagetty _____ ۶۲۸
فصل هجدهم _____ ۶۳۲
مدیریت پرینتر _____ ۶۳۲
دستور lpinfo _____ ۶۳۲
دستور lpstat _____ ۶۳۳
دستور lpc _____ ۶۳۴
دستور lpadmin _____ ۶۳۵
دستور lp _____ ۶۳۶
دستور LPR _____ ۶۳۷
دستور lpstat _____ ۶۳۸
دستور lprm _____ ۶۳۹
دستور lpq _____ ۶۳۹
دستور cupsctl _____ ۶۴۰

دستور cupsenable _____ ۶۴۲
فصل نوزدهم _____ ۶۴۴
پاک کردن داده ها در linux _____ ۶۴۴
دستور shred _____ ۶۴۴
دستور wipe _____ ۶۴۴
دستور srm _____ ۶۴۶
دستور Sfill-Secure _____ ۶۴۷
دستورsmem _____ ۶۴۷
دستور exiftool _____ ۶۴۸
فصل بیستم _____ ۶۵۱
مدیریت فایلهای آرشیو و پشتیبانی _____ ۶۵۱
مدیریت فایلهای آرشیو دستور tar _____ ۶۵۱
ایجاد آرشیو فایلtar _____ ۶۵۱
ایجاد فایل آرشیو tar.bz2 _____ ۶۵۲
ایجاد فایل آرشیو Untar وtar _____ ۶۵۲
فهرست محتوای فایل tar _____ ۶۵۳
فهرست محتوای فایلtar.gz _____ ۶۵۳
فهرست محتوای فایل tar.bz2 _____ ۶۵۴
بررسی اندازه فایل های آرشیو _____ ۶۵۵
دستور gzip _____ ۶۵۵
دستور zcat _____ ۶۵۶
دستورar _____ ۶۵۷
دستور bzless _____ ۶۶۲
دستورark _____ ۶۶۳
دستور unrar _____ ۶۶۴
استخراج فایل war _____ ۶۶۶
دستور unzip _____ ۶۶۶
دستورbackintime _____ ۶۶۷
دستور djpeg _____ ۶۶۸
دستورxz _____ ۶۶۹
فصل بیست و یکم _____ ۶۷۳
برنامه نویسی _____ ۶۷۳
دستور make _____ ۶۷۳

کامپایل کرنل لینوکس ۶۷۵
مثال‌های کاربردی دستورات make در کامپایل کرنل لینوکس ۶۷۵
دستور cc ۶۷۷
دستور autoconf ۶۷۸
دستور automake ۶۷۹
دستور cc ۶۸۲
دستور g ++ ۶۸۵
بررسی اطلاعات نسخه کامپایلر g++ ۶۸۶
دستور nasm ۶۸۷
دستور java ۶۸۸
دستور python ۶۸۹
دستور pip ۶۹۱
دستور dialog ۶۹۴
دستور expr ۶۹۵
دستور factor ۶۹۶
ادغام چند فایل pdf ۶۹۷
دستور convert ۶۹۷
دستور pdftk ۶۹۷
دستور basename ۶۹۸
دستور dirname ۷۰۰
فصل بیست و دوم ۷۰۴
مدیریت کاربران و گروه ها ۷۰۴
مشاهده کاربران در سیستم ۷۰۴
دستور users ۷۰۴
دستور whomai ۷۰۵
دستور who ۷۰۶
دستور logname ۷۰۷
دستور w ۷۰۸
دستور last ۷۰۹
دستور hostid ۷۱۱
دستور id ۷۱۲
دستور ac ۷۱۴

دستور accton _____ ۷۱۶
دستور LDAP _____ ۷۱۷
دستور ldappasswd _____ ۷۱۸
دستور ldapadd _____ ۷۱۸
دستور ldapdelete _____ ۷۱۸
دستور useradd _____ ۷۱۹
دستور adduser _____ ۷۲۱
دستور passwd _____ ۷۲۲
دستور htpasswd _____ ۷۲۳
دستور usermod _____ ۷۲۵
دستور chfn _____ ۷۲۷
دستور Finger _____ ۷۳۰
دستور lchfn _____ ۷۳۳
دستور Pinky _____ ۷۳۴
دستور groups _____ ۷۳۷
تغییر گروه اولیه کاربر _____ ۷۳۸
افزودن گروه به یک کاربر موجود _____ ۷۳۸
قفل حساب کاربری _____ ۷۳۸
فعال کردن حساب کاربری _____ ۷۳۸
تغییر UID و GID یک کاربر _____ ۷۳۹
دستور groupadd _____ ۷۳۹
دستور groupdel _____ ۷۴۰
دستور groupmod _____ ۷۴۱
دستور chpasswd _____ ۷۴۲
فصل بیست و سوم _____ ۷۴۶
دستورات ممنوعه _____ ۷۴۶
دستور stress _____ ۷۴۶
دستور حذف کامل _____ ۷۴۷
دستور فرمت هارد دیسک _____ ۷۴۷
حذف داده های دیسک با dd _____ ۷۴۷
حذف فهرست home _____ ۷۴۷
فصل بیست و چهارم _____ ۷۴۹

رمزگذاری / رمزگشایی درلینوکس ۷۴۹
دستورCcrypt ۷۴۹
دستور GnuPG ۷۵۰
دستور cryptdir ۷۵۰
دستور flock ۷۵۱
فصل بیست و پنجم ۷۵۴
ماژول‌های کرنل ۷۵۴
دستورlsmod ۷۵۴
دستورmodinfo ۷۵۵
دستورinsmod ۷۵۷
دستورrmmod ۷۵۹
دستورmodprobe ۷۶۰
دستورdepmod ۷۶۱
اضافه کردن پارامترهای کرنل ۷۶۲
دستورgrub-install ۷۶۳
دستورgrub-mkconfig ۷۶۴
initramfsچیست؟ ۷۶۶
چرا initramfs مهم است؟ ۷۶۶
اجزای اصلی:initramfs ۷۶۶
فرآیند بوت شدن با استفاده از:initramfs ۷۶۶
دستورupdate-initramfs ۷۶۶
فصل بیست و ششم ۷۶۹
مدیریت لاگ ها ۷۶۹
لاگ در لینوکس چیست؟ ۷۶۹
دستورات مدیریت لاگ ها ۷۷۳
دستورjournalctl ۷۷۵
نمایش پیام های بوت سیستم ۷۷۸
دستور logrotate ۷۸۰
فصل بیست و هفتم ۷۸۲
دستورات سرگرم کننده درLinux ۷۸۲
دستور orca ۷۸۲
دستور xeyes ۷۸۳
دستور feh ۷۸۴

دستور Cowsay _____ ۷۸۶
دستورxcowsay _____ ۷۸۷
دستور cmatrix_____ ۷۸۸
دستورfire _____ ۷۸۹
دستور sl (لوکوموتیو بخار) _____ ۷۹۱
دستورbanner _____ ۷۹۲
دستور bc _____ ۷۹۳
دستور yes _____ ۷۹۵
دستورgifdiff _____ ۷۹۶
دستورgimp _____ ۷۹۷
دستورlynx _____ ۷۹۸
فصل بیست و هشتم _____ ۸۰۱
عیب یابی _____ ۸۰۱
عیب یابی صدا _____ ۸۰۱
عیب یابیALSA _____ ۸۰۳
دستور lame _____ ۸۰۵
دستور sox _____ ۸۰۶
دستورات عیب یابی _____ ۸۰۷
پیوست _____ ۸۰۹
مقایسه دستورات shell و ms-dos _____ ۸۰۹
Devices _____ ۸۱۱
ارزیابی برنامه های رایج لینوکس و ویندوز _____ ۸۱۱

سخنی از نویسنده

با سلام و احترام به دوست گرامی،

از اینکه این کتاب را مورد توجه خود قرار داده‌اید و در این مسیر همراه من بوده‌اید، صمیمانه سپاسگزارم. امیدوارم این اثر برای علاقه‌مندان به این حوزه مفید و ثمربخش واقع شود. همچنین از شما تقاضا دارم که نواقص و لغزش‌های احتمالی کتاب را نادیده نگیرید و با ارائه نظرات و پیشنهادات ارزشمند خود، در ارتقای کیفیت آن یاری‌ام رسانید. لازم به ذکر است که این کتاب، یکی از صد کتاب لینوکسی من(بسیاری ازآنها درمرحله ترجمه یا نگارش هستند) است که در راستای پروژه‌های متن باز شخصی‌ام و با هدف طراحی و توسعه توزیع‌های لینوکس به نگارش درآمده است.

https://predator-os.ir/
https://predator-os.ir/#Little-Psycho
https://emperor-os.ir/
https://hubuntu.ir

تصمیم دارم تولید محتوای فارسی در زمینه‌های لینوکس و متن باز را در قالب‌های مختلفی مانند کتابچه، کتاب، فلش کارت، صوت و فیلم‌های تصویری در قالب یک سایت ارائه کنم. هدف من از این کار، ایجاد یکی از بزرگترین مرجع‌های آموزش فارسی در این حوزه است.

https://learninghive.ir/

جهت پیگیری آدرس سایت و اطلاع از آخرین اخبار و محتواهای آموزشی، می‌توانید از طریق راه‌های زیر با من در ارتباط باشید:

- Telegram channel: @Linuxtnt
- Telegram ID: @seilany
- Gmail : h.seilany@gmail.com
- https://github.com/hosseinseilani
- https://www.aparat.com/learninghive
- https://www.youtube.com/watch?v=QdvSMBzoNx8

تقدیم به تو

یگانه فرزانه دلم

مقدمه

تکامل لینوکس و سیستم‌عامل‌های محبوب

لینوکس یک سیستم‌عامل آزاد و متن‌باز است که از هسته لینوکس آغاز شد. این هسته توسط لینوس تورووالدز در سال ۱۹۹۱ منتشر شد و به سرعت به یک پروژه بزرگ و جهانی تبدیل شد. جامعه لینوکس از آن زمان تاکنون به رشد و توسعه خود ادامه داده است و امروزه یکی از مهم‌ترین سیستم‌عامل‌های جهان است.

اصطلاحات کلیدی

- **سیستم‌عامل (Operating System):** نرم‌افزاری است که به عنوان واسط بین سخت‌افزار و نرم‌افزارهای کاربردی عمل می‌کند .سیستم‌عامل وظایفی مانند مدیریت منابع سیستم، اجرای برنامه‌ها، و تعامل با کاربر را بر عهده دارد.
- **آزاد و متن‌باز (Free and Open Source):** به این معنی است که کد منبع نرم‌افزار به صورت عمومی در دسترس است و هر کسی می‌تواند آن را مشاهده، تغییر و توزیع کند .این ویژگی باعث شده است که لینوکس به سرعت توسعه یابد و جامعه بزرگی از توسعه‌دهندگان به آن بپیوندند.
- **هسته لینوکس (Linux Kernel):** قلب تپنده سیستم‌عامل لینوکس است .هسته وظیفه مدیریت سخت‌افزار و منابع سیستم را بر عهده دارد.
- **لینوس تورووالدز:** خالق هسته لینوکس است .او در سال ۱۹۹۱ اولین نسخه هسته لینوکس را منتشر کرد.
- **جامعه لینوکس:** به گروه بزرگی از افراد گفته می‌شود که در توسعه، پشتیبانی و استفاده از لینوکس مشارکت دارند .این جامعه شامل برنامه‌نویسان، طراحان، کاربران و شرکت‌های تجاری است.

لینوکس یک سیستم‌عاملی است که کدهای آن به صورت رایگان در دسترس همه قرار دارد .این سیستم‌عامل توسط یک نفر به نام لینوس تورووالدز شروع شد و به مرور زمان با کمک هزاران نفر توسعه پیدا کرد .امروزه لینوکس در کامپیوترهای شخصی، سرورها و حتی دستگاه‌های کوچک مانند تلفن همراه استفاده می‌شود.

اهمیت لینوکس

- **آزاد بودن:** امکان شخصی‌سازی و توسعه نرم‌افزارهای مورد نیاز را فراهم می‌کند.
- **امنیت:** به دلیل متن‌باز بودن، نقاط ضعف آن سریع‌تر شناسایی و برطرف می‌شود.
- **پایداری:** لینوکس به دلیل ساختار محکم و پشتیبانی گسترده، بسیار پایدار است.
- **مقیاس‌پذیری:** از سیستم‌های کوچک تا ابرکامپیوترها قابل استفاده است.

نرم‌افزار آزاد(Free Software)

- **دسترسی به کد منبع:** مهمترین ویژگی نرم‌افزار آزاد، دسترسی آزاد به کد منبع آن است .این بدان معناست که هر کسی می‌تواند کدهای برنامه را مشاهده، مطالعه، تغییر و بهبود بخشد.
- **آزادی در اجرا:** کاربران آزادند که نرم‌افزار را برای هر منظوری اجرا کنند.
- **آزادی در توزیع:** مجاز به کپی، توزیع و فروش نرم‌افزار هستند.
- **آزادی در تغییر:** می‌توانند نرم‌افزار را تغییر داده و نسخه‌های اصلاح شده آن را منتشر کنند.

نرم‌افزار متن‌باز(Open Source)

- **کد منبع عمومی :**مشابه نرم‌افزار آزاد، کد منبع نرم‌افزار متن‌باز نیز به صورت عمومی در دسترس است.
- **همکاری جمعی :**نرم‌افزارهای متن‌باز معمولاً توسط جامعه‌ای از توسعه‌دهندگان به صورت همکاری ایجاد و بهبود می‌یابند.

"آزاد و متن‌باز" بودن لینوکس

- **توسعه سریع :**دسترسی آزاد به کد منبع باعث شده است که تعداد زیادی از توسعه‌دهندگان به بهبود لینوکس کمک کنند و در نتیجه، لینوکس به سرعت توسعه یافته است.
- **نوآوری :**آزادی در تغییر و بهبود نرم‌افزار، زمینه را برای نوآوری و ایجاد ویژگی‌های جدید فراهم کرده است.
- **پایداری :**جامعه بزرگ توسعه‌دهندگان به رفع مشکلات و بهبود امنیت لینوکس کمک می‌کند.
- **انعطاف‌پذیری :**لینوکس می‌تواند برای طیف گسترده‌ای از کاربردها، از سیستم‌های کوچک تا ابرکامپیوترها، استفاده شود.
- **کاهش هزینه :**از آنجایی که لینوکس یک نرم‌افزار آزاد است، هزینه‌های مربوط به خرید مجوز و پشتیبانی آن بسیار پایین است.

مفاهیم پایه

- **Free Software Foundation (FSF):** بنیاد نرم‌افزار آزاد، سازمانی است که برای ترویج و دفاع از نرم‌افزار آزاد تلاش می‌کند.
- **Open Source Initiative (OSI):** ابتکار نرم‌افزار متن‌باز، سازمانی است که تعریف نرم‌افزار متن‌باز را ارائه می‌دهد و مجوزهای متن‌باز را تایید می‌کند.
- **Licenses:** مجوزها قوانینی هستند که شرایط استفاده، تغییر، توزیع و دیگر حقوق مرتبط با نرم‌افزار را تعیین می‌کنند.
- **Contracts:** قراردادها توافقنامه‌های قانونی بین دو یا چند طرف هستند که شرایط همکاری و تعهدات طرفین را مشخص می‌کنند.
- **Public domain software:** نرم‌افزار حوزه عمومی نرم‌افزاری است که هیچ حق مالکیتی بر آن وجود ندارد و هر کسی می‌تواند آن را بدون محدودیت استفاده، تغییر و توزیع کند.
- **Freeware:** نرم‌افزار رایگان نرم‌افزاری است که بدون هزینه قابل استفاده است اما ممکن است محدودیت‌هایی در استفاده، تغییر یا توزیع آن وجود داشته باشد.
- **Shareware:** نرم‌افزار اشتراکی نرم‌افزاری است که معمولاً به صورت رایگان قابل استفاده است اما برای استفاده کامل یا ادامه استفاده، نیاز به پرداخت هزینه دارد.
- **License stewards:** نگهبانان مجوز مسئولیت نظارت بر یک مجوز نرم‌افزاری و حل اختلافات مرتبط با آن را بر عهده دارند.

حقوق و مجوزها

- **Permissions to use, modify and distribute code and software:** مجوزهای استفاده، تغییر و توزیع کد و نرم‌افزار به کاربران اجازه می‌دهند تا این اعمال را تحت شرایط مشخص انجام دهند.
- **Derivative works and code reuse:** آثار مشتق شده و استفاده مجدد از کد به ایجاد نرم‌افزار جدید بر اساس نرم‌افزار موجود اشاره دارد.
- **Closed source / proprietary software:** نرم‌افزار بسته یا اختصاصی نرم‌افزاری است که کد منبع آن مخفی است و معمولاً تحت مالکیت یک شرکت یا فرد قرار دارد.
- **Paid distribution:** توزیع پولی به فروش نرم‌افزار یا خدمات مرتبط با آن با دریافت هزینه اشاره دارد.
- **Modified and unmodified software distribution:** توزیع نرم‌افزار تغییر یافته و بدون تغییر به نحوه توزیع نرم‌افزار با یا بدون تغییرات اشاره دارد.
- **Hosting software as a paid service:** میزبانی نرم‌افزار به عنوان یک سرویس به ارائه دسترسی به نرم‌افزار از طریق اینترنت با دریافت هزینه اشاره دارد.
- **License compatibility:** سازگاری مجوز به توانایی ترکیب نرم‌افزارهای تحت مجوزهای مختلف بدون ایجاد تضاد اشاره دارد.
- **Dual and multi licensing:** مجوز دوگانه و چندگانه به ارائه چندین مجوز برای یک نرم‌افزار اشاره دارد.
- **Conditional licensing:** مجوز شرطی به مجوزی اشاره دارد که شامل شرایط خاصی برای استفاده، تغییر یا توزیع نرم‌افزار است.

مفاهیم مرتبط با نرم‌افزار آزاد و متن‌باز

- **Copyleft:** یک مفهوم حقوقی است که تضمین می‌کند هر کسی که نرم‌افزار را دریافت می‌کند، حق استفاده، مطالعه، تغییر و توزیع آن را دارد. این به معنای حفظ آزادی‌های نرم‌افزار در تمام مراحل است.
- **Distributing:** توزیع به عمل پخش کردن نرم‌افزار به دیگران اشاره دارد، چه به صورت رایگان یا با دریافت هزینه.
- **Conveying:** انتقال نرم‌افزار به دیگران، چه به صورت فیزیکی یا الکترونیکی، را شامل می‌شود.
- **Tivoisierung:** اصطلاحی است که به عمل تغییر نرم‌افزار بدون بازگرداندن تغییرات به جامعه متن‌باز اشاره دارد. این عمل خلاف اصول نرم‌افزار آزاد است.

مجوزهای نرم‌افزار

- **GNU General Public License, version 2.0 (GPLv2):** یک مجوز نرم‌افزار آزاد است که به کاربران اجازه می‌دهد نرم‌افزار را اجرا، مطالعه، تغییر و توزیع کسنند. اما هرگونه تغییر باید تحت همان مجوز منتشر شود.
- **GNU General Public License, version 3.0 (GPLv3):** نسخه جدیدتری از GPL است که برخی تغییرات برای بهبود حفاظت از آزادی‌های کاربران و سازگاری با محیط دیجیتال مدرن ایجاد شده است.
- **GNU Lesser General Public License, Version 2 (LGPLv2):** یک مجوز نرم‌افزار آزاد است که به کاربران اجازه می‌دهد نرم‌افزار را به صورت استاتیک با نرم‌افزارهای دیگر ترکیب کنند بدون اینکه کل ترکیب تحت GPL قرار گیرد.
- **GNU Lesser General Public License, Version 3 (LGPLv3):** نسخه جدیدتری از LGPL است که تغییراتی برای سازگاری با GPLv3 ایجاد شده است.
- **GNU Affero General Public License, Version 3 (AGPLv3):** یک نسخه از GPL است که برای نرم‌افزارهای تحت وب طراحی شده است. این مجوز تضمین می‌کند که تغییرات در نرم‌افزار به صورت آنلاین در دسترس عموم قرار می‌گیرد.
- **Eclipse Public License (EPL), version 1.0:** یک مجوز نرم‌افزار متن‌باز است که به کاربران اجازه می‌دهد نرم‌افزار را استفاده، تغییر و توزیع کنند، اما تغییرات باید تحت همان مجوز منتشر شوند.
- **Eclipse Public License (EPL), version 2.0:** نسخه جدیدتری از EPL است که برخی تغییرات جزئی ایجاد شده است.
- **Mozilla Public Licence (MPL):** یک مجوز نرم‌افزار متن‌باز است که به کاربران اجازه می‌دهد نرم‌افزار را استفاده، تغییر و توزیع کنند، اما تغییرات می‌توانند تحت مجوزهای دیگر نیز منتشر شوند.

مجوزهای نرم‌افزاری انعطاف‌پذیر (Permissive Software Licenses)

مجوزهای نرم‌افزاری انعطاف‌پذیر نوعی از مجوزهای نرم‌افزاری هستند که محدودیت‌های کمی بر روی استفاده، تغییر و توزیع نرم‌افزار اعمال می‌کنند. این مجوزها به توسعه‌دهندگان آزادی زیادی می‌دهند تا با نرم‌افزار کار کنند بدون اینکه الزامات پیچیده‌ای داشته باشند.

- **حقوق اعطایی:** این مجوزها معمولاً حق استفاده، کپی، تغییر، توزیع و حتی فروش نرم‌افزار را به کاربران می‌دهند. در بسیاری از موارد، نیازی به انتشار کد منبع تغییرات ایجاد شده نیست.
- **تعهدات:** تنها تعهد اصلی در این مجوزها معمولاً حفظ اطلاعیه حق کپی‌رایت و متن مجوز در نسخه‌های نرم‌افزار است.
- **ویژگی‌های اصلی:** سادگی، انعطاف‌پذیری و آزادی عمل از ویژگی‌های اصلی این مجوزها هستند. آن‌ها به توسعه‌دهندگان اجازه می‌دهند تا نرم‌افزار را به دلخواه خود استفاده کنند و حتی آن را در پروژه‌های تجاری قرار دهند.
- **سازگاری:** مجوزهای انعطاف‌پذیر معمولاً با سایر مجوزها سازگار هستند، به این معنی که می‌توان نرم‌افزارهای تحت این مجوزها را با نرم‌افزارهای تحت مجوزهای دیگر ترکیب کرد.
- **مثال‌ها:** برخی از معروف‌ترین مجوزهای انعطاف‌پذیر شامل BSD دو کلاسه، BSD سه کلاسه، MIT و Apache 2.۰ هستند. این مجوزها به دلیل سادگی و شرایط آزادانه‌شان، در جامعه نرم‌افزاری بسیار محبوب هستند.

حق اختراع

- **Software patents:** حق اختراع نرم‌افزار به حق انحصاری برای استفاده، ساخت، فروش یا وارد کردن یک اختراع نرم‌افزاری اشاره دارد.
- **Explicit and implicit patent license grants:** اعطای مجوز حق اختراع صریح و ضمنی به اجازه استفاده از یک اختراع نرم‌افزاری به صورت مستقیم یا غیرمستقیم اشاره دارد.

چرا از لینوکس استفاده کنیم یا به لینوکس مهاجرت کنیم؟

یونیکس، این سیستم عامل برای اولین بار توسط دنیس ریچی و کن تامپسون در AT&T از سیستم عامل مالتیکس الگوبرداری شد. یونیکس برای اجرای بزرگترین شبکه جهان، سیستم تلفن AT&T، توسعه داده شد و این کار را نیز انجام داد. سیستم عامل یونیکس به گونه ای طراحی شده است که مقیاس پذیر، قابل اعتماد، ماژولار، ایمن و قابل حمل با قابلیت توسعه شبکه باشد.

توسعه یونیکس در نهایت منجر به استاندارد POSIX شد که بسیاری از معماری لینوکس را تعریف می کند. برخلاف بسیاری از سیستم‌عامل‌های مبتنی بر کنسول رومیزی تک کاربر، لینوکس یک سیستم‌عامل چند کاربره، چند وظیفه‌ای و فعال شبکه است که از هر نقطه‌ای در شبکه قابل دسترسی است. چندین کاربر می توانند از راه دور به یک رایانه لینوکس دسترسی داشته باشند، که هر کدام نمونه دسکتاپ شخصی خود را به طور همزمان اجرا می کنند. سیستم پنجره سازی چند کاربره فعال شبکه لینوکس/یونیکس X-Windows امکان دسترسی کامل از راه دور را فراهم می کند. این تضاد شدید با یک سیستم عامل مبتنی بر کنسول تک کاربر است که کاربر را ملزم به حضور فیزیکی در رایانه برای دسترسی به دسکتاپ و برنامه‌های خود می‌کند و دسترسی به یک کاربر را در یک زمان محدود می‌کند. سیستم عامل لینوکس/یونیکس برای پشتیبانی از دسترسی چند کاربره از راه دور و ایمن با استفاده از ssh طراحی شده است.این به همه مدیران و کاربران لینوکس/یونیکس یک رابط استاندارد از راه دور انعطاف‌پذیر قدرتمند می‌دهد، در حالی که خودرو اغلب ابزار دسترسی از راه دور اولیه سایر مدیران سیستم‌عامل مانند ویندوز است.

تنظیمات شبکه و بسیاری دیگر از پارامترهای MS/Windows برای اعمال شدن نیاز به راه اندازی مجدد دارند. هنگامی که تنظیمات رجیستری MS/Windows اصلاح می شوند نیز این امر صادق است. لینوکس به اندازه کافی ماژولار است که به سرویس خاص (مثلا شبکه) اجازه می دهد بدون خاموش کردن کل رایانه، سرویس مجددا راه اندازی شود. لینوکس همچنین دارای پارامترهای هسته زیادی است که می تواند از روش های مختلفی تنظیم شوند تا تغییرات پویا را در یک هسته در حال اجرا اعمال کنند. این امر زمان کارکرد سیستم لینوکس را تا حد زیادی افزایش می دهد و زمان تلف شده برای راه اندازی مجدد سیستم را حذف می کند.

هسته در قلب سیستم عامل لینوکس/یونیکس قرار دارد. این هسته وظیفه فعال کردن چند وظیفه، چند کاربر، چند رشته ای، چند پردازش، امنیت، رابط با سخت افزار و شبکه را بر عهده دارد. این هسته ای است که لینوس توروالدز بر اساس طرح پوزیکس/یونیکس توسعه داده است که نام لینوکس را به آن داده است. پوسته ها، برنامه های کاربردی کاربر و هر چیز دیگری با این هسته ارتباط دارند. توسعه هسته لینوکس در حال حاضر توسط OSDL.org ، کنسرسیومی از شرکت های نرم افزار و سخت افزار مخابرات و کامپیوتر[1] پشتیبانی می شود. کد منبع تحت مجوز عمومی عمومی گنواست. GPLاین مجوز برای نرم افزار منبع باز منجر به توسعه سریع، اشکال زدایی، استقرار و پذیرش با اجازه دادن و تعریف شرایطی شده است که تحت آن افراد می توانند کد منبع را کپی، اصلاح و به اشتراک بگذارند.

فروشندگان سخت افزار مانند IBM، Hewlett Packard و SGI آینده خود را روی لینوکس شرط بندی کرده اند. Oracle آن را به پلتفرم توسعه خود تبدیل کرده است. استقرار نرم افزارهای سازمانی مانند SAP و Oracle بر روی لینوکس به عنوان پلتفرم اصلی

[1] Open Source Development Labs

تضمین کرده است که لینوکس سنگ بنای زیرساخت محاسباتی شرکتی است. مجوز منبع باز، لینوکس را از شر هرگونه سوء ظن به گسترش انحصار یا ساختارهای مجوز محدود و گران رها کرده است. همچنین ماهیت نرم افزار هسته منبع باز لینوکس این است که نسبت به رقبای خود بدون اشکال و ایمن تر باشد، زیرا توسط توسعه دهندگان در همه جا به جای گروه محدودی از تحلیلگران امنیتی بررسی می شود. لینوکس به صورت تجاری بر روی مجموعه وسیعی از سخت‌افزار از PDAگرفته تا لپ‌تاپ‌ها، ایستگاه‌های wokstation، سرورها، مستقر شده است.

توسعه درآینده و پیشرفت آن تضمین می شود، زیرا کد منبع برای همه در دسترس است. در دانشگاه ها و موسسات سراسر جهان مطالعه می شود. داوطلبان دکترا و محققین شرکتی از لینوکس برای آزمایش مفاهیم پیشرفته محاسباتی استفاده می کنند. توسعه‌دهندگانی که کد منبع را ارسال می‌کنند، می‌دانند که این کد از سوی مجموعه‌ای از توسعه‌دهندگان و بازبینی‌کنندگان در سراسر جهان مورد بررسی قرار می‌گیرد، که آنها را تشویق می‌کند تا کدهای به خوبی نوشته شده ارائه کنند.

لینوکس نسبت به هر سیستم عامل دیگر یک سیستم بسیار قابل اعتماد و امن است. لینوکس و سیستم‌عامل مبتنی بر یونیکس نقص‌های امنیتی کمتری دارند، زیرا کد توسط تعداد زیادی از توسعه‌دهندگان به طور مداوم بررسی می‌شود. و هر کسی به کد منبع آن دسترسی دارد. بنابراین، دلیل امن بودن آن به دلیل منبع باز بودن آن است، به این معنی که می توانید کد منبع آن را ببینید. در نتیجه، باگ های سیستم عامل لینوکس در مقایسه با سایر سیستم عامل ها به سرعت برطرف می شوند.

مدیریت بسته لینوکس بسیار قوی تر از هر سیستم عامل دیگری است. نصب نرم افزار در لینوکس در مقایسه با ویندوز بسیار آسان است. می تواند جریان کار را برای برنامه نویسان به طرز چشمگیری افزایش دهد. این به این دلیل است که لینوکس با مدیران نرم افزار مانند apt، rpm، dpkg، synapticetc عرضه می شود.
در ویندوز، باید وب سایتی را جستجو کنید که می توانید آن را پیدا کنید. فایل exe را دانلود کنید. سپس روی فایل exe کلیک کنید. کلیک کنید، کلیک کنید، کلیک کنید، در نهایت، در بیشتر موارد، باید سیستم را مجددا راه اندازی کنید تا نرم افزار کار کند. در حالی که در لینوکس، این فرآیند بسیار ساده تر است و برای کارکرد نرم افزار نیازی به راه اندازی مجدد ندارد.

بسیاری از ابزارهای برنامه نویسی مفید و یا ابزارهایی با لینوکس از پیش نصب شده اند. به عنوان مثال، grep، wget، cron و غیره. لینوکس همچنین دارای پشتیبانی بومی از SSH است که به مدیریت سریع سرورها کمک می کند.

در لینوکس، کاربران کنترل کاملی بر به روز رسانی سیستم خود دارند. به روز رسانی سیستم در لینوکس بسیار سریعتر است. شما می توانید سیستم را در هر زمان به روز کنید، این خواسته خودتان است. از طرفی ویندوز گاهی اوقات کاربر را مجبور به آپدیت سیستم می کند.
در ویندوز، سیستم خود را روشن می‌کنید، سیستم شروع به به‌روزرسانی می‌کند، درست زمانی که ممکن است لازم باشد یک نامه فوری برای شخصی ایمیل کنید. همچنین، به‌روزرسانی‌های سیستم در ویندوز بسیار کند هستند.
ویندوز همیشه اطلاعات کاربر را جمع آوری می کند. تقریبا همه چیز. لینوکس هیچ بازی پنهان و جستجو ندارد، زیرا همیشه می توانید به کد منبع آن نگاه کنید.
وقتی صحبت از لینوکس به میان می آید، اگر به اندازه کافی از فن آوری آگاه باشید، می توانید به معنای واقعی کلمه همه چیزهایی را که از سیستم شما مخابره می شود، ببینید. حتی اگر اینطور نیستید، ده‌ها هزار چشم وجود دارند که به دنبال کد منبع هستند و به دنبال نقص‌هایی هستند که می‌تواند باعث هر گونه آسیب‌پذیری برای سیستم شود.

با خودکار کردن وظایف، می توان کارآمدتر کار کرد، که زندگی را آسان تر می کند. گاهی اوقات همان کار را بارها و بارها انجام می دهیم که زمان زیادی را صرف می کند. پوسته های Bash، SH، Korn، ZSH، Fish وجود دارد، از هر چیزی که دوست دارید استفاده کنید.

لینوکس یک سیستم عامل قابل حمل است. از طیف وسیعی از معماری های کامپیوتری پشتیبانی می کند. قابلیت حمل به این معنی است که می توانید کد خود را به راحتی از یک معماری سیستم به معماری دیگر منتقل کنید. حال، فرض کنید یک مخزن GitHub دارید که در آن مجموعه ای ازدستورالعمل ها برای نصب تمام نرم افزارهایی که استفاده می کنید فقط در یک فایل وجود دارد، سپس کاری که می توانید انجام دهید این است که فقط آن یک فایل را از هر جایی اجرا کنید تا سیستم شما آماده کار با همه باشد. نرم افزاری که در یک لینوکس کاملاً جدید نصب شده در سیستم خود استفاده می کنید. شاید در عرض ۲۰ دقیقه، سیستم شما آماده شود.

اگر چیزی را در لینوکس دوست ندارید، می توانید آن را حذف کنید یا مطابق با انتخاب خود تغییر دهید. برای مثال، اگر گنوم را دوست ندارید، می توانید آن را به KDE plasma تغییر دهید. در غیر این صورت، می توانید افزونه های زیادی را برای بهبود محیط دسکتاپ خود نصب کنید یا حتی می توانید رابط کاربری سفارشی خود را بسازید. این سطح از آزادی را در هیچ سیستم عامل دیگری نخواهید یافت. حتی می توانید لینوکس خود را بدون هیچ رابط کاربری اجرا کنید. بسیاری از افراد/شرکت ها از لینوکس فقط برای نگهداری سرورها استفاده می کنند، بنابراین تنها چیزی که نیاز دارید فقط یک ترمینال است. سیستم های مجهز به ویندوز نیاز به سخت افزار بیشتری دارند. رایانه های شخصی مجهز به لینوکس بسیار سریعتر از ویندوز هستند. دلیل اصلی اینکه لینوکس یک سیستم سبک وزن است و ویندوز مملو از نرم افزارهای غیر ضروری است. بسیاری از سیستم‌هایی که در پس‌زمینه اجرا می‌شوند، رایانه‌های شخصی ویندوز شما را کند می‌کنند. دلیل دیگر این است که سیستم های فایل در لینوکس بسیار سازماندهی شده اند. فایل‌ها در تکه‌هایی قرار دارند که نزدیک‌تر به یکدیگر هستند، که باعث می‌شود عملیات خواندن و نوشتن بسیار سریع‌تر از ویندوز باشد.

به همین دلیل است که اکثر سیستم های ابری روی لینوکس اجرا می شوند. حتی مایکروسافت لینوکس را برای اجرای Azure استفاده می کند. می‌توانید لینوکس را از سوپر رایانه‌ها تا کنسول‌های بازی، تلویزیون‌های هوشمند، ساعت‌های هوشمند، سیستم‌های سرگرمی خودرو، سیستم‌های سرگرمی پرواز، خودروهای خودران، زیردریایی‌های هسته‌ای و بسیاری دیگر اجرا کنید. ناسا برای انتقال داده ها از ماهواره ها و تلسکوپ ها به شدت به لینوکس متکی است.

فصل اول

فصل اول

معرفی لینوکس و ساختار لینوکس

سیستم عامل لینوکس یک نوع سیستم عامل است که شبیه به Unix است و بر پایه هسته لینوکس ساخته شده است. هسته لینوکس مانند مغز سیستم عامل است، زیرا نحوه مدیریت تعامل کامپیوتر با سخت‌افزارها و منابع خود را کنترل می‌کند. این هسته اطمینان می‌یابد که همه چیز به طور صحیح و بهینه کار می‌کند. اما هسته لینوکس به تنهایی کافی نیست تا یک سیستم عامل کامل را ایجاد کند. برای ایجاد یک سیستم کامل و عملی، هسته لینوکس با مجموعه‌ای از بسته‌های نرم‌افزاری و ابزارها ترکیب می‌شود که به هم Linux distributions گفته می‌شوند. این توزیع‌ها سیستم عامل لینوکس را برای کاربران آماده می‌کنند تا بتوانند برنامه‌های خود را اجرا کرده و وظایف خود را بر روی کامپیوترهای خود به صورت ایمن و موثر انجام دهند. توزیع‌های لینوکس با طعم‌های مختلف آمده و هر کدام برای تأمین نیازها و ترجیحات خاص کاربران طراحی شده‌اند.

لینوکس یک خانواده قدرتمند و انعطاف‌پذیر از سیستم‌های عامل است که استفاده و به اشتراک گذاری آن‌ها آزاد است. این سیستم‌ها توسط شخصی به نام لینوس تورووالدز در سال ۱۹۹۱ ایجاد شدند. چیز جالب این است که هرکسی می‌تواند ببیند چگونه این سیستم کار می‌کند زیرا کد منبع آن برای همه باز است تا بررسی و اصلاح شود. این باز بودن انگیزه میدهد تا افراد از سراسر جهان با یکدیگر همکاری کنند و لینوکس را بهتر و بهتر کنند. از زمان شروع خود، لینوکس به یک سیستم پایدار و ایمن تبدیل شده است که در بسیاری از موارد مختلف مانند کامپیوترها، گوشی‌های هوشمند و ابرکامپیوترهای بزرگ استفاده می‌شود. این به خاطر کارآمد بودنش شناخته شده است، به این معنی که می‌تواند به سرعت بسیاری از وظایف را انجام دهد و همچنین اقتصادی است، به این معنی که هزینه زیادی برای استفاده از آن ندارد. بسیاری از افراد عاشق لینوکس هستند و جزو یک جامعه بزرگ هستند که ایده‌ها را به اشتراک می‌گذارند و به یکدیگر کمک می‌کنند. با پیشرفت فناوری، لینوکس به روز شده و اهمیت خود را در دنیای کامپیوتر حفظ خواهد کرد.

لینوکس ابتدا برای کامپیوترهای شخصی مبتنی بر معماری اینتل x86 طراحی شد، اما از آن زمان به بیشتر محیط‌ها نسبت به سایر سیستم‌عامل‌ها منتقل شده است. شامل اندروید، لینوکس بزرگترین پایگاه نصبی از هر سیستم‌عامل عمومی است به دلیل کنترل اندروید مبتنی بر لینوکس بر روی گوشی‌های هوشمند تا ماه مه ۲۰۲۲. با این حال، لینوکس تا نوامبر ۲۰۲۲ فقط بر روی حدود ۲٫۶٪ از کامپیوترهای دسکتاپ استفاده می‌شود. همچنین، لینوکس بر روی بسیاری از سیستم‌های تعبیه‌شده اجرا می‌شود، به عبارت دیگر، دستگاه‌هایی که سیستم‌عامل آن‌ها به طور معمول در فریمور طراحی می‌شود و بسیاری به سفارش سفارشی شده است. این شامل فضاپیماها (پرسرویرنس روور، کپسول خانواده داراگون و موشک فالکن ۹)، خودروها (تویوتا، هیوندای، مرسدس بنز، آئودی و تسلا)، تلویزیون‌ها (تلویزیون‌های هوشمند ال‌جی و سامسونگ)، کنسول‌های بازی وسیله‌های هوشمند خانگی، کنترل‌های اتوماسیون و روترها را شامل می‌شود. لینوکس یکی از موفق‌ترین نمونه‌های همکاری نرم‌افزار باز و آزاد است. کد منبع می‌تواند توسط همه توزیع، اصلاح و استفاده غیرتجاری یا تجاری تحت شرایط مجوزهای مربوطه خود، مانند مجوز عمومی گنو (GPL)، توزیع شود. به عنوان مثال، هسته لینوکس بر اساس مجوز GPLv2 تأیید شده است.

تاریخچه سیستم عامل لینوکس

سیستم عامل مبتنی بر یونیکس در سال ۱۹۶۹ توسط جو آسانا، داگلاس مک‌الروی، دنیس ریچی و کن تامپسون در آزمایشگاه‌های بل ای‌تی و تی در ایالات متحده اجرا و ابداع شد. یونیکس که برای اولین بار در سال ۱۹۷۱ منتشر شد، کاملاً به زبان مونتاژ نوشته شده بود، همانطور که در آن زمان عملیات عمده‌ای بود. در سال ۱۹۷۳، دنیس ریچی به صورت پیشروانه یونیکس را به زبان سی روزگاری به‌روز کرد. در دسترس بودن یک پیاده‌سازی زبان برنامه‌نویسی بالا برای یونیکس، امکان انتقال آن به پلتفرم‌های کامپیوتری مختلف را فراهم کرد.

تورووالدز در پاییز دهه ۹۰ میلادی در حال بازدید از دانشگاه هلسینکی در یک دوره یونیکس ثبت‌نام کرد. این دوره از یک رایانه کوچک مایکروواکس که اولتریکس را اجرا می‌کرد استفاده می‌کرد و یکی از کتب مورد نیاز کتاب سیستم‌عامل‌ها: طراحی و پیاده‌سازی توسط اندروز

اس. تانبام بود. این کتاب حاوی یک نسخه از سیستم‌عامل مینیکس تانبام بود. این با این دوره بود که توروالدز ابتدا با یونیکس آشنا شد. او در سال ۱۹۹۱ به سیستم‌عامل‌ها علاقه‌مند شد. او از تنگنایی مجوز مینیکس که آن را در آن زمان تنها برای استفاده آموزشی محدود می‌کرد، ناراحت بود و شروع به کار روی هسته سیستم‌عامل خود کرد که در نهایت هسته لینوکس شد.

توروالدز شروع به توسعه هسته لینوکس بر روی مینیکس کرد و نرم‌افزار نوشته شده برای مینیکس هم بر روی لینوکس استفاده شد. بعداً، لینوکس توسعه یافت و سپس توسعه هسته لینوکس بر روی سیستم‌های لینوکس ظاهر شد. همچنین، برنامه‌های GNU تمامی مؤلفه‌های مینیکس را به‌خاطر اینکه از کد آزاد استفاده کردند، جایگزین کردند؛ از طریق پروژه GNU با مجوز GNU GPL، کد می‌تواند در سایر عملکردهای کامپیوتری دوباره استفاده شود تا زمانی که همچنان زیر یک مجوز سازگار یا همان مجوز منتشر شوند. توروالدز شروع به تغییر مجوز واقعی خود کرد که توزیع تجاری را ممنوع می‌کرد، به GNU GPLتوسعه‌دهندگان سعی کردند که با هسته لینوکس، عناصر GNU را توسعه دهند، که سیستم‌عاملی آزاد و کاملاً کارآمد ایجاد می‌کند.

پذیرش محبوب و تجاری

در محیط‌های تولید، پذیرش لینوکس ابتدا در اوایل دهه ۹۰ میلادی در جامعه محاسباتی فوق‌العاده آغاز شد به جای استفاده تنها توسط علاقمندان، جایی که سازمان‌هایی مانند NASA آغاز به جایگزینی ماشین‌های گرانقیمت خود با خوشه‌های کامپیوتری ارزان‌قیمت که لینوکس را اجرا می‌کردند، کردند. استفاده تجاری آغاز شد زمانی که IBM و Dell، پیروی شده از Hewlett-Packard، پشتیبانی لینوکس را برای فرار از انحصار مایکروسافت در بازار سیستم‌عامل‌های رومیزی فراهم کردند.

سیستم‌های لینوکس امروزه کاملاً در محاسبات استفاده می‌شوند، از سیستم‌های تعبیه‌شده تا تقریباً هر سوپرکامپیوتر، و موقعیتی در نصب‌های سروری مانند پشته برنامه‌های معروف LAMP ایجاد کرده‌اند. استفاده از توزیع‌های لینوکس در رومیزی‌های شرکتی و خانگی در حال توسعه است. همچنین، توزیع‌های لینوکس در بازار نت‌بوک معروف شده‌اند، با چندین دستگاه که با توزیع‌های لینوکس سفارشی نصب شده‌اند و گوگل که ChromeOS خود را برای نت‌بوک‌ها منتشر کرده است.

تکامل کامپیوتر

در روزهای اولیه، کامپیوترها به اندازه خانه‌ها یا پارک‌ها بودند. بنابراین می‌توانید تصور کنید چقدر دشوار بود از آنها استفاده کرد. علاوه بر این، هر کامپیوتر سیستم عامل متفاوتی داشت که باعث مشکلات بیشتری در عملکرد آنها می‌شد. هر نرم‌افزار برای یک هدف خاص طراحی شده بود و قادر به اجرا در کامپیوتر دیگر نبود. این بسیار گران‌قیمت بود و افراد عادی نه تنها نمی‌توانستند هزینه آن را بپردازند بلکه نمی‌توانستند آن را هم درک کنند.

تکامل یونیکس

در سال ۱۹۶۹، یک تیم توسعه‌دهندگان از آزمایشگاه‌های بل اقدام به شروع یک پروژه جهت ساخت یک نرم‌افزار مشترک برای تمامی کامپیوترها کردند و آن را 'یونیکس' نامیدند. این سیستم ساده و زیبا بود، از زبانC به جای زبان اسمبلی استفاده می‌کرد و کد آن قابل بازیافت بود. به عنوان یک کد قابل بازیافت، بخشی از کد آن که اکنون به طور رایج 'هسته' نامیده می‌شود برای توسعه سیستم عامل و سایر عملکردها استفاده شد و قابل استفاده در سیستم‌های مختلف بود. همچنین کد منبع آن منبع باز بود.

اولاً، یونیکس تنها در سازمان‌های بزرگی مانند دولت، دانشگاه یا شرکت‌های مالی بزرگ با مینی‌کامپیوترها و میان‌کامپیوترها (کامپیوترهای شخصی کامپیوترهای کوچک میکرو) یافت می‌شد.

گسترش یونیکس

در دهه هشتاد، بسیاری از سازمان‌ها مانند آی‌بی‌ام، اچپی و دوازده شرکت دیگر شروع به ایجاد یونیکس خود کردند. این منجر به یک آشفتگی از گویش‌های یونیکس شد. سپس در سال ۱۹۸۳، ریچارد استالمن پروژه گنو را با هدف ساخت یک سیستم عامل مانند یونیکس

به صورت آزاد و قابل استفاده برای همه توسعه داد. اما پروژه او در جذب محبوبیت شکست خورد. بسیاری از سیستم‌های عامل مانند یونیکس به وجود آمدند اما هیچ‌کدام از آنها نتوانستند محبوبیتی پیدا کنند.

تکامل لینوکس

در سال ۱۹۹۱، لینوس توروالدز، یک دانشجو در دانشگاه هلسینکی، فنلاند، به فکر داشتن یک نسخه تحصیلی از یونیکس قابل دسترسی آزاد افتاد و شروع به نوشتن کد خود کرد. بعداً این پروژه به هسته لینوکس تبدیل شد. او این برنامه را به ویژه برای کامپیوتر ۳۸۶ اینتل خود نوشت زیرا می‌خواست از یونیکس استفاده کند اما نتوانست هزینه آن را بپردازد. او این کار را روی MINIX با استفاده از کامپایلر GNU C انجام داد. کامپایلر GNU C هنوز هم انتخاب اصلی برای کامپایل کد لینوکس است اما کامپایلرهای دیگری هم استفاده می‌شود مانند کامپایلر اینتل سی. او این کار را فقط به عنوان یک سرگرمی شروع کرد اما به یک پروژه بزرگ انجامید. ابتدا می‌خواست آن را به نام 'فریکس' نامگذاری کند اما بعداً آن 'لینوکس' شد. سپس هسته لینوکس را با مجوز خود منتشر کرد و محدود به استفاده تجاری بود. لینوکس از بیشتر ابزارهای خود از نرم‌افزار گنو استفاده می‌کند و تحت حق تکثیر گنو است. در سال ۱۹۹۲، او هسته را تحت مجوز عمومی GNU منتشر کرد.

مقایسه توزیع ها

توزیع‌های[۱] لینوکس: تنها چیزی که در مورد هر توزیع منحصربه‌فرد است، نرم‌افزار مدیریت بسته‌بندی[۲] یا نصب برنامه ها است. مهم است که نسخه های کامپایلر، کتابخانه ها و هسته با هم بدون اشکال کار کنند. این برای بسته های نرم افزاری همراه نیز مهم است. پس از انتخاب نسخه های کامپایلر، کتابخانه و هسته، سایر بسته های نرم افزاری انتخاب، کامپایل و با هم جمع می شوند.

ویژگی های توزیع

تفاوت‌ها هنگام مقایسه توزیع های اوبونتو، SuSE، Debian، Slackware و Red Hat مشهود است. هر کدام یک از توزیع‌ها انتخاب متفاوتی از نسخه‌های مختلف برنامه‌ها در صورت رایج بودن و برنامه‌های نصب و پیکربندی خاص خود،استفاده می‌کنند. توزیع ها اغلب یک طرح فهرست منحصر به فرد و پیکربندی اسکریپت را انتخاب می کنند. به عنوان مثال، Red Hat از سبک اسکریپت و درخت فهرست init UNIX System V استفاده میکند در حالی که Slackware استفاده از طرح بندی سبک برکلی[۳] را انتخاب می کند.تفاوت در انتخاب کتابخانه‌ها و طرح‌بندی فهرست‌ها می‌تواند باعث ناسازگاری با نرم‌افزارهایی شود که برای نصب در یک توزیع طراحی شده اما روی توزیع دیگر نصب می‌شوند.

توزیع لینوکس (اغلب به اختصار توزیع) یک سیستم عامل ساخته شده از مجموعه نرم افزاری است که بر اساس هسته لینوکس و اغلب یک سیستم مدیریت بسته است. یک توزیع معمولی لینوکس شامل یک هسته لینوکس، ابزارها و کتابخانه‌های گنو، نرم‌افزار اضافی، اسناد، یک سیستم پنجره (متداول‌ترین آنها X Window System یا اخیراً Wayland)، یک مدیر پنجره و یک محیط دسکتاپ است. بیشتر نرم افزارهای ارائه شده آزاد و نرم افزار منبع باز هستند که هم به صورت باینری کامپایل شده و هم به صورت کدمنبع[۴] در دسترس هستند و امکان تغییرات در نرم افزار اصلی را فراهم می کنند. معمولاً، توزیع‌های لینوکس به صورت اختیاری شامل برخی از نرم‌افزارهای اختصاصی می‌شوند که ممکن است به صورت کد منبع در دسترس نباشند، مانند حباب‌های باینری مورد نیاز برای برخی از درایورهای دستگاه[۵].

---

۱ distribution
۲ Package manager
۳ Berkeley style layout
۴ Source code
۵ device drivers

تقریباً دو هزار توزیع لینوکس وجود دارد. به دلیل در دسترس بودن زیاد نرم افزارها، توزیع ها اشکال بسیار متنوعی را به خود گرفته‌اند، از جمله مواردی که برای استفاده در رایانه های رومیزی، سرورها، لپ تاپ ها، نت بوک ها، تلفن های همراه و تبلت ها مناسب هستند و همچنین محیط های حداقلی که معمولاً برای استفاده از آنها استفاده می‌شود. در سیستم های تعبیه شده توزیع‌هایی با پشتیبانی تجاری مانند فدورا لینوکس (Red Hat)، (SUSE) openSUSE و اوبونتو (Canonical Ltd.)و توزیع‌های کاملاً مبتنی بر جامعه مانند Debian، Slackware، Gentoo و Arch Linux وجود دارند. اکثر توزیع‌ها آماده استفاده و از پیش کامپایل شده برای یک مجموعه دستورالعمل خاص هستند، در حالی که برخی از توزیع‌ها (مانند جنتو) عمدتاً به صورت کد منبع توزیع می‌شوند و در حین نصب به صورت محلی کامپایل می‌شوند.

چرا مجوزهای نرم‌افزار آزاد مهم هستند؟

- **حفظ آزادی‌های کاربران :**این مجوزها تضمین می‌کنند که کاربران می‌توانند نرم‌افزار را مطالعه، تغییر، توزیع و استفاده کنند.
- **توسعه جامعه :**با ارائه دسترسی آزاد به کد منبع، مجوزهای آزاد امکان همکاری و مشارکت گسترده جامعه توسعه‌دهندگان را فراهم می‌کنند.
- **نوآوری :**آزادی در تغییر و بهبود نرم‌افزار، زمینه را برای نوآوری و ایجاد ویژگی‌های جدید فراهم می‌کند.
- **پایداری :**جامعه بزرگ توسعه‌دهندگان به رفع مشکلات و بهبود امنیت نرم‌افزار کمک می‌کند.

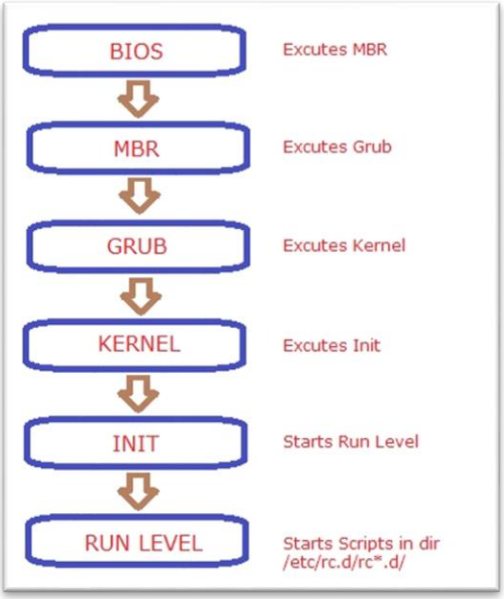
انواع اصلی مجوزهای نرم‌افزار آزاد

- **مجوز عمومی همگانی گنو :(GPL)** یکی از محبوب‌ترین مجوزهای نرم‌افزار آزاد است .این مجوز به کاربران اجازه می‌دهد تا نرم‌افزار را کپی، توزیع، مطالعه و تغییر دهند .همچنین، هر نرم‌افزاری که از کدهای GPL استفاده می‌کند، باید به صورت آزاد منتشر شود.
- **مجوز :BSD** این مجوز نسبت به GPL انعطاف‌پذیرتر است و به کاربران اجازه می‌دهد تا نرم‌افزار را بدون نیاز به انتشار تغییرات خود به صورت آزاد، استفاده کنند.
- **مجوز :MIT** این مجوز نیز بسیار ساده و انعطاف‌پذیر است و به کاربران آزادی کامل در استفاده و تغییر نرم‌افزار می‌دهد.
- **مجوز :Apache** این مجوز به طور گسترده در پروژه‌های نرم‌افزاری متن‌باز استفاده می‌شود و تعادلی بین آزادی‌های کاربران و حقوق مالکیت معنوی ایجاد می‌کند.

مراحل بوت در لینوکس

هر بار که رایانه لینوکس خود را روشن می کنید، یک سری مراحل را طی می کند تا در نهایت صفحه ورود[۱] به سیستم نمایش داده شود که نام کاربری یا رمز عبور شما را درخواست می کند .مرحله مجزا وجود دارد که هر توزیع لینوکس در یک فرآیند بوت[۲] معمولی طی می کند. توجه داشته باشید که بوت لودر[۳] GRUB2در حال حاضر درحال استفاده هستند توسط اکثریت قریب به اتفاق در توزیع های لینوکس مدرن هستند. فرآیند بوت کردن ۴ مرحله زیر را طی می کند که با جزئیات بیشتر در مورد آنها صحبت خواهیم کرد:

- بررسی یکپارچگی **بایوس (POST)**
- بارگیری بوت لودر **(GRUB2)**
- بررسی اولیه هسته[۴]
- شروع **systemd**، والد همه فرآیندها



عملیات BIOS (POST)

فرآیند بوت معمولاً زمانی آغاز می شود که کاربر دکمه روشن شدن سیستم را فشار می دهد. هنگامی که سیستم لینوکس روشن می شود، BIOS سیستم خروجی/ ورودی اولیه یک تست خودکار روشن (POST)را انجام می دهد .این یک بررسی یکپارچگی است که تعداد زیادی از بررسی های تشخیصی را انجام می دهد. عملیات POST prob عملکرد سخت افزار قطعات مانند هارد و یا SSD، صفحه کلید ، رم ، USBپورت، و هر قطعه دیگر از سخت افزا را بررسی میکند. اگر برخی از دستگاه‌های سخت‌افزاری شناسایی نشود، یا اگر نقصی در هر یک از دستگاه‌ها مانند HDD یا SSD خراب باشد، پیام خطایی روی صفحه نمایش داده می‌شود که از شما خواسته می‌شود بررسی کنید. در برخی موارد، صدای بوق را میشنوید و یا درصورت از دست رفتن RAMسیستم ریستارت یا خاموش میشود .با این حال، اگر سخت افزار مورد انتظار وجود داشته باشد و همانطور که انتظار می رود کار کند، فرآیند بوت شدن به مرحله بعدی ادامه می یابد.

---

[۱] Login
[۲] Booting
[۳] Bootloader
[۴] Initialize the kernel

بوت لودر(GRUB2)

هنگامی که سیستم روشن می‌شود، ابتدا BIOS (Basic Input/Output System) فعال می‌شود BIOS .مسئول مدیریت اجزای سخت‌افزاری سیستم است و در این مرحله، وظیفه‌اش این است که MBR (Master Boot Record) را از بخش ابتدایی هارد دیسک یا دیگر دستگاه‌های ذخیره‌سازی مثل USB یا DVD بارگیری کند.

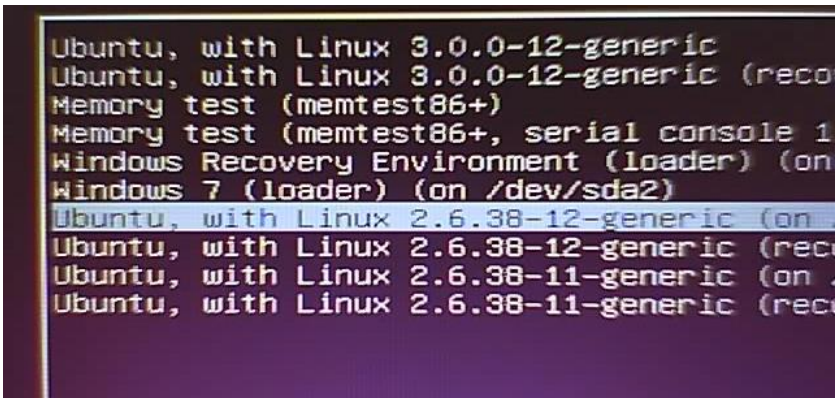
MBRیک بخش کوچک در ابتدای هارد دیسک قرار دارد که شامل یک کد ۵۱۲ بایتی است. این کد، که به عنوان Master Boot Recordشناخته می‌شود، وظیفه اصلی‌اش این است که برنامه Boot Loader را بارگذاری کند Boot Loader .سپس مسئول بارگذاری سیستم عامل اصلی سیستم شما از روی دیسک سخت است.

اگرچه MBR معمولاً بر روی هارد دیسک با مسیر `/dev/sda` یا `/dev/hda` قرار دارد، اما در بعضی موارد، مانند استفاده از دیسک‌های USB یاDVD ، MBRممکن است بر روی این دستگاه‌ها نصب شود. این نکته نشان می‌دهد که MBR نقش حیاتی‌ای در فرآیند بوت سیستم دارد و اهمیت بسیاری در راه‌اندازی صحیح سیستم دارد.

۳ نوع اصلی بوت لودر در لینوکس وجود دارد : LILO وGRUB GRUB2

GRUB2 بوت لودر آخرین و اولیه بوت لودر در توزیعهای مدرن لینوکس است. GRUB2 (GRand Unified Bootloader (version 2یک بوت لودر متن‌باز است که بعد از MBR (Master Boot Record) بارگذاری می‌شود و در فرآیند بوت سیستم عامل نقش اساسی دارد. این بوت لودر از طریق یک منوی تعاملی به کاربر اجازه می‌دهد تا سیستم عامل مورد نظر خود را انتخاب کند و بارگذاری کند. GRUB2قابلیت پشتیبانی از انواع مختلف سیستم‌های عامل را دارد، از جمله انواع مختلف لینوکس، ویندوز، macOS و سیستم‌های دیگر. این بوت لودر قادر است به صورت اتوماتیک سیستم عامل‌های موجود را شناسایی کرده و امکان اجرای آن‌ها را فراهم کند. یکی از ویژگی‌های مهم GRUB2 این است که امکان تنظیمات پیشرفته‌ای را برای کاربران فراهم می‌کند، از جمله تنظیمات بارگذاری سیستم عامل، انتخاب کرنل‌های مختلف، ویرایش فرایند بوت و افزودن پارامترهای خاص بوت. همچنین این بوت لودر امکان بوت از منابع مختلف مثل هارد دیسک‌ها، پارتیشن‌ها، دستگاه‌های USB و شبکه را فراهم می‌کند. به طور کلی، GRUB2یک ابزار بسیار قدرتمند برای مدیریت فرآیند بوت سیستم و انتخاب سیستم عامل مورد نیاز کاربر است که امکانات گسترده‌ای برای تنظیم و کنترل فرآیند بوت سیستم را فراهم می‌کند.

GRUB2مخفف GRand Unified Bootloaderنسخه ۲ است. هنگامی که BIOSبوت لودر grub2 را پیدا کرد، آن را اجرا کرده و در حافظه اصلی (RAM) بارگذاری می کند . منوی GRUB2 اجازه چند چیز را به شما میدهد. این به شما امکان می دهد نسخه هسته لینوکس را که می خواهید استفاده کنید انتخاب کنید .اگر چندسیستم عامل نصب کرده اید، میتوانید یکی از آنهارا انتخاب کنید .علاوه بر این، این امکان را به شما می دهد که برخی از پارامترهای هسته را با فشار دادن ترکیبی از کلیدهای صفحه کلید ویرایش کنید.



انتخاب نسخه هسته لینوکس از بوت منوی گرب

فایل پیکربندی grub2 فایلهای زیرهستند:

```
/boot/grub2/grub2.cfg
/boot/grub/grub.cfg
/etc/grub.d
/etc/default/grub
```

فایل‌های پیکربندی GRUB2 به شرح زیر هستند:

1. /boot/grub2/grub2.cfg: این فایل یکی از اصلی‌ترین فایل‌های پیکربندی GRUB2 است. در این فایل، تنظیمات بارگذاری سیستم عامل، انتخاب کرنل برای بوت، تنظیمات منو و سایر پیکربندی‌های مربوط به فرآیند بوت سیستم ذخیره می‌شوند. این فایل معمولاً توسط سیستم به صورت خودکار تولید و به‌روزرسانی می‌شود.

2. /boot/grub/grub.cfg: این نیز یک فایل پیکربندی مهم برای GRUB2 است. این فایل به عنوان یک نسخه پشتیبان از /boot/grub2/grub2.cfg شناخته می‌شود و ممکن است برای سازگاری با برخی توزیع‌های خاص لینوکس یا تنظیمات دیگر استفاده شود.

3. /etc/grub.d: این دایرکتوری شامل اسکریپت‌های اجرایی است که توسط GRUB2 برای تولید grub.cfg استفاده می‌شوند. این اسکریپت‌ها معمولاً مربوط به انواع مختلف سیستم عامل و کرنل‌ها هستند و به GRUB2 کمک می‌کنند تا منوی بوت را به‌روز کرده و گزینه‌های مختلف بوت را نمایش دهد.

4. /etc/default/grub: این فایل حاوی تنظیمات پیش‌فرض برای GRUB2 است. در این فایل، می‌توانید تنظیماتی مانند زمان بوت پیش‌فرض، تنظیمات رزولوشن و سایر تنظیمات مربوط به رابط کاربری GRUB2 را مشخص

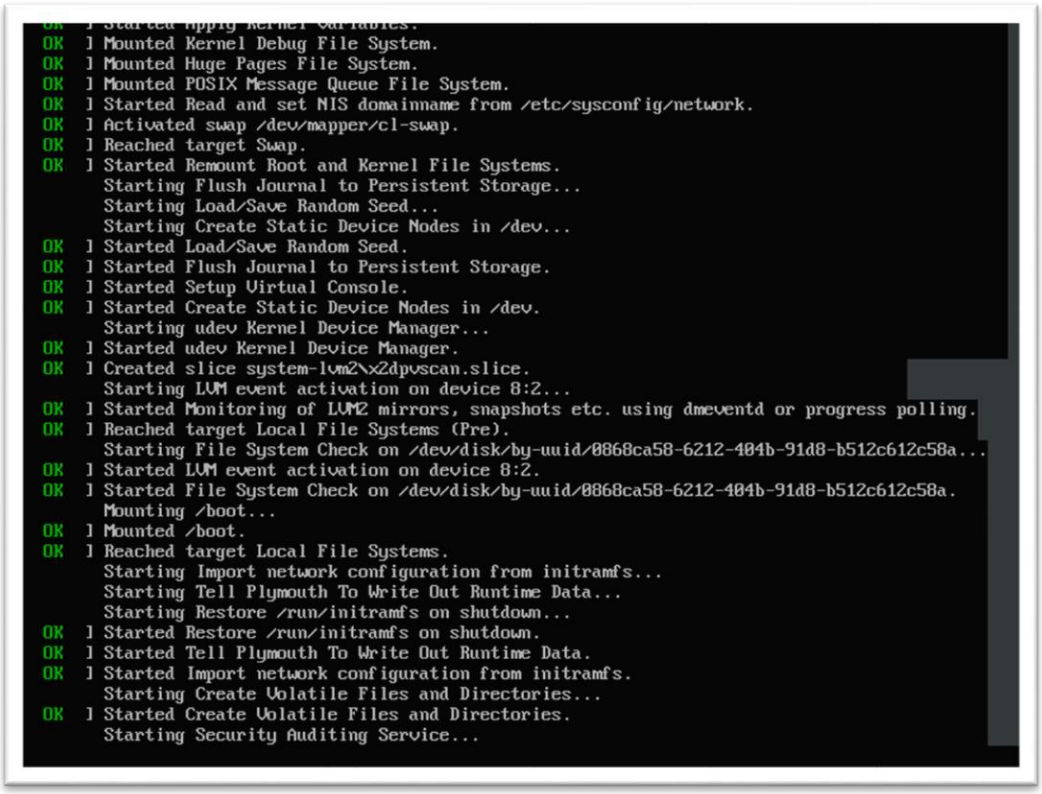
هدف اصلی GRUB بارگذاری هسته لینوکس بر روی حافظه اصلی است.

مقداردهی و ارزیابی اولیه هسته

این ارزیابی اولیه، سخت افزار رایانه شخصی را با فرآیندهای اساسی ارتباط می دهد .هسته ،تمام فرآیندهای سیستم لینوکس شما را کنترل می کند .هنگامی که هسته لینوکس انتخاب شده توسط بوت لودر بارگذاری شود، قبل از انجام هرکاری باید خود از نسخه فشرده استخراج کند .پس از استخراج خود، هسته انتخاب شده سیستم فایل ریشه[۱] را مانت می کند[۲] و برنامه /sbin/init که معمولاً به آن initگفته میشود را مقداردهی اولیه می کند .

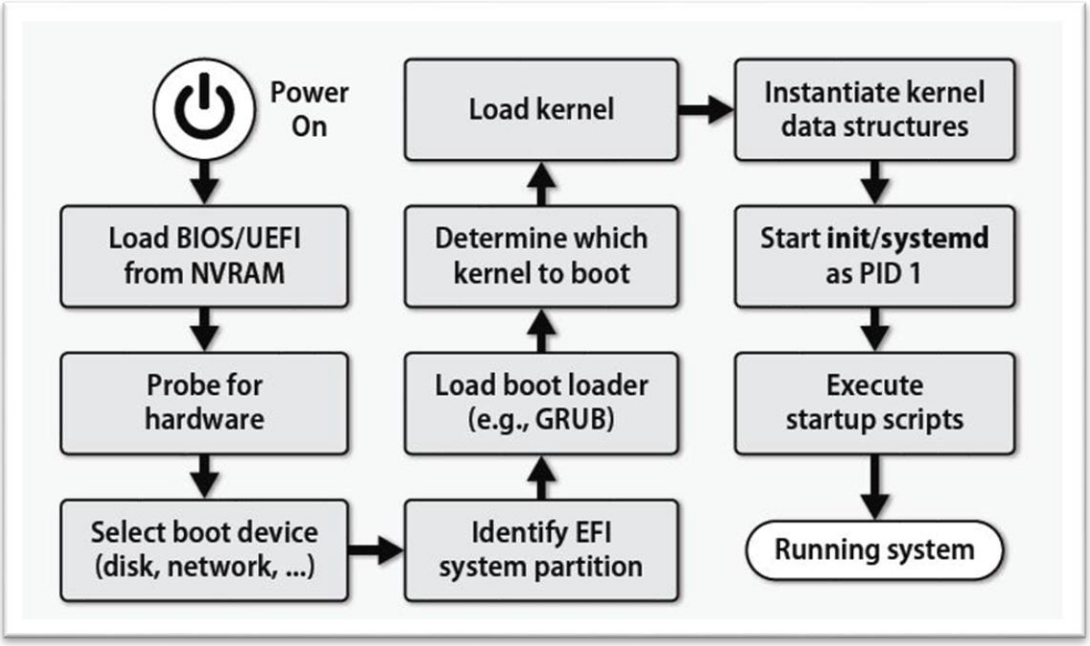
[۱] Root /

[۲] Mount



مراحل بوت اولیه- ارزیابی اولیه بوت

init همیشه اولین برنامه‌ای است که اجرا می‌شود و به آن شناسه فرآیند یا PID اختصاص داده می‌شود. این فرآیند init است که دیمون‌های[۱] مختلفی را ایجاد می‌کند و همه پارتیشن‌هایی را که در فایل /etc/fstab مشخص شده‌اند را مانت می‌کند .سپس هسته، RAMاولیه (initrd) را که یک سیستم فایل ریشه موقت است تا زمانی که سیستم فایل ریشه واقعی مانت شود، مانت می کند .



فرایند بوت سیستم عامل لینوکس

[۱] daemon

Initدر لینوکس نخستین فرایندی است که پس از بوت سیستم شروع به اجرا می‌کند و به عنوان پدر همه فرایندها (PID 1) شناخته می‌شود Init .فرایند اصلی برای مدیریت سیستم و راه‌اندازی سایر فرایندها و سرویس‌ها در سیستم عامل لینوکس است. در سیستم‌های مبتنی برsystemd ، initبه عنوان systemd تعبیه شده است.

Initبه عنوان یک فرایند اصلی سیستم عامل، وظایف اساسی زیر را بر عهده دارد:

۱ .راه‌اندازی سیستم Init :مسئول بارگذاری سایر فرایندها و سرویس‌ها در زمان بوت سیستم است. این فرایند اولین فرایند است که پس از بوت سیستم اجرا می‌شود و سایر فرایندها را به ترتیب مورد نیاز بارگذاری می‌کند.

۲ .مدیریتrunlevel ها Init :در لینوکس از طریقrunlevel هاx یاtarget ها در systemd کنترل می‌کند که کدام سرویس‌ها و فرایندها در هر زمان وضعیت خاصی فعال باشند. این وضعیت‌ها معمولاً توسط فایل‌های تنظیماتی مانند `/etc/inittab` در سیستم‌های initو `/etc/systemd/system/` در سیستم‌های systemd تعریف می‌شوند.

۳ .مدیریت فرایندها و سرویس‌ها Init :مسئول نظارت بر فرایندها، سرویس‌ها و دیمون‌های سیستم است. این شامل راه‌اندازی، توقف، بازنشانی و مانیتورینگ فرایندها و سرویس‌های مختلف در سیستم است.

۴ .مدیریت پیامدهای خطا و عملکرد سیستم Init :نقش مهمی در بررسی خطاها، رفع مشکلات و مدیریت پیامدهای ناشی از خطاها و مشکلات سیستمی دارد. این فرایند مسئول اطلاع‌رسانی به کاربران در مورد وضعیت سیستم و رفع مشکلات آن است.

به طور خلاصه، Initیکی از اهمیتی‌ترین فرایندهای سیستم عامل لینوکس است که نقش اساسی در راه‌اندازی و مدیریت سیستم دارد و بر عملکرد کلی سیستم تأثیرگذار است.

اجرای Systemd

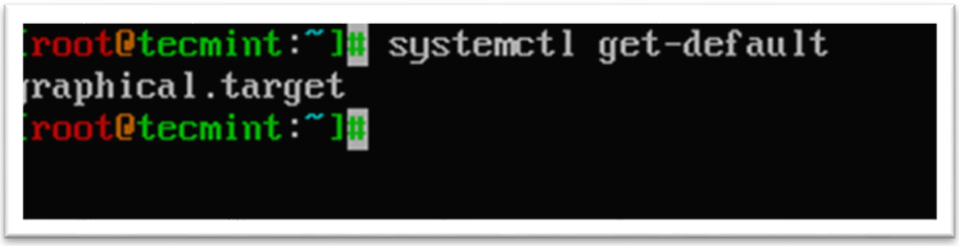
هسته لینوکس در نهایت Systemd را بارگذاری می‌کند که به عنوان یک init system جایگزین SysV init قدیمی می‌شود. Systemd مادر تمام فرآیندهای لینوکس است و علاوه بر موارد دیگر، نصب سیستم‌های فایل، راه‌اندازی و توقف سرویس‌ها را مدیریت می‌کند. برای تعیین وضعیت یا هدفی که سیستم لینوکس باید به آن بوت شود، از systemd با استفاده از فایل `/etc/systemd/system/default.target` استفاده می‌شود. برای یک ایستگاه کاری رومیزی با رابط کاربری گرافیکی، مقدار پیش‌فرض ۵ است که معادل اجرای سطح ۵ برای سیستم‌های قدیمی SysV init است.

در اینجا، سطوح اجرایی سیستم به شرح زیر تعریف شده‌اند:

- `poweroff.target` (runlevel 0) :این سطح سیستم را خاموش می‌کند.
- `rescue.target` (runlevel 1) یا Single-user :این سطح سیستم را به حالت نجات یا single-user mode منتقل می‌کند.
- `multi-user.target` (runlevel 2, 3, 4) :این سطح سیستم را برای استفاده در یک سیستم چند کاربره غیرگرافیکی (کنسول) پیکربندی می‌کند.
- `graphical.target` (runlevel 5) :این سطح سیستم را به گونه‌ای تنظیم می‌کند که از یک رابط گرافیکی چند کاربره با خدمات شبکه استفاده کند.
- `reboot.target` (runlevel 6) :این سطح سیستم را دوباره راه‌اندازی می‌کند.

برای بررسی سطح اجرایی جاری سیستم خود، دستور زیر را اجرا کنید:

```
$ systemctl get-default
```



بررسی Run Level

با اجرای دستور زیر در ترمینال می توانید از یک سطح اجرایی به سطح اجرایی دیگر سوئیچ کنید:

```
$ init runlevel-value
```

سیستم فایل

تاریخچه سیستم فایل

سیستم فایل در لینوکس، همانند سایر سیستم‌های عامل، به منظور مدیریت و ذخیره‌سازی اطلاعات بر روی دیسک‌ها و دستگاه‌های ذخیره‌سازی ایجاد شده است. از جمله ویژگی‌های برجسته سیستم فایل در لینوکس می‌توان به ساختار پویا، پشتیبانی از سطوح دسترسی و انعطاف‌پذیری آن اشاره کرد.

ساختار سیستم فایل

سیستم فایل در لینوکس از چندین بخش تشکیل شده است که هرکدام وظایف خاصی را بر عهده دارند:

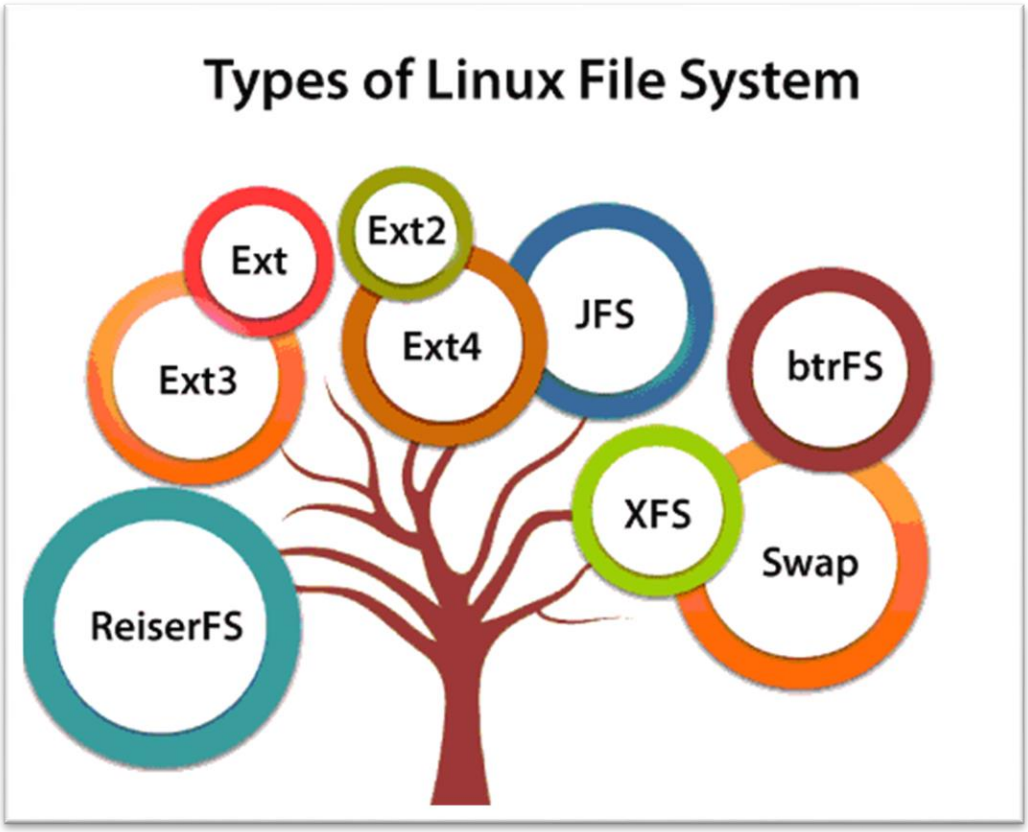
1. **Superblock**: اطلاعات کلی در مورد سیستم فایل از قبیل نوع فایل سیستم، اندازه بلاک‌ها، وضعیت فایل سیستم و ... را در خود ذخیره می‌کند.
2. **Inode Table**: هر فایل یا دایرکتوری در لینوکس متناظر با یک inode است که اطلاعات مربوط به آن را نگهداری می‌کند مانند سطوح دسترسی، اندازه، مکان ذخیره سازی و غیره.
3. **Block Bitmap**: یک نقشه بیتی است که نشان می‌دهد کدام بلاک‌ها در حال استفاده و کدام خالی هستند.
4. **Data Blocks**: بلاک‌های واقعی که اطلاعات فایل‌ها و دایرکتوری‌ها در آن‌ها ذخیره می‌شوند.

قسمت‌ها و انواع سیستم فایل در لینوکس

در لینوکس، انواع مختلفی از سیستم فایل وجود دارند که هرکدام ویژگی‌ها و استفاده‌های خاص خود را دارند، از جمله:

- Ext4: یکی از پرکاربردترین انواع سیستم فایل در لینوکس است که پشتیبانی بالا از اندازه‌های بزرگ فایل و سرعت بالا را دارد.
- XFS: یک سیستم فایل مستقر برای محیطهای سروری است که برای مدیریت فضای بزرگ و پیکربندی‌های پیچیده مناسب است.
- Btrfs: یک سیستم فایل جدید که امکاناتی مانند اسناد تصادفی، فشرده‌سازی، وsnapshot های فایل را فراهم می‌کند.
- ZFS: یک سیستم فایل پیشرفته با امکانات مانندRAID ، شفافیت داده، فشرده‌سازی، وsnapshot های محافظتی.

هر یک از این انواع سیستم فایل ویژگی‌ها و محدودیت‌های خود را دارند و بر اساس نیازهای خاص، انتخاب مناسبی برای استفاده در لینوکس می‌باشند.



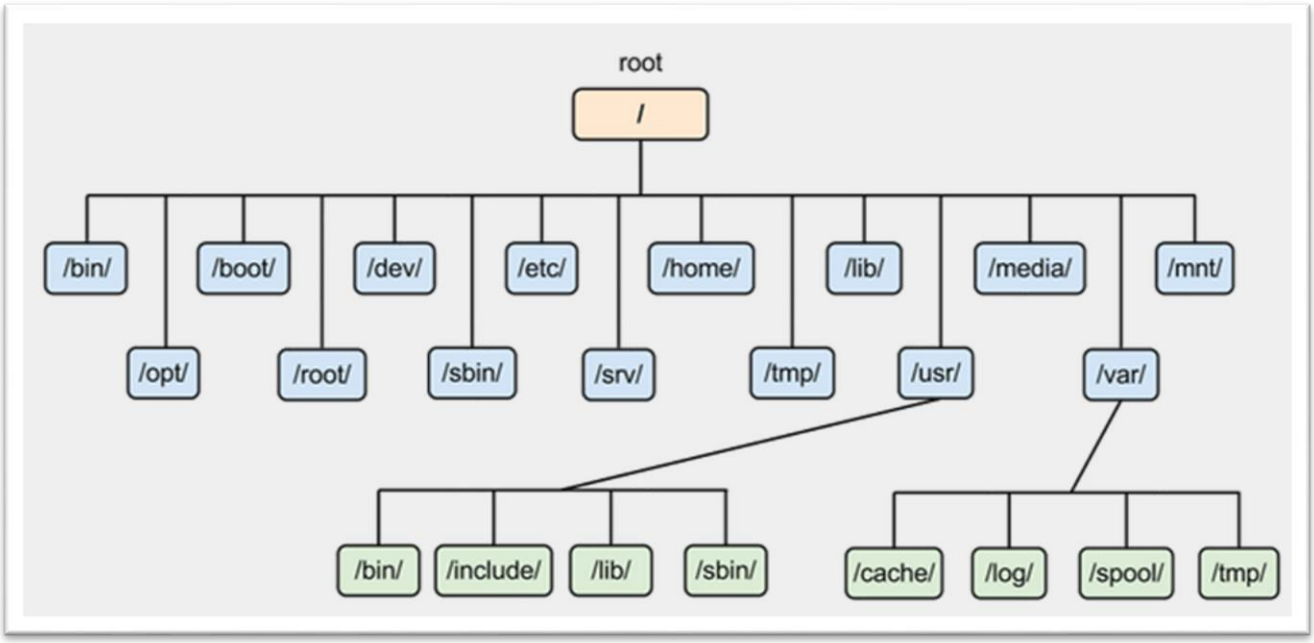
File System Hierarchy

سازمان سیستم فایل

فایل‌ها در یک سیستم لینوکس در ساختار فهرست سلسله مراتبی[۱] قرار می گیرند .این بدان معنی است که آنها در یک الگوی درختی[۲] فهرست (فهرست نامیده شده در سایر سیستم ها) سازماندهی شده اند، که ممکن است حاوی فایل ها و سایر فهرست ها باشد. اولین فهرست در سیستم فایل فهرست ریشه[۳] نامیده می‌شود .فهرست ریشه حاوی فایل ها و زیر شاخه ها[۴] هستند .اکثر محیط های گرافیکی[۵] امروز شامل یک برنامه مدیریت فایل برای مشاهده و دستکاری محتویات سیستم فایل می باشد .اغلب سیستم فایل را به صورت زیرمی بینید:

---

۱ Hierarchical
۲ tree
۳ root
۴ sudirectory
۵ GUI(graphic user interface)



File System Hierarchy

یکی از تفاوت های مهم بین سایرسیستم عامل و سیستم عامل های مشابه یونیکس مانند لینوکس این است که لینوکس مفهوم حروف درایو۱ را استفاده نمی‌کند .در حالی که حروف۲ درایو سیستم فایل را به یک سری درخت۳ های مختلف تقسیم می‌کند (یکی برای هر درایو)، لینوکس همیشه یک درخت تنها دارد .دستگاه های مختلف ذخیره سازی ممکن است شاخه های مختلف درخت داشته باشند، اما همیشه یک درخت تنها وجود دارد. در سیستم عامل لینوکس مفهوم ساختار درختی۴ بیان می شود. زیرا لینوکس برپایه یونیکس بوده و مفهوم درخت از آنجا گرفته شده است و دیگر خبری از درایو نیست.ساختار فایل به صورت درختی وارونه است که ریشه این درخت به نام root است.این ریشه در بالاترین بخش درخت قرار دارد و بقیه بخش ها زیر مجموعه آن می باشد . هرچیزی که فایل نباشد یک پردازش[۵] به شمار می آید. توجه داشته باشید که کرنل لینوکس در تمامی پارتیشن ها توزیع شده و در یک بخش وجود ندارد.

ساختار فهرست ها

فهرست‌ها در لینوکس به صورت سلسله‌مراتبی و ساختاردار سازماندهی می‌شوند. در لینوکس، فهرست‌ها با استفاده از دو نوع عنصر اصلی ساخته می‌شوند:

1 .**فایل‌ها (Files):** در لینوکس، هر چیزی از پوشه‌ها گرفته تا دستگاه‌های ورودی و خروجی به عنوان فایل شناخته می‌شود. فایل‌ها می‌توانند اطلاعات متنی، تصویری، صوتی، برنامه‌ها و غیره باشند.

2 .**پوشه‌ها (Directories):** پوشه‌ها به عنوان محلی برای سازماندهی و ذخیره فایل‌ها در سیستم عامل لینوکس عمل می‌کنند. هر پوشه می‌تواند شامل فایل‌ها و یا زیرپوشه‌های دیگری باشد.

ساختار فهرست‌ها در لینوکس به صورت درختی و سلسله‌مراتبی است، به این صورت که یک پوشه می‌تواند شامل فایل‌ها و/یا زیرپوشه‌های دیگری باشد. این امر امکان می‌دهد که اطلاعات بر روی سیستم به شکل سازماندهی شده‌ای ذخیره شوند و دسترسی به آن‌ها سهل‌تر

---

۱ drive
۲ drive letter
۳ tree structure
4 Trees (Data Structures)
۵process

باشد. هر فهرست در لینوکس یک مسیر (Path) دارد که نشان‌دهنده موقعیت آن در ساختار فهرست سلسله‌مراتبی است. به عنوان مثال، `/home/user/Documents` نشانگر فهرست Documents درون پوشه user درون پوشه home است.

به طور خلاصه، ساختار فهرست در لینوکس با استفاده از فایل‌ها و پوشه‌ها به صورت درختی و سلسله‌مراتبی سازماندهی می‌شود که این ساختار امکان مدیریت و دسترسی آسان به اطلاعات را فراهم می‌کند.

/ bin : همه برنامه های باینری[۱] اجرایی مورد نیاز در هنگام بوت شدن، تعمیر و فایل های مورد نیاز برای اجرا به حالت تک کاربره[۲] و سایر دستورات مهم، یعنی cat، du، df، tar، rpm، wc، history، و غیره در این فهرست است.

/ boot : فایلهای مهمی در طول فرایند بوت شدن، از جمله هسته linux نگهداری می کند.

/ dev : شامل فایل های دستگاه[۳] برای تمام دستگاه های سخت افزاری دستگاه مانند cdrom، cpu و غیره است.

/ etc : شامل فایل های پیکربندی برنامه[۴]، راه اندازی ، خاموش کردن ، شروع ، توقف اسکریپت

/ home : فهرست خانگی کاربران[۵] .هر بار که یک کاربر جدید ایجاد می شود، یک فهرست به نام کاربر در فهرست home ایجاد می شود که حاوی فهرست های دیگر مانند desktop ، download و غیره است.

/ lib : فهرست Lib[۶] شامل ماژول های[۷] هسته[۸] مورد نیاز برای راه اندازی system و اجرای دستورات در سیستم فایل root است.

/ lost + found : این فهرست در هنگام نصب سیستم عامل ایجاد شده است، برای بازیابی فایلهایی که ممکن است به علت خاموش شدن غیر منتظره سیستم خراب شده اند، مفید است.

/ media : فهرست موقت برای دستگاه های قابل حمل[۹] مانند cdrom / media ایجاد می شود.

/ mnt : فهرست موقت برای مونت شدن[۱۰] دستگاه ها به سیستم فایل است.

/ opt : اختیاری به صورت انتخاب مختصر استشامل نرم افزار نرم افزار شخص ثالث VIZ، جاواو غیره

/proc/system مجازی و شبه فایل که حاوی اطلاعاتی در مورد فرآیند در حال اجرا با خاص Process-id aka pid است.

/ root : این فهرست home کاربر root است و هرگز نباید با ' /' اشتباه گرفته شود.

/ run : این فهرست تنها راه حل مناسبی برای مشکل زودهنگام درایو است.

/ sbin : حاوی برنامه های اجرایی باینری، مورد نیاز توسط مدیر system ، برای تعمیر و نگهداری viz.

/ srv : سرویس[۱۱] به صورت " srv " مخفف است.این فهرست حاوی فایل های مربوط به سرویس دهنده و سرویس خاص است.

/ sys : توزیع های[۱۲] مدرن linux شامل یک فهرست sys/ به عنوان یک سیستم فایل مجازی[۱۳] است که اجازه می دهد تا دستگاه های متصل به system را اصلاح کند.

/ tmp : فهرست موقت[۱۴] system، قابل دسترسی توسط کاربر ریشه است و فایل های موقت را تا بوت بعدی برای کاربر و system ذخیره می کند.

---

۱ binary-دودویی
۲ single user
۳ device
۴ configures
۵ home dir
۶ library
۷ modules
۸ kernel
۹ portable devices
۱۰ mount
۱۱ service
۱۲ distro
۱۳ virtual file system
۱۴ temporary

usr / : حاوی مستندات[۱] ، کد منبع[۲] است. معمولاً حاوی بیشترین سهم داده در یک سیستم است. از این رو، این یکی از مهمترین فهرست ها در سیستم است زیرا شامل تمام باینری های کاربر ، اسناد آنها، کتابخانه ها ، پرونده های هدر، و غیره است.

var / :مخفف متغیر[۳] است .محتویات این فهرست همواره افزایش پیدا می کند.این فهرست حاوی پرونده های ورودی ، قفل ها، temp است.

بررسی فایل مهم سیستم فایل

linux یک system پیچیده است که نیاز به یک راه پیچیده تر و کارآمد برای شروع ، متوقف کردن ، نگهداری بر خلاف ویندوز دارد.فایل های پیکربندی[۴] به خوبی برای این امر تعریف شده اند، فایل های باینری ، صفحات man ، فایل های اطلاعات و غیره برای هر پردازش در linux وجود دارد.

/ boot / vmlinuz :فایل کرنل linux است.

/ dev / hda : فایل دستگاه برای اولین IDE HDD هارد دیسک است.

/ dev / hdc : فایل معمولی برای IDE Cdrom است.

/ dev / null : در برخی موارد خروج زباله به dev / null /هدایت می شود، به طوری که برای همیشه از بین می رود.

/ etc / bashrc :حاوی پیش فرض های system و نامهای مستعار استفاده شده توسط shell bash

/ etc / crontab : یک اسکریپت پوسته برای اجرای دستورات مشخص در یک زمان از پیش تعریف شده است.

/ etc / exports : اطلاعات سیستم فایل موجود در شبکه.

/ etc / fstab : اطلاعات دیسک درایو و نقطه اتصال[۵] آنها .

/ etc / group :اطلاعات گروه امنیتی.

/etc/grub.conf :فایل پیکربندی bootloader گراب[۶].

/etc/init.d : سرویس شروع اسکریپت.

/etc/lilo.conf : پرونده پیکربندی lilo bootloader

/ etc / hosts :اطلاعات آدرس ip و نام hostمربوطه.

/etc/hosts.allow : لیستی از hostها مجاز به دسترسی به خدمات در دستگاه[۷] محلی هستند.

/etc/host.deny : لیستی از میزبانها یی که دسترسی به خدمات در دستگاه محلی را رد میکنند.

/ etc / inittab : روند init و تعامل آن در سطوح اجرایی سیستم

/ etc / issue : اجازه می دهد تا پیام پیش از ورود را ویرایش کنید.

/etc/modules.conf :فایل های پیکربندی برای ماژول های system

/ etc / motd : motd /مخفف پیام روز است ، کاربران پیام را پس از ورود به ترمینالهای tty دریافت می کنند.

/ etc / mtab : در حال حاضر بلوک اطلاعات را نصب کرده است.

/ etc / passwd : حاوی رمز عبور از کاربران system در یک فایل سایه، یک پیاده سازی امنیتی است.

/ etc / printcap :اطلاعات چاپگر

/ etc / profile : مقادیر پیش فرض پروفایل ها

/etc/profile.d : اسکریپت کاربردی، اجرا شده پس از ورود.

/etc/rc.d :اطلاعات در مورد اسکریپت مشخصی در زمان اجرا.

---

۱ documents
۲ source code
۳ variable
۴ configure file
۵ mount point
۶ grub
۷ local host

/etc/rc.d/init.d : اسکریپت اولیه راه اندازی سطوح اجرا.
/etc/resolv.conf : سرورهای نام دامنه (DNS) که توسط system استفاده می شود.
/ etc / securetty : فهرست ترمینال، که در ورود به آن امکان پذیر است.
/ etc / skel : محتویات دایرکتوری کاربر جدید را ایجاد و تنظیم می کند.
/ etc / termcap : یک فایل ASCIIکه رفتار ترمینال ، کنسول و چاپگر را تعریف می کند.
/ etc / X11 : فایل های پیکربندیX-window
/ usr / bin :دستورات اجرایی کاربران
/ usr / include :شامل شامل فایل هایی است که توسط برنامه Cاستفاده می شود.
/ usr / share :فهرست فایل های اشتراک گذاشته شده از فایل های man ، فایل های اطلاعات
/ usr / lib : فایلهای کتابخانه برنامه ها
/ usr / sbin : دستورات کاربر برای مدیریت system
/ proc / cpuinfo : اطلاعات CPU
/ proc / filesystems :اطلاعات سیستم فایل جاری رو نگه میدارد.
/ proc / interrupts :درمورد وقفه های فعلی که در حال حاضر مورد استفاده قرار می گیرند.
/ proc / ioports : تمام آدرس های ورودی /خروجی مورد استفاده توسط دستگاه ها در سرور است.
/ proc / meminfo : اطلاعات استفاده از memory
/ proc / modules : ماژول های مورد استفاده هسته.
/ proc / mount : اطلاعات دستگاه های مونت شده
/ proc / stat : آمار دقیقی از وضعیت کنونی سیستم
/ proc / swaps : اطلاعات حافظه موقت swap
/ var / log / lastlog : اطلاعات از آخرین روند بوت شدن سیستم.
/ var / log / messages :اطلاعات پیام های تولید شده daemon syslogدر هنگام بوت شدن.
/ var / log / wtmp : لیست زمان ورود و مدت زمان هر کاربر در system نگهداری میکند.

فهرست ریشه

فهرست ریشه دارای فهرست‌های زیرین زیادی می باشد که به طور کلی آنها را معرفی می کنیم :

/bin : user binaries

در این فهرست بیشتردستورهای مدیریتی کاربر سیستم که قابل دسترسی برای همه کاربران و ازجمله فایل‌های باینری قابل اجرا هستند، قرارمی گیرد.

/sbin :system binaries

این فهرست همانند فهرست قبلی شامل دستورهای مدیریتی کاربر سیستم و هم چنین دستورهای مدیریتی مدیر سیستم را شامل می شود.

/etc:configuration files

این فهرست مثل سیستم عصبی بدن عمل می کند.کلیه تنظیمات سیستم که در قالب فایل وجود دارد دراین فهرست نگهداری و ذخیره می شوند. فهرست‌های زیادی به همراه فایل‌های تنظیمات در این مکان وجود دارد که به طور چکیده به معرفی آنها می پردازیم:

فهرست‌های موجود در /etc

/etc/x11/ : تمامی تنظیمات مربوط به پنجره موسوم به x window که تنظیمات رابط گرافیکی کاربران است در این فهرست ذخیره می شود.

/etc/apt/sources.list

محل ذخیره لیست منابع مربوط به سرورهای بروزرسانی سیستم عامل و مخازن می باشد.

/dev: device files

فهرستی برای نگهداری فایل‌های سخت افزاری و وسیله‌های متصل به سیستم است.

/proc: process information

این فهرست محل نگهداری اطلاعات مربوط به پردازش‌های در حال اجرا و مخفی‌سیستم است.هم چنین اطلاعات منابع سیستم نیزدراین فهرست موجود است.

/var : variable files

این فهرست معمولا جداگانه ایجاد می‌شود و محل نگهداری اطلاعاتی هست که مرتبا درحال تغییر و بروزرسانی هستند و به عبارتی متغییرهستند.

فهرست‌های موجود در این مسیر:

- /var/backups
- /var/cache
- /var/crash
- /var/games
- /var/log
- /var/mail
- /var/spool

/tmp: temporary files

این فهرست برای نگهداری فایل‌های موقت[1] سیستم و کاربران است و معمولا فایل‌های موجود در آن،پس از راه اندازی مجدد[2] سیستم پاک می شوند.

/usr: user programs

دراین فهرست اطلاعاتی مربوط به کتابخانه ها،سندها، راهنماها،سورس کدها، برنامه‌های نصب شده و غیره می باشند.

/home : home directories

این فهرست محل نگهداری تمامی اطلاعات شخصی[3] کاربرانی است که به سیستم عاملloginکرده اند.مثلا

/home/ali

/boot: boot loader files

محل نگهداری فایل‌های بوت سیستم است.فایل‌هایی که سیستم نیازمند آن‌ها برای راه اندازی سیستم است و همچنین فایل‌هایی که در زمان بوت به آنها نیازدارد

/lib : system libraries

محل نگهداری فایل‌های کتابخانه برنامه‌ها است.این فایل ها مثل ماژول‌های کرنل، کتابخانه‌های اشتراکی(مثل کتابخانه‌های زبان c) که نیاز است به همراه سیستم بارگذاری شوند در این محل قرارمی‌گیرند.

/opt: optional add-on applications

محل نگهداری برنامه‌هایadd-on که برنامه‌های کمکی برای دیگر نرم افزارها هستند و همیشه به صورت پیش فرض نصب نیستند.مثل staroffice یا kget.

---

[1]Temporary
[2] Reboot
[3]Privacy

/mnt: mount directory

محل نگهداری نقطه‌های متصل شده[۱] به طورموقت به سیستم مثل فلش دیسک یایک هاردخارجی[۲] است.

/media : removable media devices

این فهرست برای محل نگهداری وسایل جانبی که قابلیت ذخیره سازی[۳]را دارند، مثل فلش دیسک ها، دیسک گردان ها و غیره می باشد.

/srv: service data

محل نگهداری سرویس‌های که مربوط به سرورها می باشد و معمولا در پس زمینه[۴] در حال اجرا هستند.

سایر سیستم فایل ها

/lost+found

برای هرپارتیشن در سیستم یک فهرست برای آن درفهرستlost+found ساخته می شود. در صورتی که اطلاعات اصلی دراین مسیر موجود باشد عمل بازیابی صورت گرفته و سیستم اجرا می شود و در غیر این صورت کاربر باید برنامه ویا سیستم خود را دوباره نصب و تنظیم نمایید.

/sys

در توزیع‌های جدید لینوکس از این فهرست برای نگهداری سیستم فایل مجازی استفاد می شود.

/boot/vmlinuz

محلی برای نگهداری فایل‌های‌کرنل لینوکس

/etc/fstab

محل نگهداری هارد دیسک‌هایی که به سیستم متصل شده اند، قرار دارد.

/etc/default/grub.conf

محل نگداری فایل تنظیمات راه انداز بوت سیستم :grub

/etc/lilo.conf

محل نگداری فایل تنظیمات راه انداز بوت سیستم :lilo

/etc/init.d

محل نگداری اسکریپت‌هایstart/stop سرویس ها

/etc/hosts

محل نگهداری آدرس‌های های‌ipهاست‌های مجازی و دیگرآدرس‌هایip

/etc/passwd

محل نگهداری پسوردهای کاربران سیستم

/etc/printcap

محل ذخیره اطلاعت مربوط به پرینترهای موجود برروی سیستم عامل

/etc/x11

محل ذخیره تنظیمات پنجره‌هایx . تنظیمات رابط‌های گرافیکی سیستم

/etc/resolv.conf

محل ذخیره تنظیمات شبکه مربوط به دامین ها و سرور.(DNS) Domain Name Servers

/usr/share

محل ذخیره فهرست‌های اشتراک گذاشته شده در میان سیستم ها

---

[۱] Mount

[۲] External

[۳] Removable Devices

[۴] Background

/proc/cpuinfo
محل ذخیره اطلاعت مربوط به CPU
/proc/filesystems
محل ذخیره اطلاعات مربوط به سیستم فایل که در حال حاضر استفاده می شود.
/proc/interrupts
محل ذخیره اطلاعات درباره رخ دادهای سیستم
/proc/ioports
محل ذخیره تمام اطلاعات آدرس پورت‌های ورودی و خروجی که توسط سیستم استفاده می شوند.
/proc/meminfo
محلی برای ذخیره میزان حافظه استفاده شده توسط سیستم
/proc/modules
محل ذخیره ماژول‌های کرنل
/proc/mount
محلی برای ذخیره اطلاعات مربوط به وسایل و رسانه‌های متصل شده به سیستم
/proc/stat
محل ذخیره چکیده ای از وضعیت سیستم در قالب آمار
/proc/swaps
محل ذخیره اطلاعات مربوط به حافظه مجازی سیستم
/version
محلی برای نگهداری اطلاعات مربوط به نسخه لینوکس
/var/log
محل ذخیره رخ دادها و وقایع سیستم

مسیرهای مطلق و نسبی

درک تفاوت بین این دو نوع مسیر برای ناوبری صحیح در سیستم فایل واقعاً حیاتی است.

مسیر مطلق: یک مسیر مطلق از فهرست ریشه (‘/’) شروع می‌شود و مسیر کامل به یک مکان خاص در سیستم فایل را فراهم می‌کند. به عنوان مثال، /home/sssit مکان دقیقی را نشان می‌دهد که از فهرست ریشه شروع می‌شود.

مسیر نسبی: از سوی دیگر، یک مسیر نسبی نسبت به فهرست کاری فعلی تعریف می‌شود. این نوع مسیر با خط (‘/’) شروع نمی‌شود و مسیری نسبتی به مکان فعلی شما در سیستم فایل مشخص می‌کند. به عنوان مثال، اگر در /home/sssit باشید، مسیر نسبی به فهرست sssit به سادگی sssit خواهد بود. مهم است که توجه داشته باشید که هنگام استفاده از مسیرهای مطلق، باید مسیر کامل از فهرست ریشه مشخص شود. از سوی دیگر، مسیرهای نسبی به موقعیت فعلی شما در سیستم فایل وابسته هستند. در مثال‌های ارائه شده، برای ناوبری به فهرست home از /home/sssit نیاز به استفاده از یک مسیر مطلق (cd /homeداریم، زیرا home یک زیردایرکتور از sssit نیست. به‌طور معکوس، دسترسی به فهرست sssit می‌تواند از طریق یک مسیر نسبی (cd sssitاز فهرست فعلی صورت گیرد. اگر قبلاً در فهرست ریشه هستید، هر دو home و /home شما را به همان مقصد (فهرست homeهدایت خواهند کرد زیرا از فهرست ریشه شروع می‌کنید. درک صحیح از استفاده از مسیرهای مطلق و نسبی در لینوکس برای ناوبری بهینه سیستم فایل و اجرای دقیق دستورات بسیار حیاتی است.

توضیح مفاهیم و فایل‌های بوت

معمولاً در محیط‌های بوت شدن سیستم‌ها، به خصوص در سیستم‌های مبتنی بر لینوکس و سیستم‌های بوت چندگانه، استفاده می‌شوند . هر یک از آن‌ها نقش خاصی را در فرایند بوت شدن سیستم ایفا می‌کنند.

bootloaderها و فایل های پیکربندی

- **pxelinux: ،extlinux ،syslinux**
- این‌ها نام‌های مختلفی برای یک نوع bootloader هستند که به طور گسترده برای بوت کردن سیستم‌ها از روی فلش مموری، هارد دیسک یا شبکه (PXE) استفاده می‌شود.
- **syslinux:** یکی از قدیمی‌ترین و شناخته‌شده‌ترین bootloader ها است.
- **extlinux:** نسخه توسعه‌یافته‌تر syslinux است که از ویژگی‌های بیشتری پشتیبانی می‌کند.
- **pxelinux:** نسخه مخصوص شبکه syslinux (PXE) است که برای بوت کردن سیستم‌ها از روی شبکه استفاده می‌شود.
- **فایل پیکربندی** `syslinux.cfg` **:** یا فایل‌های مشابه در سایر bootloader ها (حاوی پیکربندی‌های بوت است .در این فایل، گزینه‌های مختلفی مانند نام منو، گزینه‌های بوت، مسیر هسته و initrd تعریف می‌شود.

- **isolinux.cfg: ،isolinux.bin**
  - این فایل‌ها معمولاً در محیط‌های ISO مانند Live CD ها (یافت می‌شوند.
  - **isolinux.bin:** فایل اجرایی bootloader است که مسئول بارگذاری هسته و سایر فایل‌های مورد نیاز است.
  - **isolinux.cfg:** فایل پیکربندی bootloader است که گزینه‌های بوت را تعریف می‌کند.
  - **وظیفه :** این فایل‌ها برای بوت کردن سیستم از روی یک محیط ISO استفاده می‌شوند.
- **isohdpfx.bin:**
  - این فایل به عنوان یک پیش‌بارگذاری برای سیستم‌های قدیمی‌تر استفاده می‌شود و به bootloader اصلی کمک می‌کند تا محیط بوت را آماده کند.

فایل های بوت UEFI

- **efiboot.img:**
  - این فایل یک تصویر از محیط بوت UEFI است و حاوی اطلاعات لازم برای بوت کردن سیستم‌های مبتنی بر UEFIاست.
- **uefi/shim.efi:**
  - فایل shim.efi یک لایه سازگاری است که به سیستم‌های UEFI اجازه می‌دهد تا bootloader های قدیمی‌تر) مانند GRUB) را اجرا کنند.
- **uefi/grubx64.efi:**
  - این فایل bootloader GRUB برای سیستم‌های ۶۴ بیتی مبتنی بر UEFI است GRUB) GRand Unified Bootloader) . یک bootloader محبوب است که برای بوت کردن سیستم‌های لینوکس و سایر سیستم عامل‌ها استفاده می‌شود.

فایل های اصلی سیستم

- **/etc/nologin:**
  - این فایل حاوی یک پیام است که هنگام تلاش برای ورود به سیستم به کاربر نمایش داده می‌شود .معمولاً زمانی استفاده می‌شود که بخواهیم دسترسی به سیستم را به طور موقت یا دائمی مسدود کنیم.
- **/etc/passwd:**

- این فایل اطلاعات اولیه کاربران سیستم را شامل می‌شود، مانند نام کاربری، UID(شناسه عددی کاربر)، گروه اصلی، دایرکتوری خانگی و پوسته پیش‌فرض.

- **/etc/shadow:**
  - این فایل حاوی اطلاعات رمزنگاری شده گذرواژه‌های کاربران است .به دلیل حساس بودن اطلاعات موجود در این فایل، دسترسی به آن معمولاً محدود به کاربران روت است.

تنظیمات شبکه

- **/etc/xinetd.d/:**
  - این دایرکتوری حاوی فایل‌های پیکربندی برای سرویس‌های شبکه‌ای است که توسط xinetd مدیریت می‌شوند . xinetdیک سرور اینترنت کوچک است که به صورت پیش‌فرض در بسیاری از توزیع‌های لینوکس نصب می‌شود.
- **/etc/xinetd.conf:**
  - این فایل پیکربندی اصلی xinetd است و تنظیمات کلی سرویس‌ها را مشخص می‌کند.
- **/etc/hosts.allow:**
  - این فایل لیستی از سرویس‌ها و میزبان‌هایی را مشخص می‌کند که به آن‌ها اجازه دسترسی داده می‌شود.
- **/etc/hosts.deny:**
  - این فایل لیستی از سرویس‌ها و میزبان‌هایی را مشخص می‌کند که به آن‌ها اجازه دسترسی داده نمی‌شود.

تنظیمات بوت و سرویس‌ها

- **/systemd.socket:**
  - این فایل واحد سیستم برای سوکت‌های ارتباطی استفاده می‌شود .سوکت‌ها نقاط پایانی ارتباطی بین فرآیندها هستند.
- **/etc/inittab:**
  - این فایل در سیستم‌های مبتنی بر initd برای پیکربندی فرآیند init استفاده می‌شود .init فرآیندی است که در هنگام بوت شدن سیستم اجرا می‌شود و سایر فرآیندها را راه‌اندازی می‌کند.
- **/etc/init.d/:**
  - این دایرکتوری حاوی اسکریپت‌های شروع و توقف سرویس‌ها است .این اسکریپت‌ها برای راه‌اندازی، توقف و مجدد راه‌اندازی سرویس‌ها استفاده می‌شوند.

فایل میزبان

- **/etc/hosts:**
  - این فایل یک پایگاه داده محلی از نام میزبان‌ها و آدرس‌های IP است .سیستم از این فایل برای ترجمه نام میزبان به آدرس IP استفاده می‌کند.

مسیرهای مربوط به SSH

- **id_rsa.pub: و ~/.ssh/id_rsa**
  - این فایل‌ها جفت کلید پیش‌فرض برای احراز هویت SSH هستند.
    - id_rsaکلید خصوصی است که باید آن را مخفی نگه دارید.
    - id_rsa.pubکلید عمومی است که باید آن را به سروری که می‌خواهید به آن وصل شوید اضافه کنید.
- **id_dsa.pub و: ... و ~/.ssh/id_dsa**
  - این فایل‌ها جفت کلیدهای جایگزین با الگوریتم‌های مختلف(DSA ، ECDSA ، Ed25519)هستند که می‌توانید با دستور ssh-keygenآن‌ها را ایجاد کنید.
- **~/.ssh/authorized_keys:**

- این فایل در سرور قرار دارد و کلیدهای عمومی کاربران مجاز برای اتصال با SSH بدون رمز عبور را ذخیره می‌کند .شما باید کلید عمومی خود را به این فایل در سرور مقصد اضافه کنید.

- **ssh_host_rsa_key.pub:و/etc/ssh/ssh_host_rsa_key**
  - این فایل‌ها جفت کلید میزبان سرور برای احراز هویت سرور SSH هستند.
    - کلید خصوصی (ssh_host_rsa_key) در سرور باقی می‌ماند.
    - کلید عمومی (ssh_host_rsa_key.pub) معمولاً همراه با اطلاعات سرور توزیع می‌شود تا به کاربران در تأیید هویت سرور کمک کند.

مسیرهای مربوط به مدیریت بسته

- **(/etc/apt/sources.list برای سیستم‌های مبتنی بر Debian و:Ubuntu)**
  - این فایل لیستی از مخازن نرم‌افزاری را مشخص می‌کند که سیستم هنگام نصب یا به‌روزرسانی بسته‌ها از آن‌ها استفاده می‌کند.
- **/etc/yum.conf و) /etc/yum.repos.d/ برای سیستم‌های مبتنی بر Red Hat و:Fedora)**
  - این فایل‌ها تنظیمات مربوط به مدیریت بسته با Yum را مشخص می‌کنند .فایل yum.conf تنظیمات کلی و فایل‌های داخل yum.repos.d مخازن نرم‌افزاری را تعریف می‌کنند.

مسیرهای دیگر

- **/etc/ld.so.conf:**
  - این فایل لیستی از دایرکتوری‌هایی را مشخص می‌کند که سیستم هنگام بارگذاری کتابخانه‌های اشتراکی (shared libraries)به آن‌ها مراجعه می‌کند.
- **/etc/updatedb.conf:**
  - این فایل تنظیمات مربوط به به‌روزرسانی پایگاه داده موقعیت فایل‌ها (locate database) را مشخص می‌کند.
- **/etc/profile:و/etc/bash.bashrc**
  - این فایل‌ها اسکریپت‌هایی هستند که هنگام ورود کاربران به سیستم با پوسته Bash اجرا می‌شوند و تنظیمات کلی محیط کاربر را تعریف می‌کنند.
- **.bash_profile/و ... (فایل‌های تنظیمات پوسته در دایرکتوری خانگی کاربر):**
  - این فایل‌ها اسکریپت‌هایی هستند که هنگام ورود کاربر به سیستم اجرا می‌شوند و تنظیمات شخصی محیط کاربر را تعریف می‌کنند .هر کدام از این فایل‌ها با پوسته‌های مختلفی مرتبط هستند) مانند bashrc برای پوسته Bash).
- **/etc/X11/xorg.conf/و:...**
  - این فایل‌ها تنظیمات مربوط به سیستم X Window که رابط گرافیکی را مدیریت می‌کند، تعریف می‌کنند.
- **/etc/localtime:و/etc/timezone**
  - این فایل‌ها منطقه زمانی سیستم را مشخص می‌کنند.
- **/etc/logrotate.d/:و/etc/logrotate.conf**
  - این فایل‌ها تنظیمات مربوط به چرخش لاگ (log rotation) را مشخص می‌کنند .چرخش لاگ به معنای فشرده‌سازی یا حذف لاگ‌های قدیمی به صورت خودکار است.

مسیرهای مربوط به شبکه

- **/etc/network/interfaces:**(در سیستم‌های قدیمی‌تر) این فایل برای پیکربندی کارت‌های شبکه استفاده می‌شد .در سیستم‌های جدیدتر، معمولاً از ابزارهای دیگری مانند NetworkManager استفاده می‌شود.
- **/etc/sysctl.conf:**این فایل تنظیمات هسته لینوکس را شامل می‌شود و می‌تواند برای تنظیم پارامترهای شبکه مانند اندازه بافر، تعداد اتصالات همزمان و ... استفاده شود.

مسیرهای مربوط به سیستم فایل

- **/etc/fstab:**این فایل حاوی اطلاعاتی در مورد سیستم فایل‌ها است، از جمله پارتیشن‌ها، دستگاه‌های بلوکی، نقاط نصب و گزینه‌های مانت کردن.
- **/etc/mtab:**این فایل یک نسخه زنده از فایل fstab است و اطلاعاتی درباره سیستم فایل‌هایی که در حال حاضر مانت شده‌اند را نشان می‌دهد.

مسیرهای مربوط به چاپ

- **/etc/printcap:**این فایل شامل اطلاعاتی درباره چاپگرها و صف‌های چاپ است (در سیستم‌های قدیمی‌تر).
- **/etc/cups/printers.conf:**در سیستم‌های جدیدتر، پیکربندی چاپگرها معمولاً در این فایل و سایر فایل‌های موجود در دایرکتوری cups انجام می‌شود.

مسیرهای مربوط به کرون(Cron)

- **/etc/crontab:**این فایل حاوی وظایفی است که باید به صورت دوره‌ای اجرا شوند.
- **/etc/cron.d/:**این دایرکتوری حاوی فایل‌های اضافی برای تنظیم وظایف کرون است.

مسیرهای مربوط به سیستم‌عامل

- **/etc/passwd:** (قبلاً توضیح داده شده است
- **/etc/group:**این فایل اطلاعات گروه‌های کاربری را شامل می‌شود.
- **/var/log:**این دایرکتوری برای ذخیره لاگ‌های سیستم استفاده می‌شود.
- **/var/run:**این دایرکتوری حاوی فایل‌های سوکت و سایر فایل‌های موقتی است که توسط فرآیندهای در حال اجرا استفاده می‌شود.

سایر مسیرهای مهم

- **/etc/hosts:** (قبلاً توضیح داده شده است
- **/etc/resolv.conf:**این فایل تنظیمات مربوط به سرورهای DNS را مشخص می‌کند.
- **/etc/default/locale:**این فایل زبان و تنظیمات منطقه‌ای پیش‌فرض سیستم را مشخص می‌کند.
- **/etc/security/limits.conf:**این فایل برای تنظیم محدودیت‌های منابع برای کاربران و گروه‌ها استفاده می‌شود.

تکمیل مسیر در لینوکس

تکمیل مسیر یک قابلیت بسیار مفید در سیستم عامل لینوکس است که به شما کمک می‌کند در وارد کردن دستورات و نام‌های فایل‌ها سریع‌تر و دقیق‌تر عمل کنید. با استفاده از این ویژگی، هنگامی که شما شروع به تایپ یک مسیر یا نام فایل می‌کنید و سپس کلید ‘Tab’ را فشار دهید، سیستم به صورت خودکار تلاش می‌کند مسیر یا نام فایل را تکمیل کند یا گزینه‌های موجود را به شما نشان دهد. با این ویژگی، شما نیاز نیست همه چیز را به صورت کامل تایپ کنید، بلکه می‌توانید با تایپ کردن قسمتی از مسیر یا نام فایل و سپس استفاده از کلید ‘Tab’، به سرعت و دقت بیشتری دسترسی پیدا کنید. این ویژگی بسیار مفید است زمانی که با نام‌های بلند یا پیچیده فایل‌ها سر و کار دارید و یا نیاز به تکمیل خودکار مسیرهای طولانی دارید.

انواع سیستم فایل پارتیشن

سیستم فایل:Ext2

-**تاریخچه:** سیستم فایل Ext2 در سال ۱۹۹۳ معرفی شد و توسط Remy Card توسعه یافت.

-استفاده در توزیع‌های Linux: Ext2 اولین سیستم فایل پیش‌فرض در برخی از توزیع‌ها مانند RedHat و Debian بود.

-محدودیت‌ها Ext2: برای غلبه بر محدودیت‌های سیستم فایل Ext استفاده می‌شد.

-حداکثر اندازه فایل: حداکثر اندازه فایل قابل ذخیره در سیستم فایلExt2 ، ۱۶ گیگابایت است.

-کاربردهای معمول Ext2: برای رسانه‌های ذخیره‌سازی معمولی مانند درایو USB فلش، کارت SD و سایر رسانه‌های ذخیره‌سازی از این نوع استفاده می‌شود.

Ext2یکی از سیستم‌های فایل ابتدایی در دنیای Linux بود که در زمان ارائه‌اش از استقبال بالایی برخوردار بود. این سیستم فایل، با توجه به محدودیت‌های خود، برای کاربردهای خاصی مانند رسانه‌های ذخیره‌سازی کوچک و محیط‌هایی که نیاز به فضای ذخیره‌سازی محدود دارند، مناسب بود.

سیستم فایل:Ext3

-**تاریخچه:** سیستم فایل Ext3 در سال ۲۰۰۱ معرفی شد و در Kernel 2.4.15 با ویژگی journaling یکپارچه شده است.

-ویژگی Journaling: اضافه شدن ویژگی journaling به Ext3 باعث بهبود قابلیت اطمینان و حذف نیاز به بررسی سیستم فایل پس از خاموش شدن می‌شود.

-حداکثر اندازه فایل: حداکثر اندازه فایل قابل ذخیره در سیستم فایلExt3 ، از ۱۶ گیگابایت تا ۲ ترابایت متغیر است.

-امکان ارتقا Ext3: امکان ارتقا از سیستم فایل Ext2 به Ext3 را بدون نیاز به پشتیبان‌گیری و بازگرداندن داده‌ها فراهم می‌کند.

سیستم فایل Ext3 با اضافه شدن ویژگیjournaling ، قابلیت اطمینان و پایداری بیشتری نسبت به Ext2 را ارائه می‌دهد. این سیستم فایل امکان ارتقا از نسخه‌های قدیمی‌تر Ext2 به Ext3 را به صورت آسان فراهم کرده و از دست دادن داده‌ها در این فرایند جلوگیری می‌کند. از آنجایی که Ext3 قابلیت journaling دارد، نیاز به بررسی سیستم فایل پس از خاموش شدن کاهش می‌یابد که این امر باعث بهبود عملکرد و پایداری سیستم می‌شود.

سیستم فایل:Ext4

تکامل از Ext4 :Ext3 به عنوان پیشگام Ext3 در اکتبر ۲۰۰۸ به عنوان کد پایدار در Kernel 2.6.28 ادغام شد.

حداکثر اندازه فایل Ext4: از ۱۶ گیگابایت تا ۱۶ ترابایت برای حداکثر اندازه فایل پشتیبانی می‌کند.

غیرفعال کردن Ext4 :Journaling امکان غیرفعال کردن ویژگی journaling را نیز فراهم می‌کند.

ویژگی‌های دیگر: از ویژگی‌های Ext4 می‌توان به مقیاس‌پذیری زیر فهرست، اختصاص چند بلاک، تخصیص تاخیر، FSCKسریع و سایر ویژگی‌های پیشرفته اشاره کرد.

سیستم فایل Ext4 به عنوان یک تکامل ازExt3 ، امکانات پیشرفته‌تری را ارائه می‌دهد. با داشتن حداکثر اندازه فایل قابل ذخیره‌سازی تا ۱۶ ترابایت، Ext4برای مواردی که نیاز به فضای ذخیره‌سازی بزرگتر دارند، مناسب است. امکان غیرفعال کردنjournaling ، امکانات مقیاس‌پذیری و اختصاص چند بلاک، همراه با سرعت بالا در) FSCK بررسی و تعمیر خطاهای سیستم فایل(، Ext4را به یک گزینه پیشرفته برای سیستم‌های فایل در محیط‌های پربار و پیچیده تبدیل کرده است.

سیستم فایل XFS

XFS - یک سیستم فایل مستقر برای محیط‌های سروری است که برای مدیریت فضای بزرگ و پیکربندی‌های پیچیده مناسب است.

سیستم فایل Btrfs

Btrfs - یک سیستم فایل جدید است که امکاناتی مانند اسناد تصادفی، فشرده‌سازی، وsnapshot های فایل را فراهم می‌کند.

سیستم فایل ZFS

ZFS - یک سیستم فایل پیشرفته است که امکاناتی مانندRAID ، شفافیت داده، فشرده‌سازی، وsnapshot های محافظتی را دارا است.

مشاهده نوع سیستم فایل

برای مشاهده نوع سیستم فایل linux، دستور زیر را در ترمینال به عنوان یک کاربر root اجرا کنید.

```
# df -hT | grep “^ / dev”
```

خروجی دستور:

```
/ dev / sda3 ext3 /
/ dev / sda1 ext3 / boot
```

یا

```
# df -hT | grep “dev”
```

دستورات XFS

سیستم فایل XFS (XFS Filesystem) یکی از سیستم‌های فایل قدرتمند و پرکاربرد در لینوکس است که برای مدیریت حجم‌های بسیار بزرگ داده طراحی شده است .برای مدیریت و نگهداری از سیستم‌های فایلXFS ، از مجموعه‌ای از دستورات استفاده می‌شود که در ادامه به توضیح هر یک از آن‌ها می‌پردازیم:

دستورxfs_info

شرح :

:این دستور اطلاعات جامعی درباره سیستم فایلXFS ، از جمله اندازه، تعداد اینودها، بلوک‌های آزاد و استفاده شده، و سایر جزئیات را نمایش می‌دهد.

ساختار دستور:

```
xfs_info /dev/device
```

- :/dev/deviceدستگاه بلوکی حاوی سیستم فایلXFS

کاربردها:

- نمایش اطلاعات کلی درباره سیستم فایل
- بررسی سلامت سیستم فایل
- مقایسه دو سیستم فایل

مثال:

```
xfs_info /dev/sda1
```

دستور xfs_check

شرح :این دستور برای بررسی یکپارچگی سیستم فایل XFS استفاده می‌شود .خطاهای احتمالی را شناسایی کرده و گاهی اوقات می‌تواند آن‌ها را تعمیر کند.

ساختار دستور:

```
xfs_check /dev/device
```

کاربردها:

- بررسی و تعمیر خطاهای سیستم فایل
- اطمینان از سلامت سیستم فایل قبل از عملیات مهم

مثال:

```
xfs_check -v /dev/sdb2
```

- `-v`: حالت مفصل (Verbose) را فعال می‌کند.

دستور xfs_repair

شرح :این دستور برای تعمیر سیستم فایل‌های XFS آسیب دیده استفاده می‌شود .در مواردی که `xfs_check` نتواند خطاها را تعمیر کند، از این دستور استفاده می‌شود.

ساختار دستور:

```
xfs_repair /dev/device
```

کاربردها:

- تعمیر سیستم فایل‌های آسیب دیده
- بازیابی داده‌های از دست رفته (در برخی موارد)

مثال:

```
xfs_repair -L /dev/sda3
```

- `-L`: تعمیر آنلاین (Online repair) را انجام می‌دهد.

دستور xfsdump وxfsrestore

شرح :این دو دستور برای ایجاد بک‌آپ و بازیابی سیستم فایل‌های XFS استفاده می‌شوند.

- **xfsdump:** برای ایجاد بک‌آپ از یک سیستم فایل XFS استفاده می‌شود.
- **xfsrestore:** برای بازیابی یک سیستم فایل XFS از بک‌آپ ایجاد شده استفاده می‌شود.

ساختار دستور:

```
xfsdump -v -f /backup/mybackup.tar /dev/sda1
xfsrestore -v -f /backup/mybackup.tar /mnt
```

کاربردها:

- ایجاد بک‌آپ از سیستم فایل
- بازیابی سیستم فایل از بک‌آپ

- مهاجرت سیستم فایل به دستگاه دیگری

مثال:

```
xfsdump -v -f /backup/mybackup.tar /dev/sda1
```

- v:-حالت مفصل (Verbose) را فعال می‌کند.
- f:-فایل بک‌آپ را مشخص می‌کند.

گزینه‌های دیگر و کاربردهای پیشرفته

هر یک از این دستورات دارای گزینه‌های دیگری نیز هستند که امکانات بیشتری را فراهم می‌کنند .برای مثال:

- **n:**-تنها بررسی می‌کند و هیچ تغییری ایجاد نمی‌کند.
- **L:**-تعمیر آنلاین را انجام می‌دهد.
- **f:**-فایل بک‌آپ یا فایل حاوی لیست فایل‌ها را مشخص می‌کند.
- **P:**-عملیات را به صورت موازی انجام می‌دهد.

ایجاد سیستم فایل

دستور mke2fs

دستور mke2fs در لینوکس برای ایجاد یک سیستم فایل ext2، ext3 یا ext4 استفاده می‌شود. این دستور به شما اجازه می‌دهد تا یک سیستم فایل جدید بر روی یک دیسک یا دستگاه راه‌اندازی کنید.

گزینه های دستور:

- t format -: این پارامتر برای تعیین فرمت سیستم فایل (ext2، ext3 یا ext4) استفاده می‌شود.
- c- -: این پارامتر برای بررسی بلاک‌های پیش از ایجاد سیستم فایل استفاده می‌شود.
- L label -: این پارامتر برای تعیین برچسب label برای سیستم فایل استفاده می‌شود.
- j -: این پارامتر برای ایجاد یک journal برای سیستم فایل استفاده می‌شود.
- m: -: این پارامتر برای تعیین درصد فضای برای استفاده سیستم فایل برای سیستم journal استفاده می‌شود.
- O -: این پارامتر برای فعال کردن ویژگی‌های خاص برای سیستم فایل استفاده می‌شود.

مثال استفاده از mke2fs:

```
mke2fs -t ext4 /dev/sdb1
```

این دستور سیستم فایل ext4 را بر روی دیسک /dev/sdb1 ایجاد می‌کند. حتما قبل از استفاده از این دستور، از اعمال آن بر روی داده‌های مهم خود پشتیبان بگیرید تا از از دست دادن اطلاعات جلوگیری شود.

ایجاد سیستم فایل Ext2

```
# mke2fs   /dev /hdXX
```

ایجاد سیستم فایل Ext3

```
# mke2fs   -j   / dev/ hdXX

یا

# mkfs.ext3   /dev/ hdXX
```

گزینه j- برای journaling استفاده می شود.

ایجاد سیستم فایل Ext4

```
# mke2fs  -t  ext4 /dev /hdXX
```

عددXX شماره درایو مورد نظر ست.

از سویچ t– برای مشخص کردن نوع سیستم فایل .

تبدیل سیستم فایلها

برای تغییر یک سیستم فایل ext2به ext3فعال کردن ویژگی ژورنالینگ، از دستور استفاده کنید. دستور tune2fs در لینوکس برای تنظیم و تغییر پارامترهای یک سیستم فایل ext2، ext3 یا ext4 استفاده می‌شود. این دستور به شما امکان می‌دهد تا پارامترهای مختلفی را برای یک سیستم فایل موجود تغییر دهید.

ساختار دستور :

```
tune2fs [ -l ] [ -c max-mount-counts ] [ -e errors-behavior ] [ -f ] [ -i interval-between-checks ] [ -j ] [ -J journal-options ] [ -m reserved-blocks-percentage ] [ -o [^]mount-options[,...] ] [ -r reserved-blocks-count ] [ -s sparse-super-flag ] [ -u user ] [ -g group ] [ -C mount-count ] [ -E extended-options ] [ -L volume-name ] [ -M last-mounted-directory ] [ -O [^]feature[,...] ] [ -T time-last-checked ] [ -U UUID ] device
```

گزینه های دستور:

```
tune2fs -L [برچسب]
```

این دستور برای تغییر برچسب (label) یک سیستم فایل استفاده می‌شود.

```
tune2fs -c [تعداد]
```

این دستور برای تغییر تعداد بلاک‌هایی که باید بلاک‌های پیش از ایجاد سیستم فایل بررسی کند، استفاده می‌شود.

```
tune2fs -m [درصد]
```

این دستور برای تغییر درصدی از فضای سیستم فایل که برای استفاده سیستم journal اختصاص داده شده است، استفاده می‌شود.

```
tune2fs -U [شناسه]
```

این دستور برای تغییر UUID یک سیستم فایل استفاده می‌شود.

```
tune2fs -r [تعداد]
```

این دستور برای تغییر تعداد بلاک‌هایی که باید بلاک‌های پشتیبان را نگهداری کند، استفاده می‌شود.

مثال های دستور :

تغییر برچسب سیستم فایل به " "mylabelبرای دستگاه /dev/sda1

```
tune2fs -L mylabel /dev/sda1
```

تغییر تعداد بلاک‌های بررسی شده قبل از ایجاد سیستم فایل به ۱۰۰ برای دستگاه /dev/sda1

```
tune2fs -c 100 /dev/sda1
```

تغییر درصد فضای سیستم فایل که برای journaling اختصاص داده شده به ۵٪ برای دستگاه /dev/sda1

```
tune2fs -m 5 /dev/sda1
```

تغییر UUID سیستم فایل به "۱۲۳۴۵۶۷۸-۱۲۳۴-۱۲۳۴-۱۲۳۴-۱۲۳۴۵۶۷۸۹abc" برای دستگاه /dev/sda1

```
tune2fs -U 12345678-1234-1234-1234-123456789abc /dev/sda1
```

تغییر تعداد بلاک‌های نگهداری شده برای بلاک‌های پشتیبان به ۵ برای دستگاه /dev/sda1

```
tune2fs -r 5 /dev/sda1
```

دستور fsck: بررسی و تعمیر سیستم فایل

شرح دستور

دستور fsck(فایل سیستم چک) برای بررسی و تعمیر خطاهای احتمالی در سیستم فایل استفاده می‌شود .این دستور به صورت خودکار در هنگام بوت شدن سیستم اجرا می‌شود تا از سلامت سیستم فایل‌ها اطمینان حاصل کند .همچنین می‌توان آن را به صورت دستی برای بررسی و تعمیر سیستم فایل‌های خاص اجرا کرد.

ساختار دستور

```
fsck [-sAVRTMNP] [-C [fd]] [-t fstype] [filesys...] [--] [fs-specific-options]
```

کاربردهای دستور

- **بررسی یکپارچگی سیستم فایل :**اطمینان از اینکه ساختار داده‌ای سیستم فایل سالم و بدون خطا است.
- **تعمیر خطاهای سیستم فایل :**شناسایی و رفع خطاهایی مانند بلوک‌های خراب، ایندکس‌های نادرست و. ...
- **پیشگیری از از دست رفتن داده‌ها :**با شناسایی و تعمیر به موقع خطاها، از از دست رفتن داده‌ها جلوگیری می‌شود.

گزینه‌های دستور و مثال‌ها

- **-s:**گزارش مختصر (Summary) از عملیات انجام شده را نمایش می‌دهد.
- **-A:**تمام سیستم فایل‌های موجود در /etc/fstab را بررسی می‌کند.
- **-V:**حالت مفصل (Verbose) را فعال می‌کند و اطلاعات بیشتری را در مورد عملیات انجام شده نمایش می‌دهد.
- **-R:**عملیات تعمیر خودکار را انجام می‌دهد.
- **-T:**عملیات تعمیر خودکار را بدون پرسیدن سوال انجام می‌دهد.
- **-M:**از ایجاد یک فایل ژورنال برای ثبت تغییرات خودداری می‌کند.
- **-N:**فقط بررسی می‌کند و هیچ تغییری در سیستم فایل ایجاد نمی‌کند.
- **-P:**خروجی را به صورت موازی (Parallel) انجام می‌دهد.
- **-C [fd]:**پیام‌های خطا را به فایل توصیفگر (file descriptor) مشخص شده هدایت می‌کند.
- **-t fstype:**فقط سیستم فایل‌های از نوع مشخص شده را بررسی می‌کند) مثلاًext4 ، xfs).
- **filesys...:**لیستی از دستگاه‌ها یا سیستم فایل‌هایی است که باید بررسی شوند.
- **--:**تمام گزینه‌های بعدی را به عنوان آرگومان‌های سیستم فایل در نظر می‌گیرد.
- **fs-specific-options:**گزینه‌های خاص برای هر نوع سیستم فایل (مثلاً برایext4 ، xfsو(...

نکات مهم

- **اجرای fsck روی سیستم فایل‌های متصل شده :**به طور معمول، نباید fsck را روی سیستم فایل‌هایی که در حال استفاده هستند اجرا کنید .این کار می‌تواند باعث از دست رفتن داده‌ها شود.

- **توقف خودکار سیستم :**در برخی موارد، fsckممکن است سیستم را به طور خودکار متوقف کند تا بتواند عملیات تعمیر را به طور کامل انجام دهد.
- **انواع سیستم فایل :**هر نوع سیستم فایل(ext2) ، ext3، ext4، xfsو (... ممکن است گزینه‌های خاص خود را داشته باشد .برای اطلاعات بیشتر، به مستندات مربوط به هر نوع سیستم فایل مراجعه کنید.

مثال‌های عملی برای دستور fsck با استفاده از گزینه‌های مختلف

در ادامه، مثال‌های متنوعی برای استفاده از دستور fsckبا گزینه‌های مختلف ارائه شده است تا درک بهتری از کاربرد هر گزینه حاصل شود.

مثال های ساده

- **بررسی کلی تمام سیستم فایل‌ها به صورت خلاصه:**

```
fsck -sA
```

- **بررسی و تعمیر خودکار سیستم فایل :/dev/sda1**

```
fsck -a /dev/sda1
```

- **بررسی سیستم فایلext4**

```
fsck -V -N -t ext4 /dev/sdb2
```

مثال های با گزینه‌های ترکیبی

- **بررسی تمام سیستم فایل‌ها به صورت مفصل و هدایت خروجی به فایل:log.txt**

```
fsck -AV -C > log.txt
```

- **بررسی سیستم فایل‌های ext4 و xfs به صورت موازی و تعمیر خودکار بدون پرسش:**

```
fsck -t ext4,xfs -PRT
```

- **بررسی سیستم فایل /dev/sda3 و هدایت پیام‌های خطا به فایل:error.log**

```
fsck -C 2> error.log /dev/sda3
```

در مثال بالا، error.log <2خطاها را به فایل error.log هدایت می‌کند.

مثال های با گزینه‌های خاص سیستم فایل

- **بررسی سیستم فایل ext4 و فعال کردن بررسی جامع:(Extended checks)**

```
fsck.ext4 -c /dev/sda2
```

گزینه c-در fsck.ext4 برای بررسی جامع استفاده می‌شود.

- **بررسی سیستم فایل XFS و تعمیر خطاهای آنلاین:**

```
xfs_repair -L /dev/sdb1
```

دستور xfs_repairبرای تعمیر سیستم فایل XFS استفاده می‌شود و گزینه L-برای تعمیر آنلاین است.

مثال برای بررسی و تعمیر سیستم فایل در حالتSingle User Mode

اگر سیستم شما به دلیل مشکلات سیستم فایل قابل بوت شدن نیست، می‌توانید از یک Live CD/USB برای تعمیر آن استفاده کنید .پس از بوت شدن ازLive CD/USB ، سیستم فایل را unmount کرده و سپس دستور fsck را اجرا کنید:

```
umount /dev/sda1
fsck /dev/sda1
```

توجه :همیشه قبل از انجام هرگونه تغییری در سیستم فایل، از داده‌های خود نسخه پشتیبان تهیه کنید.

دستورات مدیریت آرایه‌های RAID

دستور mdadm

شرح mdadm :ابزار اصلی برای مدیریت آرایه‌های RAID نرم‌افزاری در لینوکس است .این ابزار امکان ایجاد، مدیریت، و حذف آرایه‌های RAID را فراهم می‌کند.

ساختار دستور:

```
mdadm [options] [device]
```

کاربردها:

- ایجاد آرایه‌های RAID جدید
- مدیریت آرایه‌های موجود (افزودن، حذف دیسک، بررسی وضعیت)
- پیکربندی پارامترهای آرایه
- تعمیر آرایه‌های آسیب دیده

گزینه‌های مهم:

- `--create`: برای ایجاد آرایه جدید استفاده می‌شود.
- `--assemble`: برای جمع‌آوری مجدد آرایه استفاده می‌شود.
- `--stop`: برای متوقف کردن آرایه استفاده می‌شود.
- `--remove`: برای حذف یک دیسک از آرایه استفاده می‌شود.
- `--add`: برای افزودن یک دیسک به آرایه استفاده می‌شود.
- `--grow`: برای افزایش اندازه آرایه استفاده می‌شود.
- `--manage`: برای مدیریت آرایه‌ها به صورت تعاملی استفاده می‌شود.
- `--detail`: اطلاعات مفصل درباره آرایه را نمایش می‌دهد.
- `--scan`: برای اسکن سیستم و یافتن آرایه‌های موجود استفاده می‌شود.

توضیحات و مثال‌های بیشتر برای گزینه‌های دستور mdadm

ایجاد آرایه جدید

--create: برای ایجاد یک آرایه RAID جدید استفاده می‌شود.

```
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda /dev/sdb
```

- `/dev/md0`: نام دستگاه آرایه
- `--level=1`: سطح RAID را به ۱ (آینه سازی) تنظیم می‌کند

- `--raid-devices=2`: تعداد دیسک‌های مورد استفاده در آرایه را به ۲ تنظیم می‌کند
- `/dev/sda` و `/dev/sdb`: دستگاه‌های فیزیکی که در آرایه استفاده می‌شوند

جمع‌آوری مجدد آرایه

--assemble: برای جمع‌آوری مجدد یک آرایه که قبلاً ایجاد شده است، اما به دلایلی از هم گسیخته شده است، استفاده می‌شود.

```
mdadm --assemble /dev/md0
```

متوقف کردن آرایه

--stop: برای متوقف کردن یک آرایه در حال اجرا استفاده می‌شود.

```
mdadm --stop /dev/md0
```

حذف یک دیسک از آرایه

--remove: برای حذف یک دیسک از یک آرایه استفاده می‌شود .توجه داشته باشید که حذف یک دیسک از یک آرایه RAID می‌تواند باعث کاهش ظرفیت یا از دست رفتن داده‌ها شود.

```
mdadm --remove /dev/md0 /dev/sdb
```

افزودن یک دیسک به آرایه

--add: برای افزودن یک دیسک جدید به یک آرایه موجود استفاده می‌شود .این کار باعث افزایش ظرفیت آرایه می‌شود.

```
mdadm --add /dev/md0 /dev/sdc
```

افزایش اندازه آرایه

--grow: برای افزایش اندازه یک آرایه موجود استفاده می‌شود .این کار معمولاً پس از افزودن یک دیسک جدید به آرایه انجام می‌شود.

```
mdadm --grow /dev/md0 --size=2048M
```

مدیریت تعاملی آرایه‌ها

--manage: برای مدیریت آرایه‌ها به صورت تعاملی استفاده می‌شود .این حالت به شما اجازه می‌دهد تا دستورات مختلف را به صورت مرحله به مرحله اجرا کنید.

```
mdadm --manage /dev/md0
```

نمایش اطلاعات مفصل

--detail: برای نمایش اطلاعات مفصل درباره یک آرایه استفاده می‌شود.

```
mdadm --detail /dev/md0
```

اسکن سیستم برای یافتن آرایه‌های موجود

--scan: برای اسکن سیستم و یافتن آرایه‌های موجود استفاده می‌شود.

```
mdadm --scan
```

مثال‌های پیشرفته‌تر

- **ایجاد یک آرایه RAID5 با سه دیسک:**

```
mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/sda /dev/sdb /dev/sdc
```

- **تغییر سطح RAID یک آرایه:**

```
mdadm --manage /dev/md0
```

#در حالت تعاملی، دستور change level را اجرا کرده و سطح جدید را مشخص کنید

نکات مهم:

mdadm.conf

شرح :فایل پیکربندی mdadm است که اطلاعات مربوط به آرایه‌های RAID را ذخیره می‌کند .این فایل به صورت خودکار توسط mdadmهنگام ایجاد یا تغییر آرایه به‌روز می‌شود.

ساختار فایل :این فایل شامل بخش‌هایی برای تعریف هر آرایه است .هر بخش شامل اطلاعاتی مانند سطحRAID ، دستگاه‌های تشکیل‌دهنده آرایه، و سایر پارامترها است.

کاربرد:

- ذخیره اطلاعات پیکربندی آرایه‌ها
- اطمینان از اینکه آرایه‌ها هنگام بوت شدن به صورت خودکار جمع‌آوری شوند.

مثال :یک ورودی نمونه در فایل:mdadm.conf

```
ARRAY /dev/md0 metadata=1.2 name=myarray
  uuid=12345678-9abc-def0-1234-56789abcdef0
  devices=/dev/sda1 /dev/sdb1
```

/proc/mdstat

شرح :یک فایل سیستم مجازی است که اطلاعات لحظه‌ای درباره وضعیت آرایه‌های RAID را نمایش می‌دهد.

ساختار فایل :این فایل شامل اطلاعاتی مانند وضعیت آرایه، دستگاه‌های تشکیل‌دهنده آرایه، تعداد بلوک‌های داده، و وضعیت هر یک از دستگاه‌ها است.

کاربرد:

- نظارت بر وضعیت آرایه‌ها
- تشخیص خطاهای احتمالی
- بررسی پیشرفت عملیات تعمیر

مثال:

```
cat /proc/mdstat
```

خروجی نمونه:

```
Personalities : [raid1]
md0 : active raid1 sda1[1] sdb1[0] 1048576 blocks [2/2] [UU]
```

جمع‌بندی:

- **mdadm:**ابزار اصلی برای مدیریت آرایه‌های RAID نرم‌افزاری.
- **mdadm.conf:**فایل پیکربندی برای ذخیره اطلاعات آرایه‌ها.
- **proc/mdstat/:**فایل سیستم مجازی برای نمایش وضعیت لحظه‌ای آرایه‌ها.

مفهوم ژورنالینگ

سیستم فایل ژورنالیستی (JFS)توسط IBM برای AIX یونیکس توسعه داده شد که به عنوان جایگزینی برای system ext استفاده شد. JFS.جایگزینی برای ext4در حال حاضر است و مورد استفاده قرار می گیرد که در آن ثبات با استفاده از منابع بسیار کمی مورد نیاز است.وقتی قدرت CPU محدود است. سیستم فایل ژورنالی یک نوع سیستم فایل است که در لینوکس استفاده می‌شود و از یک جورنال برای ثبت تغییرات و عملیات‌های صورت گرفته بر روی فایل‌ها استفاده می‌کند. این نوع سیستم فایل بهبود امنیت و پایداری داده‌ها را ارتقا می‌دهد و از از دست دادن داده‌ها جلوگیری می‌کند.

ویژگی‌های سیستم فایل ژورنالی

Journalingجورنالینگ: یکی از اصلی‌ترین ویژگی‌های سیستم فایل ژورنالی، استفاده از یک جورنال برای ثبت تغییرات است. این جورنال به سیستم فایل کمک می‌کند تا در صورت قطع ناگهانی برق یا خطاهای سیستم، بتواند به سرعت و با دقت تغییرات را بازیابی کند.

بهبود عملکرد: سیستم فایل ژورنالی بهبود عملکرد خواندن و نوشتن داده‌ها را نیز فراهم می‌کند. با استفاده از جورنال برای ثبت تغییرات، عملیات خواندن و نوشتن بهبود می‌یابد.

پایداری داده‌ها: با استفاده از سیستم فایل ژورنالی، احتمال از دست دادن داده‌ها به دلیل قطعی ناگهانی برق یا خطاهای سیستم کاهش می‌یابد. این سیستم فایل توانایی بازیابی داده‌ها را دارد.

سازگاری با برنامه‌ها: بسیاری از برنامه‌ها و سیستم‌های عامل از سیستم فایل ژورنالی پشتیبانی می‌کنند و از قابلیت‌های آن برای ایجاد پایداری و امنیت برنامه‌ها استفاده می‌کنند.

سیستم فایل ژورنالی یکی از محبوب‌ترین انتخاب‌ها برای استفاده در لینوکس است و بسیاری از توزیع‌های لینوکس از این نوع سیستم فایل برای ایجاد پایداری و امنیت برنامه‌ها و داده‌ها استفاده می‌کنند.

سیستم فایل ReiserFS

این به عنوان یک جایگزین برای ext3با عملکرد بهبود یافته و ویژگی های پیشرفته معرفی شد.زمانی بود که پرونده SuSE linux به طور پیش فرض ReiserFSبود اما بعد از آن reiser از کسب و کار خارج شد و SuSe هیچ گزینه ای برای بازگشت به ext3 نداشت . ReiserFS از سیستم فایل Extensionبه صورت پویا پشتیبانی می کند که نسبتا یک ویژگی پیشرفته است اما سیستم فایل فاقد سطح مشخصی از عملکرد است. ReiserFS یک نوع سیستم فایل جدید و پیشرفته برای لینوکس است که توسط هانس رایزر Hans Reiserتوسعه داده شده است. این سیستم فایل از ویژگی‌ها و قابلیت‌های منحصر به فردی برخوردار است که آن را از سایر سیستم‌های فایل متمایز می‌کند.

ویژگی‌های سیستم فایل :ReiserFS

هماهنگی با فایل‌های کوچک: ReiserFS به خوبی برای مدیریت فایل‌های کوچک و بزرگ مناسب است. این سیستم فایل بهبود عملکرد خواندن و نوشتن بر روی فایل‌های کوچک را فراهم می‌کند.

همگام سازی تراکنشی: ReiserFS از روش‌های همگام سازی تراکنشی برای جلوگیری از از دست دادن داده‌ها در صورت قطع برق یا خطاهای سیستم استفاده می‌کند. این ویژگی امنیت و پایداری داده‌ها را ارتقا می‌دهد.

فشرده‌سازی داده‌ها: ReiserFS قابلیت فشرده‌سازی داده‌ها را نیز دارد که می‌تواند فضای ذخیره‌سازی را بهبود بخشیده و میزان استفاده از دیسک را کاهش دهد.

پشتیبانی از سطوح :RAIDاین سیستم فایل از سطوح (Redundant Array of Independent Disks) RAID نیز پشتیبانی می‌کند که به کاربران امکان افزایش امنیت و پایداری داده‌ها را می‌دهد.

ReiserFS یک گزینه جذاب برای استفاده در محیط‌هایی است که نیاز به مدیریت بهتر فایل‌ها، افزایش عملکرد و امنیت داده‌ها دارند. این سیستم فایل از ویژگی‌ها و قابلیت‌های منحصر به فردی برخوردار است که آن را یک گزینه قابل توجه برای استفاده در سیستم‌های لینوکس می‌سازد.

XFS سیستم فایل

XFSیک JFSبا سرعت بالا بود که با هدف پردازش همزمان I / Oانجام شد. ناسا این سیستم فایل را در سرور ۳۰۰ ترابایتی خود استفاده می کند. XFS یکی دیگر از سیستم‌های فایل محبوب برای لینوکس است که توسط شرکت SGI (Silicon Graphics International توسعه یافته است. این سیستم فایل از ویژگی‌ها و قابلیت‌های قدرتمندی برخوردار است که آن را به یک گزینه مناسب برای استفاده در محیط‌هایی با نیازهای بالا تبدیل کرده است.

ویژگی‌های سیستم فایل :XFS

پشتیبانی از حجم بزرگ: XFS قابلیت مدیریت حجم‌های بزرگ از داده‌ها را دارد و برای سرورها و سیستم‌هایی با حجم داده بالا مناسب است.

عملکرد بالا: از جمله ویژگی‌های برجسته XFS، عملکرد بالای خواندن و نوشتن داده‌ها به صورت همزمان بر روی دیسک‌ها است که این سیستم فایل را برای سرورها و برنامه‌هایی با نیاز به عملکرد بالا مناسب می‌کند.

هماهنگی با فایل‌های بزرگ: XFS به خوبی برای مدیریت فایل‌های بزرگ و حجم بالا مناسب است و از این نظر مورد توجه است.

پشتیبانی از تراکنش‌های صورت گرفته: XFS از تراکنش‌های صورت گرفته برای جلوگیری از از دست دادن داده‌ها در صورت قطعی ناگهانی برق یا خطاهای سیستم استفاده می‌کند.

XFS یکی از گزینه‌های مناسب برای استفاده در محیط‌هایی با حجم داده بالا و نیاز به عملکرد بالا محسوب می‌شود. این سیستم فایل از ویژگی‌ها و قابلیت‌های پیشرفته‌ای برای مدیریت داده‌ها و ارتقای عملکرد سیستم‌ها برخوردار است که آن را یک گزینه جذاب برای استفاده در محیط‌های تجاری و صنعتی می‌سازد

Btrfs سیستم فایل

سیستم فایل Btrfs (B-Tree) بر پیکربندی ذخیره سازی بزرگ تمرکز کرده و هنوز هم در حال توسعه است. Btrfs یک سیستم فایل مدرن و پیشرفته برای لینوکس است که از طرف شرکت Oracle توسعه یافته است. این سیستم فایل از ویژگی‌ها و قابلیت‌های منحصر به فردی برخوردار است که آن را از سایر سیستم‌های فایل متمایز می‌کند.

ویژگی‌های سیستم فایل :Btrfs

نگهداری داده‌ها با کپی تکراری Btrfs :(Copy-on-Write) از روش نگهداری داده‌ها با کپی تکراری برای جلوگیری از از دست دادن داده‌ها در صورت خطا یا قطعی ناگهانی برق استفاده می‌کند.

Snapshots: Btrfs از قابلیت ایجاد Snapshots برای ذخیره و بازیابی وضعیت قبلی فایل‌ها و سیستم‌ها استفاده می‌کند. این ویژگی به کاربران امکان بازگشت به وضعیت قبلی داده‌ها را فراهم می‌کند.

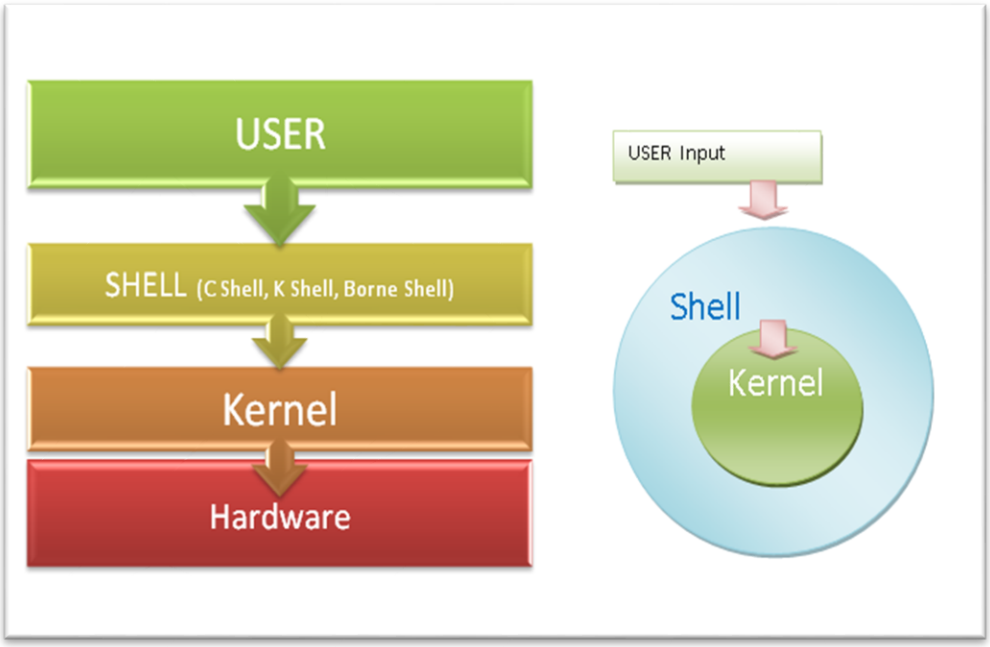
RAID و مدیریت حجم: Btrfs از پشتیبانی از سطوح RAID و امکان مدیریت حجم بر روی دیسک‌ها برای افزایش امنیت و انعطاف‌پذیری داده‌ها استفاده می‌کند.

فشرده‌سازی داده‌ها: Btrfs از قابلیت فشرده‌سازی داده‌ها بر روی دیسک‌ها برای کاهش فضای ذخیره‌سازی و بهبود عملکرد استفاده می‌کند.

Btrfs یک سیستم فایل قدرتمند با ویژگی‌ها و قابلیت‌های پیشرفته است که برای استفاده در محیط‌هایی که نیاز به انعطاف‌پذیری، امنیت و قابلیت بازیابی داده‌ها دارند، مناسب است. این سیستم فایل از روش‌های نوینی برای مدیریت داده‌ها و بهبود عملکرد سیستم‌ها استفاده می‌کند که آن را یک گزینه جذاب برای استفاده در سیستم‌های لینوکس می‌سازد.

پوسته–شل چیست؟

به عبارت ساده، پوسته[۱] یک برنامه است که دستورات را از صفحه کلید گرفته و آنها را به کرنل سیستم عامل می دهد تا اجرا شود. در زمان های قدیم، این تنها رابط کاربری موجود در سیستم یونیکس مانند لینوکس بود. امروزه ما رابطهای کاربری گرافیکی (GUI) را علاوه بر رابطهای خط دستور (CLI) مانند پوسته داریم. در اکثر سیستم های لینوکس یک برنامه به نام bash که برای Bourne Again Sheell نامیده می شود، نسخه پیشرفته یونیکس پوسته اصلی sh است که توسط Steve Bourne نوشته شده است. به عنوان برنامه پوسته عمل می کند. علاوه برbash ، دیگر برنامه های پوسته وجود دارد که می توانند در یک سیستم لینوکس نصب شوند. اینها عبارتند از: ksh ،tcsh و zsh.



ترمینال در تصویر بالا ، قسمت user قرار میگیرد که به شل متصل است و شل به کرنل.

ترمینال چیست؟

برنامه ای است که یک پنجره را باز می کند و به اجازه می دهد با پوسته ارتباط برقرار کنید. تعدادی از شبیه سازهای ترمینال مختلف می‌توانید استفاده کنید. خط فرمان Linux یک رابط متنی برای رایانه است. اغلب به عنوان پوسته ، ترمینال ، کنسول ، سریع یا نام های مختلف دیگر نامیده می‌شود، استفاده از آن پیچیده و گیج کننده است. با این وجود توانایی کپی و paste دستورات از یک وب سایت ، همراه با قدرت و انعطاف پذیری خط فرمان را دارد.

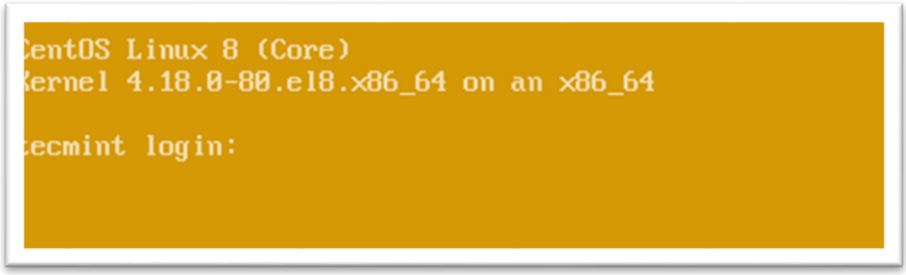
انواع ترمینالها

اکثر توزیعهای لینوکس دارای ترمینالهای[۲] پیش فرض خود هستند.به عبارت دیگر، شبیه ساز ترمینال این توانایی را دارد که یک ماشین گنگ را مانند یک کامپیوتر مشتری متصل به سرور به نظر برساند. شبیه ساز ترمینال به کاربر نهایی[۳] اجازه می‌دهد تا به کنسول[۴] و

---

۱ shell
۲ Terminal Emulators for Linux
۳ End user
۴ console

همچنین برنامه های کاربردی آن مانند رابط کاربری متنی و رابط خط فرمان[۱] دسترسی داشته باشد. ممکن است تعداد زیادی شبیه ساز ترمینال برای انتخاب در این دنیای منبع باز[۲] پیدا کنید. برخی از آنها طیف وسیعی از ویژگی ها را ارائه میدهند در حالی که برخی دیگر ویژگی های کمتری را ارائه می دهند. برای درک بهتر کیفیت نرم افزار موجود، لیستی از شبیه سازهای ترمینال شگفت انگیز برای لینوکس را گردآوری کرده ایم.



User Login Prompt

لازم به ذکر است تمامی دستورات لینوکس با حروف کوچیک می باشند. و همچنین سویچ یا همان options های دستورات به صورت بزرگ و کوچیک وجود دارد.

ترمینال Terminator

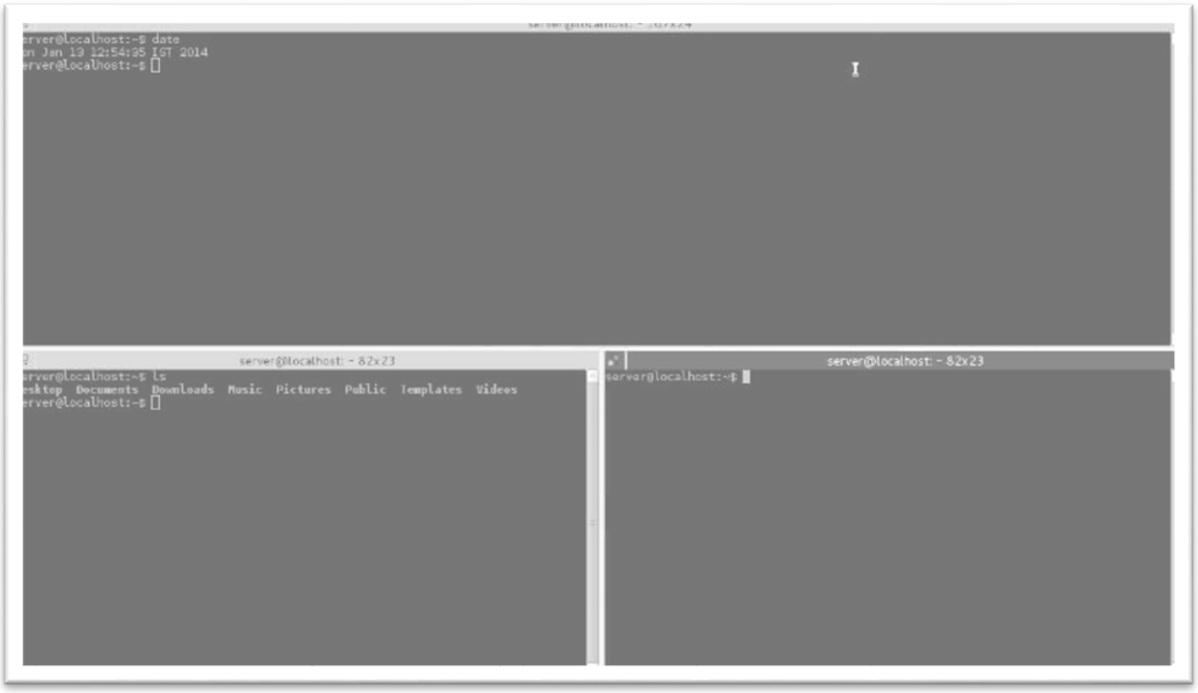
یک شبیه ساز ترمینال پیشرفته و قدرتمند است که از چندین پنجره ترمینال پشتیبانی میکند و دارای برخی عملکردهای اضافی است که در برنامه ترمینال پیش فرض لینوکس نخواهید یافت. به عنوان مثال ، در برنامه ترمیناتور، می توانید پنجره های ترمینال خود را به صورت افقی و عمودی بر اساس نیاز خود تقسیم[۳] کنید.

ویژگی ها

- پروفایل ها[۴] و طرح های رنگی[۵] خود را میتوانید سفارشی کنید[۶]، یا اندازه را متناسب با نیاز خود تنظیم کنید.
- از افزونه ها[۷] استفاده کنید تا عملکرد بیشتری داشته باشید.
- چند کلید میانبر[۸] برای سرعت بخشیدن به فعالیت های رایج دارد.
- پنجره ترمینال را به چند ترمینال مجازی تقسیم کنید و در صورت نیاز اندازه آنها را تغییر دهید.

---

۱ Command line
۲ Opensource
۳ split
۴ profile
۵ theme
۶ customize
۷ Extension
۸ shortcutkey



ترمیناتور برای لینوکس

برای نصب Terminatorدر لینوکس، همانطور که نشان داده شده است از مدیر بسته[۱] پیش فرض خود استفاده کنید.

```
$ sudo apt install terminator [ در دبیان، اوبونتو و مینت ]
$ sudo yum install terminator [ در RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux ]
$ sudo emerge -a x11-terms/terminator [ در جنتو لینوکس ]
$ sudo pacman -S terminator [ در آرچ لینوکس ]
$ sudo zypper install terminator [ در OpenSUSE ]
```

ترمینال Tilda

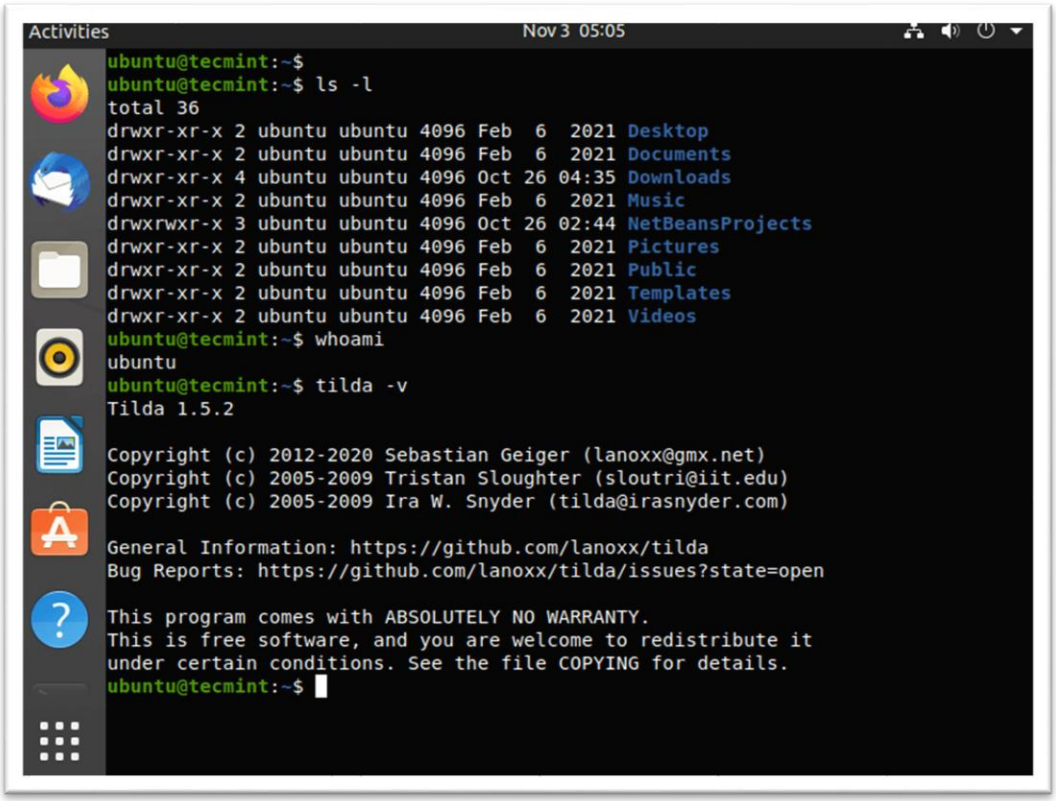
Tildaیک ترمینال کشویی شیک مبتنی بر GTK+[۲] است. با کمک یک کلید، می توانید یک پنجره Tilda جدید یا پنهان را راه اندازی کنید. با این حال، می توانید رنگ های دلخواه خود را برای تغییر ظاهر متن و پس زمینه ترمینال اضافه کنید. علاوه بر این، Tildaبسیار قابل تنظیم است، به عنوان مثال ، شما می توانید کلیدهای میانبر را برای صفحه کلید پیکربندی کنید، ظاهر را تغییر دهید، و بسیاری از گزینه هایی که عملکرد Tilda را تغییر می دهند.

ویژگی ها:

- رابط[۳] UI با سفارشی سازی زیاد.
- می توانید سطح شفافیت[۴] را برای پنجره Tilda تنظیم کنید.
- طرح های رنگی داخلی عالی.

---

۱ Package manager
۲ GUI toolkit
۳ User interface
۴ transparent



ترمینال Tilda برای لینوکس

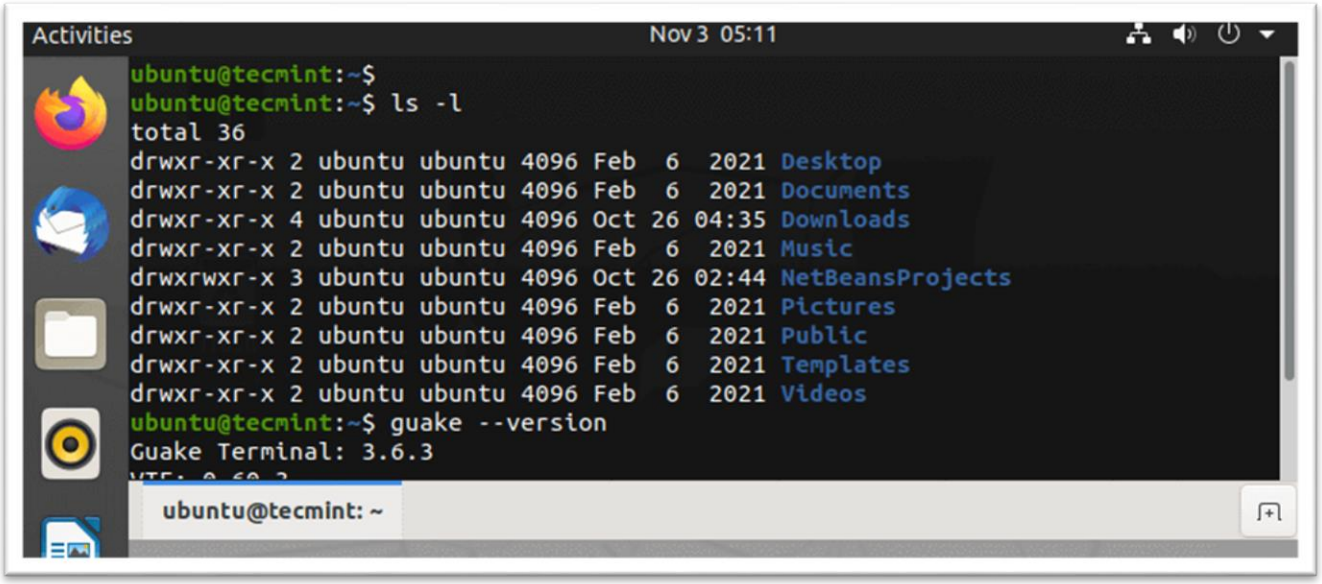
برای نصب **Tilda** در لینوکس، همانطور که نشان داده شده است از مدیر بسته پیش فرض خود استفاده کنید.

```
$ sudo apt install tilda [ در دبیان، اوبونتو و مینت ]
$ sudo yum  install tilda [ در RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux ]
$ sudo emerge -a x11-terms/tilda [ در لینوکس جنتو]
$ sudo pacman -S tilda [ در آرچ لینوکس]
$ sudo zypper install tilda [ OpenSUSE ]
```

ترمینال Guake

Guake یک ترمینال کشویی[۱] مبتنی بر پایتون[۲] است که برای محیط دسکتاپ GNOME ایجاد شده است. با فشار دادن یک کلید، فراخوانی شده و می توان با فشار دادن مجدد همان کلید، آن را پنهان کند. طراحی آن بر اساس بازی های FPS تیرانداز اول شخص مانند **Quake** مشخص شده است و یکی از اهداف اصلی آن دسترسی آسان است. **Guake** بسیار شبیه **Yakuaka** و **Tilda** است، اما آزمایشی برای ترکیب بهترین آنها در یک برنامه واحد مبتنی بر GTK است. از ابتدا با پایتون با استفاده از یک قطعه کوچک در C نوشته شده است.

[۱] Pop down
[۲] python



ترمینال Guake برای لینوکس

برای نصب **Guake** در لینوکس، همانطور که نشان داده شده است از مدیر بسته پیش فرض خود استفاده کنید.

```
$ sudo apt install guake [ در دبیان، اوبونتو و مینت]
$ sudo yum install guake [ در RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux ]
$ sudo emerge -a x11-terms/guake [ در جنتو لینوکس]
$ sudo pacman -S guake [ در آرچ لینوکس]
$ sudo zypper install  guake [ در OpenSUSE ]
```

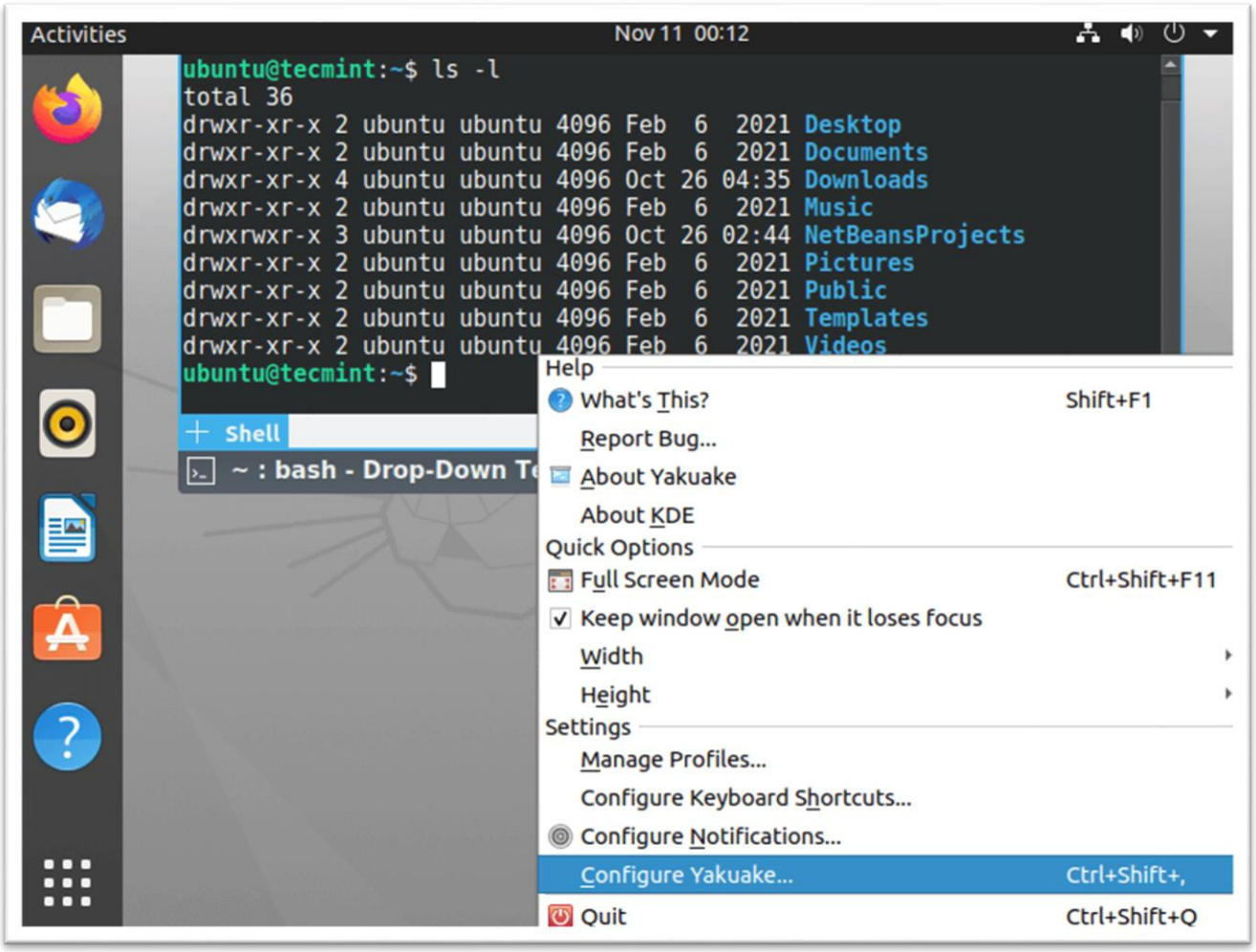
ترمینال Yakuake

Yakuake (Yet Another Kuake) یک شبیه ساز ترمینال کشویی مبتنی بر KDE است که از نظر عملکرد بسیارشبیه شبیه ساز ترمینال **Guake** است. طراحی آن از بازی های کنسول های فریم بر ثانیه مانند **Quake** الهام گرفته شده است. **Yakuake** اساساً یک برنامه KDE است که به راحتی می توان آن را روی دسکتاپ **KDE** نصب کرد، اما اگر سعی کنید **Yakuake** را در دسکتاپ **GNOME** نصب نمایید، از شما می خواهد که تعداد زیادی بسته وابسته را نصب کنید.

ویژگی ها:

- به آرامی از بالای صفحه به پایین می آید.
- رابط زبانه دار[1].
- ابعاد و سرعت انیمیشن قابل تنظیم
- قابل تنظیم.

[1] tab



ترمینال Yakuake برای لینوکس

برای نصب **Yakuake** در لینوکس، همانطور که نشان داده شده است از مدیر بسته پیش فرض خود استفاده کنید.

```
$ sudo apt install yakuake [ در دبیان، اوبونتو و مینت]
$ sudo yum install  yakuake [ در RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux ]
$ sudo emerge -a kde-apps/yakuake [ در لینوکس جنتو]
$ sudo pacman -S yakuake [ در آرچ لینوکس]
$ sudo zypper install  yakuake [ در OpenSUSE ]
```

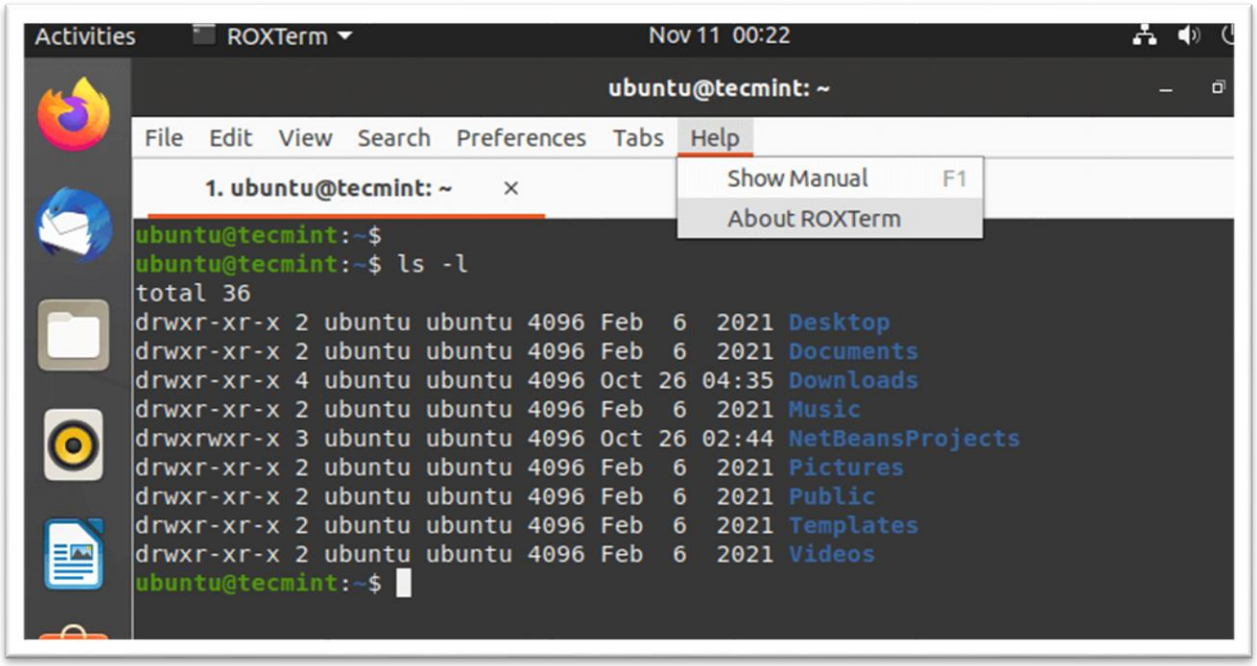
ترمینال ROXTerm

ROXterm یکی دیگر از شبیه سازهای ترمینال سبک است که برای ارائه ویژگی های مشابه gnome-terminal طراحی شده است. در ابتدا با استفاده نکردن از کتابخانه‌های Gnome و با استفاده از یک اپلت[۱] مستقل برای آوردن رابط پیکربندی (GUI) کمتر و زمان راه‌اندازی سریع‌تر ساخته شد، اما با گذشت زمان نقش آن به ارائه طیف بالاتری از ویژگی‌ها تغییر کرد. با این حال، بیشتر از gnome-terminal قابل تنظیم است. به راحتی با محیط دسکتاپ گنوم ادغام می شود و ویژگی هایی مانند کشیدن و رها کردن[۲] موارد به ترمینال را ارائه می دهد.

---

[۱] Applet

[۲] Drag&drop



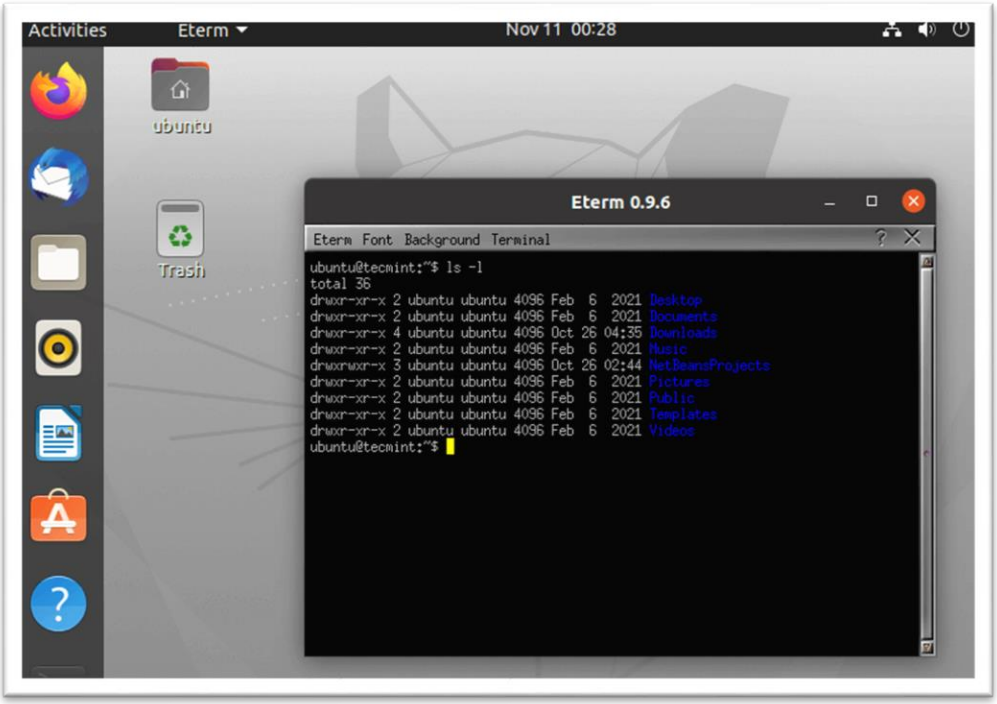
ترمینال ROXTerm برای لینوکس

برای نصب Roxterm در لینوکس، همانطور که نشان داده شده است از مدیر بسته پیش فرض خود استفاده کنید.

```
$ sudo apt install roxterm [ در دبیان، اوبونتو و مینت ]
$ sudo yum install roxterm [ در RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux ]
$ sudo emerge -a x11-terms/roxterm [ در لینوکس جنتو ]
$ sudo pacman -S roxterm [ در آرچ لینوکس ]
$ sudo zypper install roxterm [ در OpenSUSE ]
```

ترمینال Eterm

Eterm سبک ترین شبیه ساز ترمینال و رنگی است که به عنوان جایگزینی برای **xterm** طراحی شده است. این با ایدئولوژی آزادی انتخاب توسعه یافته است و به همان اندازه قدرت، انعطاف‌پذیری و آزادی را در دستان کاربر می‌گذارد. Eterm یک شبیه‌ساز ترمینال سبک و رنگی است که بخوبی به عنوان جایگزینی برای xterm طراحی شده است. این شبیه‌ساز ترمینال با ایدئولوژی آزادی انتخاب توسعه یافته است و به کاربر امکاناتی همچون قدرت، انعطاف‌پذیری و آزادی را ارائه می‌دهد. Eterm با امکانات گوناگونی که دارد، به کاربر امکان مدیریت و تنظیم ترمینال به دلخواه خود را می‌دهد. این شبیه‌ساز ترمینال قابلیت نصب افزونه‌ها و تنظیمات مختلف را دارا است که به کاربر امکان سفارشی‌سازی بیشتر را می‌دهد. از Eterm می‌توان برای انجام وظایف مختلف مانند مدیریت فایل‌ها، اجرای دستورات خاص، نمایش اطلاعات سیستم و غیره استفاده کرد. با توجه به ویژگی‌های آن، Eterm یک گزینه مناسب برای کاربرانی است که به دنبال یک ترمینال سبک، رنگی و قابل تنظیم هستند.



ترمینال Eterm

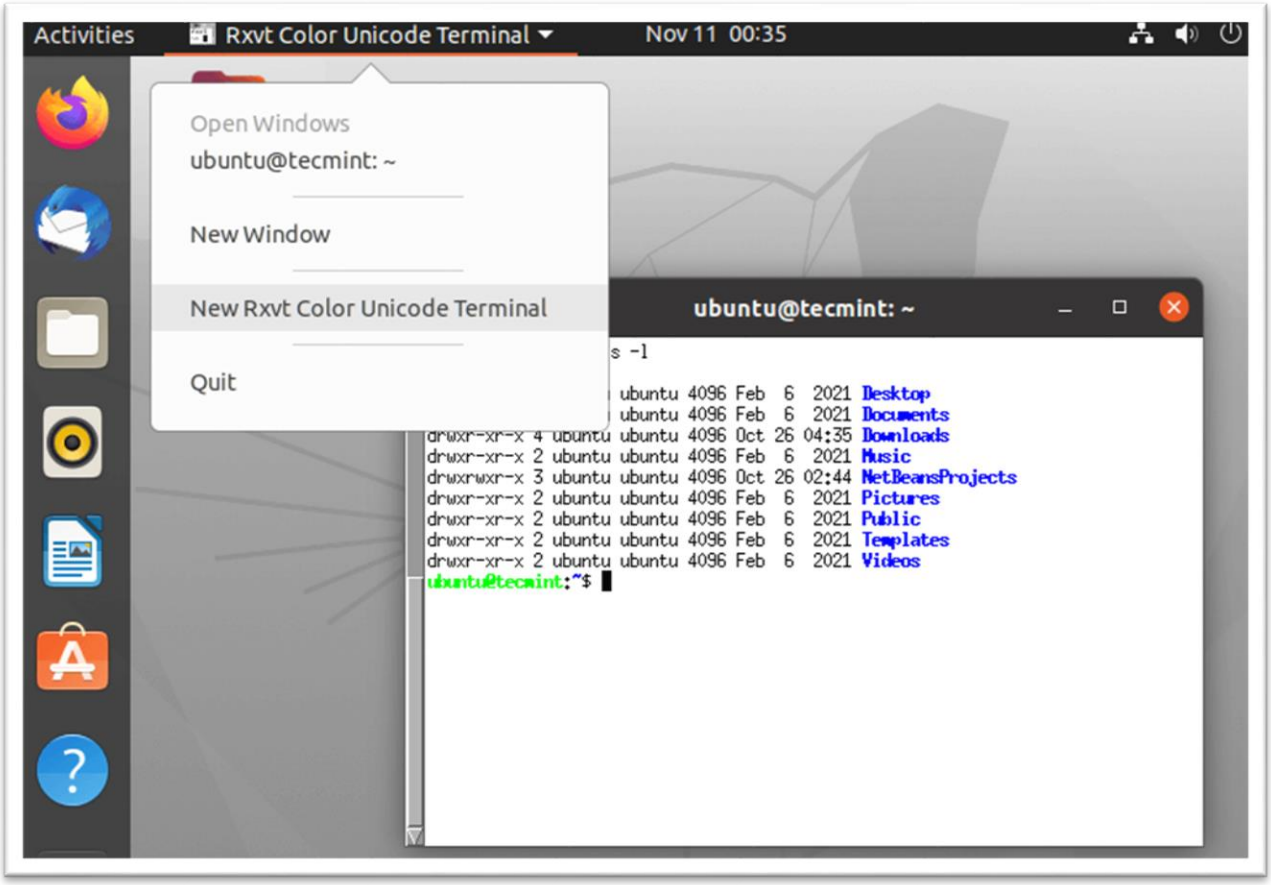
برای نصب **Eterm** در لینوکس، همانطور که نشان داده شده است از مدیر بسته پیش فرض خود استفاده کنید.

```
$ sudo apt install eterm [ در دبیان، اوبونتو و مینت]
$ sudo yum install eterm [ در RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux ]
$ sudo emerge -a x11-terms/eterm [ در لینوکس جنتو]
$ sudo pacman -S eterm [ در آرچ لینوکس]
$ sudo zypper install eterm [ در OpenSUSE ]
```

ترمینال Rxvt

Rxvt مخفف **ترمینال مجازی توسعه یافته**[1] یک برنامه ترمینال رنگی برای لینوکس است که به عنوان جایگزینی **xterm** برای کاربران قدرتمندی که نیازی به داشتن ویژگی هایی مانند شبیه سازی Tektronix 4014 و پیکربندی به سبک جعبه ابزار ندارند در نظر گرفته شده است.

[1] Extended virtual terminal



ترمینال Rxvt برای لینوکس

برای نصب **Rxvt** در لینوکس، همانطور که نشان داده شده است از مدیر بسته پیش فرض خود استفاده کنید.

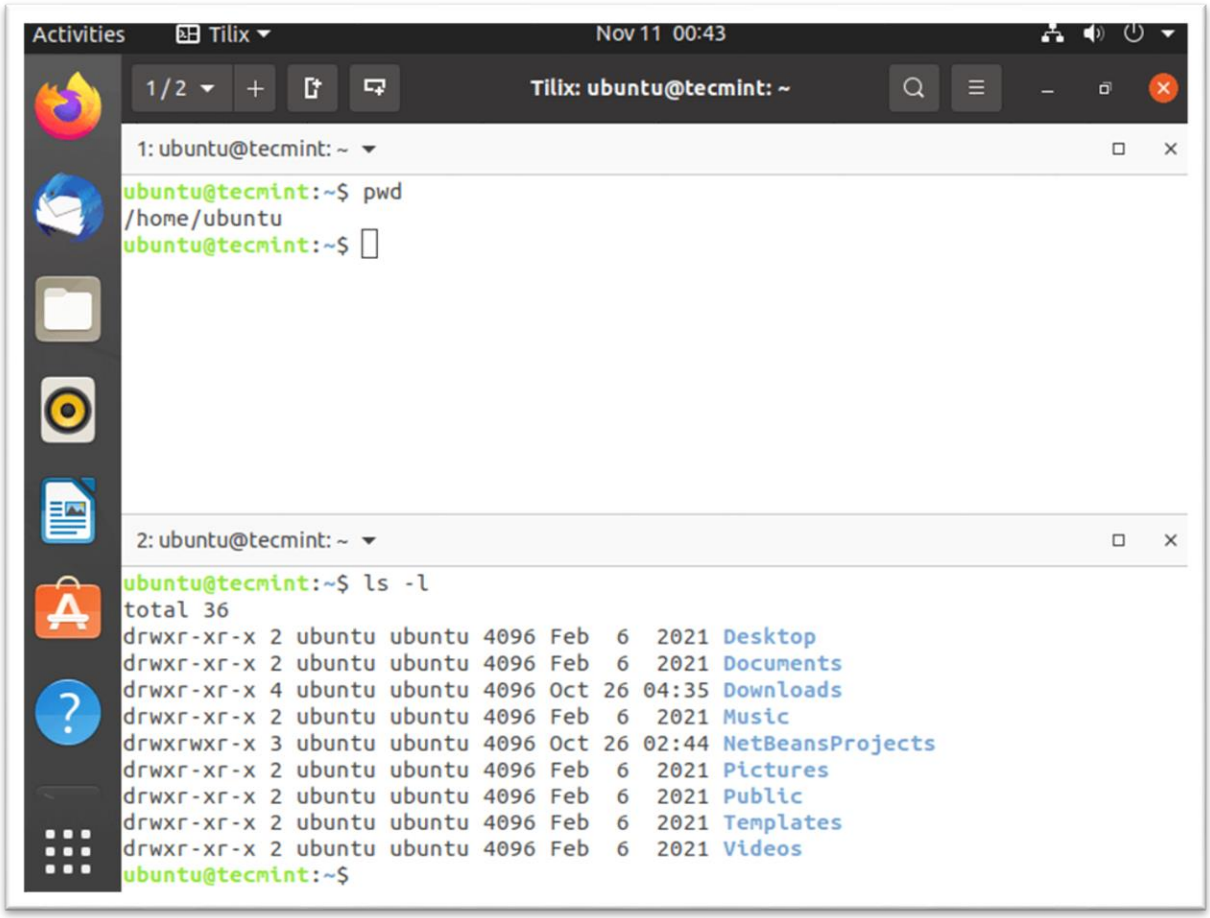
```
$ sudo apt install rxvt [در دبیان، اوبونتو و مینت]
$ sudo yum install rxvt [در RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux ]
$ sudo emerge -a x11-terms/rxvt [در لینوکس جنتو]
$ sudo pacman -S rxvt [در آرچ لینوکس]
$ sudo zypper install rxvt [در OpenSUSE ]
```

ترمینال Tilix

Tilix یک شبیه ساز ترمینال کاشی کاری[۱] مدرن GTK3 است که به شما امکان می دهد پنجره شبیه ساز خود را به چندین صفحه ترمینال شبیه به مالتی پلکسر ترمینال. tmux تقسیم کنید. Tilix یک شبیه‌ساز ترمینال کاشی‌کاری مدرن GTK3 است که امکان تقسیم پنجره ترمینال به چندین صفحه را مانند مالتی‌پلکسر ترمینال tmux فراهم می‌کند. این قابلیت به کاربر امکان می‌دهد تا بتواند بیشترین بهره را از فضای کاری خود برداشته و برنامه‌ها یا فرآیندهای مختلف را به صورت همزمان در چندین صفحه ترمینال مدیریت کند. Tilix با ویژگی‌های مدرن و کاربردی خود، یک گزینه عالی برای کاربرانی است که به دنبال ترمینالی با امکانات پیشرفته و امکانات مالتی‌تسکینگ هستند. این شبیه‌ساز ترمینال از امکانات GTK3 بهره می‌برد و تجربه کاربری را بهبود می‌بخشد.

استفاده از Tilix به کاربر امکان می‌دهد تا بتواند بهترین تنظیمات و تقسیم‌بندی‌ها را برای مدیریت کارهای خود انجام دهد و به راحتی واحدهای ترمینال خود را سفارشی کند. این شبیه‌ساز ترمینال با امکانات مالتی‌تسکینگ مناسب برای مدیریت کارهای چندگانه و همزمان است.

[۱] Tiled



ترمینال Tilix برای لینوکس

برای نصب **Tilix** در لینوکس، مطابق شکل از مدیر بسته پیش فرض خود استفاده کنید.

```
$ sudo apt install tilix [ در دبیان، اوبونتو و مینت]
$ sudo yum install tilix [ در RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux ]
$ sudo emerge -a x11-terms/tilix [ در لینوکس جنتو]
$ sudo pacman -S tilix [ در آرچ لینوکس]
$ sudo zypper install tilix [ در OpenSUSE ]
```

ترمینال LXTerminal

LXTerminal یک شبیه ساز ترمینال پیش فرض مبتنی بر VTE برای LXDE محیط رومیزی سبک ، بدون هیچ گونه وابستگی غیر ضروری است LXTerminal . یک شبیه‌ساز ترمینال پیش‌فرض است که بر اساس VTE برای محیط رومیزی سبک LXDE طراحی شده است. این ترمینال بدون هیچ گونه وابستگی غیرضروری، امکانات لازم را برای کاربران فراهم می‌کند. بین ویژگی‌های خوب LXTerminal می‌توان به موارد زیر اشاره کرد:

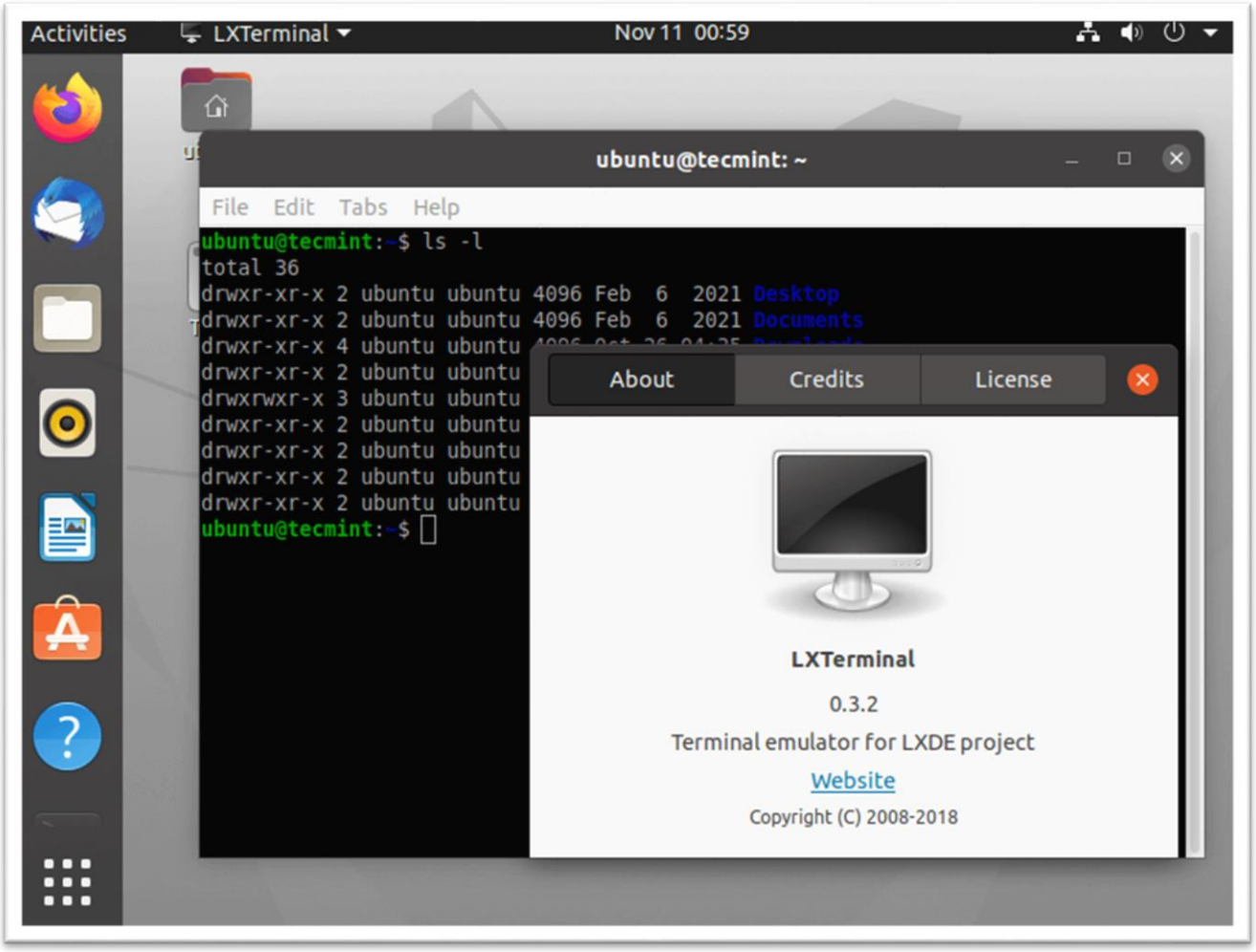
سبک و سریع: LXTerminal به عنوان یک شبیه‌ساز ترمینال سبک طراحی شده است و به کاربر امکان استفاده آسان و سریع را می‌دهد.

بدون وابستگی غیرضروری: LXTerminal بدون هیچ گونه وابستگی غیرضروری ارائه می‌شود که باعث افزایش کارایی و کاربردی بودن آن می‌شود.

امکانات مورد نیاز: این ترمینال امکانات لازم برای مدیریت فایل‌ها، اجرای دستورات، نمایش خروجی‌ها و سایر کارهای ترمینالی را فراهم می‌کند. LXTerminal با تمرکز بر روی سادگی، سرعت و کارایی، یک گزینه عالی برای کاربرانی است که به دنبال تجربه ترمینال سبک و کارامد هستند.

ترمینال دارای ویژگی های خوبی است مانند:

- پشتیبانی از چندین تب
- از دستورات رایج مانندcp ، cd، dir، mkdir، mvdirپشتیبانی می کند.
- قابلیت مخفی کردن نوار منو برای صرفه جویی در فضا
- میتوانید رنگ بندی را تغییر دهید



ترمینال LXTerminal برای لینوکس

برای نصب **LXTerminal** در لینوکس، همانطور که نشان داده شده است از مدیر بسته پیش فرض خود استفاده کنید.

```
$ sudo apt install lxterminal [ در دبیان، اوبونتو و مینت]
$ sudo yum install lxterminal [ در RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux ]
$ sudo emerge -a lxde-base/lxterminal [ در لینوکس جنتو]
$ sudo pacman -S lxterminal [ در آرچ لینوکس]
$ sudo zypper install lxterminal [ در OpenSUSE ]
```

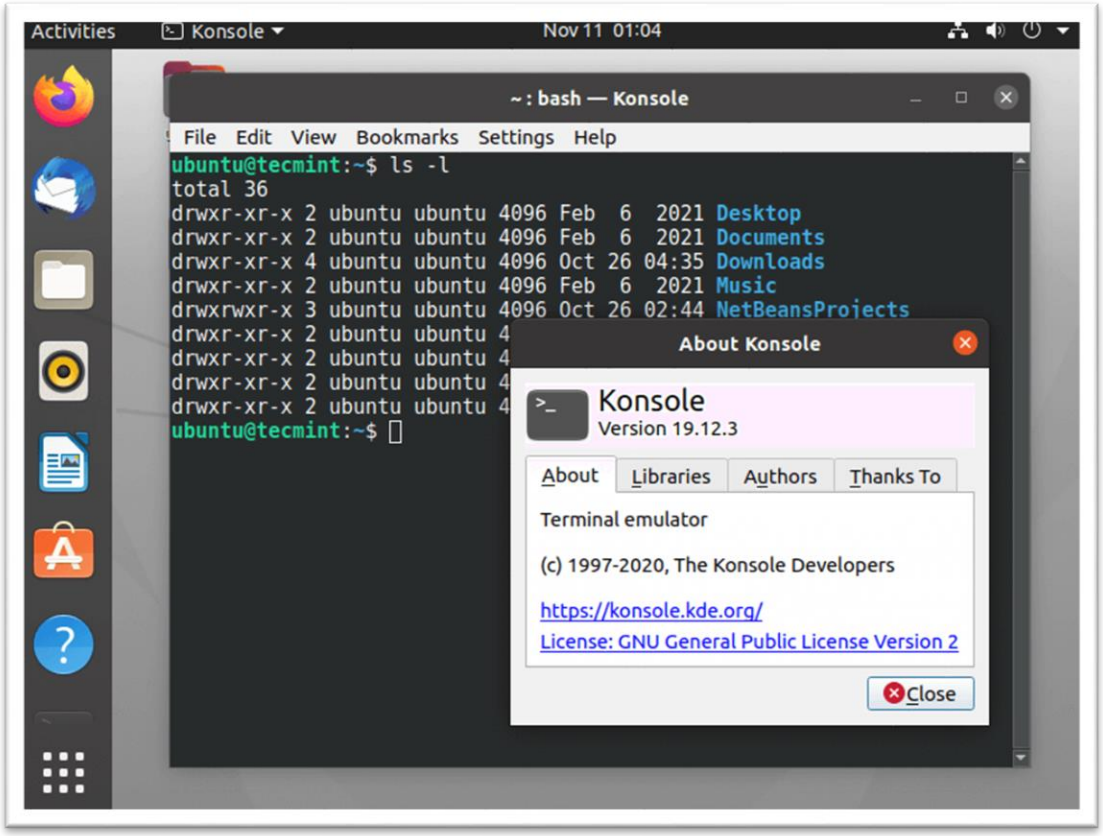
ترمینالKonsole

Konsoleیک شبیه‌ساز ترمینال آزاد **مبتنی بر KDE است** که در ابتدا توسط Lars Doelle ساخته شد .همچنین در چندین برنامه KDE دیگر ادغام شده است که دسترسی به آن را ساده تر و مناسب تر می کند. Konsole یک شبیه‌ساز ترمینال آزاد است که ببنی KDE توسعه یافته است و ابتدا توسط Lars Doelle ساخته شد. این ترمینال امکانات قدرتمندی برای کاربران KDE و محیط‌های دسکتاپ مبتنی بر آن فراهم می‌کند. یکی از ویژگی‌های برجسته Konsole ادغام آسان با برنامه‌های KDE دیگر است که دسترسی به آن را ساده‌تر و مناسب‌تر می‌کند. این امکان به کاربران این امکان را می‌دهد تا با استفاده از Konsole به راحتی با برنامه‌های دیگر KDE ارتباط برقرار کنند و تجربه‌ی یکپارچه‌تری داشته باشند. Konsole با قابلیت‌های پیشرفته و قابلیت‌های سفارشی‌سازی گسترده، یک ابزار قدرتمند برای مدیریت ترمینال و اجرای دستورات مختلف است. این ترمینال با توجه به امکانات KDE و ادغام با سایر برنامه‌های آن، یک گزینه محبوب برای کاربران این محیط دسکتاپ است.

ویژگی های کنسول:

- تب های چند زبانه.
- پس زمینه های شفاف
- پشتیبانی از حالتSplit-view
- نشانک فهرست وSSH.
- طرح های رنگی قابل تنظیم
- اتصال کلید قابل تنظیم
- هشدارهای اعلان در مورد فعالیت در ترمینال.
- جستجوی افزایشی
- پشتیبانی از فایل منیجرDolphin

export خروجی در قالب متن ساده یاHTML.



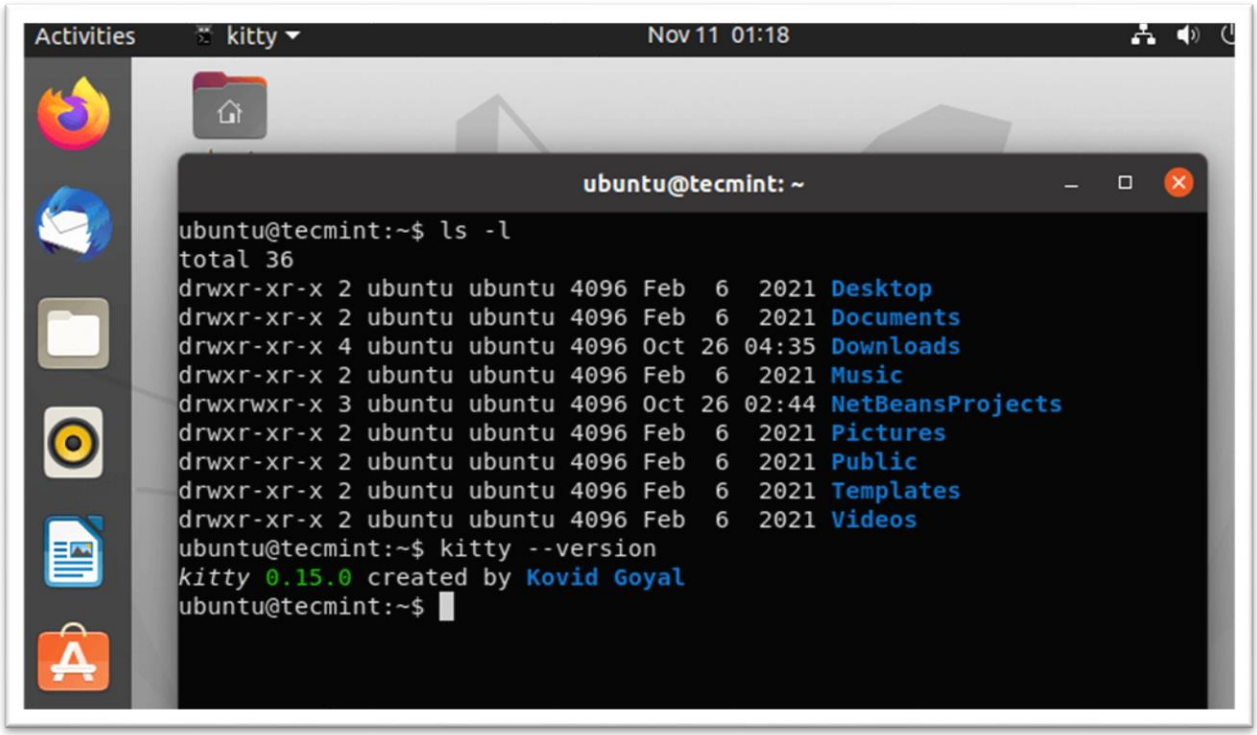
ترمینال کنسول Konsole

برای نصب **Konsole** در لینوکس، همانطور که نشان داده شده است از مدیر بسته پیش فرض خود استفاده کنید.

```
$ sudo apt install konsole [ در دبیان، اوبونتو و مینت]
$ sudo yum install konsol (RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux ]
$ sudo emerge -a kde-apps/konsole [ در جنتو لینوکس]
$ sudo pacman -S konsole [ در آرچ لینوکس]
$ sudo zypper install konsole در] کنسول OpenSUSE ]
```

ترمینال Kitty

Kitty یک شبیه ساز ترمینال آزاد، منبع باز و سریع، غنی از ویژگی های گرافیکی برای لینوکس است که از تمام ویژگی های ترمینال امروزی مانند یونیکد، رنگ واقعی، قالب بندی متن، فونت های پررنگ/مورب، کاشی کاری چندگانه پشتیبانی می کند. پنجره ها و تب ها و غیره. Kitty به زبان های برنامه نویسی C و Python نوشته شده است و یکی از معدود شبیه سازهای ترمینال با پشتیبانی از GPU به همراه Alacritty است.



ترمینال کیتی برای لینوکس

برای نصب **Kitty** در لینوکس، همانطور که نشان داده شده است از مدیر بسته پیش فرض خود استفاده کنید.

```
$ sudo apt install kitty [در دبیان، اوبونتو و مینت]
$ sudo yum install kitty [در RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux ]
$ sudo emerge -a x11-terms/kitty [در لینوکس جنتو]
$ sudo pacman -S kitty [در آرچ لینوکس]
$ sudo zypper install kitty [در OpenSUSE ]
```

ترمینال ST

stیک اجرای ترمینال ساده برای X Window است. st یک اجرای ترمینال ساده برای X Window است. این ترمینال با تمرکز بر سادگی و کارایی طراحی شده است و امکانات اساسی یک ترمینال را فراهم می‌کند.

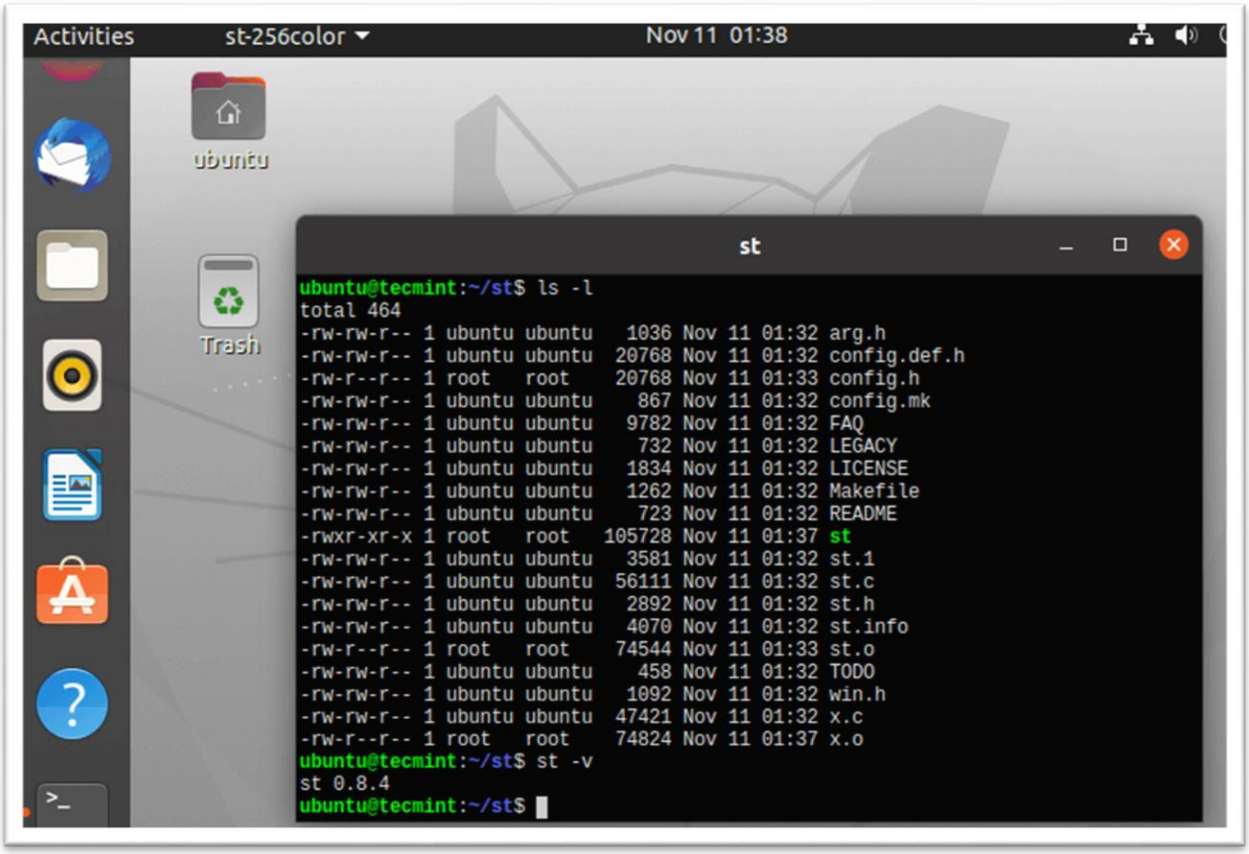
از جمله ویژگی‌های st می‌توان به موارد زیر اشاره کرد:

سادگی: st با رابط کاربری ساده و کاربر پسند خود، تجربه استفاده آسان از ترمینال را فراهم می‌کند.

سرعت: به عنوان یک ترمینال سبک، st با سرعت بالا و عدم اضافه‌وزنی که ممکن است سبب کاهش کارایی شود، عملکرد خوبی را ارائه می‌دهد.

قابلیت سفارشی‌سازی: این ترمینال امکاناتی برای سفارشی‌سازی شامل تنظیمات فونت، رنگ‌ها و دیگر تنظیمات را به کاربر ارائه می‌دهد.

با توجه به سادگی و کارایی st، این ترمینال یک گزینه مناسب برای کاربرانی است که به دنبال تجربه ترمینال سبک و کارآمد هستند.



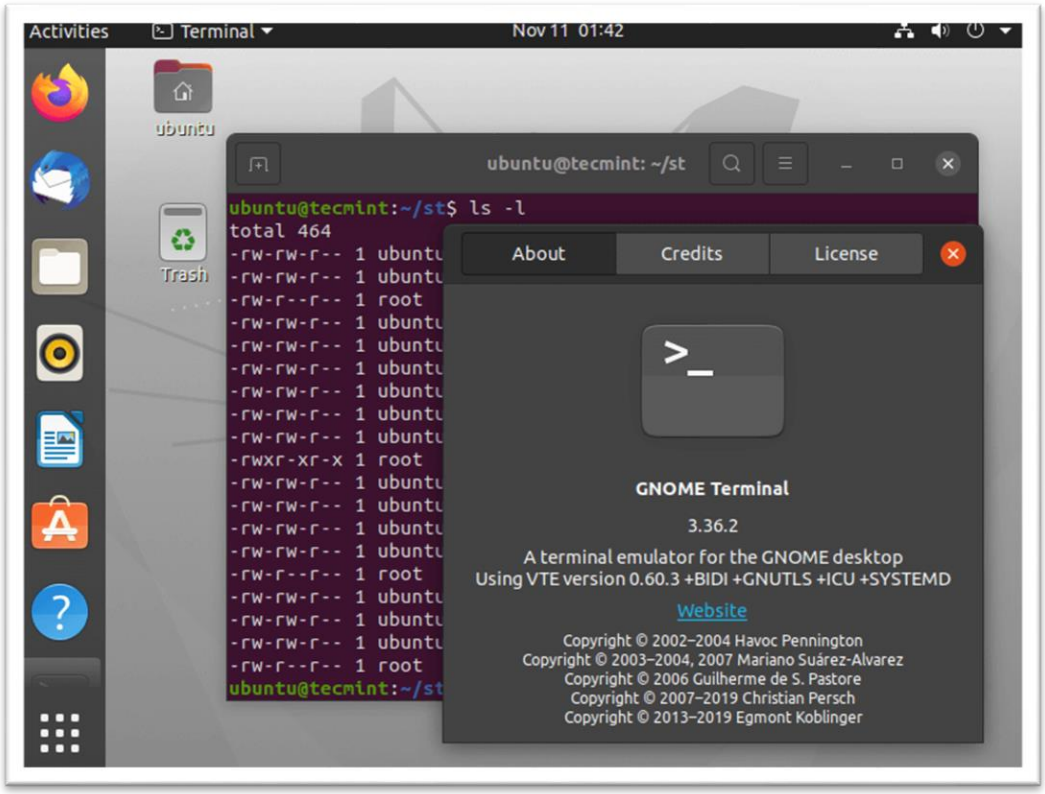
St Terminal برای لینوکس

برای نصب **st** terminal در لینوکس، مطابق شکل از مدیر بسته پیش فرض خود استفاده کنید.

```
git clone https://git.suckless.org/st
```

ترمینال Gnome-Terminal

ترمینال گنوم یک شبیه ساز ترمینال داخلی برای **محیط** دسکتاپ **گنوم** است که توسط **Havoc Pennington** و دیگران توسعه یافته است .این به کاربران اجازه می دهد تا دستورات را با استفاده از یک پوسته واقعی لینوکس در حالی که در محیط گنوم هستند اجرا کنند .ترمینال گنوم شبیه ساز ترمینال **xterm** را شبیه سازی میکند و چند ویژگی مشابه را به ارمغان می آورد. ترمینال Gnome از چندین پروفایل پشتیبانی می کند، که در آن کاربران می توانند چندین پروفایل برای حساب خود ایجاد کنند و می توانند **گزینه** های پیکربندی مانند فونت ها، رنگ ها، تصاویر پس زمینه، رفتار و غیره را در هر حساب شخصی سازی کنند و برای هر پروفایل یک نام تعریف کنند .همچنین از رویدادهای ماوس، تشخیص URL ، چندین تب و غیره پشتیبانی می کند.



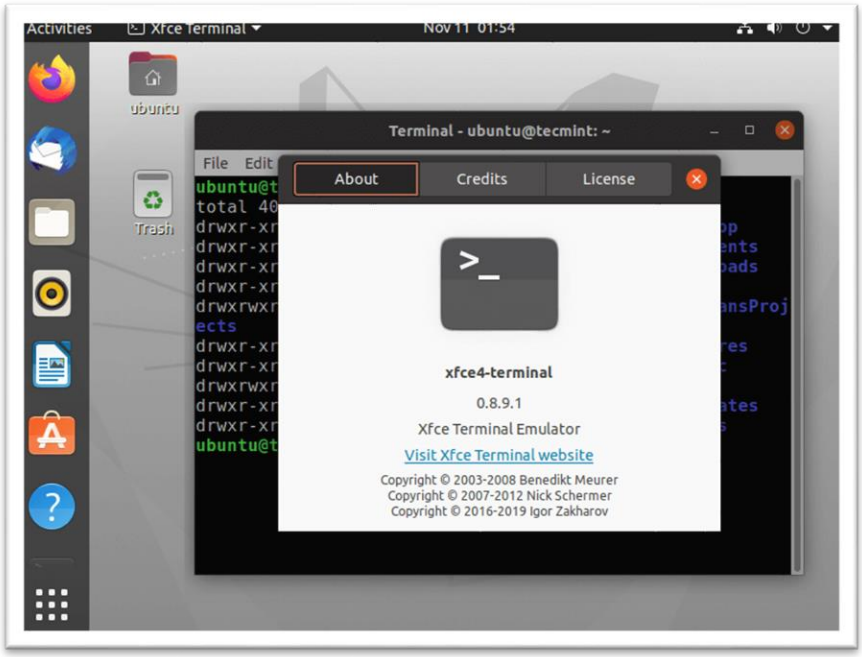
ترمینال Gnome برای لینوکس

برای نصب **Gnome-Terminal** در لینوکس، همانطور که نشان داده شده است از مدیر بسته پیش فرض خود استفاده کنید.

```
$ sudo apt install gnome-terminal [ در دبیان، اوبونتو و مینت]
$ sudo yum install gnome-terminal [ در RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux ]
$ sudo emerge -a x11-terms/gnome-terminal [ در جنتو لینوکس]
$ sudo pacman -S gnome-terminal [ در آرچ لینوکس]
$ sudo zypper install  gnome-terminal [ در OpenSUSE ]
```

ترمینال xfce4-terminal

xfce4-terminal یک شبیه ساز ترمینال سبک وزن و مدرن و با کاربری آسان است که مخصوص محیط دسکتاپ Xfce طراحی شده است. آخرین نسخه ترمینال xfce دارای برخی ویژگی های جالب جدید مانند جستجو، تغییر رنگ tab، کنسول کشویی مانند Guake یا Yakuake و بسیاری موارد دیگر است.



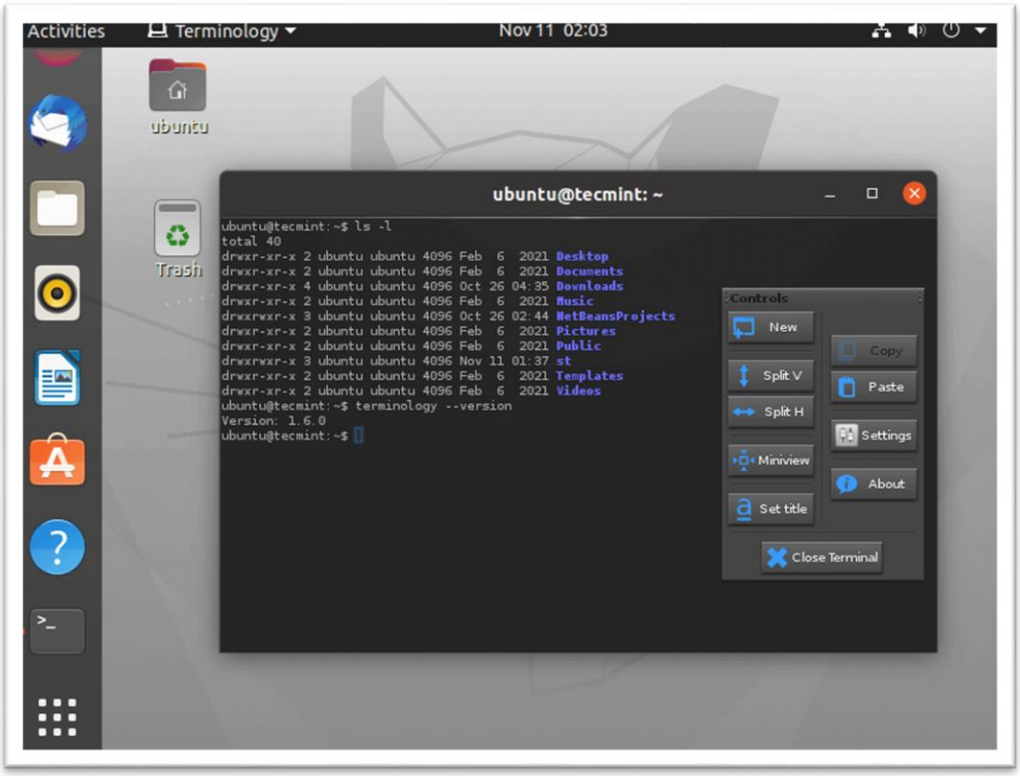
ترمینال Xfce برای لینوکس

برای نصب **Xfce Terminal** در لینوکس، همانطور که نشان داده شده است از مدیر بسته پیش فرض خود استفاده کنید.

```
$ sudo apt install xfce4-terminal [در دبیان، اوبونتو و مینت]
$ sudo yum install  xfce4-terminal [در RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux]
$ sudo emerge -a x11-terms/xfce4-terminal [در جنتو لینوکس]
$ sudo pacman -S xfce4-terminal [در آرچ لینوکس]
$ sudo zypper install  xfce4-terminal [در OpenSUSE]
```

ترمینال Terminology

Terminology یکی دیگر از شبیه سازهای ترمینال مدرن جدید است که برای دسکتاپ **Enlightenment** ایجاد شده است، اما همچنین می تواند در محیط های دسکتاپ مختلف استفاده شود. این دارای ویژگی های منحصر به فرد فوق العاده ای است که در هیچ شبیه ساز ترمینال دیگری وجود ندارد. جدا از ویژگی‌ها، Terminology چیزهای بیشتری را ارائه می‌کند مانند پیش‌نمایش ریز عکس‌ها، ویدیوها و اسناد، همچنین به شما امکان می‌دهد آن فایل‌ها را مستقیماً از Terminology ببینید.



ترمینال Terminology برای لینوکس

برای نصب **Terminology** در لینوکس، همانطور که نشان داده شده است از مدیر بسته پیش فرض خود استفاده کنید.

```
$ sudo apt install Terminology [ در دبیان، اوبونتو و مینت]
$ sudo yum install  Terminology [ در RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux ]
$ sudo emerge -a x11-terms/terminology [ در لینوکس جنتو]
$ sudo pacman –S  Terminology [ در آرچ لینوکس]
 $ sudo zypper install  Terminology [ در OpenSUSE ]
```

ترمینال Deepin Terminal

Deepin Terminal یک شبیه‌ساز ترمینال پیشرفته است که ویژگی‌های قدرتمندی را ارائه می‌دهد که شامل فضای کاری، چندین پنجره، آپلود و دانلود فایل‌ها با مدیریت از راه دور، حالت لرزش و سایر ویژگی‌های قدرتمندی است. Deepin Terminal یک شبیه‌ساز ترمینال پیشرفته است که ویژگی‌های قدرتمندی را ارائه می‌دهد. این ترمینال با امکانات پیشرفته‌ای که فراهم می‌کند، تجربه استفاده از ترمینال را بهبود می‌بخشد.

بین ویژگی‌های قدرتمند Deepin Terminal می‌توان به موارد زیر اشاره کرد:

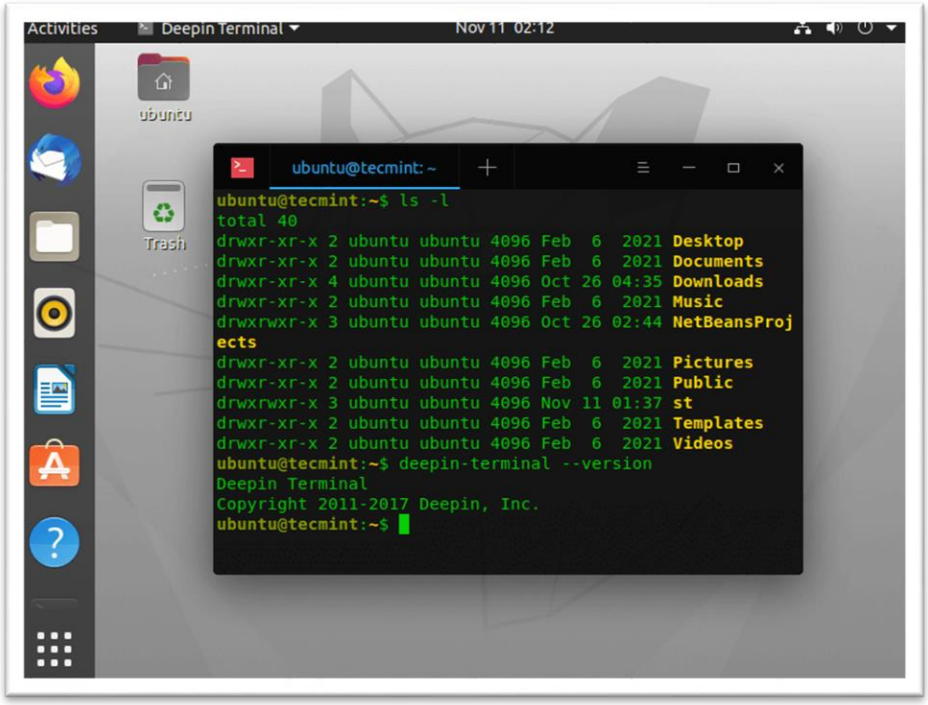
فضای کاری: Deepin Terminal امکان فراهم کردن فضای کاری مناسب برای کاربران را فراهم می‌کند تا بتوانند به راحتی با برنامه‌ها و فایل‌ها کار کنند.

چندین پنجره: این ترمینال امکان باز کردن و مدیریت چندین پنجره ترمینال را فراهم می‌کند که کاربران می‌توانند به صورت همزمان با چندین کار کار کنند.

آپلود و دانلود فایل‌ها با مدیریت از راه دور: Deepin Terminal امکان آپلود و دانلود فایل‌ها با مدیریت از راه دور را فراهم می‌کند که این امکان برای کاربرانی که به اشتراک گذاری فایل‌ها از راه دور نیاز دارند بسیار مفید است.

حالت لرزش و سایر ویژگی‌های قدرتمند: Deepin Terminal با ویژگی‌هایی مانند حالت لرزش و سایر امکانات پیشرفته، یک تجربه کاربری پیشرفته در استفاده از ترمینال ارائه می‌دهد.

با توجه به این ویژگی‌های قدرتمند، Deepin Terminal یک گزینه عالی برای کاربرانی است که به دنبال ترمینالی پیشرفته و کارآمد هستند.



Deepin Terminal برای لینوکس

برای نصب **Deepin Terminal** در لینوکس، همانطور که نشان داده شده است از مدیر بسته پیش فرض خود استفاده کنید.

```
$ sudo apt install deepin-terminal [ در دبیان، اوبونتو و مینت ]
$ sudo yum install deepin-terminal [RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux ]
$ sudo emerge -a x11-terms/deepin-terminal [ در جنتو لینوکس]
$ sudo pacman -S deepin-terminal [ در آرچ لینوکس]
$ sudo zypper install deepin-terminal [ در OpenSUSE ]
```

ترمینال xterm

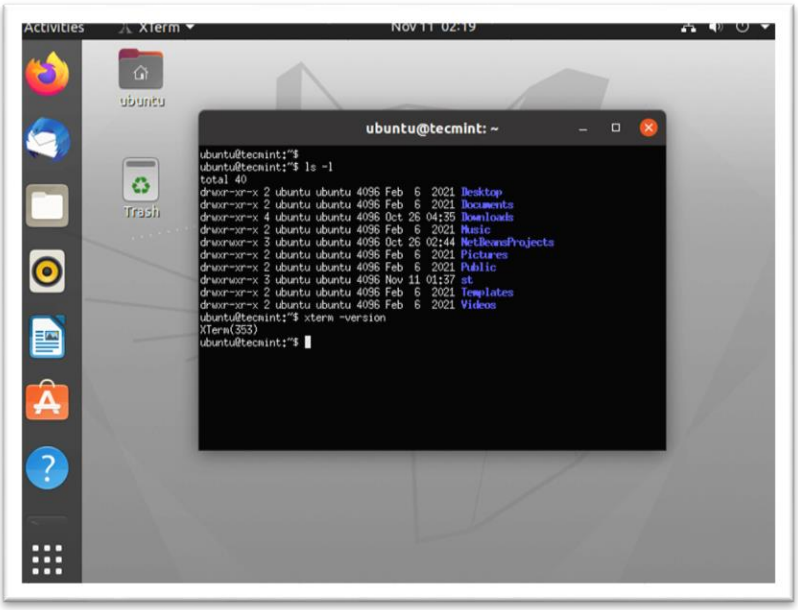
ترمینال xterm یک شبیه ساز ترمینال استاندارد برای سیستم پنجره X این است. ترمینال xterm یک شبیه‌ساز ترمینال استاندارد برای سیستم پنجره X است. این ترمینال یکی از قدیمی‌ترین و پرکاربردترین شبیه‌سازهای ترمینال برای سیستم‌های X Window است. بین ویژگی‌های xterm می‌توان به موارد z زیر اشاره کرد:

استاندارد: xterm یک ترمینال استاندارد است که در بسیاری از توزیع‌های لینوکس و سیستم‌های دیگر موجود است.

سادگی: این ترمینال با رابط کاربری ساده و کارآمد خود، تجربه استفاده راحتی را به کاربران ارائه می‌دهد.

پشتیبانی از ویژگی‌های پیشرفته: xterm از ویژگی‌های پیشرفته‌ای مانند پشتیبانی از فونت‌های مختلف، مدیریت پنجره‌ها و تنظیمات سفارشی پشتیبانی می‌کند.

از آنجایی که xterm یک ترمینال استاندارد و پرکاربرد است، بسیاری از کاربران لینوکس و سیستم‌های دیگر از آن استفاده می‌کنند برای انجام وظایف مختلف در محیط خط فرمان.



ترمینال Xterm برای لینوکس

برای نصب **Xterm** در لینوکس، **مطابق شکل** از مدیریت بسته پیش فرض خود استفاده کنید.

```
$ sudo apt install xterm [ در دبیان، اوبونتو و مینت]
$ sudo yum install xterm [ در RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux ]
$ sudo emerge -a x11-terms/xterm [ در لینوکس جنتو]
$ sudo pacman -S xterm [ در آرچ لینوکس]
$ sudo zypper install xterm [ در OpenSUSE ]
```

ترمینال LilyTerm

LilyTermیکی دیگر از منبع باز کمتر شناخته شده شبیه ساز ترمینال بر اساس libvte است که سبک و سریع است.LilyTerm یکی دیگر از منابع باز کمتر شناخته شده از شبیه‌سازهای ترمینال است که مبنی libvte بر پایه ساخته شده است. این ترمینال به دلیل سبکی و سرعت بالای خود، جذابیت خاصی برای کاربران دارد.

بین ویژگی‌های LilyTerm می‌توان به موارد زیر اشاره کرد:

پایه libvte: استفاده از libvte به LilyTerm امکان فراهم کردن ویژگی‌های پیشرفته ترمینال و امکانات متنوع را فراهم می‌کند.

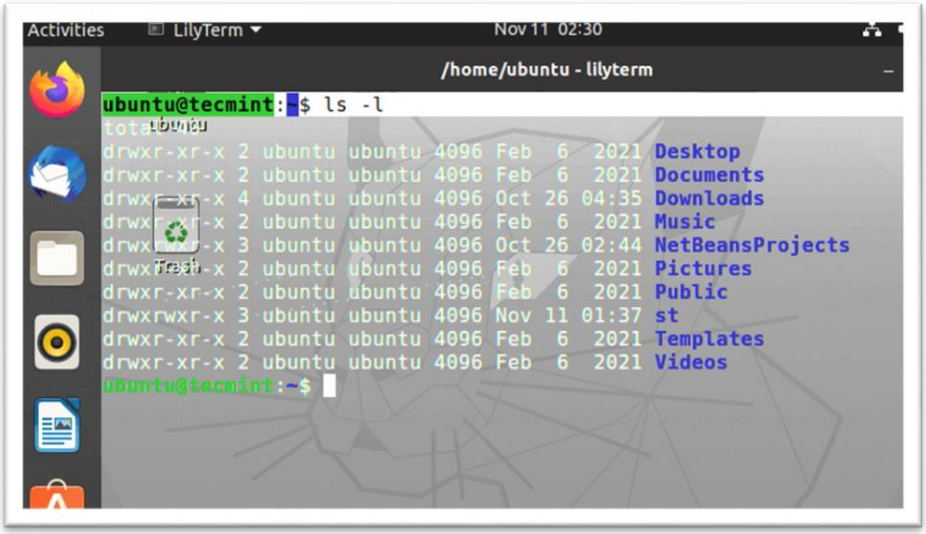
سبک و سریع: LilyTerm به دلیل سبکی و سرعت بالای خود، به کاربران این امکان را می‌دهد که به راحتی با ترمینال کار کنند و وظایف خود را به سرعت انجام دهند.

منبع باز: این ترمینال یک منبع باز است که به ارتقاء و بهبود آن توسط جامعه کاربری کمک می‌کند.

از آنجایی که LilyTerm یک ترمینال سبک، سریع و بر پایه libvte است، می‌تواند یک گزینه عالی برای کاربرانی با نیازهای خاص در استفاده از ترمینال باشد.

ویژگی های کلیدی:

- پشتیبانی از tabها، رنگ‌آمیزی و مرتب‌سازی مجدد tabها
- امکان مدیریت tab ها از طریقkeybinding
- پشتیبانی از شفافیت و اشباع پس زمینه.
- پشتیبانی از ایجاد پروفایل خاص کاربر
- چندین گزینه سفارشی سازی برای پروفایل ها
- پشتیبانی گسترده.UTF-8



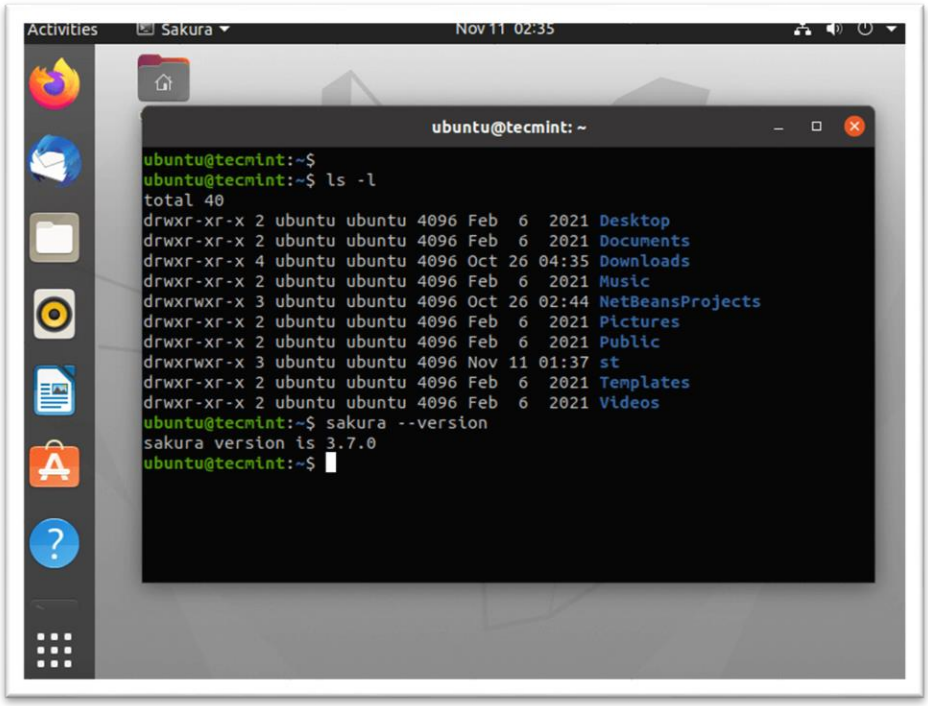
ترمینال LilyTerm برای لینوکس

برای نصب **LilyTerm** در لینوکس، همانطور که نشان داده شده است از مدیر بسته پیش فرض خود استفاده کنید.

```
$ sudo apt install lilyterm [ در دبیان، اوبونتو و مینت ]
$ sudo yum install lilyterm [ در RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux ]
$ sudo emerge -a x11-terms/lilyterm [ در لینوکس جنتو ]
$ sudo pacman -S lilyterm [ در آرچ لینوکس ]
$ sudo zypper install lilyterm [ در OpenSUSE ]
```

ترمینال sakura

ساکورا دیگر کمتر شناخته شده به سبک یونیکس شبیه ساز ترمینال توسعه یافته برای خط فرمان و همچنین برنامه ترمینال های مبتنی بر متن است .ساکورا مبتنی بر GTK و livte است و نه ویژگی‌های پیشرفته‌تری، بلکه گزینه‌های سفارشی‌سازی مانند پشتیبانی از چندین تب، رنگ متن سفارشی، فونت و تصاویر پس‌زمینه، پردازش سریع دستورات و چند مورد دیگر را ارائه می‌کند.



ترمینال ساکورا برای لینوکس

برای نصب **ساکورا** در لینوکس، مطابق شکل از مدیر بسته پیش فرض خود استفاده کنید.

```
$ sudo apt install sakura [ در دبیان، اوبونتو و مینت]
$ sudo yum install sakura [RHEL/CentOS/Fedora وRocky Linux/AlmaLinux ]
$ sudo emerge -a x11-terms/sakura [ در لینوکس جنتو]
$ sudo pacman -S sakura [ در آرچ لینوکس]
$ sudo zypper install sakura [ OpenSUSE ]
```

ترمینال Extraterm

Extratermیک شبیه ساز ترمینال مدرن آزاد و منبع باز است که هدف آن ارائه بسیاری از ویژگی های جدید برای آوردن ترمینال استاندارد به عصر مدرن است.Extraterm یک شبیه‌ساز ترمینال مدرن، آزاد و منبع باز است که هدف آن ارائه بسیاری از ویژگی‌های جدید برای بهبود تجربه استفاده از ترمینال استاندارد در عصر مدرن است.

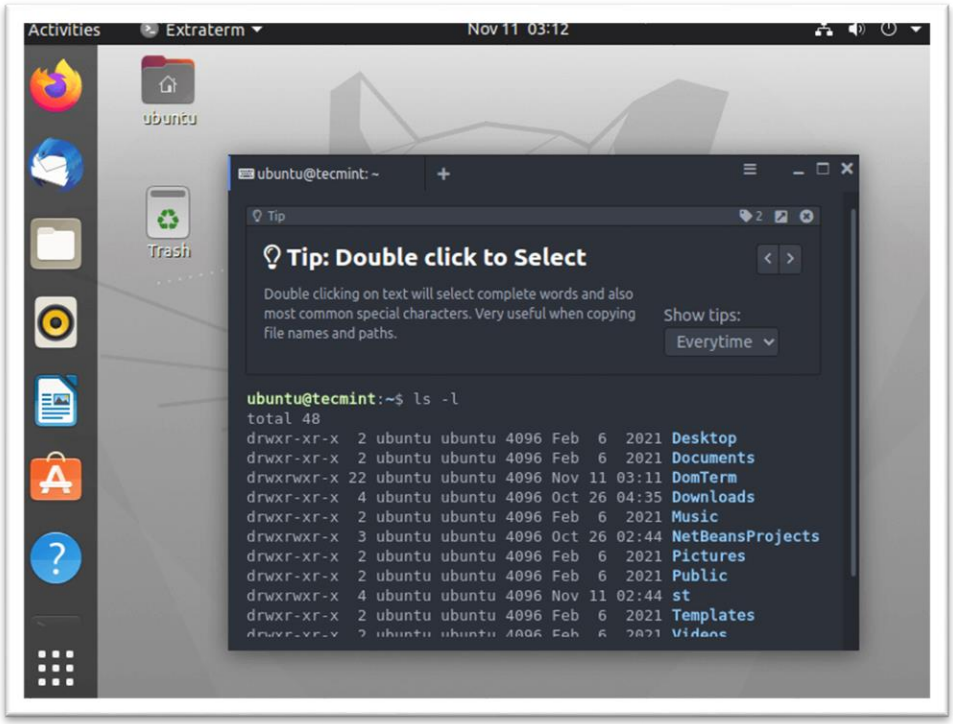
بین ویژگی‌های Extraterm می‌توان به موارد زیر اشاره کرد:

مدرن: Extraterm با ارائه ویژگی‌ها و امکانات مدرن، به کاربران امکان می‌دهد تا تجربه بهتری از استفاده از ترمینال داشته باشند.

ویژگی‌های جدید: این ترمینال با اضافه کردن ویژگی‌های جدید، امکانات بیشتری را به کاربران ارائه می‌دهد و از ترمینال استاندارد به یک سطح بالاتر ارتقا می‌دهد.

آزاد و منبع باز: Extraterm به عنوان یک پروژه منبع باز، به جامعه کاربری امکان می‌دهد که در بهبود و توسعه آن شرکت کنند و از آن بهره‌مند شوند.

از آنجایی که Extraterm تلاش دارد تا ترمینال استاندارد را به عصر مدرن بیاورد و ویژگی‌های جدید ارائه دهد، می‌تواند یک گزینه جذاب برای کاربران حرفه‌ای و علاقمند به تکنولوژی باشد.



ترمینال Extraterm برای لینوکس

ترمینالDomTerm

DomTermیک شبیه‌ساز ترمینال است که از ویژگی‌های قدرتمندی مانند صفحه‌بندی خودکار، مدیریت رویداد ماوس، **چندگانه‌سازی** صفحه ،tab ها و مدیریت جلسه[۱] مانند صفحه **tmux**یا GNUپشتیبانی می‌کند.DomTerm یک شبیه‌ساز ترمینال قدرتمند است که از ویژگی‌های متنوع و قدرتمندی برخوردار است. برخی از این ویژگی‌ها عبارتند از:

صفحه‌بندی خودکار: DomTerm قابلیت صفحه‌بندی خودکار را داراست که به کاربران امکان می‌دهد به راحتی واحدهای مختلفی از خروجی ترمینال را مشاهده کنند.

مدیریت رویداد ماوس: این ویژگی به کاربران امکان می‌دهد با استفاده از موس، به راحتی در ترمینال حرکت کنند و عملیات مختلفی را انجام دهند.

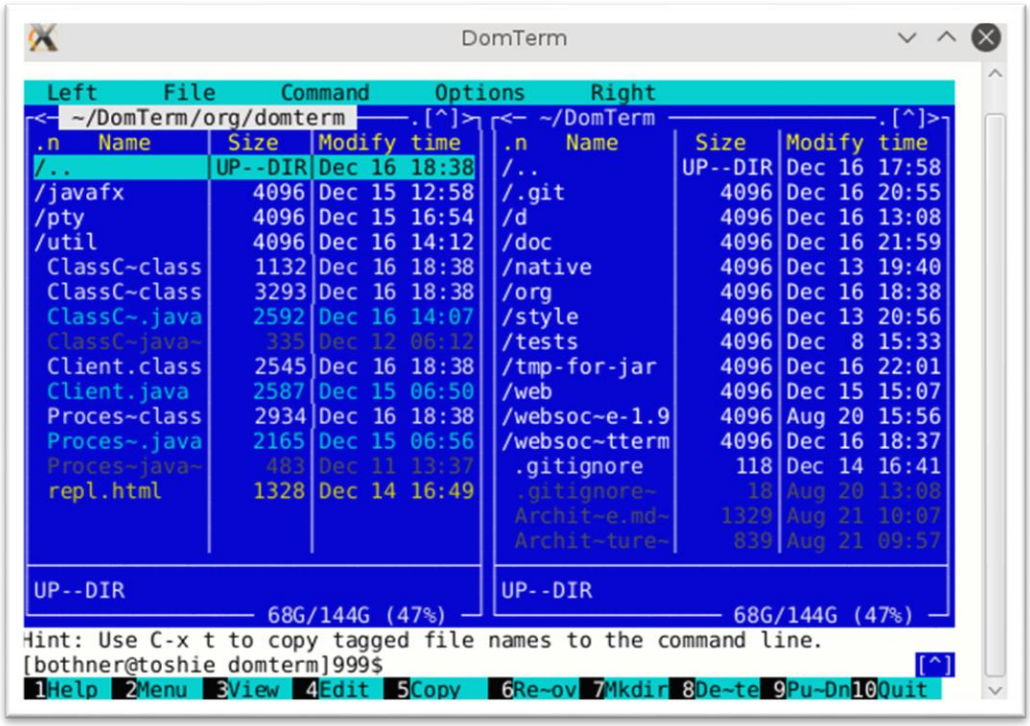
چندگانه‌سازی صفحه: DomTerm امکان چندگانه‌سازی صفحه را فراهم می‌کند که به کاربران اجازه می‌دهد چندین صفحه ترمینال را همزمان مدیریت کنند.

پشتیبانی از tabها و مدیریت جلسه: DomTerm از tabها و مدیریت جلسه مانند صفحه tmux یا GNU نیز پشتیبانی می‌کند که به کاربران امکان می‌دهد به راحتی و به صورت سازمان‌یافته با ترمینال کار کنند.

با توجه به ویژگی‌های قدرتمند DomTerm، این ترمینال می‌تواند یک گزینه عالی برای کاربرانی با نیازهای پیچیده در مدیریت ترمینال باشد.

---

[۱] session



ترمینال Domterm برای لینوکس

ترمینالTermKit

TermKitیک ترمینال زیبا است که هدف آن ایجاد جنبه‌های رابط کاربری گرافیکی با برنامه مبتنی بر خط فرمان با استفاده از موتور رندر WebKit است که بیشتر در مرورگرهای وب مانند Google Chrome و Chromium استفاده می‌شود. TermKit در اصل برای مک و ویندوز طراحی شده است، اما به دلیل انجام فورک TermKit توسط Floby که اکنون می توانید آن را تحت توزیع های مبتنی بر لینوکس نصب کنید و قدرت TermKitرا تجربه کنید.



ترمینال TermKit برای لینوکس

منابع بیشتر درباره ترمینالها

xterm:
xterm on Wikipedia: https://en.wikipedia.org/wiki/Xterm
Homepage for the current maintainer of xterm, Thomas Dickey:

http://invisible-island.net/xterm/

Tektronix 4014:
Tektronix 4014 on Wikipedia: https://en.wikipedia.org/wiki/Tektronix_4010
Some background on the 4014 at Chilton Computing: http://www.chilton-computing.org.uk/acd/icf/terminals/p005.htm

rxvt:
Home page for rxvt: http://rxvt.sourceforge.net/

urxvt (rxvt-Unicode):
Home page for the rxvt-Unicode project: http://software.schmorp.de/pkg/rxvt-unicode.html

gnome-terminal:
Help pages for gnome-terminal: https://help.gnome.org/users/gnome-terminal/stable/

konsole:
The Konsole Manual at the KDE Project:
https://docs.kde.org/stable5/en/applications/konsole/index.html

guake:
The home page for the guake project: http://guake-project.org/
The Arch Wiki entry for guake (contains a lot of useful information but some is Arch Linux specific): https://wiki.archlinux.org/index.php/Guake

terminator:
The home page for the terminator project:
http://gnometerminator.blogspot.com/p/introduction.html

Connectbot:
Connectbot at the Google Play Store:
https://play.google.com/store/apps/details?id=org.connectbot&hl=en

Hacker's Keyboard:
Hacker's Keyboard at the Google Play Store:
https://play.google.com/store/apps/details?id=org.pocketworkstation.pckeyboard&hl=en

Termux:
Termux at the Google Play Store:
https://play.google.com/store/apps/details?id=com.termux&hl=en

کار با ترمینالها

یکی از سیستم عامل های اولیه یونیکس نام داشت. این برنامه به گونه ای طراحی شده است که به عنوان یک سیستم چند کاربره در رایانه های اصلی استفاده می شود و کاربران از راه دور از طریق پایانه های جداگانه به آن متصل می شوند. این ترمینال ها با استانداردهای مدرن کاملاً اساسی بودند: فقط یک صفحه کلید و صفحه نمایش، بدون اینکه بتوانید برنامه ها را به صورت محلی اجرا کنید. درعوض، آنها فقط کلیدها را به سرور ارسال می کردند و هر داده ای را که می گرفتند روی صفحه نمایش می دادند. ماوس، گرافیک فانتزی و حتی انتخاب رنگی وجود نداشت. همه چیز به عنوان متن ارسال شد و به عنوان متن دریافت شد. بدیهی است، بنابراین، هر برنامه ای که روی سیستم اصلی اجرا شود ، باید متن را به عنوان یک خروجی تولید کند و متن را به عنوان ورودی بپذیرد. در مقایسه با گرافیک، متن از نظر مصرف منابع بسیار کمتر است. کاربران قادر به تعامل سریع و کارآمد با برنامه ها هستند. دستورات نیز بسیار کوتاه نوشته شدند تا تعداد کلیدهای مورد نیاز را کاهش دهند و استفاده مردم از ترمینال را بیش از پیش تسریع کنند. این سرعت و کارایی یکی از دلایلی است که امروزه همچنان از این رابط متن به طور گسترده استفاده می شود.

هنگامی که از طریق یک ترمینال به سیستم اصلی Unix وارد می شوید، کاربران هنوز مجبور به مدیریت نوعی از وظایف مدیریت پرونده[۱] هستند که ممکن است اکنون با یک موس و چند پنجره انجام دهید. اعم از ایجاد فایل ، تغییر نام آنها ، قرار دادن آنها در زیرشاخه ها یا جابجایی آنها روی دیسک ، کاربران دهه ۷۰ می توانند همه کارها را کاملاً با یک رابط متنی انجام دهند. هر یک از این کارها به برنامه یا دستور خاص خود احتیاج داشتند: یکی برای تغییر فهرست ها (cd) ، دیگری برای فهرست کردن محتویات آنها (ls) ، دیگری برای تغییر نام یا انتقال پرونده ها (mv) و غیره. کاربران حتی می توانند کدهای ساده ای بنام "اسکریپت های پوسته[۲]" بنویسند که می تواند برای ساده سازی کارهای پیچیده از دستورات پوسته طولانی استفاده کند. برنامه اصلی پوسته یونیکس فقط sh نامیده می شد ، اما این برنامه در طول سالها گسترش یافته و تحت نظارت قرار گرفته است ، بنابراین در یک سیستم مدرن لینوکس به احتمال زیاد از پوسته ای به نام bash استفاده می کنید.

لینوکس نوعی از نسل یونیکس است. قسمت اصلی لینوکس برای رفتار مشابه سیستم Unix طراحی شده است ، به گونه ای که اکثر پوسته های قدیمی و سایر برنامه های مبتنی بر متن کاملاً با خوشحالی روی آن اجرا می شوند. از لحاظ تئوری حتی می توانید یکی از آن ترمینال های قدیمی دهه ۱۹۷۰ را به جعبه مدرن لینوکس متصل کنید و از این طریق به پوسته دسترسی پیدا کنید. اما این روزها استفاده از ترمینال نرم افزاری بسیار رایج تر است: همان رابط متنی قدیمی به سبک یونیکس ، اما در کنار پنجره در کنار برنامه های گرافیکی شما اجرا می شود.

بازکردن ترمینال

در اکثر توزیع‌های لینوکسی دارای مدیر پنجره[۳] هستند که یک راه برای راه اندازی یک ترمینال از طریق منو دارند. میتوانید از طریق لیست برنامه ها ببینید که آیا چیزی شبیه یک شبیه ساز ترمینال وجود دارد؟به عنوان مثلا اگر یک کاربر KDE هستید، برنامه ترمینال "konsole"نام دارد، در Gnome آن را "gnome-terminal" نامگذاری می کند .یا در میزهای xfce آن را xfce4termianl می نامند. در حالی که تعدادی از شبیه سازهای مختلف ترمینال وجود دارد، همه آنها کار یکسانی را انجام می دهند. یعنی آنها به یک پوسته[۴] دسترسی دارند.

---

۱ file manager
۲ Shell script
۳ file manager
۴ session shell

روش اول:

همانگونه که در بالا گفته شد، میتوانید بافشردن کلیدsuper key در سیستم عامل ویندوز به کلید ویندوز اطلاق میشود) منوی برنامه ها را بازکنید و عبارت ترمینال را تایپ نمایید. تابرنامه ترمینال همان میزکاری شما اجرا شود. به عنوان مثال در توزیع اوبونتو:



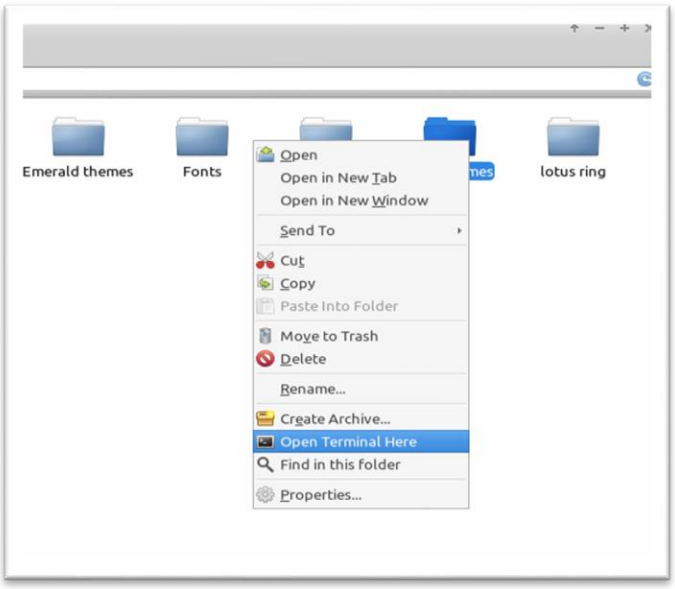
روش دوم:

اکثر سیستم های لینوکس از میانبر صفحه کلید زیر به عنوان کلید میانبر پیش فرض برای بازکردن آن استفاده می کنند:

Ctrl+Alt+T

روش سوم:

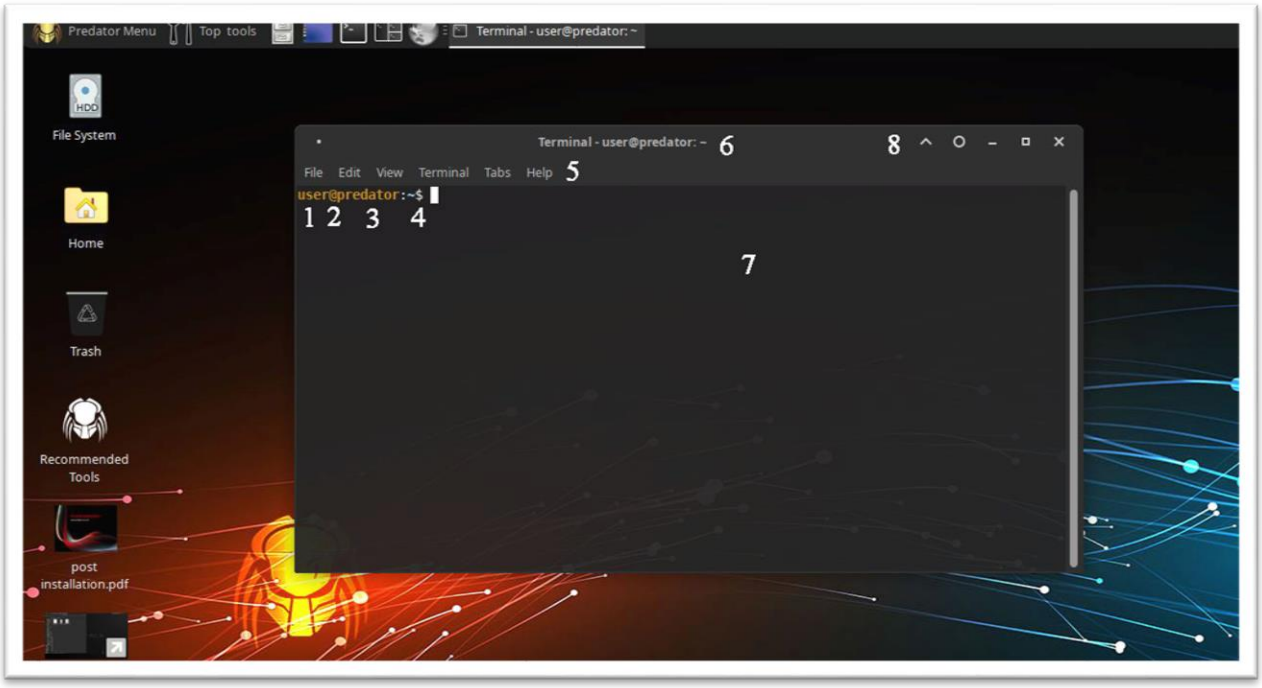
در بسیاری از توزیع های لینوکسی با راست کلیک کردن روی میزکار[1] یا داخل هر پنجره ای که باز است،گزینه ای به نام open terminal hereوجود دارد.



اجزای پنجره ترمینال

پنجره ی ترمینال را بازکنید. در ابتدا قسمت هایی را به صورت پیش فرض و همیشگی در آن مشاهده خواهید کرد. مشخصاتی مانند:

[1] desktop



۱.نام کاربر جاری[۱]

۲.علامت @ جداکننده نام کاربری و نام سیستم شما

۳. نام سیستم شما[۲]

۴.علامت $ یا علامت # مکانی که دستورات خود را تایپ مینمایید.

علامت $نشان دهنده کاربر عالی فعال است.و دسترسی شما کاربر نرمال است.

علامت # نشان دهنده کاربر ریشه فعال است.

۵. نوار منو

۶. نوار عنوان ترمینال که قابل تغییر هم می باشد.(ازطریق تنظیمات ترمینال)

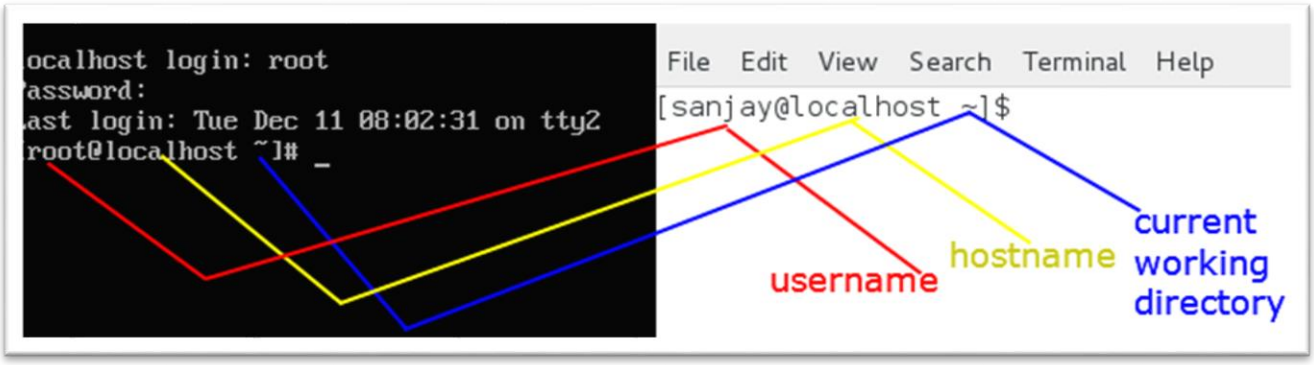
۷.بدنه ترمینال.فضایی که خروجی دستورات مشاهده میشود.

۸. دکمه های کنترل پنجره ترمینال

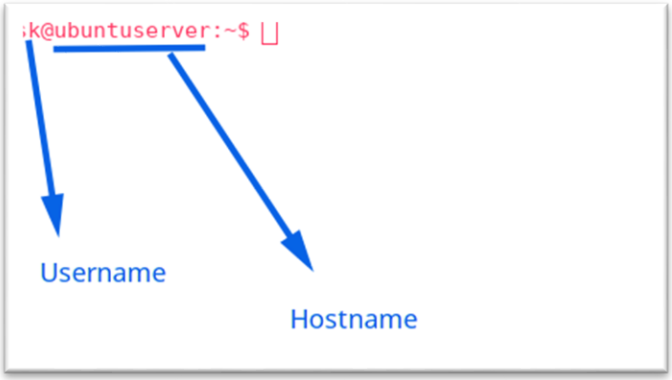
درابتدا که ترمینال خود را راه اندازی می کنید ، ممکنه با یک پنجره نسبتاً کسل کننده همراه باشید که کمی متن عجیب و غریب در بالا دارد ، بسته به سیستم لینوکس شما ممکن است رنگها یکسان نباشد ، و متن احتمالاً چیز دیگری را بیان می کند ، اما طرح کلی یک پنجره با یک متن متن بزرگ (عمدتا خالی) باید مشابه مواردی که ذکر کرده ام، باشد. اول اینکه وقتی دستور را تایپ می کنید از آن به عنوان prompt یادمی شود، یعنی درمقابل علامت $ یا #

[۱] useraccount

[۲] hostname

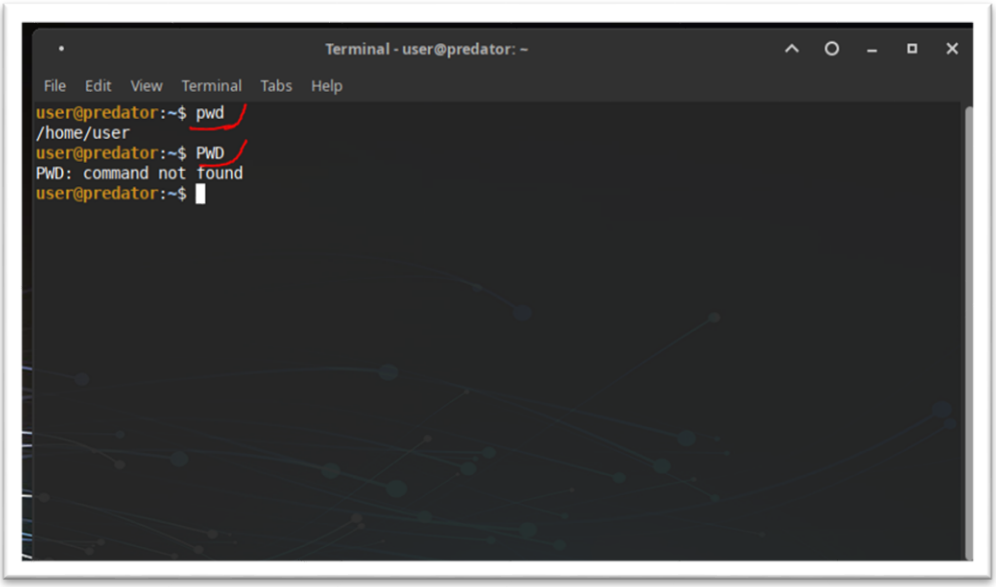


دومین چیزی که باید درک کنید این است که وقتی یک دستور را اجرا می کنید ، هر خروجی که تولید می کند ، مستقیماً در ترمینال چاپ و نمایش داده می‌شود، پس از اتمام آن یک پیام دیگر به شما نشان داده می شود. برخی از دستورات می توانند متن زیادی را چاپ کنند، برخی دیگر بی صدا کار می کنند و به هیچ وجه خروجی نخواهند داشت. اگر دستور را اجرا کردید و بلافاصله دستور دیگری ظاهر شد ، نگران نشوید ، زیرا این معمولاً به معنی موفقیت آمیز بودن دستور است.یا منتظر وارد کردن مقداری از شما می‌باشد.

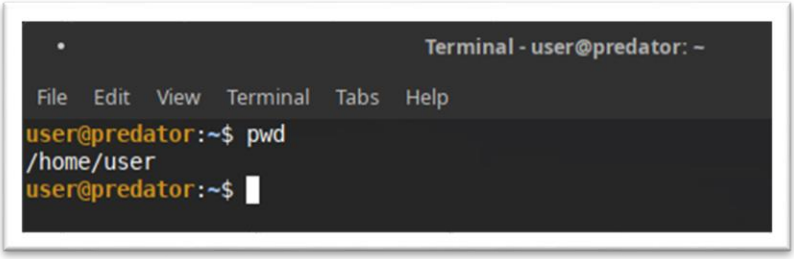


حساسیت دستورات در ترمینال

هنگام تایپ کردن در خط فرمان ترمینال، به موارد بزرگ یا کوچک تایپ کردن دستورات و مقادیر ورودی دقت کنید. تایپ کردن PWD به جای pwd خطایی ایجاد می کند، اما گاهی اوقات مورد اشتباه ممکن است منجر به اجرای دستور شود ، اما آنچه را که انتظار داشتید انجام ندهد.

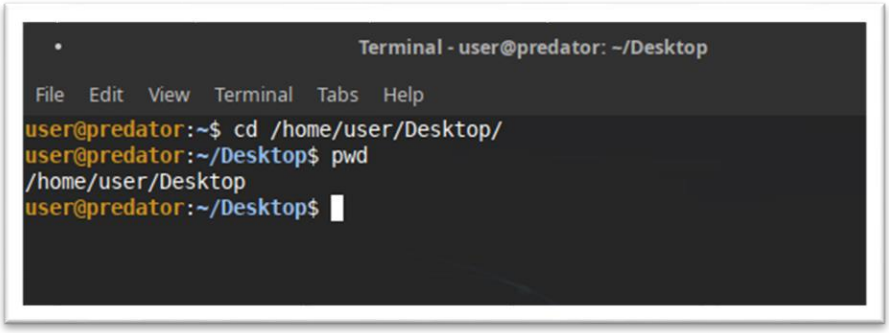


یک مفهوم دیگر که باید از آن اطلاع داشته باشید، این است که شل یک مکان پیش فرض دارد که در آن هرگونه عملیات انجام می شود.به این معنی که بعد از باز کردن ترمینال، مسیر پیش فرضی را دارد.که به آن فهرست کاری فعال[۱] گفته میشود. اگر می خواهید فایل یا فهرست جدید ایجاد کنید ، پرونده های موجود را مشاهده کنید یا حتی آنها را حذف کنید ، پوسته فرض خواهد کرد که شما آنها را در فهرست کار فعلی جستجو می کنید مگر اینکه مسیر دیگری را تعیین کنید. بنابراین کاملاً مهم است که در هر زمان مشخص در مورد پوشه ای که پوسته در آن قرار دارد ، استفاده کنید ، بعد از همه اینها ، حذف پرونده ها از پوشه اشتباه می تواند فاجعه بار باشد. اگر نیاز به دانستن مسیر جاری و کاری شل در ترمینال دارید دستور pwd دقیقاً به شما می گوید که فهرست کار فعلی شما چیست.



تغییر مسیر جاری شل در ترمینال

به منظور تغییر مسیر در ترمینال جاری باز شده از دستور cd که مخفف change directiry است، صورت می‌گیرد.



در ابتدای کار با دستورات در ترمینال سعی کنید مسیرجاری و کاری خود را با pwd مشاهده نمایید.

[۱] Proccesing working directory

میانبرهای ترمینال

کلیدهای میانبر زیر در اکثر توزیع های لینوکسی یکسان هستند.

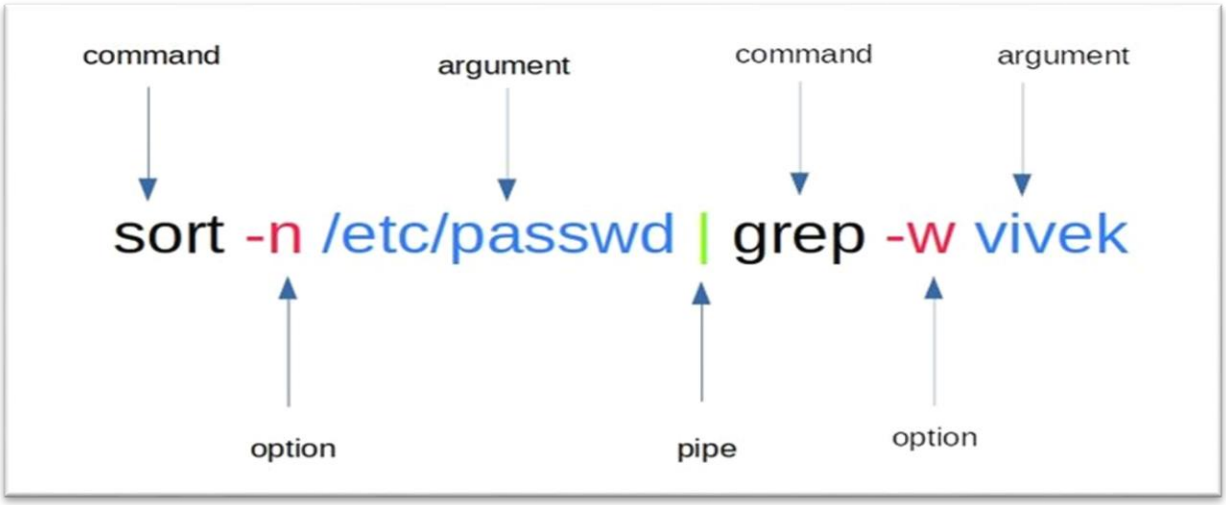
میانبر	توضیح
CTRL+A	مکان‌نما را به ابتدای خط منتقل می‌کند.
CTRL+E	مکان‌نما را به انتهای خط منتقل می‌کند.
CTRL+U	از مکان‌نما تا ابتدای خط را حذف می‌کند.
CTRL+K	از مکان‌نما تا انتهای خط را حذف می‌کند.
CTRL+W	کلمه قبل از مکان‌نما را حذف می‌کند.
CTRL+L	صفحه ترمینال را پاک می‌کند.
CTRL+C	فرآیند یا فرمان فعلی را متوقف می‌کند.
CTRL+D	از ترمینال خارج می‌شود یا از ترمینال خارج می‌شود.
CTRL+Z	روند فعلی را متوقف می‌کند (می‌توان از سر گرفت).
CTRL+R	تاریخچه فرمان را جستجو می‌کند (جستجوی به عقب).
↑	دستور قبلی را نشان می‌دهد (از تاریخچه دستورات).
↓	دستور بعدی را نشان می‌دهد (از تاریخچه دستورات).
!!	آخرین دستور را تکرار می‌کند.
!N	دستورnام را از تاریخچه تکرار می‌کند.
TAB	تکمیل خودکار دستورات، فایل‌ها یا دایرکتوری‌ها.
TABدوبار	تمام دستورات ممکن را فهرست می‌کند.

فصل دوم

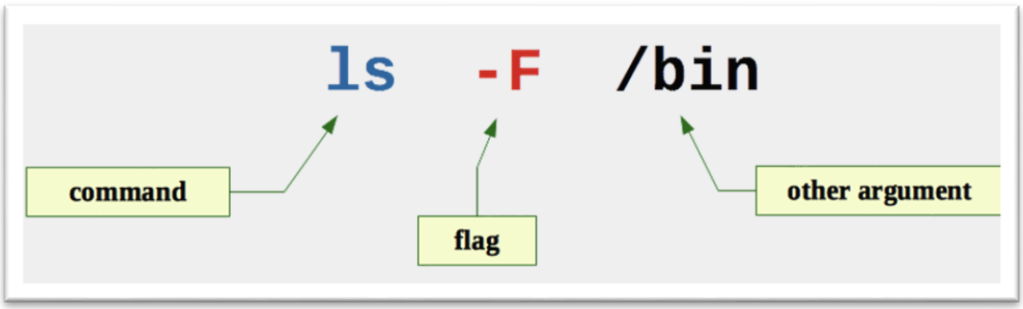
فصل دوم

آناتومی دستور

در تصویر زیرساختار یک دستور لینوکسی را مشاهده میکنید:



یا به صورت زیر:



یک دستور در سیستم عامل لینوکس از چند بخش تشکیل شده است که هر بخش نقش وظیفه‌ای خاصی دارد. ساختار یک دستور عموماً به صورت زیر است:

دستور اصلی (:(Commandاین بخش اصلی دستور است که عملیات خاصی را انجام می‌دهد. به عنوان مثال، ls برای لیست کردن فایل‌ها و پوشه‌ها، mkdir برای ایجاد یک پوشه جدید و غیره.

گزینه‌ها یا آرگومان‌ها (:(Options/Argumentsگاهی اوقات دستورات نیاز به گزینه‌ها یا آرگومان‌های اضافی دارند تا عملکرد آنها تغییر کند یا بهتر کند. این گزینه‌ها با استفاده از خط فرمان وارد می‌شوند. به عنوان مثال، ls -l برای نمایش جزئیات بیشتر در لیست فایل‌ها و پوشه‌ها.

پارامترها (:(Parameterبرخی از دستورات ممکن است نیاز به ورودی‌های اضافی داشته باشند که به عنوان پارامترها شناخته می‌شوند. این پارامترها می‌توانند اطلاعات مربوط به فایل‌ها، پوشه‌ها، یا سایر موارد مربوط به دستور باشند.

پایان خط فرمان (:(Command Terminationهر دستور در خط فرمان با فشار دادن دکمه Enter اجرا می‌شود و بازخورد مربوط به اجرای دستور به کاربر نمایش داده می‌شود.

یک دستور لینوکس دارای option های مختلفی میباشد که خروجی دستور را تغییر میدهد. مثلا در دستور ls وقتی option را –a قرار دهید، در خروجی فایل های مخفی نمایش داده میشود.

```
$ls –a
```

درمواردی به این option گاهی flag و گاهی switch نیز گفته میشود. هر دستور ممکنه است دارای پارامترهای ورودی نیز باشه که argument گفته میشود. مثلا در دستور زیر مسیر داده شده یک نوع آرگومانت می باشد.

```
$ls –a /usr/bin/
```

دستورات لینوکسی می‌تواند از انواع مختلف زیرباشند :

در لینوکس یا هر سیستم مبتنی بر یونیکس، یک دستور به دستورالعملی اشاره دارد که می تواند برای فراخوانی برنامه های کامپیوتری استفاده شود. سپس این برنامه به عنوان مفسر دستور و گزینه ها و آرگومان های آن عمل می کند و کار خاصی را انجام می دهد. در اکثر سناریوها، یک دستور به یک دستورالعمل در خط فرمان اشاره دارد که معمولاً یک پوسته مانند bash، csh، ksh و غیره است. برخی از دستوراتی که می توانند در پوسته خط فرمان استفاده شوند عبارتند از ls، cd، cat و غیره. همانطور که گفته شد، برخی از این دستورات می توانند آرگومان هایی را دریافت کنند که تعیین میکند برنامه بر روی کدام منبع عمل کند. بسیاری از دستورات، اگر نه همه، از گزینه‌های خط فرمان نیز پشتیبانی می‌کنند که عمل انجام‌شده روی منبع را تغییر می‌دهند. چند نمونه از دستورات هستند

```
$ ls -lisa /home/tom/photos/
```

در دستور بالا، /home/tom/photos/ آرگومان خط فرمان است، در حالی که –l، i-، s- و –a گزینه‌های خط فرمان هستند. ls فرمان است و عمل فهرست کردن یا چاپ فایل ها و پوشه ها را در پوشه مشخص شده انجام می‌دهد. همانطور که می دانید، صدها هزار دستور و برنامه مختلف وجود دارد که می‌توانید از خط فرمان استفاده کنید. همه این دستورات را می توان به طور کلی به چهار نوع مختلف طبقه بندی کرد. هر کدام از آنها دارای ویژگی ها و اولویت های خاصی در محیط لینوکس/یونیکس هستند.

دستورات داخلی شل(پوسته)

باتوجه به پوسته ای که استفاده می‌کنید ، چندین دستور مختلف در پوسته تعبیه شده است. برخی از این دستورات ممکن است بسته به پوسته کمی متفاوت باشند، اما بیشتر دستورات رایج در اکثر پوسته‌ها با گزینه‌های مشابه در دسترس هستند. اینها به آنها پوسته ساخته شده نیز گفته می‌شود . همانطور که از نام آن مشخص است، این دستورات دقیقاً در داخل پوسته اجرا می شوند. این به این معنی است که بسیار سریع است زیرا هیچ برنامه خارجی برای بارگیری وجود ندارد. توکارهای پوسته معمولاً کارهای ساده و بی اهمیتی هستند که به نوعی ذاتی پوسته هستند، مانند چاپ محتویات پوشه.

دستورات داخلی که در پوسته تعبیه شده اند:

برخی از نمونه‌های ساخته شده پوسته معمولی عبارتند از cd، bg، jobs، kill، local ، exit ، echo و غیره.

توابع درون پوسته

این به طور کلی به اسکریپت های پوسته یا همان برنامه نویسی اسکریپ یا shell scripting اشاره دارد که در محیط لینوکس یا پوسته گنجانده شده اند. توابع پوسته معادل متدها، توابع یا زیر روال ها در سایر زبان های برنامه نویسی هستند. اینها راه های خوبی برای استفاده مجدد از کد برای انجام کارهای تکراری در اسکریپت های پوسته هستند.برخی از کلیدواژه های رزرو شده پوسته مانند while، else، do، case و غیره که برای نوشتن اسکریپت های پوسته استفاده می شوند در این دسته قرار می گیرند.

برنامه های اجرایی

اینها احتمالاً متداول ترین نوع دستورات هستند. اینها همه برنامه هایی هستند که معمولاً روی دستگاه کامپایل[۱] و نصب می شوند، مانند برنامه هایی که در /usr/bin/ یا /usr/local/binپیدا می کنید. اینها همچنین می‌توانند به برنامه‌ها یا اسکریپت‌هایی اشاره کنند که

[۱] compile

زبان‌های تفسیری[۱] نوشته شدهاند، مانند پایتون، روبی یا پرل و غیره. برخی از نمونه های این نوع دستورات عبارتند از , wget , zip mplayer , vlcو vi .تا زمانی که این برنامه ها در متغیر $PATH هستند ، می توانید به راحتی با تایپ دستور در خط فرمان راه اندازی کنید . همچنین می توانید با استفاده از مسیر مطلق[۲] به دستور اجرایی، آنها را اجرا کنید.

روش تایپ دستورات در ترمینال

شما میتوانید یک دستور را در ترمینال نوشته و آن را اجرا کنید. همپنین به صورت های زیر، همانگونه که در تصویر میبینید، میتوانید چندین دستور را به حالت های زیر نیز اجرانمایید:

- Run commands sequentially
  cmd1 ; cmd2 ; cmd3 ...
- Run commands in parallel
  cmd1 & cmd2 & cmd3 ...
- Run commands in a subshell
  (cmd1 ; cmd2; cmd3 ...)
- Run commands conditionally
  cmd1 && cmd2 # run cmd2 only if cmd1 is successful
  cmd1 || cmd2 # run cmd2 only if cmd1 fails

همانگونه که ملاحضه میکنید:

در حالت اول: علامت سمی کالون[۳] باعث میشود که دستورات به ترتیب و پشت سر هم اجرا شوند

در حالت دوم :علام امپرسندیا & ، باعث میشود که دستور اولی و دومی و سومی به صورت موازی اجرا شوند.

در حالت سوم: دستورات در داخل پرانتز قرار دارند که در یک شل جداگانه اجرا خواهند شد.

در حالت چهارم:

۱-زمانی که بجای استفاده از یک & از دو && استفاده مینماییم، به این معنی خواهد بود که دستور دوم را اجرا کن به شرطی که دستور اول اجرا شد.

۲- زمانی که || استفاده مینماییم، به این معنی خواهد بود که دستور دوم را اجرا کن به شرطی که دستور اول اجرا نشود.

اگر بین دو دستور از یک | استفاده شود، یعنی یا دستور اولی یا دستور دومی اجرا شود.

مثال های دستور !

```
$ cd myfolder; ls   # no matter cd to myfolder successfully, run ls
```

```
$ cd myfolder && ls  # run ls only after cd to myfolder
```

```
$ cd myfolder || ls  # if failed cd to myfolder, ls will run
```

---

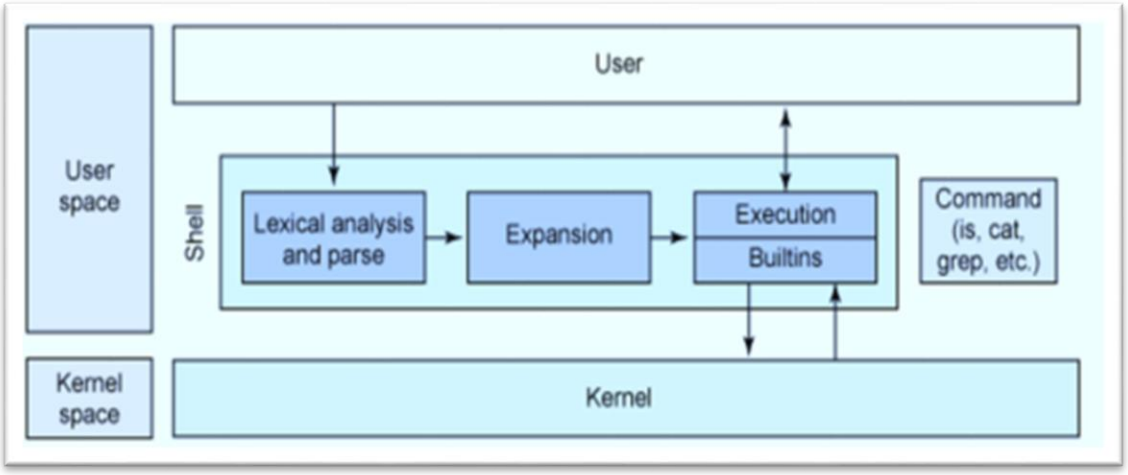
۱ Interpretive languages

۲ Absolute path

۳ ;

انواع پوسته

واژه پوسته ترجمه شده کلمه shell می باشد. از انواع رایج پوسته های لینوکسی موجود، شل بش– bash shellمی باشد. پوسته های تعاملی[۱] پوسته هایی هستند که دستورات را می پذیرند. Shell ها می‌توانند ورودی و تعاملی[۲]، غیر ورودی و غیر تعاملی، و یا هر ترکیبی دیگر باشند.شل ها دستورات شما را که از برنامه ترمینال وارد می نمایید، تفسیرمیکند و خروجی های متناظر با دستورات ورودی به شما ارائه میکند. پس شل‌ها یا همان پوسته ها رابط بین شما وکرنل هستند. و دستورات شما را به کرنل ارسال میکنند. یک پوسته دارای انواع متفاوتی هستند، همانطور که سیستم عامل ها دارای انواع متفاوتی هستند. هر نوعی از پوسته مجموعه ای از دستورات و عملکردهای شناخته شده خود را دارد.



محل قرارگیری شل یا پوسته در ساختار کرنل

چندین فایل درسیستم وجود دارد، که وابسته به شل شما هستند، علاوه بر فایل bashrc[۳] ، چندین اسکریپت[۴] دیگر نیز وجود دارد که به وسیله پوسته به طور خودکار هنگام ورود و بالا آمدن سیستم[۵] یا خروج از system تغیرات در آنها ذخیره می شوند:

- **/ etc / profile**
- **/ ~.bash_profile**
- **/ ~.bash_login**
- **/ ~.profile**
- **/ ~.bash_logout**
- **/etc/bash.bash_logout**

پوسته های رایج عبارتند از:

ویژگی های پوسته و دستور های پیش فرض آنها نیز شرح داده شده است.

---

۱ interactive

۲ CLI (command line interface

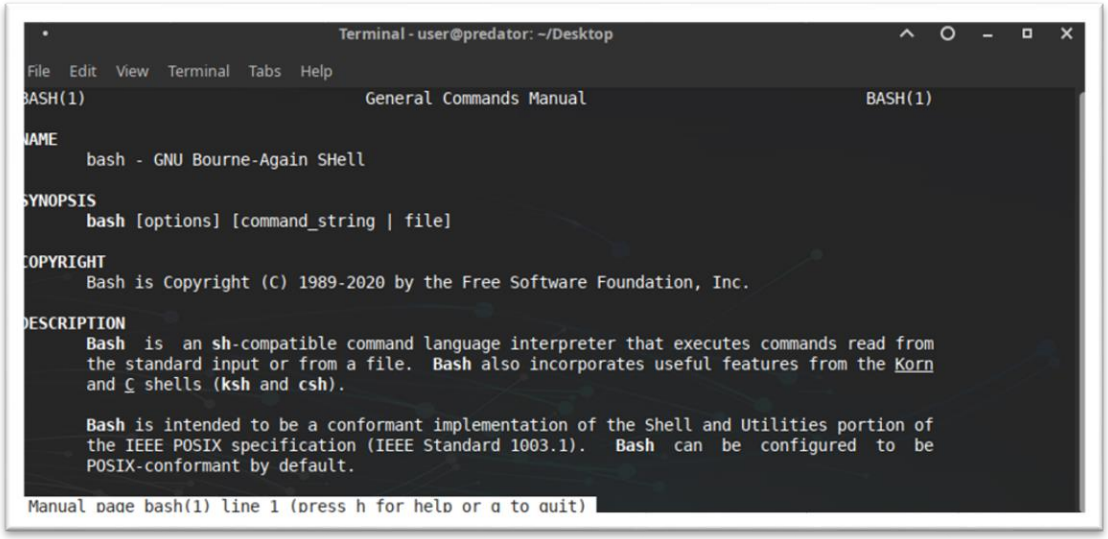
۳ /home/your-user-name/. **bashrc**

۴ script

۵ startup

پوسته بورن

پوسته بورن (sh)[1] ، که توسط استیو بورن در آزمایشگاههای AT&T Bell نوشته شده است ، پوسته اصلی UNIX است. به دلیل فشردگی و سرعت آن پوسته ترجیحی برای برنامه نویسی پوسته است. یک اشکال Bourne shell این است که فاقد ویژگی هایی برای استفاده تعاملی است ، مانند توانایی یادآوری دستورات قبلی (تاریخچه). پوسته بورن همچنین محاسبات داخلی و هندسی بیان منطقی ندارد.



همچنین پوسته Bourne پوسته پیش فرض خیلی از سیستم عامل ها است. پوسته بورن نام مسیر کامل مسیرزیر:

```
/ bin/sh
/ sbin / sh
```

Prompt کاربر غیر ریشه $ است.

Prompt کاربر ریشه # است.

پوسته C Shel

توسط بیل جوی در دانشگاه کالیفرنیا در برکلی نوشته شده است. ویژگی‌های گنجانیده شده در پوسته C Shell توسط بیل جوی شامل ویژگی‌های تعاملی و ویژگی‌های مناسب برنامه‌نویسی می‌شود:

نام‌های مستعار (:(Aliasesاین ویژگی به کاربران امکان می‌دهد که برای دستورات یا سریعتر دسترسی پیدا کنند، نام‌های دیگری را تعریف کنند. این نام‌های مستعار می‌توانند برای دستورات طولانی یا پیچیده باشند و کاربران را از تایپ مجدد دستورات طولانی نجات دهند.

تاریخچه دستورات (:(Command Historyاین ویژگی به کاربران امکان می‌دهد تا دستورات قبلی را مشاهده کرده و از آنها استفاده کنند. این ویژگی مفید است زیرا کاربران می‌توانند دستورات پیشین خود را بررسی کرده و از آنها برای تکرار عملیات استفاده کنند.

محاسبات داخلی (:(Internal Calculationsاین ویژگی به کاربران امکان محاسبات ساده و پیچیده را در داخل محیط خط فرمان فراهم می‌کند. این به کاربران امکان می‌دهد تا بدون نیاز به استفاده از برنامه‌های جداگانه برای محاسبات، محاسبات خود را در محیط خط فرمان انجام دهند.

---

[1] Bourne shell

نحو بیان C مانند (:C-like Syntax)این ویژگی به کاربران امکان می‌دهد تا دستورات و دستورات شرطی را با استفاده از نحو بیانی مشابه به زبان برنامه‌نویسی C تایپ کنند. این ویژگی به برنامه‌نویسان کمک می‌کند تا به راحتی دستورات پیچیده‌تر را در خط فرمان تایپ کنند و از قدرت برنامه‌نویسی خود استفاده کنند.

برای پوسته C:

نام مسیر کامل مسیر این پوسته:

```
/bin/csh
```

prompt برای کاربران غیر ریشه % است.

Prompt کاربر ریشه # است .

پوسته Korn

توسط دیوید کورن در سال ۱۹۸۰ در آزمایشگاههای AT&T Bell نوشته شده است .

از همه چیز در پوسته بورن پشتیبانی می کند.

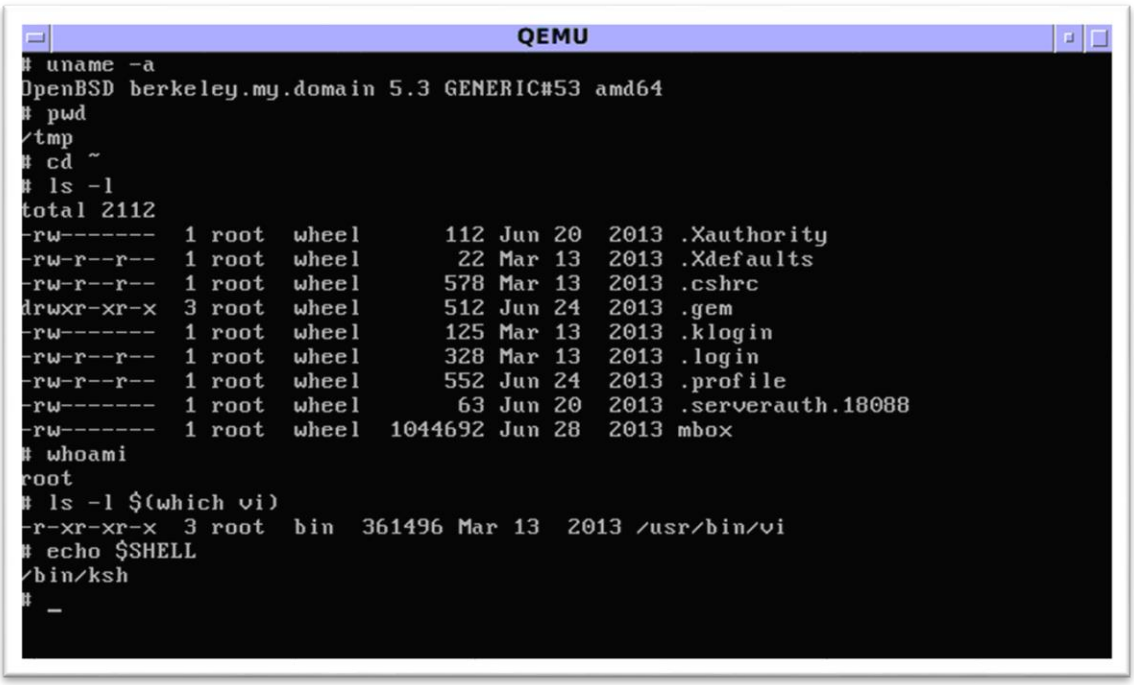
دارای ویژگی های تعاملی قابل مقایسه با ویژگی های پوسته C است.

شامل ویژگی های مناسب برنامه نویسی مانند محاسبات داخلی و آرایه های C مانند ، توابع و امکانات دستکاری رشته ای است .

سریعتر از پوسته C است.

اسکریپت های نوشته شده برای پوسته Bourne را اجرا می کند.

اعلان پیش‌فرض کاربر ریشه # است، در حالی که دستور کاربر پیش‌فرض غیر ریشه $ است.



برای پوسته Korn:

نام مسیر کامل :

```
/ bin / ksh/
```

prompt کاربر غیر ریشه $ است.

prompt کاربر ریشه # است.

پوسته GNU Bourne-Again

با پوسته بورن سازگار است.

دارای ویژگی های مفید از پوسته های Korn و C است.

دارای کلیدهای جهت دار است که برای یادآوری و ویرایش دستور به طور خودکار نقشه برداری می شود.

برای پوسته GNU Bourne-Again:

نام مسیر کامل :

```
/ bin / bash
```

prompt برای کاربر غیر ریشه# است.

prompt کاربرغیر ریشه $ است.

پوسته Z

Z Shell با zsh نشان داده می شود و پسوندی برای پوسته بورن با پیشرفت های فراوان است و به عنوان یک پوسته ورود به سیستم تعاملی استفاده می شود. این ویژگی‌ها را از پوسته های Bash، ksh و tcsh ترکیب می‌کند. همچنین به عنوان یک مفسر دستور برای اسکریپت نویسی پوسته استفاده می شود. برخی از ویژگی های مهم پوسته z عبارتند از:

- پشتیبانی از پلاگین و تم
- تولید Filename
- تصحیح املا و تکمیل تقریبی

پوسته ماهی-fish

پوسته ماهی یک پوسته کاربرپسند است که در سال ۲۰۰۵ ساخته شد. برخلاف رقبای خود، ماهی بر اساس پوسته Bourne یا پوسته C نیست. ماهی یک پوسته یونیکس با تمرکز بر تعامل و قابلیت استفاده است. فیش به گونهای طراحی شده است که به‌طور پیش‌فرض، به جای پیکربندی، ویژگی‌هایی را به کاربر بدهد. فیش دارای پیشنهادات خودکار "جستجو در حین تایپ"[۱]بر اساس تاریخچه و فهرست فعلی است. فیش قواعد نحوی[۲] کمی دارد و به جای نحو، ویژگی‌ها را به عنوان دستور ترجیح می‌دهد.

---

۱ automatic suggestions

۲ Syntax

جدول شل ها به صورت کلی:

Shell	Path	Default Prompt (non-root user)	Default Prompt (Root user)
The Bourne Shell (sh)	/bin/sh and /sbin/sh	$	#
The C Shell (csh)	/bin/csh	%	#
The Korn Shell (ksh)	/bin/ksh	$	#
The GNU Bourne-Again Shell (Bash)	/bin/bash	$	#
z shell	/bin/bash	$	#
Fish shell	/bin/bash	$	#

روند رشد و تکامل شل های لینوکس

۱۹۷۷: Bourne Shell (sh) به عنوان اولین شل در لینوکس معرفی شد.

۱۹۸۹: C Shell (csh) توسط بیل جویس توسعه یافت.

۱۹۸۹: Bourne Again Shell (Bash) توسط برایان فاکس برای بهبود Bourne Shell ایجاد شد.

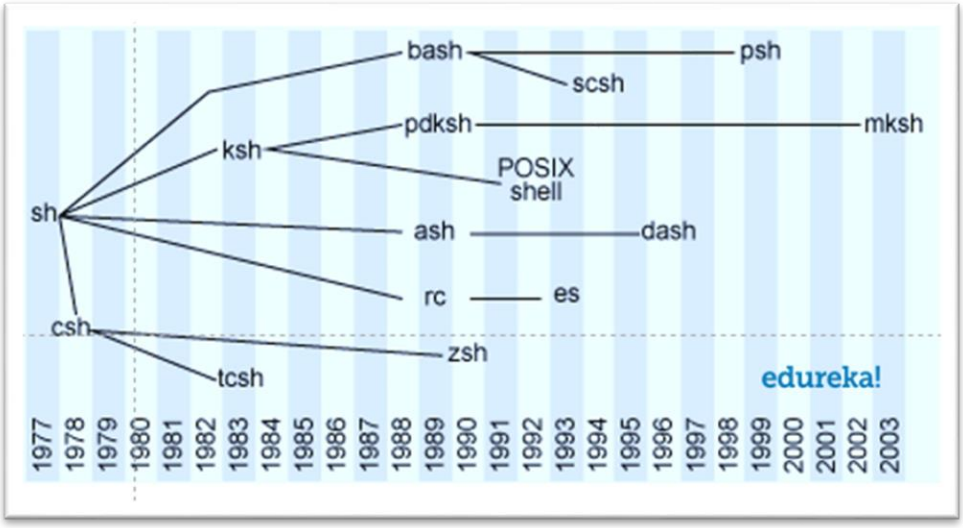
۱۹۹۰: Korn Shell (ksh) توسط دیوید کورنهایمر توسعه یافت.

۱۹۹۲: Z Shell (Zsh) به عنوان یک شل قدرتمند و پر ویژگی معرفی شد.

۲۰۰۵: Fish Shell برای سهولت استفاده توسعه یافت.

۲۰۱۹: Bash به عنوان شل پیش‌فرض در بسیاری از توزیع‌های لینوکس استفاده می‌شود.

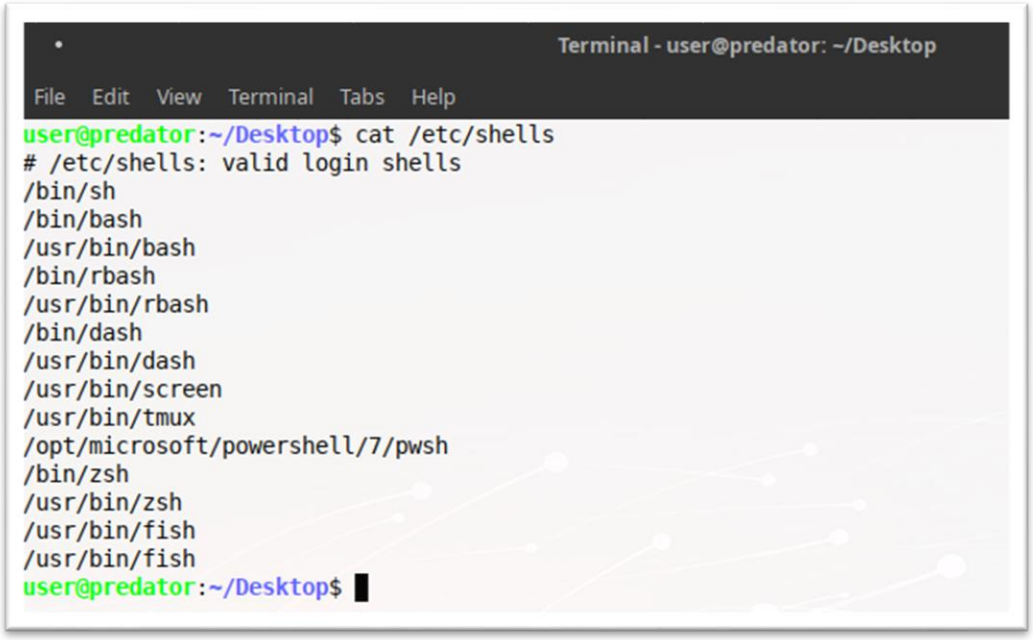
۲۰۲۱: شل‌های جدیدی مانند Tcsh و Elvish به بازار معرفی شدند.



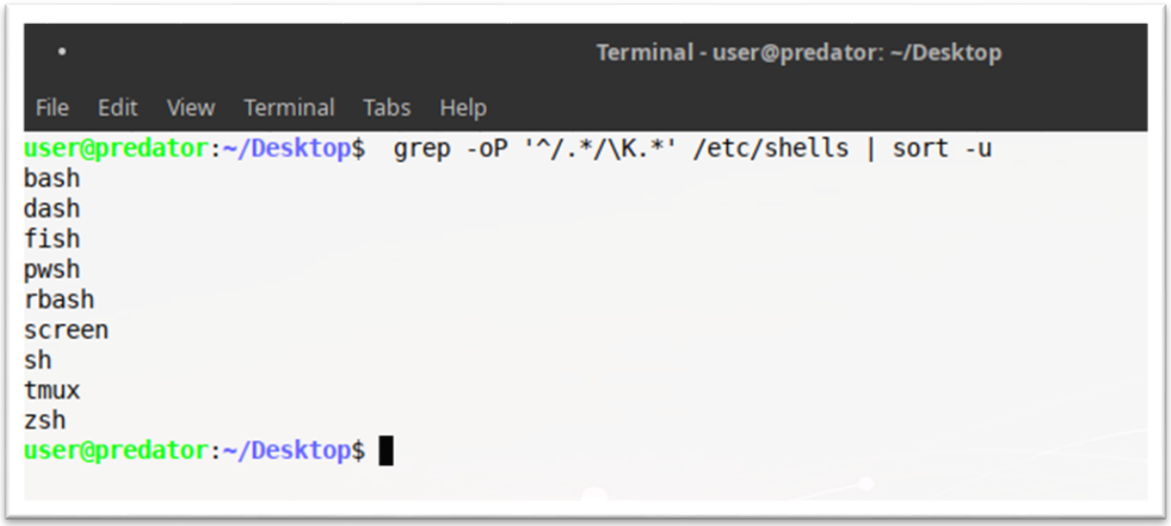
نمایش شل های نصب شده

به منظور نمایش شل های نصب شده دستورات زیر را میتوانید در ترمینال وارد نمایید.از آنجایی که همه پوسته‌های نصب شده روی سیستم شما در/etc/shells فهرست شده‌اند، همچنین می‌توانید ازدستور ps برای مشاهده شل در حال اجرا و اطلاعات مرتبط استفاده کنید.که این دستور در ادامه مثال زده میشود.

```
$ cat /etc/shells
```



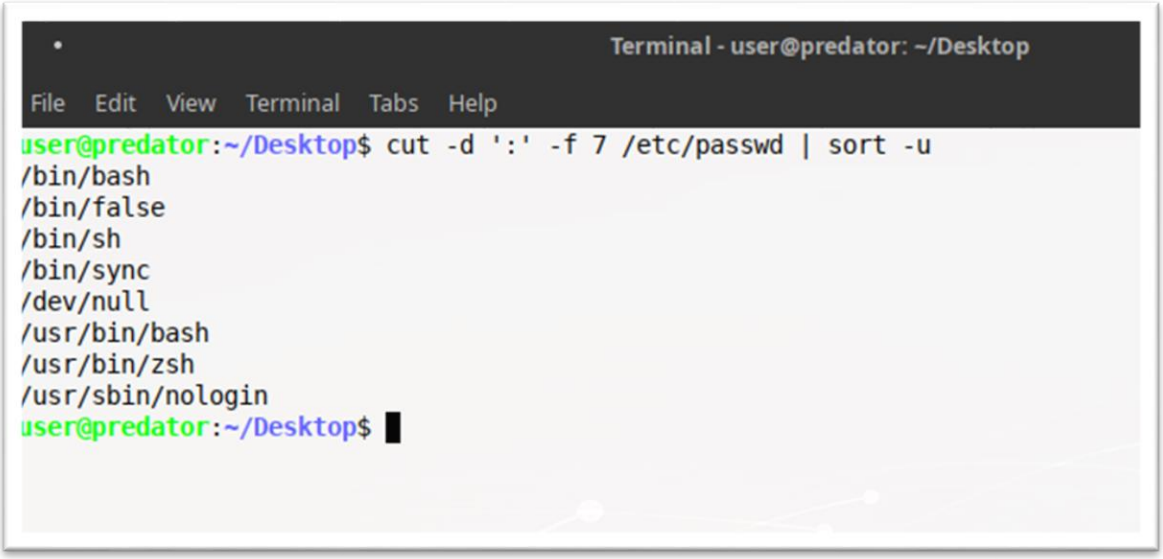
$ grep -oP ‘^/.*/\K.*’ /etc/shells | sort –u



grep ‘^[^#]’ /etc/shells



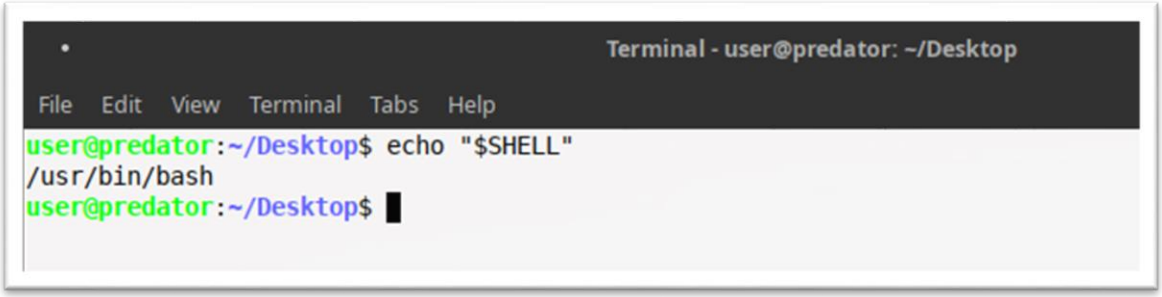

```
cut -d ':' -f 7 /etc/passwd | sort –u
```



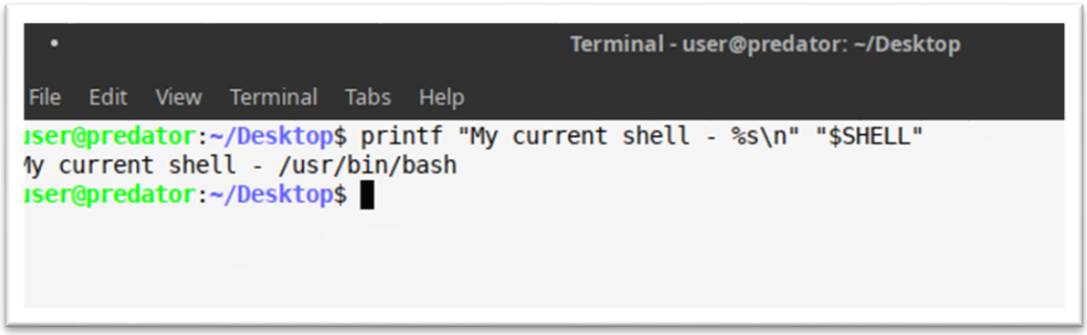
مشاهده شل فعال جاری

حال میخواهیم شل فعال را ازبین شل های نصب شده پیدا کنیم. برای این امر دستورات زیر را تایپ نمایید:

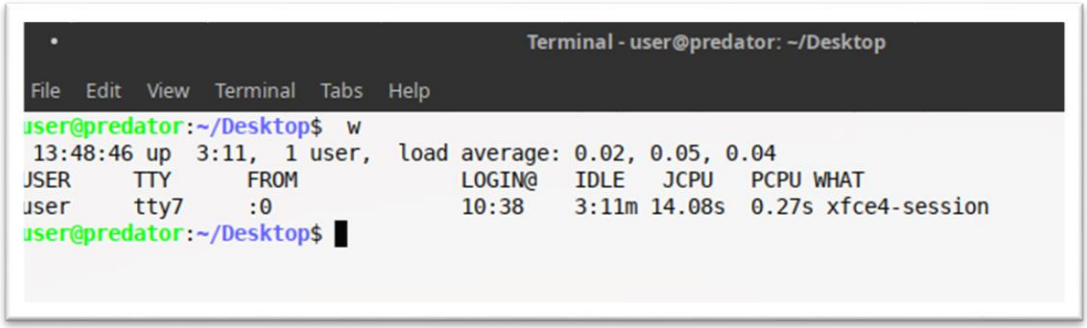
```
echo "$SHELL"
```



printf "My current shell - %s\n" "$SHELL"



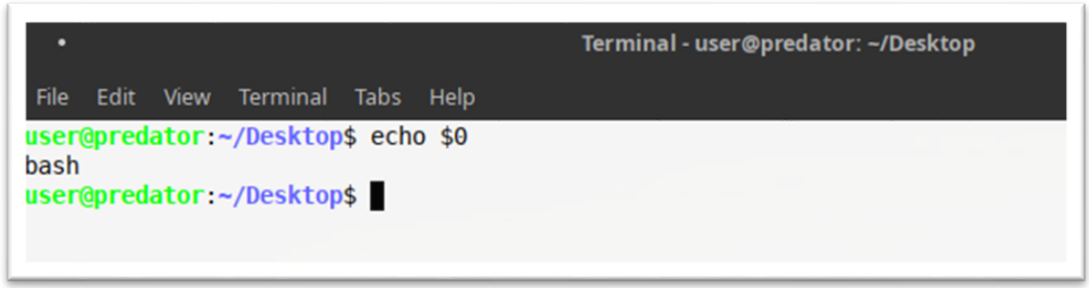
$ w



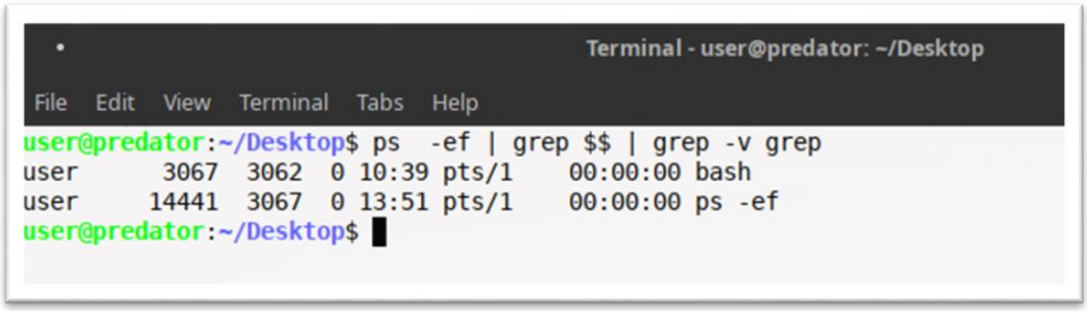
خروجی دستور:

- USER – Linux or Unix login name.
- TTY – The tty name.
- FROM The remote host or IP address.
- @Login – Login time.
- IDEL – Idle time.
- JCPU – The JCPU time is the time used by all processes attached to the tty
- PCPU – The PCPU time is the time used by the current process, named in the "what" field.
- WHAT – The command line of that users current process.

$echo $0



ps -ef | grep $$ | grep -v grep



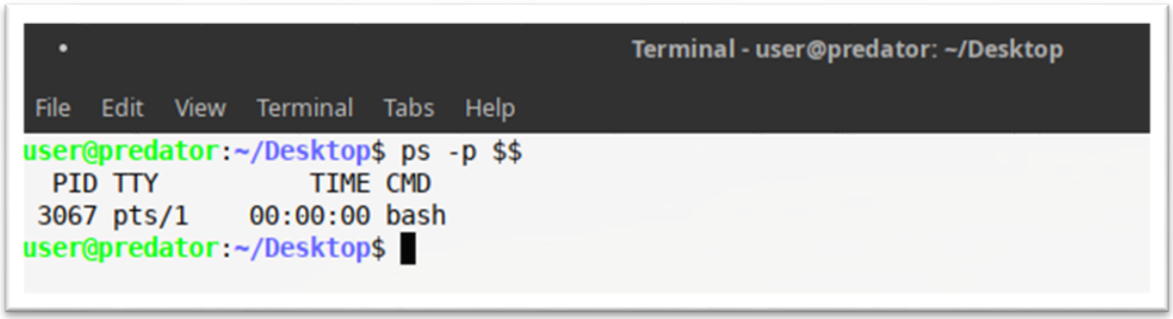
$ ps –p $$

یا

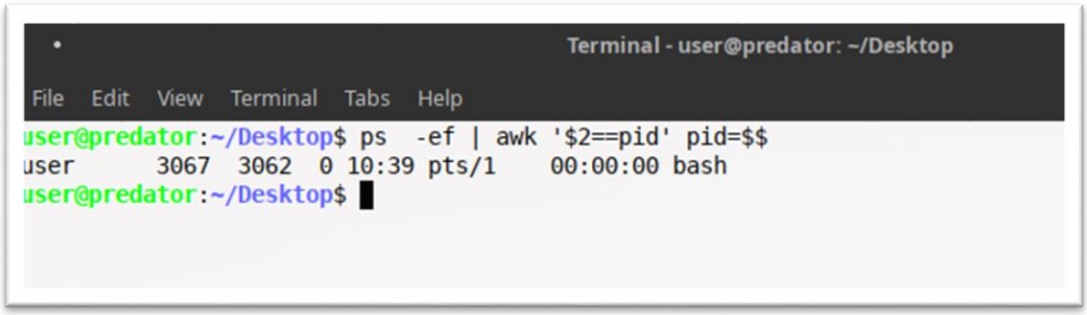
ps -p $$ -oargs=

یا

ps -p $$ -ocomm=



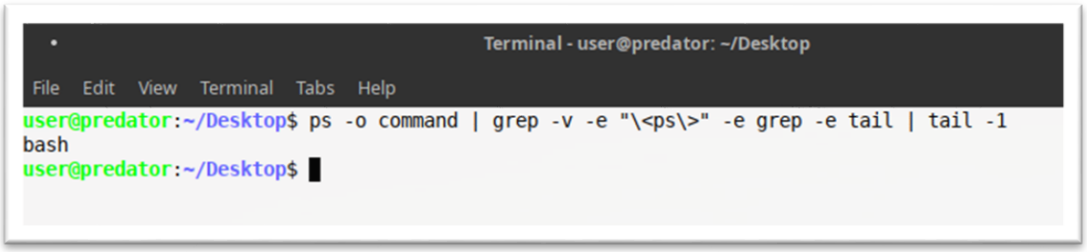
ps -ef | awk '$2==pid' pid=$$



ps -o fname --no-headers $$



ps -o command | grep -v -e "\<ps\>" -e grep -e tail | tail -1



دریافت مقدار مصرف cpu،ram در یک دستور

```
ps -p <pid> -o %cpu,%mem,cmd
```

تغییر پوسته پیش فرض

از دستور chsh در لینوکس برای تغییر شل (شل سیستم فعلی) استفاده می شود. همانگونه که گفته شد، شل یک رابط کاربری تعاملی با یک سیستم عامل است و می توان آن را یک لایه خارجی از سیستم عامل دانست. پوسته bash یکی از پرکاربردترین پوسته های ورود به سیستم در لینوکس است. این دستور به کاربر اجازه می دهد پوسته را از پوسته فعلی تغییر دهد. کاربر ریشه می تواند شل با دستور chsh که مخفف change shell است، تغییر دهد.

ساختار دستور:

```
chsh [-s shell] [-R] [-h] [username]
```

پس از پیداکردن شل های نصب شده، از دستور زیر شل خود را به شل مدنظر تغییر دهید:

```
$chsh –s /bin/fish
```

مثال های دستور:

```
chsh -s /bin/zsh ## change bash to ksh ##
```

گزینه های دستور:

- **s-**: برای تنظیم پوسته به عنوان پوسته ورود به سیستم شما استفاده می شود.

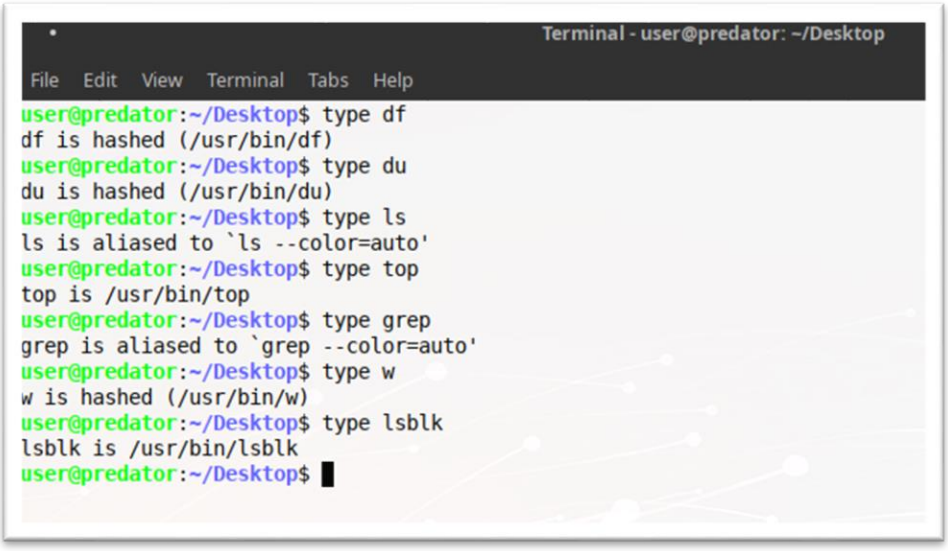
دستور type

دستور type می‌گوید که نام کامل یک دستور داده شده چیست؟

۱. **شل[۱] ساخته شده**
۲. **کلمه کلیدی شل**
۳. **نام مستعار[۲]**

---

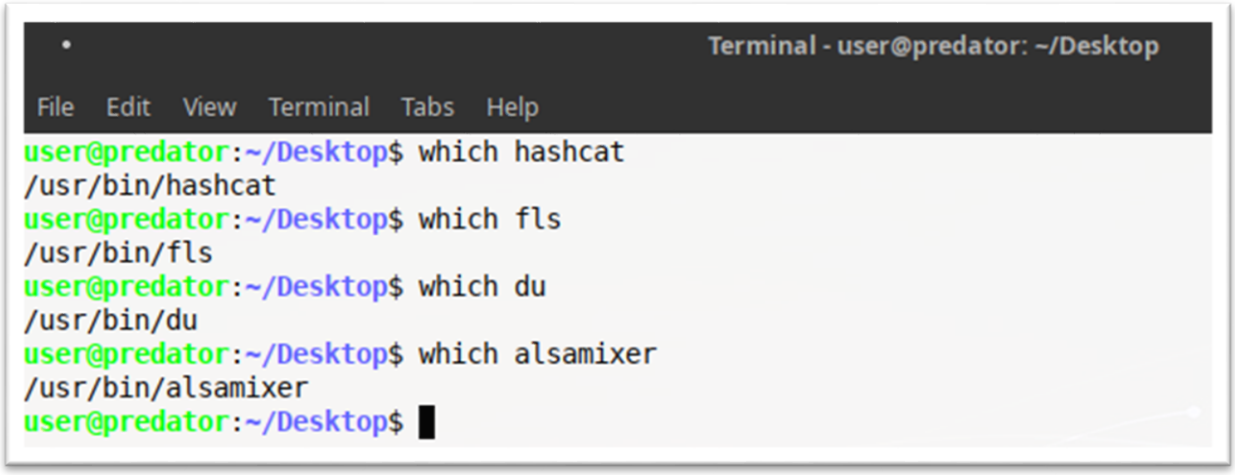
[۱] shell
[۲] alias



برای مشاهده تمام نام های مستعار ایجاد شده در system خود از دستور نام alias استفاده کنید :

دستور which

به یک دستور کمک می کند، مسیر دستور مطلق را به صورت زیر چاپ می کند:



دستور which در سیستم‌های عامل مبتنی بر یونیکس (UNIX) و لینوکس برای پیدا کردن مسیر (path) فایل اجرایی یا برنامه استفاده می‌شود. بطور کلی، which به شما اجازه می‌دهد تا ببینید که یک فایل اجرایی یا برنامه خاص در کدام مسیر سیستم قرار دارد. این دستور بیشتر در محیط خط دستوراستفاده می‌شود.

ساختار دستور:

```
$which [OPTIONS] COMMAND
```

در اینجا، COMMAND نشان‌دهنده نام فایل اجرایی یا برنامه است که می‌خواهید مسیر آن را پیدا کنید.

مثال های دستور:

پیدا کردن مسیر یک فایل اجرایی:

```
$which gcc
```

این دستور مسیر فایل اجرایی gcc را در سیستم پیدا می‌کند و مسیر آن را نمایش می‌دهد.

پیدا کردن مسیر یک برنامه:

```
$which python
```

این دستور مسیر برنامه python را در سیستم پیدا می‌کند و مسیر آن را نمایش می‌دهد.

پیدا کردن مسیر یک برنامه با نام یکسان:

```
$which -a java
```

در اینجا، با استفاده از گزینه a-، which تمام مسیرهای برنامه java را که با همین نام در سیستم وجود دارند، نمایش می‌دهد .

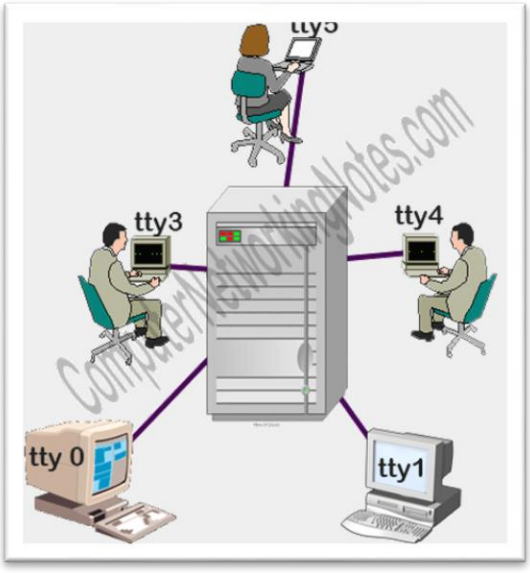
دستور which بسیار مفید است زمانی که بخواهید بفهمید که یک فایل اجرایی یا برنامه خاص در کدام مسیر سیستم قرار دارد. با استفاده از این دستور، می‌توانید مسیر صحیح را برای اجرای فایل‌ها و برنامه‌ها تعیین کنید.

فصل سوم

فصل سوم

ترمینال های مجازی

کنسول مجازی یک پوسته یک محیط غیر گرافیکی است که از طریق ماشین فیزیکی قابل دسترسی است، نه از راه دور .به طور همزمان می توان به چند کنسول مجازی[۱] دسترسی داشت.اگر هنگام نصب لینوکس یا اجرای آن با مشکل مواجه شدید، این کنسول های مجازی[۲] می توانند مفید باشند .پیام‌هایی که روی کنسول‌های نصب یا سیستم نمایش داده می‌شوند می‌توانند به مشخص کردن مشکل کمک کنند.



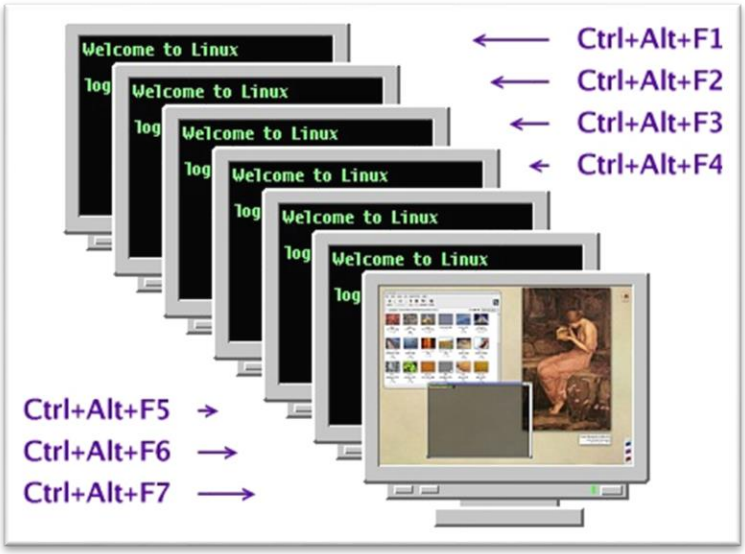
کنسول مجازی همچنین به عنوان ترمینال مجازی شناخته می شود.ترکیبی مفهومی از صفحه کلید و صفحه نمایش برای رابط کاربری رایانه است. این یکی از ویژگی‌های برخی از سیستم‌عامل‌های شبه یونیکس مانند لینوکس، BSD، illumos، UnixWare و macOS است که در آن می‌توان از کنسول سیستم رایانه برای جابه‌جایی بین چندین کنسول مجازی برای دسترسی به رابط‌های کاربری نامرتبط استفاده کرد. قدمت کنسول های مجازی حداقل به Xenix و Concurrent CP/M در دهه ۱۹۸۰ برمی گردد.

تاریخچه و مفهوم ترمینال مجازی

در روزهای قبلی که رایانه های شخصی بسیار گران بودند، شرکت ها ترجیح می دادند از رایانه های اصلی استفاده کنند. دستگاهی به نام ترمینال که عموماً یک مانیتور با صفحه کلید یکپارچه است برای اتصال به رایانه اصلی در پورت سریال[۳] استفاده می شود و از آنجایی که این ترمینال به صورت فیزیکی وجود دارد به عنوان ترمینال فیزیکی شناخته می شود. پس از اتصال، کاربر می تواند با استفاده از آن ترمینال شروع به استفاده از منابع سیستم کند. یک رایانه مرکزی می‌تواند چندین پایانه متصل به پورت‌های خود داشته باشد تا چندین کاربر همزمان بتوانند از آن استفاده کنند. اولین پایانه ها تله تایپ یا TTY نیز نامیده می شدند. مفهوم مشابهی هنوز در سیستم‌های مبتنی بر یونیکس وجود دارد که به چندین کاربر اجازه می‌دهد به سیستم وارد شوند و از آن به طور همزمان استفاده کنند. این پایانه ها با سیستم عامل ها به صورت کد پیاده سازی می شوند. بنابراین اینها به عنوان پایانه های مجازی شناخته می شوند.

---

۱ Tty (**teletype**)
۲ Virtual console
۳ Serial port



ترمینال مجازی یا کنسول مجازی یکی از ویژگی های سیستم مبتنی بر یونیکس است که به چندین کاربر اجازه می دهد به طور همزمان به منابع سیستم دسترسی داشته باشند. این یک ترمینال تمام صفحه[۱] است که در سیستم ویندوز X اجرا نمی شود. ماوس را نمی توان در ترمینال مجازی استفاده کرد.

راه اندازی ترمینال مجازی

رفتن به محیط ترمینال و کنسول

Ctrl + Alt + F1 - Will send user to `tty1` or 1st Terminal

Ctrl + Alt + F2 - Will send user to `tty2` or 2nd Terminal

Ctrl + Alt + F3 - Will send user to `tty3` or 3rd Terminal

Ctrl + Alt + F4 - Will send user to `tty4` or 4th Terminal

Ctrl + Alt + F5 - Will send user to `tty5` or 5th Terminal

Ctrl + Alt + F6 - Will send user to `tty6` or 6th Terminal

بازگشت به محیط گرافیکی

Ctrl + Alt + F7 - Will send user to the 1st GUI Terminal

Ctrl + Alt + F8 - Will send user to the 2nd GUI Terminal

Ctrl + Alt + F9 - Will send user to the 3rd GUI Terminal

Ctrl + Alt + F10 - Will send user to the 4th GUI Terminal

Ctrl + Alt + F11 - Will send user to the 5th GUI Terminal

Ctrl + Alt + F12 - Will send user to the 6th GUI Terminal

---

[۱] Full screen

چگونه بین خط فرمان (CLI) و رابط گرافیکی (GUI) در یک سرور لینوکس جابجا شویم

در این بخش، دو روش مختلف برای جابجایی بین خط فرمان و رابط گرافیکی در یک سرور لینوکس را بررسی خواهیم کرد. در زیر هر دو روش ممکن را فهرست کرده‌ایم.

مرحله ۱: جابجایی از GUI به CLI برای جابجایی از رابط گرافیکی (GUI) به خط فرمان (CLI) در لینوکس، کلیدهای Ctrl + Alt + F1 را فشار دهید. این میانبر صفحه کلید شما را به اولین ترمینال مجازی می‌برد و به شما امکان دسترسی به خط فرمان را می‌دهد. کلید: Ctrl + Alt + F1 جابجایی از GUI به CLI این میانبر صفحه کلید (Ctrl + Alt + F1) محیط دسکتاپ گرافیکی را ترک کرده و شما را به یک صفحه ورود متن در اولین ترمینال مجازی می‌برد.



مرحله ۲: جابجایی از CLI به GUI برای جابجایی از خط فرمان (CLI) به رابط گرافیکی (GUI) در لینوکس، کلیدهای Alt + F7 را فشار دهید. این میانبر صفحه کلید شما را به ترمینال مجازی اصلی برمی‌گرداند که رابط گرافیکی در آن اجرا می‌شود.
کلید: Alt + F7



پیمایش ترمینالهای مجازی

با استفاده از ctrl + alt + F1 به F6 می توانید از یک ترمینال به ترمینال دیگر سوئیچ کنید. فرض کنید می خواهید از tty1 به tty5 حرکت کنید، به سادگی ctrl + alt + F5 را فشار دهید. و برای بازگشت به tty1 دوباره ctrl + alt + F1 را فشار دهید، این شما را به ترمینال مجازی ۱ باز می‌گرداند. حالا اگر می‌خواهید بدون استفاده از کلیدهای تابع، از یک ترمینال مجازی به ترمینال مجازی دیگر سوئیچ کنید. با استفاده از دستور chvt می توانید این کار را انجام دهید. برای انتقال از tty1 به tty3 دستور زیر را در اعلان tty1 خود

اجرا کنید. در کنسول لینوکس و سایر پلتفرم‌ها، معمولاً شش کنسول مجازی اول یک پایانه متنی با یک اعلان ورود به پوسته یونیکس ارائه می‌کنند. سیستم گرافیکی X Window در کنسول مجازی هفتم شروع می شود.

دستور chvt

از دستور **chvt** در سیستم های لینوکس برای جابجایی بین پایانه های مختلف **TTY** (TeleTYpewriter) موجود استفاده می شود. دستور chvt در لینوکس برای تغییر و سوئیچ بین ترمینال‌های مجازی (Virtual Terminals)استفاده می‌شود. با استفاده از این دستور می‌توانید به صورت مستقیم به یک ترمینال مجازی خاص سوئیچ کنید. برای استفاده از دستور chvt، معمولاً نیاز به دسترسی روت (root)یا استفاده از sudo دارید.

ساختار دستور:

```
chvt N
```

این دستور یک مقدار صحیح برای N می گیرد ، از ۱ تا ۶ (از آنجا که فقط ۶ TTY در سیستم عامل Linux وجود دارد. وقتی یک عدد صحیح خارج از این محدوده مشخص شود ، یک صفحه خالی ظاهر می شود .برای خروج از این صفحه ، باید **Alt + FunKey (7)**را فشار داد.

به عنوان مثال، برای سوئیچ به ترمینال مجازی شماره ۲، می‌توانید از دستور زیر استفاده کنید:

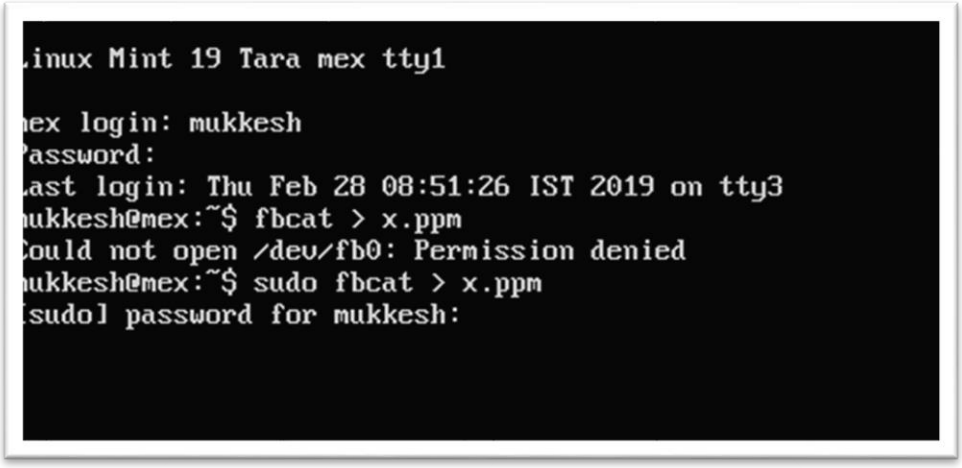
```
sudo chvt 2
```

این دستور باعث سوئیچ شدن به ترمینال مجازی شماره ۲ می‌شود. این امر می‌تواند برای اتصال به ترمینال‌های مجازی مختلف و انجام عملیات مورد نیاز در هرکدام از آن‌ها مفید باشد.

مثال ۱:

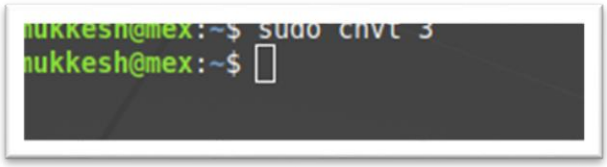
```
sudo chvt 1
```

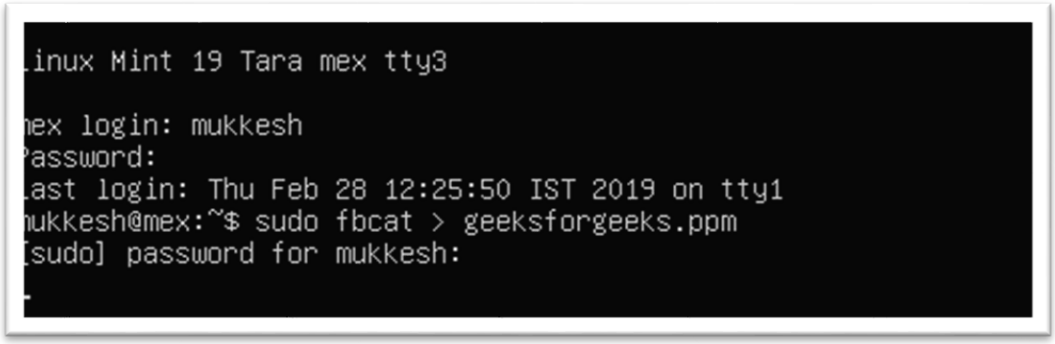




مثال ۲:

```
sudo chvt 3
```





دستور screen

می توانید یک جلسه screen را شروع کنید و سپس هر تعداد پنجره (پایانه های مجازی) را در داخل آن جلسه باز کنید. فرآیندهایی که در Screen اجرا می شوند، زمانی که پنجره آنها قابل مشاهده نباشد، حتی اگر اتصال شما قطع شود، همچنان اجرا می شوند. این برای اجرای طولانی در طول جلسه مانند اسکریپت های bash که بسیار طولانی اجرا می شوند بسیار مفید است . دستور screen یک ابزار مفید در سیستم‌عامل لینوکس است که به شما امکان می‌دهد ترمینال‌ها را در صورت نیاز به صورت مستقل اجرا کنید و با استفاده از آن‌ها می‌توانید به صورت همزمان با چندین پنجره ترمینال کار کنید. همچنین می‌توانید جلسات screen را ذخیره کنید و در صورت نیاز به آن‌ها بازگردید. دستور screen عموماً شامل مراحل زیر است:

```
$screen
```

با اجرای این دستور، یک پنجره ترمینال جدید باز می‌شود. برای خروج از پنجره screen و بازگشت به شل اصلی، از ترکیب کلیدهای Ctrl + A و سپس D استفاده کنید. برای بازگشت به پنجره screen مورد نظر، از دستور زیر استفاده کنید:

```
$screen -r
```

اگر فقط یک پنجره screen باز باشد، با اجرای این دستور به طور خودکار به آن بازگشته و در غیر این صورت، لیست پنجره‌های screen باز را نمایش می‌دهد و شما می‌توانید یکی را انتخاب کنید. برای بستن پنجره screen فعلی، درون پنجره screen به دستور exit یا Ctrl + D اجرا کنید.

به عنوان **مثال** برای پیدا کردن شناسه جلسه، می‌توانید جلسات فعلی صفحه نمایش در حال اجرا را با موارد زیر فهرست کنید:

```
$ screen -ls
```

صفحه نمایش در موارد زیر وجود دارد:

```
18787.pts-0.your-server
(Detached) 15454.pts-0.your-server
(Detached) 2 Sockets in /run/screens/S-yourserver.
```

اگر می‌خواهید صفحه ۱۸۷۸۷.pts-0 را بازیابی کنید، دستور زیر را تایپ کنید:

```
$ screen -r 18787
```

فصل چهارم

فصل چهارم

تنظیمات bash

در این قسمت آموزش برنامه نویسی بش داده نمیشود.اما مقدمه ای بر نحوه برنامه نویسی و ساختار آن ارائه می شود.

اولین خط اسکریپت BASH

دوکاراکتر عبارت اند از:

```
#!
```

پس از این، باید مسیر مترجم BASH را وارد کنید. بنابراین خط اول اسکریپت شما باید به شکل زیر باشد:

```
#!bin/bash
```

اگر پوسته پیش فرض شما BASH است، این خط:

```
#!bin/sh
```

علامت #! یا شناور شروع (shebang) در بش (Bash) یک نشانگر استفاده می‌شود تا محیط اجرایی مناسب برای اجرای اسکریپت مشخص شود. این خط در ابتدای یک اسکریپت قرار می‌گیرد و به سیستم عامل می‌گوید که از کدام برنامه باید استفاده کند تا اسکریپت را اجرا کند. به عنوان مثال، در صورتی که خط زیر را در ابتدای یک اسکریپت بش بنویسید:

```
#!/bin/bash
```

این به سیستم عامل می‌گوید که اسکریپت باید با استفاده از برنامه bash، یک محیط اجرایی متناسب با بش، اجرا شود. شناور شروع (shebang)برای تعیین محیط اجرایی مورد نیاز اسکریپت استفاده می‌شود و می‌تواند برای زبان‌ها و محیط‌های اجرایی مختلف مانند Python، Perl، Rubyو غیره استفاده شود. با استفاده از شناور شروع، می‌توانید برنامه‌ها و اسکریپت‌های خود را به صورت مستقل و قابل حمل بر روی سیستم‌های مختلف اجرا کنید، بدون اینکه نیازی به تعیین محیط اجرایی در هر بار اجرا داشته باشید.

دستور export

دستورات **export** یک دستورات builtins است، به این معنی که جزئی از شل و درون آن از قبل وجود دارد. ابتدا بدانید که ما متغییر هایی را در لینوکس داریم و باید برنامه از قبیل شل به این متغییرها دسترسی داشته باشند.از انجا که نمیتوانیم به تک تک برنامه‌ها، هربار این مقادیرمتغییرها را بدهیم.باید متغییرهای دلخواه با مقادیرشان را به شل پدر یا همان والد[۱] export بزنیم. به عنوان مثال میدانید که ترمینال ها دستورات شمارا history میکنند. و هربار که نیاز داشته باشید، دستور قبلی را میتوانید بیاورید و اجرا کنید. حال میخواهید مقدار history را برای همه شل ها تغییردهید که مثلا فقط ۵ دستور اخر را نگه دارند. با دستور زیر مقدار history دستورات در ترمینال ۵ عدد میشود.

```
export HISTSIZE=5
```

پس به صورت کلی دستور export یک متغیر محیطی را به سایر شل های فزرند ارسال میکند که در نتیجه هر شل فرزند، همه متغیرهای والد خود را به ارث ببرد. و شما هر شلی را که باز کنید مقدار HISTSIZE آن ۵ خواهد بود.

[۱] parent

متغییر های محیطی

تنظیمات محلی و گلوبال

بسیاری از تنظیمات در حالت سیستمی[۱] یا حالت global قرار دارند که برای تمام کاربران سیستم اعمال میشود.و همچنین تنظیمات محیطی هم هستند که برای کاربر محلی[۲] یا Local اعمال می شوند. کاربری که به سیستم وارده شده است.[۳] یکی از این فایلها برای نگهداری تنظیمات فایل زیر می باشد:

.bashrc

این فایل یک فایل خاص کاربر است که هر بار که کاربر یک جلسه محلی جدید(یعنی به سیستم loginمیکند) ایجاد می کند، یعنی به عبارت ساده زمانی ترمینال جدید را بازشود، این فایل بارگذاری می شود. همه متغیرهای محیطی ایجاد شده در این فایل هر بار که یک جلسه محلی جدید شروع می شود، اعمال می شوند.

.bash_profile

این فایل یک فایل ورود از راه دور مخصوص کاربر است. متغیرهای محیطی لیست شده در این فایل هر بار که کاربر از راه دور وارد سیستم می شود، یعنی با استفاده از جلسه ssh، فراخوانی می شود. اگر این فایل موجود نباشد، سیستم به دنبال فایل‌های .bash_login یا .profile می‌گردد.

/etc/environment

این فایل یک فایل گلوبال سیستم برای ایجاد، ویرایش یا حذف هر متغیر محیطی است. متغیرهای محیطی ایجاد شده در این فایل در سراسر سیستم، توسط هر کاربر، چه به صورت محلی و چه از راه دور، قابل دسترسی هستند.

/etc/bash.bashrc

فایل bashrc فایل گلوبال سیستمی است. این فایل یک بار برای هر کاربر بارگیری می شود، هر بار که کاربر یک جلسه ترمینال محلی را باز می کند. متغیرهای محیطی ایجاد شده در این فایل برای همه کاربران قابل دسترسی هستند اما فقط از طریق جلسه ترمینال محلی. هنگامی که هر کاربر در آن دستگاه از راه دور از طریق یک جلسه ورود به سیستم از راه دور قابل دسترسی باشد، این متغیرها قابل مشاهده نخواهند بود.

/etc/profile

فایل پروفایل یک فایل گلوبال سیستمی است. تمام متغیرهای ایجاد شده در این فایل برای هر کاربر در سیستم قابل دسترسی است، اما تنها در صورتی که جلسه آن کاربر از راه دور فراخوانی شود، یعنی از طریق ورود از راه دور. هیچ متغیری در این فایل برای جلسه ورود به سیستم محلی قابل دسترسی نخواهد بود، یعنی زمانی که کاربر ترمینال جدیدی را در سیستم محلی خود باز می کند.

توجه: متغیرهای محیطی ایجاد شده با استفاده از فایل‌های پیکربندی در سطح سیستم یا کاربر را می‌توان تنها با حذف آنها از این فایل‌ها حذف کرد. فقط این که پس از هر تغییر در این فایل ها، یا از سیستم خارج شوید[۴] و دوباره وارد شوید[۵] یا فقط دستور زیر را در ترمینال تایپ کنید تا تغییرات اعمال شوند:

```
$ source <file-name>
```

---

۱ system wide
۲ user specific
۳ login
۴ Logout
۵ login

ایجاد و حذف یک متغییر محیطی لوکال

یک متغییر محیلی لوکال با دستور زیر ایجاد میشود. نام متغییر و مقدار آن را بنویسیدوسپس آن را بادستورexport ارسال کنید.

```
$ var=value
OR
$ export var=value
```

لغو یک متغییر محیلی تنظیم شده

با دستور unset میتوانید آن متغییر را حذف کنید.

```
$ unset <var-name>
```

تعریف یک متغییر سیستمی یا گلوبال

الف) فایل bashrc. را در فهرست اصلی خود تغییر دهید تا متغیر محیطی را که باید اضافه یا صادرکنید. پس از آن فایل را سورس کنید تا تغییرات اعمال شوند.

```
$ sudo vi  .bashrc
```

سپس متغییر مورد نیاز خود را به خط اخر فایل اضافه کنید

```
sudo export CD=‘This is Predator-os  Home’
```

حالا دستور زیر را تایپ نمایید تا متغییر اعمال و ثبت شود

```
$ source  ~/.bashrc
$ echo  $CD
```

انواع متغیرهای محلی

یک متغیر محیطی می تواند در سه نوع باشد:

۱.متغیر محیطی محلی

یکی برای جلسه فعلی تعریف شده است. این متغیرهای محیطی فقط تا جلسه فعلی دوام دارند، خواه جلسه ورود از راه دور یا جلسه ترمینال محلی باشد. این متغیرها در هیچ فایل پیکربندی مشخص نشده اند و با استفاده از مجموعه ای از دستورات خاص ایجاد و حذف می شوند.

۲. متغیر محیط کاربر

اینها متغیرهایی هستند که برای یک کاربر خاص تعریف می شوند و هر بار که کاربر با استفاده از یک جلسه ترمینال محلی وارد می شود یا آن کاربر با استفاده از جلسه ورود از راه دور وارد سیستم می شود، بارگذاری می شوند. این متغیرها معمولاً در فایل‌های پیکربندی زیرتنظیم و بارگیری می‌شوند: فایل‌های bashrc، bash_profile، bash_login، profile. که در فهرست اصلی کاربر وجود دارند.

۳. متغیرهای محیطی گسترده سیستم

اینها متغیرهای محیطی هستند که در سراسر سیستم در دسترس هستند، یعنی برای همه کاربران حاضر در آن سیستم. این متغیرها در فایل‌های پیکربندی کل سیستم موجود در فهرست‌ها و فایل‌های زیر وجود دارند: /etc/environment، /etc/profile، /etc/profile.d/ ،/etc/bash.bashrc. این متغیرها هر بار که سیستم روشن می شود و توسط هر کاربری به صورت محلی یا از راه دور وارد سیستم می شود بارگیری می شوند.

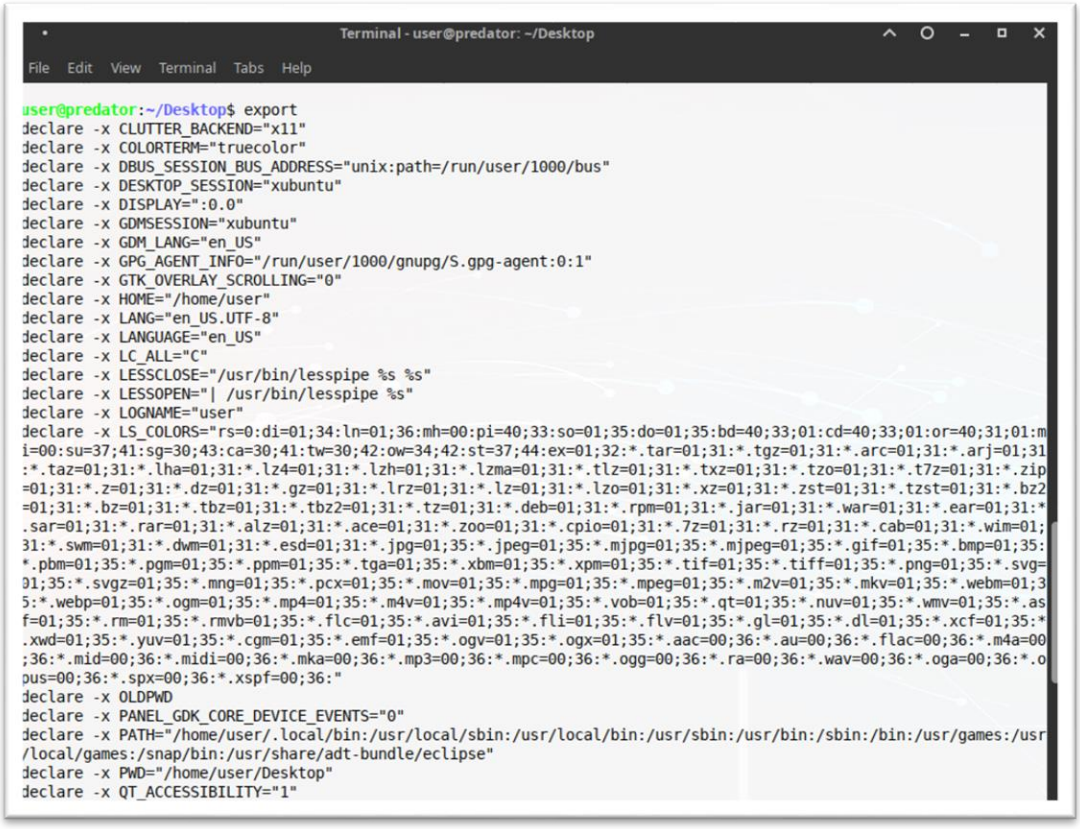
ساختار دستور:

```
export [-f] [-n] [name[=value] ...] or export -p
```

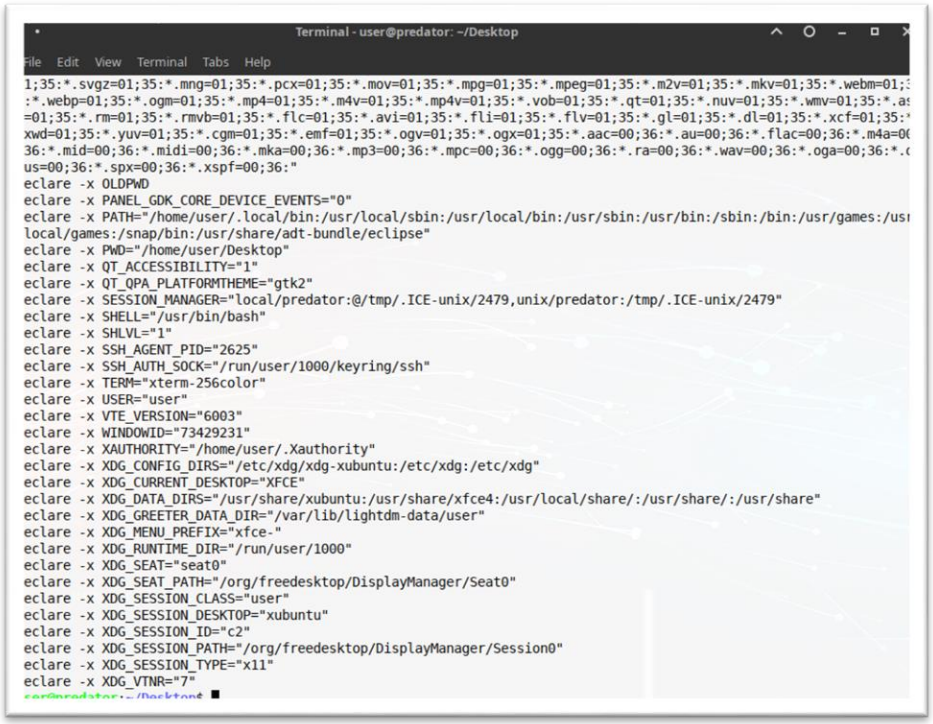
گزینه های موجود برای استفاده در دستور export به شرح زیر می باشند :

- بدون هیچ گزینه ای :برای مشاهده تمام متغیرهای صادر شده.
- سویچ p همه اسامی استخراج شده به شل اخیر را نمایش می دهد.
- سویچ n اسامی مورد نظر را از لیست برای ما حذف می کند.
- سویچ f اسامی به عنوان توابع استخراج می شوند.

مثال های دستور :



مثال :



سویچ f- اگر نامها به توابع اشاره داشته باشند باید از آن استفاده شود .اگر f- استفاده نشود ، export فرض می کند نام ها متغیر باشند.

دستور set

دستور set در سیستم‌های Unix/Linux برای تنظیم متغیرهای محلی، مشاهده تمام متغیرها و تنظیمی که در محیط فعلی وجود دارد، استفاده می‌شود. این دستور معمولاً برای نمایش تمامی متغیرها، تنظیم‌ها و تنظیم‌های محیطی مورد استفاده قرار می‌گیرد.

مثال های دستور:

۱. نمایش تمامی متغیرها و تنظیم‌ها:

```
set
```

این دستور تمامی متغیرها، تنظیم‌ها و تنظیم‌های محیطی را نمایش می‌دهد.

۲. تنظیم متغیر محلی و نمایش آن:

```
myvar=“Hello”
set | grep myvar
```

در این مثال، متغیر myvar با مقدار Hello تنظیم شده و سپس با استفاده از grep نمایش داده می‌شود.

۱. ندارد: دستور set از گزینه‌های خاصی برای استفاده ندارد و تنها برای نمایش تمامی متغیرها، تنظیم‌ها و تنظیم‌های محیطی استفاده می‌شود.

دستور env

دستور env در سیستم‌های Unix/Linux برای نمایش متغیرهای محیطی ( )environment variablesموجود و اجرای یک دستور یا برنامه در یک محیط متغیری خاص استفاده می‌شود. این دستور معمولاً برای اجرای برنامه‌ها و تنظیم متغیرهای محیطی مرتبط با آن‌ها استفاده می‌شود.

مثال های دستور:

۱. نمایش تمامی متغیرهای محیطی موجود

دستور unset

دستور unset در سیستم‌های Unix/Linux برای حذف یک یا چند متغیر محلی یا متغیر محیطی استفاده می‌شود. با استفاده از این دستور، می‌توانید یک متغیر را از محیط محلی خود یا از متغیرهای محیطی سیستم حذف کنید.

مثال های دستور:

۱. حذف یک متغیر محلی:

```
myvar=“Hello”
unset myvar
```

در این مثال، متغیر محلی myvar که مقدار “Hello” را داشت، حذف می‌شود.

۲. حذف یک متغیر محیطی:

```
unset PATH
```

با اجرای این دستور، متغیر محیطی PATH که معمولاً شامل مسیرهای جستجو برای اجرای برنامه‌ها است، حذف می‌شود.

گزینه‌های دستور:

۱. ندارد: دستور unset از گزینه‌های خاصی برای استفاده ندارد و تنها برای حذف متغیرهای محلی و متغیرهای محیطی استفاده می‌شود.

دستور env

دستور env در لینوکس/یونیکس برای چاپ لیستی از متغیرهای محیط فعلی یا اجرای یک برنامه در یک محیط سفارشی بدون تغییر محیط فعلی استفاده می شود. دستور env در لینوکس برای نمایش یا تنظیم متغیرهای محیطی (environment variables) استفاده می‌شود. این دستور به کاربر اجازه می‌دهد تا متغیرهای محیطی را مشاهده کند، تغییر دهد یا مقدار جدیدی برای آن‌ها تعیین کند.

گزینه های دستور:

- i– یا ignore-environment: این گزینه باعث می‌شود که متغیرهای محیطی فعلی نادیده گرفته شوند و یک محیط پاک و جدید برای اجرای دستور مشخص شود.
- u VAR=value– یا unset=VAR: با استفاده از این گزینه می‌توانید یک متغیر محیطی مشخص را حذف کنید یا مقدار جدیدی به آن تعیین کنید.
- C– یا chdir=DIR: این گزینه مشخص می‌کند که دستور env باید در دایرکتوری مشخص‌شده اجرا شود و متغیرهای محیطی مربوط به آن دایرکتوری را نمایش دهد.
- S– یا split-string: این گزینه باعث می‌شود که مقادیر متغیرهای محیطی که با کاراکتر فاصله جدا شده‌اند به عنوان جداکننده متغیرها در نظر گرفته شوند.

ساختاردستور:

```
$ env [OPTION]... [-] [NAME=VALUE]... [COMMAND [ARG]...]
```

مثال های دستور:

مجموعه متغیرهای محیط فعلی را چاپ میکند.

```
$ env
```

دستوری را با محیط خالی اجرا میکند

```
$ env -i command_name
```

حذف یک متغیر از محیط محلی

```
$ env -u variable_name
```

هر خروجی را با NULL پایان میدهد.

```
$  env -0
```

شروع با یک محیط خالی گزینه i:

```
$env -i
```

پایان دادن به خطوط خروجی با NUL به جای خطوط جدید :

```
$env --null
```

حذف یک متغیر محیطی

```
$env --unset=VAR_NAME
```

تغییر فهرست کاری به یک فهرست خاص

```
$env --chdir=/path/to/directory
```

فعال کردن حالت دیباگ

```
$env --debug
```

نمایش پیام راهنما

```
$env --help
```

نمایش اطلاعات نسخه

```
$env --version
```

دستور printenv

printenv مقادیر متغییرهای محیطی مشخص شده را چاپ می‌کند. متغیر(ها) . اگر VARIABLE مشخص نشده است، نام و جفت مقدار همه را چاپ میکند.

ساختاردستور:

```
$  printenv [OPTION]... PATTERN...
```

گزینه های دستور:

-0 --null به جای اینکه هر خط خروجی را چاپ کند ان را با 0 بایت پایان میدهد

help -- یک پیام راهنما را نمایش میدهد و از آن خارج میشود.

مثال های دستور:

۱. نمایش مقادیر تمام متغیرهای محیطی

```
$ printenv
```

۲. فهرست اصلی کاربر فعلی .

```
$ printenv $HOME
```

۳. برای استفاده از گزینه خط دستور null -- به عنوان کاراکتر پایانی بین ورودی های خروجی.

```
$ printenv -- null $SHELL $HOME
```

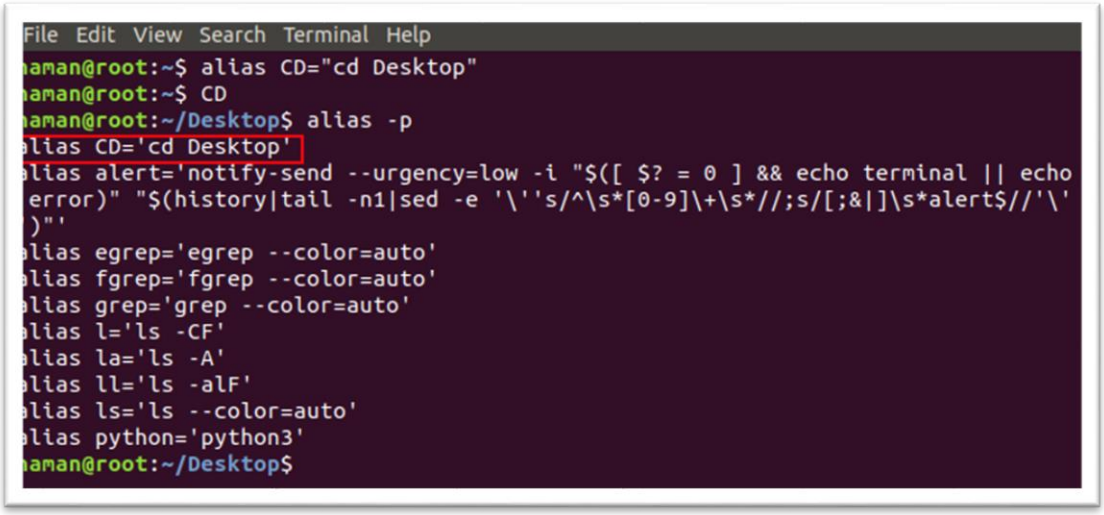
توجه: به طور پیش فرض، printenv دستور از newline به عنوان کاراکتر پایانی بین ورودی های خروجی استفاده می‌کند.

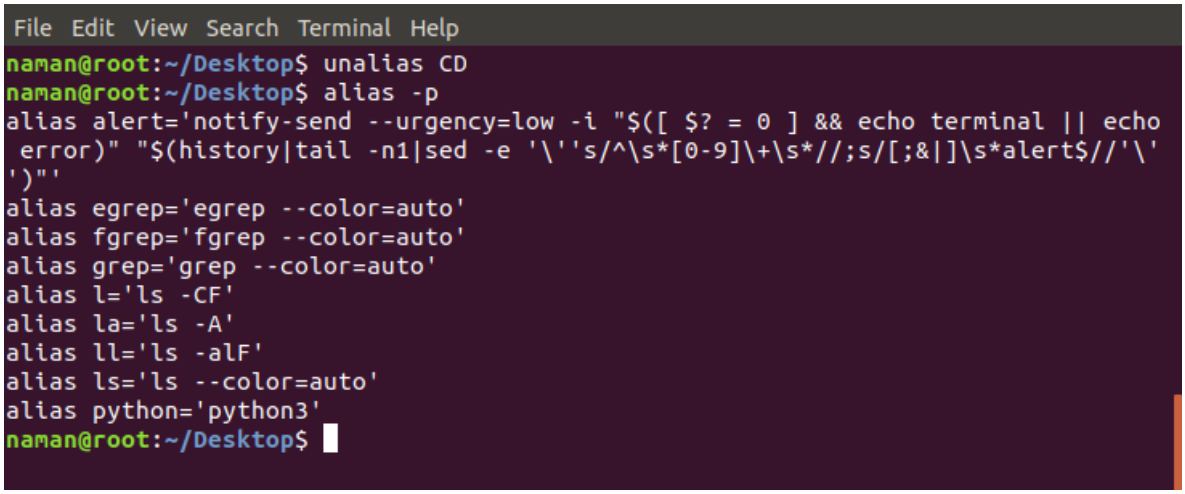
دستور alias

این دستور به معنی مستعار، شبیه به متغیرهای محیطی[۱] است، اما معمولا به صورت متفاوتی استفاده می شود برای جایگزینی دستورات طولانی با دستور کوتاه تر است. دستور alias در لینوکس برای ایجاد و مدیریت خطوط دستوری مختصر و سفارشی استفاده می‌شود. با استفاده از alias می‌توانید یک نام مختصر برای یک دستور طولانی‌تر تعریف کنید تا هنگام استفاده از آن نیازی به تایپ کامل دستور نباشد.

ایجاد نام مستعار

دستور alias و سپس دستور مورد نظر خود را که تمایل به کوتاه سازی آن را دارید وارد نمایید.





گزینه های دستور:

-p option :این گزینه چاپ تمام نام های مستعار تعریف شده با قالب قابل ذخیره است.

```
alias -p
```

[۱] environment variable

```
File  Edit  View  Search  Terminal  Help
naman@root:~$ alias -p
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo
 error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\'
')"'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
alias python='python3'
naman@root:~$
```

گزینهhelp– : اطلاعات راهنما را نمایش می دهد.

ساختار دستور:

alias -- help

```
File  Edit  View  Search  Terminal  Help
naman@root:~/Desktop$ alias --help
alias: alias [-p] [name[=value] ... ]
    Define or display aliases.

    Without arguments, `alias' prints the list of aliases in the reusable
    form `alias NAME=VALUE' on standard output.

    Otherwise, an alias is defined for each NAME whose VALUE is given.
    A trailing space in VALUE causes the next word to be checked for
    alias substitution when the alias is expanded.

    Options:
      -p        print all defined aliases in a reusable format

    Exit Status:
    alias returns true unless a NAME is supplied for which no alias has been
    defined.
naman@root:~/Desktop$
```

حذف یک نام مستعار

برای این منظور از دستور unailas استفاده می کنیم.

```
File  Edit  View  Search  Terminal  Help
eek@geek-laptop:~$ alias CD="cd Desktop"
eek@geek-laptop:~$ CD
eek@geek-laptop:~/Desktop$ unalias CD
eek@geek-laptop:~/Desktop$ CD
D: command not found
eek@geek-laptop:~/Desktop$
```

دستور unalias

دستور unalias در سیستم‌عامل‌های مبتنی بر Unix/Linux برای حذف یک نام مستعار (alias) تعریف شده توسط کاربر به کار می‌رود. نام مستعارها معمولاً برای تعریف دستورات کوتاهتر و سفارشی‌تر استفاده می‌شوند. با اجرای unalias و ذکر نام مستعار مورد نظر، آن نام مستعار از سیستم حذف خواهد شد. دستور unalias برخی گزینه‌های مفید را نیز پشتیبانی می‌کند.

گزینه های دستور:

-a یا --all : با استفاده از این گزینه، تمام نام‌های مستعار تعریف شده را حذف می‌کند. به عبارت دیگر، تمام نام‌های مستعار تعریف شده توسط کاربر به حالت پیش‌فرض باز می‌گردند.

-v یا --verbose : با استفاده از این گزینه، پیامی که نمایش می‌دهد که نام مستعار حذف شده است.

مثال های دستور:

حذف یک نام مستعار خاص:

```
$unalias ll
```

حذف تمام نام‌های مستعار تعریف شده:

```
$unalias -a
```

حذف یک نام مستعار با نمایش پیام:

```
$unalias -v gs
```

با استفاده از این گزینه‌ها، شما می‌توانید نام‌های مستعار تعریف شده را با دقت حذف کنید و به حالت پیش‌فرض برگردانید.

ایجاد نام مستعار دائمی

ما می توانیم نام مستعار را در پرونده های پیکربندی پوسته ذخیره کنیم .به عنوان مثال برای Bash Shell، می توانیم در / ~ bashrc.بنویسیم. فایل bashrc را باز کنید و دستور خود را وارد نمایید.

```
alias CD=“cd Desktop”
```

دستور history

اجرای دستور ساده تاریخچه از ترمینال لیست کاملی از آخرین دستورات اجرا شده با شماره خطوط را به شما نشا می دهد.

```
# history
 1  sudo add-apt-repository ppa:tualatrix/ppa
 2  sudo apt-get update
 3  sudo apt-get install ubuntu-tweak
 4  sudo add-apt-repository ppa:diesch/testing
 5  sudo apt-get update
 6  sudo apt-get install indicator-privacy
 7  sudo add-apt-repository ppa:atareao/atareao
 8  sudo apt-get update
 9  sudo apt-get install my-weather-indicator
```

پاک کردن تاریخچه دستورات

با فلش های بالا و پایین ، میتوانیم دستور استفاده شده قبلی را مشاهده کنیم که ممکن است مفید باشد یا ممکن است شما را عصبانی کند .حذف یا پاک کردن تمام ورودی‌ها از لیست سابقه bash با گزینه‌های ’ c- ‘.

```
[ ali@predator-os  ~]$ history -c
```

دستور builtin:

دستور builtin در محیط خود یک شل ((shellمانند Bash برای اجرای یک دستور داخلی ((built-inمشخص در خود شل به جای اجرای آن به صورت یک برنامه خارجی استفاده می‌شود. این دستور اجازه می‌دهد تا دستورات داخلی موجود در شل را به صورت مستقیم اجرا کرد.

گزینه‌های دستور:

بدون گزینه: اجرای یک دستور داخلی built-inمشخص در شل.

```
builtin echo “Hello, World!”
```

خروجی دستور: نمایش پیام “ Hello, World!”بدون ایجاد یک پردازه جدید

-p, --posix اجرای یک دستور داخلی POSIX به جای دستور داخلی اصلی.

```
builtin -p cd /home/user
```

خروجی دستور: تغییر مسیر فعلی به /home/user با استفاده از دستور داخلی POSIX cd

اجرای یک دستور داخلی مشخص به جای دستور موجود در خط فرمان.

```
builtin -c pwd
```

خروجی دستور: نمایش مسیر فعلی به جای اجرای دستور pwd به صورت خارجی

-v, --version نمایش نسخه فعلی دستور داخلی.

```
builtin -v echo
```

خروجی دستور: نمایش نسخه دستور داخلی echo

-h, --help نمایش راهنمای استفاده از دستور builtin.

```
builtin -h
```

خروجی دستور: راهنمای استفاده از دستور builtin

-q, --quiet اجرای دستور داخلی به صورت ساکت بدون نمایش پیام خروجی.

```
builtin -q pwd
```

خروجی دستور: نمایش مسیر فعلی بدون پیام خروجی

-a, --all نمایش تمامی دستورات داخلی موجود در شل.

```
builtin -a
```

خروجی دستور: نمایش لیست تمامی دستورات داخلی موجود در شل

فصل پنجم

فصل پنجم

دستورات پایه ای و مقدماتی

تفاوت بین su و sudo

سیستم لینوکس نسبت به سایر همتایان خود بسیار ایمن است .یکی از راه‌های پیاده‌سازی امنیت در لینوکس ، خط‌مشی مدیریت کاربر و مجوز کاربر است و کاربران عادی مجاز به انجام هیچ عملیات سیستمی نیستند. اگر یک کاربر معمولی نیاز به انجام هر گونه تغییر در سطح سیستم داشته باشد، باید از دستور su یا sudo استفاده کند.

su : شما را مجبور می کند **رمز عبور ریشه** خود را با سایر کاربران به اشتراک بگذارید در حالی که ' sudo ' اجرای دستورات سیستم را بدون رمز عبور **root**ممکن می کند ' sudo '.به شما امکان می دهد از رمز عبور خود برای اجرای دستورات سیستم استفاده کنید، یعنی مسئولیت سیستم را بدون رمز عبور rootمحول می کند .

sudo : یک setuidباینری ریشه است که دستورات root را از طرف کاربران مجاز اجرا می کند و کاربران باید رمز عبور خود را برای اجرای دستور سیستم و به دنبال آن ' sudo ' وارد کنند.

در تعریفی دیگر:

su : این دستور به معنای "Switch User" یا تغییر کاربر است. با استفاده از دستورsu ، شما می‌توانید از کاربر جاری خود به یک کاربر دیگر در سیستم تغییر کنید. وقتی از su استفاده می‌کنید، شما باید رمز عبور کاربر مورد نظر را وارد کنید تا بتوانید به این کاربر تغییر کنید.

sudo : این دستور به معنای "Super User Do" است و به شما اجازه می‌دهد دسترسی به دستورات مدیریتی سیستم را با دسترسی مدیر (root) داشته باشید. با استفاده ازsudo ، شما می‌توانید دستوراتی را با دسترسی مدیر اجرا کنید بدون اینکه به طور کامل به حساب rootوارد شوید. وقتی از sudo استفاده می‌کنید، شما باید رمز عبور خود را وارد کنید، نه رمز عبور کاربر .

دستور visudo

در لینوکس، دستور visudo برای ویرایش فایل تنظیمات sudoers از طریق ویرایشگر متنی vi استفاده می‌شود. فایل sudoers حاوی تنظیمات مربوط به دسترسی‌ها و مجوزهای استفاده از دستور sudo توسط کاربران سیستم است. وقتی از دستور visudo استفاده می‌کنید، فایل sudoers باز می‌شود و شما می‌توانید تنظیمات مجوزها را ویرایش کنید. این دستور از ویرایشگر vi استفاده می‌کند تا از اشتباهات جلوگیری کند و اطمینان حاصل کند که فرمت فایل sudoers صحیح باشد.

ساختار دستور:

```
visudo [-chqsV] [-f sudoers]
```

برای استفاده از دستور visudo، به عنوان کاربر root یا با دسترسی sudo، می‌توانید دستور زیر را اجرا کنید:

```
sudo visudo
```

با اجرای این دستور، فایل sudoers باز می‌شود و اگر تغییراتی اعمال می‌کنید، پس از ذخیره کردن تغییرات، فایل به صورت اتوماتیک بررسی می‌شود تا از اشتباهات جلوگیری شود.

ویرایش فایل sudoers از اهمیت بسیاری برخوردار است، زیرا تعیین مجوزها و دسترسی‌های صحیح برای کاربران و گروه‌های مختلف، از نظر امنیتی بسیار حیاتی است.

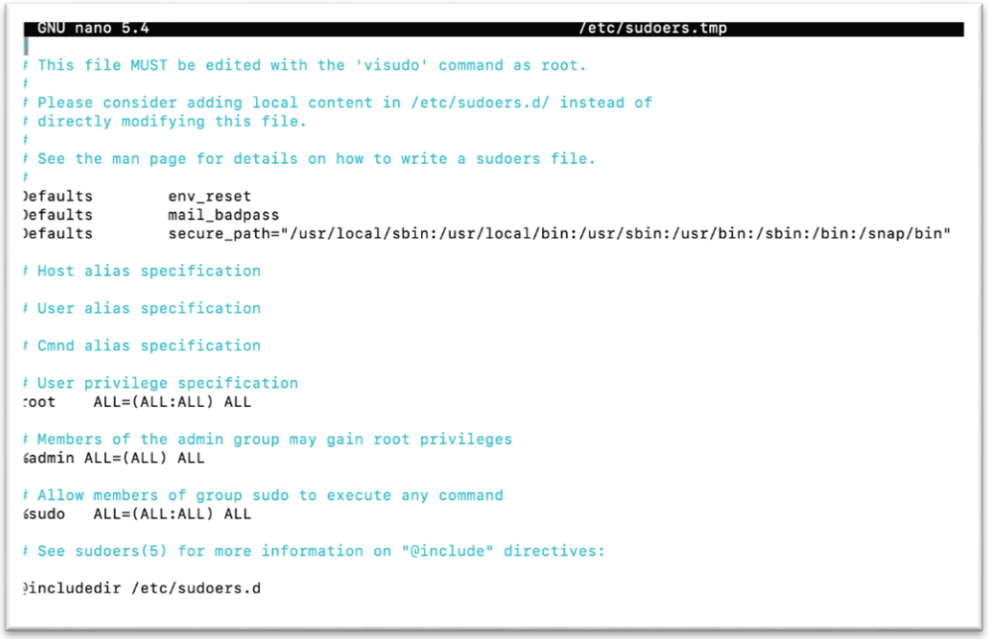
ما نیز می‌توانیم « /usr/sbin/visudo » را برای افزودن/حذف لیست کاربرانی که می‌توانند « sudo » را اجرا کنند، اجرا کنیم .

```
$ sudo /usr/sbin/visudo
```

یا دستور:

```
sudo visudo /etc/sudoers
```

اسکرین شات فایل ' /usr/sbin/visudo ' چیزی شبیه به این است:




```
root  ALL=(ALL) ALL
```

توجه :برای ویرایش فایل /usr/sbin/visudo باید در حالت rootباشید .

اعطای دسترسی sudo

در بسیاری از موقعیت‌ها، مدیر سیستم ،

```
ALL=(ALL) ALL
```

ویرایش فایل '/usr/sbin/visudo ' به چیزی شبیه الگوی زیر ممکن است واقعاً بسیار خطرناک باشد، مگر اینکه همه کاربران لیست شده را کاملاً اعتمادسازی شده باشد.

```
root ALL=(ALL) ALL
ali ALL=(ALL) ALL
hoda ALL=(ALL) ALL
hossein ALL=(ALL) ALL
```

یا اضافه کردن یک کاربر به گروه sudoers با دستور :

```
$sudo usermod -aG sudo your_username
```

مثلا:

```
$sudo usermod -aG sudo predator-os
```

دستور su

دستور su برای انتقال دسترسی از یک کاربر به یکی دیگر استفاده می شود یا در طول یک جلسه ورود، root می شود.توجه داشته باشید هنگامی که suبدون نام کاربری فراخوانی می شود، به طور پیش فرض به root تبدیل می شود. در لینوکس، su به شما اجازه می‌دهد تا دستورات را با شناسه کاربر و گروه جایگزین اجرا کنید. هنگامی که بدون آرگومان فراخوانی می شود، su به طور پیش فرض یک پوسته تعاملی را به عنوان ریشه اجرا می‌کند.

```
$ su
```

اگر می‌خواهید به کاربری به نام ali سوئیچ کنید، می‌توانید با اجرای دستور زیر این کار را انجام دهید:

```
$  su ali
```

ساختاردستور: |

دستور su به صورت زیر است:

```
su [OPTION]... [-] [USER [ARG]...]
```

گزینه های دستور:

برای گزینه :p-

```
$su -p
```

این دستور باعث می‌شود که متغیرهای محیط را بازنشانی نکند.

برای گزینه :w-

```
$su -w
```

این دستور متغیرهای مشخص شده را بازنشانی نمی‌کند.

برای گزینه :g-

```
$su -g groupname
```

این دستور گروه اولیه را به groupnameتغییر می‌دهد.

برای گزینه :G-

```
$su -G groupname
```

این دستور یک گروه مکمل را به groupnameتغییر می‌دهد.

برای گزینه :l-

```
$su -l
```

این دستور پوسته را به پوسته ورود تبدیل می‌کند.

برای گزینه :s-

```
$su -s /bin/bash
```

این دستور /bin/bashرا به عنوان پوسته اجرا می‌کند (اگر در /etc/shellاجازه دهد.)

برای گزینه :p-

```
$su -p
```

این دستور یک شبه ترمینال جدید را ایجاد می‌کند.

برای گزینه :h-

```
$su -h
```

این دستور راهنمای دستور su را نمایش می‌دهد.

برای گزینه :v-

```
$su -v
```

این دستور نسخه su را نمایش می‌دهد.

دستور sudo

دستورsudo اجازه می دهد یک کاربر مجاز برای اجرای دستور به عنوان root یا یک کاربر دیگر، همانطور که توسط سیاست امنیتی مانند sudoers تعریف شده است. در این مورد، شناسه کاربری واقعی (نه موثر) کاربر که sudoاجرا می شود برای تعیین نام کاربری که با آن به پرس و جو از سیاست امنیتی استفاده می شود استفاده می شود.

```
$ sudo apt update
$ sudo useradd  hosein
$ sudo passwd  hosein
```

دستور sudo (“substitute user do” یا “super user do”) به کاربر با مجوزهای مناسب اجازه می‌دهد تا دستوری را به عنوان کاربر دیگری مانند superuser اجرا کند. این معادل گزینه run as administrator در ویندوز است. دستور sudo به شما این امکان را می‌دهد تا حساب کاربری فعلی خود را ارتقا دهید تا از امتیازات روت برخوردار شوید. همچنین، امتیاز root در sudo فقط برای مدت زمان موقت معتبر است. پس از انقضای این زمان، باید رمز عبور خود را دوباره وارد کنید تا امتیاز root را دوباره به دست آورید.

ساختار دستور:

```
sudo -h | -K | -k | -V

sudo -v [-AknS] [-g group name | #gid] [-p prompt] [-u user name | #uid]

sudo -l[l] [-AknS] [-g group name | #gid] [-p prompt] [-U user name] [-u user name | #uid] [command]

sudo [-AbEHnPS] [-C fd] [-g group name | #gid] [-p prompt] [-r role] [-t type] [-u user name | #uid]
[VAR=value] -i | -s [command]
```

گزینه های دستور:

گزینه V:

```
$ sudo -V
```

این دستور نسخه sudo را چاپ کرده و از آن خارج می‌شود.

گزینه L:

```
$ sudo -L
```

این دستور پارامترهای تنظیم شده در sudo را فهرست می‌کند.

گزینه h:

```
$ sudo -h
```

این دستور یک پیام راجع به استفاده از sudo را چاپ کرده و خارج می‌شود.

گزینه v:

```
$ sudo -v
```

این دستور برچسب زمانی کاربر را به روز می‌کند و در صورت لزوم رمز عبور کاربر را درخواست می‌کند. تایم‌اوت sudo نیز تمدید می‌شود.

گزینه K:

```
$ sudo -K
```

این دستور برچسب زمانی کاربر را حذف می‌کند و نیازی به رمز عبور ندارد.

گزینه u:

```
$ sudo -u <username> <command>
```

این دستور دستور مشخص شده را با نام کاربری موردنظر اجرا می‌کند. می‌توانید به جای نام کاربری، یک uid را به صورت #uid استفاده کنید.

گزینه s:

```
$ sudo -s
```

این دستور پوسته مشخص شده توسط متغیر محیطی SHELL را اجرا می‌کند، یا اگر تنظیم نشده باشد، پوسته را همانطور که در فایل مشخص شده است اجرا می‌کند.

گزینه:flag--

```
$ sudo --flag <command>
```

این دستور نشان می‌دهد که sudo باید پردازش دستور را متوقف کند. آرگومان‌های خطی نیز با این گزینه و در ارتباط با گزینه s- بسیار مفید است

دسترسی کابر ریشه

اگر قصد دارید سیستم را مدیریت کنید، برای انجام وظایف به عنوان root وارد سیستم شوید. در بسیاری از موارد، شخص به عنوان یک کاربر وارد سیستم می‌شود و مایل است برخی از وظایف root سیستم مدیریت را انجام دهد. در اینجا به این صورت است:

```
su  -
```

و سپس از شما خواسته می شود رمز عبور root را وارد کنید.

اجازه دسترسی رابط کاربری گرافیکی

دستور xhost +localhost به معنای اجازه دسترسی به سرور X برای تمامی کاربران محلی localاست. با اجرای این دستور، تمام کاربران محلی می‌توانند به سرور X محلی شما متصل شوند و از امکانات گرافیکی آن استفاده کنند. اگر اجرای این دستور با موفقیت انجام شود، این بدان معنی است که تمامی کاربران محلی دسترسی به سرور X شما دارند. این کار ممکن است امنیت سیستم را کاهش دهد، بنابراین باید با دقت و با دانش کافی از این دستور استفاده کنید. در صورتی که شما می‌خواهید اجازه دسترسی به سرور X را به تمام کاربران محلی بدهید، دستور xhost +localhost یکی از روش‌های اعطای دسترسی است که می‌توانید استفاده کنید.

اجازه دسترسی به کنسول توسط کاربر دیگری از همان سیستم:

```
xhost +localhost
```

ابتدا کاربر را به ریشه تغییر دهید و سپس از شما خواسته می شود رمز عبور root را وارد کنید:

```
su -
```

متغیر محیطی را برای جلسات رابط کاربری گرافیکی تنظیم کنید تا نمایش استفاده شده نمایش محلی باشد. (به یاد داشته باشید که X- شبکه ای است که سیستم پنجره را فعال می کند windows)

```
export DISPLAY=:0.0
```

برنامه رابط کاربری گرافیکی را اجرا کنید.

نکته: دستور su با یک "-" انتهایی صادر شد که به این معنی است که فرآیند ورود به سیستم مراحل ورود کاربر را دنبال می کند و پروفایل های آن را اجرا می کند. از آنجایی که هیچ شناسه کاربری داده نشده است، پیش فرض کاربر root است. این همه معادل است با:

```
su - root
```

برخی از سیستم ها ممکن است به گونه ای پیکربندی شوند که فقط دستور سوئیچ کاربر ((suبدون تمام پیکربندی پنجره X مورد نیاز باشد.

دستور pwd

ازآنجایی که ترمینال نمیتواند تصاویر گرافیکی ساختار سیستم فایل را ارائه کند، باید روش دیگری برای نشان دادن آن را داشته باشیم. درخت سیستم فایل را به عنوان یک پیچ و خم در نظر بگیرید و در آن ایستاده اید. در هر لحظه ای مشخص،در یک فهرست قرار گرفته اید. در داخل آن فهرست، می‌توانید فایلها و مسیر آن را به فهرست اصلی خود و مسیرهای مربوط به زیر فهرست فهرست که در آنجا

ایستاده اید مشاهده کنید. فهرست که در آن ایستاده اید، فهرست کار می شود. برای پیدا کردن نام فهرست کار، از دستور pwd استفاده کنید. برای کسانی که با خط فرمان لینوکس کار می کنند، دستور pwd بسیار مفید است، که می گوید کجا هستید – در کدام فهرست، با شروع از ریشه (/). مخصوصاً برای مبتدیان لینوکس، که ممکن است هنگام ناوبری در میان فهرست ها در رابط خط فرمان گم شوند، دستور " "pwd برای نجات می آید. Pwd مخفف ' 'Print Working Directoryاست. همانطور که از نامش مشخص است، دستور pwdفهرست فعلی یا به سادگی کاربر فهرست در حال حاضر را چاپ می کند. نام فهرست فعلی را با مسیر کامل که از ریشه (/) شروع می شود چاپ می کند. این دستور در دستور shell ساخته شده است و در اکثر پوسته ها – bash، bourne shell، ksh، zsh و غیره موجود است.

```
# pwd [OPTION]
```

Options	توضیح
-L (logical)	نمایش همه لینک های نمادین–میانبرها
-P (physical)	عدم نمایش لینک های نمادین– میانبرها
–help	نمایش راهنمای دستور
–version	نمایش خروجی

```
 $  pwd
/home/me
```

دستور زیر مسیر فهرست کاری را چاپ میکند، حتی اگر حاوی پیوندهای نمادین باشد.

```
ali@predator-os :~$ /bin/pwd   -L
/home/ali/htm
```

فهرست فعلی فیزیکی

```
ali@predator-os :~$ /bin/pwd  -P
/var/www/html
```

بررسی خروجی دستور pwd و pwd -Pیکسان است یا خیر؟

```
ali@predator-os :~$ /bin/pwd
/var/www/html
```

دستور زیر تمام مکان های حاوی فایل اجرایی با نام pwd را چاپ میکند.

```
ali@predator-os :~$ type  -a  pwd

pwd is a shell builtin
pwd is /bin/pwd
```

دستور cd

برای تغییر فهرست جاری از دستور cd استفاده می کنید. برای انجام این کار، cdرا تایپ کنید و پس از نام مسیر[۱] فهرست مورد نظر را وارد نمایید. نام مسیر، مسیری است که در طول شاخه های درخت می گیرید تا به فهرست که می خواهید دسترسی پیدا کنید. نام ها را می توان در یکی از دو روش مختلف مشخص کرد.نام مسیرها به صورت مطلق یا نام آنها به صورت نسبی[۲]، بیایید ابتدا با نام مطلق[۳] نگاهی بیاندازیم.

---

[۱] path
[۲] releative path
[۳] absolute path

```
$ cd [directory_name]
```

نام محلی مطلق با فهرست ریشه شروع می شود و پس از ریشه توسط شاخه ها تا مسیر فهرست دلخواه ادامه می یابد. به عنوان مثال ، یک فهرست در سیستم وجود دارد که اکثر برنامه ها نصب شده اند. نام فهرست فهرست bin /usr/ است. این بدان معنی است که از فهرست ریشه (که با نماد اسلش / نشان داده میشود) یک فهرست به نام "usr" وجود دارد که حاوی فهرست به نام "bin" است. این مسیر یک فهرست مطلق است.

```
$ cd /usr/bin
$ pwd
/usr/bin
```

مسیر جاری bin را نشان می دهد. با دستور ls محتوای درون فهرست را مشاهده میتوان کرد.

```
$ ls
lwp-request
2to3                lwp-rget
2to3-2.6            lxterm
a2p                 lz
aalib-config        lzcat
aconnect            lzma
acpi_fakekey        lzmadec
acpi_listen         lzmainfo
add-apt-repository  m17n-db
addpart             magnifier
```

علامت نقطه در دستور cd به فهرست جاری[1] اشاره می کند و علامت ".." به فهرست قبلی از فهرست جاری اشاره می کند. هم اکنون به چگونگی کارکرد آن می پردازیم. اجازه دهید فهرست کاری را دوباره به bin / usr / تغییر دهیم:




```
cd /
```

این دستور برای تغییر فهرست به فهرست ریشه استفاده می شود ، فهرست ریشه اولین فهرست در سلسله مراتب سیستم فایل شما است.

[1] current dir



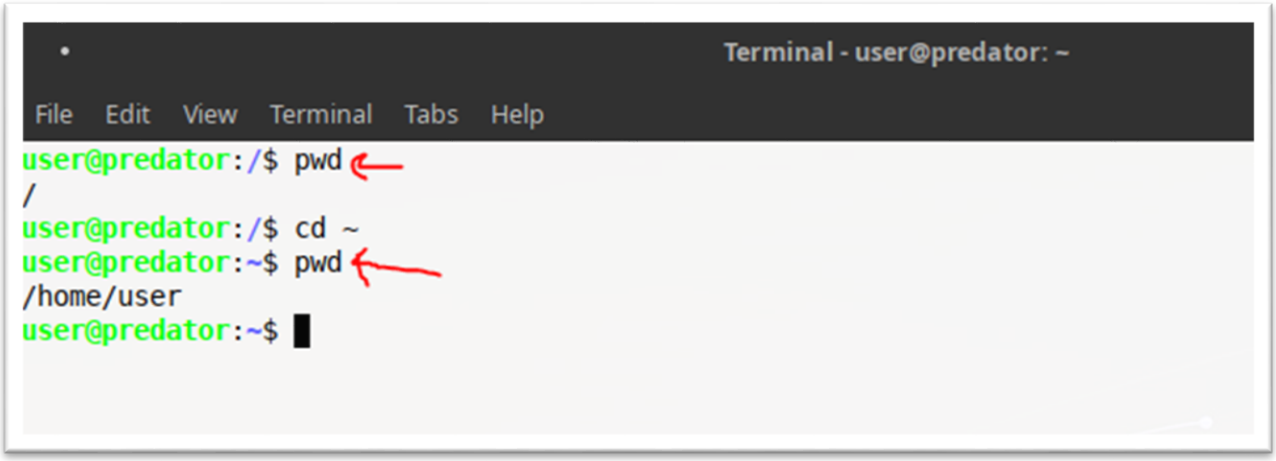
در بالا ، / نماینده فهرست ریشه است.

cd dir_1 / dir_2 / dir_3: از این دستور برای جابجایی درون فهرست از فهرست استفاده می شود.

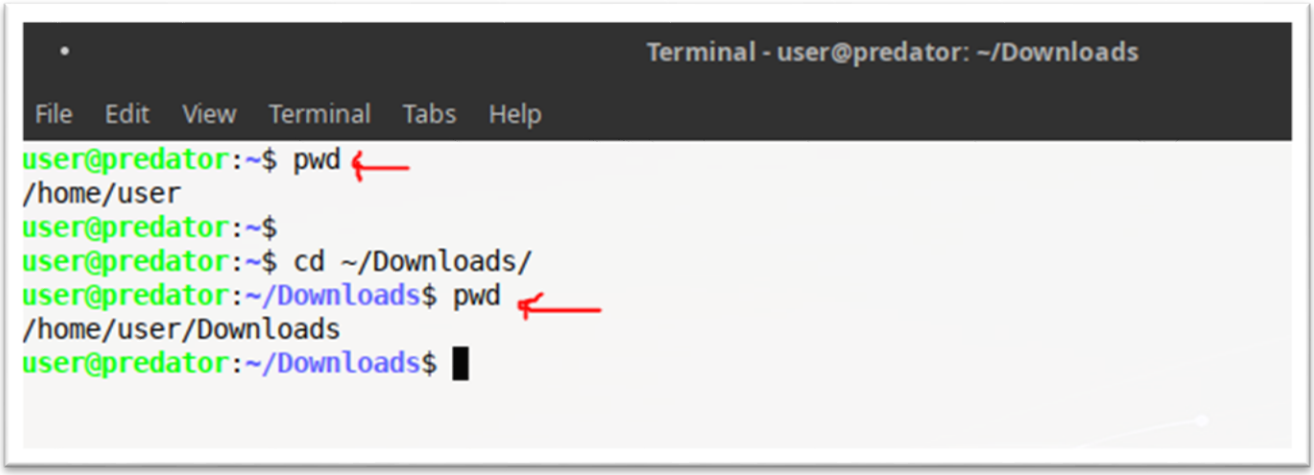
```
$ cd dir_1 / dir_2 / dir_3
$cd /home/ali/Desktop/mp3
```

```
cd ~
```

این دستور به فهرست کاربر جاری اشاره دارد.



به عنوان مثال میتوانید با دستور زیر به فهرست download از هر مسیری که باشید وارد شوید:

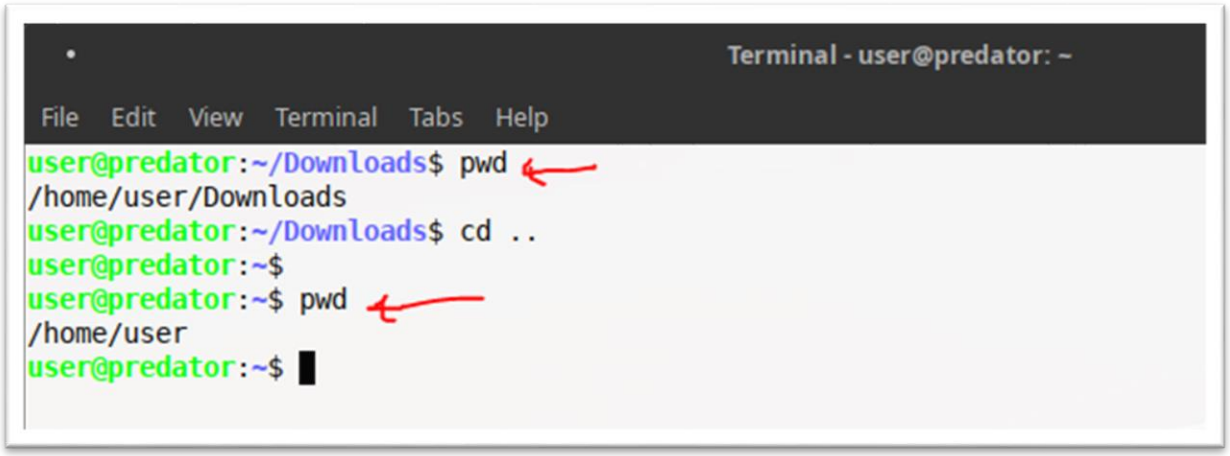


یا به مسیر فهرست کاربر جاری:

```
$cd  ~username
```

```
cd ..
```

این دستور مسیر جاری را به فهرست قبلی برمی گرداند.



دستور clear

دستور اجازه می دهد صفحه نمایش ترمینال را پاک[۱] کنید، به سادگی تایپ کنید.

```
$ clear
```

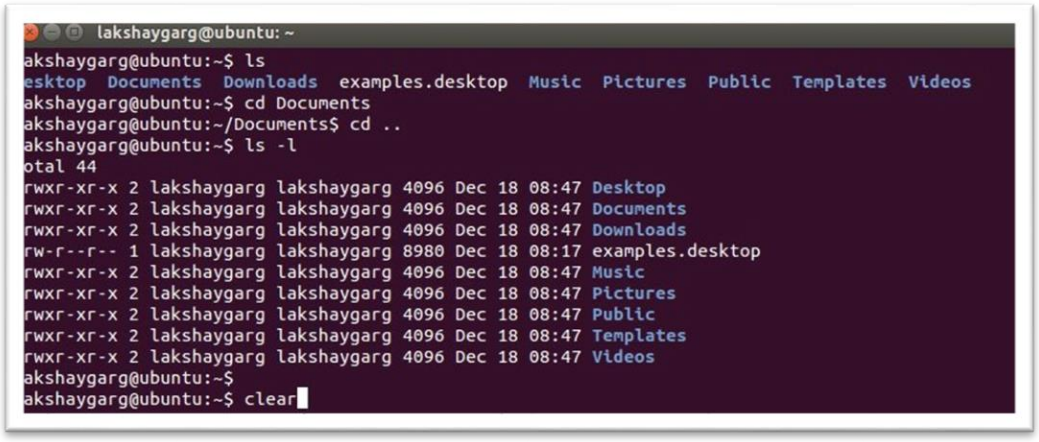
clearیک دستور استاندارد سیستم عامل رایانه یونیکس است که برای پاک کردن صفحه ترمینال استفاده می شود .این دستور ابتدا به دنبال یک نوع ترمینال در محیط است و پس از آن ، پایگاه داده **terminfo** را برای نحوه پاک کردن صفحه پیدامی‌کند و این دستورپارامترهای خط فرمان موجود را نادیده میگیرد .همچنین ، دستور **clear** هیچ سویچی ندارد و تقریباً مشابه دستور **cls** در تعدادی از سیستم عامل های دیگر است.

ساختار دستور:

```
$clear
```

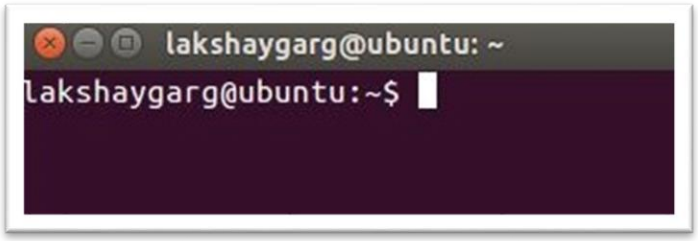
مثال های دستور:

ترمینال قبل از اجرای دستورclear :



ترمینال پس از اجرای دستور clear:

[۱] clear



میانبر دستور clear

```
ctrl + l
```

(کلید ctrl و سپس L کوچک را بزنید)

دستور sleep

دستور sleep در سیستم‌عامل‌های Unix/Linux برای توقف اجرای یک پروسه به مدت زمان مشخص شده استفاده می‌شود.

گزینه‌های دستور :

-s, --seconds=NUM-تعیین مدت زمان به واحد ثانیه.

```
sleep 5s
```

این دستور باعث توقف اجرای پروسه برای ۵ ثانیه می‌شود.

-m, --minutes=NUM-تعیین مدت زمان به واحد دقیقه.

```
sleep 3m
```

این دستور باعث توقف اجرای پروسه برای ۳ دقیقه می‌شود.

-h, --hours=NUM-تعیین مدت زمان به واحد ساعت.

```
sleep 1h
```

این دستور باعث توقف اجرای پروسه برای ۱ ساعت می‌شود.

-d, --debug-نمایش اطلاعات اشکال زدایی و اجرای دستور sleep.

```
sleep -d 10s
```

این دستور باعث توقف اجرای پروسه برای ۱۰ ثانیه می‌شود و اطلاعات اشکال زدایی نمایش داده می‌شود.

-help-نمایش راهنما و خروجی.

```
sleep --help
```

نمایش راهنما مربوط به دستور sleep.

-version-نمایش اطلاعات نسخه و خروجی.

```
sleep --version
```

نمایش اطلاعات نسخه sleep.

دستورات پیمایش

سیستم فایل های مدرن فهرست درختی دارند، جایی که یک فهرست یک فهرست root است که ما آن را "parent" می نامیم. system های Unix فقط یک فهرست root ای با نام \ دارند . هنگام کار در یک سیستم فایل ، کاربر همیشه در داخل یک فهرست کار می کند که ما آن را فهرست فعلی یا فهرست کاری[۱] می نامیم .

[۱] working or current directory

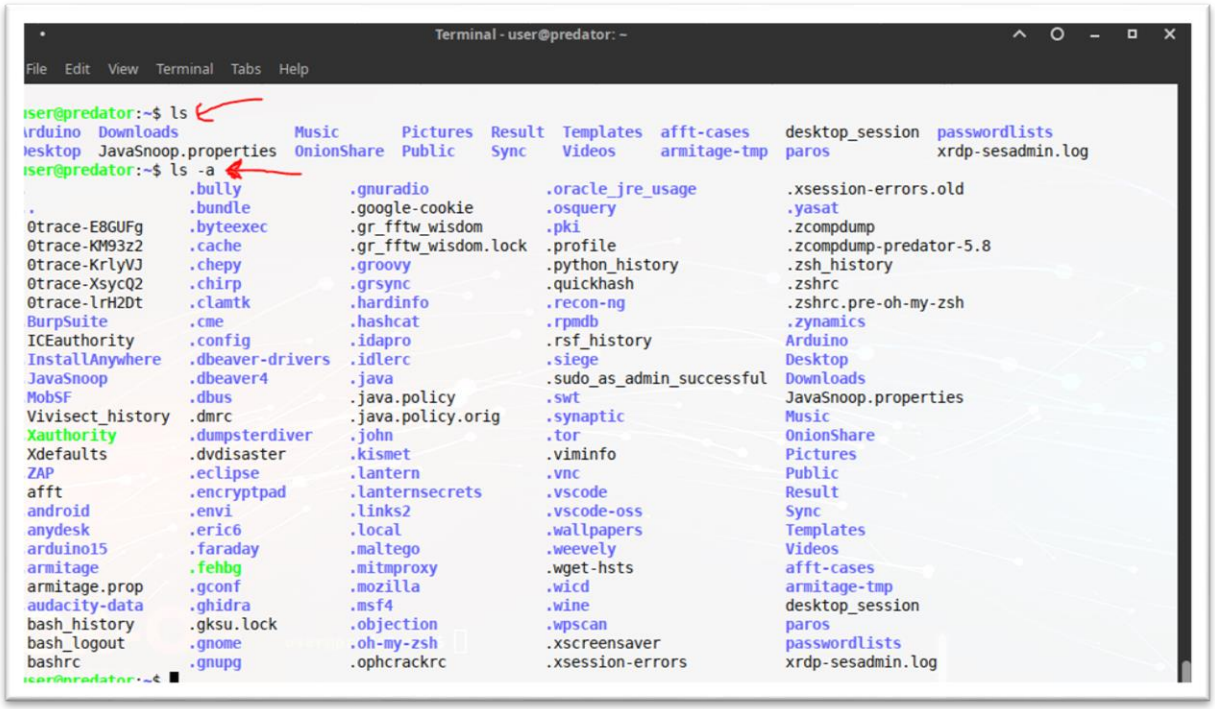
فهرست جاری کاربر را با pwd چاپ کنید :

```
$ pwd
/ home / hossein
```

لیست محتویات این فهرست (فایل ها و / یا فهرست های فرزند و غیره) را با ls :

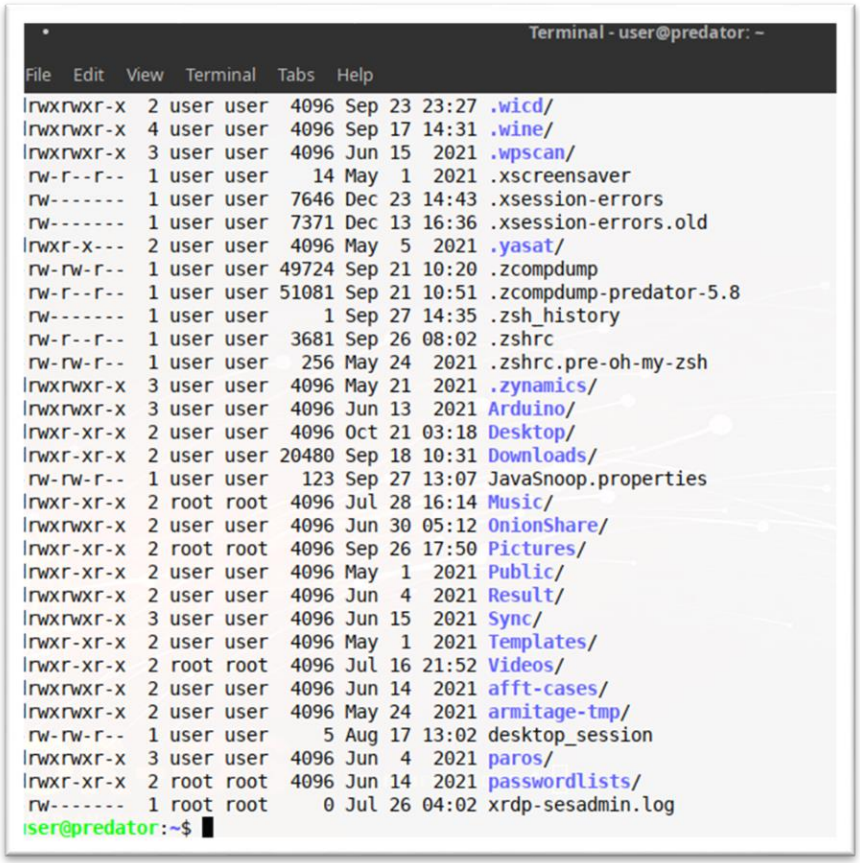
```
$ ls
```

نمایش فایل های cache (“با علامت نقطه”)



نمایش جزئیات فایل

```
$ ls -l
```



ترکیب چندین سویچ

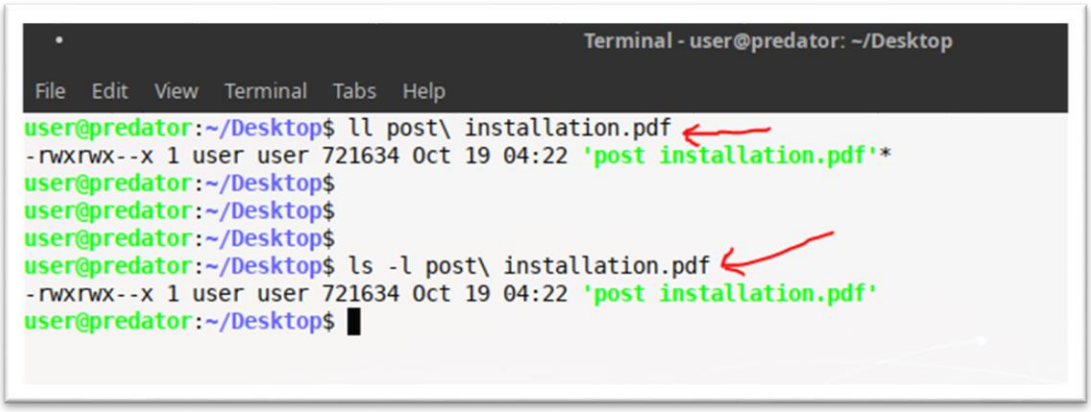
```
ls –l -a
```

دستور ls برای فهرست محتویات یک فهرست استفاده می شود .احتمالا دستور لینوکس معمولی استفاده شده است .این را می توان در تعدادی از روش های مختلف استفاده می شود .در اینجا چند نمونه است :

دستور	خروجی
ls	فهرست فایل ها را در فهرست کاری نشان می دهد
ls / bin	فهرست فایل ها را در فهرست bin / نشان می دهد
ls -l	فایلهای موجود در فهرست کاری را در قالب نوع دسترسی فهرست میکند
ls -l / etc / bin	فهرست فایل ها در فهرست bin / و فهرست etc / در فرمت دسترسی نشان می دهد
ls -la ..	لیست تمام فایل ها (حتی آنها یی که اسامی که معمولا پنهان هستند) در فهرست والدین نشان داده می شوند

نگاهی دقیق تر به فرمت خروجی دستور ls

اگر از سویچ -l با ls استفاده می کنید، لیستی را که حاوی اطلاعات فراوانی در مورد فایل مربوط است، دریافت خواهید کرد .در اینجا یک مثال داریم:



خروجی شامل اطلاعات فایل از قبیل: نام فایل، زمان ویرایش فایل، اندازه فایل، نام گروه، مالکیت فایل، مجوزهای فایل .

نام فایل

نام فایل یا فهرست .

زمان اصلاح

آخرین بار فایل تغییر کرده و اگر آخرین ویرایش گذشته، بیش از شش ماه رخ داده باشد، تاریخ و سال نمایش داده می شود .در غیر این صورت، زمان روز نمایش داده می شود .

اندازه

اندازه فایل در حالت بایت .

گروه

نام گروهی که مجوزهای فایل علاوه بر صاحب پرونده[۱] دارد .

صاحب

نام کاربری که فایل را مالکیت دارد .

مجوزهای پرونده

مجوزهای دسترسی[۲] به پرونده را نشان میدهد.

```
-rw  --  --  --- 1 bshotts bshotts 576 Apr 17 1998 weather.txt
drwxr-xr-x 6 bshotts bshotts 1024 9 ۱۹۹۹ اکتبر web_page
```

اولین کاراکتر مشخصه این است که نوعی فایل است "-" .یک فایل regular (عادی) را نشان می دهد.

"d" نشان می دهد که یک فهرست است. مجموعه دوم از سه کاراکتر، خواندنread، نوشتنwrite و اجرای execute مربوط به دسترسی صاحب فایل است .

مثال های دستور:

در اینجا، ls -l (-l)یک کاراکتر است، نه یک (فایل یا فهرست، اندازه، تاریخ و زمان اصلاح شده، نام فایل یا پوشه و صاحب فایل و مجوز آن را نشان می دهد.

```
# ls -l
```

```
total 176
-rw-r--r--. 1 root root   683 Aug 19 09:59 0001.pcap
```

---

[۱] owner

[۲] permission

```
-rw-------. 1 root root  1586 Jul 31 02:17 anaconda-ks.cfg
drwxr-xr-x. 2 root root  4096 Jul 31 02:48 Desktop

drwxr-xr-x. 2 root root  4096 Jul 31 02:48 Documents
drwxr-xr-x. 4 root root  4096 Aug 16 02:55 Downloads
-rw-r--r--. 1 root root 21262 Aug 12 12:42 fbcmd_update.php
-rw-r--r--. 1 root root 46701 Jul 31 09:58 index.html
-rw-r--r--. 1 root root 48867 Jul 31 02:17 install.log
-rw-r--r--. 1 root root 11439 Jul 31 02:13 install.log.syslog
drwxr-xr-x. 2 root root  4096 Jul 31 02:48 Music
drwxr-xr-x. 2 root root  4096 Jul 31 02:48 Pictures
drwxr-xr-x. 2 root root  4096 Jul 31 02:48 Public
drwxr-xr-x. 2 root root  4096 Jul 31 02:48 Templates
drwxr-xr-x. 2 root root  4096 Jul 31 02:48 Videos
```

مشاهده فایل های مخفی در لینوکس

فهرست همه فایل ها از جمله فایل های مخفی که با "." شروع می شود.

```
# ls -a
```

```
.                .bashrc         Documents        .gconfd
install.log      .nautilus       .pulse-cookie
..               .cache          Downloads        .gnome2
install.log      .netstat.swp    .recently-used.xbel
0001.pcap        .config         .elinks          .gnome2_private
.kde             .opera          .spice-vdagent
anaconda-ks.cfg  .cshrc          .esd_auth          .gtk-bookmarks
.libreoffice      Pictures       .tcshrc
.bash_history    .dbus           .fbcmd             .gvfs
```

فهرست فایل ها با فرمت خوانا توسط انسان

با ترکیبی از گزینه lh-، اندازه ها را در قالب قابل خواندن توسط انسان نشان می دهد.

```
# ls -lh
```

```
total 176K
-rw-r--r--. 1 root root  683 Aug 19 09:59 0001.pcap
-rw-------. 1 root root 1.6K Jul 31 02:17 anaconda-ks.cfg
drwxr-xr-x. 2 root root 4.0K Jul 31 02:48 Desktop
drwxr-xr-x. 2 root root 4.0K Jul 31 02:48 Documents
drwxr-xr-x. 4 root root 4.0K Aug 16 02:55 Downloads
-rw-r--r--. 1 root root  21K Aug 12 12:42 fbcmd_update.php
-rw-r--r--. 1 root root  46K Jul 31 09:58 index.html
-rw-r--r--. 1 root root  48K Jul 31 02:17 install.log
-rw-r--r--. 1 root root  12K Jul 31 02:13 install.log.syslog
drwxr-xr-x. 2 root root 4.0K Jul 31 02:48 Music
drwxr-xr-x. 2 root root 4.0K Jul 31 02:48 Pictures
drwxr-xr-x. 2 root root 4.0K Jul 31 02:48 Public
drwxr-xr-x. 2 root root 4.0K Jul 31 02:48 Templates
```

فهرست فایل ها و فهرست ها با نویسه "/" در پایان

استفاده از گزینه F- با دستور ls، کاراکتر '/' را در انتهای هر فهرست اضافه می کند.

```
# ls -F
```

```
0001.pcap         Desktop/     Downloads/         index.html
install.log.syslog  Pictures/  Templates/
onda-ks.cfg  Documents/  fbcmd_update.php  install.log
Music/               Public/     Videos/
```

فهرست کردن فایل ها به ترتیب معکوس در لینوکس

دستور زیر با گزینه r- ls فایل ها و فهرست ها را به ترتیب معکوس نمایش می دهد.

```
# ls -r
```

```
Videos      Public     Music                install.log
fbcmd_update.php  Documents  anaconda-ks.cfg
Templates   Pictures   install.log.syslog   index.html
Downloads           Desktop     0001.pcap
```

فهرست فهرست های فرعی را در لینوکس به صورت بازگشتی

گزینه R- ls درخت های فهرست فهرست بسیار طولانی را فهرست می کند. نمونه ای از خروجی دستور را ببینید.

```
# ls -R
```

```
total 1384
-rw-------. 1 root     root      33408 Aug  8 17:25 anaconda.log
-rw-------. 1 root     root      30508 Aug  8 17:25 anaconda.program.log

./httpd:
total 132
```

فهرست کردن فایل ها و فهرست ها به ترتیب معکوس در لینوکس

ترکیبی از ltr– آخرین ویرایش فایل یا تاریخ فهرست را نشان می دهد.

```
# ls -ltr
```

```
Total 176
-rw-r--r--. 1 root root 11439 Jul 31 02:13 install.log.syslog
-rw-r--r--. 1 root root 48867 Jul 31 02:17 install.log
-rw-------. 1 root root  1586 Jul 31 02:17 anaconda-ks.cfg
drwxr-xr-x. 2 root root  4096 Jul 31 02:48 Desktop
drwxr-xr-x. 2 root root  4096 Jul 31 02:48 Videos
drwxr-xr-x. 2 root root  4096 Jul 31 02:48 Templates
drwxr-xr-x. 2 root root  4096 Jul 31 02:48 Public
drwxr-xr-x. 2 root root  4096 Jul 31 02:48 Pictures
```

مرتب سازی فایل ها بر اساس اندازه فایل در لینوکس

با ترکیبی از lS– اندازه فایل را به ترتیب نمایش می دهد، ابتدا در اندازه بزرگ نمایش داده می شود.

```
# ls -lS
```

```
total 176
-rw-r--r--. 1 root root 48867 Jul 31 02:17 install.log
-rw-r--r--. 1 root root 46701 Jul 31 09:58 index.html
-rw-r--r--. 1 root root 21262 Aug 12 12:42 fbcmd_update.php
-rw-r--r--. 1 root root 11439 Jul 31 02:13 install.log.syslog
drwxr-xr-x. 2 root root  4096 Jul 31 02:48 Desktop
drwxr-xr-x. 2 root root  4096 Jul 31 02:48 Documents
drwxr-xr-x. 4 root root  4096 Aug 16 02:55 Downloads
```

```
drwxr-xr-x. 2 root root  4096 Jul 31 02:48 Music
```

نمایش شماره Inode فایل یا فهرست

می‌توانیم تعدادی اعداد را که قبل از نام فایل/فهرست چاپ شده‌اند، ببینیم. با گزینه های –i فایل/فهرست را با یک عدد inode لیست می کند.

```
# ls -i
```

```
2112 0001.pcap        23610 Documents         23793 index.html
23611 Music            23597 Templates         23564 anaconda-ks.cfg
23595 Downloads        22 install.log         23612 Pictures  23613 Videos
23594 Desktop          23585 fbcmd_update.php     35 install.log.syslog
23601 Public
```

بررسی نسخه دستور ls

نسخه دستور ls را بررسی کنید.

```
# ls --version
```

نمایش UID و GID فایل ها

برای نمایش UID و GID فایل ها و فهرست ها. از گزینه –n با دستور ls استفاده کنید.

```
# ls -n
```

```
total 36
drxr-xr-x. 2 500 500 4096 Aug  2 01:52 Downloads
drwxr-xr-x. 2 500 500 4096 Aug  2 01:52 Music
drwxr-xr-x. 2 500 500 4096 Aug  2 01:52 Pictures
```

دستور ls در یک نگاه

گزینه ls	توضیحات
ls -a	در لینوکس، فایل‌های مخفی با نماد . (نقطه) شروع می‌شوند و در فهرست معمولی قابل مشاهده نیستند. دستور (ls -a) لیست کامل فهرست فعلی را با اعمال فایل‌های مخفی نشان می‌دهد.
ls -l	لیست را به صورت فرمت لیست بلند نمایش می‌دهد.
ls -lh	این دستور اندازه فایل‌ها را به صورت خوانا برای انسان نمایش می‌دهد. اندازه فایل بسیار دشوار است وقتی به صورت بایت نمایش داده می‌شود. دستور (ls -lh) اطلاعات را به صورت مگابایت، گیگابایت، ترابایت و غیره نمایش می‌دهد.
ls -lhS	اگر می‌خواهید فایل‌های خود را به ترتیب نزولی (بزرگترین در بالا) بر اساس اندازه‌شان نمایش دهید، می‌توانید از دستور (ls -lhS) استفاده کنید.
ls -l --block-size=[SIZE]	برای نمایش فایل‌ها در یک فرمت اندازه خاص استفاده می‌شود. در اینجا، در [SIZE] می‌توانید اندازه را بر اساس نیاز خود تعیین کنید.
ls -d */	برای نمایش تنها زیرفهرست‌ها استفاده می‌شود.
ls -lG یا ls -g	با این دستور می‌توانید ستون اطلاعات گروه و مالک را حذف کنید.
ls -n	برای چاپ شناسه گروه و شناسه مالک به جای نام‌هایشان استفاده می‌شود.
ls --color=[VALUE]	این دستور برای چاپ لیست به رنگی یا بی‌رنگ استفاده می‌شود.

ls -li	این دستور شماره شاخص را چاپ می‌کند اگر فایل در ستون اول باشد.
ls -p	برای شناسایی آسان فهرست با علامت خط (/) نشان می‌دهد.
ls -r	برای چاپ لیست به ترتیب برعکس استفاده می‌شود.
ls -R	محتوای زیرفهرست‌ها را نیز نمایش می‌دهد.
ls -lX	فایل‌ها با انتهای یکسان را در لیست گروه‌بندی می‌کند.
ls -lt	با نمایش فایل‌های اخیراً تغییر یافته در بالای لیست، لیست را مرتب می‌کند.
ls ~	محتوای فهرست خانه را نشان می‌دهد.
ls ../	محتوای فهرست والدین را نشان می‌دهد.
ls --version	ورژن دستور ls را بررسی می‌کند.

تغییر به یک فهرست دیگر با cd

در لینوکس دستور (cdتغییر فهرست) یکی از مهمترین و پرکاربردترین دستورات برای مبتدیان و همچنین مدیران سیستم است. برای ادمین‌های سرور cdتنها راه برای پیمایش به فهرست برای بررسی گزارش، اجرای یک برنامه و برای هر کار دیگری است. برای مبتدیان این یکی از دستورات اولیه است که با آن دست های خود را کثیف می کنند.

```
$ cd TEST
```

```
cd  A
pwd
A
cd ..
```

برای تغییر به فهرست parent" استفاده می شود:

```
# cd..
/ home / hossein / TEST
```

cd ~ یا فقط cd برای " تغییر مسیر به به فهرست اصلی است.

```
cd~
```

نکته

cd ~ کاربر به معنای " cd به فهرست خانگی کاربر است

می توانید چند فهرست را با cd .. /..تغییر دهید

```
cd ../..
```

بازگشت به آخرین فهرست با cd–

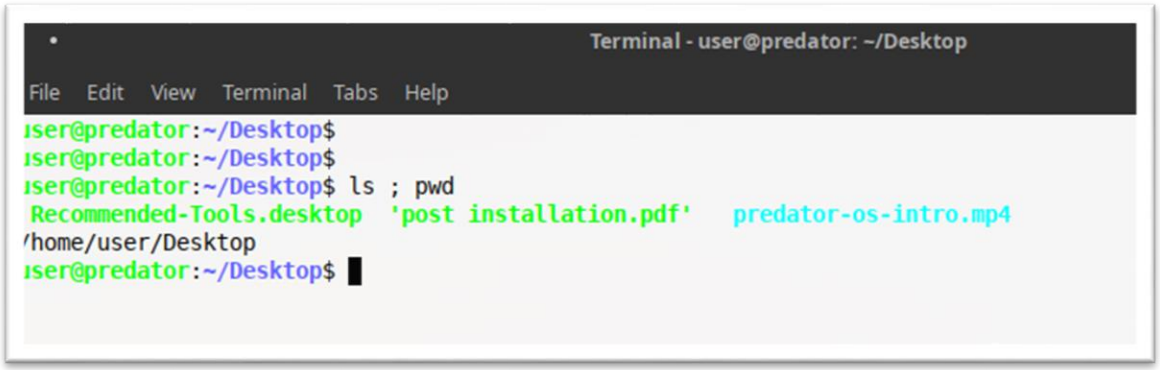
```
cd -
```

چیزهایی که ما در خط دستور وارد میکنیم ، دستورات نامیده میشوند و آنها همواره برخی از کد ماشین[۱] را که در جایی از رایانه ذخیره شده اند اجرا می کنند. گاهی اوقات این کد دستگاه یک دستور linux داخلی است، گاهی اوقات این برنامه است. گاهی اوقات، ما می خواهیم یک دستور را درست بعد از دیگری اجرا کنیم. برای انجام این کار می توانیم از ؛ (سمیکالن[۲]) استفاده کنیم.

---

[۱] machine code

[۲] semicolon



در بالا، semicolon به این معنی است که من برای اولین بار (ls) فهرست محتویات فهرست کار را لیست کرده و سپس (pwd) محل فهرست جاری خود را چاپ می کنم.

مثال های دستور

۱.از فهرست فعلی به /usr/local تغییر مسیر دهید.

```
ali@predator-os : ~$ cd /usr/local
ali @ predator-os :/usr/local$
```

۲. با استفاده از مسیر مطلق، از فهرست فعلی به /usr/local/lib تغییر دهید.

```
ali@predator-os :/usr/local$ cd /usr/local/lib

ali@predator-os :/usr/local/lib$
```

۳. با استفاده از مسیر نسبی، از فهرست کاری فعلی به /usr/local/lib تغییر دهید.

```
mypc@predator-os :/usr/local$ cd lib

mypc@predator-os :/usr/local/lib$
```

۴. به فهرست قبلی که قبلاً در آن کار می کردید، برگردید.

```
mypc@predator-os :/usr/local/lib$ cd -

/usr/local
mypc@predator-os :/usr/local$
```

فهرست فعلی را به فهرست والد تغییر دهید.

```
mypc@predator-os :/usr/local/lib$ cd ..

mypc@predator-os :/usr/local$
```

۵.نمایش آخرین فهرست کاری از جایی که ما حرکت کردیم (از سوئیچ '–' استفاده کنید) همانطور که نشان داده شده است.

```
mypc@predator-os :/usr/local$ cd --
/home/mypc
```

۶.دو فهرست را از جایی که اکنون هستید به بالا ببرید.

```
mypc@predator-os :/usr/local$ cd ../ ../

mypc@predator-os :/usr$
```

۷. از هر کجا به فهرست اصلی کاربران بروید.

```
mypc@predator-os :/usr/local$ cd ~
```

```
mypc@predator-os :~$
یا
mypc@predator-os :/usr/local$ cd
mypc@predator-os :~$
```

۸.فهرست کاری را به فهرست کاری فعلی تغییر دهید (به نظر می رسد در حالت کلی هیچ استفاده ای از آن وجود ندارد).

```
mypc@predator-os :cd ~/Downloads
```

۹. فهرست فعلی شما "/usr/local/lib/python3.4/dist-packages/" است، آن را در یک دستور با بالا رفتن در فهرست به "/home/mypc/Desktop/"تغییر دهید. '/' سپس با استفاده از مسیر مطلق.

```
mypc@predator-os :/usr/local/lib/python3.4/dist-packages$ cd ../../../../../home/mypc/Desktop/

mypc@predator-os :~/Desktop$
```

۱۰. از فهرست کاری فعلی به /var/www/html بدون تایپ کامل با استفاده از TAB تغییر دهید.

```
mypc@predator-os :/var/www$ cd /v<TAB>/w<TAB>/h<TAB>

mypc@predator-os :/var/www/html$
```

۱۱. از فهرست کاری فعلی خود به /etc/v___ _ بروید، اوه !شما نام فهرست را فراموش کرده اید و قرار نیست از TAB استفاده کنید.

```
mypc@predator-os :~$ cd /etc/v*

mypc@predator-os :/etc/vbox$
```

توجه :فقط در صورتی که فقط یک فهرست وجود داشته باشد که با " v " شروع می شود، به " vbox " منتقل می شود .اگر بیش از یک فهرست که با ' v ' شروع می شود وجود داشته باشد، و هیچ معیار بیشتری در خط فرمان ارائه نشده باشد، به اولین فهرست که با ' v ' شروع می شود، بر اساس حروف الفبا به عنوان حضور آنها در فرهنگ لغت استاندارد منتقل می شود.

۱۲. بدون استفاده از TAB

باید مسیر کامل را وارد نمایید :

```
mypc@predator-os : cd /home/ali
mypc@predator-os :~$
```

۱۳. Pushd و Popd در لینوکس چیست؟

Pushdو popd دستورات لینوکس در bash و پوسته خاصی هستند که مکان فهرست کاری فعلی را در حافظه ذخیره می کنند و به ترتیب از حافظه به عنوان فهرست کاری فعلی و همچنین فهرست تغییرات به فهرست می آورند.

```
mypc@predator-os :~$ pushd /var/www/html
/var/www/html
mypc@predator-os :/var/www/html$
```

دستور بالا مکان فعلی را در حافظه ذخیره می کند و به فهرست درخواستی تغییر می دهد .به محض اینکه popd فعال می شود، مکان فهرست ذخیره شده را از حافظه دریافت می کند و آن را به فهرست فعلی تبدیل می کند.

```
mypc@predator-os :/var/www/html$ popd
~
mypc@predator-os :~$
```

۱۴. به فهرست حاوی فاصله با استفاده از علامت اسلش= \ تغییر مسیردهید.

```
mypc@predator-os :~$ cd test\predator-os /

mypc@ predator-os :~/test predator-os $

یا

mypc@predator-os :~$ cd ‘test predator-os ‘
 mypc@predator-os :~/test predator-os $

 یا

mypc@predator-os :~$ cd “test predator-os “/
 mypc@predator-os :~/test predator-os $
```

۱۵. از فهرست کاری فعلی به Downloads تغییر مسیردهید و تمام تنظیمات آن را یکجا فهرست کنید.

```
mypc@predator-os :/usr$ cd ~/downloads && ls
```

دستور dir

در این قسمت مثال هایی از دستور dir برای فهرست کردن محتویات یک فهرست را نشان می دهد.. اگرچه کمتر شبیه دستور ls است که اکثر کاربران لینوکس ترجیح می دهند از آن استفاده کنند. ما در مورد دستور dir بحث خواهیم کرد که در آن نحوه استفاده از گزینه ها و آرگومان های مختلف را بررسی خواهیم کرد.

ساختار دستور:

```
# dir [OPTION] [FILE]
```



خروجی دستور:

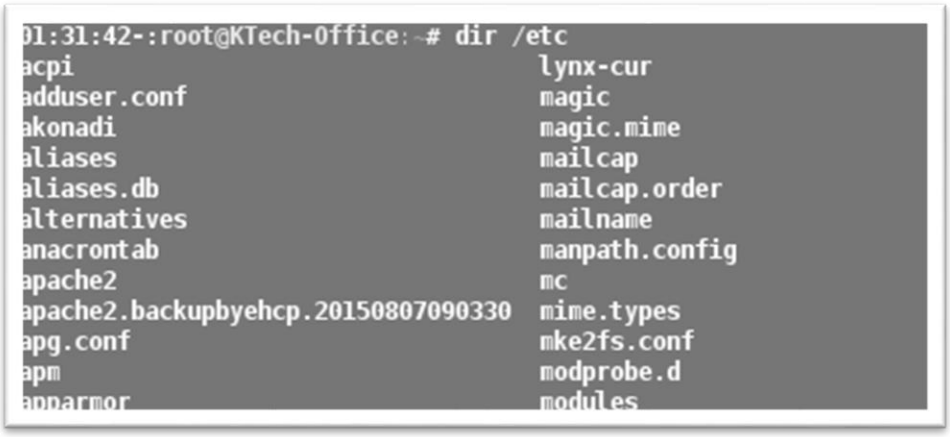
```
# dir /
```



خروجی دستور:

خروجی دستور dirبا فایل فهرست etc/به شرح زیر است.همانطور که از خروجی می بینید، تمام فایلهای موجود در فهرست etc /لیست نشده اند.

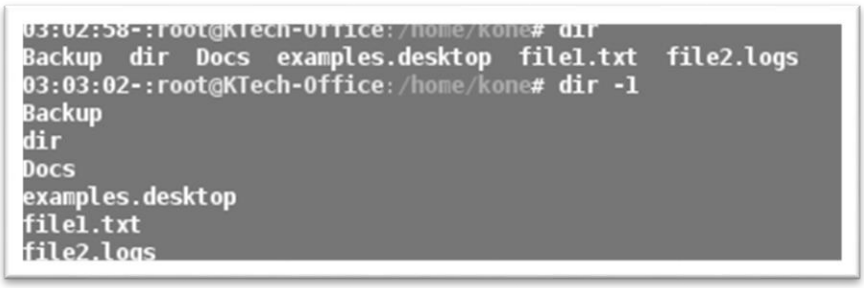
```
# dir / etc
```



لیست / etc فهرست

برای لیست یک فایل در هر خط از گزینه 1-به صورت زیر استفاده کنید.

```
# dir
 # dir -1
```



لیست فایل ها در هر خط

مشاهده تمام فایل ها

برای فهرست همه فایل ها در یک فهرست از جمله فایل های مخفی، از گزینه a-استفاده کنید. می توانید گزینه l-را برای فرمت خروجی به عنوان یک لیست اضافه کنید.

```
# dir -a
# dir -al
```



فهرست فایل های مخفی

```
01:23:56-:root@KTech-Office:~# dir -al
total 188
drwxr-xr-x 29 kili kili  4096 Hag 14 01:00 .
drwxr-xr-x  5 root root  4096 Hag  7 12:00 ..
drwx------  3 kili kili  4096 Ado 28 15:56 .adobe
drwxr-xr-x  3 kili kili  4096 Hag 14 00:09 BACKUP
-rw-------  1 kili kili 12237 Hag 11 21:03 .bash_history
-rw-r--r--  1 kili kili   220 Ado 24 05:21 .bash_logout
-rw-r--r--  1 kili kili  3643 Ado 24 12:23 .bashrc
drwx------ 24 kili kili  4096 Hag 12 00:57 .cache
drwx------  3 kili kili  4096 Ado 26 06:43 .compiz
drwx------ 25 kili kili  4096 Hag  8 21:47 .config
drwx------  3 root root  4096 Hag  7 12:07 .dbus
drwxr-xr-x  2 kili kili  4096 Ado 26 06:44 Desktop
-rw-r--r--  1 kili kili    29 Hag  7 13:32 .dmrc
drwxr-xr-x  4 kili kili  4096 Hag 14 01:17 Documents
```

فهرست صفحات cache فایل

نمایشهای فهرست به جای محتوا

هنگامی که فقط باید به جای محتویات فهرست فهرست را وارد کنید، می توانید از گزینه d-استفاده کنید.در خروجی زیر گزینه d-لیست ورودی ها را برای فهرست etc /قرار می دهد.

```
# dir -d / etc
# dir -dl / etc
```

```
01:32:13-:root@KTech-Office:~# dir -d /etc
/etc
01:35:13-:root@KTech-Office:~# dir -dl /etc
drwxr-xr-x 164 root root 12288 Hag 13 23:54 /etc
01:35:19-:root@KTech-Office:~#
```

فهرست \

نمایش شماره فهرست فایلها

در صورتی که می خواهید شماره شاخص هر فایل را ببینید، از گزینه i-استفاده کنید.از خروجی زیر، می توانید ببینید که ستون اول شماره ها را نشان می دهد.این اعداد به نام inodes گفته می شوند که بعضی اوقات به عنوان گره های شاخص یا عدد شاخص اشاره می شود. Inodeدر system های linux ذخیره سازی داده ها در یک سیستم فایل است که اطلاعات مربوط به یک فایل را ذخیره می کند به جز نام فایل و داده های واقعی آن.

```
# dir -il
```

```
01:45:34-:root@KTech-Office:~# dir -il
total 44
3151642 drwxr-xr-x  3 kili kili 4096 Hag 14 00:09 BACKUP
2883594 drwxr-xr-x  2 kili kili 4096 Ado 26 06:44 Desktop
2883598 drwxr-xr-x  4 kili kili 4096 Hag 14 01:17 Documents
2883595 drwxr-xr-x 12 kili kili 4096 Hag 12 00:29 Downloads
3145786 drwxr-xr-x 21 kili kili 4096 Hag  7 11:34 ehcp
2890663 -rw-rw-r--  2 kili kili    0 Hag 13 21:13 file_hardlink
2897228 lrwxrwxrwx  1 kili kili    8 Hag 13 21:44 file_symlink -> file.txt
2890663 -rw-rw-r--  2 kili kili    0 Hag 13 21:13 file.txt
2883599 drwxr-xr-x  2 kili kili 4096 Ado 24 05:27 Music
2883600 drwxr-xr-x  3 kili kili 4096 Ado 31 16:40 Pictures
2883597 drwxr-xr-x  2 kili kili 4096 Ado 24 05:27 Public
2883840 drwxrwxr-x 20 kili kili 4096 Ado  2 12:23 SmartSACCO
2883596 drwxr-xr-x  2 kili kili 4096 Ado 24 05:27 Templates
2897890 lrwxrwxrwx  1 kili kili    4 Hag 14 00:10 tmp-link -> /tmp
2883601 drwxr-xr-x  2 kili kili 4096 Hag 13 21:52 Videos
```

تعداد فایلها و فهرست ها

لیست فایل ها و اندازه های اختصاص داده شده آنها را در بلوک ها می توانید اندازه فایل ها را با استفاده از گزینه **-s** مشاهده کنید.

در این مورد همچنین باید از گزینه h– برای مشاهده اندازه فایل ها در قالب فرمت قابل خواندن استفاده کنید.

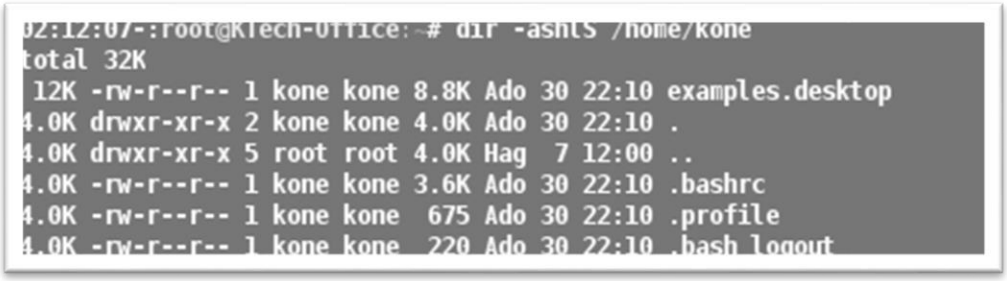
```
# dir -shl
```

```
02:04:33-:root@KTech-Office:~# dir -shl
total 44K
4.0K drwxr-xr-x  3 kili kili 4.0K Hag 14 00:09 BACKUP
4.0K drwxr-xr-x  2 kili kili 4.0K Ado 26 06:44 Desktop
4.0K drwxr-xr-x  4 kili kili 4.0K Hag 14 01:17 Documents
4.0K drwxr-xr-x 12 kili kili 4.0K Hag 12 00:29 Downloads
4.0K drwxr-xr-x 21 kili kili 4.0K Hag  7 11:34 ehcp
   0 -rw-rw-r--  2 kili kili    0 Hag 13 21:13 file_hardlink
   0 lrwxrwxrwx  1 kili kili    8 Hag 13 21:44 file_symlink -> file.txt
   0 -rw-rw-r--  2 kili kili    0 Hag 13 21:13 file.txt
4.0K drwxr-xr-x  2 kili kili 4.0K Ado 24 05:27 Music
4.0K drwxr-xr-x  3 kili kili 4.0K Ado 31 16:40 Pictures
4.0K drwxr-xr-x  2 kili kili 4.0K Ado 24 05:27 Public
4.0K drwxrwxr-x 20 kili kili 4.0K Ado  2 12:23 SmartSACCO
4.0K drwxr-xr-x  2 kili kili 4.0K Ado 24 05:27 Templates
   0 lrwxrwxrwx  1 kili kili    4 Hag 14 00:10 tmp-link -> /tmp
4.0K drwxr-xr-x  2 kili kili 4.0K Hag 13 21:52 Videos
```

لیست فایل ها با اندازه

در خروجی بالا، ستون اول حجم فایل ها را در کیلوبایت نشان می دهد.خروجی زیر یک لیست مرتب شده از فایل ها را با توجه به اندازه آنها با استفاده از گزینه S-نشان می دهد.

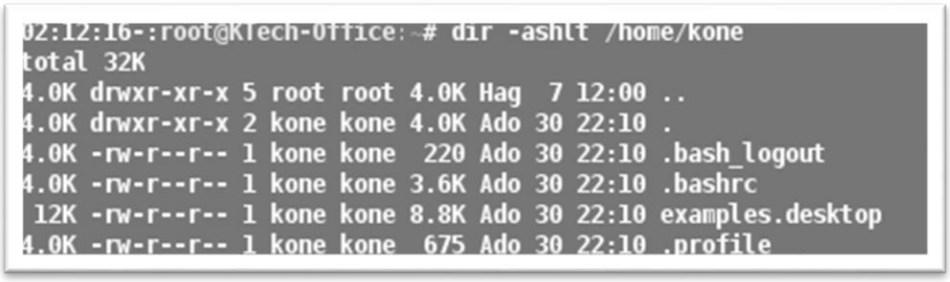
```
# dir -ashlS /home/ali
```



مرتب سازی فایل ها برحسب سایز

همچنین می توانید مرتب سازی[1] بر اساس زمان اصلاح، با فایل که به تازگی اصلاح شده ظاهر می شود در ابتدا در لیست.این را می توان با استفاده از گزینه t-انجام داد.

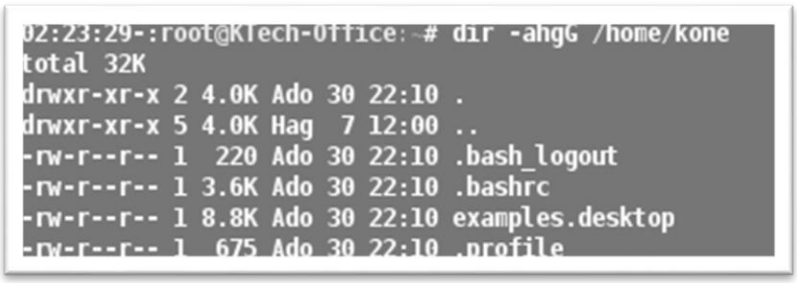
```
# dir -shlt / home / ali
```



فهرست فایلها بدون مالک یا مالک گروه

برای لیست کردن فایلها بدون نام مالک آنها ، باید از گزینه g-که مانند گزینه l-کار می کند فقط از آن استفاده کنید که صاحب فایل را چاپ نکند.و برای لیست کردن فایلها بدون مالک گروه از گزینه G-به صورت زیر استفاده کنید.

```
# dir  -ahgG  /home/ali
```



لیست فایلها بدون مالک

همانطورکه می توانید از خروجی بالا مشخص کنید نام مالک فایل و مالک گروه چاپ نمی شود. همچنین می توانید نویسنده فایل را با استفاده از سویچ author-به شرح زیر مشاهده کنید.

```
# dir  -al   --author / home / ali
```

[1] sort

```
02:34:47-:root@KTech-Office:~# dir -al --author /home/kone/
total 32
drwxr-xr-x 2 kone kone kone 4096 Ado 30 22:10 .
drwxr-xr-x 5 root root root 4096 Hag  7 12:00 ..
-rw-r--r-- 1 kone kone kone  220 Ado 30 22:10 .bash_logout
-rw-r--r-- 1 kone kone kone 3637 Ado 30 22:10 .bashrc
-rw-r--r-- 1 kone kili kone 8980 Ado 30 22:10 examples.desktop
-rw-r--r-- 1 kone kone kone  675 Ado 30 22:10 .profile
```

مشاهده نویسنده فایلها

فهرست ها را قبل از فایل های دیگر فهرست کنید

ممکن است بخواهید قبل از تمام فایل های دیگر فهرست را مشاهده کنید و این کار را با استفاده از سویچ: group-directories first به صورت زیر انجام دهید.

```
# dir -l  --group-directories-first
```

```
02:45:59-:root@KTech-Office:/home/kone# dir -l --group-directories-first
total 24
drwxr-xr-x 2 root root 4096 Hag 14 02:44 Backup
drwxr-xr-x 2 root root 4096 Hag 14 02:44 dir
drwxr-xr-x 2 root root 4096 Hag 14 02:44 Docs
-rw-r--r-- 1 kone kili 8980 Ado 30 22:10 examples.desktop
-rw-r--r-- 1 root root    0 Hag 14 02:45 file1.txt
-rw-r--r-- 1 root root    0 Hag 14 02:45 file2.logs
```

لیست کردن فهرستهای داخلی یک فهرست

می توانید زیر فهرست ها را به صورت بازگشتی مشاهده کنید، به این معنی که می توانید تمام زیر شاخه های دیگر در یک فهرست را با استفاده از گزینه R-به صورت زیر فهرست کنید.

```
# dir -R
```

```
03:10:25-:root@KTech-Office:/home/kone# dir -R
.:
Backup  dir  Docs  examples.desktop  file1.txt  file2.logs

./Backup:
mariadb  mysql

./Backup/mariadb:

./Backup/mysql:

./dir:

./Docs:
Books  Tuts

./Docs/Books:

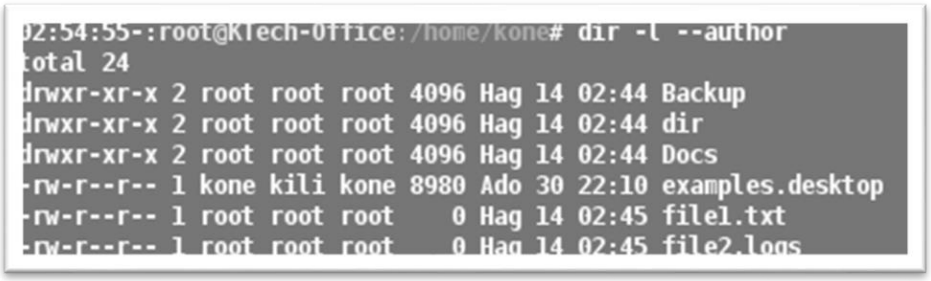
./Docs/Tuts:
03:11:11-:root@KTech-Office:/home/kone#
```

لیست فهرستها را به صورت مجزا در خروجی بالا، علامت (.)یعنی فهرست فعلی است. سه زیر فهرست را دار است که Backup، dir و Docs نام دارد. زیرگروهBackup دارای دو زیرشاخه دیگر است که mariadbو mysqlهستند که

هیچ فهرستی ندارد. به جای اسمها، شناسه کاربری و گروه را مشاهده کنید برای مشاهده **شناسه** کاربری و گروه، باید از گزینه - n استفاده کنید.بگذارید تفاوت بین دو خروجی بعدی را مشاهده کنیم.

خروجی بدون گزینه n–

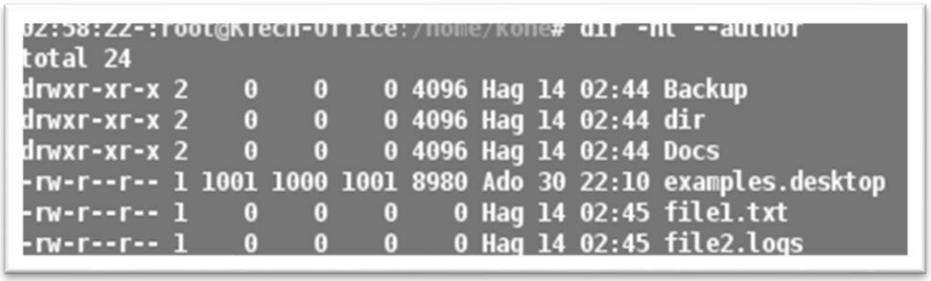
```
# dir -l  --author
```



فهرست فایلها بدون شناسه

خروجی با سویچ n-

```
# dir -nl  --author
```



لیست فایل ها با شناسه

میتوانید بین اسم فهرست ها با یک کاما جدا کنید که با سویچ m–صورت میگیرد.

```
# dir -am
```



دستور dirs

این دستوریک دستور shell builtin برای نمایش لیستی از فهرست های موجود در حال حاضر به کار می رود .به طور پیش فرض ، این فهرست شامل فهرست ای است که شما در حال حاضر در آن هستید. یک فهرست می تواند از طریق دستور **pushd به** دنبال آن نام dir وارد لیست شود و از طریق دستور **popd** قابل حذف است.

ساختار دستور:

```
dirs [-clpv] [+ N] [-N]
```

این برای جریان آسان و سریع از طریق فهرستهای موجود در خط فرمان مفید است .فرض کنید می خواهید به یک فهرست بروید ، فقط می توانید از دستور **pushd dir-name** استفاده کنید و به آن فهرست هدایت می شود .همچنین ، با استفاده از دستور **popd** می توانید به سادگی به قبلی برگردید.

گزینه های دستور:

c :لیست کاملی از فهرستهای ذخیره شده را پاک می کند.

- **l**:مسیر فهرست اصلی خود را در path-name فهرست خود نشان نمیدهد.
- **p**: نمایش یک ورودی در هر خط.
- **v**:نمایش یک ورودی در هر پیشوند با پیشوند شاخص با شروع با ۰.

مولفه های:

- **N +**:ورودی N را از سمت چپ نشان می دهد.
- **N-**:ورودی N را از راست نشان می دهد.

مثال های دستور:

عملکرد اساسی:

```
vivek@vivek-X556UQK:~/Rocket$ dirs
~/Rocket /dev /etc ~
vivek@vivek-X556UQK:~/Rocket$ dirs -c
vivek@vivek-X556UQK:~/Rocket$ dirs
~/Rocket
```

دستور vdir

دستور **vdir** در لینوکس یکی از دستوراتی است که برای نمایش اطلاعات فایل‌ها با فرمت‌های مختلف مورد استفاده قرار می‌گیرد. این دستور به صورت مشابه دستور **ls** عمل می‌کند، اما اطلاعات بیشتری از فایل‌ها را نمایش می‌دهد.

گزینه‌های دستور

- **l**-نمایش جزئیات کامل فایل‌ها
- **a**-نمایش تمام فایل‌ها، حتی فایل‌های مخفی
- **S**-مرتب سازی فایل‌ها بر اساس اندازه
- **t**-مرتب سازی فایل‌ها بر اساس زمان اصلاح
- **r**-نمایش فایل‌ها به صورت معکوس
- **h**-نمایش اندازه فایل‌ها به صورت خوانا
- U-نمایش تاریخ اصلی فایل‌ها
- R-نمایش فایل‌ها به صورت بازگشتی
- g-نمایش فایل‌ها بدون نام گروه
- G-نمایش فایل‌ها بدون نام کاربر
- i-نمایش شماره inode فایل‌ها
- F-نمایش نوع فایل‌ها (برچسب)
- Q-نمایش نام فایل‌ها در دو قرینه
- k-نمایش اندازه فایل‌ها به کیلوبایت
- Z-نمایش اطلاعات امنیتی SELinux
- c-نمایش تاریخ ایجاد فایل‌ها
- d-نمایش اطلاعات فهرست‌ها به جای فایل‌ها

مثال های دستور:

نمایش فایل‌ها به همراه جزئیات کامل:

```
vdir -l
```

نمایش تمام فایل‌ها، حتی فایل‌های مخفی:

```
vdir -a
```

مرتب سازی فایل‌ها بر اساس اندازه:

```
vdir -S
```

مرتب سازی فایل‌ها بر اساس زمان اصلاح:

```
vdir -t
```

نمایش فایل‌ها به صورت معکوس:

```
vdir -r
```

نمایش اندازه فایل‌ها به صورت خوانا:

```
vdir –h
```

نمایش نام فایل‌ها در دو قرینه:

```
vdir -Q
```

نمایش اندازه فایل‌ها به کیلوبایت:

```
vdir -k
```

نمایش اطلاعات امنیتی SELinux:

```
vdir -Z
```

نمایش تاریخ ایجاد فایل‌ها:

```
vdir -c
```

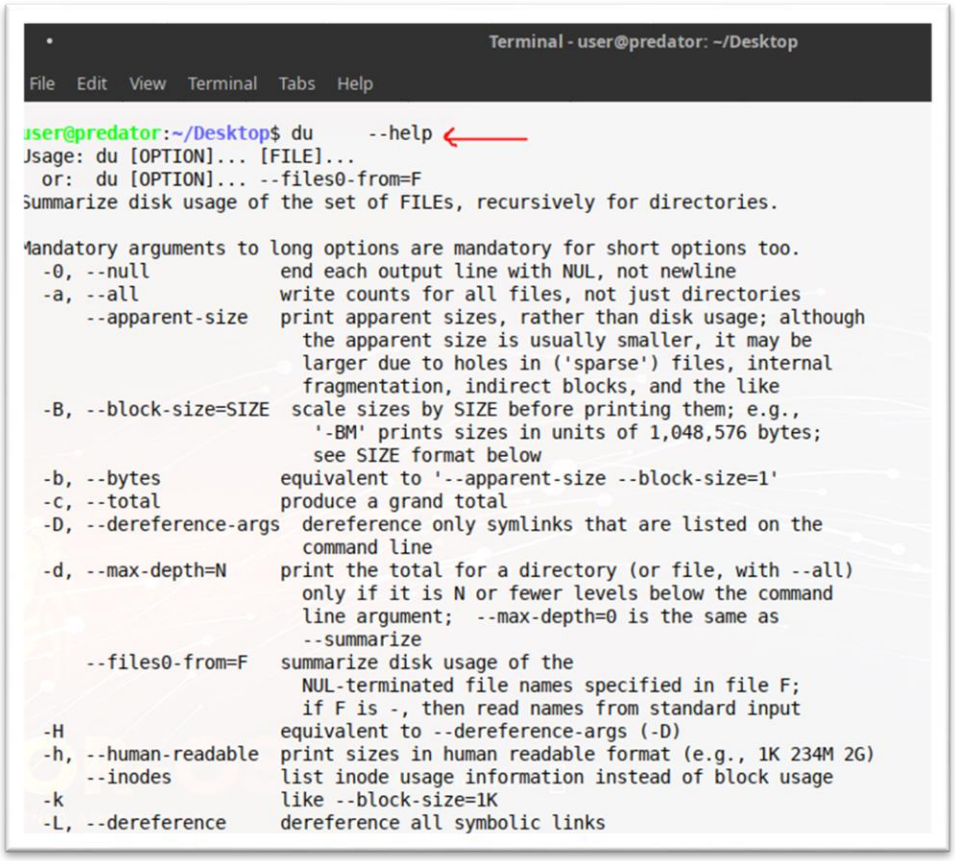
نمایش اطلاعات فهرست‌ها به جای فایل‌ها:

```
vdir -d
```

مشاهده راهنمای دستورات

سویچ h – یا help – – پس از هر دستور را یک منو راهنما برای دستور باز می کند:

```
du --help
```



دستور apropos

دستور apropos برای جستجو و نمایش توضیحات دستور استفاده می شود. دستور apropos در سیستم‌های Unix/Linux استفاده می‌شود تا به کاربران کمک کند تا دستورات مرتبط با یک موضوع خاص را پیدا کنند. این دستور به کاربر امکان می‌دهد که بر اساس یک کلمه کلیدی (keyword)، نام یک دستور، توضیحات یک دستور یا موضوع مورد نظر خود را جستجو کند.

ساختار دستور:

apropos keyword ..

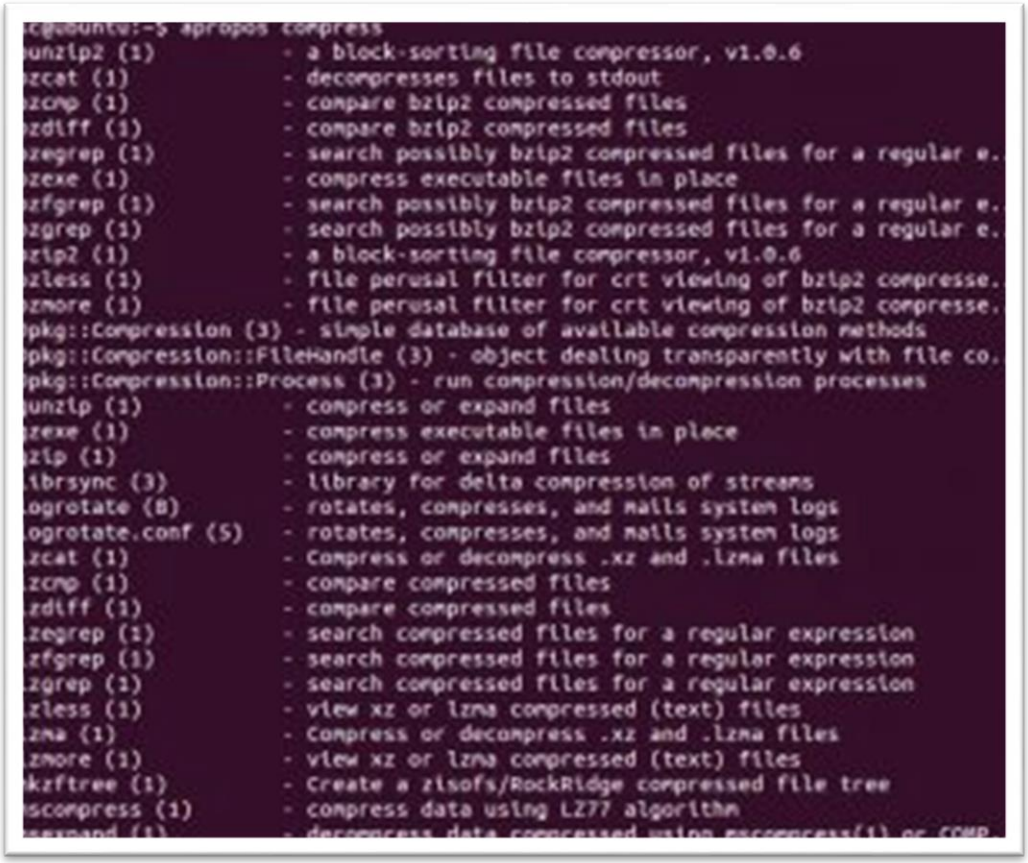
مثال ۱ : فرض کنید شما نمی دانید که چگونه یک فایل را فشرده کنید ، پس می توانید دستور زیر را در ترمینال تایپ کنید و همه دستورات مربوطه و شرح مختصر یا عملکرد آن را نشان می دهد.

ورودی :



پس از اجرای دستور فوق ، مشاهده خواهید کرد که یک دسته از دستورات در ترمینال لیست شده است که نه تنها به نحوه فشرده سازی یک فایل بلکه همچنین به منظور گسترش یک فایل فشرده ، جستجو در یک فایل فشرده ، مقایسه یک فایل فشرده و غیره می پردازد.

خروجی دستور:

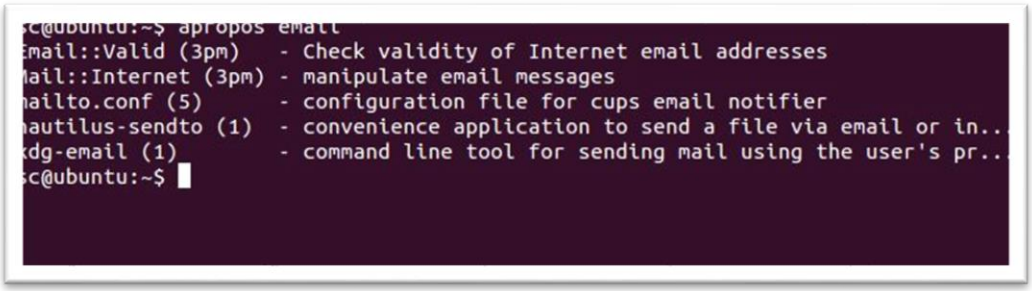


مثال ۲ :اگر به عنوان آرگومان ارائه شود ، *apropos* از چندین کلمه کلیدی پشتیبانی می کند ، همچنین می توانیم بیش از یک کلمه کلیدی را برای جستجوی بهتر ارائه دهیم .بنابراین ، اگر دو کلمه کلیدی ارائه شود ، دستور apropos تمام لیست دستورات را نمایش می دهد که شامل اولین کلمه کلیدی در توصیف صفحه اصلی آن یا کلمه کلیدی دوم است.

- **ورودی ۱ با یک کلمه کلیدی**

```
sc@ubuntu:~$ apropos email
```

خروجی دستور:



- **ورودی ۲ با چندین کلمه کلیدی:**

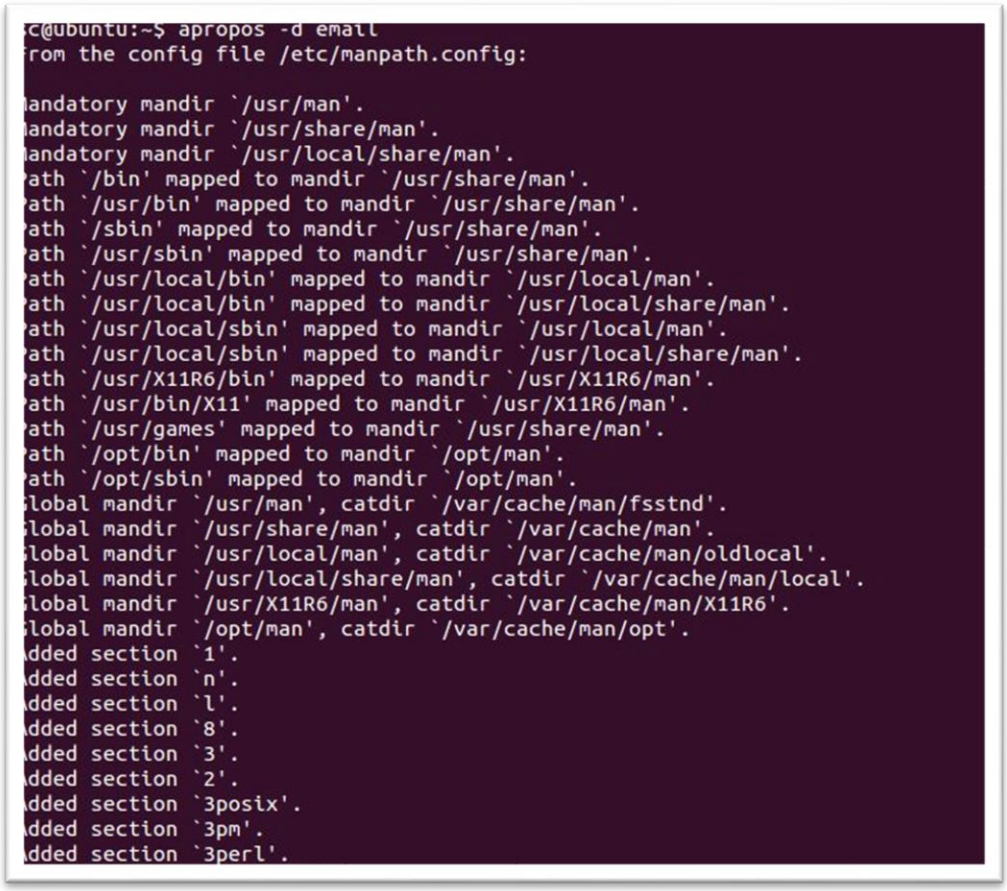
```
sc@ubuntu:~$ apropos email address
```

خروجی دستور:

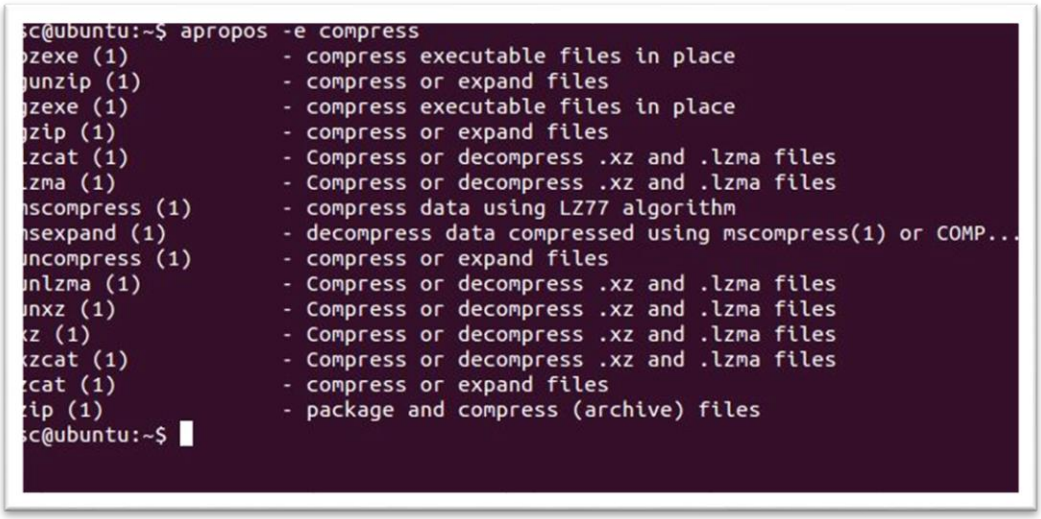
```
sc@ubuntu:~$ apropos email address
addr2line (1)        - convert addresses into file names and line numbers.
avahi-autoipd (8)    - IPv4LL network address configuration daemon
avahi-publish (1)    - Register an mDNS/DNS-SD service or host name or addres..
avahi-publish-address (1) - Register an mDNS/DNS-SD service or host name or a..
avahi-publish-service (1) - Register an mDNS/DNS-SD service or host name or a..
avahi-resolve (1)    - Resolve one or more mDNS/DNS host name(s) to IP addres..
avahi-resolve-address (1) - Resolve one or more mDNS/DNS host name(s) to IP a..
avahi-resolve-host-name (1) - Resolve one or more mDNS/DNS host name(s) to IP..
bridge (8)           - show / manipulate bridge addresses and devices
Email::Valid (3pm)   - Check validity of Internet email addresses
ether_aton (3)       - Ethernet address manipulation routines
ether_aton_r (3)     - Ethernet address manipulation routines
ether_hostton (3)    - Ethernet address manipulation routines
ether_line (3)       - Ethernet address manipulation routines
ether_ntoa (3)       - Ethernet address manipulation routines
ether_ntoa_r (3)     - Ethernet address manipulation routines
ether_ntohost (3)    - Ethernet address manipulation routines
ethers (5)           - Ethernet address to IP number database
freeaddrinfo (3)     - network address and service translation
freehostent (3)      - get network hostnames and addresses
freeifaddrs (3)      - get interface addresses
gai_cancel (3)       - asynchronous network address and service translation
gai_error (3)        - asynchronous network address and service translation
gai_strerror (3)     - network address and service translation
gai_suspend (3)      - asynchronous network address and service translation
get_myaddress (3)    - library routines for remote procedure calls
getaddrinfo (3)      - network address and service translation
getaddrinfo_a (3)    - asynchronous network address and service translation
gethostip (1)        - convert an IP address into various formats
getifaddrs (3)       - get interface addresses
getipnodebyaddr (3)  - get network hostnames and addresses
getipnodebyname (3)  - get network hostnames and addresses
getnameinfo (3)      - address-to-name translation in protocol-independent ma..
```

گزینه های دستور:

d- :از این گزینه برای انتشار پیام های رفع اشکال استفاده می شود .مخفف debug است.هنگامی که از این گزینه استفاده می شود ، ترمینال فهرست های man ، مسیر جهانی ، فهرست مسیر ، هشدارها و غیره هر دستور را که مربوط به کلمه کلیدی جستجو است برمی گرداند.



- **v-** : این گزینه برای چاپ پیام های هشدار دهنده پرکاربرد استفاده می شود.
- **e-**یا **exact–** از این گزینه برای جستجوی دقیق هر کلید واژه استفاده می شود .اگر از هیچ گزینه ای استفاده نشده باشد ، دستور apropos لیستی از تمام دستوراتی را که توضیحات آن در توضیحات صفحه man با کلمه کلیدی مطابقت دارد یا به نوعی با کلمه کلیدی ذکر شده در استدلال مرتبط هستند ، برمی گرداند .با این حال ، هنگامی که از گزینه *e*- استفاده می شود ، apropos فقط فرمانی را برمی گرداند که توضیحات آن دقیقاً با کلمه کلیدی مطابقت داشته باشد.



- **w-**یا **wildcard–**این گزینه هنگامی استفاده می شود که کلمات کلیدی حاوی باشند . به طور مستقل نام صفحه و توضیحات مطابق با کلمه کلیدی (ها) را جستجو می کند.

```
sc@ubuntu:~$ apropos -w sudo
sudo (8)              - execute a command as another user
sudo.conf (5)         - configuration for sudo front end
sudo_plugin (8)       - Sudo Plugin API
sudoers (5)           - default sudo security policy plugin
sudoreplay (8)        - replay sudo session logs
sc@ubuntu:~$
```

وقتی sudo به عنوان wildcard داده می شود ، تمام دستورات مربوط به sudo ذکر می شوند.

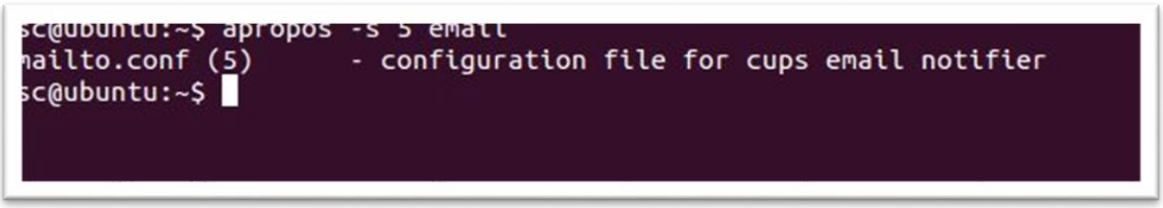
- **a**-یا **and**– این گزینه زمانی استفاده می شود که بخواهیم همه کلمات کلیدی با هم مطابقت داشته باشند .اگر هیچ یک از کلمات کلیدی ارائه شده در صفحه یا توضیحات انسان مطابقت نداشته باشد ، هیچ چیزی بر نمی گرداند .در ورودی زیر ، دو کلمه کلیدی داده شده است و فقط دو دستور در نتیجه نمایش داده می شود زیرا فقط یک دستور وجود دارد که هر دو کلمه کلیدی را شامل می شود.

```
sc@ubuntu:~$ apropos -a email send
nautilus-sendto (1)   - convenience application to send a file via email or in...
xdg-email (1)         - command line tool for sending mail using the user's pr...
sc@ubuntu:~$
```

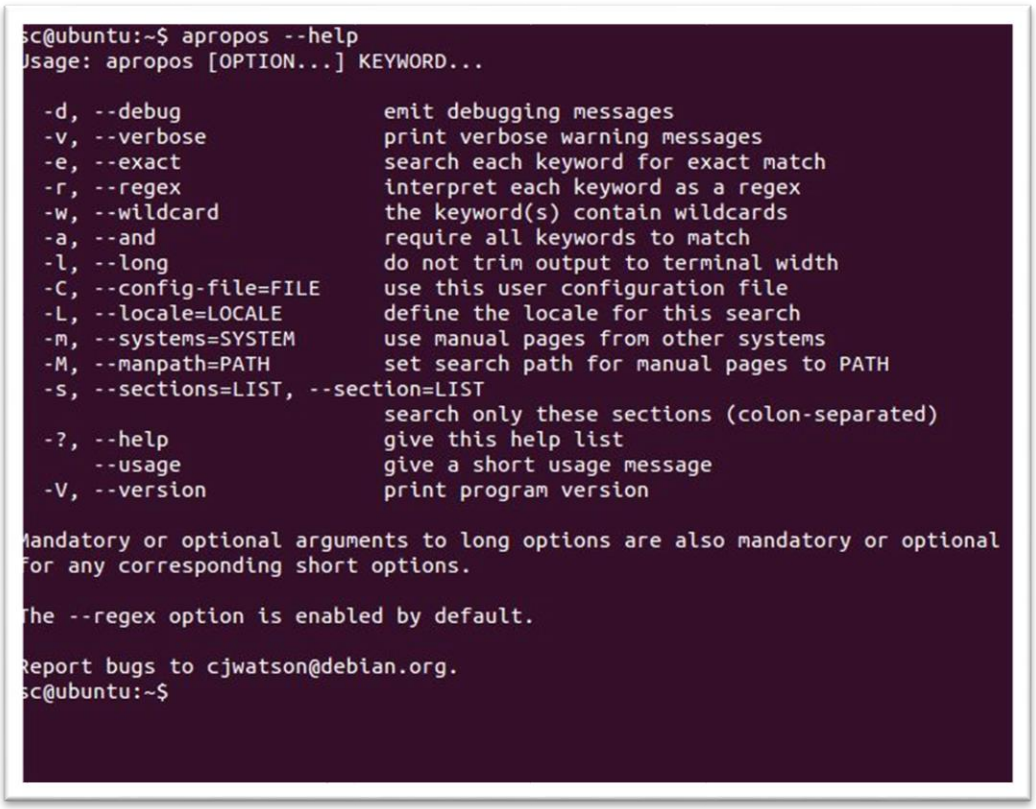
- **l**-یا **long**– به طور پیش فرض ، خروجی به عرض ترمینال کوتاه می شود .این گزینه زمانی مفید است که نمی خواهیم نتیجه کوتاه شود.

```
sc@ubuntu:~$ apropos -l email
Email::Valid (3pm)    - Check validity of Internet email addresses
Mail::Internet (3pm)  - manipulate email messages
mailto.conf (5)       - configuration file for cups email notifier
nautilus-sendto (1)   - convenience application to send a file via email or insta
nt messenger
xdg-email (1)         - command line tool for sending mail using the user's prefe
rred e-mail composer
sc@ubuntu:~$
```

- **C**- این گزینه زمانی استفاده می شود که ما نمی خواهیم از پرونده پیکربندی پیش فرض (/ manpath) استفاده کنیم.
- **L**- محلی را برای این جستجو مشخص کنید.
- **m**-یا **systems**– در این گزینه از صفحات دستی سیستمهای دیگر استفاده می شود .این گزینه زمانی مفید است که بخواهیم توضیحات صفحه man را برای سیستم عامل قابل دسترسی دیگر جستجو کنیم.
- **M**-یا **manpath**– مسیر جستجوی صفحات دستی را به جای *MANPATH* $ پیش فرض ، روی PATH تنظیم کنید.
- **s**-یا **sections**– یا **section**– این گزینه زمانی مورد استفاده قرار می گیرد که بخواهیم فقط بخشهای خاصی را جستجو کنیم (جدا شده از روده بزرگ) که در آرگومان آورده شده است.



- **-? یا help–** این گزینه لیست **راهنما** را نمایش می دهد.



- **V-یا version–** برای چاپ نسخه برنامه استفاده می شود.



- **r-یا regex–** این گزینه هر کلمه کلیدی را به عنوان(regex عبارت منظم) تفسیر می کند .کلمه کلیدی به طور مستقل با نام صفحه و توضیحات مطابقت دارد.

یافتن یک دستور باینری

دانستن جزئیات دستورات linux نه تنها به کاربر linux کمک می کند تا دستورات متعدد را مدیریت کند، بلکه کاربر را نیز قادر می‌سازد که عملیات مربوط به system را از آنها برای استفاده از خط دستور یا یک اسکریپت استفاده کند . برای مشاهده دستورات جدید بر روی system ، به تمام فهرست ها در متغیر محیطی PATHنگاه کنید.این فهرست تمام دستورات / برنامه های install شده در system

را ذخیره می کند. هنگامی که یک نام دستور جالب پیدا کردید، قبل از اینکه به خواندن بیشتر در مورد آن احتمالا در صفحه man[۱] بپردازید، سعی کنید اطلاعات جزئی درباره آن را به صورت زیر جمع آوری کنید.

دستور whatis

اکنون توضیحات و مکان دستور ها را با استفاده از روشهای مختلف در linux پیدا کنید.

```
whatis
```

whatis برای نمایش توضیحات استفاده می شود که به عنوان یک argument وارد می شود.

اگر توضیحات بیش از حد طولانی[۲] باشد برخی از خروجی به طور پیش فرض مرتب شده اند، از سویچ l-برای نشان دادن شرح کامل استفاده کنید.

```
$ whatis top
$ whatis -l top
```

دستور compgen

- compgen -c تمام دستوراتی را که می توانید اجرا کنید فهرست می کند.
- compgen -a تمام نام‌های مستعار را که می‌توانید اجرا کنید فهرست می‌کند.
- compgen -b تمام داخلی‌هایی را که می‌توانید اجرا کنید فهرست می‌کند.
- compgen -k تمام کلمات کلیدی را که می توانید اجرا کنید فهرست می کند.
- compgen -f یک تابع تمام توابعی را که می توانید اجرا کنید فهرست می کند.
- compgen -A function -abck همه موارد فوق را یکجا فهرست می کند.

```
compgen -ac | grep searchstr
```

دستور TLDR

یکی از روشهای رایج ترین و قابل اطمینان برای دریافت راهنما در سیستم های یونیکس از طریق صفحه های شخصی است .صفحات Manمستندات استاندارد برای هر یونیکس مانند سیستم هستند و آنها را به کتابچه های آنلاین برای برنامه ها، توابع، کتابخانه ها، تماس های سیستم، استانداردهای رسمی و کنوانسیون ها، فرمت های فایل و غیره مطابقت دارد .

TLDRبه معنای Too Long; Didn't Read خلاصه ای از دستورات در سیستم عامل های مختلف از جمله لینوکس است . محتوای صفحات TLDR به طور آشکار تحت مجوز MIT مجاز است.

```
Install TLDR
$ sudo npm install -g tldr
$ sudo snap install tldr
$ sudo dnf install tldr
$ sudo apt install tldr
```

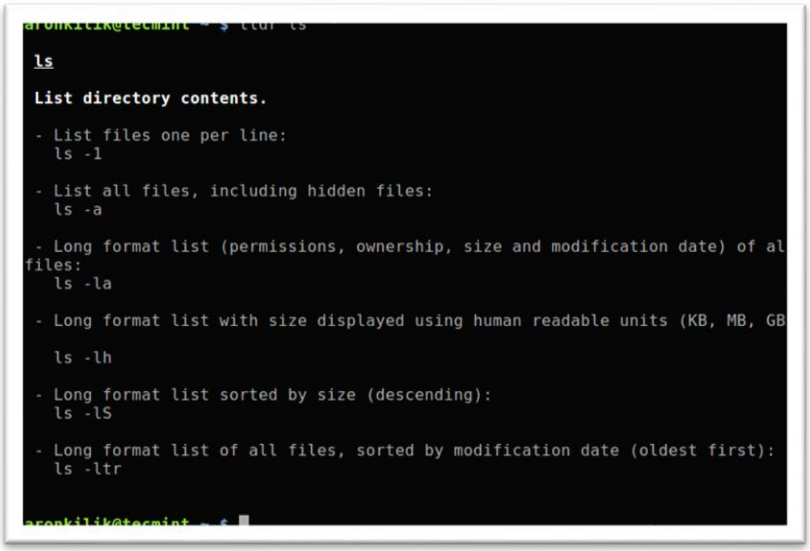
مثال:

```
$ tldr ls
```

---

[۱] manual

[۲] long format



دستور tldr یک ابزار خط فرمان است که برای نمایش خلاصه‌های ساده و قابل فهم از دستورات خط فرمان استفاده می‌شود. نام tldr مخفف عبارت " "Too Long; Didn't Readاست و هدف آن ارائه توضیح مختصر و ساده از دستورات خط فرمان است تا کاربران بتوانند به سرعت و با سادگی درک کنند چگونه از هر دستور استفاده کنند. با اجرای دستور tldrبه همراه نام دستور خط فرمانی که می‌خواهید راهنمایی درباره آن را ببینید، می‌توانید توضیحات خلاصه و مثال‌های کاربردی را در مورد آن دستور ببینید. این توضیحات معمولاً به صورت مختصر، ساده و قابل درک برای کاربران عادی ارائه می‌شود. برای استفاده از tldr، باید ابتدا آن را در سیستم خود نصب کنید. این ابزار در برخی توزیع‌های لینوکس و سیستم‌عامل macOS وجود دارد و می‌توانید آن را از طریق یک مدیر بسته نصب کنید. بعد از نصب، می‌توانید tldrرا با استفاده از دستور زیر فراخوانی کنید:

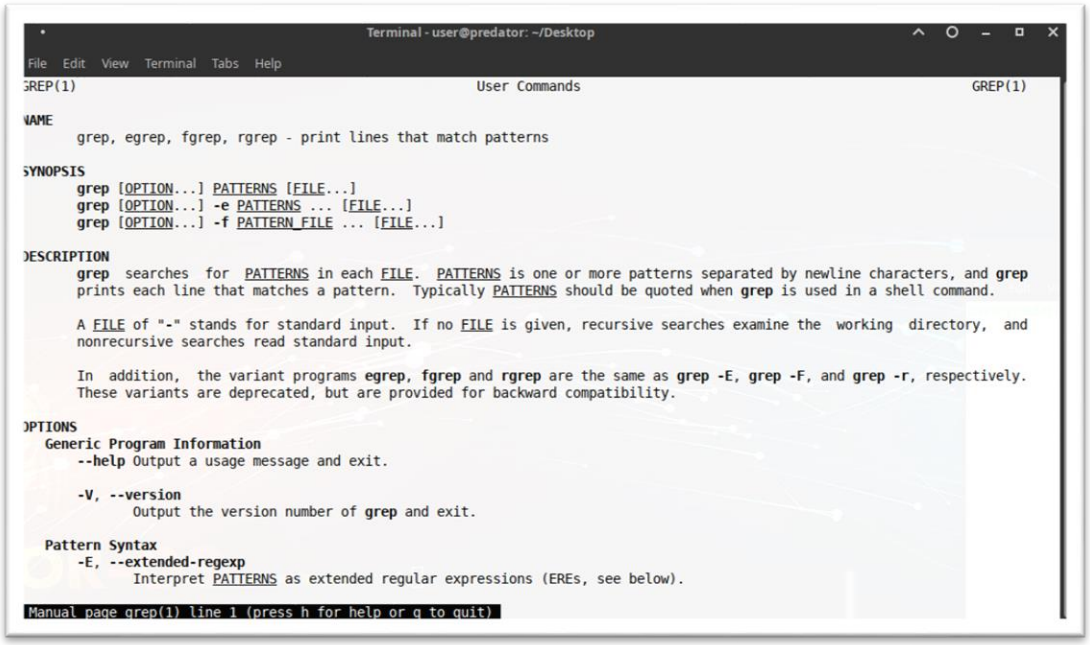
نام دستور tldr

بعد از اجرای این دستور، خلاصه و مثال‌هایی از استفاده صحیح دستور مورد نظر را خواهید دید. این راهنما می‌کند تا به سرعت از دستورات خط فرمان استفاده کنید و به سادگی و با فهم بیشتر با آنها کار کنید.

دستور man

دستور man قبل از هر دستور manual آن را نشان می دهد. man یک اصطلاح کوتاه برای صفحه دستی است .در سیستم عامل های یونیکس مانند لینوکس، manرابطی برای مشاهده راهنمای مرجع سیستم است. کاربر می تواند با تایپ man و سپس یک فاصله و سپس آرگومان، درخواست نمایش یک صفحه man را بدهد .در اینجا آرگومان آن می تواند یک دستور، ابزار یا تابع باشد .یک صفحه دستی مرتبط با هر یک از این آرگومان ها نمایش داده می شود. اگر شماره بخش را در دستور ارائه کنید، به man هدایت می‌شود تا شماره بخش راهنما را بررسی کند و صفحه آن بخش نمایش داده می‌شود .و اگر نه، به طور پیش فرض صفحه اول را نمایش می دهد و باید تمام بخش ها را به صورت از پیش تعریف شده مرور کنید. manیک کتابچه راهنمای داخلی برای استفاده از دستورات لینوکس است .به کاربران اجازه می دهد تا به راهنمای مرجع یک ابزار یا دستوری برای اجرا در ترمینال دسترسی داشته باشند .دستور man حاوی توضیحات فرمان، مثال‌ها، پرچم‌ها، گزینه‌های قابل اجرا و سایر عناصر اطلاعاتی است.

man grep



man ls

این دستور تمام اطلاعات مربوط به فرمان 'ls' را همانطور که در تصویر نشان داده شده است نمایش می دهد. به‌طور پیش‌فرض، فرمان manهر جزء موجود کتابچه راهنمای کاربر را جستجو می‌کند و مطابقت اولیه را نمایش می‌دهد (حتی زمانی که صفحه در بسیاری از بخش‌ها در دسترس است) .با دادن شماره بخش، دستور man برای جستجو در یک بخش خاص را نشان می دهد.

و بخش دستی به شرح زیر فهرست شده و توضیح داده شده است:

دستورات عمومی :دستوراتی را که در ترمینال استفاده می شود را نشان می دهد.

توابع کتابخانه :توابع موجود در کتابخانه های برنامه را نشان می دهد.

فراخوانی سیستم :اینها توابع ارائه شده توسط هسته را نشان می دهند.

فایل های ویژه :دستگاه ها معمولا در فهرست / dev و درایورهای مربوطه یافت می شوند.

بازی ها :توضیحات دستوری را نشان می دهد که نقل قول های پایگاه داده را نشان می دهد.

فرمت ها و قراردادهای فایل: فرمت های فایل مانند etc/passwd را نشان می دهد.

متفرقه :توضیحات مختلف، مانند پارامترهای بوت، قراردادها و بسته‌های ماکرو و موارد دیگر.

روال کرنل :اطلاعات مربوط به عملیات داخلی هسته را نمایش می دهد.

دستورات مدیریت سیستم :اکثر دستورات برای root محفوظ هستند.

چگونه از دستور man در لینوکس استفاده کنیم؟

دستور man را تایپ کنید و به دنبال آن نام فرمان لینوکس که مایلیم صفحه man را در صفحه ترمینال ببینیم، وارد کنید.

نتیجه طولانی است .ما می توانیم از چرخ اسکرول ماوس، کلیدهای بالا و پایین یا کلیدهای PgUp و PgDn برای کار از طریق آن استفاده کنیم. می‌توانیم روی دکمه H کلیک کنیم تا بخش راهنما و یک جدول کلید احتمالی را پیدا کنیم تا پس از اجرای دستور man در خروجی حرکت کنیم.

برای خروج می توانیم دکمه Q را فشار دهیم.

خروجی فرمان عنوان موجود صفحه man را برای دستور نامگذاری شده نشان می دهد.

در زیر لیست سرفصل ها وجود دارد:

Name : نام فرمان را نشان می دهد.

خلاصه :نحو دستور را نشان می دهد.

مثال‌ها :در این عنوان، مثال‌های زیادی استفاده از دستور را نشان می‌دهند.

پیکربندی :اطلاعات پیکربندی یک دستگاه را نشان می دهد.

پیش فرض ها :آنها عملکرد پیش فرض فرمان و نحوه کنترل آنها را نشان می دهند.

گزینه های دستور: آنها لیستی از پرچم ها و گزینه هایی را که دستور می گیرد نشان می دهند.

وضعیت خروج :فهرستی از مقادیر وضعیت خروج ممکن برای یک فرمان را نشان می دهد.

فایل ها :آنها یک لیست فایل را نشان می دهند که توسط دستور استفاده می شود.

Environment: شرح و لیستی از متغیرهای محیطی که بر دستور تأثیر می گذارند را نمایش می دهد.

همچنین ببینید :این دستورات مربوط به موضوع مشخص شده را نشان می دهد.

تاریخچه :تاریخچه توسعه فرمان را نشان می دهد.

نویسندگان :این عنوان شخصی را نشان می دهد که دستور را نوشته و مدیریت کرده است.

یادداشت ها :این عنوان یادداشت های بسیاری از جمله وابستگی ها، مجوزهای مورد نیاز و غیره را نشان می دهد.

اشکالات :این عنوان مشکلات شناخته شده در نسخه برنامه را نشان می دهد.

مهم :صفحات ممکن است شامل تعداد کمتر یا بیشتر عنوان با تکیه بر محتوای صفحه مرد باشند.

نحوه نمایش خروجی

manخروجی خود را از طریق پیجر نمایش می دهد .پیجر برنامه‌ای است که خروجی‌های خود را در یک زمان پر از صفحه نمایش می‌دهد، به این معنی که کل متن به یکباره ظاهر نمی‌شود و گزینه‌ای برای اسکرول کردن صفحه وجود ندارد.

دو نقطه در پایین انتهای صفحه روی صفحه را نشان می دهد .برای رفتن به صفحه بعدی می توانید از space bar یا f و برای رفتن به صفحه عقب می توانید از 'b' استفاده کنید .برای خروج از صفحه روی صفحه از 'q' استفاده کنید و به برنامه پوسته هدایت خواهید شد .و برای راهنما، 'h'را فشار دهید.

بخش ها در صفحه man

صفحه man به بخش های مختلفی تقسیم می شود .هر بخش بر اساس موضوع خاصی تقسیم شده است.

صفحات man دارای یک عدد در داخل پرانتز بعد از دستور هستند .این اعداد نشان دهنده شماره بخش است .در تصویر بالا، LS(1) را در بالا مشاهده می کنید که نشان می دهد از بخش ۱ است .همانطور که قبلاً گفتم، اگر شماره بخش خاصی را ذکر کنید، صفحه مرد شما به آن بخش هدایت می شود .در غیر این صورت باید کل بخش ها را به صورت از پیش تعریف شده طی کنید تا به قسمت مورد نظر خود بروید.

بخش ها:

- برنامه های اجرایی و دستورات پوسته
- تماس های سیستمی
- تماس های کتابخانه
- فایل های خاص
- فرمت ها و قراردادهای فایل

- بازی ها
- متفرقه
- دستورات مدیریت سیستم
- روتین های هسته
- یک زبان برنامه نویسی Tcl/Tk (

گزینه های دستور

- man –aw: نمایش تمام بخش‌های موجود یک دستور.
- man –a : نمایش همه صفحات man یک دستور.
- man -k (apropos) نمایش لیستی از نتایج در صفحه man که شامل یک کلمه کلیدی هستند.
- man -f, whatis: نمایش توضیحات از صفحه دستوری اگر موجود باشد.
- man whereis: استفاده برای تعیین مکان یک صفحه man.
- بدون گزینه: نمایش کامل راهنمای دستور.
- man Section-num: نمایش فقط یک بخش خاص از راهنما.
- man –w: نمایش مکانی که صفحه man یک دستور مشخص در آن قرار دارد.
- man –I: اجرای دستور به صورت حساس به بزرگی/کوچکی حروف.
- man --default, -D: بازنشانی رفتار دستور man به حالت پیش‌فرض.
- man --debug, -d: چاپ جزئیات اشکال‌زدایی.
- man --config-file=file, -C file: استفاده از فایل برای پیکربندی کاربر.
- man --warnings[=warnings]: فعال‌سازی هشدارها با groff.
- man --global-apropos, -K: جستجوی خشن در هر صفحه man.
- man --local-file, -l: نمایش فایل‌های راهنمای محلی.
- man --location, --path, --where, -w: نشان دادن مکان فایل nroff منبع.

مثال های دستور:

man -aw ls

نمایش تمام بخش‌های موجود برای دستور ls.

man -k search

نمایش لیستی از نتایج مرتبط با کلمه کلیدی "search".

man -f grep

نمایش توضیحات در مورد دستور grep.

man whereis tar :

تعیین مکان دستور tar.

man ls

نمایش کامل راهنمای دستور ls.

man 5 passwd

نمایش بخش مربوط به دستور passwd از بخش ۵.

```
man -w printf
```

نمایش مکان فایل man برای دستور printf.

```
man -I grep
```

اجرای دستور grep با حساسیت به بزرگی/کوچکی حروف.

```
man --default ls
```

بازنشانی رفتار دستور ls به حالت پیش‌فرض.

```
man --debug cat
```

چاپ جزئیات اشکال‌زدایی برای دستور cat.

دستور ln

دستور lnدر سیستم عامل‌های Unix/Linux برای ایجاد لینک‌های سخت و نرم به فایل‌ها یا فهرست‌ها استفاده می‌شود. لینک‌ها به کاربر امکان می‌دهند تا به فایل یا فهرست‌های دیگر از طریق یک مسیر دیگر نیز دسترسی پیدا کنند.

ساختار دستور:

```
ln [OPTION]... [-T] TARGET LINK_NAME (1st form)
ln [OPTION]... TARGET (2nd form)
ln [OPTION]... TARGET... DIRECTORY (3rd form)
ln [OPTION]... -t DIRECTORY TARGET... (4th form)
```

گزینه‌های دستور:

- b- پیوند سخت (hard link)ایجاد می‌کند.
- s- پیوند نمادین (symbolic link)ایجاد می‌کند.
- f- اگر فایل مقصد وجود داشته باشد، آن را حذف و سپس پیوند جدید را ایجاد می‌کند.
- i- قبل از ایجاد پیوند، از کاربر تأیید می‌گیرد.
- n- پیوند را بدون ایجاد آن، فقط شبیه‌سازی می‌کند.
- v- اطلاعات مربوط به پیوندهای ایجاد شده را نمایش می‌دهد.

مثال‌های دستور:

۱. ایجاد لینک سخت:

این دستور برای ایجاد یک لینک سخت به یک فایل یا فهرست استفاده می‌شود.

```
ln file1.txt link1
```

۲. ایجاد لینک نرم:

این دستور برای ایجاد یک لینک نرم به یک فایل یا فهرست استفاده می‌شود.

```
ln -s file2.txt link2
```

۳. لینک‌سازی روی فهرست:

این دستور برای ایجاد لینک به یک فهرست استفاده می‌شود.

```
ln -s /path/to/directory link-to-directory
```

٤. تغییر نام لینک:

این دستور برای تغییر نام یک لینک استفاده می‌شود.

```
ln -f oldlink newlink
```

۵. لینک‌سازی با استفاده از مسیر مطلق:

این دستور برای ایجاد لینک با استفاده از مسیر مطلق به جای مسیر نسبی استفاده می‌شود.

```
ln -s /absolute/path/to/file link3
```

٦. ایجاد لینک سخت به صورت بازگشتی:

این دستور برای ایجاد لینک سخت به صورت بازگشتی برای یک فهرست استفاده می‌شود.

```
ln -r /path/to/directory link-to-directory
```

دستور readlink

دستور readlink در سیستم عامل‌های یونیکس و لینوکس برای خواندن محتوای یک لینک نمادین symbolic link استفاده می‌شود.

این دستور به کاربر اجازه می‌دهد تا مسیر واقعی فایل یا فهرست مورد اشاره توسط لینک را بدست آورد.

گزینه های دستور:

- f– مسیر واقعی فایل یا دایرکتوری مقصد را نمایش می‌دهد، حتی اگر پیوندهای نمادین در مسیر وجود داشته باشند.
- e– پیوندهای نمادین را دنبال می‌کند و مسیر واقعی فایل یا دایرکتوری مقصد را نمایش می‌دهد.
- n– تعداد پیوندهای نمادین را در مسیر نمایش می‌دهد.
- q– اگر مسیر یک پیوند نمادین باشد، هیچ خروجی‌ای نمایش نمی‌دهد.
- v-اطلاعات مربوط به پیوند را نمایش می‌دهد.

مثال های دستور:

f- نمایش مسیر مطلق فایل یا فهرست هدف لینک.

```
readlink -f symbolic_link
```

خروجی دستور: /home/user/example_file.txt

e- نمایش مسیر مطلق فایل یا فهرست هدف لینک و اگر لینک وجود نداشته باشد، خطایی نشان داده می‌شود.

```
readlink -e symbolic_link
```

خروجی دستور: /var/www/example_directory

n-نمایش مسیر نسبی فایل یا فهرست هدف لینک.

```
readlink -n symbolic_link
```

خروجی دستور: ../documents/example.doc

m-نمایش مسیر مطلق فایل یا فهرست هدف لینک، حتی اگر لینک به یک لینک دیگر اشاره کند.

```
readlink -m symbolic_link
```

خروجی دستور: /home/user/another_directory/file.txt

q-حذف خطاهایی که ممکن است به خاطر عدم وجود لینک یا فایل‌های هدف ایجاد شوند.

```
readlink -q symbolic_link
```

خروجی دستور: /var/log/example.log/

s- نمایش محتوای لینک نمادین.

```
readlink -s symbolic_link
```

خروجی دستور: target_file.txt

c- نمایش مسیر مطلق فایل یا فهرست هدف لینک، حتی اگر لینک به یک لینک دیگر اشاره کند و لینک نمادین نیز وجود داشته باشد.

```
readlink -c symbolic_link
```

خروجی دستور: /var/www/public_html/index.html/

v- نمایش اطلاعات تکمیلی و اورورهای مربوط به عملیات readlink.

```
readlink -v symbolic_link
```

خروجی دستور: symbolic_link: symbolic link to target_file.txt

دستور touch

از این دستور برای ایجاد یک فایل به صورت خالی یا با پسوند مشخص استفاده میشود .در لینوکس هر فایل منفرد با مُهرهای زمانی[1] مرتبط است و هر فایل اطلاعات آخرین زمان دسترسی، آخرین زمان تغییر و آخرین زمان تغییر را ذخیره می‌کند. بنابراین، هر زمان که فایل جدیدی ایجاد می‌کنیم، به فایل موجود دسترسی پیدا می‌کنیم یا آن را اصلاح می‌کنیم، مُهر زمانی آن فایل به‌طور خودکار به‌روزرسانی می‌شود. در این مقاله به چند نمونه کاربردی از دستورلینوکس touch خواهیم پرداخت. دستورلمسی یک برنامه استاندارد برای سیستم عامل های یونیکس/لینوکس است که برای ایجاد، تغییر و اصلاح مُهرهای زمانی یک فایل استفاده می شود. قبل از رفتن به نمونه‌های دستورلمسی، لطفاً گزینه‌های زیر را بررسی کنید.

```
touch [options]
```

- **a-**، فقط زمان دسترسی را تغییر دهید
- **c-**، اگر فایل وجود ندارد، آن را ایجاد نکنید
- **d-**، زمان دسترسی و اصلاح را به روز کنید
- **m-**، فقط زمان اصلاح را تغییر دهید
- **r-**، از زمان دسترسی و اصلاح فایل استفاده کنید
- **t-**، یک فایل را با استفاده از یک زمان مشخص ایجاد می کند.

چگونه یک فایل خالی بسازیم؟

دستور touch زیر یک فایل خالی (صفر بایت) جدید به نام ali ایجاد می کند .

```
# touch ali
```

[1] Time stamp

نحوه ایجاد چندین فایل

با استفاده از دستور touch، می توانید بیش از یک فایل را نیز ایجاد کنید .برای مثال دستور زیر ۳ فایل به نام‌های ali، reza و hoda را ایجاد می‌کند .

```
# touch hoda reza ali
```

نحوه تغییر زمان دسترسی و تغییر فایل

برای تغییر یا به روز رسانی آخرین زمان دسترسی و اصلاح فایلی به نام ali از گزینه a – به صورت زیر استفاده کنید. دستور زیر زمان و تاریخ فعلی یک فایل را تنظیم می کند. اگر فایل ali وجود نداشته باشد، فایل خالی جدید را با نام ایجاد می کند.

```
# touch -a ali
```

چگونه از ایجاد فایل جدید جلوگیری کنیم

استفاده از گزینه c – با دستور touch از ایجاد فایل های جدید جلوگیری می کند. برای مثال دستور زیر اگر فایلی به نام ail وجود نداشته باشد، ایجاد نخواهد کرد.

```
touch -c ali
```

نحوه تغییر زمان اصلاح فایل

اگر دوست دارید تنها زمان تغییر فایلی به نام ali را تغییر دهید، از گزینه m – با دستور touch استفاده کنید. لطفاً توجه داشته باشید که فقط آخرین زمان‌های اصلاح (نه زمان دسترسی) فایل را به‌روزرسانی می‌کند.

```
# touch -m ali
```

زمان دسترسی و اصلاح را به صراحت تنظیم کنید

شما می توانید به صراحت زمان را با استفاده از گزینه c – و t – با دستور touch تنظیم کنید. قالب به صورت زیر خواهد بود.

```
# touch -c -t YYDDHHMM ali
```

به عنوان مثال دستور زیر تاریخ و زمان دسترسی و تغییر را برای یک فایل ali به عنوان ۱۷:۳۰ (۱۷:۳۰ بعد از ظهر) ۱۰ دسامبر سال جاری (۲۰۲۳) تنظیم می کند.

```
# touch -c -t 20231730 ali
```

نحوه استفاده از مهر زمانی timestamp یک فایل دیگر

دستور touch زیر با گزینه r – ، مهر زمان فایل ali را با مهر زمان فایل reza به روز می کند. بنابراین، هر دو فایل دارای مهر زمانی یکسانی هستند.

```
# touch -r reza ali
```

سایر روش های ایجاد فایل |

با دستور cat

```
cat > test.txt
```

با استفاده از redirect (>)

```
> test5.txt
```

با استفاده از دستور echo

```
echo ” File content” > test6.txt
```

با استفاده از دستور printf

```
printf ” File content” > test7.txt
```

دستور truncate

دستور truncate در سیستم‌عامل‌های Unix/Linux برای تغییر اندازه یک فایل به طور مستقیم بدون نیاز به باز کردن و ویرایش فایل استفاده می‌شود. این دستور به کاربر این امکان را می‌دهد تا اندازه یک فایل را تغییر دهد و فایل را به صورت خالی یا با اندازه مشخص شده پر کند.

گزینه های دستور:

- c– فقط تعداد مشخصی از بایت‌های ابتدای فایل را حذف می‌کند.
- s– اندازه فایل را به مقدار مشخصی تغییر می‌دهد.
- r– به صورت بازگشتی، محتویات تمام زیرشاخه‌های یک دایرکتوری را حذف می‌کند.
- v– اطلاعات مربوط به عملیات truncate را نمایش می‌دهد.

مثال های دستور:

تغییر اندازه یک فایل به یک حجم خالی: این دستور برای تغییر اندازه یک فایل به حجم خالی (صفر بایت) استفاده می‌شود.

```
truncate -s 0 myfile.txt
```

تغییر اندازه یک فایل به حجم مشخص شده: این دستور برای تغییر اندازه یک فایل به یک حجم مشخص شده (بر حسب بایت) استفاده می‌شود.

```
truncate -s 1M myfile.txt
```

افزایش اندازه فایل با پر کردن با صفرها: این دستور برای افزایش اندازه یک فایل با پر کردن با صفرها به اندازه مشخص شده استفاده می‌شود.

```
truncate -s +100K myfile.txt
```

افزایش اندازه فایل با پر کردن با مقدار خاص: این دستور برای افزایش اندازه یک فایل با پر کردن با مقدار خاص (مثلاً Aبه اندازه مشخص شده استفاده می‌شود.

```
truncate -s +1G --string=A myfile.txt
```

کوچک کردن اندازه فایل: این دستور برای کوچک کردن اندازه یک فایل به حجم مشخص شده استفاده می‌شود.

```
truncate -s 100 myfile.txt
```

دستور rm

دستورrm برای حذف فایل ها یا فهرست ها به صورت زیر استفاده می شود.

```
$ rm file1
$ rm -rf my-directory
```

لطفا توجه داشته باشید که قبل از استفاده از این دستور، باید با دقت عمل کرده و از اطلاعات خود پشتیبانی تهیه کنید، زیرا حذف فایل‌ها و فهرست‌ها غیرقابل بازیابی است.

گزینه های دستور:

حذف یک فایل با نام : file.txt

```
rm file.txt
```

حذف چندین فایل با نام‌های file1.txt ، file2.txt و : file3.txt

```
rm file1.txt file2.txt file3.txt
```

حذف یک فهرست به نام directory و تمامی محتویات آن:

```
rm -r directory
```

حذف فایل‌ها بدون نیاز به تأیید کاربر:

```
rm -f file.txt
```

حذف فهرست‌ها و فایل‌ها با نمایش پیام راهنما قبل از حذف هر فایل یا فهرست:

```
rm -i file.txt
```

حذف فهرست‌ها و فایل‌ها به صورت صامت (بدون نمایش پیام خطا):

```
rm -f -silent file.txt
```

حذف فهرست‌ها و فایل‌ها با استفاده از الگوهای خاص:

```
rm *.txt
```

این دستور تمام فایل‌های با پسوند .txt را در فهرست فعلی حذف می‌کند.

حذف یک فهرست و تمامی محتویات آن در یک مسیر خاص:

```
rm -r /path/to/directory
```

در این مثال، فهرست directory در مسیر /path/to/ و تمامی محتویاتش حذف می‌شود.

دستور mkdir

از دستور mkdirبرای ایجاد فهرست‌ها با تنظیمات مختلف و عملکردهای مختلف استفاده کنید.حالا بیایید با یک مثال ساده به شما نشان دهم که چگونه از دستور mkdirبه همراه این گزینه‌ها استفاده کنید:

۱. برای استفاده از گزینه mیا mode=MODE برای تنظیم حالت فایل (سطوح دسترسی) می‌توانید از دستور زیر استفاده کنید:

```
mkdir -m 755 my_directory
```

این دستور یک فهرست به نام my_directoryبا سطوح دسترسی ۷۵۵ ایجاد می‌کند.

۲. برای استفاده از گزینه – pیا –– parentsبرای ایجاد فهرست‌های والدین به صورت خودکار می‌توانید از دستور زیر استفاده کنید:

```
mkdir -p parent_dir/child_dir
```

این دستور parent_dirرا ایجاد می‌کند و در داخل آن فهرست child_dirرا ایجاد می‌کند، حتی اگر parent_dirوجود داشته باشد.

۳. برای استفاده از گزینه – vیا –– verboseبرای چاپ پیام برای هر فهرست ایجاد شده، می‌توانید از دستور زیر استفاده کنید:

```
mkdir -v dir1 dir2 dir3
```

این دستور سه فهرست به نام dir1، dir2و dir3 ایجاد می‌کند و برای هرکدام پیامی چاپ می‌کند.

دستور mktemp

دستور mktemp در لینوکس برای ایجاد و نمایش یک فایل یا فهرست موقت با نام یکتا استفاده می‌شود. این دستور امکان ایجاد فایل‌ها یا فهرست‌های موقت با نام‌های منحصر به فرد و امن را فراهم می‌کند.

ساختار دستور:

```
mktemp [OPTION]... [TEMPLATE]
```

گزینه‌های دستور:

- -d یا --directory: ایجاد یک فهرست موقت به جای یک فایل.
- -u یا --dry-run: نمایش نام فایل یا فهرست موقت بدون ایجاد آن.
- -p template یا --tmpdir=template: تعیین مسیر موقت برای ایجاد فایل یا فهرست.

مثال‌های دستور:

ایجاد یک فایل موقت:

```
mktemp
```

ایجاد یک فهرست موقت:

```
mktemp -d
```

نمایش نام فایل موقت بدون ایجاد آن:

```
mktemp -u
```

تعیین مسیر موقت برای ایجاد فایل یا فهرست:

```
mktemp --tmpdir=/path/to/directory
```

استفاده از الگوی خاص برای ایجاد فایل یا فهرست موقت:

```
mktemp mytempfile_XXXXXX
```

دستور rmdir

دستور rmdir به حذف / حذف فهرست های خالی به صورت زیر کمک می کند.

```
$ rmdir / backup / all
```

دستور rmdir در سیستم عامل‌های مبتنی بر یونیکس (UNIX) و لینوکس برای حذف یک فهرست خالی استفاده می‌شود. این دستور تنها قادر به حذف فهرست‌هایی است که خالی هستند و دارای هیچ فایل یا فهرست دیگری نیستند. برای استفاده از دستور rmdir، شما باید نام فهرست را به عنوان ورودی به آن بدهید.

مثال‌های دستور:

حذف یک فهرست خالی به نام : directory

```
rmdir directory
```

حذف چندین فهرست خالی با نام‌های dir1، dir2 و : dir3

```
rmdir dir1 dir2 dir3
```

حذف یک فهرست خالی و نمایش پیام تأیید قبل از حذف:

```
rmdir -i directory
```

حذف یک فهرست خالی و نمایش پیام‌های خطا در صورت بروز مشکل:

```
rmdir -v directory
```

حذف چندین فهرست خالی در یک دستور:

```
rmdir dir1 dir2 dir3
```

توجه داشته باشید که دستور rmdir تنها برای حذف فهرست‌های خالی قابل استفاده است. در صورتی که فهرست دارای فایل‌ها یا فهرست‌های دیگر باشد، از دستور rm -r استفاده کنید تا فهرست و تمامی محتویات آن را حذف کنید.

دستور CP

در linux دستورcp برای کپی کردن فایل ها از یک فهرست به دیگری استفاده می شود، ساده ترین نحو استفاده از آن به شرح زیر است:

```
# cp    فایل یا فهرست- منبع    مقصد
```

دستور cpبرای کپی فایل ها و فهرست ها از یک مکان به مکان دیگر استفاده می شود.

```
$ cp   / home/ hosein /file1   /home/ hosein
```

همچنین می‌توانید از دستور پیشرفته کپی استفاده کنیدکه نوار پیشرفت[۱] را در هنگام کپی کردن پرونده ها / فهرست های بزرگ در linux نشان می دهد. سعی کنید از طریق صفحات man از دستورات cp، echo وxargs برای find اطلاعات مفید و پیشرفته استفاده کنید:

```
$ man cp
$ man echo
$ man xargs
```

دستورAdvacned-Copy

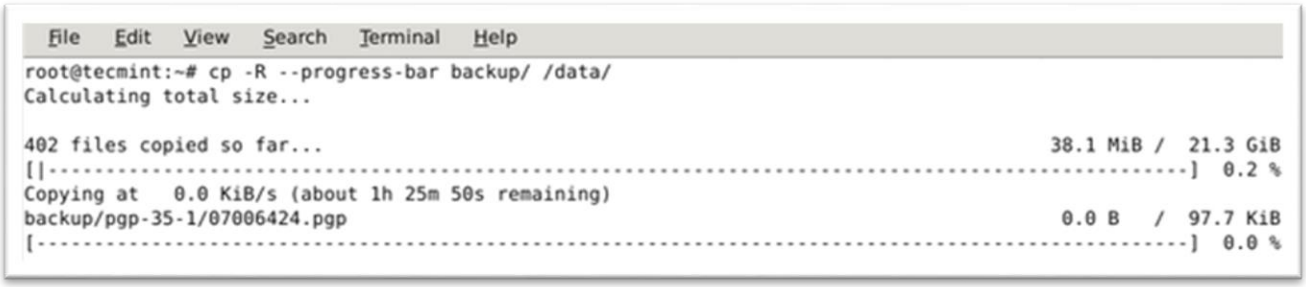
دستور همان cp است، ولی با اضافه کردن گزینه " g- " یا " progress-bar- " با دستور cpاست.
گزینه " R- " برای کپی مستقیم فهرست ها است.در اینجا نمونه ای از عکس های یک فرایند کپی با استفاده از دستور کپی پیشرفته است.

```
# cp -gR  /hosein .com/   / data /

یا

# cp -R -   /hosein /      / data /ali
```





[۱] process bar

دستور scp

دستورscp را قادر می سازد به طور مثال فایل های ایمن را در بین hostها در یک شبکه کپی کنید. دراین دستور حرف s مخفف secure میباشد.

```
$ scp ~/names.txt root@192.168.56.10 :/root/names.txt
```

دستور scpدر سیستم عامل‌های مبتنی بر یونیکس (UNIX) و لینوکس برای انتقال فایل‌ها و فهرست‌ها بین دو سیستم از طریق شبکه استفاده می‌شود scp. مخفف "Secure Copy" است و برای انتقال امن فایل‌ها از طریق پروتکل SSH (Secure Shell) استفاده می‌کند.

ساختار دستور:

```
scp [OPTIONS] SOURCE DESTINATION
```

در اینجا، SOURCEنشان‌دهنده مبدا فایل یا فهرست است و DESTINATIONنشان‌دهنده مقصد انتقال است.

برخی از استفاده‌های رایج scpعبارتند از:

انتقال یک فایل از سیستم محلی به یک سرور از طریق SSH:

```
scp file.txt user@server:/path/to/destination
```

انتقال یک فایل از یک سرور به سیستم محلی:

```
scp user@server:/path/to/file.txt /local/path/destination
```

انتقال یک فهرست از سیستم محلی به یک سرور:

```
scp -r directory user@server:/path/to/destination
```

انتقال یک فهرست از یک سرور به سیستم محلی:

```
scp -r user@server:/path/to/directory /local/path/destination
```

انتقال فایل‌ها و فهرست‌ها با استفاده از پورت خاص:

```
scp -P port file.txt user@server:/path/to/destination
```

انتقال چندین فایل با استفاده از نماد جایگزینی ویژه:

```
scp file*.txt user@server:/path/to/destination
```

انتقال فایل به دسته‌ای از سرورها:

```
scp file.txt user1@server1:/path/to/destination1 user2@server2:/path/to/destination2
```

این مثال‌ها تعدادی از رایج‌ترین استفاده‌ها از دستور scpرا نشان می‌دهند. با استفاده از این دستور و تنظیمات مختلف، می‌توانید فایل‌ها و فهرست‌ها را بین سیستم‌ها به صورت امن منتقل کنید.

دستور rsync

rsync احتمالاً یکی از پرکاربردترین دستورات است. آنجا. برای کپی امن فایل ها از یک سرور به سرور دیگر از طریق SSH استفاده می شود. در مقایسه با دستور scp که کار مشابهی را انجام می‌دهد، rsync انتقال را بسیار سریع‌تر می‌کند و در صورت وقفه، می‌توانید فرآیند انتقال را بازیابی/ازسرگیری کنید.

ساختار دستور:

```
Local:  rsync [OPTION...] SRC... [DEST]
Access via remote shell:
  Pull: rsync [OPTION...] [USER@]HOST:SRC... [DEST]
  Push: rsync [OPTION...] SRC... [USER@]HOST:DEST
```

```
Access via rsync daemon:
  Pull: rsync [OPTION...] [USER@]HOST::SRC... [DEST]
        rsync [OPTION...] rsync://[USER@]HOST[:PORT]/SRC... [DEST]
  Push: rsync [OPTION...] SRC... [USER@]HOST::DEST
        rsync [OPTION...] SRC... rsync://[USER@]HOST[:PORT]/DEST
```

```
$ rsync -avz user@your-remote-server.com:/home/user/dir/
```

دستور بالا تمام فایل ها و فهرست ها را از پوشه فعلی سرور شما به سرور راه دور کپی می‌کند.

گزینه های دستور:

- a– به طور پیش فرض، تمام مجوزها، مالکیت، زمان‌های دسترسی و پیوندها حفظ می‌شوند.
- b– از الگوریتم انتقال دلتا برای بهینه‌سازی فرآیند همگام‌سازی استفاده می‌کند.
- c– فقط در صورت بروز تغییر در فایل‌ها، آنها را همگام‌سازی می‌کند.
- e– از یک برنامه خاص برای انتقال داده‌ها استفاده می‌کند.
- f– از حذف فایل‌های اضافی در مقصد جلوگیری می‌کند.
- h– راهنمای دستور را نمایش می‌دهد.
- i– قبل از هر گونه تغییر، از کاربر تأیید می‌گیرد.
- n– فقط فرآیند همگام‌سازی را شبیه‌سازی می‌کند، بدون اینکه هیچ تغییری ایجاد کند.
- p– مجوزها، مالکیت و زمان‌های دسترسی را حفظ می‌کند.
- q– خروجی دستور را محدود می‌کند.
- r– دایرکتوری‌ها را به طور بازگشتی همگام‌سازی می‌کند.
- t– فقط فایل‌های متنی را همگام‌سازی می‌کند.
- u– فقط فایل‌هایی را که در مقصد جدیدتر هستند، همگام‌سازی می‌کند.
- v– جزئیات مربوط به فرآیند همگام‌سازی را نمایش می‌دهد.
- z– از الگوریتم فشرده‌سازی برای بهینه‌سازی فرآیند همگام‌سازی استفاده می‌کند.

اگر سرویس SSH روی سرور راه دور روی پورت استاندارد ۲۲ اجرا نمی شود ، می توانید از rsync با یک پورت SSH ویژه استفاده کنید:

```
$ rsync -avz user@your-remote server.com:/home/user/dir/
```

مثال های دستور:

مثال‌هایی برای گزینه‌های دستور rsync:

a:–

```
rsync -a /home/user/Documents /backup/Documents
```

این دستور تمام فایل‌ها و دایرکتوری‌های موجود در دایرکتوری Documents را به دایرکتوری backup/Documents همگام‌سازی می‌کند.

b:–

```
rsync -b /home/user/Desktop /remote/Desktop
```

این دستور فقط فایل‌هایی را که در دایرکتوری Desktop تغییر کرده‌اند، به دایرکتوری remote/Desktop همگام‌سازی می‌کند.

c:–

```
rsync -c /home/user/Pictures /archive/Pictures
```

این دستور فقط در صورتی که فایل‌ها در دایرکتوری Pictures جدیدتر باشند، آنها را به دایرکتوری archive/Pictures همگام‌سازی می‌کند.

e:–

```
rsync -e ssh user@remotehost:/home/user/Music /local/Music
```

این دستور از طریق SSH برای همگام‌سازی فایل‌ها از دایرکتوری home/user/Music/ در سیستم راه دور با نام remotehost به دایرکتوری local/Music/ در سیستم محلی استفاده می‌کند.

f:–

```
rsync -f /home/user/Videos /media/Videos
```

این دستور تمام فایل‌ها و دایرکتوری‌های موجود در دایرکتوری Videos را به دایرکتوری media/Videos همگام‌سازی می‌کند، حتی اگر فایل‌هایی در دایرکتوری مقصد وجود داشته باشند که در دایرکتوری منبع وجود ندارند.

h:–

```
rsync -h
```

این دستور راهنمای دستور rsync را نمایش می‌دهد.

i:–

```
rsync -i /etc/passwd /etc/passwd.bak
```

قبل از همگام‌سازی فایل /etc/passwd با /etc/passwd.bak، از کاربر تأیید می‌گیرد.

n:–

```
rsync -n /home/user/Downloads /cloud/Downloads
```

فرآیند همگام‌سازی را شبیه‌سازی می‌کند، بدون اینکه هیچ تغییری ایجاد کند.

p:–

```
rsync -p /home/user/Projects /external/Projects
```

مجوزها، مالکیت و زمان‌های دسترسی را در حین همگام‌سازی فایل‌ها از دایرکتوری home/user/Projects/ به external/Projects/ حفظ می‌کند.

q:–

```
rsync -q /var/log /backup/log
```

خروجی دستور را فقط به اطلاعات ضروری محدود می‌کند.

r:–

```
rsync -r /home/user/Website /public_html
```

دایرکتوری Website را به طور بازگشتی به دایرکتوری public_html همگام‌سازی می‌کند.

t:–

```
rsync -t /home/user/Documents/text /remote/Documents/text
```

فقط فایل‌های متنی را در دایرکتوری Documents/text به دایرکتوری remote/Documents/text همگام‌سازی می‌کند.

u:–

```
rsync -u /home/user/Music /mp3/Music
```

فقط فایل‌هایی را که در دایرکتوری Music جدیدتر از فایل‌های موجود در دایرکتوری mp3/Music هستند، همگام‌سازی می‌کند.

v:-

```
rsync -v /home/user/Pictures /gallery/Pictures
```

جزئیات مربوط به فرآیند همگام‌سازی فایل‌ها از دایرکتوری home/user/Pictures/ به gallery/Pictures/ را نمایش می‌دهد.

z:-

```
rsync -z /home/user/Videos /archive/Videos
```

از الگوریتم فشرده‌سازی برای بهینه‌سازی فرآیند همگام‌سازی فایل‌ها از دایرکتوری home/user/Videos/ به archive/Videos/ استفاده می‌کند.

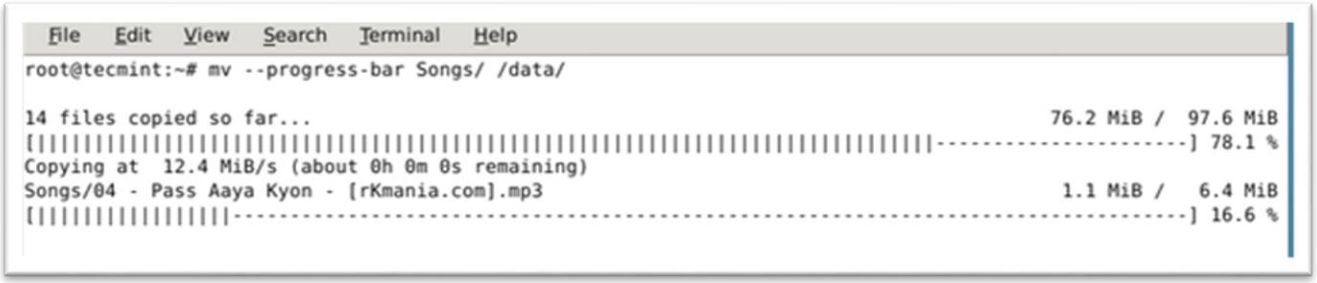
دستور mv

دستور mv مخفف move برای انتقال فایل به مکان دیگر است. در اینجا نمونه ای از دستور mvبا صفحه نمایش است.

```
# mv  --progress-bar  /mp3/*  /data/
```

یا

```
# mv  –g  /mp3  /data /
```



دستور date

تاریخ وزمان system را نمایش و تنظیم میکند. مانند این.

ساختار دستور:

```
date [OPTION]... [+FORMAT]
date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
```

گزینه‌های دستور:

- d- تاریخ و زمان را به صورت دلخواه تنظیم می‌کند.
- f- فرمت دلخواه برای نمایش تاریخ و زمان را مشخص می‌کند.
- h- راهنمای دستور را نمایش می‌دهد.
- I- تاریخ و زمان را به صورت ISO 8601 نمایش می‌دهد.
- j- تاریخ و زمان را به صورت روز سال نمایش می‌دهد.
- n- بدون تغییر تاریخ و زمان سیستم، فقط آن را نمایش می‌دهد.
- R- تاریخ و زمان را به صورت (Epochتعداد ثانیه‌ها از ۱ ژانویه ۱۹۷۰) نمایش می‌دهد.
- s- تاریخ و زمان سیستم را تنظیم می‌کند.

- t- منطقه زمانی را برای نمایش تاریخ و زمان مشخص می‌کند.
- u- تاریخ و زمان را به صورت UTC نمایش می‌دهد.
- V- اطلاعات مربوط به نسخه دستور را نمایش می‌دهد.

```
$ date
```

نمایش خروجی در فرمت دلخواه

```
date '+%A %W %Y %X'
```

دریافت و مشاهده زمان بر حسب میلی ثانیه

```
date +"%T.%N"
```

زمان فعلی را با نانوثانیه برمی گرداند.

06:46:41.431857000

date +"%T.%6N" زمان فعلی را با نانوثانیه های گرد شده به ۶ رقم اول برمی گرداند که میکروثانیه است.

06:47:07.183172

date +"%T.%3N" زمان فعلی را با نانوثانیه های گرد شده به ۳ رقم اول برمی گرداند که میلی ثانیه است.

06:47:42.773

دستور hwclock

از دستور زیر نیز میتوان خروجی تاریخ را بدست آورد. دستور hwclock برای تنظیم و خواندن زمان از ساعت سخت افزاری (RTC) در سیستم های لینوکس استفاده می شود.

```
hwclock
```

ساختار دستور:

```
hwclock [functions] [options]
```

گزینه های دستور:

- h- راهنمای دستور را نمایش می دهد.
- r- زمان فعلی را از RTC می خواند و آن را به عنوان زمان سیستم تنظیم می کند.
- s- زمان سیستم را به عنوان زمان RTC تنظیم می کند.
- w- تنظیمات RTC را به طور دائمی در فایل /etc/adjtime ذخیره می کند.
- f- فایل حاوی تنظیمات RTC را مشخص می کند.
- d- تاریخ و زمان را به صورت دستی تنظیم می کند.
- u- تنظیمات UTC را به جای تنظیمات محلی اعمال می کند.
- t- منطقه زمانی را مشخص می کند.
- V- نسخه برنامه را نمایش می دهد.

مثال های دستور:

خواندن زمان از RTC:

```
hwclock -r
```

تنظیم زمان سیستم به عنوان زمان RTC:

```
hwclock -s
```

ذخیره دائمی تنظیمات RTC:

```
hwclock -w
```

تنظیم دستی تاریخ و زمان:

```
hwclock -d “2023-11-15 12:34:56”
```

تنظیم زمان به صورت UTC:

```
hwclock -u
```

مشخص کردن منطقه زمانی:

```
hwclock -t “Asia/Tehran”
```

نمایش نسخه برنامه:

```
hwclock -V
```

دستور rename

یک ابزار خط دستوربرای تغییر نام چندین فایل در لینوکس

ما اغلب از دستور mvبرای تغییر نام یک فایل در لینوکس استفاده می کنیم .با این حال، تغییر نام چند یا گروهی از فایل ها به سرعت کار را در ترمینال بسیار دشوار میکند .لینوکس دارای یک ابزار داخلی بسیار قدرتمند به نام تغییر نام است .تغییر نام دستور برای تغییر نام فایل ها تغییر نام متعدد یا گروهی از فایل ها، به حروف کوچک، فایل های تغییر نام به حروف بزرگ است.

```
$ which rename
/usr/bin/rename
```

دستور تغییر نام همراه با چند آرگومان اختیاری همراه با عبارت اجباری پرل است که دستور تغییر نام را برای انجام کار واقعی راهنمایی می کند.

```
rename [ -v ] [ -n ] [ -f ] perlexpr [ files ]
```

گزینه های دستور:

- -v چاپ نام فایل هایی که با موفقیت تغییر نام داده اند.
- -nنشان می دهد که نام چه فایل هایی تغییر می کند.
- -fبازنویسی اجباری فایل های موجود.

دستور rename در سیستم‌های عامل مبتنی بر یونیکس ((UNIXو لینوکس برای تغییر نام یک یا چند فایل یا فهرست به صورت دسته‌ای (با استفاده از الگوها) استفاده می‌شود. این دستور قابلیت جستجو و جایگزینی الگوهای مشخصی در نام فایل‌ها را دارد. ساختار کلی دستور rename به صورت زیر است:

```
rename [OPTIONS] ‘s/OLD_PATTERN/NEW_PATTERN/’ FILES
```

در اینجا، OLD_PATTERN نشان‌دهنده الگویی است که می‌خواهید در نام فایل‌ها جستجو شود و NEW_PATTERN نشان‌دهنده الگویی است که می‌خواهید جایگزین الگوی قبلی در نام فایل‌ها شود. FILES نیز نشان‌دهنده فایل‌ها یا فهرست‌هایی است که می‌خواهید تغییر نام دهید.

```
rename ‘s/old_name/new_name/’  file.txt
```

تغییر نام چندین فایل با استفاده از الگوی جستجو و جایگزینی:

```
rename ‘s/old_name/new_name/’  file1.txt file2.txt file3.txt
```

تغییر نام فایل‌ها با الگوی خاص:

```
rename ‘s/(.*)\.txt/$1_new.txt/’ *.txt
```

در این مثال، الگوی جستجوی (.*)\.txt برای هر فایل با پسوند .txt مطابقت می‌کند و با استفاده از الگوی جایگزینی new.txt_\1$ نام فایل را تغییر می‌دهد. الگوی (.*) هر چیزی را که قبل از txt قرار دارد، گرفته و آن را در الگوی جایگزینی new.txt_\1$ قرار می‌دهد. دستور rename بسیار قابل تنظیم و قدرتمند است و می‌توانید با استفاده از الگوها و قواعد مورد نیاز خود، تغییرات دسته‌ای روی نام فایل‌ها اعمال کنید. قبل از استفاده از این دستور، بهتر است ابتدا با مستندات و مثال‌های مربوطه آشنا شوید تا از صحیح و ایمن بودن تغییرات اطمینان حاصل کنید.

دستور file

دستور file در سیستم‌های Unix/Linux استفاده می‌شود تا نوع یا محتوای یک فایل را تشخیص دهد. با اجرای دستور file بر روی یک فایل، سیستم تلاش می‌کند تا نوع فایل (مثلاً متنی، تصویر، فایل اجرایی و غیره) را شناسایی کند و اطلاعات مربوط به آن را نمایش دهد.

ساختار دستور:

```
file [نام_فایل]
```

گزینه های دستور:

- b– نام فایل را از خروجی حذف می‌کند.
- c– نوع فایل را به صورت خلاصه نمایش می‌دهد.
- e– اطلاعات مربوط به رمزگذاری فایل را نمایش می‌دهد.
- f– فایلی را که می‌خواهید نوع آن را تشخیص دهید، مشخص می‌کند.
- h– راهنمای دستور را نمایش می‌دهد.
- i– اطلاعات مربوط به نوع MIME فایل را نمایش می‌دهد.
- m– اطلاعات مربوط به جادوی فایل را نمایش می‌دهد.
- n– تعداد خطوطی که باید از ابتدای فایل خوانده شوند را مشخص می‌کند.
- p– اطلاعات مربوط به نوع فایل را به صورت تفصیلی نمایش می‌دهد.
- s– اندازه فایل را نمایش می‌دهد.
- v– جزئیات مربوط به فرآیند تشخیص نوع فایل را نمایش می‌دهد.
- z– اطلاعات مربوط به فشرده‌سازی فایل را نمایش می‌دهد.

مثال های دستور:

۱. file filename: این دستور برای نمایش نوع فایل مورد نظر استفاده می‌شود.

```
file example.txt
```

۲. file -b filename: این دستور برای نمایش نوع فایل مورد نظر بدون اطلاعات اضافی استفاده می‌شود.

```
file -b example.txt
```

۳. file -i filename: این دستور برای نمایش نوع MIME (Multipurpose Internet Mail Extensions) فایل مورد نظر استفاده می‌شود.

```
file -i example.txt
```

۴. file -z filename: این دستور برای نمایش نوع فایل مورد نظر به همراه اطلاعات فشرده‌سازی استفاده می‌شود.

```
file -z example.zip
```

۵. file -L filename: این دستور برای نمایش نوع فایل مورد نظر به همراه لینک‌های سمبولیک استفاده می‌شود.

```
file -L example_symlink
```

مثال های بیشتر:

تشخیص نوع فایل بدون نام:

```
file file.txt
```

خروجی:

```
file.txt: ASCII text
```

تشخیص نوع فایل به صورت خلاصه:

```
file -c file.txt
```

خروجی:

text

نمایش اطلاعات مربوط به رمزگذاری فایل:

```
file -e file.txt
```

خروجی:

```
file.txt: UTF-8 Unicode text
```

تشخیص نوع فایل با نام:

```
file -f /path/to/file.txt
```

خروجی:

/path/to/file.txt: ASCII text

نمایش راهنمای دستور:

```
file -h
```

نمایش اطلاعات مربوط به نوع MIME فایل:

```
file -i file.txt
```

خروجی:

```
file.txt: text/plain
```

نمایش اطلاعات مربوط به جادوی فایل:

```
file -m file.txt
```

خروجی:

۰ ۲۰ ۷۳ ۶۹ ۲۰ ۷۳ ۶۹ ۶۸ ۵۴ :۰۰۰۰۰۰۰۰a 00 00 00 00 00 00 00

۰۰ ۰۰ ۰۰ ۰۰ ۰۰ ۰۰ ۰۰ ۰۰ ۰۰ ۰۰ ۰۰ ۰۰ ۰۰ ۰۰ ۰۰ ۰۰ :۰۰۰۰۰۰۱۰

نمایش ۱۰ خط اول فایل:

```
file -n 10 file.txt
```

خروجی:

This is a text file.
It has multiple lines.

نمایش اطلاعات مربوط به نوع فایل به صورت تفصیلی:

```
file -p file.txt
```

خروجی:

file.txt: ASCII text, with CRLF line terminators

نمایش اندازه فایل:

```
file -s file.txt
```

خروجی:

۱۲۳bytes

نمایش جزئیات مربوط به فرآیند تشخیص نوع فایل:

```
file -v file.txt
```

خروجی:

```
file: trying to detect file type...
file: from `file.txt':
file: trying to find magic...
file: found 1 magic matches:
file: `file.txt': ASCII text
```

نمایش اطلاعات مربوط به فشرده‌سازی فایل:

```
file -z file.txt
```

خروجی:

```
file.txt: uncompressed
```

این مثال‌ها تعدادی از استفاده‌های معمول دستور fileدر لینوکس هستند. این دستورات همراه با گزینه‌های مختلف می‌توانند به شما کمک کنند تا اطلاعات مفیدی درباره فایل‌های مختلف در سیستم خود دریافت کنید.

دستور stat

stat برای نشان دادن وضعیت فایل یا سیستم فایل مانند این استفاده می شود. سویچ f- برای مشخص کردن یک سیستم فایل.

```
$ stat file1
```

دستور stat در سیستم‌های عامل مبتنی بر یونیکس (UNIX) و لینوکس برای نمایش اطلاعات جامع در مورد یک فایل یا فهرست استفاده می‌شود. این دستور اطلاعاتی مانند سایز فایل، تاریخ و زمان ایجاد، تاریخ و زمان آخرین تغییرات، مجوزهای دسترسی و مالکیت فایل را نمایش می‌دهد.

ساختار دستور:

```
stat [OPTIONS] FILE
```

در اینجا، FILE نشان‌دهنده مسیر و نام فایل یا فهرست مورد نظر است.

مثال های دستور:

نمایش اطلاعات جامع یک فایل:

```
stat file.txt
```

نمایش اطلاعات جامع یک فهرست:

```
stat directory
```

نمایش اطلاعات جامع چندین فایل در یک دستور:

```
stat file1.txt file2.txt file3.txt
```

نمایش اطلاعات جامع یک فایل به صورت فرمت‌بندی شده:

```
stat -c “File: %n | Size: %s bytes | Permissions: %a | Owner: %U | Group: %G” file.txt
```

در این مثال، از گزینه -c استفاده شده است تا فرمت خاصی برای نمایش اطلاعات انتخاب شود. در اینجا %n برای نام فایل، %s برای سایز فایل، %a برای مجوزهای دسترسی، %U برای نام مالک فایل و %G برای نام گروه مالک فایل استفاده شده است.

دستور chacl

دستور chacl در لینوکس برای تغییر و مدیریت لیست کنترل دسترسی‌ها (ACLs) روی فایل‌ها و فهرست‌ها استفاده می‌شود. این دستور به کاربران امکان می‌دهد تا ACLs را برای هر فایل یا فهرست به صورت دلخواه تنظیم کنند.

ساختار دستور:

```
chacl acl pathname...
chacl -b acl dacl pathname...
chacl -d dacl pathname...
chacl -R pathname...
chacl -D pathname...
chacl -B pathname...
chacl -l pathname...
chacl -r pathname...
```

گزینه‌های دستور:

b – نشان می‌دهد که دو ACL برای تغییر وجود دارد، ACL دسترسی فایل اول و ACL پیش‌فرض دایرکتوری دوم است.

d – برای تنظیم فقط ACL پیش‌فرض یک دایرکتوری استفاده می‌شود.

R – تنها ACL دسترسی فایل را حذف می‌کند.

D – تنها ACL پیش‌فرض دایرکتوری را حذف می‌کند.

B – تمام ACL ها را حذف می‌کند.

l ACL – دسترسی و احتمالا ACL پیش‌فرض مرتبط با فایل یا دایرکتوری مشخص شده را لیست می‌کند. این گزینه در زمان پورت لینوکس XFS اضافه شده است و با IRIX سازگار نیست.

r ACL – دسترسی را به صورت بازگردانی برای هر زیردرختی با ریشه در مسیر(های) تعیین شده تنظیم می‌کند. این گزینه همچنین در زمان پورت لینوکس XFS اضافه شده است و با IRIX سازگار نیست.

مثال‌های دستور:

تنظیم ACL برای یک فایل یا فهرست:

```
chacl user::rwx,group::r--,other::--- myfile.txt
```

تنظیم ACL و Default ACL برای یک فایل یا فهرست:

```
chacl -b user::rwx,group::r--,other::--- default:user::rwx,default:group::r--,default:other::---
mydir
```

حذف Default ACL برای یک فایل یا فهرست:

```
chacl -d mydir
```

بازگردانی ACL های یک فایل یا فهرست به حالت اولیه:

```
chacl -R myfile.txt
```

حذف تمامی ACL ها از یک فایل یا فهرست:

```
chacl -D mydir
```

پشتیبان‌گیری از ACL های یک فایل یا فهرست:

```
chacl -B myfile.txt
```

نمایش ACL های یک فایل یا فهرست:

```
chacl -l myfile.txt
```

بازگردانی ACL های یک فایل یا فهرست به حالت پیش‌فرض:

```
chacl -r mydir
```

دستور cksum

cksum فرمانی در سیستم‌عاملهای شبه یونیکس است جهت تولید سرجمع یا چکسام برای فایلها یا جریان‌های اطلاعاتی (مانند ورودی و خروجی‌های استاندارد)است. فرمان Cksum فایل یا فایلهایی را بعنوان آرگومان دریافت می‌کند و از آنها یک چک‌سام، کد افزونگی چرخشی (CRC) و تعداد بایتهای موجود را محاسبه می‌کند. در خروجی این فرمان، چک‌سام، تعداد بایتها و در انتها نام فایل نوشته خواهد شد. فرمان cksum می‌تواند برای بررسی کردن برابری دو فایل در کامپیوترهای مختلف استفاده شود. مثلاً زمانی که بخواهیم دو فایل را که توسط یک مسیر انتقال پرنویز کپی شدهاند با هم مقایسه کنیم و مطمئن شویم که کاملاً مشابه همدیگرند و عملیات کپی به‌درستی انجام شده است.

ساختار دستور:

```
$ cksum   FILE]]
$ cksum   [OPTION]
```

مجموع بررسی CRC و تعداد بایت هر فایل را چاپ میکند. دستور "cksum" برای محاسبه و بررسی مقادیر کنترلی (checksum) فایل‌ها استفاده می‌شود. در زیر چند مثال برای استفاده از گزینه‌های مختلف این دستور آورده شده است:

انتخاب نوع الگوریتم مقدار کنترلی:(digest)

```
$cksum -a sha256 file.txt
```

تولید مقدار کنترلی به صورت base64 به جای هگزادسیمال:

```
$cksum --base64 file.txt
```

بررسی مقادیر کنترلی از فایل‌ها:

```
$cksum -c checksums.txt file1.txt file2.txt
```

تولید مقدار کنترلی به طول مشخص (به بیت):

```
$cksum -l 256 file.txt
```

تولید مقدار کنترلی به صورت دودویی (به جای هگزادسیمال):

```
$cksum --raw file.txt
```

تولید مقدار کنترلی به سبک BSD پیش‌فرض

```
$cksum --tag file.txt
```

تولید مقدار کنترلی به سبک معکوس بدون نوعdigest :

```
$cksum --untagged file.txt
```

پایان هر خط خروجی را با NUL نهnewline و غیرفعال کردن escaping نام فایل:

```
$cksum -z file.txt
```

علاوه بر این، برای بررسی مقادیر کنترلی، می‌توانید از گزینه‌های زیر استفاده کنید:
عدم خطا یا گزارش برای فایل‌هایی که موجود نیستند:

```
$cksum --ignore-missing checksums.txt
```

عدم چاپ OK برای هر فایل که با موفقیت بررسی شده است:

```
$cksum --quiet checksums.txt
```

عدم نمایش هیچ مقداری و استفاده از کد وضعیت برای نشان دادن موفقیت:

```
$cksum --status checksums.txt
```

خروجی غیرصحیح برای خطوط مقدار کنترلی نادرست:

```
$cksum --strict checksums.txt
```

هشدار در مورد خطوط مقدار کنترلی به صورت نادرست:

```
$cksum --warn checksums.txt
```

نشان دادن اینکه کدام پیاده‌سازی استفاده شده است:

```
$cksum --debug file.txt
```

نمایش راهنما و خروج:

```
$cksum --help
```

نمایش اطلاعات نسخه و خروج:

```
$cksum --version
```

لطفاً توجه داشته باشید که مسیر و نام فایل‌ها و checksums.txt در مثال‌ها باید با موارد واقعی جایگزین شوند. همچنین، مقادیر کنترلی و خروجی‌ها برای هر مثال ممکن است متفاوت باشند و بسته به الگوریتم و ورودی‌های استفاده شده، متفاوت باشند.

دستور md5sum

md5sum یک برنامه کامپیوتری است که هش های MD5 128 بیتی را محاسبه و تأیید می کند، همانطور که در RFC 1321 توضیح داده شده است. هش MD5 به عنوان اثر انگشت دیجیتال فشرده یک فایل عمل می کند. مانند همه این الگوریتم‌های هش‌سازی، از نظر تئوری تعداد نامحدودی از فایل‌ها وجود دارد که دارای هر هش MD5 هستند. با این حال، بسیار بعید است که هر دو فایل غیر یکسان در دنیای واقعی دارای هش MD5 یکسان باشند، مگر اینکه به طور خاص برای این کار ایجاد شده باشند. الگوریتم MD5 زیربنایی دیگر امن تلقی نمی شود. بنابراین، در حالی که md5sum برای شناسایی فایل‌های شناخته شده در موقعیت‌هایی که مربوط به امنیت نیستند، مناسب است، اگر احتمال دستکاری عمدی و مخرب فایل‌ها وجود داشته باشد، نباید به آن اعتماد کرد. در مورد دوم، استفاده از ابزار هش جدیدتر مانند sha256sum توصیه می شود. معمولاً یافتن دو فایل مجزا که منجر به رشته های یکسان می شوند بسیار دشوار است. بنابراین، می‌توانید از md5sum برای بررسی یکپارچگی داده‌های دیجیتال با تعیین اینکه فایل یا ISOی که دانلود کرده‌اید یک کپی بیت به بیت از فایل راه دور یا ISO استفاده کنید.

دستورزیر یک مقدار هش برای فایل به شرح زیر تولید:

```
$ md5sum group.csv

bc527343c7ffc103111f3a694b004e2f  group.csv
```

هنگامی که سعی می کنید محتوای فایل را با حذف خط اول تغییر دهید `root:x:0:` و سپس دستور را برای بار دوم اجرا کنید، سعی کنید مقدار هش را ببینید:

```
$ md5sum group.csv

46798b5cfca45c46a84b7419f8b74735  group.csv
```

متوجه خواهید شد که مقدار هش اکنون تغییر کرده است، که نشان می دهد محتویات فایل تغییر کرده است.

حالا خط اول فایل را برگردانید `root:x:0:` و نام آن را به **group_file.txt** تغییر دهید و دستور زیر را اجرا کنید تا مقدار هش آن دوباره ایجاد شود:

```
$ md5sum group_list.txt

bc527343c7ffc103111f3a694b004e2f   group_list.txt
```

از خروجی بالا، حتی زمانی که فایل با محتوای اصلی آن تغییر نام داده است، مقدار هش همچنان یکسان است.

md5 sums: فقط با محتوای فایل به جای نام فایل تایید/کار می کند.

فایل group_list.txt تکراری از group.csv است ، بنابراین سعی کنید مقدار هش فایل ها را همزمان به صورت زیر تولید کنید . خواهید دید که هر دو دارای مقادیر هش مساوی هستند، زیرا محتوای آنها دقیقاً یکسان است.

```
$ md5sum group_list.txt groups.csv

bc527343c7ffc103111f3a694b004e2f   group_list.txt
 bc527343c7ffc103111f3a694b004e2f   group.csv
```

می توانید مقدار هش یک فایل(ها) را به یک فایل متنی هدایت کنید و ذخیره کنید، آنها را با دیگران به اشتراک بگذارید .برای دو فایل بالا، می توانید دستور زیر را صادر کنید تا مقادیر هش تولید شده را به یک فایل متنی برای استفاده بعدی هدایت کنید:

```
$ md5sum groups_list.txt groups.csv > myfiles.md5
```

برای بررسی اینکه فایل‌ها از زمانی که checksum را ایجاد کرده‌اید تغییرنکرده‌اند، دستور بعدی را اجرا کنید .شما باید بتوانید نام هر فایل را همراه با "OK"مشاهده کنید .

`-c` یا `--check` گزینه مقادیر MD5 فایل ها را بررسی کنید.

```
$ md5sum -c myfiles.md5
```

به یاد داشته باشید که پس از ایجاد چک‌سوم، نمی‌توانید نام فایل‌ها را تغییر دهید وگرنه وقتی می‌خواهید فایل‌ها را با نام‌های جدید تأیید کنید، با خطای فایل یا فهرستی وجود ندارد مواجه می‌شوید.

برای مثال:

```
$ mv group_list.txt new.txt
$ mv group.csv file.txt
$ md5sum -c myfiles.md5
```

این مفهوم همچنین برای رشته ها به طور یکسان کار می کند، در دستورات زیر، `-n` به این معنی است که خط جدید انتهایی را خروجی نکنید:

```
$ echo -n "Predator-os  How-Tos" | md5sum -

afc7cb02baab440a6e64de1a5b0d0f1b -
$ echo -n "Predator-os  How-To" | md5sum -

65136cb527bff5ed8615bd1959b0a248 -
```

دستور sha256sum

sha256sumیک دستور در لینوکس است که برای محاسبه مقدار SHA-256 یک فایل استفاده می‌شود. به عنوان مثال، فرض کنید می‌خواهید فایلی به نام example.txt را با استفاده از sha256sum بررسی کنید و مشخص شود آیا مقدار SHA-256 آن با یک مقدار خاص مطابقت دارد یا خیر. برای این منظور، ابتدا یک مقدار SHA-256 معتبر برای فایل example.txt تولید می‌کنیم. سپس با استفاده از گزینه c– یا check–– وارد مرحله بررسی مطابقت می‌شویم.

۱. تولید مقدار SHA-256 برای فایل:

```
echo -n “Hello, World!” | sha256sum
```

خروجی دستور:

4c4fcd2f5b8e4f1b8f7f8056b9c6a1e0cb5f55e7a5a5e8c3df3b7d1f3b8c3a5d

۲. بررسی مطابقت فایل با مقدار SHA-256:

```
sha256sum -c <<< “4c4fcd2f5b8e4f1b8f7f8056b9c6a1e0cb5f55e7a5a5e8c3df3b7d1f3b8c3a5d example.txt”
```

example.txt: OK

دستور sha512sum

به عنوان مثال، فرض کنید می‌خواهید فایلی به نام example.txt را با استفاده از sha512sum بررسی کنید و مشخص شود آیا مقدار SHA-512 آن با یک مقدار خاص مطابقت دارد یا خیر. برای این منظور، ابتدا یک مقدار SHA-512 معتبر برای فایل example.txt تولید می‌کنیم. سپس با استفاده از گزینه c– یا check–– وارد مرحله بررسی مطابقت می‌شویم.

تولید مقدار SHA-512 برای فایل:

```
echo -n “Hello, World!” | sha512sum
```

خروجی دستور:

۲ef7bde608ce5404e97d5f042f95f89f1c232871a467dd5f9178568e9d9ec4c7c7d1e3f3a759b3e4d45c16d3edba7f3552dd9a78e6f7a739b6ac3b3f146c29b5 -

بررسی مطابقت فایل با مقدار SHA-512:

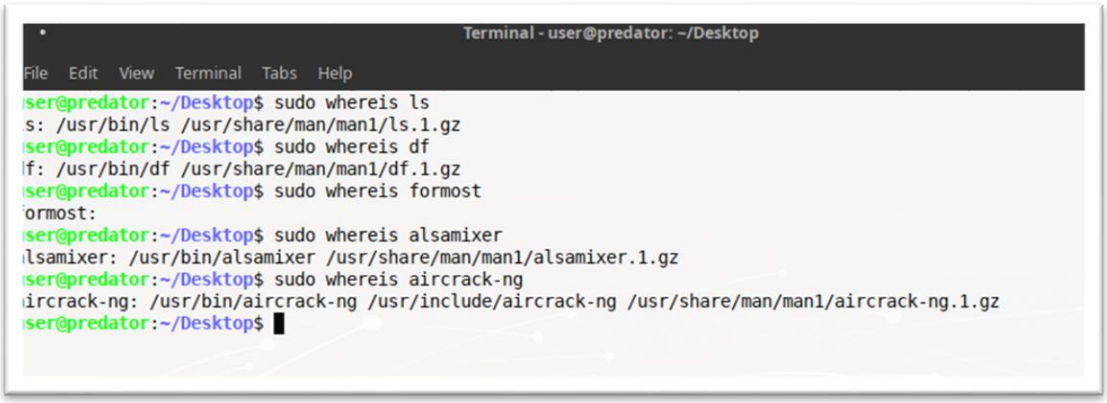
```
sha512sum -c <<<“2ef7bde608ce5404e97d5f042f95f89f1c232871a467dd5f9178568e9d9ec4c7c7d1e3f3a759b3e4d45c16d3edba7f3552dd9a78e6f7a739b6ac3b3f146c29b5 example.txt”
```

خروجی دستور:

example.txt: OK

مشاهده محل اجرا دستورات

دستور whereis فایل های باینری، منبع و فایل های صفحه دستی را برای نام دستور به صورت زیر ارائه می کند:



bash دارای دو ویژگی بزرگ است که به کمک می کند دستورات کامل و دوباره اجرا کنید، به سادگی قسمت اول یک دستور را تایپ کنید، کلید <tab> را بزنید و ترمینال حدس میزند چه کاری انجام میخواهید انجام دهید.

```
ls <ENTER>
apro   <TAB>
```

با دستور tab دستور کامل می شود.

فصل ششم

فصل ششم

مدیریت بسته ها

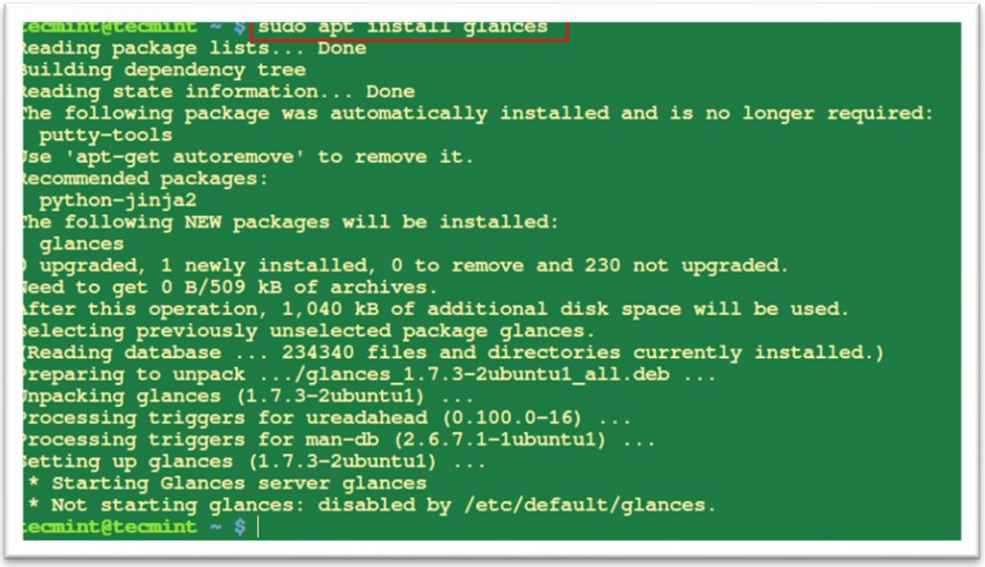
یکی از مهم ترین مواردی است که تحت system linux / سرور اداره می شود مدیریت بسته با استفاده از ابزارهای مختلف مدیریت بسته است .توزیع های مختلف linux برنامه های کاربردی را در یک بسته پیش ساخته شده که فایل های باینری، فایل های پیکربندی و همچنین اطلاعات مربوط به وابستگی های برنامه را شامل می شوند، نصب می کند. ابزار مدیریت بسته به مدیران سرور به روش های مختلف از جمله:

۱. دانلود و نصب نرم افزار
۲. نرم افزار را از مخزن استاندارد نصب میکند
۳. بروز اوری تمام نرم افزار نصب شده
۴. نگه داشتن اطلاعات در مورد نرم افزار نصب شده

مدیربسته apt

می توانید یک بسته را به صورت زیر تنظیم ونصب کنید، با نام یک بسته واحد را مشخص کنید یا بسته های زیادی را با یک لیست می توانید نصب کنید.

```
$ sudo apt install glances
```



دستور زیرکمک خواهد کرد که لیستی از تمام فایل هایی که در بسته ای به نام glancesوجود دارد .

```
ecmint@tecmint ~ $ sudo apt content glances
.
etc
etc/default
etc/default/glances
etc/glances
etc/glances/glances.conf
etc/init.d
etc/init.d/glances
usr
usr/bin
usr/bin/glances
usr/lib
usr/lib/python2.7
usr/lib/python2.7/dist-packages
usr/lib/python2.7/dist-packages/glances
usr/lib/python2.7/dist-packages/Glances-1.7.3.egg-info
usr/lib/python2.7/dist-packages/Glances-1.7.3.egg-info/dependency_links.txt
usr/lib/python2.7/dist-packages/Glances-1.7.3.egg-info/entry_points.txt
usr/lib/python2.7/dist-packages/Glances-1.7.3.egg-info/PKG-INFO
usr/lib/python2.7/dist-packages/Glances-1.7.3.egg-info/requires.txt
usr/lib/python2.7/dist-packages/Glances-1.7.3.egg-info/top_level.txt
usr/lib/python2.7/dist-packages/glances/data
usr/lib/python2.7/dist-packages/glances/data/css
usr/lib/python2.7/dist-packages/glances/data/css/default.css
usr/lib/python2.7/dist-packages/glances/data/html
usr/lib/python2.7/dist-packages/glances/data/html/base.html
usr/lib/python2.7/dist-packages/glances/data/html/default.html
usr/lib/python2.7/dist-packages/glances/data/img
```

پیدا کردن فایل های نصب شده با apt

دستور زیرکمک می کند تا اطلاعات درباره وابستگی های خاص بسته ای که مشخص می کنید نمایش داده شود.

```
$ sudo apt depends glances
```

```
tecmint@tecmint ~ $ sudo apt depends glances
glances
  Depends: python-psutil
  Depends: <python:any>
    python:i386
    python
  Depends: <python:any>
    python:i386
    python
  Depends: python
  Depends: python-pkg-resources
  Depends: adduser
  Depends: lsb-base
  Recommends: python-jinja2
tecmint@tecmint ~ $
```

جستجو برای بسته با apt

گزینه جستجو برای نام بسته داده شده جستجو می کند و تمام بسته های متناسب را نشان می دهد.

```
$ sudo apt search apache2
```

```
tecmint@tecmint ~ $ sudo apt search apache2
ih  apache2                          - Apache HTTP Server
p   apache2:i386                     - Apache HTTP Server
v   apache2-api-20120211             -
v   apache2-api-20120211:i386        -
i A apache2-bin                      - Apache HTTP Server (binary files and modul
p   apache2-bin:i386                 - Apache HTTP Server (binary files and modul
i A apache2-data                     - Apache HTTP Server (common files)
v   apache2-data:i386                -
p   apache2-dbg                      - Apache debugging symbols
p   apache2-dbg:i386                 - Apache debugging symbols
p   apache2-dev                      - Apache HTTP Server (development headers)
p   apache2-dev:i386                 - Apache HTTP Server (development headers)
p   apache2-doc                      - Apache HTTP Server (on-site documentation)
p   apache2-mpm-event                - transitional event MPM package for apache2
p   apache2-mpm-event:i386           - transitional event MPM package for apache2
p   apache2-mpm-itk                  - transitional itk MPM package for apache2
p   apache2-mpm-itk:i386             - transitional itk MPM package for apache2
p   apache2-mpm-prefork              - transitional prefork MPM package for apach
p   apache2-mpm-prefork:i386         - transitional prefork MPM package for apach
p   apache2-mpm-worker               - transitional worker MPM package for apache
p   apache2-mpm-worker:i386          - transitional worker MPM package for apache
v   apache2-prefork-dev              -
v   apache2-prefork-dev:i386         -
p   apache2-suexec                   - transitional package for apache2-suexec-pr
p   apache2-suexec:i386              - transitional package for apache2-suexec-pr
p   apache2-suexec-custom            - Apache HTTP Server configurable suexec pro
```

مشاهده اطلاعات مربوط به بسته با apt

این کمک می کند که اطلاعات مربوط به بسته یا بسته ها را نمایش دهید، دستور زیر را با مشخص کردن تمام بسته هایی که می خواهید اطلاعات را نمایش دهید، اجرا کنید.

```
$ sudo apt  show firefox
```

```
tecmint@tecmint ~ $ sudo apt show firefox
Package: firefox
State: installed
Automatically installed: no
Version: 43.0+linuxmint1+rosa
Priority: optional
Section: web
Maintainer: Ubuntu Mozilla Team <ubuntu-mozillateam@lists.ubuntu.com>
Architecture: amd64
Uncompressed Size: 106 M
Depends: lsb-release, libasound2 (>= 1.0.16), libatk1.0-0 (>= 1.12.4), libc6 (>=
         2.17), libcairo2 (>= 1.2.4), libdbus-1-3 (>= 1.0.2), libdbus-glib-1-2
         (>= 0.78), libfontconfig1 (>= 2.9.0), libfreetype6 (>= 2.2.1), libgcc1
         (>= 1:4.1.1), libgdk-pixbuf2.0-0 (>= 2.22.0), libglib2.0-0 (>= 2.31.8),
         libgtk2.0-0 (>= 2.24.0), libpango-1.0-0 (>= 1.22.0),
         libpangocairo-1.0-0 (>= 1.14.0), libstartup-notification0 (>= 0.8),
         libstdc++6 (>= 4.6), libx11-6, libxcomposite1 (>= 1:0.3-1), libxdamage1
         (>= 1:1.1), libxext6, libxfixes3, libxrender1, libxt6
Recommends: xul-ext-ubufox, libcanberra0, libdbusmenu-glib4, libdbusmenu-gtk4
Suggests: ttf-lyx
Conflicts: firefox
Replaces: kubuntu-firefox-installer, kubuntu-firefox-installer
Provides: gnome-www-browser, iceweasel, www-browser
Description: Safe and easy web browser from Mozilla
 Firefox delivers safe, easy web browsing. A familiar user interface, enhanced
 security features including protection from online identity theft, and
 integrated search let you get the most out of the web.
```

تأیید بسته برای هر وابستگی ناقص با apt

گاهی اوقات درطول نصب بسته، ممکن است اشتباهات مربوط به وابستگی بسته را ایجاد شودبرای ایجاد وابستگی[۱] های یک بسته دستور زیر را واردنماید.

[۱] dependency

```
tecmint@tecmint ~ $ sudo apt check firefox
Reading package lists... Done
Building dependency tree
Reading state information... Done
tecmint@tecmint ~ $
```

نمایش بسته های از دست رفته با apt

```
$ sudo apt recommends  apache2
```

```
tecmint@tecmint ~ $ sudo apt recommends apache2

No missing recommended packages were found for apache2
```

نمایش نسخه بسته نصب شده با apt

```
$ sudo apt version firefox
```

```
tecmint@tecmint ~ $ sudo apt version firefox
43.0+linuxmint1+rosa
tecmint@tecmint ~ $ sudo apt version apache2
2.4.7-1ubuntu4.8
tecmint@tecmint ~ $ sudo apt version perl
5.18.2-2ubuntu1
tecmint@tecmint ~ $
```

به روز رسانی بسته های سیستم با apt

این کمک می کند تا لیستی از بسته های مخازن مختلف موجود در system خود را بارگیری کنید و آنها را به روز رسانی کنید، زمانی که نسخه های جدید بسته ها و وابستگی های آنها وجود دارد.

```
$ sudo apt update
```

```
tecmint@tecmint ~ $ sudo apt update
Hit http://download.virtualbox.org trusty InRelease
Hit http://download.virtualbox.org trusty/contrib amd64 Packages
Hit http://download.virtualbox.org trusty/contrib i386 Packages
Ign http://download.virtualbox.org trusty/contrib Translation-en_IN
Ign http://download.virtualbox.org trusty/contrib Translation-en
Ign http://dl.google.com stable InRelease
Hit http://dl.google.com stable Release.gpg
Hit http://dl.google.com stable Release
Hit http://dl.google.com stable/main amd64 Packages
Get:1 http://security.ubuntu.com trusty-security InRelease [65.9 kB]
Ign http://archive.ubuntu.com trusty InRelease
Hit http://ppa.launchpad.net trusty InRelease
Ign http://packages.linuxmint.com rosa InRelease
Ign http://extra.linuxmint.com rosa InRelease
Get:2 http://archive.ubuntu.com trusty-updates InRelease [65.9 kB]
Hit http://ppa.launchpad.net trusty InRelease
Hit http://packages.linuxmint.com rosa Release.gpg
Hit http://extra.linuxmint.com rosa Release.gpg
Hit http://ppa.launchpad.net trusty InRelease
Get:3 http://security.ubuntu.com trusty-security/main amd64 Packages [454 kB]
```

ارتقا دادن سیستم با apt

این کمک می کند تا نسخه های جدیدی از تمام بسته های موجود در system خود را بروزرسانی کنید.

```
$ sudo apt upgrade
```

```
tecmint@tecmint ~ $ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  libvncclient0
Use 'apt-get autoremove' to remove it.
The following packages have been kept back:
  apache2 apache2-bin apache2-data libdrm-dev libdrm-intel1 libdrm-intel1:i386
  libdrm-nouveau2 libdrm-nouveau2:i386 libdrm-radeon1 libdrm-radeon1:i386
  libdrm2 libdrm2:i386
The following packages will be upgraded:
  apache2-utils apt-transport-https apt-utils atril atril-common base-files
  bind9-host ca-certificates caja caja-common coreutils cpio cpp-4.8 curl
  dnsutils ecryptfs-utils eom eom-common ffmpeg firefox firefox-locale-en
  flashplugin-installer g++-4.8 gcc-4.8 gcc-4.8-base gcc-4.8-base:i386
  gcc-4.9-base gcc-4.9-base:i386 gir1.2-caja gir1.2-gtk-2.0 gir1.2-ibus-1.0
  gir1.2-javascriptcoregtk-3.0 gir1.2-mate-panel gir1.2-webkit-3.0 git-man
  glib-networking glib-networking:i386 glib-networking-common
  glib-networking-services google-chrome-stable gtk2-engines-pixbuf
  gtk2-engines-pixbuf:i386 hexchat hexchat-common ibus-gtk:i386 ifupdown
```

حذف بسته نصب شده با apt

هنگام نصب یک بسته جدید در system خود، وابستگی ها نیز نصب شده و از برخی از کتابخانه های system با بسته های دیگر استفاده می کنند.پس از حذف این بسته خاص، وابستگی ها در system باقی خواهند ماند، بنابراین برای حذف آنها از autoremove به صورت زیر استفاده می شود:

```
$ sudo apt autoremove
```

```
tecmint@tecmint ~ $ sudo apt autoremove
Reading package lists... Done
Building dependency tree
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 229 not upgraded.
tecmint@tecmint ~ $
```

پاک کردن مخزن قدیمی بسته های دانلود شده با apt

گزینه 'clean' یا 'autoclean' تمام مخزن محلی قدیمی فایل های بسته دانلود شده را حذف می کند.

```
$ sudo autoclean
```

یا

```
$ sudo apt clean
```

```
tecmint@tecmint ~ $ sudo apt autoclean
Reading package lists... Done
Building dependency tree
Reading state information... Done
tecmint@tecmint ~ $ sudo apt clean
tecmint@tecmint ~ $
```

حذف بسته ها با فایل های پیکربندی آن با apt

برای حذف کامل بسته ها و وابستگی ها ان از سویچ purge استفاده میکنیم.

```
tecmint@tecmint ~ $ sudo apt purge glances
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  glances*
0 upgraded, 0 newly installed, 1 to remove and 229 not upgraded.
After this operation, 1,040 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 234394 files and directories currently installed.)
Removing glances (1.7.3-2ubuntu1) ...
 * Stopping Glances server glances
 * PID file not found
Purging configuration files for glances (1.7.3-2ubuntu1) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
tecmint@tecmint ~ $
```

نصب بسته ها با پسوند .deb با apt

برای نصب یک فایل deb. دستور زیر را با نام فایل به عنوان یک توضیح به صورت زیر اجرا کنید:

```
$ sudo apt deb atom-amd64.deb
```

```
tecmint@tecmint ~ $ sudo apt deb atom-amd64.deb
Selecting previously unselected package atom.
(Reading database ... 234336 files and directories currently installed.)
Preparing to unpack atom-amd64.deb ...
Unpacking atom (1.6.2) ...
Setting up atom (1.6.2) ...
Processing triggers for mime-support (3.54ubuntu1.1) ...
Processing triggers for desktop-file-utils (0.22-1ubuntu1) ...
tecmint@tecmint ~ $
```

راهنمای دستور apt

دستور زیر تمام گزینه های را با شرح آن در مورد نحوه استفاده از APTدر system لیست می کند.

```
tecmint@tecmint ~ $ apt help
apt
Usage: apt command [options]
       apt help command [options]

Commands:
add-repository  - Add entries to apt sources.list
autoclean       - Erase old downloaded archive files
autoremove      - Remove automatically all unused packages
build           - Build binary or source packages from sources
build-dep       - Configure build-dependencies for source packages
changelog       - View a package's changelog
check           - Verify that there are no broken dependencies
clean           - Erase downloaded archive files
contains        - List packages containing a file
content         - List files contained in a package
deb             - Install a .deb package
depends         - Show raw dependency information for a package
dist-upgrade    - Perform an upgrade, possibly installing and removing packages
download        - Download the .deb file for a package
dselect-upgrade - Follow dselect selections
held            - List all held packages
help            - Show help for a command
hold            - Hold a package
install         - Install/upgrade packages
policy          - Show policy settings
purge           - Remove packages and their configuration files
recommends      - List missing recommended packages for a particular package
rdepends        - Show reverse dependency information for a package
reinstall       - Download and (possibly) reinstall a currently installed package
remove          - Remove packages
search          - Search for a package by name and/or expression
show            - Display detailed information about a package
source          - Download source archives
sources         - Edit /etc/apt/sources.list with nano
unhold          - Unhold a package
update          - Download lists of new/upgradable packages
upgrade         - Perform a safe upgrade
version         - Show the installed version of a package
                  This apt has Super Cow Powers
tecmint@tecmint ~ $
```

دستور apt-cache

دستور apt-cache برای جستجوی بسته نرم افزاری مورد استفاده قرار می گیرد.به عبارت ساده، این ابزار برای جستجو بسته های نرم افزاری مورد استفاده قرار می گیرد، اطلاعات بسته ها را جمع آوری می کند و همچنین برای جستجو برای بسته های موجود برای نصب در system های مبتنی بر دبیان یا ubuntu مورد استفاده قرار می گیرد.

لیست بسته های نصب شده موجود با apt

برای لیست تمام بسته های موجود، دستور زیر را تایپ کنید.

```
$ apt-cache pkgnames
esseract-ocr-epo
pipenightdreams
mumudvb
libsvm-java
libmrpt-hmtslam0.9
libboost-timer1.50-dev
kcm-touchpad
g ++ - 4.5-multilib
```

مشاهده نام و جزئیات بسته با apt

برای جستجو نام بسته و با توضیحات قبل از نصب، از سویچ search استفاده کنید.با استفاده از جستجو با apt-cache لیستی از بسته ها با توضیحات کوتاه نمایش داده می شود.بگذارید بگوییم دوست دارید توضیحات بسته vsftpdرا بیابید، سپس دستور می شود:

```
$ apt-cache search vsftpd
```

برای جستجو و فهرست کردن تمام بسته هایی که از vsftpdشروع می شوند، می توانید از دستور زیر استفاده کنید.

```
$ apt-cache pkgnames vsftpd
vsttpd
```

بررسی Cache بسته ها با apt

دستور stats مربوط به cache memory را نمایش می دهد.به عنوان مثال ، دستور زیر نمایش داده خواهد شد Packets مجموع تعداد بسته های موجود در کش است.

```
$ apt-cache stats
```

ارتقا یک بسته نرم افزاری با apt

دستور ' upgrade ' برای ارتقاء تمام بسته های نصب شده در system استفاده می شود. با این حال، اگر می خواهید ارتقا دهید، از اینکه آیا بسته های نرم افزاری برای تکمیل وابستگی ها اضافه می شوند یا حذف می شوند، از "دستور dist-upgrade "استفاده کنید.

```
$ sudo apt dist-upgrade
```

نصب چند بسته همزمان با apt

می توانید نام چند بسته را همراه با دستور اضافه کنید تا چندین بسته را در یک زمان نصب کنید.به عنوان مثال ، دستور زیر بسته های " nethogs "و " goaccess " را نصب خواهد کرد.

```
$ sudo apt install nethogs goaccess
```

نصب چندبسته با Wildcard با apt

با استفاده از عبارت منظم می توانید چندین بسته با یک رشته اضافه کنید.برای مثال ، ما برای استفاده از چندین بسته ی که شامل رشته " * name * "هستند استفاده می کنیم.

```
$ sudo apt  install '* name *'
```

با استفاده از دستور زیر ' -no-upgrade ' بسته های نصب شده از بروزرسانی جلوگیری می کند.

```
$ sudo apt install packageName    --no-upgrade
```

ارتقا یک بسته خاص با apt

دستور " only-upgrade " بسته های جدیدی را نصب نمی کند، اما فقط بسته های نصب شده را ارتقا می دهد و بسته های جدید را غیر فعال می کند.

```
$ sudo apt install packageName  -only-upgrade
```

نصب یک نسخه خاص با apt

به سادگی با استفاده از ' = ' با نام پکیج و اضافه کردن نسخه دلخواه.

```
$ sudo apt install vsftpd = 2.3.5-3ubuntu1
```

دانلود یک بسته بدون نصب با apt

برای دانلودکد منبع بسته خاص، از گزینه ' -download-only source ' با نام 'package-name' به عنوان نشان داده شده استفاده کنید.

```
$ sudo apt -download-only  source vsftpd
```

دانلود سورس یک بسته با apt

برای دانلود و باز کردن کد منبع یک بسته به یک فهرست خاص، دستور زیر را تایپ کنید.

```
$ sudo apt  source vsftpd
```

مدیربسته dnf

مقدمه ای بر دستورDNF

DNF (Dandified Yum)یک ابزار مدیریت بسته در توزیع‌های لینوکس مبتنی بر RPM مانندRHEL ، CentOS، Rocky Linux، OELو Fedora است .می تواند بسته های نرم افزاری روی سیستم را نصب، به روز رسانی یا حذف کند .این نسخه نسل بعدی مدیریت بسته yum در نظر گرفته می شود .در نظر گرفته شده است که yum را با بهبود ویژگی هایی مانند حل وابستگی و استفاده از حافظه جایگزین کند .این اولین بار در فدورا ۱۸ معرفی شد و از زمان فدورا ۲۲ مدیر بسته پیش فرض فدورا بوده است.

هر دو dnf و yum برای نصب، حذف یا به‌روزرسانی بسته‌ها در سیستم مبتنی بر RPM استفاده می‌شوند DNF .جایگزینی به روز شده برای yum به دلیل مشکلات حل نشده در YUM است .مشکلات عبارتند از عملکرد ضعیف، استفاده از حافظه بالا و وضوح وابستگی کند DNF .از کتابخانه 'libsolv' برای حل وابستگی ها استفاده می کند، در حالی که yum از API عمومی استفاده می

کند dnf .به زبان هایC ، C++و Python نوشته شده است که باعث عملکرد بهتر و استفاده کم از حافظه می شود .از آنجایی که yumفقط در پایتون نوشته می شود، عملکرد پایینی دارد.

نحوه عملکردdnf

DNFیک ابزار مدیریت بسته است که برای نصب/به روز رسانی/کاهش/حذف بسته ها به مخازن متکی است .بنابراین شما باید یک مخزن داشته باشید تا بتوانید از dnf استفاده کنید. به طور پیش فرض، dnfاز فایل پیکربندی جهانی در /etc/dnf/dnf.conf/و تمام فایل های مخزن (.repo) موجود در /etc/yum.repos.d استفاده می کند .

.repoفایل شامل نام مخزن، مسیر مخزن، چک GPG و پارامتر را فعال کنید .بر اساس این پارامتر، dnfاز مخزن برای بسته های موجود پرس و جو می کند.

مثال های دستور:

اکثر dnfدستورات مشابه دستورات yum هستند .برای اجرای دستور dnf به امتیاز root نیاز است .ممکن است dnf از قبل در سیستم شما نصب نشده باشد

نصب مدیر بسته dnf

```
# yum install dnf
```

شما قادر خواهید بود dnfدستور را در ترمینال خود اجرا کنید .در زیر پر استفاده ترین dnfدستورات برای مدیریت بسته ها و وابستگی های آنها در سیستم های مبتنی بر RPM آورده شده است.

نصب بسته و وابستگی های با dnf

می توانید از این دستور برای نصب آخرین نسخه یک بسته و وابستگی های آن استفاده کنید.

```
# dnf install package_name
```

حذف یک بسته و وابستگی ها با dnf

برای حذف بسته های نصب شده و وابستگی های آن می توانید از دستور زیر استفاده کنید.

```
# dnf remove package_name
```

خروجی نمونه:

```
[root@rockylinux ~]# # dnf remove iotop
```

بروزرسانی یک بسته و وابستگی ها با dnf

این دستور به شما امکان می دهد تا بسته های نصب شده و وابستگی های آنها را به آخرین نسخه ها به روز کنید.

```
# dnf update package_name
```

شما می توانید از این دستور بدون هیچ آرگومان برای به روز رسانی هر بسته در سیستم استفاده کنید.

```
# dnf update
```

ارتقاء یک بسته و وابستگی ها با dnf

این دستور عملکرد مشابهی مانند updateدستور دارد که یک بسته و وابستگی های آن را به جدیدترین نسخه ارتقا می دهد.

```
# dnf update package_name
```

خروجی دستور:

```
[root@rockylinux ~]# dnf upgrade rsyslog
```

استفاده از این دستور بدون بسته به عنوان آرگومان، تمام بسته های نصب شده روی سیستم را ارتقا می دهد.

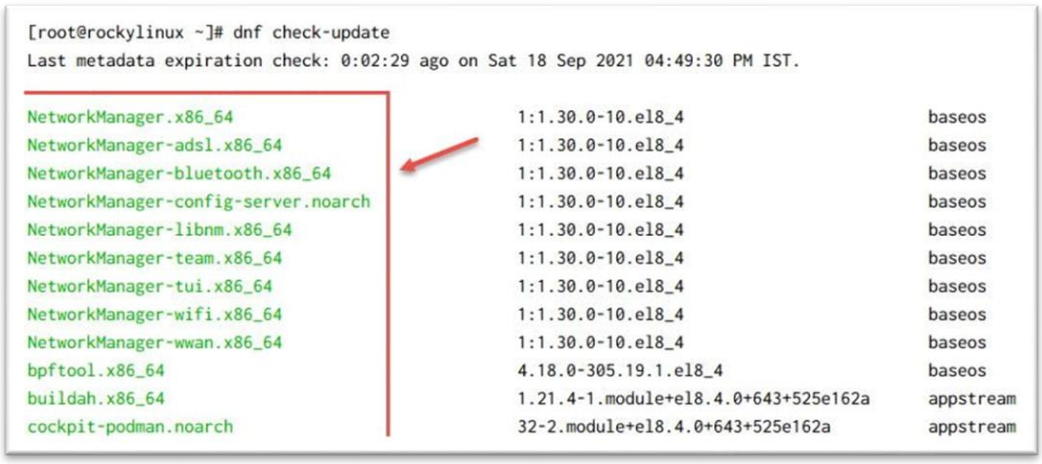
```
# dnf upgrade
```

بررسی به روز رسانی های موجود با dnf

می توانید به روز رسانی های موجود برای بسته های نصب شده و وابستگی های آنها را در سیستم خود بررسی کنید.

```
#dnf check-update
```

خروجی دستور:



```
[root@rockylinux ~]# dnf check-update
Last metadata expiration check: 0:02:29 ago on Sat 18 Sep 2021 04:49:30 PM IST.

NetworkManager.x86_64                    1:1.30.0-10.el8_4                    baseos
NetworkManager-adsl.x86_64               1:1.30.0-10.el8_4                    baseos
NetworkManager-bluetooth.x86_64          1:1.30.0-10.el8_4                    baseos
NetworkManager-config-server.noarch      1:1.30.0-10.el8_4                    baseos
NetworkManager-libnm.x86_64              1:1.30.0-10.el8_4                    baseos
NetworkManager-team.x86_64               1:1.30.0-10.el8_4                    baseos
NetworkManager-tui.x86_64                1:1.30.0-10.el8_4                    baseos
NetworkManager-wifi.x86_64               1:1.30.0-10.el8_4                    baseos
NetworkManager-wwan.x86_64               1:1.30.0-10.el8_4                    baseos
bpftool.x86_64                           4.18.0-305.19.1.el8_4                baseos
buildah.x86_64                           1.21.4-1.module+el8.4.0+643+525e162a appstream
cockpit-podman.noarch                    32-2.module+el8.4.0+643+525e162a     appstream
```

لیست کردن تمام بسته ها با dnf

این دستور همه بسته‌ها (نصب شده و در دسترس برای نصب) را برای توزیع لینوکس مبتنی بر RPM فهرست می‌کند.

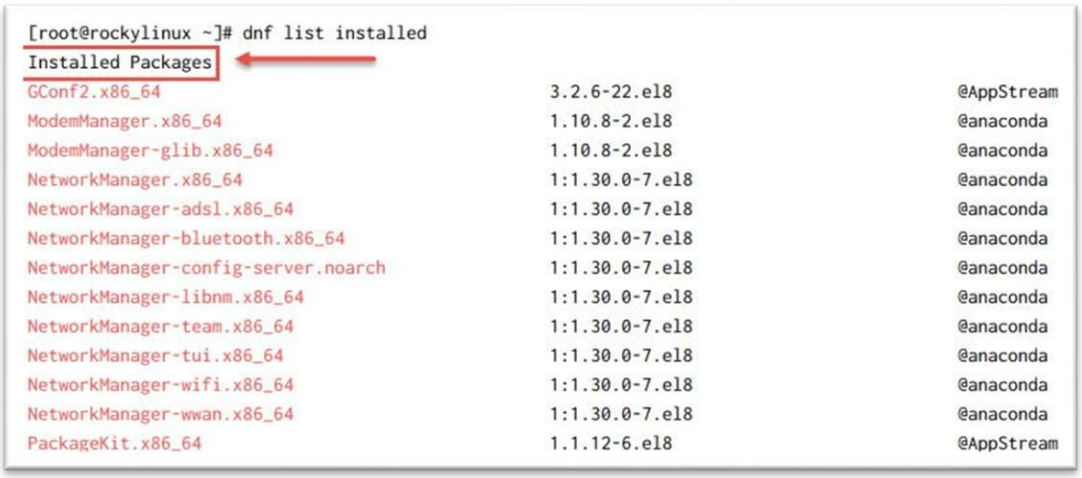
```
# dnf lsit
```

دریافت تنها لیست بسته های نصب شده با dnf

برای دریافت لیست تنها بسته های نصب شده روی سیستم می توانید از دستور زیر استفاده کنید.

```
# dnf list installed
```

خروجی دستور:



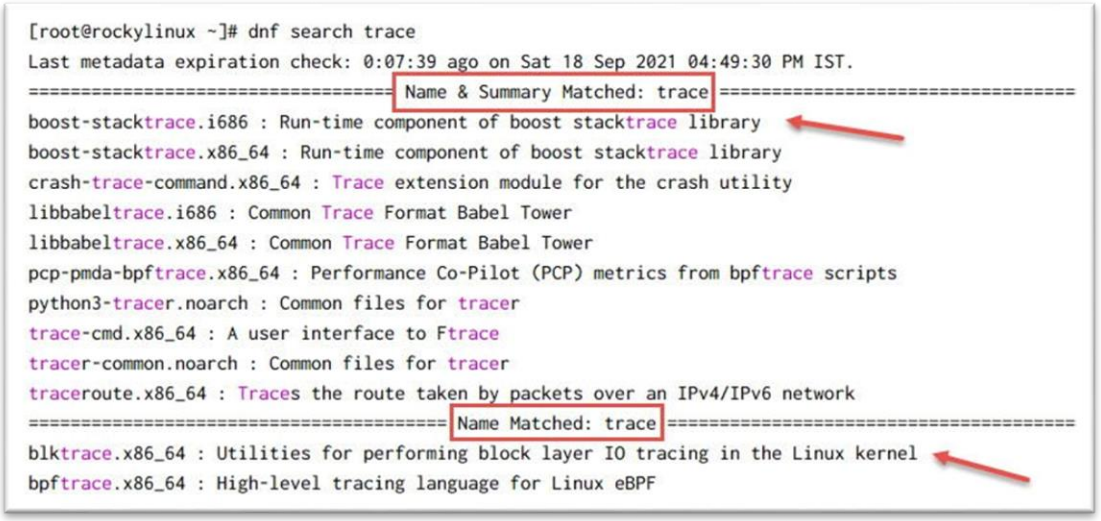
```
[root@rockylinux ~]# dnf list installed
Installed Packages
GConf2.x86_64                            3.2.6-22.el8                         @AppStream
ModemManager.x86_64                      1.10.8-2.el8                         @anaconda
ModemManager-glib.x86_64                 1.10.8-2.el8                         @anaconda
NetworkManager.x86_64                    1:1.30.0-7.el8                       @anaconda
NetworkManager-adsl.x86_64               1:1.30.0-7.el8                       @anaconda
NetworkManager-bluetooth.x86_64          1:1.30.0-7.el8                       @anaconda
NetworkManager-config-server.noarch      1:1.30.0-7.el8                       @anaconda
NetworkManager-libnm.x86_64              1:1.30.0-7.el8                       @anaconda
NetworkManager-team.x86_64               1:1.30.0-7.el8                       @anaconda
NetworkManager-tui.x86_64                1:1.30.0-7.el8                       @anaconda
NetworkManager-wifi.x86_64               1:1.30.0-7.el8                       @anaconda
NetworkManager-wwan.x86_64               1:1.30.0-7.el8                       @anaconda
PackageKit.x86_64                        1.1.12-6.el8                         @AppStream
```

جستجوی یک بسته در مخازن موجود با dnf

این دستور به شما امکان می دهد با استفاده از کلمه یک بسته خاص را جستجو کنید .اگر کلمه با نام یا خلاصه بسته مطابقت داشته باشد، نتیجه را چاپ می کند.

```
# dnf search [package_name]
```

خروجی دستور:



دریافت لیست بسته های خاص با dnf

شما همچنین می توانید لیست یک بسته خاص را با ارسال نام بسته به عنوان آرگومان دریافت کنید .اما مطمئن شوید که نام دقیق بسته را دارید وگرنه این دستور کار نخواهد کرد و ممکن است برگردد

Error: No matching Packages to list

```
# dnf list package_name
```

خروجی نمونه:



اطلاعات بسته را با دستور dnf دریافت کنید با dnf

با این دستور می توانید اطلاعات دقیق هر بسته موجود در مخازن را مشاهده کنید .این دستور نام دقیق بسته را انتظار دارد.

```
# dnf info [Package_name]
```

خروجی دستور:

```
[root@rockylinux ~]# dnf info vim-common
Last metadata expiration check: 0:13:51 ago on Sat 18 Sep 2021 04:49:30 PM IST.
Installed Packages
Name         : vim-common
Epoch        : 2
Version      : 8.0.1763
Release      : 15.el8
Architecture : x86_64
Size         : 27 M
Source       : vim-8.0.1763-15.el8.src.rpm
Repository   : @System
From repo    : AppStream
Summary      : The common files needed by any version of the VIM editor
URL          : http://www.vim.org/
License      : Vim and MIT
Description  : VIM (VIsual editor iMproved) is an updated and improved version of the
             : vi editor.  Vi was the first real screen-based editor for UNIX, and is
             : still very popular.  VIM improves on vi by adding new features:
             : multiple windows, multi-level undo, block highlighting and more.  The
             : vim-common package contains files which every VIM binary will need in
             : order to run.
```

نصب مجدد بسته و وابستگی ها با dnf

گاهی اوقات، ممکن است هنگام استفاده از یک بسته خاص، خطاهایی رخ دهد .به جای حذف آن و نصب مجدد، می توانید مستقیماً یک بسته و وابستگی های آن را مجدداً نصب کنید.

```
# dnf reinstall [package_name]
```

دانلود بسته به جای نصب با dnf

همچنین می توانید بسته را به جای نصب در سیستم دانلود کنید .برای انجام این کار، می توانید از --downloadonly گزینه استفاده کنید و می توانید فهرست دانلود را با استفاده از --downloaddir=<DIR_PATH> گزینه مشخص کنید.

```
# dnf install package_name --downloadonly --downloaddir=DIR_PATH
```

خروجی دستور:

```
[root@rockylinux ~]#
[root@rockylinux ~]# dnf -q -y install iotop --downloadonly --downloaddir=/home/deepak
[root@rockylinux ~]#
[root@rockylinux ~]# ls -l /home/deepak/iotop-0.6-16.el8.noarch.rpm
-rw-r--r--. 1 root root 66728 Sep 18 17:04 /home/deepak/iotop-0.6-16.el8.noarch.rpm
[root@rockylinux ~]#
[root@rockylinux ~]# rpm -q iotop
package iotop is not installed
[root@rockylinux ~]#
```

لیست کردن مخازن فعال فعلی با dnf

برای چاپ لیست تمام مخازن فعال می توانید از دستور زیر استفاده کنید.

```
# dnf repolist
```

یا

```
# dnf repolist enabled
```

خروجی دستور:

```
[root@rockylinux ~]# dnf repolist
repo id                                 repo name
appstream                               Rocky Linux 8 - AppStream
baseos                                  Rocky Linux 8 - BaseOS
extras                                  Rocky Linux 8 - Extras
plus                                    Rocky Linux 8 - Plus
```

لیست کردن مخازن غیرفعال فعلی با dnf

برای دریافت لیست تمامی مخازن غیرفعال شده می توانید از گزینه disabled استفاده کنید.

```
# dnf repolist disabled
```

خروجی دستور:

```
[root@rockylinux ~]# dnf repolist disabled
repo id                                 repo name
appstream-source                        Rocky Linux 8 - AppStream - Source
baseos-source                           Rocky Linux 8 - BaseOS - Source
devel                                   Rocky Linux 8 - Devel WARNING! FOR BUILDROOT USE ONLY!
extras-source                           Rocky Linux 8 - Extras - Source
ha                                      Rocky Linux 8 - HighAvailability
ha-source                               Rocky Linux 8 - High Availability - Source
media-appstream                         Rocky Linux 8 - Media - AppStream
media-baseos                            Rocky Linux 8 - Media - BaseOS
```

لیست کردن تمام مخازن با dnf

همچنین می توانید تمام مخازنی که روی سیستم شما پیکربندی شده اند را فهرست کنید.

```
# dnf repolist all
```

خروجی دستور:

```
[root@rockylinux ~]# dnf repolist all
repo id                                 repo name                                       status
appstream                               Rocky Linux 8 - AppStream                       enabled
appstream-source                        Rocky Linux 8 - AppStream - Source              disabled
baseos                                  Rocky Linux 8 - BaseOS                          enabled
baseos-source                           Rocky Linux 8 - BaseOS - Source                 disabled
devel                                   Rocky Linux 8 - Devel WARNING! FOR BUILDROOT USE ONLY!  disabled
extras                                  Rocky Linux 8 - Extras                          enabled
extras-source                           Rocky Linux 8 - Extras - Source                 disabled
ha                                      Rocky Linux 8 - HighAvailability                disabled
ha-source                               Rocky Linux 8 - High Availability - Source      disabled
media-appstream                         Rocky Linux 8 - Media - AppStream               disabled
media-baseos                            Rocky Linux 8 - Media - BaseOS                  disabled
plus                                    Rocky Linux 8 - Plus                            enabled
plus-source                             Rocky Linux 8 - Plus - Source                   disabled
```

فعال کردن مخزن خاص با dnf

با استفاده از enablerepo--گزینه می توانید بسته ای را از یک مخزن خاص نصب کنید .همچنین می تواند یک مخزن غیرفعال را برای نصب یک بسته فعال کند.

```
# dnf install package_name --enablerepo=repo_name
```

یا شما می توانید تمام مخازن را غیرفعال کنید و تنها از یک مخزن استفاده کنید:

```
# dnf install package_name -disablerepo=* --enablerepo=repo_name
```

به عنوان مثال ، من `vsftpd` بسته `appstream` را فقط با استفاده از repoنصب می کنم و تمام مخزن های دیگر را غیرفعال می کنم.

غیرفعال کردن مخزن خاص با dnf

disablerepo--به شما اجازه می دهد تا مخزن را برای اجرای زمان اجرا غیرفعال کنید dnf .را برای نصب بسته ای از آن مخزن محدود می کند .اگر بسته در سایر مخازن فعال موجود نباشد، ممکن است نتوانید بسته را نصب کنید.

```
# yum install package_name --disablerepo=repo_name
```

پیدا کردن فایل خاص یک بسته با dnf

با استفاده از این دستور می توانید متوجه شوید که کدام بسته ها حاوی یک فایل خاص هستند.

```
# dnf provides file_name
```

یا

```
# dnf whatprovides file_name
```

```
[root@rockylinux ~]# dnf provides python3
Last metadata expiration check: 0:14:51 ago on Sat 18 Sep 2021 05:04:05 PM IST.
python36-3.6.8-2.module+el8.3.0+120+426d8baf.x86_64 : Interpreter of the Python programming language
Repo        : @System
Matched from:
Provide    : python3 = 3.6.8-2.module+el8.3.0+120+426d8baf

python36-3.6.8-2.module+el8.3.0+120+426d8baf.x86_64 : Interpreter of the Python programming language
Repo        : appstream
Matched from:
Provide    : python3 = 3.6.8-2.module+el8.3.0+120+426d8baf
```

همانطور که می بینید، نام RPM را در اختیار ما قرار می دهد که مسئول نصب باینری python3 است .ممکن است چند فایل دیگر با همین نام وجود داشته باشد .برای دریافت لیست کامل فایل ها با نام python3، می توانید از regex با دستور dnf استفاده کنید.

```
[root@rockylinux ~]# dnf provides */python3
Last metadata expiration check: 0:16:08 ago on Sat 18 Sep 2021 05:04:05 PM IST.
bash-completion-1:2.7-5.el8.noarch : Programmable completion for Bash
Repo        : @System
Matched from:
Filename    : /usr/share/bash-completion/completions/python3

bash-completion-1:2.7-5.el8.noarch : Programmable completion for Bash
Repo        : baseos
Matched from:
Filename    : /usr/share/bash-completion/completions/python3

freeradius-3.0.20-3.module+el8.4.0+576+2d493411.x86_64 : High-performance and highly configurable
                                                       : free RADIUS server
Repo        : appstream
Matched from:
Filename    : /etc/raddb/mods-available/python3
```

پیدا کردن وابستگی های بسته با dnf

شما می توانید از deplist استفاده کنید .دستور فهرست کردن وابستگی های بسته و بسته هایی که آنها را ارائه می دهد.

```
# dnf deplist package_name
```

به عنوان مثال ، برای دریافت لیستی از rpm های وابسته مورد نیاز برای نصب `nmap` بسته، می توانیم از دستور زیر استفاده کنیم:

```
nmap dnf deplist
```

خروجی دستور:

```
[root@rockylinux ~]# dnf deplist nmap
Last metadata expiration check: 0:19:46 ago on Sat 18 Sep 2021 05:04:05 PM IST.
package: nmap-2:7.70-5.el8.x86_64
  dependency: libc.so.6(GLIBC_2.15)(64bit)
   provider: glibc-2.28-151.el8.x86_64
  dependency: libcrypto.so.1.1()(64bit)
   provider: openssl-libs-1:1.1.1g-15.el8_3.x86_64
  dependency: libcrypto.so.1.1(OPENSSL_1_1_0)(64bit)
   provider: openssl-libs-1:1.1.1g-15.el8_3.x86_64
  dependency: libdl.so.2()(64bit)
   provider: glibc-2.28-151.el8.x86_64
  dependency: libdl.so.2(GLIBC_2.2.5)(64bit)
   provider: glibc-2.28-151.el8.x86_64
  dependency: libgcc_s.so.1()(64bit)
   provider: libgcc-8.4.1-1.el8.x86_64
  dependency: libgcc_s.so.1(GCC_3.0)(64bit)
```

لیست کردن تمام بسته های گروه با dnf

بسته گروهی مجموعه ای از بسته های متعدد در یک گروه خاص است .می توانید از دستور زیر برای لیست کردن تمام گروه های موجود برای سیستم خود استفاده کنید.

```
# dnf grouplist
```

```
[root@rockylinux ~]# dnf grouplist
Last metadata expiration check: 0:21:23 ago on Sat 18 Sep 2021 05:04:05 PM IST.
Available Environment Groups:
   Server
   Minimal Install
   Workstation
   Virtualization Host
   Custom Operating System
Installed Environment Groups:
   Server with GUI
Installed Groups:
   Container Management
   Headless Management
Available Groups:
   .NET Core Development
   RPM Development Tools
   Development Tools
```

نصب بسته گروهی با dnf

هنگامی که یک بسته گروهی را نصب می کنید، تمام بسته های داخل گروه نصب می شوند .برای نصب بسته گروهی می توانید از این دستور استفاده کنید.

```
# dnf install group_package
```

حذف بسته گروهی با dnf

هنگامی که یک بسته گروهی را حذف می کنید، تمام بسته های متعلق به آن گروه نیز حذف می شوند.

```
# dnf group_package remove
```

بروزرسانی بسته گروهی با dnf

برای به روز رسانی یک بسته گروهی نصب شده در سیستم خود می توانید از این دستور استفاده کنید.

```
# dnf update  groupupdate group_package
```

مشاهده تاریخچه تمام دستورات dnf اجرا شده

با استفاده از این دستور می توانید اطلاعات مربوط به آنچه در اجرای قبلی دستور dnf رخ داده است را مشاهده کنید.

```
# dnf  history
```

```
[root@rockylinux ~]# dnf history
ID     | Command line                                | Date and time    | Action(s)      | Altered
-------------------------------------------------------------------------------------------------
     5 | install whois --disablerepo=baseos -y       | 2021-09-18 17:16 | Install        |    2
     4 | install vsftpd --disablerepo=* --enablerepo=ap | 2021-09-18 17:12 | Install        |    1
     3 | -y install nss-tools                        | 2021-08-12 10:51 | Install        |    1
     2 | -y install openldap-servers openldap-clients | 2021-08-12 10:17 | I, U           |    3 EE
     1 |                                             | 2021-08-12 09:31 | Install        | 1384 EE
```

لغو و انجام مجدد تغییرات با dnf

می توانید تغییرات ایجاد شده با اجرای دستور dnf را لغو و دوباره انجام دهید .شما به یک شناسه تراکنش نیاز دارید که می توانید از دستور history دریافت کنید.

برای واگرد تغییر، می توانید از موارد زیر استفاده کنید:

```
#  dnf undo transaction_id
```

لغو تمام تراکنش ها پس از تراکنش مشخص شده با dnf

همچنین می توانید با استفاده از دستور rollback با dnf تمام تراکنش ها را پس از تراکنش های مشخص شده لغو کنید .به عنوان مثال ، اگر شما ۱۵ تراکنش در تاریخچه داشته باشید و شناسه تراکنش ۱۰ را مشخص کنید، تمام تراکنش های بعد از ۱۰ یعنی ۱۱، ۱۲، ۱۳، ۱۴، ۱۵ را خنثی می کند.

```
# dnf hisroty rollback transaction_id
```

شما باید با احتیاط از dnf rollback استفاده کنید زیرا تمام تراکنش‌های بین شناسه تراکنش فعلی و ارائه‌شده را برمی‌گرداند .در سناریوهای پیچیده که چندین تراکنش وجود دارد، دستور dnf در انجام بازگشت ناموفق خواهد بود.

پاک کردن تمام حافظه کش مدیربسته با dnf

برای حذف تمام کش های ذخیره شده توسط dnf می توانید از دستور زیر استفاده کنید.

```
# dnf clean all
```

ورود به پوسته dnf

شما می توانید از پوسته dnf تعاملی استفاده کنید که در آن می توانید تمام دستورات dnf را اجرا کنید.

```
# dnf shell
```

مدیر بسته pacman برای آرچ بیس ها

نصب یک یا چند بسته با pacman

```
pacman -S    package_name1    package_name2
```

نصب بسته از مخزن خاص با pacman

```
pacman -S extra/package_name
```

حذف یک بسته نرم افزاری با pacman

```
pacman -R package_name
```

حذف یک بسته نرم افزاری با تمام وابستگی ها با pacman

```
pacman -Rs package_name
```

آپدیت سیستم از مخازن با pacman

```
pacman -Syu
```

جستجوی یک بسته نرم افزاری در مخازن با pacman

```
pacman -Ss   uget
```

جستجوی بسته نصب شده روی سیستم عامل با pacman

```
pacman -Qs uget
```

لیست فایل های نصب شده به همراه یک بسته با pacman

```
pacman -Ql   package_name
```

مشاهده لیست تمام بسته های نصب شده با pacman

```
pacman -Qdt
```

مشاهده لیست برنامه های نصب شده به همراه برنامه های وابسته با pacman

```
pactree    package_name
```

دانلود بسته بدون نصب شدن با pacman

```
pacman -Sw package_name
```

حذف بسته های دانلود شده با pacman

بسته ها در مسیر /var/cache/pacman/pkg/ قرار دارند.

```
pacman -Sc
paccache -r
```

مدیر بسته dpkg

dpkgیک ابزار خط فرمان است که برای نصب، حذف، ساخت و مدیریت بسته‌های دبیان (.deb) به کار می‌رود .این دستور به عنوان پایه و اساس سیستم مدیریت بسته در توزیع‌های مبتنی بر دبیان مانند اوبونتو، دبیان و بسیاری دیگر عمل می‌کند.

ساختار دستور:

```
dpkg [options] action package
```

- **options:**گزینه‌های مختلفی که رفتار دستور را تغییر می‌دهند.
- **action:**عملیاتی که قرار است روی بسته انجام شود (مثلاً نصب، حذف، ساخت و...).
- **package:**نام بسته‌ای که می‌خواهید روی آن عمل انجام شود.

کاربردهای دستور dpkg:

- **نصب بسته‌ها :**با استفاده از گزینه `i-`یا `--install`می‌توانید بسته‌های deb. را نصب کنید.
- **حذف بسته‌ها :**با استفاده از گزینه `r-`یا `--remove`می‌توانید بسته‌های نصب شده را حذف کنید.
- **ساخت بسته‌ها :**با استفاده از گزینه `b-`یا `--build`می‌توانید از یک دایرکتوری منبع، یک بسته deb. بسازید.
- **نمایش اطلاعات در مورد بسته‌ها :**با استفاده از گزینه‌های مختلف می‌توانید اطلاعاتی مانند نسخه، وابستگی‌ها و فایل‌های تشکیل‌دهنده یک بسته را مشاهده کنید.
- **بروزرسانی پایگاه داده بسته‌ها :**با استفاده از گزینه `--configure`می‌توانید پایگاه داده بسته‌ها را بروزرسانی کنید.

گزینه‌های کامل دستور:

برای مشاهده لیست کامل گزینه‌ها و توضیحات آن‌ها می‌توانید از دستور `man dpkg`در ترمینال استفاده کنید .اما در اینجا برخی از مهم‌ترین گزینه‌ها را معرفی می‌کنیم:

- **-i, --install:**نصب بسته
- **-r, --remove:**حذف بسته
- **-b, --build:**ساخت بسته
- **-l, --list:**لیست تمام بسته‌های نصب شده
- **-s, --status:**نمایش وضعیت یک بسته
- **-c, --configure:**پیکربندی مجدد بسته
- **-P, --purge:**حذف کامل یک بسته به همراه فایل‌های پیکربندی
- **-L, --listfiles:**نمایش لیست فایل‌های یک بسته
- **-S, --search:**جستجوی بسته بر اساس نام یا توضیحات

توضیحات گزینه‌های اضافی دستور dpkg

نصب و باز کردن بسته

- **-i, --install package-file...:**این گزینه برای نصب بسته‌های deb. استفاده می‌شود .می‌توانید چندین فایل بسته را به عنوان آرگومان مشخص کنید.
- **--unpack package-file...:**این گزینه باعث باز کردن بسته می‌شود بدون اینکه آن را پیکربندی کند .اگر گزینه‌های `--recursive`یا `R-`نیز استفاده شده باشند، باید به جای فایل بسته، یک دایرکتوری مشخص کنید.

پیکربندی بسته

- **--configure package...|-a|--pending:**این گزینه برای پیکربندی بسته‌های نصب شده استفاده می‌شود .
- گزینه‌های `a-`و `--pending`برای پیکربندی تمام بسته‌های منتظر استفاده می‌شوند.

حذف بسته

- **r, --remove, -P, --purge package...|-a|--pending:** این گزینه‌ها برای حذف بسته‌ها استفاده می‌شوند . تفاوت بین r- و P- در این است که P- علاوه بر حذف بسته، فایل‌های پیکربندی آن را نیز حذف می‌کند .گزینه‌های a- و pending-- برای حذف تمام بسته‌های نصب شده استفاده می‌شوند.

بررسی و مدیریت بسته‌ها

- **V, --verify [package-name...]:** این گزینه برای بررسی یکپارگی بسته استفاده می‌شود .می‌توانید نام بسته خاصی را مشخص کنید یا بدون آرگومان برای بررسی تمام بسته‌ها استفاده کنید.
- **--update-avail, --merge-avail Packages-file:** این گزینه‌ها برای به‌روزرسانی اطلاعات در مورد بسته‌های موجود استفاده می‌شوند .فایل Packages-file حاوی اطلاعات مربوط به بسته‌ها است.
- **A, --record-avail package-file...:** این گزینه برای ثبت اطلاعات در مورد بسته‌های موجود استفاده می‌شود .فایل package-file حاوی اطلاعات مربوط به بسته‌ها است.
- **--forget-old-unavail:** این گزینه برای حذف اطلاعات قدیمی در مورد بسته‌های ناموجود استفاده می‌شود.
- **--clear-avail:** این گزینه برای پاک کردن تمام اطلاعات در مورد بسته‌های موجود استفاده می‌شود.
- **C, --audit:** این گزینه برای بررسی مشکلات بسته‌ها استفاده می‌شود.
- **--get-selections [package-name-pattern...]:** این گزینه برای دریافت لیستی از بسته‌های انتخاب شده استفاده می‌شود .می‌توانید یک الگوی نام بسته را نیز مشخص کنید.
- **--set-selections:** این گزینه برای تنظیم انتخاب‌های بسته استفاده می‌شود.
- **--clear-selections:** این گزینه برای پاک کردن انتخاب‌های بسته استفاده می‌شود.
- **--yet-to-unpack:** این گزینه برای لیست کردن بسته‌های منتظر باز شدن استفاده می‌شود.

مدیریت معماری

- **--add-architecture architecture:** این گزینه برای افزودن یک معماری جدید به لیست معماری‌های پشتیبانی شده استفاده می‌شود.
- **--remove-architecture architecture:** این گزینه برای حذف یک معماری از لیست معماری‌های پشتیبانی شده استفاده می‌شود.
- **--print-architecture:** این گزینه برای چاپ معماری بسته‌هایی که dpkg نصب می‌کند استفاده می‌شود.
- **--print-foreign-architectures:** این گزینه برای چاپ لیستی از معماری‌های خارجی که dpkg برای نصب بسته‌ها مجاز است استفاده می‌شود.

سایر گزینه‌ها

- **--compare-versions ver1 op ver2:** این گزینه برای مقایسه نسخه‌های نرم‌افزار استفاده می‌شود.
- **?, --help:** این گزینه برای نمایش پیام راهنما استفاده می‌شود.
- **--force-help:** این گزینه برای نمایش اطلاعات بیشتر در مورد گزینه‌های force-- استفاده می‌شود.
- **Dh, --debug=help:** این گزینه برای نمایش اطلاعات در مورد گزینه‌های دیباگ استفاده می‌شود.

- **version:--**این گزینه برای نمایش نسخه dpkg استفاده می‌شود.

مثال‌ها:

- **نصب بسته vim:**

```
sudo dpkg -i vim.deb
```

- **حذف بسته firefox:**

```
sudo dpkg -r firefox
```

- **نمایش لیست تمام بسته‌های نصب شده:**

```
dpkg -l
```

- **نمایش اطلاعات در مورد بسته apache2:**

```
dpkg -s apache2
```

نکات مهم:

- برای استفاده از دستور dpkgمعمولاً نیاز به دسترسی روت (sudo) دارید.
- dpkgیک ابزار سطح پایین است و برای مدیریت پیچیده بسته‌ها، ابزارهایی مانند aptitude وaptکه بر روی dpkgساخته شده‌اند، مناسب‌تر هستند.
- قبل از حذف یا نصب هر بسته‌ای، بهتر است از وضعیت سیستم پشتیبان تهیه کنید.

دستور dpkg-reconfigure:

دستور dpkg-reconfigure ابزاری قدرتمند برای پیکربندی مجدد بسته‌های نرم‌افزاری نصب شده در سیستم‌های مبتنی بر دبیان است .با استفاده از این دستور، می‌توان تنظیمات مختلف یک بسته را پس از نصب یا در هر زمان دیگری تغییر داد.

ساختار دستور:

```
dpkg-reconfigure package
```

- **package:**نام بسته‌ای که می‌خواهید پیکربندی مجدد کنید.

کاربردهای دستور:

- **تغییر تنظیمات پیش‌فرض :**بسیاری از بسته‌ها دارای تنظیمات پیش‌فرضی هستند که ممکن است برای همه کاربران مناسب نباشد .با استفاده از dpkg-reconfigureمی‌توانید این تنظیمات را مطابق با نیاز خود تغییر دهید.
- **فعال یا غیرفعال کردن ویژگی‌ها :**برخی از بسته‌ها دارای ویژگی‌های اضافی هستند که می‌توان آن‌ها را فعال یا غیرفعال کرد.
- **تغییر پیکربندی سرویس‌ها :**برای سرویس‌هایی که توسط بسته‌ها ارائه می‌شوند، می‌توان تنظیمات سرویس را تغییر داد.
- **رفع مشکلات پیکربندی :**اگر با مشکلی در پیکربندی یک بسته مواجه شدید، می‌توانید با استفاده از dpkg-reconfigureآن را برطرف کنید.

گزینه‌های رایج:

- **بدون گزینه :**اگر بدون هیچ گزینه‌ای از dpkg-reconfigureاستفاده کنید، یک رابط متنی تعاملی برای پیکربندی بسته باز می‌شود.

- **--force-yes:** این گزینه باعث می‌شود که تمام سوالات پیکربندی با پاسخ "بله" پاسخ داده شوند .با احتیاط از این گزینه استفاده کنید.
- **--force-no:** این گزینه باعث می‌شود که تمام سوالات پیکربندی با پاسخ "خیر" پاسخ داده شوند.
- **--edit-config:** این گزینه به شما اجازه می‌دهد فایل پیکربندی بسته را با یک ویرایشگر متن باز کنید.

مثال‌ها:

- **پیکربندی مجدد سرور پایگاه داده:MySQL**

```
sudo dpkg-reconfigure mysql-server
```

- **پیکربندی مجدد سرور وب:Apache**

```
sudo dpkg-reconfigure apache2
```

- **پیکربندی مجدد بسته:Vim**

```
sudo dpkg-reconfigure vim
```

نکات مهم:

- برای اجرای `dpkg-reconfigure` معمولاً نیاز به دسترسی روت (sudo) دارید.
- قبل از پیکربندی مجدد یک بسته، بهتر است از تنظیمات فعلی آن نسخه پشتیبان تهیه کنید.
- برخی از بسته‌ها ممکن است اسکریپت‌های پیکربندی پیچیده‌ای داشته باشند و تغییر تنظیمات آن‌ها نیاز به دانش فنی داشته باشد.
- در صورت بروز هرگونه مشکل پس از پیکربندی مجدد، می‌توانید از گزینه `--edit-config` برای بررسی و اصلاح فایل پیکربندی استفاده کنید.

نصب میزهای کارلینوکسی

میزهای معروف و پراستفاده لینوکسی

```
GNOME
  Plasma Desktop (KDE)
  Pantheon
  XFCE
  Cinnamon
  Mate
  Trinity (TDE)
  Budgie
  Windowmaker
  Enlightenment
  Fluxbox
  Openbox
  TWM
  FVWM
  Lx{QT,DE}
  PekWM
  JWM
  Ratpoison
  HerbsluftWM
  Lumina
  UDE
  Rox
  Equinox
```

یکی از نقاط قوت دسکتاپ لینوکس، انتخابی است که برای کاربرانش فراهم می کند .اگر منوی برنامه خود را در پایین سمت چپ صفحه خود دوست ندارید، می توانید آن را جابجا کنید .اگر روشی را که مدیر فایلتان اسنادتان را سازماندهی می‌کند دوست ندارید، می‌توانید از یکی دیگر استفاده کنید، با این حال، مسلماً این می تواند برای کاربران جدیدی که عادت ندارند در مورد نحوه استفاده از رایانه خود نظر بدهند، گیج کننده باشد .اگر به دنبال نصب لینوکس هستید، یکی از گزینه‌هایی که باید انجام دهید این است که از کدام دسکتاپ می‌خواهید استفاده کنید، و بهترین راه برای انجام این کار این است که چند مورد مختلف را امتحان کنید تا زمانی که مورد احساس را پیدا کنید.

محیط های مختلف دسکتاپ

میز MATE

تاریخچه:

MATE یک محیط دسکتاپ متن‌باز و آزاد برای سیستم‌عامل‌های لینوکسی است که ابتدا برگرفته از GNOME 2 شد و با هدف ارائه تجربه کلاسیک و ساده برای کاربران توسعه یافت. توسعه MATE در سال ۲۰۱۱ آغاز شد و پس از توقف توسعه GNOME 2 و انتقال به GNOME 3، MATE به عنوان یک جایگزین محبوب برای کاربرانی که به دنبال تجربه کلاسیک GNOME 2 بودند، معرفی شد.

ویژگی‌ها:

کلاسیک و ساده: MATE طراحی شده است تا تجربه کاربری کلاسیک و ساده را فراهم کند، شامل منوها و پنل‌های سنتی.

سبک و سریع: MATE یک محیط دسکتاپ سبک است که به روال سیستم‌های کم توان و قدیمی نیز مناسب است و عملکرد سریعی را ارائه می‌دهد.

پشتیبانی از افزونه‌ها: MATE امکان نصب و استفاده از افزونه‌ها و پوسته‌های مختلف را فراهم می‌کند تا کاربران بتوانند ظاهر و عملکرد محیط دسکتاپ را دلخواه خود تغییر دهند.

پشتیبانی از ویجت‌ها: MATE از انواع ویجت‌ها برای سفارشی‌سازی دسکتاپ پشتیبانی می‌کند و به کاربران امکان می‌دهد تا دسکتاپ خود را به دلخواه خود تنظیم کنند.

پشتیبانی از ابزارهای مدیریتی: MATE شامل ابزارهای مدیریتی گوناگون مانند مدیریت فایل، تنظیمات سیستم و شبکه است که کاربران می‌توانند از آن‌ها برای مدیریت سیستم خود استفاده کنند.

مزایا:

- ارائه تجربه کاربری کلاسیک و ساده
- عملکرد سریع و سبک
- قابلیت سفارشی‌سازی بالا
- پشتیبانی از افزونه‌ها و ویجت‌ها
- دارای ابزارهای مدیریتی کارآمد



میز Cinnamon

تاریخچه:

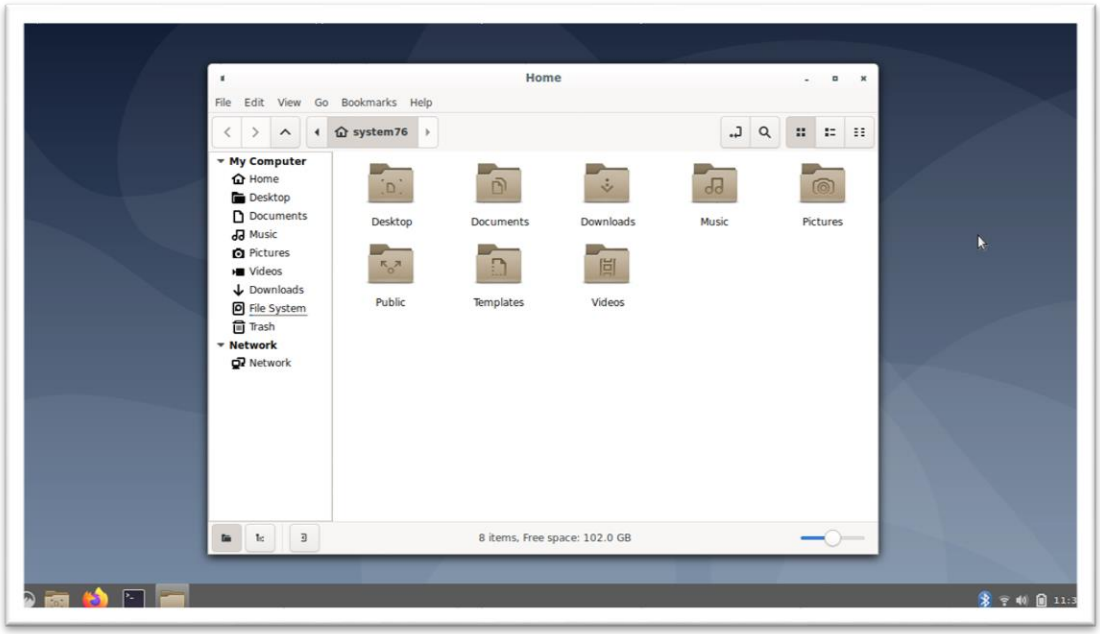
Cinnamon یک محیط دسکتاپ متن‌باز برای سیستم‌عامل‌های لینوکسی است که توسط تیم توسعه‌دهنده‌های محبوب ویندوز Mint توسعه یافته است. این محیط دسکتاپ ابتدا در نسخه ۱۲ از Mint معرفی شد و با هدف ایجاد یک تجربه کاربری زیبا، ساده و کاربرپسند برای کاربران توسعه یافت.

ویژگی‌ها:

ویندوزپسند: Cinnamon طراحی شده است تا به نحوی شبیه به ویندوز باشد که کاربران ویندوز به راحتی می‌توانند به آن عادت کنند.
زیبایی و طراحی مدرن: Cinnamon دارای ظاهری زیبا و مدرن است که با امکانات سفارشی‌سازی بالا، کاربران می‌توانند ظاهر دسکتاپ خود را به دلخواه تغییر دهند.
عملکرد قوی: Cinnamon علاوه بر ظاهر زیبا، عملکرد سریع و قدرتمندی را نیز ارائه می‌دهد که به کاربران امکان استفاده روان از سیستم را می‌دهد.
پشتیبانی از افزونه‌ها و اپلیکیشن‌های زیاد: Cinnamon قابلیت نصب و استفاده از افزونه‌ها و اپلیکیشن‌های مختلف را فراهم می‌کند تا کاربران بتوانند تجربه دسکتاپ خود را بهبود بخشند.
امکانات مدیریتی: Cinnamon دارای ابزارهای مدیریتی متنوعی مانند مدیریت فایل، تنظیمات سیستم و تنظیمات دسکتاپ است که به کاربران کمک می‌کند تا سیستم خود را بهتر مدیریت کنند.

مزایا:

- ظاهر زیبا و مدرن
- امکانات سفارشی‌سازی بالا
- عملکرد سریع و قدرتمند
- پشتیبانی از افزونه‌ها و اپلیکیشن‌های زیاد
- ابزارهای مدیریتی کارآمد



میز GNOME

تاریخچه

GNOME یک محیط دسکتاپ متن‌باز برای سیستم‌عامل‌های لینوکسی است که ابتدا در سال ۱۹۹۹ توسط پروژه GNOME توسعه یافت. هدف اصلی این محیط دسکتاپ ارائه یک تجربه کاربری ساده، زیبا و کارآمد برای کاربران لینوکسی بوده است.

ویژگی‌ها

طراحی ساده و کاربرپسند: GNOME با طراحی ساده و کاربرپسند خود، به کاربران امکان استفاده راحت و سریع از سیستم را می‌دهد.

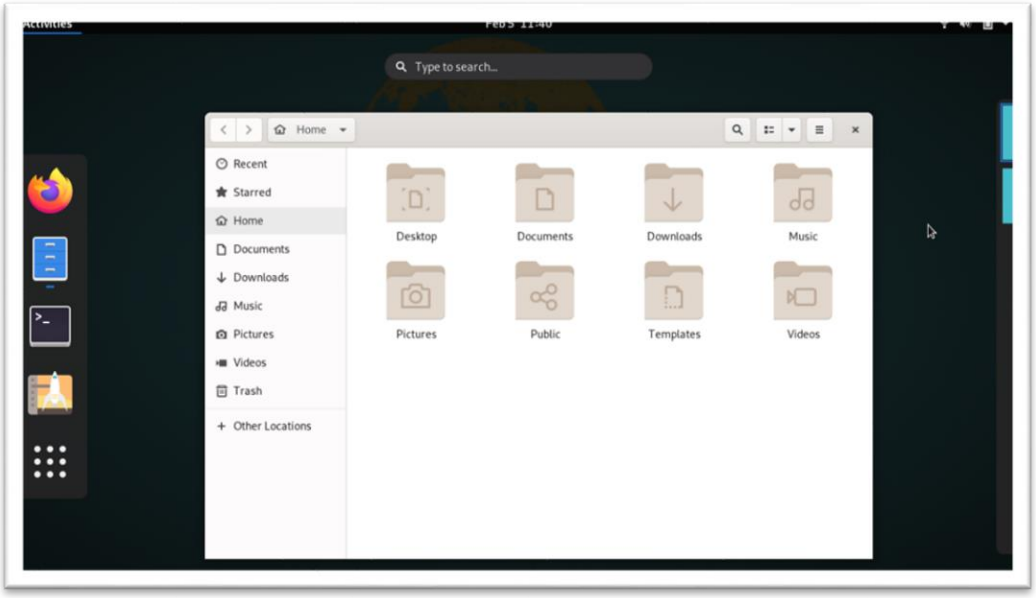
امکانات گسترده: GNOME دارای امکانات گسترده‌ای از جمله مدیریت فایل، تنظیمات سیستم، ابزارهای دسترسی و ابزارهای دیگر است که به کاربران امکانات کاملی را ارائه می‌دهد.

پشتیبانی از افزونه‌ها: GNOME قابلیت نصب و استفاده از افزونه‌ها را دارد که کاربران می‌توانند با اضافه کردن این افزونه‌ها، تجربه دسکتاپ خود را بهبود بخشند.

امکانات چندرسانه‌ای: GNOME دارای امکانات چندرسانه‌ای مانند پخش صدا و ویدیو، مدیریت عکس و ویدیو و امکانات دیگر است که به کاربران امکان استفاده آسان از این امکانات را می‌دهد.

مزایا

- طراحی ساده و کاربرپسند
- امکانات گسترده
- پشتیبانی از افزونه‌ها
- امکانات چندرسانه‌ای



میز پلاسما

تاریخچه

Plasma یک محیط دسکتاپ متن‌باز برای سیستم‌عامل‌های لینوکسی است که توسط KDE توسعه یافته است. این محیط دسکتاپ ابتدا در سال ۲۰۰۸ معرفی شد و به مرور زمان با بهبود و توسعه مستمر، تبدیل به یکی از محبوب‌ترین محیط‌های دسکتاپ برای کاربران لینوکسی شده است.

ویژگی‌ها

چیدمان انعطاف پذیر: Plasma دارای چیدمان انعطاف پذیری است که به کاربران امکان تنظیم دقیق و شخصی‌سازی ظاهر دسکتاپ را می‌دهد.

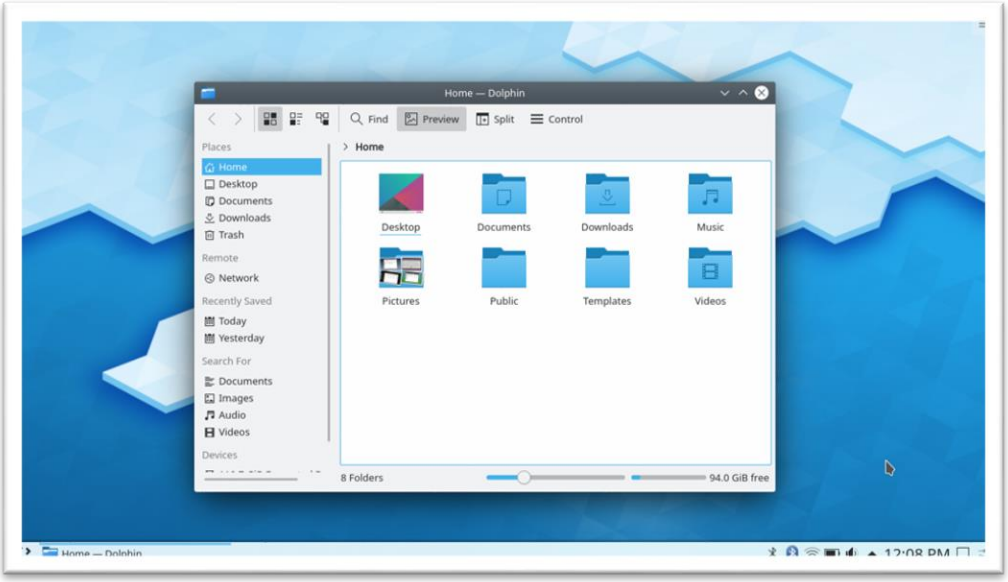
امکانات قدرتمند: Plasma دارای امکانات بسیار قدرتمندی از جمله مدیریت فایل، ابزارهای دیتابیس، ابزارهای توسعه و امکانات دیگر است که به کاربران امکانات کاملی را ارائه می‌دهد.

پشتیبانی از ویدجت‌ها: Plasma از ویدجت‌ها پشتیبانی می‌کند که به کاربران امکان اضافه کردن ابزارهای کاربردی و دلخواه به دسکتاپ را می‌دهد.

امکانات چندرسانه‌ای: Plasma دارای امکانات چندرسانه‌ای مانند پخش صدا و ویدیو، مدیریت عکس و ویدیو و امکانات دیگر است که تجربه کاربری را بهبود می‌بخشد.

مزایا

- چیدمان انعطاف پذیر
- امکانات قدرتمند
- پشتیبانی از ویدجت‌ها
- امکانات چندرسانه‌ای



میز UKUI

تاریخچه:

UKUI یک محیط دسکتاپ متن‌باز برای سیستم‌عامل‌های لینوکسی است که توسط چین در سال ۲۰۱۵ توسعه یافت. این محیط دسکتاپ بر اساس محیط دسکتاپ MATE بنا شده است و به منظور ارائه تجربه کاربری بهتر برای کاربران چینی طراحی شده است.

ویژگی‌ها:

طراحی ساده و زیبا: UKUI دارای طراحی ساده و زیبا است که باعث ارائه یک تجربه کاربری دلپذیر برای کاربران می‌شود.

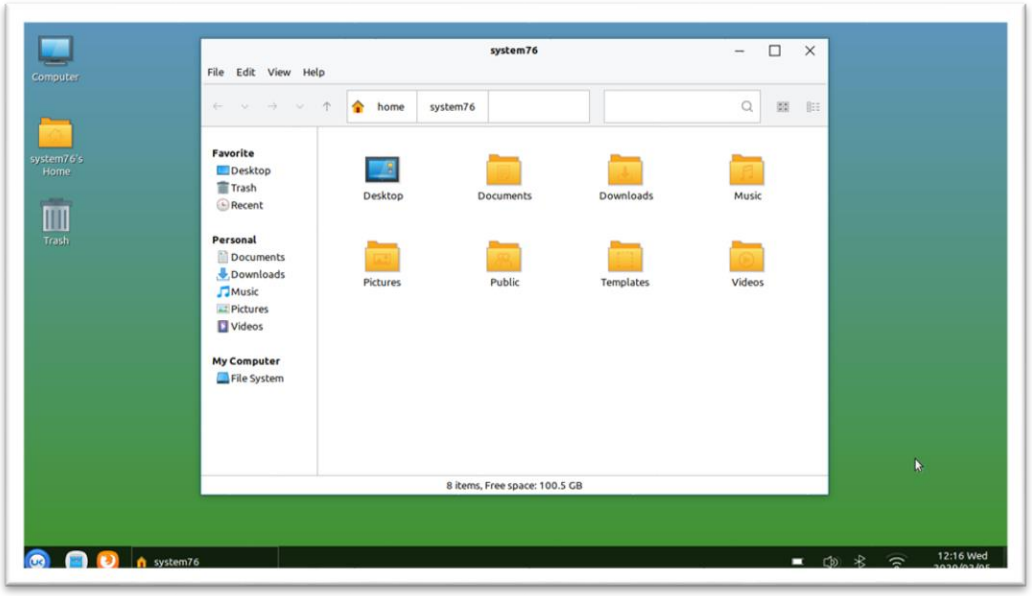
پشتیبانی از زبان چینی: UKUI پشتیبانی کامل از زبان چینی را داراست که به کاربران چینی امکان استفاده راحت و آسان از این محیط دسکتاپ را می‌دهد.

امکانات چندرسانه‌ای: UKUI دارای امکانات چندرسانه‌ای مانند پخش صدا و ویدیو، مدیریت عکس و ویدیو و امکانات دیگر است که تجربه کاربری را بهبود می‌بخشد.

سازگاری با ابزارهای چینی: UKUI به خوبی با ابزارها و سرویس‌های محبوب در چین سازگاری دارد که به کاربران این منطقه امکان استفاده بهتر از این محیط دسکتاپ را می‌دهد.

مزایا:

- طراحی ساده و زیبا
- پشتیبانی از زبان چینی
- امکانات چندرسانه‌ای
- سازگاری با ابزارهای چینی



میز XFCE

تاریخچه

Xfce یک محیط دسکتاپ متن‌باز برای سیستم‌عامل‌های لینوکسی است که ابتدا در سال ۱۹۹۶ توسط Olivier Fourdan توسعه یافت. این محیط دسکتاپ با تمرکز بر کارایی و سبکی طراحی شده و به مرور زمان توانسته است جایگاه خود را به عنوان یکی از محیط‌های دسکتاپ محبوب در دنیای لینوکسی تثبیت کند.

ویژگی‌ها

سبکی و کارایی: Xfce به دلیل سبکی و کارایی بالا، بر روی سیستم‌های با سخت‌افزار کمتر نیز به خوبی عمل می‌کند و تجربه کاربری سریع و پرقدرتی را ارائه می‌دهد.

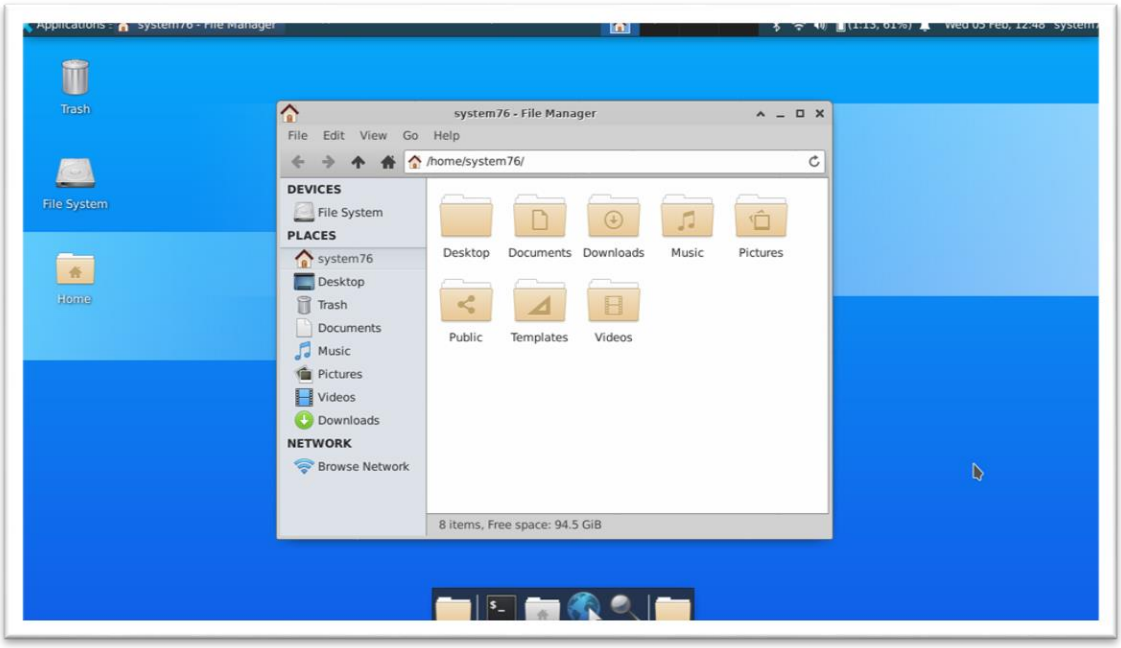
پیکربندی انعطاف پذیر: Xfce دارای امکانات پیکربندی بسیار زیادی است که به کاربران امکان شخصی‌سازی دسکتاپ خود را می‌دهد.

پشتیبانی از افزونه‌ها: Xfce از افزونه‌ها و ویجت‌ها پشتیبانی می‌کند که به کاربران امکان اضافه کردن ابزارها و ویژگی‌های جدید به دسکتاپ را می‌دهد.

استقرار آسان: Xfce به دلیل سبکی خود، به راحتی قابل استقرار بر روی انواع سیستم‌های لینوکسی است و نیاز به منابع سخت‌افزاری بالا ندارد.

مزایا

- ✓ سبکی و کارایی بالا
- ✓ پیکربندی انعطاف پذیر
- ✓ پشتیبانی از افزونه‌ها
- ✓ استقرار آسان



میز LXDE

تاریخچه

LXDE یک محیط دسکتاپ سبک و سریع برای سیستم‌عامل‌های لینوکسی است که ابتدا در سال ۲۰۰۶ توسط Hong Jen Yee توسعه یافت. LXDE به منظور ارائه یک محیط دسکتاپ سبک با مصرف کم منابع سخت‌افزاری طراحی شده و به مرور زمان توانسته است جایگاه خود را در جامعه لینوکسی برجسته کند.

ویژگی‌ها

سبکی و سرعت: LXDE به دلیل طراحی سبک و کم‌مصرف منابع، بسیار سریع و پرقدرت است و بر روی سیستم‌های قدیمی و با منابع محدود نیز به خوبی عمل می‌کند.

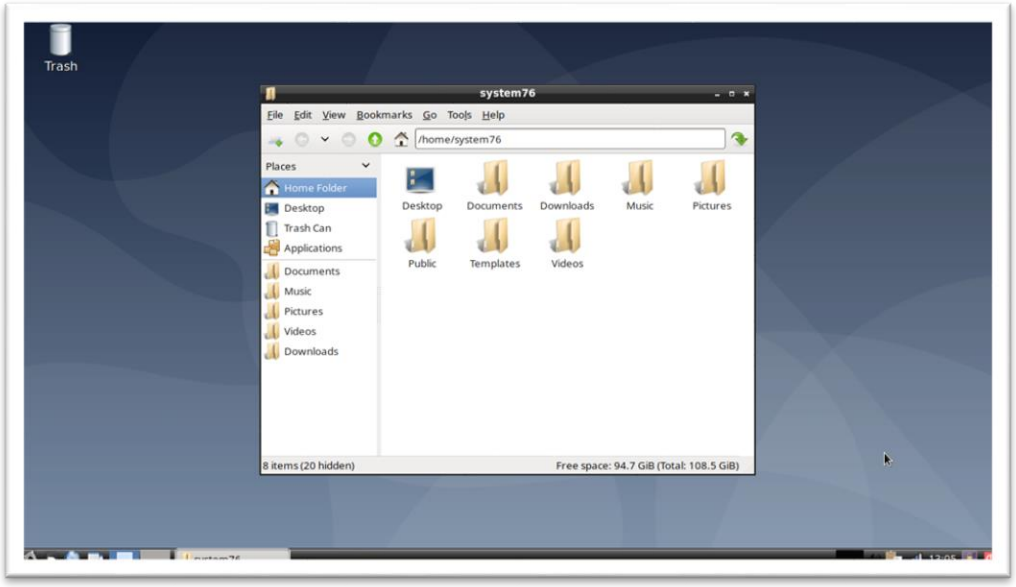
سادگی: LXDE دارای رابط کاربری ساده و آسان است که امکان استفاده راحت و سریع از آن را برای کاربران فراهم می‌کند.

پیکربندی انعطاف‌پذیر: LXDE امکان پیکربندی بسیار زیادی را به کاربران می‌دهد تا دسکتاپ خود را به دلخواه شخصی‌سازی کنند.

پشتیبانی از افزونه‌ها: LXDE از افزونه‌ها و ویجت‌ها پشتیبانی می‌کند که به کاربران امکان اضافه کردن امکانات و ویژگی‌های جدید به دسکتاپ را می‌دهد.

مزایا

- ✓ سبکی و سرعت بالا
- ✓ سادگی رابط کاربری
- ✓ پیکربندی انعطاف‌پذیر
- ✓ پشتیبانی از افزونه‌ها



میز LXQt

تاریخچه:

LXQt یک محیط دسکتاپ سبک و مدرن برای سیستم‌عامل‌های لینوکسی است که ابتدا توسط LXDE-Qt توسعه یافت و در نسخه ۰٫۷ از LXDE منتشر شد. LXQt از Qt به عنوان کتابخانه گرافیکی استفاده می‌کند و با هدف ارائه یک تجربه کاربری مدرن و کارآمد طراحی شده است.

ویژگی‌ها:

سبکی و کارایی: LXQt از Qt به عنوان کتابخانه گرافیکی استفاده می‌کند که باعث افزایش کارایی و سبکی این محیط دسکتاپ می‌شود.

طراحی مدرن: LXQt دارای طراحی مدرن و زیبا است که تجربه کاربری بهتری را ارائه می‌دهد.

پشتیبانی از ابزارهای Qt: LXQt از ابزارها و ویژگی‌های Qt پشتیبانی می‌کند که امکان اضافه کردن ابزارها و امکانات جدید به دسکتاپ را فراهم می‌کند.

پیکربندی انعطاف‌پذیر: LXQt امکانات پیکربندی بسیار زیادی دارد که به کاربران اجازه می‌دهد دسکتاپ خود را به دلخواه شخصی‌سازی کنند.

مزایا:

- ✓ سبکی و کارایی بالا
- ✓ طراحی مدرن
- ✓ پشتیبانی از ابزارهای Qt

فصل هفتم

فصل هفتم

شخصی سازی میزهای لینوکس

سفارشی‌سازی میزگنوم

نرم‌افزار GNOME Tweaks یک ابزار ساده و قدرتمند برای تنظیمات و سفارشی‌سازی بر روی محیط گرافیکی GNOME است. با استفاده از این ابزار، کاربران می‌توانند انواع تنظیمات مربوط به ظاهر، رفتار و عملکرد GNOME Shell را تغییر دهند و به سلیقه خود تنظیم کنند.

نصب GNOME Tweaks برای سفارشی‌سازی میزگنوم:

```
sudo apt install gnome-tweaks -y
```

نصب افزونه‌های GNOME Shell به کاربران امکان می‌دهد تا قابلیت‌ها و ویژگی‌های جدیدی را به محیط گرافیکی GNOME Shell اضافه کنند. این افزونه‌ها می‌توانند به شکل‌دهی به منوها، پوسته‌ها، نوارها و دیگر عناصر محیط گرافیکی کمک کرده و تجربه کار با GNOME Shell را بهبود بخشند.

نصب افزونه‌های GNOME Shell:

```
sudo apt install gnome-shell-extensions -y
```

دستور gsettings

تنظیم تصویر پس‌زمینه میز گنوم

```
gsettings set org.gnome.desktop.background picture-uri ‘wallpaper1.png’
```

تنظیم نمایش آیکن های میزکار میز گنوم

```
gsettings set org.gnome.desktop.background show-desktop-icons true
```

سفارشی کردن گفتگوی اعلان

Xfce گفتگوی اعلان را به خودش تغییر می دهد. با این دستور می توان آن را پیکربندی کرد:

```
xfce4-notifyd-config
```

گزینه های تکراری را از صفحه ورود حذف کنید. برخی از محیط های دسکتاپ بیش از یک جلسه ارائه می دهند. به عنوان مثال ، دارچین یک جلسه دو بعدی و سه بعدی را ارائه می دهد. گزینه های موجود در لاگین در پوشه usr/share/xsessions/ قرار دارند و گزینه های غیر ضروری را می توان با حذف فایل های مربوطه حذف کرد. به عنوان مثال ، برای حذف گزینه اضافی دارچین:

```
sudo rm /usr/share/xsessions/cinnamon2d.desktop
```

راه‌اندازی مجدد GNOME Shell

این دستور با استفاده از دستور killall پروسه GNOME Shell را متوقف می‌کند و سپس با استفاده از سیگنال ۳(SIGQUIT) ، GNOME Shellرا مجدداً راه‌اندازی می‌کند. این کار می‌تواند به حل مشکلات عملکردی یا نمایشی در محیط گرافیکی GNOME کمک کند و تغییرات اعمال شده را اعمال کند.

```
$ sudo killall -3 gnome-shell
```

نصب ابزارهای ضروری

این دستور در توزیع‌های مبتنی بر Debian و Ubuntu اجازه می‌دهد تا ابزارهای ضروری برای توسعه نرم‌افزار را نصب کنید. build-essential شامل ابزارهایی مانند کامپایلر C/C++، make و سایر ابزارهای لازم برای کامپایل برنامه‌ها است. git یک سیستم کنترل نسخه توزیع شده است که برای مدیریت کدهای منبع استفاده می‌شود.

```
sudo apt install build-essential git -y
```

نصب در توزیع‌های لینوکس دیگر:

- در توزیع‌های مبتنی بر Fedora و CentOS:

```
sudo dnf install @development-tools git -y
```

- در توزیع‌های مبتنی بر Arch Linux:

```
sudo pacman -S base-devel git
```

غیرفعال کردن لیست کاربران

این دستور gsettings برای تنظیم و مدیریت تنظیمات دسکتاپ GNOME به کار می‌رود. در اینجا، ما تنظیم disable-user-list را در محیط ورود GNOME تغییر می‌دهیم. وقتی مقدار این تنظیم به true تغییر داده می‌شود، لیست کاربران موجود در صفحه ورود غیرفعال می‌شود و کاربران باید نام کاربری و رمز ورود خود را به صورت دستی وارد کنند به جای انتخاب از یک لیست.

به طور خاص، این دستور تغییراتی در تنظیمات محیط ورود GNOME (login screen) اعمال می‌کند تا لیست کاربران آنلاین در صفحه ورود غیرفعال شود و کاربران باید نام کاربری و رمز عبور خود را به صورت دستی وارد کنند.

```
gsettings set org.gnome.login-screen disable-user-list true
```

قفل شدن سیستم و رفتن به حالت لاگین

این دستور gsettings برای تنظیم تنظیمات جلسه دسکتاپ GNOME به کار می‌رود. با این دستور، ما تنظیم idle-delay را تغییر می‌دهیم که زمان تاخیر برای ورود به حالت بی‌فعالیت idleرا تعیین می‌کند. در اینجا، مقدار ۱۸۰ به معنای ۱۸۰ ثانیه است، به این معنی که بعد از سه دقیقه بی‌فعالیت، سیستم به حالت بی‌فعالیت ورود می‌کند. بنابراین، با اجرای این دستور، زمان تاخیر برای ورود به حالت بی‌فعالیت در جلسه دسکتاپ GNOME به ۱۸۰ ثانیه (سه دقیقه) تنظیم می‌شود.

```
gsettings set org.gnome.desktop.session idle-delay 180
```

غیرفعال‌سازی و متوقف کردن انواع hybrid sleep، sleep، hibernate

ما ابتدا هر یک از حالت‌های hybrid sleep، sleep، hibernate و suspend را غیرفعال می‌کنیم و سپس آن‌ها را متوقف می‌کنیم تا از اجرا شدن آن‌ها جلوگیری شود.

Hybrid Sleep: حالتی است که ترکیبی از حالت sleep و hibernate است و در برخی سیستم‌ها وضعیت فعلی را در حافظه RAM ذخیره کرده و سپس به حالت hibernate وارد می‌شود.

Sleep: حالتی است که سیستم به حالت استراحت وارد می‌شود و مصرف انرژی کمتری دارد.

Hibernate: در این حالت، وضعیت جاری سیستم به روی دیسک ذخیره می‌شود و سیستم کاملاً خاموش می‌شود.

Suspend: حالتی است که سیستم به حالت استراحت می‌رود اما برخلاف sleep، برخی از قسمت‌های سخت‌افزاری همچون RAM فعال می‌مانند.

Disable and stop hybrid sleep

```
sudo systemctl disable hybrid-sleep.service
sudo systemctl stop hybrid-sleep.service
sudo systemctl mask hybrid-sleep.service
```

Disable and stop sleep

```
sudo systemctl disable sleep.target
sudo systemctl stop sleep.target
sudo systemctl mask sleep.target
```

Disable and stop hibernate

```
sudo systemctl disable hibernate.target
sudo systemctl stop hibernate.target
sudo systemctl mask hibernate.target
```

Disable and stop suspend

```
sudo systemctl disable suspend.target
sudo systemctl stop suspend.target
sudo systemctl mask suspend.target
```

فعال کردن screen saver

این دستور gsettings برای فعال‌سازی نمایشگر صفحه نمایش (screensaver) در محیط دسکتاپ GNOME به کار می‌رود. با اجرای این دستور، تنظیم idle-activation-enabled به true تغییر می‌یابد که به معنای فعال بودن قابلیت فعال‌سازی نمایشگر صفحه نمایش بر اساس بی‌فعالیت شدن سیستم است. بنابراین، با اجرای این دستور، نمایشگر صفحه نمایش فعال شده و هنگامی که سیستم به حالت بی‌فعالیت وارد شود، نمایشگر صفحه نمایش فعال خواهد شد.

```
gsettings set org.gnome.desktop.screensaver idle-activation-enabled true
```

دستور pactl

این دستور به کاربردی جهت فعال‌سازی افزایش بیش از حد صدای صوتی در سیستم صدا (Over-Amplification) در لینوکس می‌پردازد. با اجرای این دستور، از دستور pactl برای تنظیم میزان صدای خروجی استفاده می‌شود. در این دستور، set-sink- -- volume 0 150% به معنی تنظیم میزان صدا برای sink (0) به مقدار ۱۵۰٪ است.

بنابراین، با اجرای این دستور، میزان صدا برای sink انتخابی به مقدار ۱۵۰٪ تنظیم می‌شود که موجب افزایش بیش از حد صدای صوتی در سیستم می‌شود. این امکان ممکن است در مواردی که نیاز به صدای بیشتری دارید، مفید باشد.

```
pactl -- set-sink-volume 0 150%
```

غیرفعال کردن بلوتوث

```
# Disable Bluetooth by default
sudo sed -i 's/NoEnable=true/NoEnable=false/' /etc/bluetooth/main.conf
```

غیرفعال کردن وب کم

```
# Disable webcam by default
sudo modprobe -r uvcvideo
echo "blacklist uvcvideo" | sudo tee /etc/modprobe.d/blacklist-uvcvideo.conf
```

غیرفعال کردن میکروفن

```
# Disable microphone by default
sudo modprobe -r snd_hda_intel
echo "blacklist snd_hda_intel" | sudo tee /etc/modprobe.d/blacklist-snd_hda_intel.conf
```

نصب درایور تاچ پد

```
# Install touchpad driver
sudo apt-get install xserver-xorg-input-synaptics -y
```

نصب درایور پرینتر

```
# Install CUPS
sudo apt-get install cups -y
```

نصب درایور اسکنر

```
# Install XSane and SANE
sudo apt-get install xsane sane -y
```

فصل هشتم

فصل هشتم

پردازش متون

دستور nl

ابزار nlیک ابزار مفید برای شمارش خطوط در فایل‌ها است. با استفاده از این ابزار، می‌توانید تعداد خطوط فایل‌های خود را به راحتی محاسبه کنید. همچنین، با ترکیب این ابزار با ابزارهای دیگر، می‌توانید کارهای مفیدی با آن انجام دهید.

گزینه های دستور:

- b- شماره خطوط را فقط برای خطوط خالی نمایش می‌دهد.
- ba- شماره خطوط را فقط برای خطوط خالی و خطوطی که با فضای خالی شروع می‌شوند نمایش می‌دهد.
- bn- شماره خطوط را فقط برای خطوطی که با شماره شروع می‌شوند نمایش می‌دهد.
- e- شماره خطوط را به جای اعداد، با حروف الفبا نمایش می‌دهد.
- f- شماره خطوط را از یک شماره دلخواه شروع می‌کند.
- i- شماره خطوط را با فاصله مشخصی از ابتدای خط نمایش می‌دهد.
- l- شماره خطوط را در سمت چپ خطوط نمایش می‌دهد.
- n- شماره خطوط را به صورت پیش فرض (قبل از هر خط) نمایش می‌دهد.
- p- شماره خطوط را به صورت پیش فرض (بعد از هر خط) نمایش می‌دهد.
- s- کاراکتر جداکننده بین شماره خط و متن را مشخص می‌کند.
- v- شماره خطوط را به صورت عمودی در سمت چپ صفحه نمایش می‌دهد.
- w- تعداد کاراکترهای اختصاص داده شده به شماره خطوط را مشخص می‌کند.

مثال های دستور:

۱. شماره‌گذاری خطوط بدنه:

```
nl -b a file.txt
```

۲. محدودکننده‌های بخش:

```
nl -d “|” file.txt
```

۳. شماره‌گذاری خطوط پانویس:

```
nl -f a file.txt
```

۴. شماره‌گذاری خطوط سربرگ:

```
nl -h a file.txt
```

۵. افزایش شماره خط:

```
nl -i 5 file.txt
```

۶. ادغام خطوط خالی:

```
nl -l 2 file.txt
```

۷. فرمت شماره‌گذاری:

```
nl -n rz file.txt
```

۸. عدم بازنشانی:

```
nl -p file.txt
```

۹. جداکننده شماره:

```
nl -s ": " file.txt
```

۱۰. شماره خط ابتدایی:

```
nl -v 10 file.txt
```

۱۱. عرض شماره:

```
nl -w 3 file.txt
```

۱۲. نمایش اطلاعات راهنما

```
nl --help
```

۱۳. نمایش اطلاعات نسخه:

```
nl --version
```

دستور head

```
head  / tail / cat / less
```

چند خط اول یک فایل را نشان می دهد. سویچ n– تعداد خطوط دلخواهی را نشان می دهد (به طور پیش فرض ۱۰ است). دستورhead ده خط اول نام هر فایلی را می خواند. دستور اصلی دستور head به صورت زیر است:

```
head [options] [file(s)]
```

برای مثال ، دستور زیر ده خط اول فایل با نام /etc/passwdرا نمایش می دهد.

```
# head /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
```

اگر بخواهید تعداد خطوط بیشتری نسبت به ده پیش‌فرض بازیابی شود، از گزینه n به همراه یک عدد صحیح که تعداد خطوطی را که باید بازیابی می‌شوند، استفاده می‌شود. به عنوان مثال ، دستور زیر ۳ خط اول فایل را نمایش می دهد.

```
# prints the first 3 lines
$ head-n 3 c
this
```

```
file
has
tail
```

دستور head در سیستم عامل‌های Unix و مشتقات آن مانند Linux برای نمایش بخش ابتدایی یک فایل استفاده می‌شود. این دستور به صورت پیش‌فرض اولین ۱۰ خط از یک فایل را نمایش می‌دهد. عمدتا برای مشاهده اطلاعات اولیه یک فایل یا خروجی‌هامورد استفاده قرار می‌گیرد.

مثال های دستور:

۱. نمایش ۵ خط اول از یک فایل:

```
head -n 5 file.txt
```

۲. نمایش ۲۰ خط اول از چند فایل:

```
head -n 20 file1.txt file2.txt
```

۳. نمایش ابتدای یک فایل با نمایش شماره خطها:

```
head -n 10 -v file.txt
```

۴. نمایش ابتدای یک خروجی:

```
ls -l | head
```

۵. نمایش بخشی از یک فایل بر اساس بایت:

```
head -c 100 file.txt
```

۶. نمایش خطوط بدون نمایش خطهای تکراری:

```
head -q file.txt
```

۷. نمایش ابتدای یک فایل بدون نمایش نام فایل:

```
head -h file.txt
```

۸. نمایش تعداد خطها بدون نمایش پیام اطلاعاتی:

```
head -q -v file.txt
```

۹. نمایش ابتدای یک فایل به همراه نام فایل:

```
head -v -f file.txt
```

۱۰. نمایش ابتدای یک فایل به صورت معکوس:

```
head -n -10 file.txt
```

دستور tail

چند خط آخر یک فایل را نمایش می دهد.

```
# prints the end of the file, beginning with the 4th line
$ tail-n +4 c
exactly
six
lines
```

دستور tail یک ابزار است که در سیستم‌های عامل مبتنی بر یونیکس (UNIX) و لینوکس استفاده می‌شود. این دستور برای نمایش آخرین محتوای یک فایل متنی استفاده می‌شود.

ساختار دستور:

```
tail [OPTIONS] [FILE]
```

در اینجا، FILE نشان‌دهنده نام فایلی است که می‌خواهید محتوای آخر آن را ببینید. اگر FILE مشخص نشود، tail به صورت پیش‌فرض محتوای آخر فایل استاندارد ورودی را نمایش می‌دهد. برخی از استفاده‌های رایج tail عبارتند از:

نمایش آخرین n خط از یک فایل:

```
tail -n [NUMBER] [FILE]
```

با استفاده از گزینه n- و تعیین یک عدد مثبت برای [NUMBER]، tail آخرین NUMBER خط از فایل مورد نظر را نمایش می‌دهد.

نمایش آخرین n بایت از یک فایل:

```
tail -c [NUMBER] [FILE]
```

با استفاده از گزینه c- و تعیین یک عدد مثبت برای [NUMBER]، tail آخرین NUMBER بایت از فایل مورد نظر را نمایش می‌دهد.

دنبال کردن (follow) تغییرات در یک فایل:

```
tail -f [FILE]
```

با استفاده از گزینه f-، tail به صورت تعاملی محتوای آخرین بخش از فایل را نمایش می‌دهد و تغییرات جدید را به صورت زنده نمایش می‌دهد. این بسیار مفید است زمانی که می‌خواهید تغییرات جدید در یک فایل را به صورت زنده پیگیری کنید، مانند فایل‌های لاگ. دستور tail بسیار مفید است زمانی که می‌خواهید به سرعت آخرین بخش از یک فایل را ببینید، به ویژه در مواردی که فایل بسیار بزرگ است. با استفاده از این دستور، می‌توانید به راحتی به آخرین خطوط یا بایت‌ها دسترسی پیدا کنید و تغییرات جدید را ببینید.

مثال های دستور:

نمایش ۵ خط آخر از یک فایل:

```
tail -n 5 file.txt
```

نمایش ۲۰ خط آخر از چند فایل:

```
tail -n 20 file1.txt file2.txt
```

نمایش انتهای یک فایل با نمایش شماره خطها:

```
tail -n 10 -v file.txt
```

نمایش انتهای یک خروجی:

```
ps aux | tail
```

نمایش بخشی از یک فایل بر اساس بایت:

```
tail -c 100 file.txt
```

نمایش خطوط بدون نمایش خطاهای تکراری:

```
tail -q file.txt
```

نمایش انتهای یک فایل بدون نمایش نام فایل:

```
tail -h file.txt
```

نمایش تعداد خطاهای مشخص از انتهای یک فایل:

```
tail -n +15 file.txt
```

نمایش انتهای یک فایل به همراه نام فایل:

```
tail -v -f file.txt
```

نمایش انتهای یک فایل به صورت معکوس:

```
tail -n -10 file.txt
```

دستور cat

محتوای فایل را به خروجی (معمولا ترمینال) می فرستد. می تواند فقط با یک فایل یا چندین فایل استفاده شود و اغلب به سرعت آنها را نشان می دهد.

```
$ cat a.txt
this is a file
```

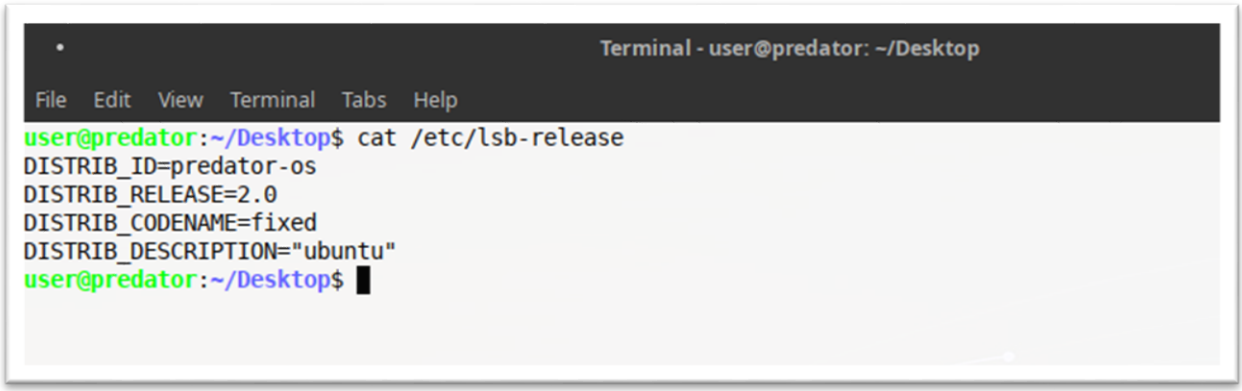
```
cat  a.txt  b.txt
file a
file b
```

دستور Cat الحاق به طور مکرر در Linux استفاده می شود .داده ها را از پرونده می خواند و محتوای آنها را به عنوان خروجی می دهد .این به ما کمک می کند تا پرونده ها را ایجاد ، مشاهده و ادغام کنیم .بنابراین اجازه دهید برخی از دستورات cat که اغلب استفاده می شوند را ببینیم.

مشاهده محتوای یک فایل

بعد از دستور مسیر فایل خود را وارد نمایید.

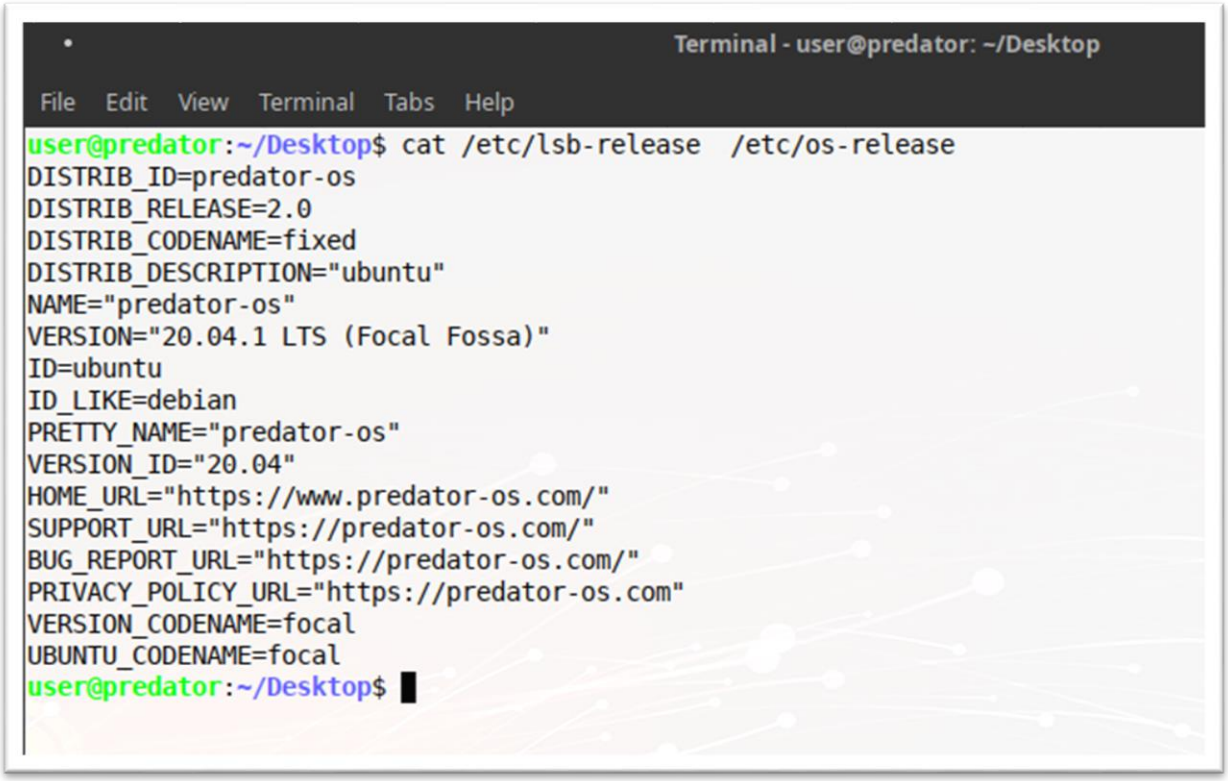
```
cat   /etc/lsb-release
```



مشاهده محتوای چندین فایل

```
$ cat file1 file2
```

خروجی دستور:



مشاهده محتویات یک پرونده قبل با شماره خط

```
$cat -n filename
```

خروجی دستور:

این محتوا را با شماره خط نشان می دهد

```
cat  -n  ali.txt
```

این مثال است و یک آرایه منحصر به فرد است. دستور زیر یک پرونده ایجاد میکند.

```
$ cat > newfile
```

خروجی دستور:

ارسال محتوای یک پرونده را در پرونده دیگر

```
$cat [filename-whose-contents-is-to-be-copied] > [destination-filename]
```

محتوا در پرونده مقصد کپی می شود.

دستور Cat می تواند خطوط خالی تکراری را در دستورخروجی ازبین ببرد

```
$ cat -s ali.txt
```

دستور Cat می تواند محتویات یک فایل را به انتهای پرونده دیگری اضافه کند.

```
$ cat file1 >> file2
```

دستور Cat می تواند با استفاده از دستور tac محتوا را به ترتیب معکوس نمایش دهد.

```
$tac filename
```

دستور Cat می تواند انتهای خط را برجسته کند.

```
$ cat -E "نام پرونده"
$cat -E "filename"
```

اگر می خواهید از گزینه v- ، E-و T- با هم استفاده کنید ، به جای نوشتن vET- در دستور ، می توانید از گزینه A- خط فرمان استفاده کنید.

```
$ cat -A "نام پرونده"
$cat -A "filename"
```

دستور Cat برای باز کردن فایل های dashed

```
$ cat - "-dashfile"
```

دستور Cat درصورتی که فایل محتوای زیادی داشته باشد و در ترمینال قرار نگیرد.

```
$cat "filename" | more
```

دستور Cat برای ادغام محتویات چندین پرونده.

```
$ cat "filename1" "filename2" "filename3"> "merged_filename"
```

دستور Cat برای نمایش محتوای تمام فایل های متنی در پوشه.

```
$ cat * .txt.
```

مشاهده محتویات فایل های چندگانه در ترمینال

در مثال زیر، محتویات فایل testو test1 در ترمینال نمایش داده می شود.

```
# cat test test1
```

یک فایل را با دستور cat ایجاد می کند. ما فایلی با نام test2 را با دستور زیر ایجاد خواهیم کرد.

```
# cat  >  test2
```

پس از زدن دستور، متن مورد نظر را تایپ کرده و CTRL + Dرا فشار دهید . متن در فایل test2 نوشته خواهد شد. می توانید محتویات فایل را با دستور catزیر دنبال کنید.

```
# cat  test2
```

از دستور Cat با گزینه های بیشتر و کمتر استفاده کنید اگر فایل با تعداد زیادی از محتوا که در ترمینال خروجی مناسب نیست و صفحه نمایش بسیار سریع حرکت می کند، ما می توانیم با استفاده از پارامترهای بیشتر و کمتر با دستور catبه عنوان مثال بالا استفاده کنیم.

```
# cat song.txt |more
# cat song.txt |less
```

نمایش شماره خط در فایل

با گزینه n- می توانید شماره خطوط یک فایل را در ترمینال خروجی ببینید.

```
# cat  -n  song.txt
```

نمایش $ در پایان فایل

در ادامه میتوانید با گزینه e– علامت " $ " در انتهای خطوط قراردهید.

```
# cat  -e  test
```

خروجی دستور:

```
سلام به همه، چگونه می توانم انجام دهم؟ $
 $
سلام، من خوبم $
چگونه آموزش خود را ادامه می دهی؟ $
 $
```

نمایش فاصله بین خطوط جدا شده در فایل

در خروجی پایین، می توانیم بفهمیم فاصله ها با کاراکتر ' ^ ' پر شده است.

```
# cat  –T  test
```

خروجی دستور:

```
سلام  ^من همه، چطور؟
سلام  ^من خوبم
^ ^ ^ 
```

ضمیمه خروجی استاندارد با اپراتور هدایتگر

در فایل جاری با نماد « >> » مقادیر اضافه به فایل داده شده اضافه میشود.در اینجا، محتویات فایل تست در پایان فایل test1 اضافه می شود.

```
# cat   test >> test1
```

انتقال استاندارد ورودی با اپراتور هدایتگر

از نام فایل test2به عنوان ورودی برای یک دستور استفاده می کند و خروجی در ترمینال نشان داده می شود.

```
# cat < test2
```

مسیر فایل های چندگانه در یک فایل

این یک فایل به نام test3ایجاد خواهد کرد و تمام خروجی ها در یک فایل جدید ایجاد شده هدایت می شوند.

```
# cat  test  test1  test2  > test3
```

مرتب سازی محتویات فایل های چندگانه در یک فایل

در این مثال یک فایل تست ۴ ایجاد می شود و خروجی های دستور cat به صورت مرتب در یک فایل تازه ایجاد شده هدایت می شود.

```
# cat   test  test1  test2  test3 |  ordered > test4
```

هدایت گرها–Redirect

تغییر مسیر خروجی به یک فایل، اگر فایل وجود داشته باشد، آن را بازنویسی می کند

```
echo “Hello, World!” > output.txt
```

اضافه کردن خروجی به یک فایل، اگر فایل وجود داشته باشد، به آن اضافه می کند

```
echo “Hello, again!” >> output.txt
```

تغییر مسیر ورودی از یک فایل

```
sort < unsorted.txt
```

تغییر مسیر ورودی از یک دستور

```
wc -l << EOF
one
two
three
EOF
```

تغییر مسیر خروجی و خطاها به یک فایل

```
ls /nonexistentdirectory &> errors.txt
```

اضافه کردن خروجی و خطاها به یک فایل

```
ls /nonexistentdirectory &>> errors.txt
```

تغییر مسیر خطاها به یک فایل

```
ls /nonexistentdirectory 2> errors.txt
```

اضافه کردن خطاها به یک فایل

```
ls /nonexistentdirectory 2>> errors.txt
```

pipe کردن خروجی یک دستور به دستور دیگر

```
ls -l | grep ".txt"
```

تغییر مسیر خروجی به یک فایل و همچنین نمایش آن در stdout

```
ls -l | tee output.txt
```

تغییر مسیر ورودی از یک رشته

```
grep "Hello" <<< "Hello, World!"
```

باز کردن یک فایل برای خواندن و نوشتن

```
exec 3<> file.txt
```

تغییر مسیر خروجی فایل descriptor

```
exec 3>&1
```

تغییر مسیر ورودی از یک فرآیند

```
cat <(ls)
```

تغییر مسیر خروجی به یک فرآیند

```
echo "Hello, World!" >(cat)
```

ضبط یک جلسه ترمینال

```
script session.txt
```

تغییر مسیر خروجی به /dev/null دور انداختن خروجی)

```
command > /dev/null
```

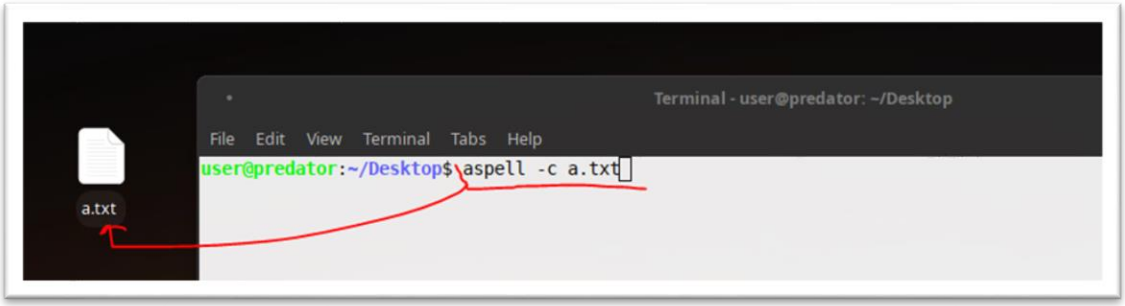
دستور aspell

به عنوان املاً در لینوکس استفاده می شود .به طور کلی ، پرونده های داده شده یا هر چیزی را از ورودی استاندارد اسکن می کند و سپس غلط املایی را بررسی می کند .سرانجام به کاربر امکان می دهد کلمات را به صورت تعاملی اصلاح کند.

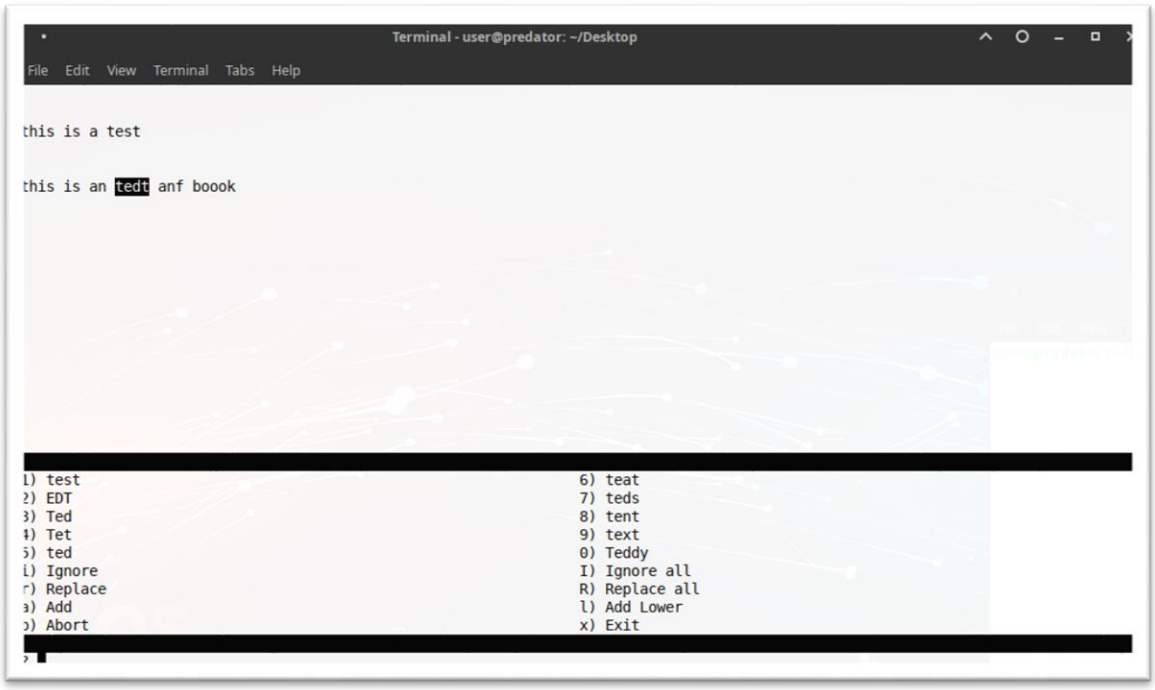
ساختار دستور:

```
aspell check [options] filename
```

مثال ۱ :یک پرونده را از نظر غلط املایی بررسی کنید



بررسی و خطایابی



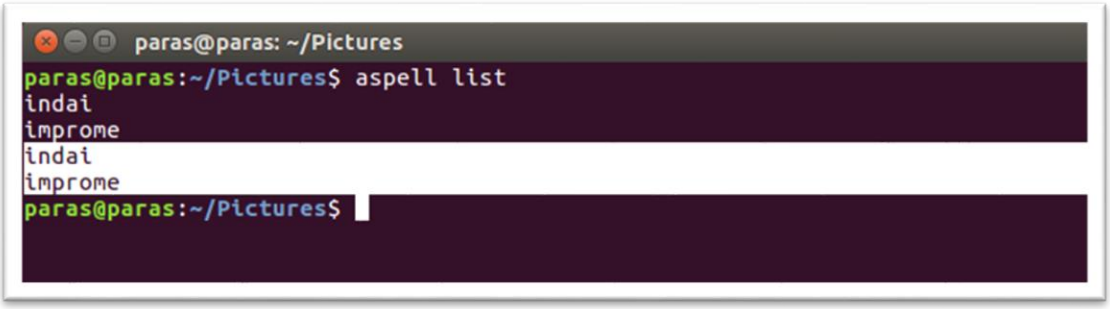
مثال ۲ :استفاده از aspell برای بررسی کلمات منفرد.

```
aspell -a
```

توجه :به محض اجرای این دستور ، منتظر ورود کاربر می ماند .یک کلمه را در این حالت تایپ کنید ، enterرا فشار دهید ، و خواهید دید که **spell** پیشنهادهای هجی را در stdout ارائه می دهد.



مثال ۳ :استفاده از **aspell** برای بررسی کلمات به صورت انبوه است ، منتظر ورود کاربراست .هر تعداد کلمه که می خواهید اضافه کنید و پس از اتمام ، Ctrl + Dرا فشار دهید .سپس خواهید دید که **aspell** کلمات نادرست هجی شده را در زیر ورودی شما ارائه می دهد.



مثال های دستور:

۱. بررسی املای صحیح یک فایل متنی:

```
aspell check document.txt
```

۲. بررسی املای صحیح یک فایل و نمایش خطاها به صورت جزئیات:

```
aspell -c document.txt
```

۳. اصلاح املای صحیح در یک فایل متنی:

```
aspell -t -c file.txt
```

۴. اضافه کردن یک کلمه به لغت‌نامه شخصی:

```
aspell --lang=en create master personal.en.pws < new_word.txt
```

۵. بررسی املای صحیح با استفاده از یک لغت‌نامه خاص:

```
aspell --master=my_dict.txt check document.txt
```

۶. نمایش کلمات پیشنهادی برای اصلاح املای صحیح:

```
aspell list < misspelled_words.txt
```

۷. اصلاح املایی خودکار برای یک فایل متنی:

```
aspell --mode=html --lang=en --encoding=utf-8 -c file.html
```

۸. اجرای aspell بدون نمایش پیغام‌های اطلاعاتی:

```
aspell -q check text.txt
```

۹. استفاده از aspell برای چک کردن املای یک جمله:

```
echo “Hello, wrld!” | aspell pipe
```

۱۰. اصلاح املایی یک کلمه خاص:

```
aspell -a <<< “recieve”
```

دستور column

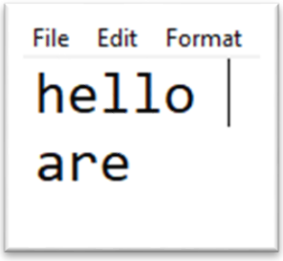
از دستور column در لینوکس برای نمایش محتویات یک فایل در ستون استفاده می شود .ورودی ممکن است از ورودی استاندارد یا از پرونده گرفته شود .این دستور اساساً ورودی چند ستون را می شکند .سطرها قبل از ستون ها پر می شوند .خطوط خالی از ورودی نادیده گرفته می شوند مگر اینکه از گزینه **-e** استفاده شود.

ساختار دستور:

```
column [-entx] [-c columns] [-s sep] [file ...]
```

مثال :

فرض کنید یک فایل متنی با محتوای زیر دارید:



برای نمایش اطلاعات فایل متنی به صورت ستون ، دستور زیر را وارد می کنید:

```
column filename.txt
```

```
hello   are
```

فرض کنید ، شما می خواهید ورودی هایی را که توسط جدا کننده های خاص جدا شده اند ، به ستون های مختلف مرتب کنید .به عنوان مثال ، این نمونه فایل متنی:

```
File  Edit  Format  View  Help
No.|Dish|Y/N
01|Idli|Y
02|Samosa|Y
03|Chole bhature|Y
04|Chowmein|N
05|Pasta|Y
06|Burger|Y
```

برای جدا کردن ستون بر اساس جداکننده "|" ، شما دستور زیر را می دهید ، که به نوبه خود خروجی داده شده را تولید می کند:

```
hruvkaushal1997@LAPTOP-4DNJAQAV:/mnt/c/Users/Hp/Desktop$ column hello.txt -t -s "|"
o.  Dish           Y/N
1   Idli           Y
2   Samosa         Y
3   Chole bhature  Y
4   Chowmein       N
5   Pasta          Y
6   Burger         Y
```

درصورتی که چندین جداکننده (همان نوع) بدهید ، این دستور با آنها به عنوان یک جداگانه رفتار می کند .مثلا:

```
File  Edit  Format  View  Help
No.|Dish|Y/N
01|Idli|Y
02|Samosa|Y
03|||Chole bhature|Y
04|Chowmein|||N
05|Pasta|||Y
06|Burger||Y
```

با استفاده از دستور ، خروجی حاصل می شود:

```
dhruvkaushal1997@LAPTOP-4DNJAQAV:/mnt/c/Users/Hp/Desktop$ column hello.txt -t -s "|"
No.  Dish           Y/N
01   Idli           Y
02   Samosa         Y
03   Chole bhature  Y
04   Chowmein       N
05   Pasta          Y
06   Burger         Y
```

گزینه های دستور:

-Jاین گزینه سیستم را برای نمایش خروجی در قالب JSON مشخص می‌کند.

c – عرض خروجی را براساس تعداد کل کاراکترهای مجاز در یک ردیف تعریف می‌کند.

d – برای تعریف اینکه سربرگ چاپ نمی‌شود استفاده می‌شود.

N – این نام ستون‌ها را مشخص می‌کند و هر نام با یک ویرگول از هم جدا می‌شود.

H – برای پنهان کردن نمایش ستون‌های جداگانه استفاده می‌شود.

V – اطلاعات نسخه را نمایش می‌دهد و خارج می‌شود.

h – اطلاعات راهنما را نشان می‌دهد و خارج می‌شود.

s – جداکننده ستون را برای خروجی تعریف می‌کند.

t – برای تعیین جدول با تعیین تعداد ستون‌ها اعمال می‌شود.

R – برای ترازبندی متن در یک ستون خاص استفاده می‌شود.

T – به سیستم اجازه می‌دهد متن را در هر کجا که لازم باشد کوتاه کند تا بتواند عرض ستون را تنظیم کند تا اطمینان حاصل کند که ستون بیش از حد گسترده نمی‌شود.

O – به کاربر اجازه می‌دهد تا ترتیب نمایش ستون‌ها را تعریف کند.

L – خطوط خالی را بعد از هر سطر درج می‌کند.

دستور column

دستور column در سیستم عامل‌های Unix و مشتقات آن مانند Linux برای فرمت‌بندی و نمایش داده‌ها در ستون‌های قابل خواندن استفاده می‌شود. این دستور به کاربر امکان می‌دهد تا خروجی یک فایل یا خروجی یک دستور را به صورت ستون‌های منظم و زیبا نمایش دهد. column معمولاً برای سازماندهی و نمایش اطلاعات به صورت منظم و زیبا در خروجی‌های متنی استفاده می‌شود.

مثال‌های دستور:

۱. نمایش فایل متنی در ستون‌های خوانا:

```
column file.txt
```

۲. نمایش خروجی یک دستور در ستون‌های منظم:

```
ls | column
```

۳. نمایش داده‌ها در ستون‌های با عرض مشخص:

```
column -t -s, data.csv
```

۴. نمایش داده‌ها در ستون‌های جدا شده توسط فاصله:

```
column -s' ' file.txt
```

۵. نمایش داده‌ها در ستون‌های با تعداد ستون‌ها مشخص:

```
column -c 3 file.txt
```

۶. نمایش داده‌ها در ستون‌های با استفاده از جداکننده خاص:

```
column -o '|' file.txt
```

۷. نمایش داده‌ها در ستون‌های با جا به جایی عمودی:

```
column -x file.txt
```

۸. نمایش داده‌ها در ستون‌های با استفاده از فاصله به عنوان جداکننده:

```
column -s ' ' file.txt
```

۹. نمایش داده‌ها در ستون‌های به صورت افقی:

```
column -c 80 file.txt
```

۱۰. نمایش داده‌ها در ستون‌های با تعداد سطرهای مشخص:

```
column -c 4 file.txt
```

دستور rev

دستور rev در سیستم عامل‌های یونیکس و لینوکس برای برعکس کردن محتوای یک خط متن استفاده می‌شود. این دستور به کاربر اجازه می‌دهد تا محتوای یک خط متن را از راست به چپ برعکس کند.

گزینه‌های دستور:

بدون گزینه: برعکس کردن محتوای یک خط متن.

```
echo “Hello, World!” | rev
```

خروجی دستور: !dlroW ,olleH

برعکس کردن کلمات در هر خط متن.

```
echo “This is a sample text.” | rev -c
```

خروجی دستور: sihT si a elpmas .txet

برعکس کردن ترتیب کاراکترهای خطوط متن.

```
echo “12345” | rev -t
```

خروجی دستور: ۵۴۳۲۱

نمایش اطلاعات نسخه و خروجی دستور rev.

```
rev -V
```

خروجی دستور: rev from util-linux 2.36

برعکس کردن کلمات در هر خط متن با استفاده از فاصله به عنوان جداکننده.

```
echo “This is another example” | rev -w
```

خروجی دستور: sihT si rehtona elpmaxe

حذف خطاهایی که ممکن است در اجرای دستور rev ایجاد شوند.

```
rev -q
```

خروجی دستور: (بدون خروجی در صورت عدم وقوع خطا)

برعکس کردن ترتیب کاراکترهای درون هر خط متن بدون تغییر ترتیب خطوط.

```
echo -e “123\nabc\ndef” | rev -l
```

خروجی دستور: ncba\nfed\۳۲۱

نمایش راهنمای استفاده از دستور rev.

```
rev -h
```

خروجی دستور: راهنمای استفاده از دستور rev

دستور Sort

این دستور برای چاپ خطوط فایل های متنی ورودی و الحاق همه فایل ها به ترتیب مرتب شده استفاده می شود. دستور مرتب سازی فضای خالی را به عنوان جداکننده فیلد و کل فایل ورودی را به عنوان کلید مرتب سازی می گیرد. توجه به این نکته مهم است که دستور مرتب سازی در واقع فایل ها را مرتب نمی کند بلکه فقط خروجی مرتب شده را چاپ می کند تا زمانی که خروجی را تغییر مسیر دهید. .دستور SORT دارای ویژگی های مختلفی است که در نتیجه دستورات از آن پیروی می کند .اول اینکه خطوط دارای اعداد قبل از خطوط الفبایی قرار می گیرند .خطوطی که حروف کوچک دارند زودتر از خطوطی که نویسه‌های مشابه با حروف بزرگ دارند نمایش داده می‌شوند.

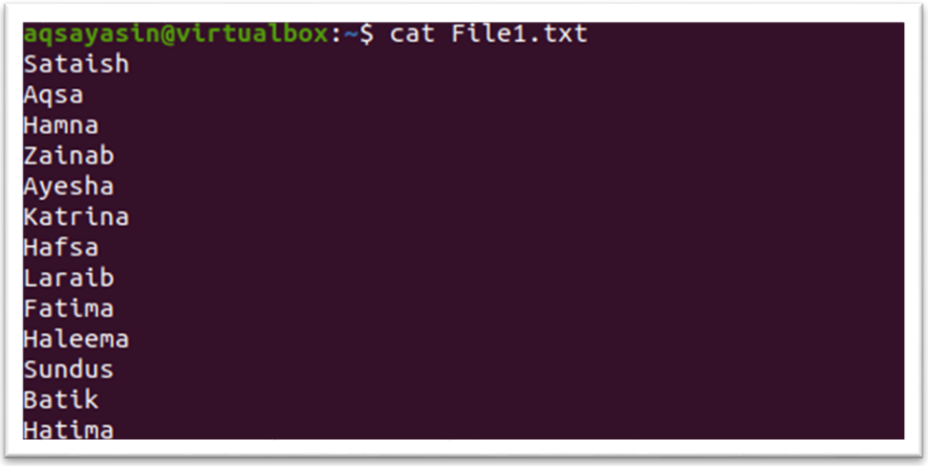
ساختار دستور:

```
sort [options]  file
```

مثال :

این یک مثال ساده از مرتب‌سازی فایلی است که داده‌های نام دارد. بنابراین، فایلی به نام file1.txt را در نظر بگیرید .با استفاده از دستور ضمیمه شده، محتویات فایل را نمایش خواهیم داد:

```
$ Cat file1.txt
```



اکنون از دستور برای مرتب کردن متن در فایل استفاده کنید:

```
$ sort file1.txt
```

```
aqsayasin@virtualbox:~$ sort File1.txt
Aqsa
Ayesha
Batik
Fatima
Hafsa
Haleema
Hamna
Hatima
Katrina
Laraib
Sataish
Sundus
Zainab
```

ذخیره خروجی دستور در فایل

با استفاده از دستور sort متوجه خواهید شد که نتیجه آن فقط نمایش داده می شود اما ذخیره نمی شود .برای گرفتن نتیجه باید آن را ذخیره کنیم .برای این منظور از گزینه o– در دستور sort استفاده می شود. نمونه ای از نام نمونه۱ txt.را با نام خودروها در نظر بگیرید .می خواهیم آنها را مرتب کنیم و داده های حاصل را در یک فایل جداگانه ذخیره کنیم .فایلی با نام result.txt در زمان اجرا ایجاد می شود و خروجی مربوطه در آن ذخیره می شود .داده های sample1.txt به فایل حاصل منتقل می شود و سپس با کمک- oداده های مربوطه مرتب می شوند .ما داده ها را با استفاده از دستور cat نمایش داده ایم:

```
aqsayasin@virtualbox:~$ cat sample1.txt
Ford
Meclaran
Honda
Toyota
Suzuki
Hondai
Yamaha
Ferari
B.M.W
```

```
aqsayasin@virtualbox:~$ sort sample1.txt > result.txt
aqsayasin@virtualbox:~$ sort -o result.txt sample1.txt
aqsayasin@virtualbox:~$ cat result.txt
B.M.W
Ferari
Ford
Honda
Hondai
Meclaran
Suzuki
Toyota
Yamaha
```

خروجی نشان می دهد که داده ها مرتب شده و در فایل دیگری ذخیره شده اند.

مرتب سازی برای شماره ستون

مرتب سازی تنها بر روی یک ستون انجام نمی شود. ما می توانیم یک ستون را به دلیل ستون دوم مرتب کنیم. اجازه دهید نمونه ای از یک فایل متنی داشته باشیم که در آن نام و نشان دانش آموزان وجود دارد. ما می خواهیم آنها را به ترتیب صعودی سازماندهی کنیم. بنابراین از کلمه کلیدی k– در دستور استفاده خواهیم کرد. در حالی که n– برای مرتب سازی عددی استفاده می شود.

```
aqsayasin@virtualbox:~$ cat file3.txt
Arham 20
sajid 40
mazhar60
Wahab 20
Yasin 10
Jawad 60
sajid 40
Yasin 10
```

```
$ sort –k 2n file3.txt
```

همانطور که دو ستون وجود دارد، بنابراین ۲ با n استفاده می شود.

وضعیت مرتب شده یک فایل را بررسی کنید

اگر مطمئن نیستید که فایل حاضر مرتب شده است یا نه، با استفاده از دستوری که ابهام را روشن می کند و پیام را نمایش می دهد، این شک را برطرف کنید. ما به دو مثال اساسی خواهیم پرداخت:

داده های مرتب نشده

دستور از کلمه کلیدی c– استفاده می کند. با این کار بررسی می شود که آیا داده های فایل مرتب شده اند یا خیر. اگر داده ها مرتب نشده باشند، خروجی شماره خط اولین کلمه را که در آن مرتب سازی نشده در فایل وجود دارد و همچنین کلمه را نمایش می دهد.

```
$ sort –c sample2.txt
```

```
aqsayasin@virtualbox:~$ sort -c sample2.txt
sort: sample2.txt:3: disorder: Lady finger
```

داده های مرتب شده

در این مورد، زمانی که داده ها از قبل سازماندهی شده اند، نیازی به انجام کار دیگری نیست. یک فایل result.txt را در نظر بگیرید.

```
aqsayasin@virtualbox:~$ cat result.txt
B.M.W
Ferari
Ford
Honda
Hondai
Meclaran
Suzuki
Toyota
Yamaha
```

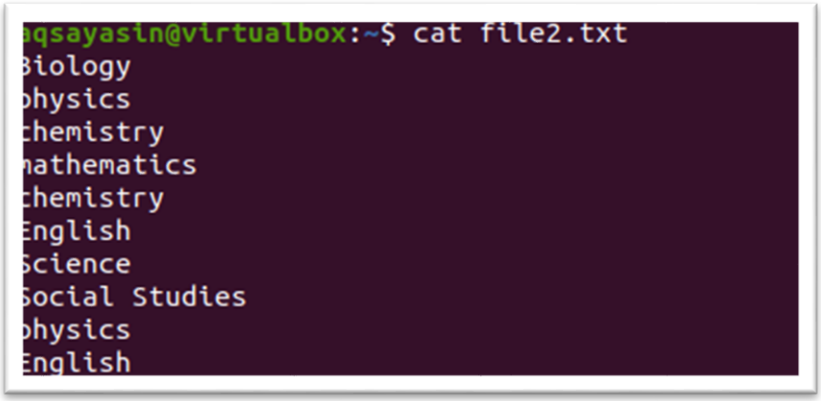
```
$ sort –c result.txt
```



از نتیجه، می بینید که هیچ پیامی نشان داده نمی شود که نشان دهد داده های فایل مربوطه قبلا مرتب شده اند.

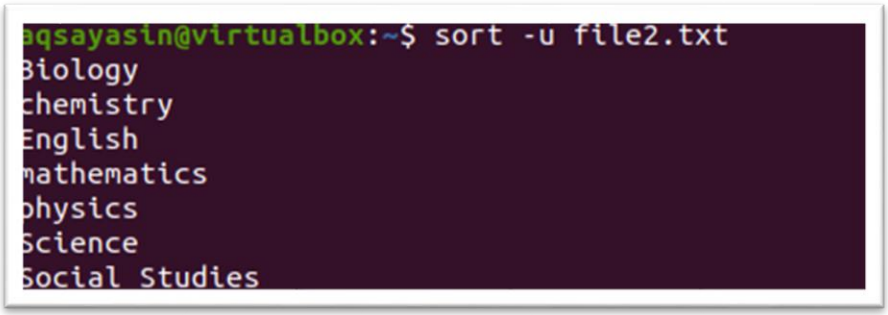
حذف موارد تکراری

در اینجا مفیدترین گزینه به نوعی است .این به حذف کلمات تکراری در یک فایل کمک می کند و آیتم فایل را نیز سازماندهی می کند .همچنین یکنواختی داده های موجود در فایل را حفظ می کند. نام فایل file2.txt را در نظر بگیرید که نام موضوعات را داشته باشد اما یک موضوع چندین بار تکرار شود .سپس دستور مرتب سازی از کلمه کلیدی u– برای حذف موارد تکراری و مرتبط استفاده می کند:




```
$ sort –u file2.txt
```

اکنون می بینید که موارد تکرار شده از خروجی حذف شده و داده ها نیز مرتب شده اند.



مرتب سازی با استفاده از pipe در یک فرمان

اگر بخواهیم داده‌های فایل را با ارائه فهرستی از فهرست مربوط به اندازه فایل مرتب کنیم، همه داده‌های مربوط به فهرست را فهرست می‌کنیم ls .در دستور استفاده می شود و l- آن را نمایش می دهد Pipe .به نمایش فایل ها به صورت سازماندهی شده کمک می کند.

```
$ ls –l / home / aqsayasin /     sort –nk5
```

```
aqsayasin@virtualbox:~$ ls -l /home/aqsayasin/ | sort -nk5
total 64
-rw-rw-r-- 1 aqsayasin aqsayasin   20 18:07 15    جون file4.txt
-rw-rw-r-- 1 aqsayasin aqsayasin   58 11:54 17    جون sample2.txt
-rw-rw-r-- 1 aqsayasin aqsayasin   61 11:53 17    جون sample1.txt
-rw-rw-r-- 1 aqsayasin aqsayasin   61 11:59 17    جون result.txt
-rw-rw-r-- 1 aqsayasin aqsayasin   67 11:56 17    جون sample3.txt
-rw-rw-r-- 1 aqsayasin aqsayasin   72 12:13 17    جون file3.txt
-rw-rw-r-- 1 aqsayasin aqsayasin   89 18:31 15    جون File1.txt
-rw-rw-r-- 1 aqsayasin aqsayasin   95 17:49 15    جون file2.txt
drwxr-xr-x 2 aqsayasin aqsayasin 4096 12:00 15    جون Desktop
drwxr-xr-x 2 aqsayasin aqsayasin 4096 12:00 15    جون Documents
drwxr-xr-x 2 aqsayasin aqsayasin 4096 12:00 15    جون Downloads
drwxr-xr-x 2 aqsayasin aqsayasin 4096 12:00 15    جون Music
drwxr-xr-x 2 aqsayasin aqsayasin 4096 12:00 15    جون Pictures
drwxr-xr-x 2 aqsayasin aqsayasin 4096 12:00 15    جون Public
drwxr-xr-x 2 aqsayasin aqsayasin 4096 12:00 15    جون Templates
drwxr-xr-x 2 aqsayasin aqsayasin 4096 12:00 15    جون Videos
```

مرتب سازی تصادفی

گاهی اوقات، در حین انجام هر عملکردی، می توانید ترتیب را به هم بزنید .اگر می خواهید داده ها را در هر ترتیبی مرتب کنید و معیاری برای مرتب سازی وجود ندارد، مرتب سازی تصادفی ترجیح داده می شود .فایلی به نام sample3.txt را با نام قاره ها در نظر بگیرید.

```
aqsayasin@virtualbox:~$ cat sample3.txt
Asia
Africa
Australia
Antarctica
Europe
North Ameria
South America
```

۱. مرتب‌سازی خطوط یک فایل متنی به ترتیب صعودی:

```
sort file.txt
```

۲. مرتب‌سازی خطوط یک فایل متنی به ترتیب نزولی:

```
sort -r file.txt
```

۳. مرتب‌سازی خطوط یک فایل متنی و نادیده گرفتن حروف بزرگ و کوچک:

```
sort -f file.txt
```

۴. مرتب‌سازی خطوط یک فایل متنی بر اساس عدد ابتدایی هر خط:

```
sort -n file.txt
```

۵. مرتب‌سازی خطوط یک فایل متنی و حذف تکرارها:

```
sort -u file.txt
```

۶. مرتب‌سازی خطوط یک فایل متنی بر اساس ستون دوم:

```
sort -k 2 file.txt
```

۷. مرتب‌سازی خطوط یک فایل متنی بر اساس مقدار عددی ستون سوم:

```
sort -k 3n file.txt
```

۸. مرتب‌سازی خطوط یک فایل متنی بر اساس تاریخ و زمان:

```
sort -t' ' -k 3,3 file.txt
```

۹. مرتب‌سازی خطوط یک فایل متنی بر اساس ستون سوم به صورت منظم:

```
sort -t',' -k 3 file.txt
```

۱۰. مرتب‌سازی خطوط یک فایل متنی بر اساس مقدار عددی ستون چهارم به صورت نزولی:

```
sort -k 4nr file.txt
```

دستور comm

commدو فایل مرتب شده را خط به خط مقایسه کرده و در خروجی استاندارد مینویسد .خطوطی که مشترک هستند و خطوطی که منحصر به فرد هستند. فرض کنید شما دو لیست از افراد دارید و از شما خواسته می شود اسامی موجود در یکی را پیدا نکنید و در اسم دیگر نیست و یا حتی نام های مشترک هر دو است **comm** .دستوری است که به شما در دستیابی به این هدف کمک می کند .به دو فایل مرتب شده نیاز دارد که آنها را خط به خط مقایسه می کند.

```
$comm [OPTION]... FILE1 FILE2
```

با استفاده از comm ، ما سعی می کنیم دو پرونده را با هم مقایسه کنیم بنابراین دستور دستور comm به دو نام پرونده به عنوان آرگومان نیاز دارد. بدون استفاده از گزینه ، commخروجی سه ستونی تولید می کند که ستون اول شامل خطوط منحصر به FILE1 است ، ستون دوم شامل خطوط منحصر به فرد برای FILE2 و ستون سوم و آخرین شامل خطوط مشترک برای هر دو پرونده است. دستور comm فقط در صورتی درست کار می کند که دو پرونده از قبل مرتب شده را با هم مقایسه کنید.

مثال :فرض کنید دو فایل مرتب شده وجود دارد file1.txt :و file2.txt و اکنون برای مقایسه این دو از دستور comm استفاده خواهیم کرد.

نمایش محتویات فایل ۱

```
$ cat file1.txt
```

نمایش محتویات فایل ۱

```
$ file file2.txt
```

اکنون ، دستور comm را به صورت زیر اجرا کنید:

```
$ comm file1.txt file2.txt
```

گزینه های دستور:

- check-order -- بررسی می‌کند که ورودی به درستی مرتب شده باشد، حتی اگر همه خطوط ورودی جفت شوند.
- nocheck-order -- بررسی نمی‌کند که ورودی به درستی مرتب شده باشد.
- output-delimiter=STR -- ستون‌های جداگانه با رشته STR.
- help -- نمایش یک پیام راهنما و خروج.

توجه :از گزینه های ۴ تا ۸ بندرت استفاده می شود اما گزینه های ۱ تا ۳ از نظر میزان خروجی مورد نظر کاربر بسیار مفید هستند. با استفاده از گزینه **:**check-order- - این گزینه برای بررسی مرتب بودن یا نبودن پرونده های ورودی استفاده می شود و در صورتی که هر یک از این دو پرونده به اشتباه مرتب شده باشند ، دستور comm با یک پیام خطا از کار می افتد.

```
$comm - -check-order f1.txt f2.txt
```

مثال های دستور :

دستور comm در سیستم عامل‌های Unix و مشتقات آن مانند Linux برای مقایسه و نمایش تفاوت‌ها و تطابق‌های بین دو فایل متنی مرتب استفاده می‌شود. این دستور به کاربر امکان می‌دهد تا خطوط مشترک، خطوط منحصر به فرد هر فایل و خطوط موجود فقط در یکی از فایل‌ها را به صورت جداگانه نمایش دهد. comm معمولاً برای پیدا کردن تفاوت‌ها و تطابق‌های بین داده‌ها در دو فایل متنی به کار می‌رود.

مثال های دستور:

۱. نمایش خطوط مشترک بین دو فایل:

```
comm file1.txt file2.txt
```

۲. نمایش خطوط منحصر به فرد فایل اول:

```
comm -23 file1.txt file2.txt
```

۳. نمایش خطوط منحصر به فرد فایل دوم:

```
comm -13 file1.txt file2.txt
```

۴. نمایش خطوط مشترک و خطوط منحصر به فرد فایل اول:

```
comm -12 file1.txt file2.txt
```

۵. نمایش خطوط مشترک و خطوط منحصر به فرد فایل دوم:

```
comm -21 file1.txt file2.txt
```

۶. نمایش خطوط مشترک بدون خروجی‌های مشترک:

```
comm -3 file1.txt file2.txt
```

۷. نمایش خطوط مشترک بدون خروجی‌های فایل اول:

```
comm -12 -3 file1.txt file2.txt
```

۸. نمایش خطوط مشترک بدون خروجی‌های فایل دوم:

```
comm -21 -3 file1.txt file2.txt
```

۹. نمایش خطوط مشترک و خطوط منحصر به فرد هر دو فایل:

```
comm -3 file1.txt file2.txt
```

۱۰. نمایش خطوط مشترک و خطوط منحصر به فرد هر فایل با عنوان‌های سفید:

```
comm -123 file1.txt file2.txt
```

دستور ed

از دستور **ed** در Linux برای راه اندازی ویرایشگر متن **ed** استفاده می شود که یک ویرایشگر متن مبتنی بر خط است و دارای حداقل رابط کاربری است که باعث می شود کار در پرونده های متنی یعنی ایجاد ، ویرایش ، نمایش و دستکاری پرونده ها از پیچیدگی کمتری برخوردار باشد .این قدیمی ترین ویرایشگر ساخته شده در یونیکس است .توسط ویرایشگر متن **vi** و **emacs** موفق می شود.

ساختار دستور:

```
ed [options] [file]
```

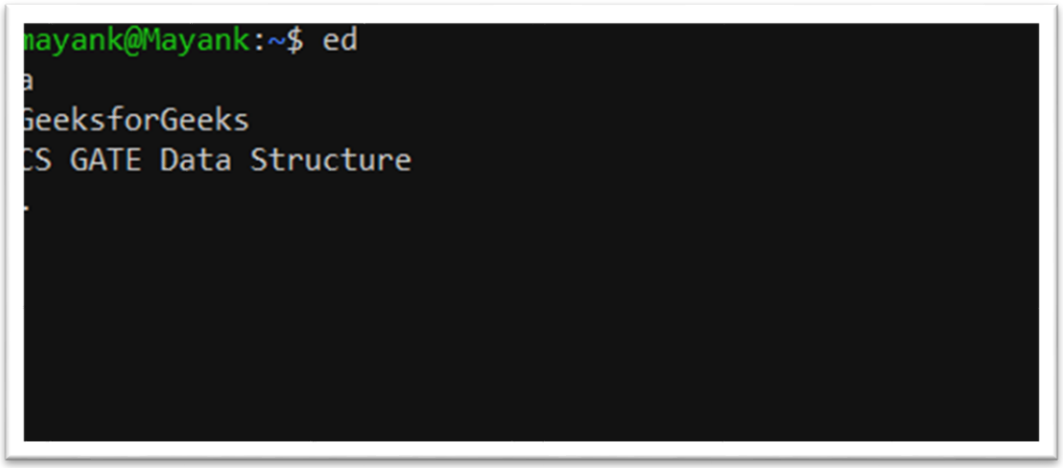
دستور ed بدون هیچ گزینه :ویرایشگر ed را با یک بافر خالی برای نوشتن ما راه اندازی می کند ، این همان ویرایشگر مبتنی بر خط فرمان یا تصویری دیگر است که بدون نام پرونده فراخوانی می شود .ساده ترین راه برای شروع ویرایشگر متن ویرایش ، تایپ کردن " **ed** "در ترمینال است.

مثال ۱:



در اینجا ما نام فایل را مشخص نکرده ایم بنابراین این یک بافر خالی برای نوشتن ایجاد می کند.

مثال ۲ :برای ورود به حالت درج ، " **a** "را فشار دهید و وقتی کار نوشتن تمام شد آن را با نقطه متوقف کنید.



مثال ۳ :برای مشاهده ، آخرین خط " **p** " را در خط فرمان ed وارد کنید.

```
mayank@Mayank:~$ ed
a
GeeksforGeeks
CS GATE Data Structure
.
p
CS GATE Data Structure
```

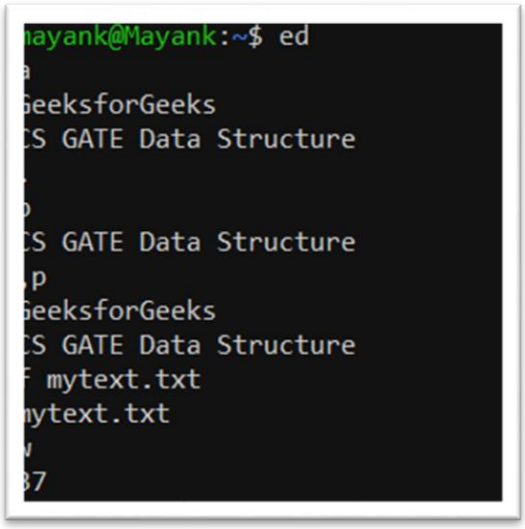
مثال ٤ :برای چاپ تمام خطوطی که با استفاده از " p" ، "در بافر قرار داده ایم.

```
mayank@Mayank:~$ ed
a
GeeksforGeeks
CS GATE Data Structure
.
p
CS GATE Data Structure
,p
GeeksforGeeks
CS GATE Data Structure
```

مثال ٥ :برای ذخیره این خطوط در یک پرونده ، "f [filename]"را بنویسید.

```
mayank@Mayank:~$ ed
a
GeeksforGeeks
CS GATE Data Structure
.
p
CS GATE Data Structure
,p
GeeksforGeeks
CS GATE Data Structure
f mytext.txt
mytext.txt
```

مثال ٦ " w " :را در خط فرمان ed تایپ کنید تا داده ها را در پرونده بنویسید و ببینید چند بایت نوشته شده است.



مثال ۷ :پرونده اکنون دارای داده است و ما می توانیم با فشار دادن "Q" در خط فرمان **ed** به ترمینال برویم.



دستور grep

فیلتر grep یک فایل را برای یک الگوی خاص از نویسه ها جستجو می کند و تمام خطوطی را که حاوی آن الگو هستند نمایش می دهد .از الگویی که در پرونده جستجو می شود به عنوان عبارت منظم یاد می شود (grep مخفف عبارت globally search for منظم و چاپ کردن است.

ساختار دستور:

```
grep [options] pattern [files]
```

- c-تعداد خطوطی که با یک الگو مطابقت دارند را چاپ می‌کند.
- h-خطوط همسان را نمایش می‌دهد، اما نام پرونده‌ها را نمایش نمی‌دهد.
- i-برای تطبیق، بدون توجه به حروف بزرگ و کوچک، عمل می‌کند.
- l-فقط لیست نام پرونده‌ها را نمایش می‌دهد.
- n-شماره خطها را نمایش دهید.
- v-تمام خطوطی که با الگو مطابقت ندارند را چاپ می‌کند.
- f file-الگوها را از پرونده دریافت می‌کند، یکی در هر خط.
- E-الگو را به عنوان یک عبارت منظم گسترده (ERE) در نظر می‌گیرد.
- w-تنها کلمات کامل را مطابقت دهید.
- o-فقط بخش‌های مطابق یک خط را چاپ می‌کند.

مثال بیشتر:

فایل زیر را به عنوان ورودی در نظر بگیرید.

```
$ cat > mypasswords.txt
```

جستجوی عدم حساسیت به پرونده :گزینه i- امکان جستجوی یک مورد رشته ای را به صورت غیر حساس در پرونده داده فراهم می کند .با کلماتی مانند "UNIX" ، "Unix"، "unix"مطابقت دارد.

```
$ grep -i “UNix”  mypasswords.txt
```

نمایش تعداد تعداد تطبیق ها :می توانیم تعداد خطوطی را که با رشته / الگوی داده شده مطابقت دارد پیدا کنیم

```
$ grep -c “unix”  mypasswords.txt
```

یا

```
$ grep -l “unix” *
```

یا

```
 $ grep -l “unix” f1.txt f2.txt f3.xt f4.txt
```

خروجی دستور:

```
mypasswords.txt
```

بررسی کل کلمات در یک پرونده :به طور پیش فرض ، grepبا رشته / الگوی داده شده مطابقت دارد حتی اگر به عنوان زیر رشته در یک فایل پیدا شود .گزینه w- برای grep باعث می شود که فقط با کل کلمات مطابقت داشته باشد.

```
$ grep -w “unix”  mypasswords.txt
```

نمایش فقط الگوی همسان :به طور پیش فرض ، grepکل خطی را که دارای رشته همسان است نشان می دهد .با استفاده از گزینه o- می توانیم grep را طوری نمایش دهیم که فقط رشته همسان را نشان دهد.

```
$ grep -o “unix”  mypasswords.txt
```

نمایش شماره خط هنگام نمایش خروجی با استفاده از :grep -n برای نشان دادن شماره خط پرونده با خط همسان.

```
$ grep -n “unix”  mypasswords.txt
```

معکوس کردن مطابقت الگو :با استفاده از گزینه v- می توانید خطوطی را که با الگوی جستجوی مشخص شده مطابقت ندارند نمایش میدهد.

```
$ grep -v “unix”  mypasswords.txt
```

تطبیق خطوطی که با یک رشته شروع می شوند :الگوی بیان منظم ^ شروع یک خط را مشخص می کند .این را می توان در grepبرای مطابقت با خطوطی که با رشته یا الگوی داده شده شروع می شوند ، استفاده کرد.

```
$ grep “^ unix”  mypasswords.txt
```

مطابقت خطوطی که با یک رشته خاتمه می یابند :الگوی بیان $ منظم انتهای یک خط را مشخص می کند .این را می توان در grepبرای مطابقت با خطوطی که با رشته یا الگوی داده شده مطابقت دارند استفاده کرد.

```
$ grep “os $”  mypasswords.txt
```

بیان را با گزینه e- مشخص می کند .می تواند چندین بار استفاده کند:

```
$ grep –e “Agarwal” –e “Aggarwal” - e “Agrawal”  mypasswords.txt
```

f- گزینه پرونده الگوها را از پرونده می گیرد ، یکی در هر خط.

```
$ cat pattern.txt
```

```
$ grep –f pattern.txt  mypasswords.txt
```

دستور egrep

در سیستم عامل‌های Unix/Linux برای جستجوی الگوهای با قابلیت استفاده از عبارات بازگردنده (Regular Expressions) در محتوای فایل‌ها استفاده می‌شود. این دستور از ترکیب دو دستور grep و egrep برای اجرای جستجوی پیشرفته‌تر و قوی‌تر استفاده می‌کند.

مثال های دستور:

۱. جستجوی الگوی خاص در یک فایل:

این دستور برای جستجوی یک الگوی خاص در محتوای یک فایل استفاده می‌شود.

```
egrep ‘pattern’ file.txt
```

۲. جستجوی الگوی خاص در تمام فایل‌های یک فهرست:

این دستور برای جستجوی یک الگوی خاص در تمام فایل‌های موجود در یک فهرست استفاده می‌شود.

```
egrep ‘pattern’ /path/to/directory/*
```

۳. جستجو با نمایش شماره خطوط:

این دستور برای جستجوی الگو با نمایش شماره خطوط مطابق استفاده می‌شود.

```
egrep -n ‘pattern’ file.txt
```

۴. جستجوی الگو برعکس:

این دستور برای جستجوی الگوی برعکس یک الگو خاص استفاده می‌شود.

```
egrep -v ‘pattern’ file.txt
```

۵. جستجوی الگو با نمایش خطوط مطابق:

این دستور برای جستجوی الگو با نمایش خطوط مطابق با الگو استفاده می‌شود.

```
egrep -o ‘pattern’ file.txt
```

۶. جستجو با عبارت بازگردنده پیچیده:

این دستور برای جستجو با استفاده از عبارات بازگردنده پیچیده برای الگو استفاده می‌شود

```
egrep -E ‘pattern1|pattern2’ file.txt
```

دستور exec

دستور **exec** در Linux برای اجرای یک دستور از خود bash استفاده می شود .این دستور فرایند جدیدی ایجاد نمیکند بلکه فقط دستور اجرا شده را جایگزین bash می کند .اگر دستور exec موفقیت آمیز باشد ، به فرایند فراخوانی بر نمی گردد.

ساختار دستور:

```
exec [-cl] [-a name] [command [arguments]] [redirection ...]
```

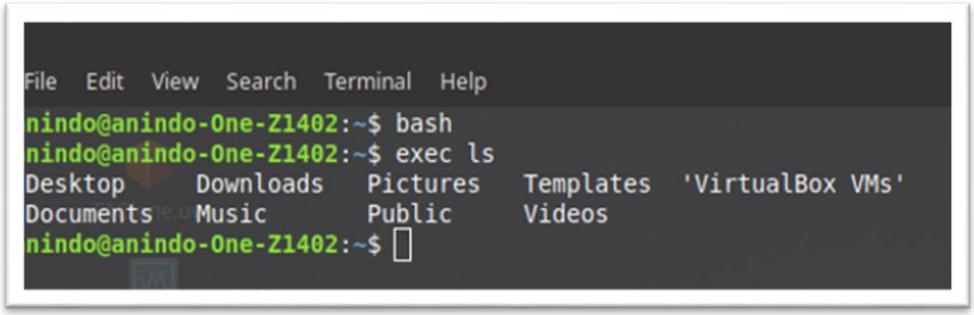
گزینه های دستور:

- **c** برای اجرای دستور با محیط خالی استفاده می شود.
- **a name** برای گذراندن نام به عنوان آرگومان صفر دستور استفاده می شود.
- **l** برای عبور خط تیره به عنوان آرگومان صفر دستور استفاده می شود.

توجه :دستور *exec* فرآیند جدیدی ایجاد نمی کند .هنگامی که ما دستور exec را از ترمینال اجرا می کنیم ، فرایند ترمینال در حال اجرا با فرمانی که به عنوان آرگومان دستور exec ارائه می شود ، جایگزین می شود. از دستور exec می توان در دو حالت استفاده کرد:

Execبا یک دستور به عنوان آرگومان :در حالت اول ، execسعی می کند آن را به عنوان دستوری اجرا کند که آرگومان های باقیمانده را در صورت وجود به آن دستور منتقل کرده و در صورت وجود تغییر مسیرها را مدیریت کند.

مثال ۱:



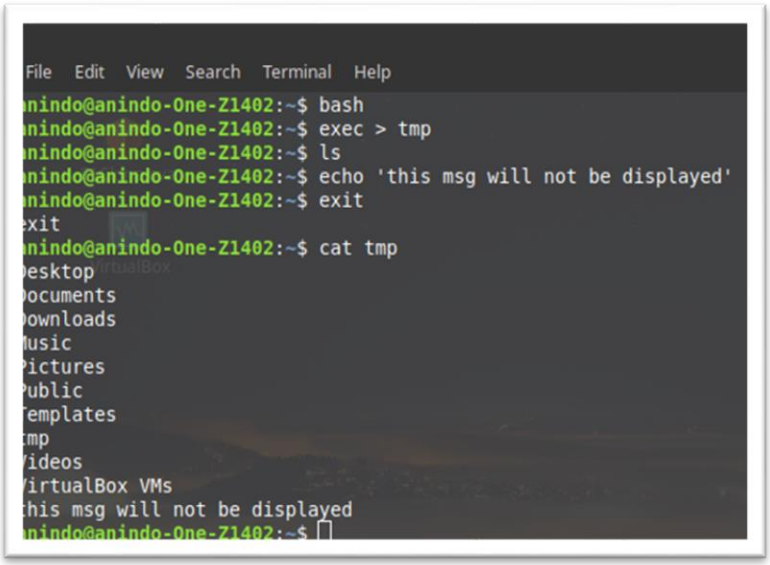
مثال ۲:



دستور exec مسیر ذکر شده در متغیر *PATH $* را جستجو می کند تا دستوری را برای اجرا پیدا کند .اگر دستور پیدا نشد ، دستور execو همچنین پوسته در یک خطا خارج می شوند.

اجرای بدون دستور :اگر هیچ دستوری ارائه نشود ، می توان از تغییر مسیرها برای تغییر محیط پوسته فعلی استفاده کرد .این مفید است زیرا به ما امکان می دهد توصیف کننده پرونده پوسته را مطابق میل خود تغییر دهیم .این روند حتی پس از دستور exec بر خلاف حالت قبلی ادامه دارد اما اکنون ورودی ، خروجی و خطای استاندارد با توجه به تغییر مسیرها اصلاح می شود.

مثال :



در اینجا دستور exec استاندارد را از پوسته به پرونده tmp تغییر می دهد و بنابراین تمام دستورات اجرا شده پس از دستور exec نتایج خود را در آن پرونده می نویسند .این یکی از متداول ترین روش های استفاده از exec بدون هیچ گونه دستوری است.

مثالهای بیشتر

۱. اجرای یک دستور خاص به جای پروسه جاری:

```
exec ls -l
```

۲. اجرای یک اسکریپت shell به جای پروسه جاری:

```
exec ./myscript.sh
```

۳. اجرای یک برنامه جایگزین با استفاده از مسیر مطلق:

```
exec /usr/bin/python3 myprogram.py
```

۴. اجرای یک برنامه جایگزین با پاس دادن آرگومان‌ها:

```
exec myprogram arg1 arg2
```

۵. اجرای یک برنامه جایگزین و بستن تمام فایل‌های باز در پروسه جاری:

```
exec myprogram <&- >&- 2>&-
```

۶. اجرای یک برنامه جایگزین و تغییر شماره پردازه:

```
exec -a myprocess myprogram
```

۷. اجرای یک برنامه جایگزین و تغییر فایل‌های ورودی و خروجی دستور:

```
exec < input.txt > output.txt myprogram
```

۸. اجرای یک برنامه جایگزین و تنظیم متغیرهای محیطی:

```
exec -l myprogram
```

۹. اجرای یک برنامه جایگزین در پس‌زمینه:

```
exec myprogram &
```

۱۰. اجرای یک برنامه جایگزین و تغییر نام پروسه:

```
 exec -l myprogram
```

دستور exo-open

استفاده از برنامه پیش فرض در محیط دسکتاپ استفاده می شود. برای همه گزینه های این دستور، مثال های زیر را می توان ارائه داد:

```
$exo-open file.txt
```

این دستور فایل متنی با نام "file.txt" را باز می کند.

```
$exo-open /home/user/Documents
```

این دستور فهرست "Documents" را در مسیر "/home/user" باز می کند.

```
$exo-open https://www.seilany.ir
```

این دستور وبسایت "www.seilany.ir" را در مرورگر پیش فرض سیستم باز می کند.

```
$exo-open image.jpg
```

این دستور تصویر با نام "image.jpg" را باز می کند، در صورتی که برنامه پیش فرض برای باز کردن تصاویر تنظیم شده باشد. توجه داشته باشید که ممکن است نیاز به نصب برنامه exo-open بر روی سیستم شما باشد تا از این دستور استفاده کنید.

دستور exit

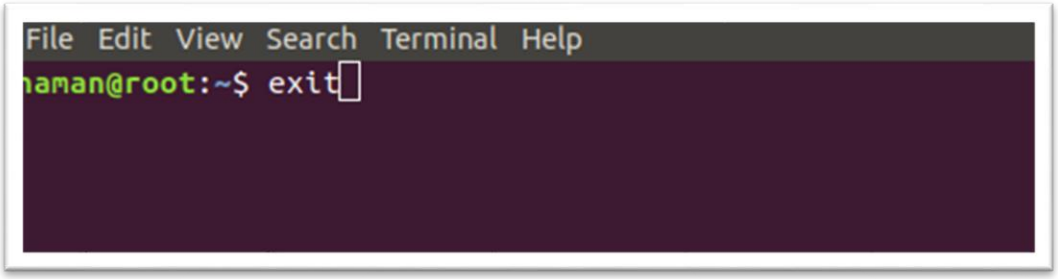
از دستور exit در linux برای خروج از پوسته ای که در حال حاضر در آن اجرا می شود استفاده می شود .یک پارامتر دیگر به عنوان [N] طول می کشد و با بازگشت وضعیت N از پوسته خارج می شود . اگر n ارائه نشده باشد ، به سادگی وضعیت آخرین دستوری را که اجرا می شود برمی گرداند.

ساختار دستور:

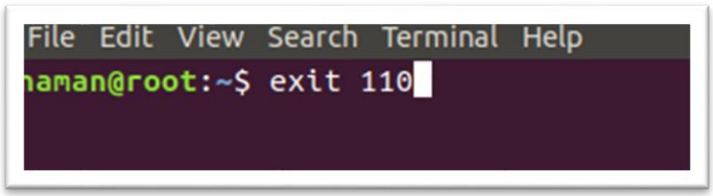
```
exit [n]
```

گزینه های دستور:

exit:خروج بدون پارامتر



exit [n]:خروج با پارامتر



پس از فشار دادن enter ، پنجره ترمینال بسته می شود و وضعیت ۱۱۰ را برمی گرداند. وضعیت بازگشت مهم است زیرا گاهی اوقات می توان آنها را برای تشخیص خطا ، هشدارها و اعلان ها ترسیم کرد .به عنوان مثال

- وضعیت بازگشت:"0"به این معنی است که برنامه با موفقیت اجرا شده است.
- "1"به معنای این است که برنامه دارای خطاهای جزئی است.
- exit n:با استفاده از "sudo su" ما به فهرست ریشه می رویم و سپس از فهرست ریشه با وضعیت بازگشت ۵ خارج می شویم. پس از بازگشت ، نحوه نمایش کد وضعیت بازگشت نشان داده می شود.

```
File  Edit  View  Search  Terminal  Help
naman@root:~$ sudo su
[sudo] password for naman:
root@root:/home/naman# exit 5
exit
naman@root:~$ echo $?
5
naman@root:~$
```

```
echo $?
```

از این دستور برای دیدن آخرین وضعیت بازگشت استفاده می شود.

```
exit –help
```

اطلاعات راهنما را نمایش می دهد.

```
naman@root:~$ exit --help
exit: exit [n]
    Exit the shell.

    Exits the shell with a status of N.  If N is omitted, the exit status
    is that of the last command executed.
naman@root:~$
```

مقایسه فایل ها

به طور معمول، برای مقایسه دو فایل در linux، از diffاستفاده می کنیم. یک ابزار خط دستور یونیکس ساده و اصلی است که تفاوت بین دو فایل کامپیوتری را نشان می دهد، فایل ها را به صورت خطی مقایسه می کند و از آن آسان است که استفاده می شود، با بسیاری از توزیع های linux نصب می شود .سوال این است که چگونه می توانیم بین دو فهرست در linux تفاوت ایجاد کنیم؟در اینجا، ما می خواهیم بدانیم که کدام فایل ها / زیر شاخه ها در دو فهرست مشترک هستند، کسانی که در یک فهرست موجود هستند، اما نه در فهرست.

دستور diff

دستورالعمل متعارف برای اجرای diff به شرح زیر است:

```
$ diff [option] ... files
$ diff dir1 dir2
```

به طور پیش فرض، خروجی آن به ترتیب حروف الفبا توسط نام فایل / زیر شاخه به ترتیب نشان داده شده است.در این دستور، q– گزارش را فقط زمانی که پرونده ها متفاوت هستند، گزارش می کند.

```
$ diff -q directory-1 / directory-2 /
```

دستور diff به subdirectories عمل نمیکند، اما می توانیم از سویچ r-برای خواندن subdirectories ها نیز استفاده کنیم.

```
$ diff -qr directory-1 / directory-2 /
```

دستور diff در سیستم عامل‌های Unix و مشتقات آن مانند Linux برای مقایسه محتوای دو فایل متنی به کار می‌رود. این دستور تفاوت‌های بین دو فایل را نمایش می‌دهد و به کاربر امکان می‌دهد تا خطوطی که در یک فایل وجود دارند اما در فایل دیگر نیستند یا برعکس را مشاهده کند.

مثال های دستور:

۱. مقایسه دو فایل و نمایش تفاوت‌ها به صورت خروجی استاندارد:

```
diff file1.txt file2.txt
```

۲. مقایسه دو فایل و نمایش تفاوت‌ها به صورت کامل (تفاوت‌ها و تطابق‌ها):

```
diff -u file1.txt file2.txt
```

۳. مقایسه دو فایل و نمایش تنها خطوط متفاوت:

```
diff -q file1.txt file2.txt
```

۴. مقایسه دو فایل و نمایش تفاوت‌ها به صورت کوتاه:

```
diff -c file1.txt file2.txt
```

۵. مقایسه دو فهرست و نمایش تفاوت‌ها:

```
diff -r directory1 directory2
```

۶. مقایسه دو فایل و ذخیره نتیجه در یک فایل خروجی دستور:

```
diff file1.txt file2.txt > differences.txt
```

۷. مقایسه دو فایل و نمایش تفاوت‌ها به صورت خط به خط:

```
diff -y file1.txt file2.txt
```

۸. مقایسه دو فایل و نمایش تغییرات به صورت خلاصه:

```
diff -e file1.txt file2.txt
```

۹. مقایسه دو فایل و نمایش تفاوت‌ها به صورت کلمه به کلمه:

```
diff -w file1.txt file2.txt
```

۱۰. مقایسه دو فایل و نمایش تفاوت‌ها به صورت باز و بسته:

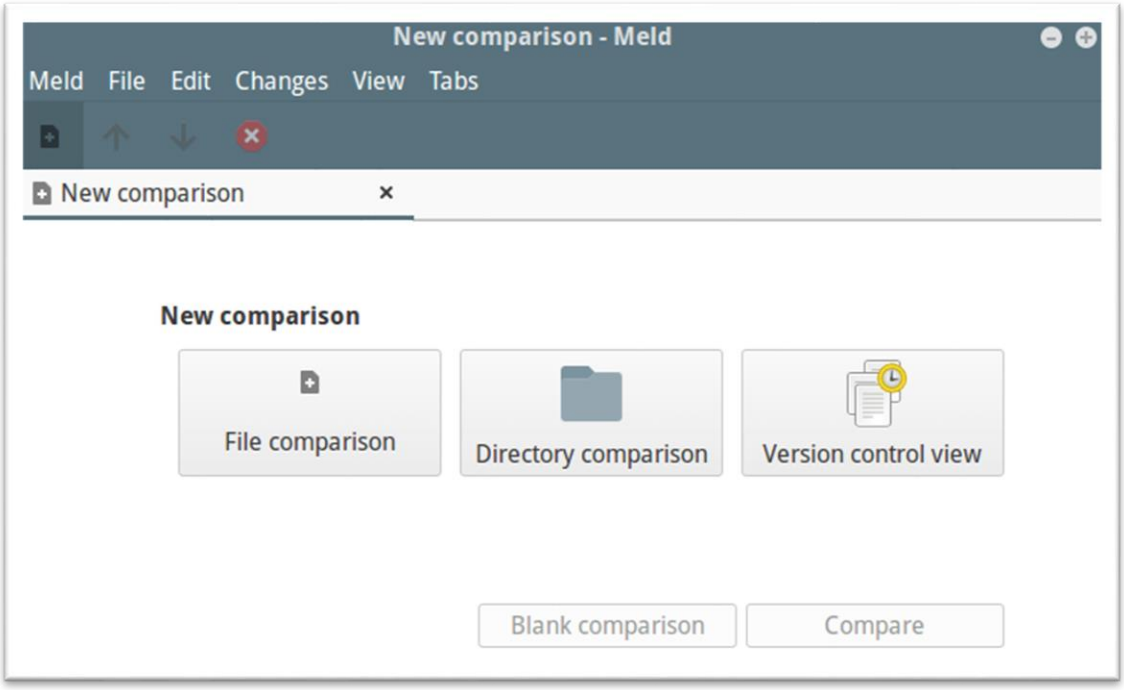
```
diff -B file1.txt file2.txt
```

ابزار Meld

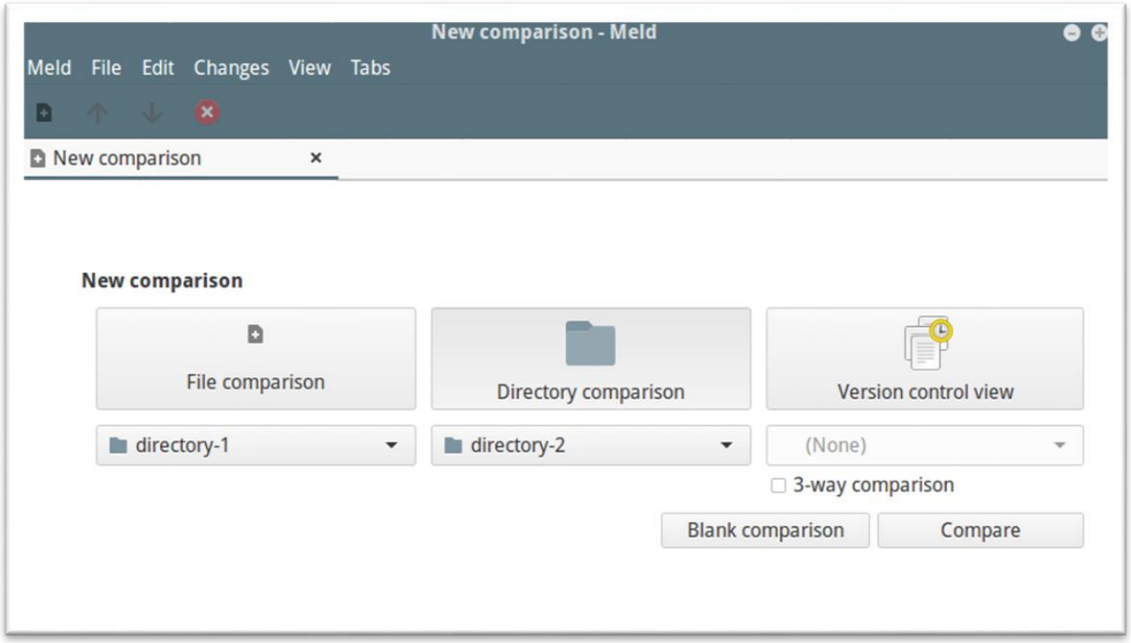
یک ابزار گرافیکی meld یک ابزار گرافیکی برای مشاهده تفاوت ها است که می توانید آن را به صورت زیر نصب کنید.داخلش کروشه[۱] توضیح است و جز دستور نیست. روش نصب:

```
$ sudo apt install meld    [system / های دبیان ubuntu]
$ sudo yum install meld      [systemهای  RHEL / CentOS]
$ sudo dnf install meld    [+فدورا ۲۲]
```

در ادامه رابط گرافیکی برنامه Meld را میتوانیدببینید، جایی که می توانید فایل یا فهرست را مقایسه کنید.

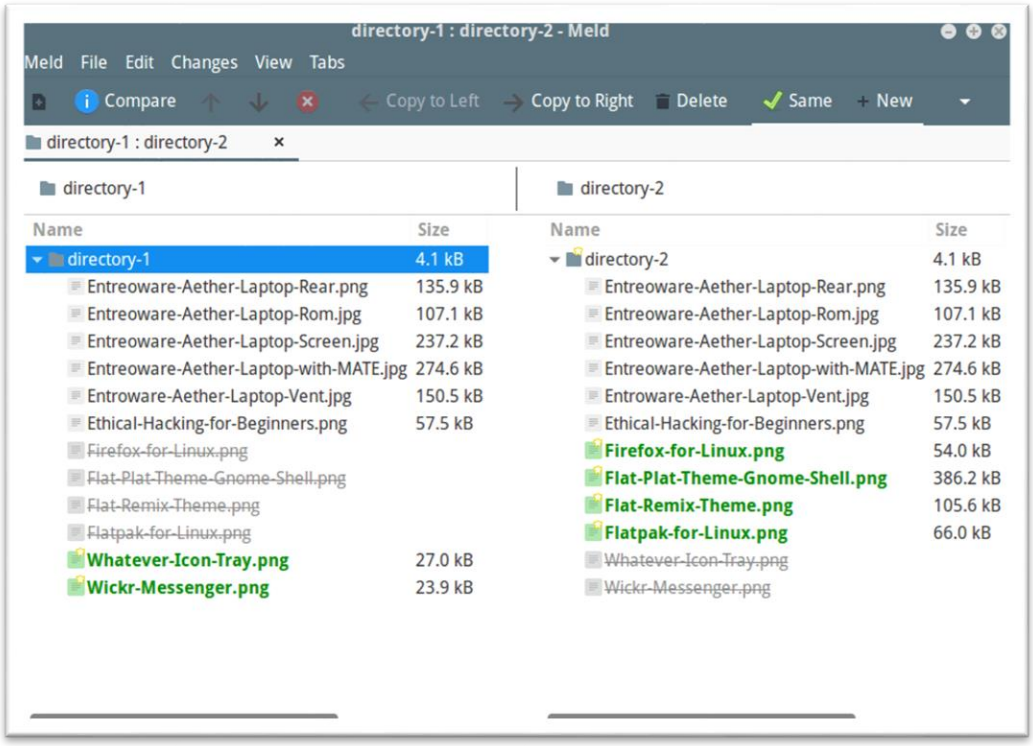


ابزار مقایسه **Meld**



[۱] Brackets

پس از انتخاب فهرست ها، بر روی " مقایسه "کلیک کنید.



دستور meld

Meld یک ابزار گرافیکی مقایسه و ادغام فایل‌ها در محیط گرافیکی برای سیستم عامل‌های Unix و مشتقات آن مانند Linux است. این ابزار به کاربر امکان می‌دهد تا دو یا چند فایل را مقایسه کرده و تفاوت‌ها را به صورت دیداری ویرایش کند.

مثال های دستور:

۱. باز کردن Meld و مقایسه دو فایل:

```
meld file1.txt file2.txt
```

۲. مقایسه دو فهرست و نمایش تفاوت‌ها:

```
meld directory1 directory2
```

۳. ادغام تغییرات از چند فایل به یک فایل:

```
meld file1.txt file2.txt file3.txt
```

۴. استفاده از رابط گرافیکی Meld برای مقایسه و ادغام فایل‌ها:

```
meld
```

۵. نمایش تغییرات با استفاده از رنگ‌های مختلف در Meld:

```
meld --color file1.txt file2.txt
```

۶. استفاده از فیلترها برای نمایش تغییرات مشخص در Meld:

```
meld --filter=regexp file1.txt file2.txt
```

۷. نمایش تغییرات به صورت خط به خط در Meld:

```
meld --diff file1.txt file2.txt
```

۸. نمایش تاریخچه تغییرات و بازنگری تغییرات در Meld:

```
meld --history file1.txt file2.txt
```

۹. انتخاب قالب نمایش تغییرات در Meld:

```
meld --layout tabbed file1.txt file2.txt
```

۱۰. نمایش تغییرات به صورت ساختار گرافیکی در Meld:

```
meld --graph file1.txt file2.txt
```

دستور cmp

از دستور **cmp** در Linux برای مقایسه دو فایل بایت به بایت استفاده می شود و به شما کمک میکند تا بفهمید این دو پرونده یکسان هستند یا نه. هنگامی که cmp برای مقایسه بین دو فایل استفاده می شود ، در صورت مشاهده تفاوت و عدم تفاوت یافتن، به عنوان مثال فایلهای مقایسه شده یکسان هستند ، محل عدم تطابق اول را به صفحه نمایش می دهد. cmpهیچ پیامی را نمایش نمی دهد و اگر پرونده های مقایسه شده یکسان باشد، به راحتی دستور را برمی گرداند.

```
cmp [OPTION] ... FILE1 [FILE2 [SKIP1 [SKIP2]]]
```

SKIP1 ، SKIP2و OPTION اختیاری هستند .

و FILE1 و FILE2 به نام پرونده ها اشاره می کنند. **نحو** دستور cmp کاملاً ساده است .اگر ما در حال مقایسه دو پرونده هستیم ، بدیهی است که به نام آنها به عنوان آرگومان به عنوان FILE1 و FILE2 نیاز خواهیم داشت .علاوه بر این ، SKIP1و SKIP2 اختیاری تعداد بایت هایی را برای پرش در ابتدای هر فایل مشخص می کند که به طور پیش فرض صفر است و OPTION به گزینه های سازگار با این دستور اشاره می کند که در ادامه در مورد آنها بحث خواهیم کرد .همانطور که توضیح داده شد اگر cmp تفاوت پیدا کند ، بایت و شماره خط را گزارش می کند .حال بیایید با کمک یک مثال همان را دریابیم .فرض کنید دو فایل وجود دارد که می خواهید آنها را مقایسه کنید یکی file1.txt و دیگری file2.txt است:

```
$ cmp file1.txt file2.txt
```

۱. **اگر پرونده ها یکسان نباشند :**خروجی دستور بالا خواهد بود:

```
$ cmp file1.txt file2.txt
```

file1.txt file2.txtمتفاوت است: بایت ۹ ، خط ۲

۲. **اگر فایل ها یکسان باشند :**چیزی شبیه به این را در صفحه خود مشاهده خواهید کرد:

```
$ cmp file1.txt file2.txt
```

-b (print-bytes) :اگر می خواهید cmp در صورت استفاده با گزینه **-b** ، بایتهای مختلف خروجی را نمایش می دهد.

دستور cmp با گزینه b- استفاده می شود

```
$ cmp -b file1.txt file2.txt
```

-i [bytes-to-skip] :اکنون ، این گزینه هنگامی که با دستور cmp استفاده می شود کمک می کند تا تعداد مشخصی از بایت های اولیه را از هر دو پرونده رد کنید و سپس پس از جستجوی فایل ها ، آنها را مقایسه می کند .این کار را می توان با تعیین تعداد بایت ها به عنوان آرگومان به گزینه خط فرمان i- انجام داد.

```
$ cmp -i 10 file1.txt file2.txt
```

توجه داشته باشید که در مواردی از این قبیل (که برای رد کردن بایت ها از i- استفاده می کنید) ، بیتی که در آن مقایسه آغاز می شود به عنوان بایت شماره صفر در نظر گرفته می شود.

-l این گزینه باعث می شود که دستور cmp موقعیت بایت و مقدار بایت را برای همه بایت های مختلف چاپ کند.

```
$ cmp -l file1.txt file2.txt
20 12 56
21 124 12
22 150 124
150 151 150
241 163 151
25 40 163
26 146 40
27 150 151
28 12 24
29 124 145
30 157 163
```

اولین ستون در خروجی نشان دهنده موقعیت (تعداد بایت) بایت های مختلف است .ستون دوم مقدار بایت متفاوت بایت را در پرونده اول نشان می دهد ، در حالی که ستون سوم مقدار بایت متفاوت بایت را در پرونده دوم نشان می دهد.

سویچ **-s** این امکان را برای شما فراهم می کند تا خروجی تولید شده به طور معمول توسط دستور cmp را سرکوب کنید یعنی دو فایل را بدون نوشتن پیام ارسال می کند .این در صورت یکسان بودن پرونده ها مقدار ۰ ، در صورت متفاوت بودن مقدار ۱ یا در صورت بروز پیغام خطا مقدار ۲ می دهد.

```
$ cmp -s file1.txt file.txt
```

```
1
```

سویچ **-n .تعداد بایت برای مقایسه** :این گزینه به شما امکان می دهد تعداد بایت هایی را که می خواهید مقایسه کنید محدود کنید ، مثل این که فقط نیاز به مقایسه حداکثر ۲۵ یا ۵۰ بایت باشد.

```
$ cmp -n 50 file1.txt file2.txt
```

سویچ **-v:** - این اطلاعات خروجی را می دهد و خارج می شود.

سویچ **-help:** - این یک پیام **راهنما را** نمایش می دهد و خارج می شود.

دستور command

command برای مقایسه دو فایل مرتب شده به صورت خطی به صورت زیر نشان داده شده است.

```
$ commnd file1 file2
```

در حالی که یادگیری linux، همیشه برای تازه کارها عادی است که دستورات متعددی را برای انجام یک کار ساده انجام می دهند.این قابل درک است به ویژه هنگامی که کسی فقط به استفاده از ترمینال عادت کرده است.

دستور wc

دستور wc (word count)در system عامل های یونیکس / linux برای find تعداد شمارهٔ خط جدید ، شمارش کلمه ، بایت و شمارش کاراکتر در فایل های مشخص شده توسط های فایل استفاده می شود.نحو دستور wcهمانطور که در زیر نشان داده شده است.

```
# wc [options] نام فایل
```

بنابراین، بیایید ببینیم چگونه می توانیم از دستور wcاستفاده کنیم با تعداد چند استدلال و نمونه های موجود در این کتاب.ما از فایل hosein .txtبرای آزمایش دستورات استفاده کرده ایم.بیایید خروجی فایل را با استفاده از دستورcat به زیر نشان بدهیم.

```
$ sudo cat  hosein .txt
```

مثال اصلی از دستورWC

دستور ' wc ' بدون عبور از هر پارامتری یک نتیجه اولیه از فایل hosein .txtرا نمایش می دهد.سه شماره زیر نشان داده شده در زیر 12تعداد خطوط( ) 16تعداد کلمات (و) 112تعداد بایت (فایل است.

```
$ sudo wc  hosein .txt
```

1216112 hosein .txt

تعداد خطوط

برای شمارش تعداد خطوط جدید در یک فایل از گزینه ' l- ' استفاده کنید، که تعداد خطوط را از یک فایل داده شده چاپ می کند. ، دستور زیر تعداد خطوط جدید در یک فایل را نمایش می دهد.در خروجی اولین فیلد به عنوان شمارش اختصاص داده شده و فیلد دوم نام فایل است.

```
$ sudo wc -l  hosein .txt
```

12 hosein .txt

نمایش تعداد واژگان

با استفاده از آرگومان " w- " با دستور " wc " تعداد کلمات را در یک فایل چاپ می کند.دستور زیر را برای شمارش کلمات در یک فایل تایپ کنید.

```
$ sudo wc -w  hosein .txt
```

16 hosein .txt

شمارش تعداد بایت ها و کاراکتر ها

هنگام استفاده از گزینه های ' c- ' و ' m- ' با دستور wc، تعداد کل بایت ها و کاراکتر ها در یک فایل به ترتیب انجام می شود.

```
$ sudo wc -c  hosein .txt
```

112 hosein .txt

```
$ sudo wc-m  hosein .txt
```

112 hosein .txt

نمایش طولانی ترین خط

دستور wcبه استدلال " L- " اجازه می دهد، آن را می توان برای چاپ طول طولانی ترین خط در یک فایل استفاده می شود.

```
$ sudo wc -L  hosein .txt
16  hosein .txt
```

برای اطلاعات بیشتر و کمک در دستور wc، ساده ' wc -help ' یا ' man wc ' را از خط دستور اجرا کنید.

```
$ sudo wc   --help
```

```
استفاده: wc [OPTION] ... [FILE] ...
```

یا: wc [OPTION] ... --files0-from = F

تعداد خطوط جدید، کلمه و بایت را برای هر FILE چاپ می کنید.

دستور bzcmp

دستور bzcmp در سیستم‌عامل‌های مبتنی بر Unix/Linux برای مقایسه محتوای دو فایل فشرده شده با فرمت Bzip2 استفاده می‌شود.

گزینه‌های دستور:

بدون گزینه: مقایسه دو فایل فشرده Bzip2 و نمایش خروجی در صورت تفاوت.

```
bzcmp file1.bz2 file2.bz2
```

خروجی دستور: نمایش تفاوت‌ها بین دو فایل فشرده

--silent: ,sعدم نمایش خروجی در صورت تطابق محتوای دو فایل.

```
bzcmp -s file1.bz2 file2.bz2
```

خروجی دستور: بدون نمایش هیچ خروجی در صورت تطابق محتوای دو فایل

--brief: ,نمایش پیام خروجی کوتاه در صورت تفاوت محتوای دو فایل.

```
bzcmp -q file1.bz2 file2.bz2
```

خروجی دستور: نمایش پیام کوتاه در صورت تفاوت محتوای دو فایل

--list: ,lنمایش تمام خطوطی که تفاوت دارند.

```
bzcmp -l file1.bz2 file2.bz2
```

خروجی دستور: نمایش تمام خطوطی که تفاوت دارند بین دو فایل

--help: ,hنمایش راهنمای استفاده از دستور bzcmp.

```
bzcmp -h
```

خروجی دستور: نمایش راهنمای استفاده از دستور bzcmp

--version: ,نمایش نسخه فعلی دستور bzcmp.

```
bzcmp -V
```

خروجی دستور: نمایش نسخه فعلی دستور bzcmp

دستور bzdiff

دستور bzdiff در سیستم‌عامل‌های مبتنی بر Unix/Linux برای مقایسه و نمایش تفاوت‌های بین دو فایل فشرده با فرمت Bzip2 استفاده می‌شود.

گزینه‌های دستور:

بدون گزینه: مقایسه دو فایل فشرده Bzip2 و نمایش تفاوت‌ها به صورت متنی.

```
bzdiff file1.bz2 file2.bz2
```

خروجی دستور: نمایش تفاوت‌ها بین دو فایل فشرده به صورت متنی

--context: ,نمایش خطوط محیطی ( )contextحول خطوط با تفاوت.

```
bzdiff -c file1.bz2 file2.bz2
```

خروجی دستور: نمایش خطوط محیطی حول خطوط با تفاوت بین دو فایل

--unified: ,نمایش تفاوت‌ها به صورت یکپارچه و قابل خواندن برای ادیتورها.

```
bzdiff -u file1.bz2 file2.bz2
```

خروجی دستور: نمایش تفاوت‌ها به صورت یکپارچه و قابل خواندن برای ادیتورها

--brief: ,qنمایش پیام خروجی کوتاه در صورت تفاوت محتوای دو فایل.

```
bzdiff -q file1.bz2 file2.bz2
```

خروجی دستور: نمایش پیام کوتاه در صورت تفاوت محتوای دو فایل

--help: ,نمایش راهنمای استفاده از دستور bzdiff.

```
bzdiff -h
```

خروجی دستور: نمایش راهنمای استفاده از دستور bzdiff

--version: ,Vنمایش نسخه فعلی دستور bzdiff.

```
bzdiff -V
```

خروجی دستور: نمایش نسخه فعلی دستور bzdiff

دستور uniq

دستور uniq در سیستم عامل‌های Unix و مشتقات آن مانند Linux برای حذف خطوط تکراری متوالی از ورودی استفاده می‌شود. این دستور به کاربر این امکان را می‌دهد که خطوط تکراری را تشخیص داده و حذف کند.

مثال های دستور:

۱. حذف خطوط تکراری متوالی از ورودی:

```
uniq file.txt
```

۲. حذف خطوط تکراری متوالی و نمایش تعداد تکرارها:

```
uniq -c file.txt
```

۳. حذف خطوط تکراری متوالی و نمایش تعداد تکرارها بر اساس تعداد:

```
uniq -d file.txt
```

۴. حذف خطوط تکراری متوالی و نمایش خطوط غیرتکراری:

```
uniq -u file.txt
```

۵. حذف خطوط تکراری متوالی و نمایش تنها تکرارها:

```
uniq -D file.txt
```

۶. حذف خطوط تکراری متوالی و نمایش تکرارها به همراه تعداد تکرار:

```
uniq -dc file.txt
```

۷. حذف خطوط تکراری متوالی و نمایش تکرارها به همراه تعداد تکرار بر اساس تعداد:

```
uniq -f 2 file.txt
```

۸. حذف خطوط تکراری متوالی و نمایش تکرارها با توجه به تعداد اولیه خطوط:

```
uniq -s 5 file.txt
```

۹. حذف خطوط تکراری متوالی و نمایش تکرارها بدون در نظر گرفتن حروف بزرگ و کوچک:

```
uniq -i file.txt
```

۱۰. حذف خطوط تکراری متوالی و نمایش تکرارها بر اساس تعداد و حروف بزرگ و کوچک:

```
uniq -c -i file.txt
```

دستور expand

دستور expand در لینوکس برای تبدیل فاصله‌های هرزبرداشت شده به تب‌ها tabاستفاده می‌شود. این دستور به کاربر امکان می‌دهد تا فاصله‌های موجود در یک فایل را به تب‌ها تبدیل کرده و نمایش دهد.

گزینه‌های دستور:

- N: – tتعیین عرض تب به N کاراکتر
- i: – iاجازه تبدیل فاصله‌های درونی ( )internal spacesبه تب
- ۸–: استفاده از فاصله به جای تب (به عنوان ۸ فاصله)
- i -t N: – iترکیب دو گزینه بالا

مثال های دستور:

تبدیل فاصله‌ها به تب با عرض تب ۴ کاراکتر:

```
expand -t 4 file.txt
```

تبدیل فاصله‌های درونی به تب:

```
expand -i file.txt
```

استفاده از فاصله به جای تب (۸ فاصله):

```
expand -8 file.txt
```

ترکیب دو گزینه بالا (تبدیل فاصله‌های درونی به تب با عرض تب ۶ کاراکتر):

```
expand -i -t 6 file.txt
```

دستور tr

دستور tr در سیستم عامل‌های Unix و مشتقات آن مانند Linux برای تغییر یا حذف کاراکترها در ورودی استفاده می‌شود. این دستور به کاربر این امکان را می‌دهد که با استفاده از الگوها، کاراکترهای مورد نظر را جایگزین کند یا حذف کند.

مثال های دستور:

۱. جایگزینی کاراکترهای انتخاب شده با کاراکتر مورد نظر:

```
echo “hello” | tr ‘l’ ‘L’
```

۲. جایگزینی کاراکترهای انتخاب شده با تبدیل به حروف بزرگ:

```
echo “hello” | tr ‘[:lower:]’ ‘[:upper:]’
```

۳. حذف کاراکترهای انتخاب شده:

```
echo “hello, world!” | tr -d ‘[:punct:]’
```

۴. جایگزینی کاراکترهای انتخاب شده با کاراکترهای انتخاب شده دیگر:

```
echo “12345” | tr ‘123’ ‘abc’
```

۵. جایگزینی کاراکترهای انتخاب شده با کاراکترهای انتخاب شده دیگر و حذف کاراکترهای اضافی:

```
echo “apple, banana, cherry” | tr -s ‘, ‘ ‘\n’
```

۶. جایگزینی کاراکترهای انتخاب شده با کاراکترهای انتخاب شده دیگر و تبدیل به حروف بزرگ:

```
echo “hello, world!” | tr ‘[:lower:]’ ‘[:upper:]’
```

۷. جایگزینی کاراکترهای انتخاب شده با کاراکترهای انتخاب شده دیگر و حذف کاراکترهای اضافی:

```
echo “apple, banana, cherry” | tr -s ‘, ‘ ‘\n’
```

۸. جایگزینی کاراکترهای انتخاب شده با کاراکترهای انتخاب شده دیگر و تبدیل به حروف بزرگ:

```
echo “hello, world!” | tr ‘[:lower:]’ ‘[:upper:]’
```

۹. حذف کاراکترهای انتخاب شده و نگه‌داشتن کاراکترهای دیگر:

```
echo “123-456-7890” | tr -d ‘-’
```

۱۰. جایگزینی کاراکترهای انتخاب شده با کاراکترهای انتخاب شده دیگر و تبدیل به حروف کوچک:

```
echo “HELLO” | tr ‘[:upper:]’ ‘[:lower:]’
```

دستور split

دستور split در سیستم عامل‌های Unix و مشتقات آن مانند Linux برای تقسیم کردن یک فایل بزرگ به بخش‌های کوچکتر استفاده می‌شود. این دستور به کاربر این امکان را می‌دهد که یک فایل را براساس تعداد خطوط یا براساس اندازه بایتی مشخص شده تقسیم کند.

مثال‌های دستور:

۱. تقسیم فایل بر اساس تعداد خطوط (مثال: تقسیم یک فایل به بخش‌های حاوی ۱۰ خط):

```
split -l 10 file.txt
```

۲. تقسیم فایل بر اساس اندازه بایتی (مثال: تقسیم یک فایل به بخش‌های حداکثر ۱ مگابایت):

```
split -b 1M file.txt
```

۳. تغییر پیشوند نام فایل‌های خروجی (مثال: تقسیم فایل و تغییر پیشوند به “output_”):

```
split --additional-suffix=.txt -l 100 file.txt output_
```

۴. تقسیم فایل بدون ایجاد خروجی‌های با حجم خالی (مثال: تقسیم فایل بدون خروجی‌های خالی):

```
split --filter=‘cat > $FILE’ -l 50 file.txt
```

۵. تقسیم فایل با تعیین پیشوند و تعداد بایت‌ها برای هر قسمت (مثال: تقسیم فایل به بخش‌های ۵۰۰ بایتی با پیشوند “part_”):

```
split -b 500 -d --additional-suffix=.txt file.txt part_
```

۶. تقسیم فایل و نمایش پیغام‌های اطلاعاتی (مثال: تقسیم فایل و نمایش پیام‌های اطلاعاتی):

```
split -l 20 --verbose file.txt
```

۷. تقسیم فایل با تعیین حداکثر تعداد خطوط برای هر بخش (مثال: تقسیم فایل به بخش‌های حاوی حداکثر ۵۰ خط):

```
split --lines=50 file.txt
```

۸. تقسیم فایل به بخش‌های دارای حداقل تعداد بایت (مثال: تقسیم فایل به بخش‌های حداقل ۲۰۰ بایت):

```
split --min=200 file.txt
```

۹. تقسیم فایل با استفاده از پیشوند و پسوند دلخواه (مثال: تقسیم فایل با پیشوند و پسوند دلخواه):

```
split -l 50 file.txt output_part_
```

۱۰. تقسیم فایل با تعیین پیشوند و تعداد بایت‌ها برای هر قسمت (مثال: تقسیم فایل به بخش‌های ۵۰۰ بایتی با پیشوند "part_"):

```
split -b 500 -d --additional-suffix=.txt file.txt part_
```

دستور od

دستور od در سیستم عامل‌های Unix و مشتقات آن مانند Linux برای نمایش محتوای باینری یک فایل یا ورودی‌های دیگر به صورت اعشار، اکتال و یا هگزادسیمال استفاده می‌شود. این دستور به کاربر این امکان را می‌دهد که ساختار داده‌های باینری را به صورت خوانا و قابل فهم ترجمه کند.

مثال‌های دستور:

۱. نمایش محتوای باینری یک فایل به صورت اعشار:

```
od -An -t d1 file.bin
```

۲. نمایش محتوای باینری یک فایل به صورت اکتال:

```
od -An -t o1 file.bin
```

۳. نمایش محتوای باینری یک فایل به صورت هگزادسیمال:

```
od -An -t x1 file.bin
```

۴. نمایش محتوای باینری یک فایل به صورت اعشار و هگزادسیمال:

```
od -An -t d2x2 file.bin
```

۵. نمایش محتوای باینری یک فایل به همراه آدرس‌ها:

```
od -Ax -t x1z -j 16 -N 32 file.bin
```

۶. نمایش محتوای باینری یک ورودی از ورودی استاندارد به صورت اعشار:

```
echo "Hello, world!" | od -An -t d1
```

۷. نمایش محتوای باینری یک فایل به صورت اعشار و اضافه کردن آدرس فایل:

```
od -t d1 -v -j 16 -w8 file.bin
```

۸. نمایش محتوای باینری یک فایل به صورت هگزادسیمال و نمایش آدرس‌ها به صورت اعشار:

```
od -Ax -t x1z -j 16 -N 32 -v file.bin
```

۹. نمایش محتوای باینری یک فایل به صورت اکتال و نمایش آدرس‌ها:

```
od -Ax -t o1z -j 16 -v file.bin
```

۱۰. نمایش محتوای باینری یک فایل به صورت اعشار و نمایش آدرس‌ها به صورت هگزادسیمال:

```
od -Ax -t d2X2z -j 16 -v file.bin
```

دستور cut

دستور cut در سیستم عامل‌های Unix و مشتقات آن مانند Linux برای استخراج بخش‌های مشخصی از خطوط یک فایل متنی مانند متن، جدول و بر اساس محدودکننده‌هایی نظیر جداکننده delimiterو موقعیت‌های قطعه‌ها fieldsاستفاده می‌شود.

مثال‌های دستور:

۱. استخراج یک فیلد مشخص از یک خط متنی با جداکننده فضا:

```
echo “Hello World” | cut -d’ ‘ -f2
```

نتیجه: World

۲. استخراج چند فیلد مشخص از یک خط متنی با جداکننده کاما:

```
echo “apple,banana,orange” | cut -d’,’ -f1,3
```

نتیجه: apple,orange

۳. استخراج بخش مشخصی از یک فایل متنی با جداکننده تب و بر اساس موقعیت:

```
cut -f2 file.txt
```

۴. استخراج بخش مشخصی از یک فایل متنی با جداکننده دلخواه و بر اساس موقعیت:

```
cut -d’:’ -f1,3 data.txt
```

۵. استخراج بخش مشخصی از یک فایل متنی با جداکننده و بدون نمایش خطوط تکراری:

```
cut -d’,’ -f1,2 -s data.csv
```

۶. استخراج بخش مشخصی از یک فایل متنی با نمایش فیلدهای بیشتر از یک موقعیت:

```
cut -c1-5,10-15 file.txt
```

۷. استخراج بخش مشخصی از یک فایل متنی با نمایش خطوط بدون محتوای فیلد:

```
cut -d’,’ -f2 --complement data.csv
```

۸. استخراج بخش مشخصی از یک فایل متنی با نمایش فیلدها بدون استفاده از جداکننده:

```
cut -c1-5,10-15 --output-delimiter=“ “ file.txt
```

۹. استخراج بخش مشخصی از یک ورودی متنی با جداکننده فاصله و نمایش فیلدهای بدون محتوا:

```
echo “1 2 3” | cut -d’ ‘ -f2 --output-delimiter=“,”
```

۱۰. استخراج بخش مشخصی از یک ورودی متنی با جداکننده فاصله و نمایش تمامی فیلدها:

```
echo “1 2 3” | cut -d’ ‘ --output-delimiter=“,” -f-
```

دستور sed

sed مخفف stream editor است. یک ویرایشگر جریان برای انجام تبدیل های متنی اساسی در یک جریان ورودی (فایل یا ورودی) استفاده می شود. به عنوان **مثال**، می تواند بسیاری از عملکردها را روی فایل ها مانند جستجو، یافتن و جایگزینی، درج یا حذف انجام دهد. در حالی که از برخی جهات شبیه ویرایشگری است که ویرایش‌های اسکریپت‌شده را مجاز می‌کند (مانند ed)، sed تنها با یک عبور از ورودی(ها) کار می‌کند و در نتیجه کارآمدتر است. اما این توانایی sed برای فیلتر کردن متن در خط pipe است که آن را از سایر انواع ویرایشگرها متمایز می‌کند. دستور sed برای جایگزینی یا find and replace است. با استفاده از sed می‌توانید فایل‌ها را حتی بدون باز کردن آن ویرایش میکند، که راه بسیار سریع‌تری برای یافتن و جایگزین کردن چیزی در فایل است. از عبارات منظم اولیه و توسعه یافته پشتیبانی می‌کند که به شما امکان می‌دهد الگوهای پیچیده را مطابقت دهید. اکثر توزیع های لینوکس با گنو ارائه می شوند و sed به طور پیش فرض از قبل نصب شده است. دستور sed یک ابزار قدرتمند در لینوکس و سیستم عامل‌های مشابه است که برای ویرایش و تغییر محتوای خطوط فایل‌ها استفاده می‌شود. این دستور قابلیت جستجو و جایگزینی الگوها در فایل‌ها را دارد و می‌تواند برای انجام تغییرات چندین خط یا پرونده به صورت همزمان استفاده شود.

ساختار دستور:

```
$sed [گزینه‌ها] ’دستور‘ [فایل‌ها]
```

گزینه های دستور:

- -e :استفاده از چندین دستور در یک خط دستور.
- -f :خواندن دستورها از یک فایل.
- -i :ویرایش فایل‌ها در حالت مستقیم (دستورات اصلاحی را در فایل‌ها ذخیره می‌کند).
- -n :عدم چاپ خطوط به صورت خودکار.
- -r یا --regexp-extended :استفاده از عبارات منظم گسترده.

مثال های دستور:

جایگزینی الگو:

```
$sed ‘s/foo/bar/’ filename
```

جایگزینی تمام نمونه‌های الگو در یک خط:

```
$sed ‘s/foo/bar/g’ filename
```

جایگزینی الگو در یک بازه خطی:

```
$sed ‘2,4s/foo/bar/’ filename
```

حذف خطوط با الگوی مشخص:

```
$sed ‘/pattern/d’ filename
```

افزودن یک خط پس از یک الگو:

```
$sed ‘/pattern/a\New Line’ filename
```

افزودن یک خط قبل از یک الگو:

```
$sed ‘/pattern/i\New Line’ filename
```

نمایش خطوط با الگوی مشخص:

```
$sed -n ‘/pattern/p’ filename
```

دستور tee

دستور tee در سیستم عامل‌های Unix و مشتقات آن مانند Linux برای خواندن و نوشتن همزمان اطلاعات از یک ورودی (مانند خروجی یک دستور)، آن را به یک یا چند فایل متنی ارسال می‌کند و همچنین اطلاعات را به خروجی استاندارد می‌فرستد.

مثال‌های دستور:

۱. نوشتن خروجی یک دستور به فایل و نمایش آن در خروجی استاندارد:

```
ls -l | tee file.txt
```

۲. اضافه کردن خروجی یک دستور به انتهای یک فایل موجود و نمایش آن در خروجی استاندارد:

```
echo “New line” | tee -a file.txt
```

۳. نوشتن خروجی یک دستور به چند فایل متنی مختلف:

```
ps aux | tee process1.txt process2.txt
```

۴. نوشتن خروجی یک دستور به یک فایل متنی و انتقال آن به یک فایل دیگر:

```
ls -l | tee file1.txt | tee file2.txt
```

۵. استفاده از tee به همراه pipe کشی ((pipingبرای انتقال خروجی یک دستور به فایل:

```
cat file.txt | grep “pattern” | tee filtered_output.txt
```

۶. نوشتن خروجی یک دستور به فایل با دسترسی خواندنی و قابل اجرا:

```
ls -l | tee file.txt
chmod +x file.txt
```

۷. نوشتن خروجی یک دستور به فایل با دسترسی فقط خواندنی:

```
ps aux | tee file.txt
chmod -r file.txt
```

۸. نوشتن خروجی یک دستور به فایل و نمایش پیامی پس از انجام عملیات:

```
ls -l | tee file.txt && echo “Output saved to file.txt”
```

۹. نوشتن خروجی یک دستور به یک فایل و انتقال خروجی به یک دستور دیگر:

```
cat file.txt | tee -a output.txt | grep “pattern”
```

۱۰. نوشتن خروجی یک دستور به فایل و نمایش پیام خطایی در صورت خطا:

```
ls -l | tee file.txt || echo “Error: Failed to save output to file.txt”
```

دستور tac

دستور tac در سیستم عامل‌های Unix و مشتقات آن مانند Linux برای خواندن خطوط یک فایل متنی از انتها به ابتدا استفاده می‌شود. نام tac به عنوان برعکس کلمه cat که خطوط را از ابتدا به انتها نمایش می‌دهد تعبیر می‌شود.

مثال‌های دستور:

۱. نمایش خطوط یک فایل از انتها به ابتدا:

```
tac file.txt
```

۲. نمایش خروجی یک دستور از انتها به ابتدا:

```
ls -l | tac
```

۳. ادغام چند فایل و نمایش محتوای آنها از انتها به ابتدا:

```
tac file1.txt file2.txt
```

۴. نمایش خطوط یک فایل به همراه شماره‌گذاری از انتها به ابتدا:

```
tac -n file.txt
```

۵. نمایش خروجی یک دستور به همراه شماره‌گذاری از انتها به ابتدا:

```
ps aux | tac -n
```

۶. نمایش خطوط یک فایل به صورت عکس العمل (reverse) از انتها به ابتدا:

```
tac -r file.txt
```

۷. نمایش خطوط یک فایل به صورت عکس العمل و با شماره‌گذاری از انتها به ابتدا:

```
tac -r -n file.txt
```

۸. نمایش خطوط یک فایل به صورت عکس العمل و با جداکننده خطوط مشخص:

```
tac -s ‘,’ file.csv
```

۹. نمایش خروجی یک دستور به صورت عکس العمل و با جداکننده خطوط مشخص:

```
find . -type f | tac -s ‘,’
```

۱۰. نمایش خطوط یک فایل به صورت عکس العمل و فقط تا تعداد خطوط مشخص:

```
tac -n -r -m 10 file.txt
```

دستور iconv

دستور iconv در سیستم‌عامل‌های Unix/Linux برای تبدیل کدگذاری encodingمتن‌ها استفاده می‌شود. این دستور به کاربر امکان می‌دهد تا متن‌هایی که با یک کدگذاری خاص نوشته شده‌اند را به یک کدگذاری دیگر تبدیل کند.

مثال های دستور:

۱. تبدیل فایل با کدگذاری UTF-8 به ASCII و ذخیره نتیجه در فایل جدید:

```
iconv -f UTF-8 -t ASCII input.txt > output.txt
```

۲. تبدیل متن با کدگذاری ISO-8859-1 به UTF-8 و نمایش نتیجه در خروجی دستور:

```
echo " | "مثالiconv -f ISO-8859-1 -t UTF-8
```

۱. – f: مشخص کردن کدگذاری مبدا (from encoding).

```
iconv -f ISO-8859-1
```

۲. – t: مشخص کردن کدگذاری مقصد (to encoding).

```
iconv -t UTF-8
```

۳. – o: تعیین نام فایل خروجی برای ذخیره نتیجه تبدیل.

```
iconv -o output.txt
```

۴. – l: نمایش لیست کاملی از کدگذاری‌های موجود.

```
iconv -l
```

۵. – c: نادیده گرفتن کاراکترهایی که نتواند توسط کدگذاری مقصد نمایش داده شود.

```
iconv -c
```

دستور wall

دستور wall برای ارسال / نمایش پیام به تمامی کاربران system استفاده می شود. wall مخفف کلمه write all است. wall برای ارسال پیغام به همه است ، برای ارسال پیام فقط به کاربر خاص از دستور write استفاده کنید. معمولاً ، مدیران سیستم پیام هایی را برای اعلام تعمیر و نگهداری ارسال می کنند و از کاربران می خواهند همه برنامه های باز شده را از سیستم خارج و بسته کنند. پیام ها با ترمینال باز به همه کاربران وارد شده نشان داده می شوند.کاربرانی که از یک محیط دسکتاپ گرافیکی و بدون ترمینال باز استفاده می کنند پیام ها را نمی بینند.

ساختار دستور:

```
wall [OPTIONS] [<FILE>|<MESSAGE>]
```

اگر فایلی مشخص نشده باشد ، wall پیام را از ورودی استاندارد می خواند.ساده ترین راه برای ارسال پیام فراخوانی دستور wall با پیام به عنوان آرگومان است:

```
$ wall "The system will be restarted in 10 minutes."
```

خروجی به این شکل است :

```
Broadcast message from root@linuxize.host (pts/0) (Sun Oct 4 19:22:07 2020):
The system will be restarted in 10 minutes.
```

پیام برای همه کاربرانی که در حال حاضر وارد سیستم شده اند ارسال خواهد شد. برای سرکوب بنر و نمایش تنها متنی که تایپ می کنید به کاربران وارد شده ، دستور را با گزینه (–nobanner) n – فراخوانی کنید:

```
wall -n "The system will be restarted in 10 minutes."
```

خروجی به این شکل است :

```
The system will be restarted in 10 minutes.
```

دستور biff

biffیک سیستم اطلاع رسانی نامه برای یونیکس است که هنگام ورود نامه جدید کاربر را در خط دستورمطلع میکند و از آن شخص می گوید.

ساختار دستور:

```
biff [y | n]
```

```
BIFF(1)
      BSD General Commands Manual
                                BIFF(1)

NAME
     biff – be notified if mail arrives and who it is from

SYNOPSIS
     biff [ny]

DESCRIPTION
     Biff informs the system whether you want to be notified when mail arrives d
uring the current terminal session.

     Options supported by biff:

     n      Disables notification.

     y      Enables notification.

     When mail notification is enabled, the header and first few lines of the me
ssage will be printed on your screen whenever mail arrives.  A “biff y” command
is often included in the file .login
     or .profile to be executed at each login.

     Biff operates asynchronously via the comsat(8) service. If that service is
not enabled, biff will not do anything. In that case, or for synchronous notific
ation, use the MAIL variable of sh(1)
     or the mail variable of csh(1).

SEE ALSO
     csh(1), mail(1), sh(1), comsat(8)

HISTORY
     The biff command appeared in 4.0BSD.

BUGS
     su(1), and biff don't seem to get on too well. This is probably due to the
tty still being owned by the person using su. This can result in “Permission den
ied” messages when attempting to
     change the biff status of your session.

     Please report bugs to netbug@ftp.uk.linux.org including diffs/patches, comp
iler error logs or as complete a bug report as is possible.
```

گزینه های دستور:

- **Y** این گزینه برای فعال کردن اعلان نامه است.
- **N** این گزینه غیرفعال کردن اعلان نامه است.

توجه داشته باشید:

- وقتی اعلان نامه فعال باشد ، هر زمان نامه به شما رسید ، سرصفحه و چند خط اول پیام روی صفحه شما چاپ می شود .برای فعال کردن اعلان ها باید این دستور را تصویب کنید:

```
biff  y
```

- biff از طریق سرویس comat بصورت همزمان انجام می شود .اگر آن سرویس فعال نباشد ، biffهیچ کاری نمی کند.

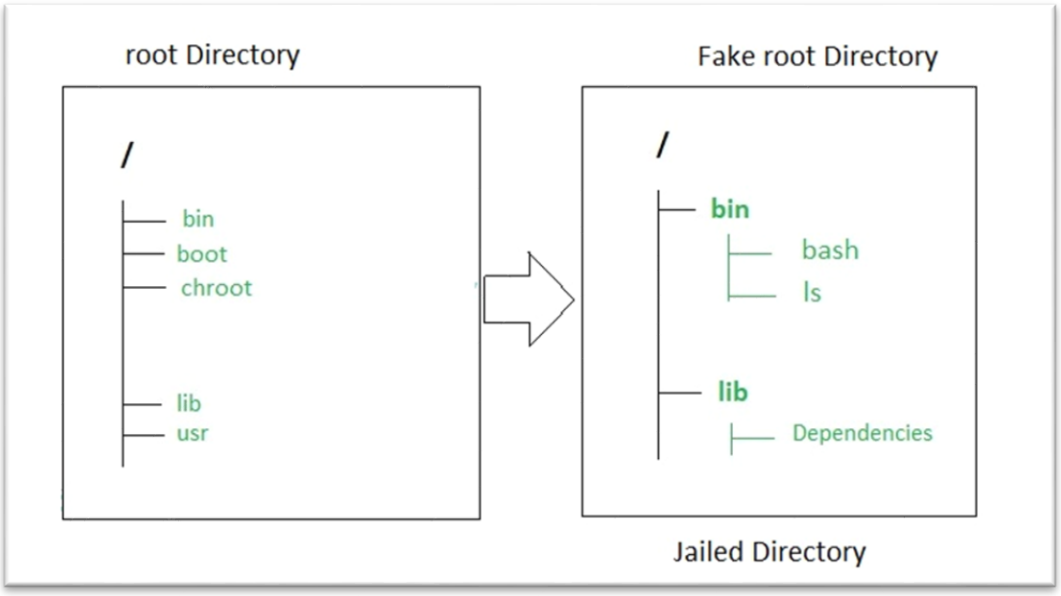
دستور agetty

agetty برنامه ای است که ترمینال های فیزیکی یا مجازی را مدیریت می کند و توسط initفراخوانی می شود .هنگامی که یک اتصال را تشخیص داد، یک پورت tty را باز می کند، نام ورود کاربر را می خواهد و دستور /bin/login را فرا می خواند agetty .جایگزین لینوکس gettyاست :

```
$agetty -L 9600 ttyS1 vt100
```

دستور chroot

از دستور **chroot** در سیستم لینوکس / یونیکس برای تغییر فهرست ریشه استفاده می شود .هر پردازش / فرمانی مانند سیستم های Linux / Unixدارای یک فهرست فعال در حال حاضر به نام **root Directory** است . این فهرست ریشه را برای فرآیندهای در حال اجرا و همچنین فرایندهای فرزند خود تغییر می دهد. یک فرآیند / دستوری که در چنین محیط اصلاح شده ای اجرا می شود ، نمی تواند به پرونده های خارج از فهرست ریشه دسترسی پیدا کند .این محیط اصلاح شده به " chroot jail " یا" **زندان" معروف است** . برخی از کاربران ریشه مجاز به استفاده از دستور chroot هستند.



دستور "chroot" می تواند بسیار مفید باشد :

- برای ایجاد یک محیط تست.
- برای بازیابی سیستم یا رمز عبور.
- برای نصب مجدد بوت لودر.

ساختار دستور:

```
chroot / path / to / new / root
```

یا

```
chroot / path / to / new / root / path / to / server
```

یا

```
chroot [options] /path/to/new/root /path/to/server
```

گزینه های دستور:

- userspec=USER:GROUP -- توصیف کاربر و گروه مورد استفاده را انجام می‌دهد. می‌توان از نام یا شناسه عددی برای تعیین کاربر و گروه استفاده کرد.
- groups=G_LIST -- توصیف گروه‌های مکمل به صورت g1، g2، ... gN را انجام می‌دهد.
- help -- نمایش پیام راهنما و خروج از دستور.
- version -- نمایش اطلاعات نسخه و خروج از دستور.

مثال :

- مرحله ۱ :ما یک mini-jail فقط با دستورات bash و اساسی ایجاد خواهیم کرد .بیایید یک پوشه "jail" در داخل فهرست "home" ایجاد کنیم ، که ریشه جدید ما خواهد بود.

```
$ mkdir $HOME/jail
```

مرحله ۲ :ایجاد فهرست در داخل : "$ HOME / jail"

```
$ mkdir -p $HOME/jail/{bin, lib64}
$ cd $HOME/jail
```

مرحله ۳ :با استفاده از دستور cp / bin / bash و bin / ls را در $ HOME / jail / bin / location کپی کنید:

```
$ cp -v / bin / {bash ،ls} $ HOME / jail / bin
```

مرحله ۴ :از دستور ldd برای چاپ کتابخانه های مشترک استفاده کنید:

```
$ ldd / bin / bash
```

```
architgarg@Archit:~$ mkdir -p $HOME/jail/{bin,lib64}
architgarg@Archit:~$ ldd /bin/bash
        linux-vdso.so.1 (0x00007ffc1e1bf000)
        libtinfo.so.6 => /lib/x86_64-linux-gnu/libtinfo.so.6 (0x00007f2b26945000)
        libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007f2b26940000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f2b2677f000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f2b26ab2000)
architgarg@Archit:~$
```

مرحله ۵ :با استفاده از دستور cp کتابخانه های مورد نیاز را در $ HOME / jail / lib64 / location کپی کنید:

```
cp -v libraries/displayed/by/above/command $HOME/jail/lib64
```

به همین ترتیب ، کتابخانه های دستور ls را در مکان $ HOME / jail / lib64 کپی کنید.

مرحله ۶ :سرانجام ، در مینی زندان خود chroot کنید:

```
$ sudo chroot $ HOME / jail / bin / bash
```

اکنون کاربر فهرست $ HOME / jail را به عنوان فهرست ریشه خود می بیند . این یک تغییر بزرگ در امنیت است.

دستور ldd

lddمخفف عبارت **list dynamic dependencies**است و یک ابزار خط فرمان در سیستم‌عامل‌های یونیکس‌مانند، به ویژه لینوکس، است که برای نمایش وابستگی‌های کتابخانه‌ای دینامیکی یک برنامه اجرایی (فایل با پسوند .so) استفاده می‌شود.

ساختار دستور:

```
ldd executable_file
```

- **executable_file:** مسیر کامل فایل اجرایی که می‌خواهید وابستگی‌های آن را بررسی کنید.

کاربردهای دستور ldd:

- **شناسایی کتابخانه‌های مورد نیاز :** با اجرای این دستور، لیستی از تمام کتابخانه‌های دینامیکی که برنامه برای اجرا به آن‌ها نیاز دارد، نمایش داده می‌شود.
- **حل مشکلات مربوط به کتابخانه‌ها :** اگر برنامه‌ای با خطای "cannot find library" یا خطای مشابه مواجه شود، با استفاده از ldd می‌توان فهمید که کدام کتابخانه‌ها نصب نشده‌اند یا در مسیر اشتباهی قرار دارند.
- **بررسی سازگاری برنامه با سیستم :** با مقایسه خروجی ldd با کتابخانه‌های نصب شده در سیستم، می‌توان بررسی کرد که آیا برنامه بر روی سیستم مورد نظر قابل اجرا است یا خیر.
- **یافتن نسخه‌های کتابخانه‌ها :** در برخی موارد، lddمی‌تواند نسخه‌های کتابخانه‌های مورد استفاده توسط برنامه را نیز نمایش دهد.

گزینه‌های مهم دستور ldd:

- **version--:** نسخه دستور ldd را نمایش می‌دهد.
- **verbose--:** اطلاعات بیشتری در مورد هر کتابخانه، مانند مسیر کامل و نسخه آن، نمایش می‌دهد.

مثال‌ها:

- **نمایش وابستگی‌های یک برنامه:**

```
ldd /bin/ls
```

این دستور لیستی از کتابخانه‌های مورد نیاز برنامه ls را نمایش می‌دهد.

- **نمایش اطلاعات دقیق‌تر:**

```
ldd --verbose /usr/bin/firefox
```

این دستور اطلاعات دقیقی در مورد تمام کتابخانه‌های مورد نیاز مرورگر فایرفاکس را نمایش می‌دهد، از جمله مسیر کامل، نسخه و نوع کتابخانه.

نکات مهم:

- **کتابخانه‌های دینامیکی :** کتابخانه‌های دینامیکی فایل‌هایی هستند که حاوی کدهای مشترک مورد استفاده توسط چندین برنامه هستند .این کتابخانه‌ها در زمان اجرا توسط برنامه بارگذاری می‌شوند.

- **وابستگی‌های استاتیکی :**برخی از برنامه‌ها ممکن است وابستگی‌های استاتیکی نیز داشته باشند که در زمان کامپایل به برنامه لینک می‌شوند .این وابستگی‌ها توسط دستور ldd نمایش داده نمی‌شوند.
- **خطاهای رایج :**اگر کتابخانه‌ای در سیستم یافت نشود، lddخطایی مبنی بر عدم وجود کتابخانه را نمایش می‌دهد .این خطا معمولاً با نصب کتابخانه مورد نظر برطرف می‌شود.

دستور ldconfig

مقدمه

در سیستم‌های لینوکس، کتابخانه‌های مشترک (shared libraries) فایل‌هایی هستند که حاوی کدهای قابل استفاده مجدد هستند و توسط چندین برنامه می‌توانند به اشتراک گذاشته شوند .دستور ldconfigابزاری است که برای مدیریت این کتابخانه‌ها و اطمینان از اینکه لینکر دینامیک (dynamic linker) می‌تواند آن‌ها را پیدا کند، استفاده می‌شود.

ساختار دستور

ldconfig [options]

- **ldconfig:**نام خود دستور است.
- **options:**گزینه‌های مختلفی که می‌توان برای کنترل نحوه عملکرد دستور استفاده کرد.

کاربردهای دستورldconfig

- **به‌روزرسانی کش کتابخانه‌ها :**مهمترین وظیفه ldconfig، به‌روزرسانی کش کتابخانه‌های سیستم است .این کش حاوی اطلاعاتی در مورد مکان کتابخانه‌های موجود در سیستم است و لینکر دینامیک از آن برای پیدا کردن کتابخانه‌های مورد نیاز برنامه‌ها استفاده می‌کند.
- **اسکن دایرکتوری‌های کتابخانه** ldconfig :می‌تواند دایرکتوری‌های مشخصی را برای یافتن کتابخانه‌های جدید اسکن کند.
- **ایجاد فایل‌های کش :**این دستور فایل‌های کش کتابخانه را ایجاد یا به‌روزرسانی می‌کند.

گزینه‌های مهم دستورldconfig

- **-v, --verbose:**اطلاعات بیشتری در مورد فرآیند به‌روزرسانی کش نمایش می‌دهد.
- **-n, --print-cache:**محتوای فایل کش فعلی را نمایش می‌دهد.
- **-C, --config file:**فایل پیکربندی جایگزینی را مشخص می‌کند.
- **-X, --add-directory directory:**یک دایرکتوری جدید را به لیست دایرکتوری‌های کتابخانه اضافه می‌کند.
- **-d, --delete-directory directory:**یک دایرکتوری را از لیست دایرکتوری‌های کتابخانه حذف می‌کند.

مثال‌های دستور

- **به‌روزرسانی کش کتابخانه‌ها:**

```
sudo ldconfig
```

این دستور معمولاً پس از نصب یا حذف یک بسته نرم‌افزاری که شامل کتابخانه‌های مشترک است، اجرا می‌شود.

- **نمایش محتوای فایل کش:**

```
sudo ldconfig -n
```

- **اضافه کردن یک دایرکتوری به لیست دایرکتوری‌های کتابخانه:**

```
sudo ldconfig -X /opt/mylibs
```

این دستور دایرکتوری /opt/mylibs را به لیست دایرکتوری‌هایی که ldconfig برای یافتن کتابخانه‌ها جستجو می‌کند، اضافه می‌کند.

چه زمانی از ldconfig استفاده می‌شود؟

- **پس از نصب یا حذف یک بسته نرم‌افزاری :**برای اطمینان از اینکه لینکر دینامیک می‌تواند کتابخانه‌های جدید یا حذف شده را پیدا کند.
- **پس از تغییر در ساختار دایرکتوری‌های کتابخانه :**برای به‌روزرسانی اطلاعات کش کتابخانه‌ها.
- **هنگامی که با خطاهایی در ارتباط با کتابخانه‌های مشترک مواجه می‌شوید :**برای بررسی و رفع این خطاها.

نکات مهم

- **اجرای با دسترسی روت :**برای اجرای ldconfig با دسترسی روت (sudo) نیاز است.
- **تغییر در فایل پیکربندی :**فایل پیکربندی ldconfig حاوی اطلاعاتی در مورد دایرکتوری‌های کتابخانه و سایر تنظیمات است .تغییر در این فایل باید با دقت انجام شود.
- **توجه به وابستگی‌ها :**هنگام نصب یا حذف کتابخانه‌ها، باید به وابستگی‌های آن‌ها توجه کنید تا از ایجاد مشکلات در عملکرد برنامه‌ها جلوگیری کنید.

دستور hexdump

از دستور**hexdump** در لینوکس برای فیلتر کردن و نمایش پرونده های مشخص شده یا ورودی استاندارد در قالب مشخص شده قابل خواندن توسط انسان استفاده می شود .به عنوان مثال، اگر می خواهید کد اجرایی یک برنامه را مشاهده کنید ، می توانید**hexdump** برای این کار استفاده کنید.

ساختار دستور:

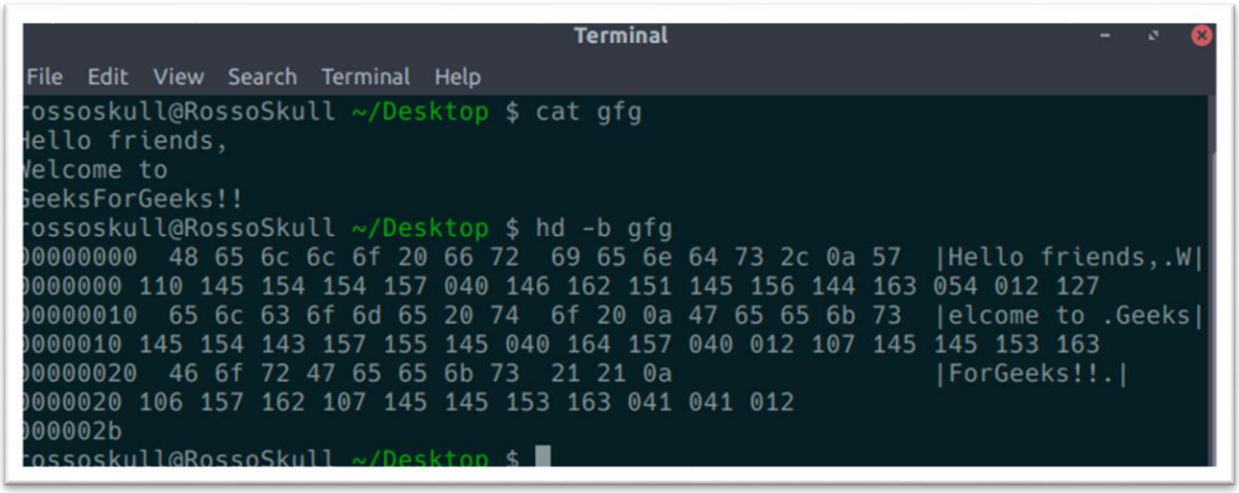
```
hd [پرونده ها] [... گزینه ها]
hd [OPTIONS...] [FILES...]
```

گزینه های دستور:

-b :نمایش یک هشت بیتی یک بایت .جابجایی ورودی را در هگزادسیمال نشان می دهد و به دنبال آن شانزده بایت داده ورودی با فضای جداگانه ، سه ستون ، صفر پر شده ، به صورت هشت ضلعی، در هر خط.

ساختار دستور:

hd -b input.txt

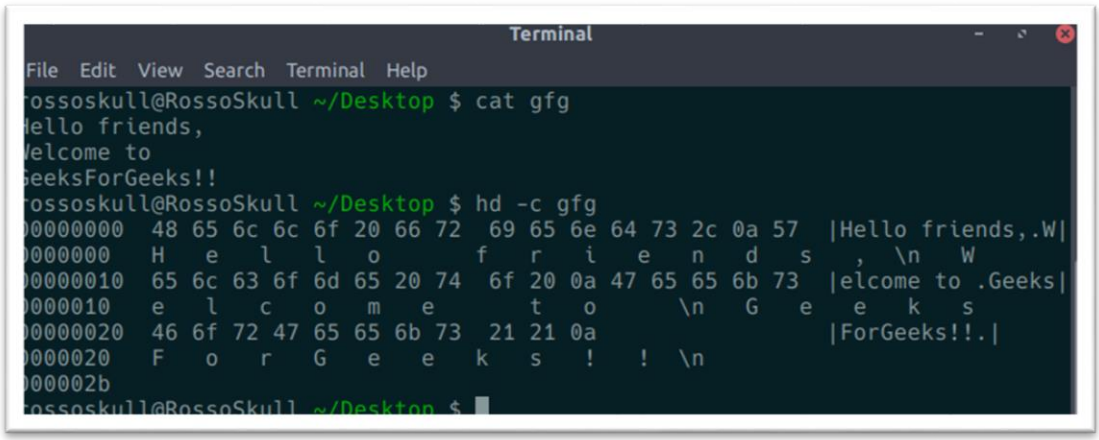


ستون اول خروجی نشان دهنده جبران ورودی در پرونده است.

-c :نمایش کاراکتر یک بایت .جابجایی ورودی را در هگزادسیمال نشان می دهد و به دنبال آن شانزده نویسه داده ورودی در هر خط ، فضای جدا شده ، سه ستون ، فضای خالی وجود دارد.

ساختار دستور:

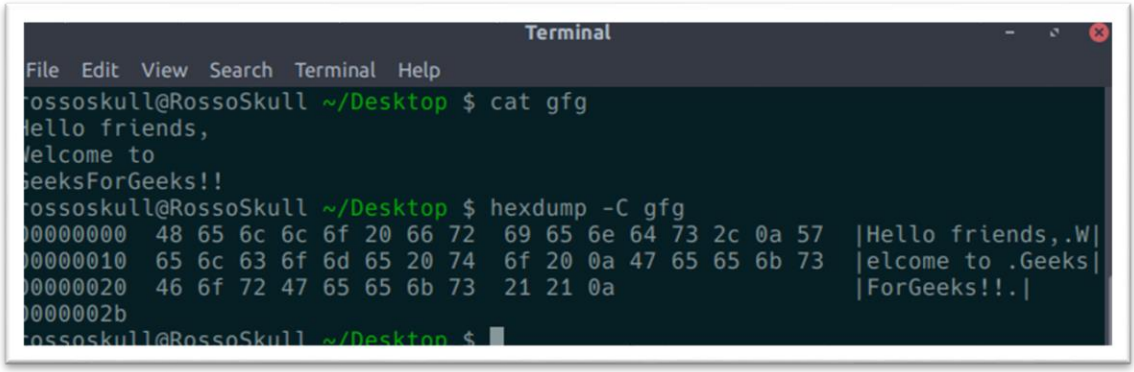
hd -c input.txt



-C :نمایشگر hex + ASCII متعارف .جابجایی ورودی را در هگزادسیمال نشان می دهد و به دنبال آن شانزده بایت هگزادسیمال با دو ستون ، دو ستون و به دنبال آن شانزده بایت با فرمت٪ p_ضمیمه شده در "|" شخصیت ها.

ساختار دستور:

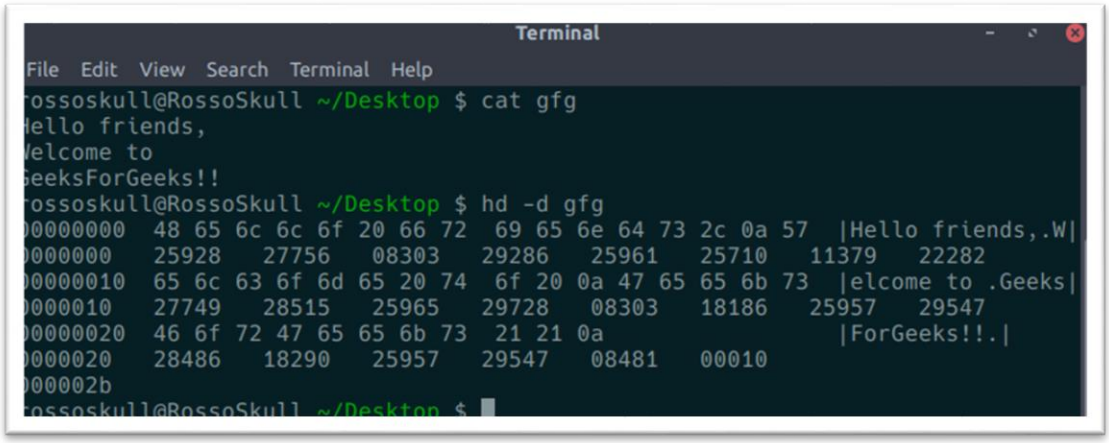
hexdump -C input.txt



-d :نمایش اعشاری دو بایت .جابجایی ورودی را در هگزادسیمال نشان می دهد ، و پس از آن هشت واحد فضای ورودی ، پنج ستون ، صفر پر شده ، دو بایت داده ورودی ، با اعشار بدون علامت ، در هر خط.

ساختار دستور:

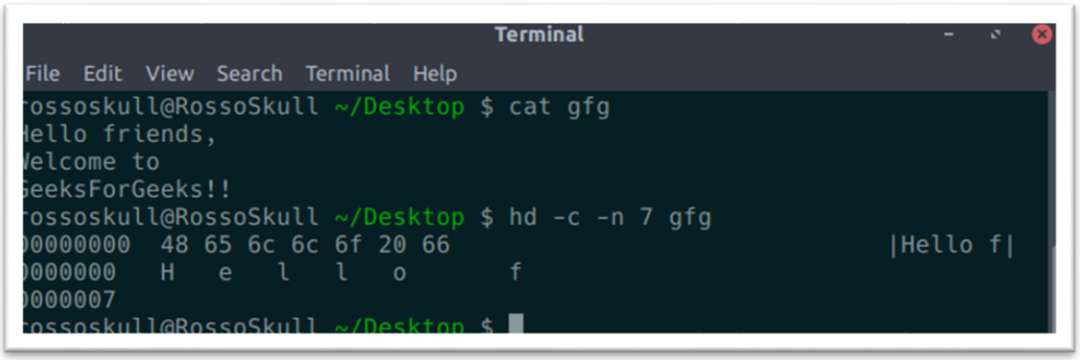
```
hd -d input.txt
```



-n length :جایی که طول یک عدد صحیح است .فقط "طول" بایت خروجی را تفسیر می کند.

ساختار دستور:

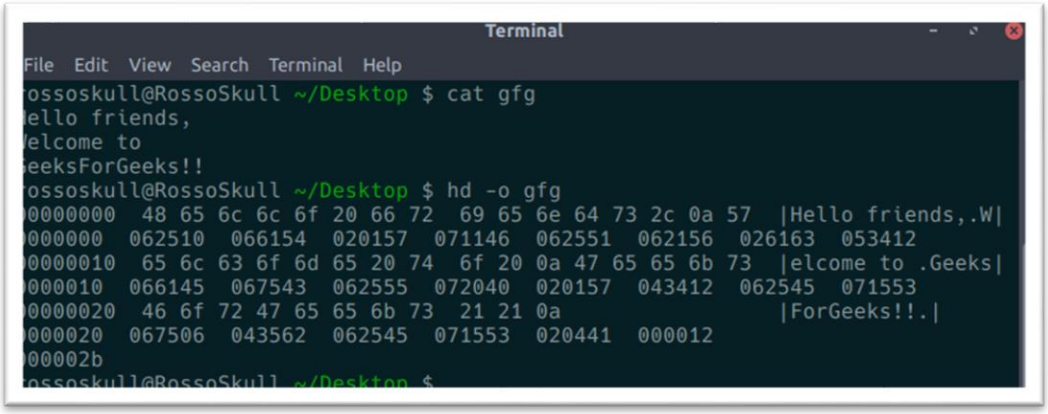
```
hd -n length input.txt
```



-o :نمایش اکتال دو بایت .جابجایی ورودی را در هگزادسیمال نشان می دهد ، و پس از آن هشت مقدار فضای ورودی ، شش ستون ، صفر پر ، دو بایت مقدار ورودی داده ، به صورت هشت ردیف ، در هر خط.

ساختار دستور:

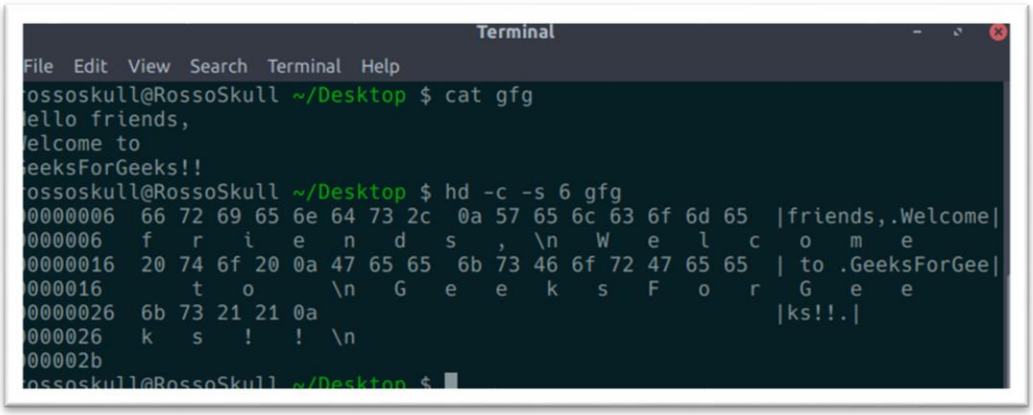
```
hd -o input.txt
```



-s offset: از ابتدای ورودی بایت "offset" را رد کنید. به طور پیش فرض ، جبران به عنوان یک عدد اعشاری تفسیر می شود. با ۰x یا ۰X پیشرو ، افست به عنوان یک عدد هگزادسیمال تفسیر می شود ، در غیر این صورت ، با ۰ پیشرو ، جبران به عنوان یک عدد هشت ضرب تفسیر می شود. افزودن نویسه های b ، k یا m برای جبران باعث می شود که به ترتیب از مضرب ۵۱۲ ، ۱۰۲۴ یا ۱۰۴۸۵۷۶ تفسیر شود.

ساختار دستور:

```
hd -s offset input.txt
```



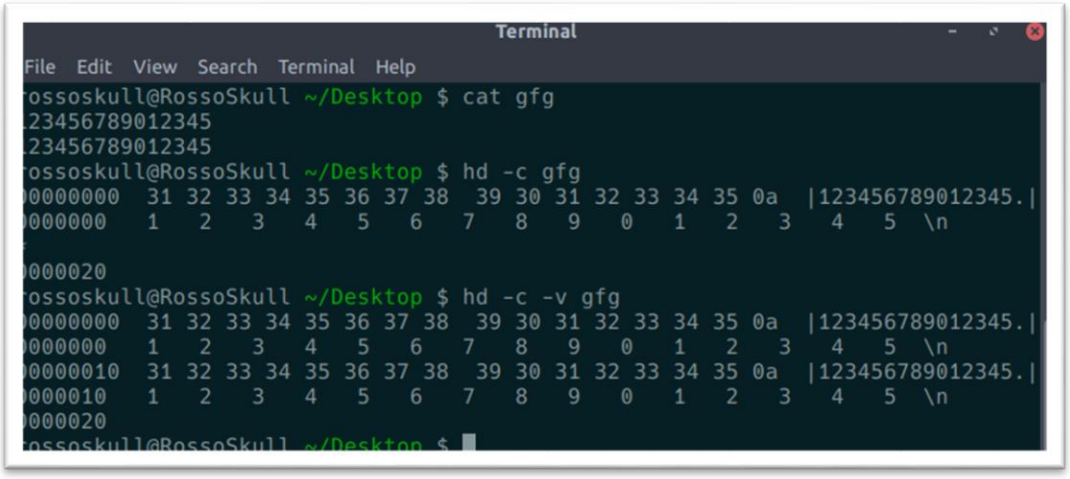
همانطور که مشاهده می کنید ، در خروجی ، ۶ کاراکتر اول ، یعنی "Hello" حذف می شوند.

-v: باعث نمایش hexdump تمام داده های ورودی می شود. بدون گزینه -v ، هر تعداد گروه از خطوط خروجی ، که یکسان با گروه بلافاصله قبل از خطوط خروجی (به استثنای جبران های ورودی) ، با یک خط متشکل از یک ستاره جایگزین می شوند.

ساختار دستور:

```
hd -v input.txt
```

ما در هنگام نمایش خروجی با استفاده از flag -c ، شاهد استفاده از این گزینه خواهیم بود.

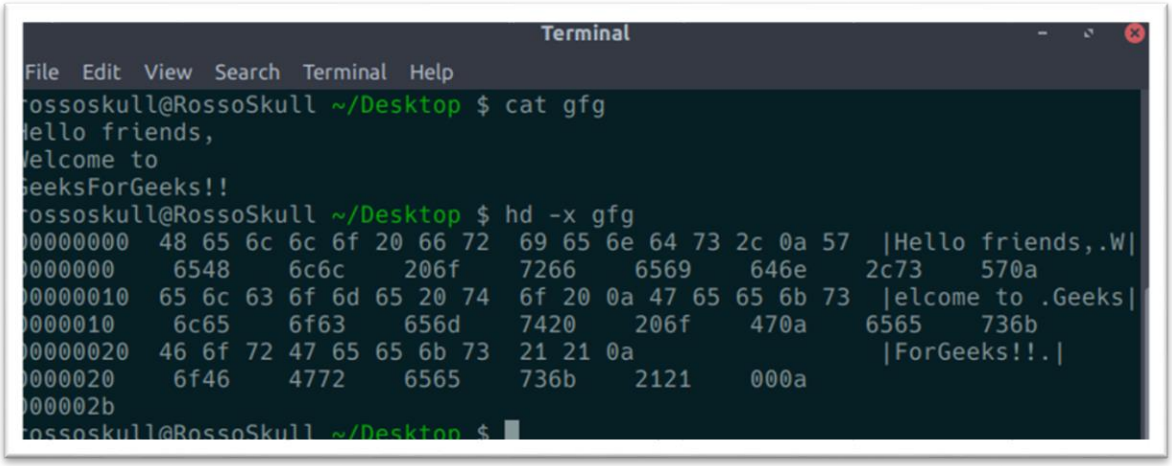


همانطور که مشاهده می کنید ، هنگامی که ما hd برای اولین بار بدون -v استفاده می کنیم ، وقتی خروجی مشابه ظاهر می شود ، یک ستاره (*) چاپ می شود. اما وقتی یک -v پرچم را رد می کنیم ، تمام خطوط خروجی را می‌گیریم.

-x: نمایش هگزادسیمال دو بایت. جابجایی ورودی را در هگزادسیمال نشان می دهد ، به دنبال آن هشت ، فضای جدا شده ، چهار ستون ، مقدار صفر پر شده ، دو بایت داده ورودی ، در هگزادسیمال ، در هر خط.

ساختار دستور:

```
hd -x input.txt
```



دستور nano

دستور nano به شما امکان ایجاد/ویرایش فایل های متنی را در پنجره می‌دهد.

نصب و راه اندازی:

ویرایشگر متن نانو روی macOS و اکثر توزیع‌های لینوکس از پیش نصب شده است. این برنامه جایگزینی برای vi و vim است. برای بررسی اینکه آیا روی نوع سیستم شما نصب شده است یا خیر دستور زیر را وارد نمایید:

```
$ nano --version
```

اگر نانو را نصب نکرده اید، می‌توانیداز دستور زیر و با استفاده از مدیریت بسته‌ها آن را نصب کنید:

نصب در اوبونتو یا دبیان:

```
$ sudo apt install nano
```

فدورا:

```
$ sudo dnf install nano
```

مثال های دستور:

۱.برای باز کردن یک فایل ، ابتدا nano را تایپ کنید و سپس مسیر فایل را تایپ کنید:

```
$ nano /path/to/filename
```

۲. برای ایجاد یک فایل جدید ، nano و سپس نام فایل را تایپ کنید:

```
$ nano filename
```

۳. برای باز کردن یک فایل را با مکان نما روی یک خط و کاراکتر خاص از دستور زیر استفاده کنید:

```
$ nano +line_number,character_number filename
```

مروری بر برخی از میانبرها و عملکردهای آنها:

توضیحات میانبرها:

- Ctrl + S ذخیره فایل فعلی
- Ctrl + O پیشنهاد نوشتن فایل ("ذخیره به عنوان") جدید
- Ctrl + X بستن بافر، خروج از نانو
- Ctrl + K خط فعلی را به cutbuffer برش میدهد

- Ctrl + U محتویات cutbuffer را paste میکند
- Alt + 6 خط فعلی را در cutbuffer کپی میکند.
- Alt + U برای Undo کردن آخرین اقدام
- Alt + E انجام مجدد آخرین اقدام لغو شده–redo

دستور recode

دستور recode در سیستم‌عامل‌های Unix/Linux برای تبدیل یک فایل از یک کدگذاری به دیگری (مثلاً از UTF-8 به ASCII) استفاده می‌شود. این دستور به کاربر این امکان را می‌دهد تا کدگذاری یک فایل را تغییر داده و آن را با فرمت مورد نظر ذخیره کند.

مثال های دستور:

تبدیل فایل از UTF-8 به ASCII: این دستور برای تبدیل یک فایل از فرمت UTF-8 به ASCII استفاده می‌شود.

```
recode UTF-8..ASCII file.txt
```

تبدیل یک فایل از ISO-8859-1 به UTF-8: این دستور برای تبدیل یک فایل از کدگذاری ISO-8859-1 به UTF-8 استفاده می‌شود.

```
recode ISO-8859-1..UTF-8 file.txt
```

تبدیل یک فهرست از UTF-16 به UTF-8: این دستور برای تبدیل تمام فایل‌های موجود در یک فهرست از کدگذاری UTF-16 به UTF-8 استفاده می‌شود.

```
recode UTF-16..UTF-8 directory/*
```

تبدیل متن ورودی از ASCII به UTF-8: این دستور برای تبدیل یک متن ورودی از کدگذاری ASCII به UTF-8 استفاده می‌شود.

```
echo “Hello, world!” | recode ASCII..UTF-8
```

تبدیل فایل از Latin-1 به UTF-16: این دستور برای تبدیل یک فایل از کدگذاری Latin-1 به UTF-16 استفاده می‌شود.

```
recode Latin-1..UTF-16 file.txt
```

دستور echo

دستور echo به شما امکان می‌دهد خط متن/رشته ای را که به عنوان آرگومان ارسال می شود را نمایش دهید. دستور echo در لینوکس و سیستم‌عامل‌های مشابه استفاده می‌شود. این دستور برای نمایش رشته‌ها (متن) در ترمینال استفاده می‌شود.

ساختار دستور:

```
$echo [رشته‌ها] [گزینه‌ها]
```

در این ساختار، شما می‌توانید از گزینه‌ها و رشته‌ها برای تنظیمات و نمایش متن موردنظر استفاده کنید. البته استفاده از گزینه‌ها و رشته‌ها اختیاری است و می‌توانید تنها با استفاده از echo بدون هیچ پارامتری، یک خط خالی را در ترمینال نمایش دهید.

گزینه های دستور:

- n – این گزینه باعث می‌شود echo عبارت را چاپ کند اما به انتهای خط جدید رفتار نکند. به عبارت دیگر، خط جدیدی که به طور پیش‌فرض پس از چاپ عبارت ایجاد می‌شود، حذف می‌شود.
- e – این گزینه به echo می‌گوید که رشته ورودی را بررسی کند و کاراکترهای خاص را تفسیر کند. برخی از کاراکترهای خاص که توسط این گزینه تفسیر می‌شوند عبارتند از (\n خط جدید)، (\t تب) و (\\ کاراکتر برگشتی).
- E – این گزینه باعث می‌شود echo کاراکترهای خاص را تفسیر نکند و به عنوان رشته‌های عادی چاپ کند.
- به علاوه، می‌توانید رشته‌های متغیر را با دستور echo نمایش دهید. برای این کار، رشته مورد نظر را با $ قبل از نام متغیر محاصره کنید.

مثال های دستور:

```
name="John"
$echo "My name is $name"
```

در این مثال، با استفاده از دستور echo، رشته " My name is John" در ترمینال نمایش داده می‌شود. دستور echo در کاربردهای مختلف مانند اسکریپت‌نویسی و تنظیمات سیستم بسیار مفید است و می‌توانید از آن برای نمایش و چاپ رشته‌ها در ترمینال استفاده کنید.

یک خط از متن را وارد کنید و در خروجی استاندارد نمایش داده شود.

```
$ echo hosein and linux
```

خروجی متن زیر است:

```
hosein and linux
```

یک متغیر را تایپ کنید و مقدار آن را echo کنید.به عنوان مثال ، اعلام یک متغیر از xو اختصاص مقدار آن 10

```
$ x = 10
```

مقدار خروجی:

```
$ echo x
```

مقدار متغیر x = 10

نکته :گزینه ' e- ' تمام فضاهای بین خطوط را حذف می کند.

```
$ echo -e " hosein \ bis \ ba \ bcommunity \ bof \ bLinux \ bNerds"
```

با استفاده از گزینه - ' n \ ' خط جدید را از جایی که از آن استفاده می شود ایجاد می کند.

```
$ echo -e " hosein \ nis \ na \ ncommunity \ nof \ n linux \ n"
```

با استفاده از گزینه ' c \ ' خط جدید ادامه می دهد.

```
$ echo -e " hosein Community /c of Linux Nerds "
```

تمامی فایل ها / فهرست را با استفاده از دستور چاپ کنید.

```
$ echo *
```

```
103.odt 103.pdf 104.odt 104.pdf 105.odt 105.pdf 106.odt 106.pdf 107.odt 107.pdf 108a.odt 108.odt
108.pdf 109.odt 109.pdf 110b.odt 110. odt 110.pdf 111.odt 111.pdf 112.odt 112.pdf 113.odt linux-
headers-3.16.0-customkernel_1_amd64.deb linux-image-3.16.0-customkernel_1_amd64.deb
network.jpeg
```

فایلهای یک نوع خاص را چاپ کنید.به عنوان مثال ، فرض کنید میخواهید تمام فایلهای jpeg.را چاپ کنید، از دستور زیر استفاده کنید.

```
$ echo * .jpeg
```

مثال های دستور:

۱.برای نمایش خط متن یا رشته ارسال شده به عنوان آرگومان:

```
$ echo Hello There
```

۲. برای نمایش همه فایل ها/پوشه ها مشابه دستور ls :

```
$ echo *
```

۳. برای ذخیره متن در فایلی به نام foo.bar:

```
$ echo “Hello There” > foo.bar
```

۴. برای افزودن متن به فایلی به نام foo.bar:

```
$ “echo Hello There” >> foo.bar
```

معمولاً در اسکریپت‌های پوسته یاهمون شل و فایل‌های دستهای ،برای خروجی متن به صفحه یا یک فایل استفاده می‌شود. –e که همراه آن استفاده می‌شود، تفسیر بک اسلش را امکان‌پذیر می‌کند.

\b	تمام فاصله های بین متن را حذف می‌کند
\ c	برای ادامه بدون ایجاد خط جدید،
\ n	خط جدیدی را از جایی که استفاده می شود ایجاد می‌کند
\ t	horizontal tabایجاد می‌کند
\ r	carriage با مفسر e-‘ backspace’ برمی گردد تا در خروجی بازده بار مشخصی داشته باشد
\ v	vertical tabایجاد می‌کند
\ a	هشدار صوتی برمی‌گردد
– n	خط جدید دنباله دار را حذف می‌کند.

نمایش یک رشته ساده:

```
$echo “Hello, World!”
```

خروجی دستور:

```
Hello, World!
```

نمایش رشته به همراه کاراکتر خاص (مثلاً خط جدید):

```
$echo -e “Line 1\nLine 2”
```

خروجی دستور:

```
Line 1
Line 2
```

نمایش متغیرها:

```
name=“John”
age=25
echo “My name is $name and I am $age years old.”
```

خروجی دستور:

```
My name is John and I am 25 years old.
```

نمایش تاریخ و زمان فعلی:

```
$echo “Today is $(date +%A), $(date +%F) and the time is $(date +%T).”
```

خروجی دستور:

```
Today is Wednesday, 2022-09-28 and the time is 14:30:00.
```

استفاده از گزینه -n برای حذف خط جدید:

```
$echo -n “This is a “
$echo “single line.”
```

خروجی دستور:

```
This is a single line.
```

مقادیر environment variables

نام متغییر	توضیح دهید
BASH_VERSION	شماره نسخه فعلی نمونه پوستهBash
EDITOR	ویرایشگر پیش فرض استفاده شده توسط برخی از دستورات پوسته
GROUPS	عضویت گروهی حساب کاربری
HISTFILE	نام فایل تاریخچه فرمان پوسته کاربر
HISTSIZE	حداکثر تعداد دستورات ذخیره شده در فایل تاریخچه
HOME	نام فهرست اصلی کاربر فعلی
HOSTNAME	نام میزبان سیستم فعلی
LANG	دسته بندی محلی برای پوسته
LC_*	تنظیمات محلی مختلف که LANG را لغو می کنند
LC_ALL	دسته محلی برای پوسته ای که LANG را لغو می کند
LD_LIBRARY_PATH	فهرست فهرست های کتابخانه جدا شده با دو نقطه برای جستجو قبل از جستجو در فهرست های استاندارد کتابخانه
PATH	فهرست فهرست های جدا شده با دو نقطه برای جستجوی دستورات
PS1	رشته درخواست رابط خط فرمان پوسته اولیه
PS2	رشته اعلان رابط خط فرمان پوسته ثانویه
PWD	فهرست کار فعلی حساب کاربر
SHLVL	سطح پوسته فعلی
TZ	منطقه زمانی کاربر، اگر با منطقه زمانی سیستم متفاوت باشد
UID	شماره شناسایی کاربر حساب کاربری

دستور printf

این دستور به شما این امکان را می‌دهد که مقدار یک متغیر را با قالب بندی با استفاده از قوانین چاپ کنید. بسیار شبیه به printf در زبان C است.

ساختار دستور:

```
$printf [-v variable_name] format [arguments]
```

رشته FORMAT را می توان به سه بخش مختلف استخراج کرد.

- **کاراکترهای معمولی** ، که دقیقاً همان کاراکترهایی که در ابتدا در خروجی استفاده شده بود کپی می شوند.
- **کاراکتر تفسیر شده**، که با یک اسلش (“\”) تعیین می کنند.
- **RGUMENTها** به عنوان بخشی از خروجی بیان می شوند.

می توانید آن قسمت ها را در این **مثال** ببینید،

```
$  printf “ %s is where over %d million developers shape \”the future of sofware.\” “ Github 65
```

خروجی دستور:

```
Github is where over 65 million developers shape “the future of sofware.”
```

رشته های تبدیل همیشه از چپ به راست تفسیر می شوند.

مثال:

ورودی

```
$  printf “We know %f is %s %d” 12.07 “larger than” 12
```

خروجی دستور:

```
We know 12.070000 is larger than 12
```

مثال بالا ۳ آرگومان، ۱۲,۰۷ ، بزرگتر از و ۱۲ را نشان می‌دهد. هر یک از آنها از چپ به راست یک به یک با ۳ رشته تبدیل داده شده (%s, %d , f) تفسیر شدند. دنباله کاراکترهایی که توسط printf به عنوان کاراکترهای خاص تفسیر می شوند :

شرح کاراکتر

\a	هشدار می‌دهد. ASCII BEL
\b	یک بک اسپیس چاپ می‌کند
\c	به دستور می‌دهد هیچ خروجی دیگری تولید نکند
\e	یک کاراکتر فرار را چاپ می‌کند (کد اسکی ۲۷)
\f	یک فید فرم را چاپ می‌کند
\n	یک خط جدید چاپ می‌کند
\r	بازگشت کالسکه را چاپ می‌کند

گزینه های دستور:

- % b :آرگومان‌ها را با گسترش دنباله‌های فرار بک اسلش چاپ می‌کند.

- %q: آرگومان‌ها را در قالب نقل قول شده پوسته چاپ می‌کند که به عنوان ورودی قابل استفاده مجدد است.
- %i, %d: آرگومان‌ها را در قالب اعداد صحیح اعشاری امضا شده چاپ می‌کند.
- %u: آرگومان‌ها را در قالب اعداد صحیح اعشاری بدون علامت چاپ می‌کند.
- %o: آرگومان‌ها را در قالب octal بدون علامت (پایه ۸) چاپ می‌کند.
- %x, %X: آرگومان‌ها را در قالب اعداد صحیح هگزادسیمال (پایه ۱۶) بدون علامت چاپ می‌کند. حروف کوچک و بزرگ را نیز چاپ می‌کند.
- %e, %E: آرگومان‌ها را در قالب اعداد ممیز شناور در نماد نمایی چاپ می‌کند. حروف کوچک و بزرگ را نیز چاپ می‌کند.
- %a, %A: آرگومان‌ها را در قالب اعداد ممیز شناور در نماد کسری هگزادسیمال (پایه ۱۶) چاپ می‌کند. حروف کوچک و بزرگ را نیز چاپ می‌کند.
- %g, %G: آرگومان‌ها را در قالب اعداد ممیز شناور در نمادهای معمولی یا نمایی چاپ می‌کند، هر کدام که برای مقدار و دقت داده شده مناسب تر باشد. حروف کوچک و بزرگ را نیز چاپ می‌کند.
- %c: آرگومان‌ها را به صورت تک کاراکتر چاپ می‌کند.
- %f: آرگومان‌ها را به صورت اعداد ممیز شناور چاپ می‌کند.
- %s: آرگومان‌ها را به صورت رشته چاپ می‌کند.
- %%: نماد "%" را چاپ می‌کند.

مثال های بیشتر:

```
$ printf 'Hello\nyoung\nman!'
```

خروجی دستور:

```
young man
```

این دو جمله را به ۳ قسمت از کلمه تقسیم می کنند.

ورودی:

```
$ printf "%f\n" 2.5 5.75
```

خروجی:

```
۲,۵۰۰۰۰۰
۵,۷۵۰۰۰۰
```

مشخص‌کننده %f همراه با \n دو آرگومان را به شکل نقطه شناور در خطوط جدید جدا شده تفسیر می‌کند.

دستور fold

دستور fold در لینوکس هر خط را در یک فایل ورودی قرار می‌دهد تا با عرض مشخصی مطابقت داشته باشد و آن را در خروجی استاندارد چاپ می‌کند. به طور پیش فرض، خطوط را با حداکثر عرض ۸۰ ستون می پیچد، اما این قابل تنظیم است. برای تا کردن ورودی با استفاده از دستور fold یک فایل یا ورودی استاندارد را به دستور ارسال میکند.

ساختاردستور:

```
$ fold [OPTION]... [FILE]...
```

گزینه های دستور:

w – : با استفاده از این گزینه در دستور fold می توانیم عرض را با تعداد ستون ها محدود کنیم. با استفاده از این دستور، عرض ستون را از عرض پیش فرض ۸۰ تغییر می دهیم.

ساختاردستور:

```
$ fold -w[n] [FILE]
```

مثال: خطوط file1.txt را به عرض ۶۰ ستون بپیچید

```
$ fold -w60 file1.txt
```

b –: این گزینه از دستور fold برای محدود کردن عرض خروجی با تعداد بایت ها به جای تعداد ستون ها استفاده می شود. با استفاده از این می توانیم عرض خروجی را به تعداد بایت ها اعمال کنیم.

```
$ fold -b[n] [FILE]
```

به عنوان **مثال:** عرض خروجی فایل را به ۴۰ بایت محدود میکند و دستور خروجی را در ۴۰ بایت کاهش می‌دهد.

```
$ fold -b40 file1.txt
```

s – : از این گزینه برای شکستن خطوط روی فاصله استفاده می شود تا کلمات شکسته نشوند. اگر قسمت هایی از خط دارای یک کاراکتر خالی در اولین موقعیت‌های ستون عرضی باشد، پس از آخرین نویسه خالی که محدودیت‌های عرض را برآورده می‌کند، خط را میشکند.

```
$ fold -w[n] -s [FILE]
```

دستور shuf

دستور shuf در سیستم‌عامل‌های Unix/Linux برای تصادفی کردن و مخلوط کردن خطوط ورودی استفاده می‌شود.

گزینه‌های دستور:

e, --echo هر ARG را به عنوان یک خط ورودی در نظر بگیرید.

```
shuf -e “apple” “banana” “orange”
```

خروجی دستور: میکس شده‌ی خطوط “apple”, “banana”, “orange”.“

i, --input-range=LO-HI هر عدد از LO تا HI را به عنوان یک خط ورودی در نظر بگیرید.

```
shuf -i 1-10
```

خروجی دستور: میکس شده‌ی اعداد از ۱ تا ۱۰.

n, --head-count=COUNT حداکثر تعداد COUNT خط را نمایش دهید.

```
shuf -n 5 file.txt
```

خروجی دستور: نمایش تصادفی ۵ خط از فایل file.txt.

o, --output=FILE نتیجه را به جای خروجی استاندارد در فایل FILE بنویسید.

```
shuf -o output.txt file.txt
```

خروجی دستور: نتیجه تصادفی را در فایل output.txt ذخیره کنید.

random-source=FILE بایت‌های تصادفی را از فایل FILE بگیرید.

```
shuf --random-source=random.bin file.txt
```

خروجی دستور: استفاده از بایت‌های تصادفی از فایل random.bin برای میکس کردن خطوط فایل file.txt.

r, --repeatخطوط خروجی ممکن است تکرار شوند.

```
shuf -r file.txt
```

خروجی دستور: خطوط تصادفی از فایل file.txt که ممکن است تکرار شوند.

z, --zero-terminatedجداکننده خط (ASCII 0) NUL است، نه خط جدید.

```
shuf -z file.txt
```

خروجی دستور: میکس شده‌ی خطوط فایل file.txt با استفاده از NUL به جای خط جدید.

helpنمایش راهنما و خروجی.

```
shuf --help
```

خروجی دستور: نمایش راهنما و کمک‌های مربوط به دستور shuf.

Versionنمایش اطلاعات نسخه و خروجی.

```
shuf --version
```

خروجی دستور: نمایش اطلاعات نسخه shuf.

فصل نهم

فصل نهم

مدیریت سیستم فایل،دیسک و پارتیشن ها

دستور smartctl

دستور smartctl در سیستم عامل‌های Unix/Linux برای دسترسی و نمایش اطلاعات SMART (Self-Monitoring, Analysis, and Reporting Technology) دیسک‌های سخت (hard driveو SSD) استفاده می‌شود. این دستور به شما امکان می‌دهد تا وضعیت و عملکرد دیسک‌های خود را بررسی کنید و از پیش‌اخطارهای ممکن درباره خرابی یا نقص‌های احتمالی مطلع شوید.

مثال های دستور:

نمایش اطلاعات SMART دیسک: این دستور برای نمایش جزئیات و اطلاعات SMART دیسک مورد استفاده قرار می‌گیرد.

```
smartctl -a /dev/sda
```

تست کامل دیسک: این دستور برای انجام یک تست کامل بر روی دیسک و بررسی وضعیت آن استفاده می‌شود.

```
smartctl -t long /dev/sdb
```

نمایش وضعیت خطاهای دیسک: این دستور برای نمایش وضعیت خطاهای دیسک و تعداد بلوک‌های پیش‌اخطاری استفاده می‌شود.

```
smartctl -l error /dev/sdc
```

تنظیم و ذخیره‌سازی تنظیمات SMART: این دستور برای تنظیم و ذخیره‌سازی تنظیمات SMART بر روی دیسک استفاده می‌شود.

```
smartctl -s on -S /dev/sdd
```

تست سریع دیسک: این دستور برای انجام یک تست سریع بر روی دیسک و بررسی وضعیت آن استفاده می‌شود.

```
smartctl -t short /dev/sde
```

دستور badblocks

دستور badblocks در سیستم‌عامل‌های Unix/Linux برای تشخیص و شناسایی بلاک‌های خراب یا بد در یک دیسک سخت (هارد دیسک) استفاده می‌شود. این دستور به کاربر این امکان را می‌دهد تا بلاک‌های خراب یا بد را بررسی کرده و اطلاعات مربوط به آن‌ها را دریافت کند.

مثال های دستور:

بررسی بلاک‌های خراب در یک دیسک سخت: این دستور برای بررسی بلاک‌های خراب در یک دیسک سخت با استفاده از حالت خواندن نوشتن تصادفی (read-write mode) استفاده می‌شود.

```
badblocks -w /dev/sda
```

بررسی بلاک‌های خراب با استفاده از حالت خواندن نوشتن تصادفی: این دستور برای بررسی بلاک‌های خراب در یک دیسک سخت با استفاده از حالت خواندن نوشتن تصادفی و نمایش اطلاعات جامع در خروجی استفاده می‌شود.

```
badblocks -vw /dev/sdb
```

بررسی و نمایش بلاک‌های خراب در حالت خواندن: این دستور برای بررسی بلاک‌های خراب در یک دیسک سخت با استفاده از حالت خواندن و نمایش اطلاعات جامع در خروجی استفاده می‌شود.

```
badblocks -n /dev/sdc
```

بررسی بلاک‌های خراب و ذخیره نتایج در فایل: این دستور برای بررسی بلاک‌های خراب در یک دیسک سخت و ذخیره نتایج در یک فایل خروجی استفاده می‌شود.

```
badblocks -o badblocks_output.txt /dev/sdd
```

بررسی و تعمیر بلاک‌های خراب در حالت خواندن و نوشتن: این دستور برای بررسی و تعمیر بلاک‌های خراب در یک دیسک سخت با استفاده از حالت خواندن و نوشتن استفاده می‌شود.

```
badblocks -c 10240 -w -s /dev/sde
```

دستور dd

دستور dd برای کپی فایل ها، تبدیل و قالب بندی با توجه به سویچ های ارائه شده در خط دستور استفاده می شود.مثال زیر نشان می دهد ایجاد یک دستگاه USB با قابلیت بوت :

```
$ dd if = /home/hosein/kali-linux-3-amd64.iso of =/dev/sdc1 bs = 512M
```

دستور df

در اینترنت مقدار زیادی ابزار برای بررسی استفاده از فضای دیسک در linux پیدا خواهید کرد.با این حال، linux یک ابزار قدرتمند ساخته شده را به نام ' df ' ساخته است.دستور " df " برای " سیستم فایل دیسک " است، برای خلاص شدن کامل از در دسترس بودن و استفاده از فضای استفاده از دیسک system در linux system استفاده می شود .با استفاده از پارامتر « -h » با (df -h)(آمار فضای دیسک فایلی را در فرمت » قابل خواندن برای انسان[1] نشان می دهد، به این معنی که جزئیات در بایت، مگا بایت و گیگابایت را می دهد. دستور " df " اطلاعات نام دستگاه، بلوک های کل، فضای دیسک کل، فضای دیسک مورد استفاده، فضای دیسک در دسترس و نقاط سوپاپ در یک سیستم فایل را نمایش می دهد.

```
$ sudo df
```

هنگام استفاده از لینوکس ممکن است شرایطی پیش بیاید که بخواهید از میزان فضای مصرف شده توسط یک سیستم فایل خاص در سیستم LINUX خود یا میزان فضای موجود در یک فایل سیستم خاص مطلع شوید . اگر هیچ نام پرونده ای به عنوان آرگومان با دستور df منتقل نشود ، آنگاه فضای موجود در تمام سیستم های پرونده نصب شده در حال حاضر را نشان می دهد. این چیزی است که شما ممکن است بخواهید بدانید زیرا دستور **df** قادر به نشان دادن فضای موجود در سیستم فایلهای پیاده نشده است و دلیل این امر این است که برای انجام این کار در برخی از سیستم ها نیاز به دانش بسیار عمیقی در مورد ساختار سیستم فایل پرونده دارد.
به طور پیش فرض ، **df** فضای دیسک را در **1** بلوک **K** نشان می دهد.

[1] human read

ساختار دستور:

```
df [OPTION] ... [FILE] ...
```

فرض کنید فایلی به نام kt.txt دارید و می خواهید از فضای دیسک استفاده شده در سیستم فایل که شامل این فایل است ، بدانید ، بنابراین می توانید از **df** در این حالت استفاده کنید:

```
$ df kt.txt
```

حال ، اگر هیچ نام پرونده ای به t df ندهید ، در این صورت df اطلاعات استفاده از دیسک را برای تمام سیستم های پرونده نصب شده به صورت زیر نشان می دهد:

```
$ df
```

نمایش اطلاعات تمام سیستم فایل استفاده از فضای دیسک

اطلاعات system های فایل ساختگی همراه فضای استفاده از دیسک و استفاده از memory را نمایش می دهد.

```
$ sudo df -a
```

نمایش استفاده از فضای دیسک در فرمت قابل خواندن انسانی

متوجه شده اید که دستورات بالا اطلاعات را در بایت نمایش می‌دهد، که هنوز قابل خواندن نیست، چرا که ما عادت به خواندن اندازه ها در مگابایت، گیگابایت و غیره هستیم زیرا بسیار آسان است که بتوانیم آن را درک کنیم و به یاد داشته باشیم .دستور DFگزینه ای برای نمایش اندازه در فرمت های خوانده شده انسان فراهم می کند با استفاده از ’h-‘ چاپ نتایج در فرمت قابل خواندن انسانی به عنوان مثال :

```
$ sudo df -h
```

```
/ dev / cciss / c0d0p2 75G 23G 49G 32 ٪/
/ dev / cciss / c0d0p5 24G 22G 1.2G 95 ٪/ home
/ dev / cciss / c0d0p3 29G 25G 2.6G 91 ٪/ data
/ dev / cciss / c0d0p1 289M 22M 253M 8 ٪/ boot
```

نمایش اطلاعات / سیستم فایل home

برای مشاهده اطلاعات تنها سیستم فایل /home در فرمت قابل خواندن برای انسانی، از دستور زیر استفاده کنید.

```
$ sudo df -hT / home
```

```
/dev/cciss/c0d0p5 ext3 24G 22G 1.2G 95 ٪home
```

نمایش اطلاعات سیستم فایل درحالت Bytes

برای نمایش تمام اطلاعات سیستم فایل و استفاده در بلوک های 1024بایت ، از گزینه k- به صورت زیر استفاده کنید.

```
$ sudo df -k
```

خروجی دستور:

```
/ dev / cciss / c0d0p2 78361192 23187212 51129216 32 ٪/
/ dev / cciss / c0d0p5 24797380 22273432 1243972 95 ٪/ home
/ dev / cciss / c0d0p3 29753588 25503792 2713984 91 ٪/ data
```

```
/ dev / cciss / c0d0p1 295561 21531 258770 8 ٪/ boot
tmpfs 257476 0 257476 0 ٪/ dev / shm
```

نمایش اطلاعات سیستم فایل درMB

برای نمایش اطلاعات تمام استفاده از سیستم فایل در MB مگا بایت از گزینه به عنوان " m- " استفاده کنید.

```
$ sudo df -m
```

خروجی دستور:

```
/ dev / cciss / c0d0p2 76525 22644 49931 32 ٪/
/ dev / cciss / c0d0p5 24217 21752 1215 95 ٪/ home
/ dev / cciss / c0d0p3 29057 24907 2651 91 ٪/ data
/ dev / cciss / c0d0p1 289 22 253 8 ٪/ boot
tmpfs 252 0 252 0 ٪/ dev / shm
```

نمایش اطلاعات سیستم فایل درGB

برای نمایش اطلاعات تمام آمار سیستم فایل در GBگیگابایت از گزینه به عنوان ' df -h ' استفاده کنید.

```
$ sudo df -h
```

```
/ dev / cciss / c0d0p2 75G 23G 49G 32 ٪/
/ dev / cciss / c0d0p5 24G 22G 1.2G 95 ٪/ home
/ dev / cciss / c0d0p3 29G 25G 2.6G 91 ٪/ data
/ dev / cciss / c0d0p1 289M 22M 253M 8 ٪/ boot
tmpfs 252M 0 252M 0 ٪/ dev / shm
```

نمایش سیستم فایل Inodes

با استفاده از سوئیچ " i- " اطلاعاتی از تعداد inode ها و درصد آنها برای سیستم فایل نمایش داده خواهد شد.

```
$ sudo df -i
```

نمایش سیستم فایل نوع

اگرتمام خروجی های دستورات بالا را مشاهده می کنید، خواهید دید که هیچ نوع سیستم فایل وجود ندارد که در نتایج ذکر شده باشد.برای بررسی نوع سیستم فایل خود از گزینه ' T ' استفاده کنید.این نوع سیستم فایل را همراه با سایر اطلاعات نمایش می دهد.

```
$ sudo df -T
```

خروجی دستور:

```
/ dev / cciss / c0d0p2 ext3 78361192 23188812 51127616 32 ٪/
/ dev / cciss / c0d0p5 ext3 24797380 22273432 1243972 95 ٪home
/ dev / cciss / c0d0p3 ext3 29753588 25503792 2713984 91 ٪/ data
/ dev / cciss / c0d0p1 ext3 295561 21531 258770 8 ٪/ boot
```

تعیین نوع خاص سیستم فایل

اگر می خواهید نوع خاصی از سیستم فایل را نمایش دهید، از گزینه ' t- ' استفاده کنید.به عنوان مثال ، دستور زیر فقط سیستم فایل ext3را نمایش می دهد.

```
$ sudo df -t ext3
```

خروجی دستور df:

```
/ dev / cciss / c0d0p2 78361192 23190072 51126356 32 %/
 / dev / cciss / c0d0p5 24797380 22273432 1243972 95 %/ home
 / dev / cciss / c0d0p3 29753588 25503792 2713984 91 %/ data
 / dev / cciss / c0d0p1 295561 21531 258770 8 %/ boot
```

مثالهای بیشتر:

a– :در صورت نیاز به نمایش تمام سیستم های پرونده همراه با آنهایی که اندازه بلوک آنها صفر است ، از گزینه **a-** با df استفاده کنید.

```
$ df -a
```

h– :این امر برای نمایش دستور df در خروجی با فرمت قابل خواندن توسط انسان استفاده می شود.

```
$ df -h kt.txt
```

k– :این اطلاعات و استفاده از سیستم فایل را در بلوک های ۱ K نمایش می دهد.

```
$ df -k kt.txt
```

استفاده از گزینه **total- -** برای تولید ستون های اندازه ، استفاده شده و در دسترس در خروجی استفاده می شود.

```
$ df --total
```

T– :با کمک این گزینه می توانید نوع مربوط به سیستم فایل را همانطور که نشان داده شده مشاهده کنید.

```
$ df -T kt.txt
```

t– :این مورد هنگامی مورد استفاده قرار می گیرد که شما بخواهید اطلاعات مربوط به دیسک سیستم های پرونده فقط دارای نوع خاصی باشد.

```
$ df -t squashfs
```

x– :اکنون ، شما همچنین می توانید به df بگویید که اطلاعات استفاده از دیسک را در تمام سیستم های پرونده نمایش دهد ، به جز موارد خاص با کمک گزینه. **x-**

```
$ df -x squashfs
```

i– Using .8 :از این گزینه برای نمایش اطلاعات **inode** در خروجی استفاده می شود.

```
$ df -i kt.txt
```

sync- - :به طور پیش فرض ، دستور df با گزینه **no -sync- -** خروجی تولید می کند که قبل از بازنشر اطلاعات استفاده ، تماس سیستم همگام سازی را انجام نمی دهد .اکنون می توانیم از گزینه sinc- - استفاده کنیم که همگام سازی را به نتیجه می رساند و خروجی کاملاً به روز است.

```
$ df - -sync
```

l– :وقتی دستور df را اجرا می کنیم ، به طور پیش فرض این سیستم پرونده های نصب شده در خارج از کشور را نشان می دهد که شامل آنهایی هستند که از سرورهای NFS یا Samba خارجی استفاده می کنند .ما می توانیم اطلاعات این سیستم های پرونده خارجی را از خروجی پنهان کنیم با نحو گزینه l- که در زیر نشان داده شده است.

```
$ df -l
```

دستور pydf

یک ابزار خط فرمان پیشرفته و جایگزین خوبی برای df لینوکس است. برای نمایش مقدار فضای دیسک استفاده شده و موجود در سیستم فایل لینوکس، مانند دستور df، اما در رنگ های مختلف استفاده می شود. خروجی دستور pydf با توجه به نیاز شما قابل تنظیم است.

نصب ابزار pydf

روش نصب این ابزار در توزیع های مختلف به صورت زیر است:

On RHEL/CentOS/Fedora/Scientific Linux

```
# yum install pydf
```

On Debian/Ubuntu/Linux Mint

```
$ sudo apt-get install pydf
```

نحوه استفاده از دستور pydf

اگر فقط دستور pydfرا بدون تعیین آرگومان اجرا کنید، فقط اطلاعات استفاده از فضای دیسک را به همراه تمام سیستم های فایل نصب شده نمایش می دهد. نصب pydf در لینوکس می‌تواند به روش‌های مختلفی انجام شود. در زیر دو روش نصب pydf را برای شما توضیح می‌دهم:

۱. استفاده از مدیر بسته‌ها

بیشتر توزیع‌های لینوکس، از جمله Ubuntu، Debian و Fedora، دارای مدیر بسته هستند که امکان نصب و مدیریت بسته‌های نرم‌افزاری را فراهم می‌کنند. در این حالت، می‌توانید از مدیر بسته مورد نظر خود برای نصب pydf استفاده کنید. به طور مثال، اگر از Ubuntu استفاده می‌کنید، می‌توانید از دستور زیر برای نصب pydf استفاده کنید:

```
sudo apt install pydf
sudo dnf install pydf
```

۲. استفاده از pip

pip یک ابزار مدیریت بسته برای پایتون است که به شما امکان نصب بسته‌های پایتون از منابع خارجی را می‌دهد. اگر pip را نصب کرده‌اید، می‌توانید از دستور زیر برای نصب pydf استفاده کنید:

```
pip install pydf
```

با اجرای این دستور، pydf و تمامی وابستگی‌های آن نصب خواهند شد. بعد از نصب، می‌توانید pydf را به صورت مستقیم اجرا کنید تا از آن استفاده کنید. به طور معمول، دستور زیر را برای اجرای pydf در ترمینال وارد می‌کنید: با اجرای این دستور، اطلاعات دیسک‌ها و پارتیشن‌های شما به صورت یک جدول فرمت‌بندی شده نمایش داده می‌شود.

```
# pydf
```

```
Filesystem Size  Used Avail Use%                                Mounted on
/dev/sda2   49G 3244M   44G  6.4 [.....................................] /
/dev/sda1  194M   43M  140M 22.4 [.....................................] /boot
/dev/sdb1  492G  345G  122G 70.2 [#.....................] /data
/dev/sda3   39G 4043M   33G 10.0 [#.....................................] /home
/dev/sda6   55G   33G   19G 60.0 [.....................] /var
/dev/sdc1  492G  262G  205G 53.3 [.....................] /videos
```

```
# pydf -a
```

```
Filesystem  Size  Used Avail Use% Mounted on
/dev/sda2    49G 3244M   44G  6.4 [##.....] /
/dev/sda1   194M   43M  140M 22.4 [##.....] /boot
/dev/sdb1   492G  345G  122G 70.2 [#.....] /data
devpts        0    0    0    - [.....] /dev/pts
tmpfs      3995M    0 3995M  0.0 [.....] /dev/shm
/dev/sda3    39G 4043M   33G 10.0 [#.....] /home
proc          0    0    0    - [.....] /proc
none          0    0    0    - [.....] /proc/sys/fs/binfmt_misc
sysfs         0    0    0    - [.....] /sys
/dev/sda6    55G   33G   19G 60.0 [#.....] /var
sunrpc        0    0    0    - [.....] /var/lib/nfs/rpc_pipefs
/dev/sdc1   492G  262G  205G 53.3 [#.....] /videos
```

برای چاپ اندازه ها در کیلوبایت (به عنوان مثال ، اندازه بلوک ۱۰۲۴) از گزینه kمطابق شکل زیر استفاده کنید.

```
# pydf -k
```

برای چاپ خلاصه خروجی در مگابایت (به عنوان مثال ، اندازه بلوک ۱۰۴۸۵۷۶) از گزینه "-m" مطابق شکل زیر استفاده کنید.

```
pydf -m
```

برای چاپ اندازه ها در گیگابایت (به عنوان مثال ، ۱۰۷۳۷۴۱۸۲۴)، گزینه gرا مطابق شکل زیر مشخص کنید.

```
# pydf -g
```

دستور du

" du " (Disk Usage)یک دستورالعمل استاندارد linux است که برای بررسی اطلاعات استفاده دیسک از فایل ها و فهرست ها در یک ماشین استفاده می شود.دستور duدارای گزینه های پارامترهای بسیاری است که می تواند برای نتایج در بسیاری از فرمت ها استفاده شود.دستور duهمچنین فایل ها و اندازه های فهرست را به صورت بازگشتی نمایش می دهد.

```
[ root@ hosein ] # du / home /  hosein
```

```
40 / home /  hosein  / downloads
4 /home/ hosein /.mozilla/plugins
4 /home/ hosein /.mozilla/extensions
12 /home/ hosein /.mozilla
12 /home/ hosein /.ssh
689112 /home/ hosein /Ubuntu-12.10
689360 / home /  hosein
```

خروجی دستور بالا تعداد بلوک های دیسک را در فهرست /home/hosein همراه با زیر شاخه های آن نمایش می دهد. با استفاده از گزینه " h- " با دستور " du " در "فرم قابل خواندن برای انسان" نتایج حاصل میشود.یعنی می توانید اندازه های Bytes، Kilobytes، Megabytes، Gigabytesو غیره را ببینید

```
[ root@ hosein ] # du -h / home /  hosein
```

```
40K / home /  hosein  / downloads
4.0K /home/ hosein /.mozilla/plugins
4.0K /home/ hosein /.mozilla/extensions
12K /home/ hosein /.mozilla
```

برای دریافت خلاصه ای از حجم کل حجم دیسک فهرست از گزینه " s- " به صورت زیر استفاده کنید.

```
[ root@ hosein ] # du -sh / home / hosein
```

```
674M / home / hosein
```

استفاده از سویچ " a- " با دستور " du " استفاده از دیسک از تمام فایل ها و فهرست ها را نمایش می دهد.

```
[ root@ hosein ] # du-a / home / hosein
```

```
4 /home/ hosein /.bash_logout
12 /home/ hosein /downloads/uploadprogress-1.0.3.1.tgz
24 /home/ hosein /downloads/Phpfiles-org.tar.bz2
40 / home / hosein / downloads
12 /home/ hosein /uploadprogress-1.0.3.1.tgz
4 /home/ hosein /.mozilla/plugins
4 /home/ hosein /.mozilla/extensions
12 /home/ hosein /.mozilla
4 /home/ hosein /.bashrc
689108 /home/ hosein /Ubuntu-12.10/ubuntu-12.10-server-i386.iso
689112 /home/ hosein /Ubuntu-12.10
689360 / home / hosein
```

با استفاده از " a- " سویچ همراه با " h- " استفاده از دیسک از تمام فایل ها و فهرست ها را در فرمت قابل نمایش برای انسان[1] نمایش می دهد.خروجی زیر آسان تر است زیرا فایلی را در کیلوبایت ، مگابایت و غیره نشان می دهد.

```
[ root@ hosein ] # du -ah / home / hosein
```

```
4.0K /home/ hosein /.bash_logout
12K /home/ hosein /downloads/uploadprogress-1.0.3.1.tgz
24K /home/ hosein /downloads/Phpfiles-org.tar.bz2
40K / home / hosein / downloads
12K /home/ hosein /uploadprogress-1.0.3.1.tgz
4.0K /home/ hosein /.mozilla/plugins
4.0K /home/ hosein /.mozilla/extensions
12K /home/ hosein /.mozilla
4.0K /home/ hosein /.bashrc
673M /home/ hosein /Ubuntu-12.10/ubuntu-12.10-server-i386.iso
673M /home/ hosein /Ubuntu-12.10
674M / home / hosein
```

find استفاده از درایو یک درخت فهرست با زیرمجموعه آن در بلوک های کیلوبایت.از " k- " استفاده کنید (نمایش اندازه در واحد 1024بایت)

```
[ root@ hosein ] # du-k / home / hosein
40 / home / hosein / downloads
4 /home/ hosein /.mozilla/plugins
4 /home/ hosein /.mozilla/extensions
12 /home/ hosein /.mozilla
12 /home/ hosein /.ssh
689112 /home/ hosein /Ubuntu-12.10
689360 / home / hosein
```

[1] human read

برای به دست آوردن خلاصه ای از فضای استفاده از سیستم فایل ها همراه با زیر شاخه های آن در حالت مگابایت تنها از گزینه " -mh "به صورت زیر استفاده کنید.سویچ " -m " بلوک های موجود در واحد MBرا شمارش می کند و " -h " برای فرمت قابل خواندن انسان است.

```
[ root@ hosein ] # du-mh / home / hosein
```

```
40K / home / hosein / downloads
4.0K /home/ hosein /.mozilla/plugins
4.0K /home/ hosein /.mozilla/extensions
12K /home/ hosein /.mozilla
12K /home/ hosein /.ssh
673M /home/ hosein /Ubuntu-12.10
674M / home / hosein
```

سویچ " -c " یک فضای استفاده کل دیسک را محاسبه و در آخرین خط را نشان می دهد.

```
[ root@ hosein ] # du -ch / home / hosein
```

```
40K / home / hosein / downloads
4.0K /home/ hosein /.mozilla/plugins
4.0K /home/ hosein /.mozilla/extensions
12K /home/ hosein /.mozilla
12K /home/ hosein /.ssh
673M /home/ hosein /Ubuntu-12.10
674M / home / hosein
مجموع ۶۷۴ مگابایت
```

دستور زیر مقدار استفاده تمام فایل ها و فهرست ها را نمایش می دهد، اما فایل هایی را که با الگوی داده شده مطابقت دارد، حذف می کند.دستور زیر محتویات فایل های " .txt " را محاسبه نمی کند.

```
[ root@ hosein ] # du -ah -exclude = "* .txt"
```

```
/ home / hosein 4.0K /home/ hosein /.bash_logout 12K /home/ hosein /downloads/uploadprogress-
1.0.3.1.tgz 24K / home / hosein /downloads/Phpfiles-org.tar.bz2 40K / home / hosein / downloads 12K
/home/ hosein /uploadprogress-1.0.3.1.tgz 4.0K /home/ hosein /.bash_history 4.0K /home/ hosein
/.bash_profile 4.0K
```

استفاده از فضای دیسک را بر اساس زمان نمایش دهید، از سویچ " -time " استفاده کنید، همانطور که در زیر نشان داده شده است.

```
[ root@ hosein ] # du -ha -time / home / hosein
```

```
4.0K 2012-10-12 22:32 /home/ hosein /.bash_logout
12K 2013-01-19 18:48 /home/ hosein /downloads/uploadprogress-1.0.3.1.tgz
24K 2013-01-19 18:48 /home/ hosein /downloads/Phpfiles-org.tar.bz2
40K 2013-01-19 18:48 / home / hosein / downloads
12K 2013-01-19 18:32 /home/ hosein /uploadprogress-1.0.3.1.tgz
4.0K 2012-10-13 00:11 /home/ hosein /.bash_history
4.0K 2012-10-12 22:32 /home/ hosein /.bash_profile
0 2013-01-19 18:32 /home/ hosein /xyz.txt
```

```
0 2013-01-19 18:32 /home/ hosein /abc.txt
4.0K 2012-10-12 22:32 /home/ hosein /.mozilla/plugins
4.0K 2012-10-12 22:32 /home/ hosein /.mozilla/extensions
12K 2012-10-12 22:32 /home/ hosein /.mozilla
4.0K 2012-10-12 22:32 /home/ hosein /.bashrc
24K 2013-01-19 18:32 /home/ hosein /Phpfiles-org.tar.bz2
4.0K 2013-01-19 18:32 /home/ hosein /geoipupdate.sh
4.0K 2012-10-12 22:32 /home/ hosein /.zshrc
120K 2013-01-19 18:32 /home/ hosein /goaccess-0.4.2.tar.gz.1
673M 2013-01-19 18:51 /home/ hosein /Ubuntu-12.10/ubuntu-12.10-server-i386.iso
673M 2013-01-19 18:51 /home/ hosein /Ubuntu-12.10
674M 2013-01-19 18:52 / home /  hosein
```

دستور ncdu

ncdu (NCurses Disk Usage) یک دستور معروف du است. این یک راه سریع برای دیدن فهرست هایی است که از فضای دیسک شما استفاده می کنند.

۱.حالت بی صدا

```
$ ncdu -q
```

۲. فهرست های نصب شده را حذف میکند

```
$ ncdu -q -x
```

ساختار دستور:

```
$ ncdu [-hqvx] [--exclude PATTERN] [-X FILE] dir
```

- h – یک پیام راهنمایی کوچک را چاپ میکند
- q – حالت سکوت هنگام محاسبه فضای دیسک، ncdu به طور پیش فرض صفحه را ۱۰ بار در ثانیه به روز می‌کند، این مقدار در حالت بی صدا به هر ۲ ثانیه یک بار کاهش می یابد. از این ویژگی برای صرفه جویی در پهنای باند از طریق اتصالات راه دور استفاده کنید.
- v – نسخه چاپی.
- x – فقط فایل‌ها و فهرست‌ها را در سیستم فایلی که dir مشخص شده است، میشمارد.
- exclude Exclude –– فایل هایی که با PATTERN مطابقت دارند را نادیده میگیرد.
- exclude -f Exclude –– فایل هایی که با هر الگوی در FILE مطابقت دارند را نادیده میگیرد.

مثال های دستور:

-h :این گزینه حجم فایل‌ها و فهرست‌ها را به صورت خوانا نمایش می‌دهد. به عنوان مثال:

```
$ncdu -h /path/to/directory
```

-q :با استفاده از این گزینه، ncduبه صورت ساکت عمل می‌کند و هیچ پیام پیشرفتی نشان نمی‌دهد. مثال:

```
$ncdu -q /path/to/directory
```

-v :این گزینه به شما امکان می‌دهد جزئیات بیشتری درباره فایل‌ها و فهرست‌ها مشاهده کنید. به عنوان مثال:

```
$ncdu -v /path/to/directory
```

-x :با استفاده از این گزینه، ncduفقط فایل‌ها و فهرست‌های در دسترس را بررسی می‌کند و به فایل‌ها و فهرست‌هایی که قابل دسترسی نیستند تغییر نمیدهد. مثال:

```
$ncdu -x /path/to/directory
```

--exclude PATTERN: این گزینه به شما امکان می‌دهد الگوهایی را که می‌خواهید از بررسی مستثنی کنید، مشخص کنید. به عنوان مثال:

```
$ncdu --exclude '*.log' /path/to/directory
```

-X FILE: با استفاده از این گزینه، می‌توانید یک فایل حاوی الگوهای مستثنی را مشخص کنید. مثال:

```
$ncdu -X excluded.txt /path/to/director
```

دستور mount

همه پرونده ها در سیستم فایل لینوکس به صورت یک درخت بزرگ ریشه در " / " مرتب شده اند . این پرونده ها را می توان در دستگاه های مختلف بر اساس جدول پارتیشن شما پخش کرد ، در ابتدا فهرست والدین شما به این درخت نصب می شو. از دستور **mount** برای لیست سیستم فایل موجود در دستگاه به ساختار درخت بزرگ که در " / " ریشه دارد استفاده می شود.

ساختار دستور:

```
mount -t type device dir
```

مثال های دستور:

```
mount [-l | -h | -V]
```

```
mount -a [-fFnrsvw] [-t fstype] [-O optlist]
```

```
mount [-fnrsvw] [-o options] device|dir
```

```
mount [-fnrsvw] [-t fstype] [-o options] device dir
```

/ etc / fstab معمولاً حاوی اطلاعاتی در مورد اینکه کدام دستگاه باید در کجا نصب شود وجود دارد. بیشتردستگاه‌ها توسط پرونده هایی مانند /dev/sda4 و غیره نشان داده می‌شوند اما برای‌سیستم های پرونده خاص می‌تواند متفاوت باشد. اشکال مختلف تا حدودی مورد بحث قرار می گیرند زیرا محدودیت های خاصی در هسته های مختلف دارد.

مثال های دستور:

- اطلاعات مربوط به سیستم فایل های نصب شده را نمایش می دهد:

```
vivek@vivek-X556UQK:~$ sudo mount -l -t ext4
/dev/sda3 on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)
/dev/sda4 on /media/vivek type ext4 (rw,relatime,data=ordered)
/dev/sda5 on /media/vivek type ext4 (rw,relatime,data=ordered)
vivek@vivek-X556UQK:~$ sudo mount -l -t fuseblk
/dev/sda6 on /media/vivek type fuseblk (rw,relatime,user_id=0,group_id=0,allow_o
ther,blksize=4096)
vivek@vivek-X556UQK:~$
```

- سیستم های پرونده را ماونت می کند:

```
vivek@vivek-X556UQK:~$ sudo mount -l -t fuseblk
/dev/sda6 on /media/vivek type fuseblk (rw,relatime,user_id=0,group_id=0,allow_o
ther,blksize=4096)
vivek@vivek-X556UQK:~$
```

- اطلاعات نسخه را نمایش می دهد:

```
vivek@vivek-X556UQK:~$ sudo mount -V
mount from util-linux 2.27.1 (libmount 2.27.0: selinux, assert, debug)
vivek@vivek-X556UQK:~$
```

- پیاده سازی سیستم های پرونده:

```
vivek@vivek-X556UQK:~$ sudo umount /dev/sda6
vivek@vivek-X556UQK:~$ sudo umount /dev/sda5
vivek@vivek-X556UQK:~$ sudo umount /dev/sda4
vivek@vivek-X556UQK:~$ sudo mount -l -t fuseblk
vivek@vivek-X556UQK:~$ sudo mount -l -t ext4
/dev/sda3 on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)
vivek@vivek-X556UQK:~$
```

مونت کردن cd/dvd

برای نصب cdیا دی وی دی در لینوکس مراحل زیر را دنبال کنید .با قرار دادن دیسک در رایانه خود شروع کنید، سپس مراحل زیر را طی کنید: ابتدا سعی کنید از blkidدستور استفاده کنید تا ببینیددستگاه شما در چه پارتیشنی قراردارد. .معمولاً این مسیر خواهد بود /dev/sr0، اما ممکن است مال شما مسیر دیگری باشد.

بعد، یک نقطه اتصال برای جایی که می‌خواهید سی‌دی را در آن نصب کنید، ایجاد کنید .یا، اگر از قبل جایی در ذهن دارید، فقط یک فهرست خالی در جایی از رایانه خود انتخاب کنید.

```
$ sudo mkdir /mnt/cdrom
```

اکنون می توانیم از دستور mount برای نگاشت فایل دستگاه به فهرست که به تازگی ایجاد کرده ایم استفاده کنیم.

```
$ sudo mount /dev/sr0 /mnt/cdrom
```

CDیا DVD شما اکنون باید در فهرست که در آن نصب شده است قابل دسترسی باشد.

```
$ls /mnt/
$ cd /mnt/cdrom
```

همچنین می توانید با استفاده از mount دستور بدون گزینه های بیشتر تأیید کنید که mount موفقیت آمیز بوده است .

```
$ mount | grep cdrom
```

وقتی کارتان تمام شد، فقط cdرا جدا کرده و از دیسک خارج کنید.

```
$ sudo umount /mnt/cdrom
```

دستور dump

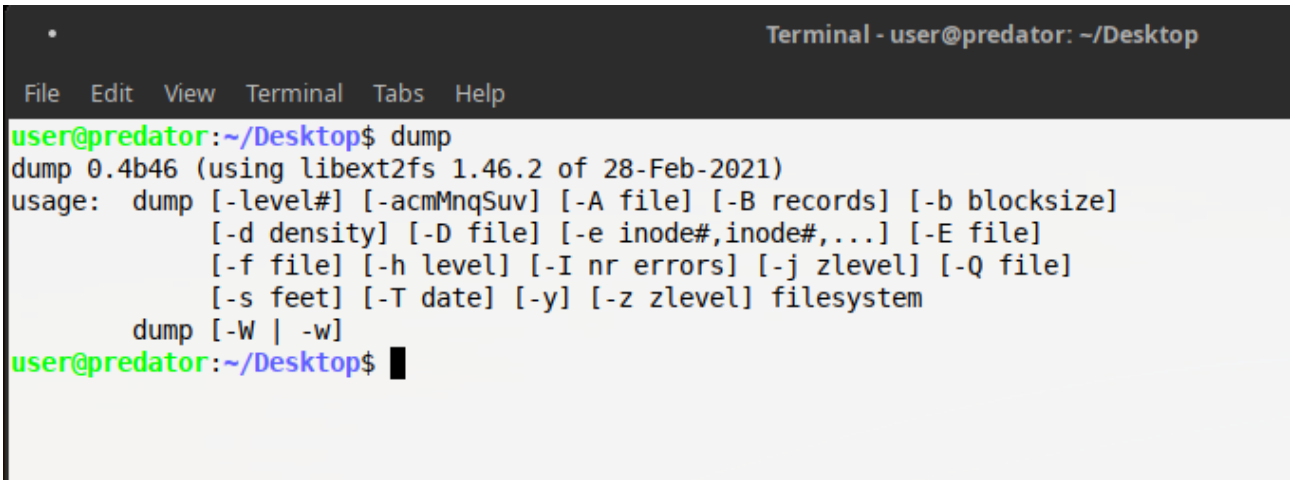
برای تهیه نسخه پشتیبان از سیستم فایل در برخی از دستگاه های ذخیره سازی استفاده می شود .از سیستم فایل کامل پشتیبان تهیه می کند و از پرونده های منفرد .به عبارت دیگر ، از پرونده های مورد نیاز برای ذخیره امن در نوار ، دیسک یا هر دستگاه ذخیره سازی دیگر پشتیبان تهیه می کند . تنها با این نسخه سیستم فایل ext3 / ext2کار میکند. منظور از پشتیبان گیری افزایشی ، کاربر می تواند یک طرح پشتیبان تهیه کند که طبق آن مثلا هفتگی یا طبق برنامه زمانی از سیستم فایلها پشتیبان تهیه کند یا فقط از آن فایلهایی که اخیراً تغییر کرده یا اضافه شده اند ، پشتیبان تهیه می شود.

ساختار دستور:

```
dump [-level#] [-a autosize] [-A file] [-B records] [-b blocksize]
[-d density] [-D file] [-e inode numbers] [-E file] [-f file]
[-F script] [-h level] [-I nr errors] [-jcompression level] [-L label]
[-Q file] [-s feet] [-T date] [-y] [-zcompression level] files-to-dump
```

دستور dump بدون هیچ گزینه

عمومی دستور را به همراه گزینه های مختلفی که می توان با دستور dump استفاده کرد چاپ می‌کند .همچنین شماره نسخه دستور dumpمورد استفاده را چاپ می کند.



گزینه های دستور:

- **-level** :سطح dump که یک عدد صحیح و از ۰–۹ است .اگر کاربر خواسته است پشتیبان گیری کامل یا پشتیبان گیری از فقط فایلهای جدید اضافه شده پس از آخرین dump سطح پایین را انجام دهد.
- **-f** : این فایلی را مشخص می کند که در آن نسخه پشتیبان تهیه شود .این پرونده می تواند یک درایو نوار ، یک فلاپی دیسک ، یک فایل معمولی یا یک خروجی استاندارد باشد.
- **-u** : این پشتیبان گیری را در پرونده ***etc/dumpdates*** / ضبط و به روز می کند.

```
sudo dump 0uf / dev / qft0 / dev / sda6
```

```
anubhav@anubhav-ThinkPad-Yoga-460:~
anubhav@anubhav-ThinkPad-Yoga-460:~$ sudo dump 0uf  /dev/qft0 /dev/sda6
[sudo] password for anubhav:
 DUMP: Date of this level 0 dump: Mon Jun  3 11:19:35 2019
 DUMP: Dumping /dev/sda6 (/) to /dev/qft0
 DUMP: Label: none
 DUMP: Writing 10 Kilobyte records
 DUMP: mapping (Pass I) [regular files]
 DUMP: mapping (Pass II) [directories]
 DUMP: estimated 19563251 blocks.
 DUMP: Volume 1 started with block 1 at: Mon Jun  3 11:19:39 2019
 DUMP: dumping (Pass III) [directories]
 DUMP: dumping (Pass IV) [regular files]
 DUMP: End of tape detected
 DUMP: Closing /dev/qft0
 DUMP: Volume 1 completed at: Mon Jun  3 11:21:44 2019
 DUMP: Volume 1 3929900 blocks (3837.79MB)
 DUMP: Volume 1 took 0:02:05
 DUMP: Volume 1 transfer rate: 31438 kB/s
 DUMP: Change Volumes: Mount volume #2
 DUMP: Is the new volume mounted and ready to go?: ("yes" or "no")
```

B-مقدار داده ای را که می تواند در نوار جای‌گیرد را نشان می‌دهد .این همیشه لازم نیست زیرا **dump** میتواند پایان رسانه یا فضای ذخیره سازی را تشخیص دهد .این مقدار عددی می گیرد و در ترکیب با **b**- استفاده می شود

b-اندازه بلوک تعداد کیلوبایت در هر سابقه dump را مشخص می‌کند .مقدار پیش فرض اندازه بلوک **10 است** . در اینجا توجه کنید که چگونه در خط ۴ DUMP اندازه بلوک به جای ۱۰ ، ۲۰ است.

sudo dump 0uBbf / dev / qft0 / dev / sda6

```
anubhav@anubhav-ThinkPad-Yoga-460:~$ sudo dump 0uBbf 5000000 20  /dev/qft0 /dev/sda6
[sudo] password for anubhav:
 DUMP: Date of this level 0 dump: Mon Jun  3 11:35:43 2019
 DUMP: Dumping /dev/sda6 (/) to /dev/qft0
 DUMP: Label: none
 DUMP: Writing 20 Kilobyte records
 DUMP: mapping (Pass I) [regular files]
 DUMP: mapping (Pass II) [directories]
 DUMP: estimated 19575066 blocks on 3.91 tape(s).
 DUMP: Volume 1 started with block 1 at: Mon Jun  3 11:35:48 2019
 DUMP: dumping (Pass III) [directories]
 DUMP: dumping (Pass IV) [regular files]
 DUMP: End of tape detected
 DUMP: Closing /dev/qft0
 DUMP: Volume 1 completed at: Mon Jun  3 11:37:37 2019
 DUMP: Volume 1 3929920 blocks (3837.81MB)
 DUMP: Volume 1 took 0:01:49
 DUMP: Volume 1 transfer rate: 36053 kB/s
 DUMP: Change Volumes: Mount volume #2
 DUMP: Is the new volume mounted and ready to go?: ("yes" or "no") yes
 DUMP: Volume 2 started with block 3929861 at: Mon Jun  3 11:38:27 2019
 DUMP: Volume 2 begins with blocks from inode 8003216
 DUMP: End of tape detected
 DUMP: Closing /dev/qft0
 DUMP: Volume 2 completed at: Mon Jun  3 11:40:30 2019
 DUMP: Volume 2 3929919 blocks (3837.81MB)
 DUMP: Volume 2 took 0:02:03
 DUMP: Volume 2 transfer rate: 31950 kB/s
 DUMP: Change Volumes: Mount volume #3
 DUMP: Is the new volume mounted and ready to go?: ("yes" or "no")
```

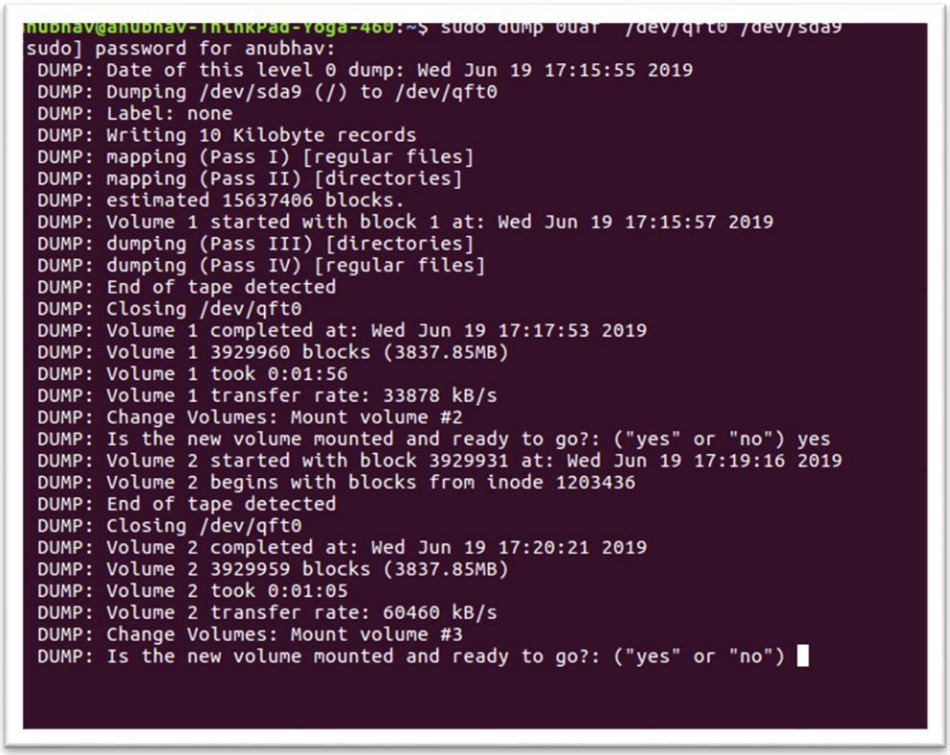
W- سیستم های پرونده ای را که باید پشتیبان تهیه شوند لیست می کند.

dump -W

```
anubhav@anubhav-ThinkPad-Yoga-460:~$ dump -W
Last dump(s) done (Dump '>' file systems):
  /dev/sda9      (     /) Last dump: Level 0, Date Tue Jun 18 12:53:45 2019
anubhav@anubhav-ThinkPad-Yoga-460:~$
```

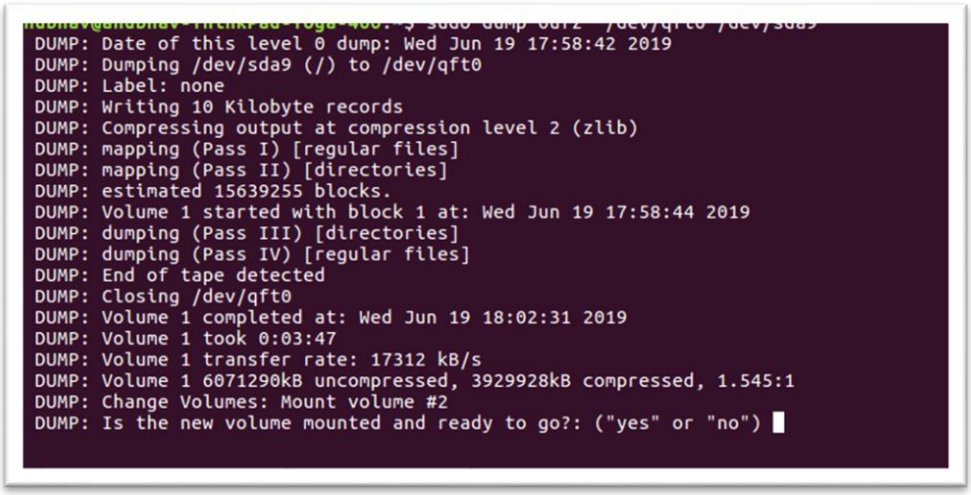
a- مخفف 'auto-size' است .این مفید است زیرا به ما کمک می کند تا همه محاسبات طول نوار را دور بزنیم و وقتی مجبور به پیوستن به دیسک موجود هستیم بهترین عملکرد را دارد.

sudo dump 0uaf / dev / qft0 / dev / sda9



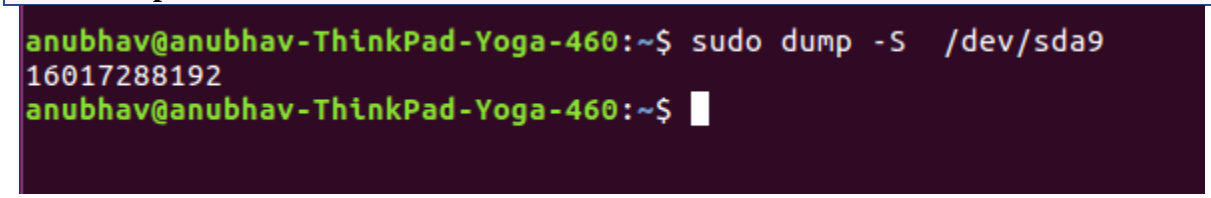
z- با استفاده از کتابخانه zlib ، هر بلوکی را که باید روی نوار نوشته شود فشرده می کند .سطح فشرده سازی پیش فرض ۲ است.

sudo dump 0ufz / dev / qft0 / dev / sda6



گزینه **S** را حتی قبل از انجام آن تخمین می زنید .خروجی تعداد تخمینی بایت های dump شده است .درصورت dump های افزایشی برای تعیین میزان استفاده از حجم رسانه ها(فضای ذخیره سازی) مفید است.

sudo dump S / dev / sda6



I- خطاهای NR نادیده می گیرد. به طور پیش فرض اول ۳۲ عنوان خوانده شده خطا .این مقدار را می توان با توجه به انتخاب کاربر به صورت دستی به هر مقداری تغییر داد.

sudo dump 0ufI20 / dev / qft0 / dev / sda9

```
anubhav@anubhav-ThinkPad-Yoga-460:~$ sudo dump 0ufI20  /dev/qft0 /dev/sda9

  DUMP: Date of this level 0 dump: Wed Jun 19 18:58:38 2019
  DUMP: Dumping /dev/sda9 (/) to /dev/qft0
  DUMP: Label: none
  DUMP: Writing 10 Kilobyte records
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 15627219 blocks.
  DUMP: Volume 1 started with block 1 at: Wed Jun 19 18:58:39 2019
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: End of tape detected
  DUMP: Closing /dev/qft0
  DUMP: Volume 1 completed at: Wed Jun 19 19:00:41 2019
  DUMP: Volume 1 3929960 blocks (3837.85MB)
  DUMP: Volume 1 took 0:02:02
  DUMP: Volume 1 transfer rate: 32212 kB/s
  DUMP: Change Volumes: Mount volume #2
  DUMP: Is the new volume mounted and ready to go?: ("yes" or "no")   DUMP: "Yes" or "No"?
  DUMP: Is the new volume mounted and ready to go?: ("yes" or "no") yes
  DUMP: Volume 2 started with block 3929931 at: Wed Jun 19 19:09:28 2019
  DUMP: Volume 2 begins with blocks from inode 1203436
  DUMP: 25.15% done at 39 kB/s, finished in 0:06
  DUMP: End of tape detected
  DUMP: Closing /dev/qft0
  DUMP: Volume 2 completed at: Wed Jun 19 19:10:31 2019
  DUMP: Volume 2 3929959 blocks (3837.85MB)
  DUMP: Volume 2 took 0:01:03
  DUMP: Volume 2 transfer rate: 62379 kB/s
  DUMP: Change Volumes: Mount volume #3
  DUMP: Is the new volume mounted and ready to go?: ("yes" or "no")
```

دستور dumpe2fs

از دستور dumpe2fs برای چاپ super block استفاده می شود و اطلاعات گروه را برای سیستم فایل موجود در دستگاه مسدود می کند . دستور dumpe2fs یک ابزار مفید در سیستم‌عامل‌های لینوکس است که برای نمایش اطلاعات مربوط به سیستم فایل ext2، ext3یا ext4 استفاده می‌شود. این دستور امکان مشاهده جزئیات فایل سیستم، مانند اطلاعات مربوط به بلاک‌ها، inodeها، تاریخچه سیستم فایل و غیره را فراهم می‌کند.

umount device

ساختار دستور:

dumpe2fs [-bfhixV] [-o superblock=superblock] [-o blocksize=block-size] device

گزینه های دستور:

- **b** برای بدست آوردن بلاکهایی که به عنوان بد در سیستم فایل ذخیره شده اند ، استفاده می شود.
- **o superblock = superblock** در بررسی سیستم سیستم از superblock استفاده می کند .این گزینه معمولاً مورد نیاز نیست مگر توسط یک سیستم فایل که در حال بررسی بقایای یک سیستم پرونده بسیار خراب است.
- **o blocksize = blocksize** از blockize در هنگام بررسی سیستم فایل استفاده می کند .این گزینه معمولاً مورد نیاز نیست مگر توسط یک سیستم فایل که در حال بررسی بقایای یک سیستم پرونده بسیار خراب است.
- **f** برای مجبور کردن dumpe2fs برای نمایش سیستم پرونده استفاده میشود حتی اگر ممکن است دارای برخی از پرچم های ویژگی سیستم فایل باشد که dumpe2fs ممکن است درک نکند.

- **h** این فقط اطلاعات superblock را نمایش میدهد و نه هیچ یک از اطلاعات جزئیات توصیف کننده گروه.block
- **i** برای نشان دادن داده‌های سیستم فایل از یک فایل تصویری ایجاد شده توسط e2image ، با استفاده از دستگاه به عنوان نام فایل پرونده استفاده می شود.
- **x** نمایش شماره بلوک دقیق اطلاعات گروه در قالب هگزادسیمال.
- **V** شماره نسخه dumpe2fs را نمایش داده و خارج شوید.

Superblock این یک رکورد از مشخصات سیستم فایل است .این شامل اطلاعاتی در مورد: اندازه ، اندازه بلوک ، خالی و بلوک های پر شده است . لینوکس همچنین کپی superblock خود را در حافظه نگهداری میکند .اگر نمی توانید دستگاه خود را نصب کنید ، این می تواند به دلیل خرابشدن superblock باشد.

مثال‌ها:

نمایش اطلاعات:superblock

```
sudo dumpe2fs -h /dev/sda3
```

```
vivek@vivek-X556UQK:~$ sudo dumpe2fs -h /dev/sda3
dumpe2fs 1.42.13 (17-May-2015)
Filesystem volume name:   <none>
Last mounted on:          /
Filesystem UUID:          1affe225-2aa1-4562-8136-632ad37a15e1
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype needs_recovery extent flex_bg sparse_super large_file huge_file uninit_bg dir_nlink extra_isize
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              12500992
Block count:              50000128
Reserved block count:     2500006
Free blocks:              46282972
Free inodes:              12176106
First block:              0
Block size:               4096
Fragment size:            4096
Reserved GDT blocks:      1012
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         8192
Inode blocks per group:   512
Flex block group size:    16
Filesystem created:       Wed Mar  6 01:05:48 2019
Last mount time:          Wed Mar  6 19:27:51 2019
Last write time:          Wed Mar  6 19:27:46 2019
Mount count:              5
Maximum mount count:      -1
Last checked:             Wed Mar  6 01:05:48 2019
Check interval:           0 (<none>)
Lifetime writes:          21 GB
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               256
Required extra isize:     28
Desired extra isize:      28
Journal inode:            8
First orphan inode:       1583960
Default directory hash:   half_md4
Directory Hash Seed:      c212ddd9-a688-4600-b454-4214ddf17813
Journal backup:           inode blocks
Journal features:         journal_incompat_revoke
Journal size:             128M
Journal length:           32768
Journal sequence:         0x00005cb8
Journal start:            8221
```

نمایش اطلاعات گروه های بلوک:

```
sudo dumpe2fs /dev/sda3
```

```
Group 0: (Blocks 0-32767) [ITABLE_ZEROED]
  Checksum 0x4f00, unused inodes 8176
  Primary superblock at 0, Group descriptors at 1-12
  Reserved GDT blocks at 13-1024
  Block bitmap at 1025 (+1025), Inode bitmap at 1041 (+1041)
  Inode table at 1057-1568 (+1057)
  23513 free blocks, 8177 free inodes, 2 directories, 8176 unused inodes
  Free blocks: 9255-32767
  Free inodes: 14, 17-8192
Group 1: (Blocks 32768-65535) [INODE_UNINIT, ITABLE_ZEROED]
  Checksum 0xcdf4, unused inodes 8192
  Backup superblock at 32768, Group descriptors at 32769-32780
  Reserved GDT blocks at 32781-33792
  Block bitmap at 1026 (bg #0 + 1026), Inode bitmap at 1042 (bg #0 + 1042)
  Inode table at 1569-2080 (bg #0 + 1569)
  30918 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes
  Free blocks: 33793-34335, 34417-34431, 34773-34783, 34814-42431, 42489-42495, 42637-42687, 42741-42815, 42850-59300, 59389-65535
  Free inodes: 8193-16384
Group 2: (Blocks 65536-98303) [INODE_UNINIT, ITABLE_ZEROED]
  Checksum 0x7465, unused inodes 8192
  Block bitmap at 1027 (bg #0 + 1027), Inode bitmap at 1043 (bg #0 + 1043)
  Inode table at 2081-2592 (bg #0 + 2081)
  21939 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes
  Free blocks: 65601-66559, 67071, 67486-67519, 67639-67647, 67766-67839, 68018-68031, 68094-68095, 68224-68255, 68280-68351, 68639-68863, 69009-69599,
2159, 94027-98303
  Free inodes: 16385-24576
Group 3: (Blocks 98304-131071) [INODE_UNINIT, ITABLE_ZEROED]
  Checksum 0x75c1, unused inodes 8192
  Backup superblock at 98304, Group descriptors at 98305-98316
  Reserved GDT blocks at 98317-99328
  Block bitmap at 1028 (bg #0 + 1028), Inode bitmap at 1044 (bg #0 + 1044)
  Inode table at 2593-3104 (bg #0 + 2593)
  23957 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes
  Free blocks: 99336-99375, 99386-99388, 99570-99583, 99713-99750, 101043-101119, 101134-101247, 101257-124927
  Free inodes: 24577-32768
Group 4: (Blocks 131072-163839) [INODE_UNINIT, ITABLE_ZEROED]
  Checksum 0x6cfb, unused inodes 8192
  Block bitmap at 1029 (bg #0 + 1029), Inode bitmap at 1045 (bg #0 + 1045)
  Inode table at 3105-3616 (bg #0 + 3105)
  4414 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes
  Free blocks: 135082-135103, 135154-135167, 154651-154655, 154674-154687, 154734-154751, 154836-154879, 155013-155135, 155473-155519, 155617-159743
  Free inodes: 32769-40960
Group 5: (Blocks 163840-196607) [INODE_UNINIT, ITABLE_ZEROED]
  Checksum 0xd262, unused inodes 8192
  Backup superblock at 163840, Group descriptors at 163841-163852
  Reserved GDT blocks at 163853-164864
  Block bitmap at 1030 (bg #0 + 1030), Inode bitmap at 1046 (bg #0 + 1046)
  Inode table at 3617-4128 (bg #0 + 3617)
  14673 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes
  Free blocks: 165048-165179, 165810-165887, 167936-182271, 190337-190463
  Free inodes: 40961-49152
Group 6: (Blocks 196608-229375) [INODE_UNINIT, ITABLE_ZEROED]
:
```

این اطلاعات مربوط به گروه های بلوک را نشان می دهد .

```
Group 0: (Blocks 0-32767) [ITABLE_ZEROED]
  Checksum 0x4f00, unused inodes 8176
  Primary superblock at 0, Group descriptors at 1-12
  Reserved GDT blocks at 13-1024
  Block bitmap at 1025 (+1025), Inode bitmap at 1041 (+1041)
  Inode table at 1057-1568 (+1057)
  23513 free blocks, 8177 free inodes, 2 directories, 8176 unused inodes
  Free blocks: 9255-32767
  Free inodes: 14, 17-8192
```

مشاهده بیشتر درباره superblock ها:

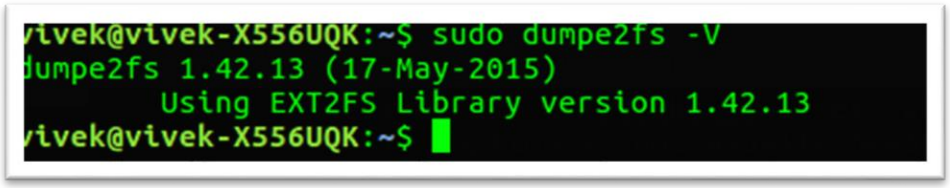
```
sudo dumpe2fs /dev/sda4 | grep -i superblock
```

```
dumpe2fs 1.42.13 (17-May-2015)
  Primary superblock at 0, Group descriptors at 1-12
  Backup superblock at 32768, Group descriptors at 32769-32780
  Backup superblock at 98304, Group descriptors at 98305-98316
  Backup superblock at 163840, Group descriptors at 163841-163852
  Backup superblock at 229376, Group descriptors at 229377-229388
  Backup superblock at 294912, Group descriptors at 294913-294924
  Backup superblock at 819200, Group descriptors at 819201-819212
  Backup superblock at 884736, Group descriptors at 884737-884748
  Backup superblock at 1605632, Group descriptors at 1605633-1605644
  Backup superblock at 2654208, Group descriptors at 2654209-2654220
  Backup superblock at 4096000, Group descriptors at 4096001-4096012
  Backup superblock at 7962624, Group descriptors at 7962625-7962636
  Backup superblock at 11239424, Group descriptors at 11239425-11239436
  Backup superblock at 20480000, Group descriptors at 20480001-20480012
  Backup superblock at 23887872, Group descriptors at 23887873-23887884
vivek@vivek-X556UQK:~$
```

تمام فوق بلوک های موجود را با مخلوط کن ها نمایش می دهد.

نمایش نسخه:dumpe2fs

```
sudo dumpe2fs -V
```



دستور fdisk

fdiskهمچنین به عنوان disk ابزار مدیریت دیسک شناخته می شود ، یک دستورمحاوره ای در Linux است که برای ایجاد و دستکاری جدول پارتیشن دیسک استفاده می شود .برای مشاهده ، ایجاد ، حذف ، تغییر ، تغییر اندازه ، کپی و جابجایی پارتیشن ها بر روی یک دیسک سخت با استفاده از رابط دیالوگ محور استفاده می شود. fdiskبه شما امکان میدهد حداکثر چهار پارتیشن اصلی ایجاد کنید و تعداد پارتیشن منطقی به اندازه دیسک سختی که استفاده می کنید بستگی دارد .این به کاربر اجازه می دهد:

- ب‌رای ایجاد فضای برای پارتیشن های جدید.
- س‌ازماندهی فضا برای درایوهای جدید.
- س‌ازماندهی مجدد درایوهای قدیمی.
- ک‌پی یا انتقال داده ها روی دیسک های جدید –پارتیشن ها

```
fdsik [options] device
```

یا

```
fdisk -l [device...]
```

مشاهده همه پارتیشن های دیسک :

اولین کاری که باید قبل از انجام هر کاری با دیسک ها و پارتیشن انجام شود مشاهده جزئیات اساسی در مورد تمام پارتیشن های موجود در سیستم است .از دستور زیر برای لیست پارتیشن های سیستم شما و دیدن نام **dev** / آنها استفاده می شود .به عنوان مثال :

```
/ dev / sda  یا / dev/sdb یا /dev/sdc
$ sudo fdisk -l
```

```
linux@ubuntu:~$ sudo fdisk -l
Disk /dev/loop0: 140.9 MiB, 147722240 bytes, 288520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop1: 140.7 MiB, 147496960 bytes, 288080 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop2: 14.5 MiB, 15196160 bytes, 29680 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop3: 3.7 MiB, 3887104 bytes, 7592 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop4: 13 MiB, 13619200 bytes, 26600 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop5: 34.7 MiB, 36323328 bytes, 70944 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop6: 91 MiB, 95416320 bytes, 186360 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop7: 34.8 MiB, 36503552 bytes, 71296 sectors
Units: sectors of 1 * 512 = 512 bytes
```

مشاهده پارتیشن در یک دیسک خاص:

از دستور زیر برای مشاهده تمام پارتیشن های دیسک در دستگاه **/dev/** **sda** استفاده می شود.

```
$ sudo fdisk -l /dev/sda
```

```
linux@ubuntu:~$ sudo fdisk -l /dev/sda
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xa4f02e0d

Device     Boot Start      End  Sectors Size Id Type
/dev/sda1  *     2048 41940991 41938944  20G 83 Linux
linux@ubuntu:~$
```

مشاهده تمام دستوراتfdisk

برای دیدن همه دستوراتی که تحت دستور fdisk در دسترس هستند می توانید از پارتیشن **sda / dev /** با دستور fdisk استفاده کنید.

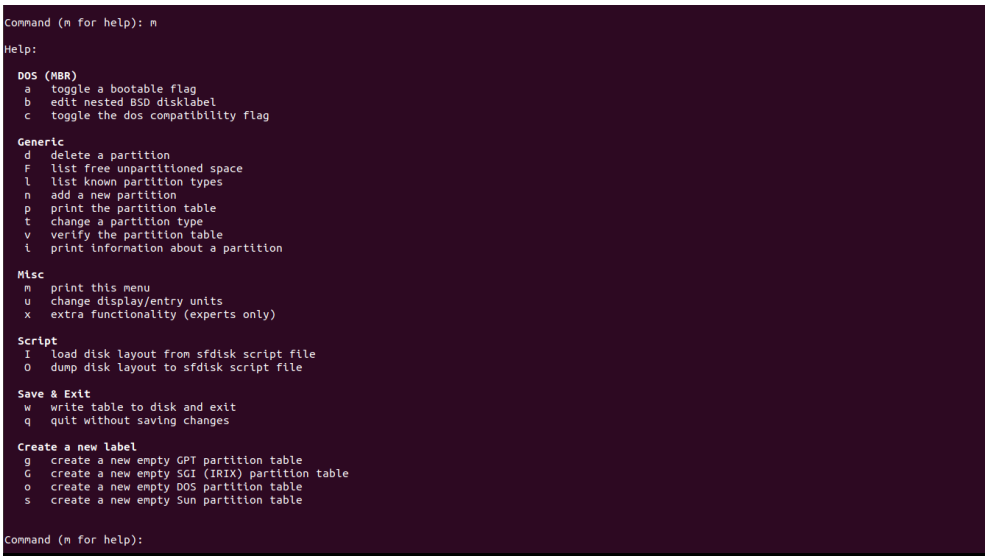
```
$ sudo fdisk /dev/sda
```

```
linux@ubuntu:~$ sudo fdisk /dev/sda

Welcome to fdisk (util-linux 2.31.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help):
```

توجه :این یک دستوررا درخواست می کند .برای دیدن همه عملیاتی که می توانند در **/dev/sda** انجام دهند ، **m را** تایپ کنید . پس از فشردن **m** شما بدست خواهید آورد:



ایجاد یک پارتیشن هارد دیسک :

برای این کار وارد پارتیشن هارد دیسک که پارتیشن **dev / sda /** است ، شوید و از دستور زیر استفاده کنید:

```
$ sudo fdisk /dev/sda
```

اکنون برای ایجاد یک پارتیشن جدید باید **n را** تایپ کنید و سپس **p را** برای ساخت یک پارتیشن اصلی و **e را** برای ساخت یک پارتیشن توسعه یافته یا منطقی بسته به نوع پارتیشن ایجاد کنید.



برای نوشتن تغییرات و راه اندازی مجدد سیستم ، دستور **w را** اجرا کنید.

حذف یک پارتیشن هارد دیسک :

برای حذف یک پارتیشن برای هارد دیسک و آزاد کردن فضای اشغال شده توسط آن پارتیشن برای مثال:

dev / sdb /با استفاده از موارد زیر به منوی دستور بروید:

```
$ sudo fdisk /dev/sda
```

و سپس **d را** تایپ کنید تا به منوی Delete partition بروید .این شماره پارتیشنی را که می خواهید حذف کنید ، فوری می کند (شماره را تایپ کنید)

```
linux@ubuntu:~$ sudo fdisk /dev/sda1

Welcome to fdisk (util-linux 2.31.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): d
Partition number (1,5, default 5): 5

Partition 5 has been deleted.

Command (m for help):
```

برای نوشتن تغییرات و راه اندازی مجدد سیستم ، دستور **w را** اجرا کنید.

نحوه مشاهده اندازه پارتیشن خود:

```
$ sudo fdisk -s /dev/sda
```

```
linux@ubuntu:~$ sudo fdisk -s /dev/sda
20971520
linux@ubuntu:~$
```

توجه داشته باشید:

برای بررسی صفحه دستی دستور fdisk ، از دستور زیر استفاده کنید:

```
$ man fdisk
```

برای دیدن پیام راهنما و لیست همه گزینه ها ، از گزینه دستور زیر استفاده کنید:

```
$ sudo fdisk -h
```

```
linux@ubuntu:~$ sudo fdisk -h

Usage:
 fdisk [options] <disk>      change partition table
 fdisk [options] -l [<disk>] list partition table(s)

Display or manipulate a disk partition table.

Options:
 -b, --sector-size <size>      physical and logical sector size
 -B, --protect-boot            don't erase bootbits when creating a new label
 -c, --compatibility[=<mode>]  mode is 'dos' or 'nondos' (default)
 -L, --color[=<when>]          colorize output (auto, always or never)
                                 colors are enabled by default
 -l, --list                    display partitions and exit
 -o, --output <list>           output columns
 -t, --type <type>             recognize specified partition table type only
 -u, --units[=<unit>]          display units: 'cylinders' or 'sectors' (default)
 -s, --getsz                   display device size in 512-byte sectors [DEPRECATED]
     --bytes                   print SIZE in bytes rather than in human readable format
 -w, --wipe <mode>             wipe signatures (auto, always or never)
 -W, --wipe-partitions <mode>  wipe signatures from new partitions (auto, always or never)

 -C, --cylinders <number>      specify the number of cylinders
 -H, --heads <number>          specify the number of heads
 -S, --sectors <number>        specify the number of sectors per track

 -h, --help                    display this help
 -V, --version                 display version

Available output columns:
 gpt: Device Start End Sectors Size Type Type-UUID Attrs Name UUID
 dos: Device Start End Sectors Cylinders Size Type Id Attrs Boot End-C/H/S Start-C/H/S
 bsd: Slice Start End Sectors Cylinders Size Type Bsize Cpg Fsize
 sgi: Device Start End Sectors Cylinders Size Type Id Attrs
 sun: Device Start End Sectors Cylinders Size Type Id Flags

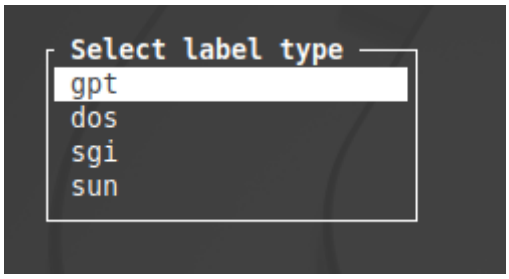
For more details see fdisk(8).
```

دستور cfdisk

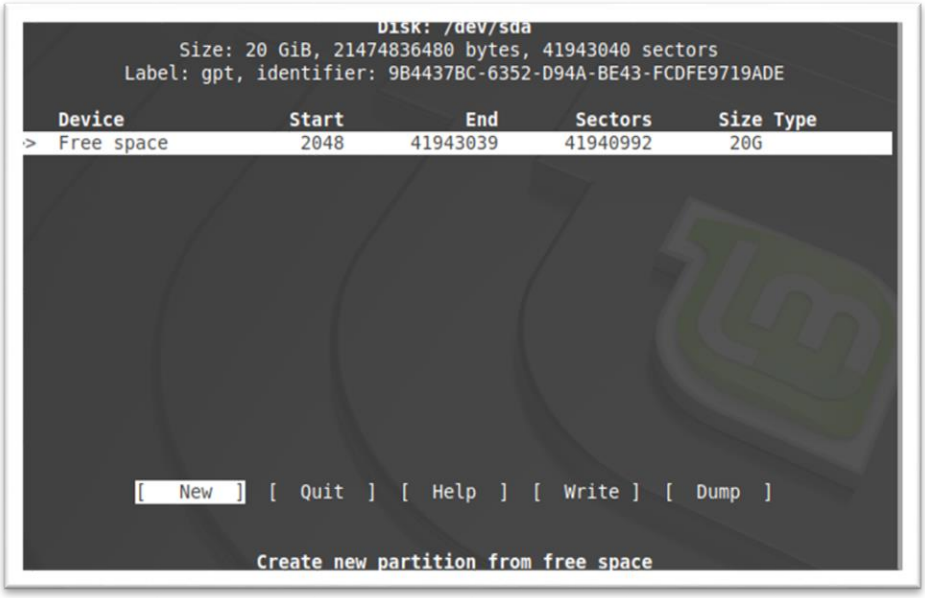
از دستور **cfdisk** برای ایجاد، حذف و اصلاح پارتیشن ها در دستگاه دیسک استفاده می شود .با تهیه یک رابط گرافیکی مبتنی بر متن ، جدول پارتیشن دیسک را نمایش یا دستکاری می کند.

```
cfdisk / dev / sda
```

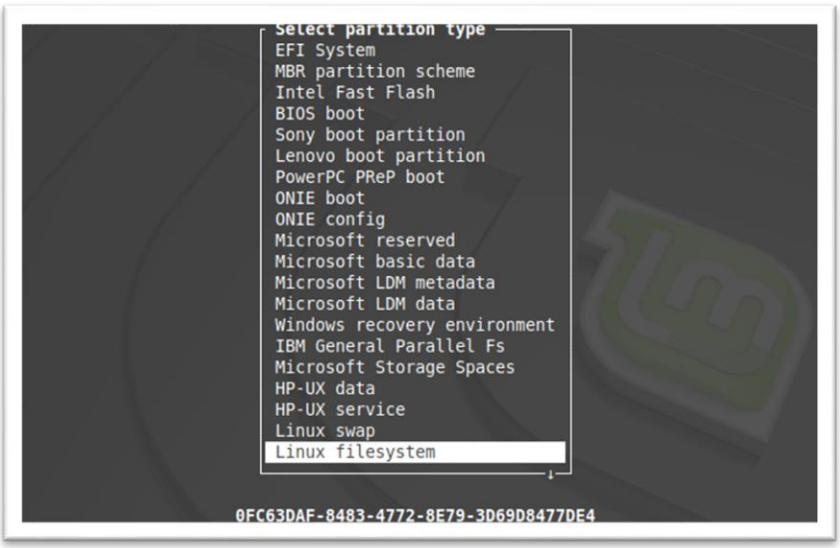
پس از آن ، مانند این موارد اعلانی دریافت می کنید:



gptرا از لیست انتخاب کنید . اکنون یک جدول پارتیشن مانند این مشاهده خواهید کرد:

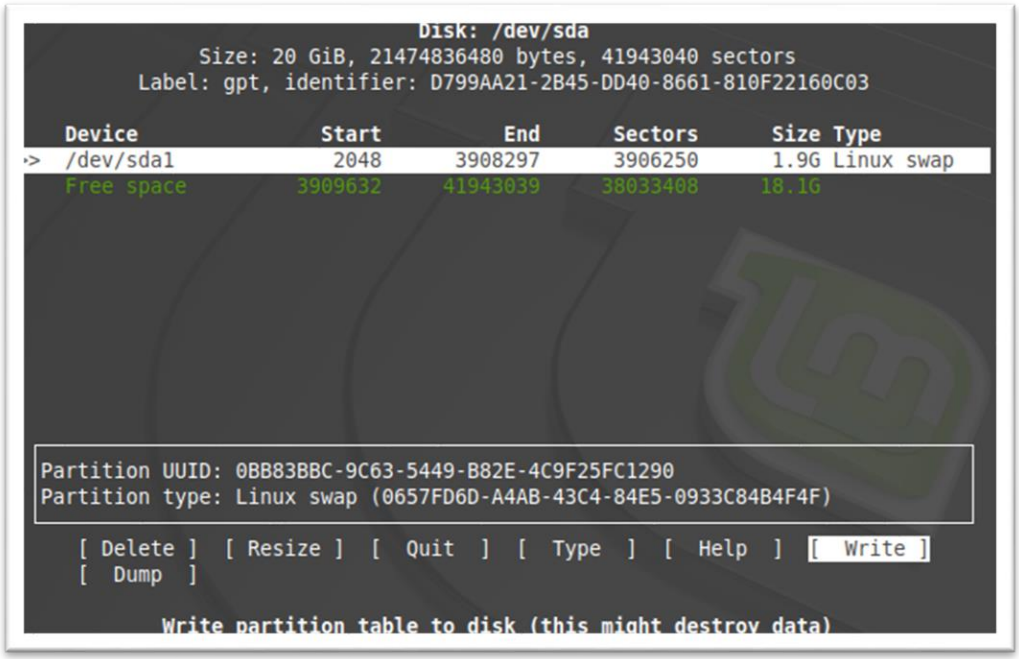


فضای آزاد موجود را مشاهده کنید .در اینجا ما ۲۰ گیگابایت داریم NEW .را انتخاب کرده و یک پارتیشن جدید ایجاد کنید .برای پیمایش از کلیدهای جهت دار به سمت بالا استفاده کنید و برای انتخاب وارد شوید. در صورت نصب سیستم جدید با رابط خط فرمان ، می توانید کارهای زیادی را انجام دهید ، می توانید گزینه ای را برای استفاده از فضای انتخاب شده به عنوان پارتیشن اصلی مشاهده کنید. **مثال** :اندازه ۲ گیگابایت را انتخاب کنید. گزینه > انتخاب کنید . به همین ترتیب می توانیم یک پارتیشن منطقی نیز انجام دهیم.

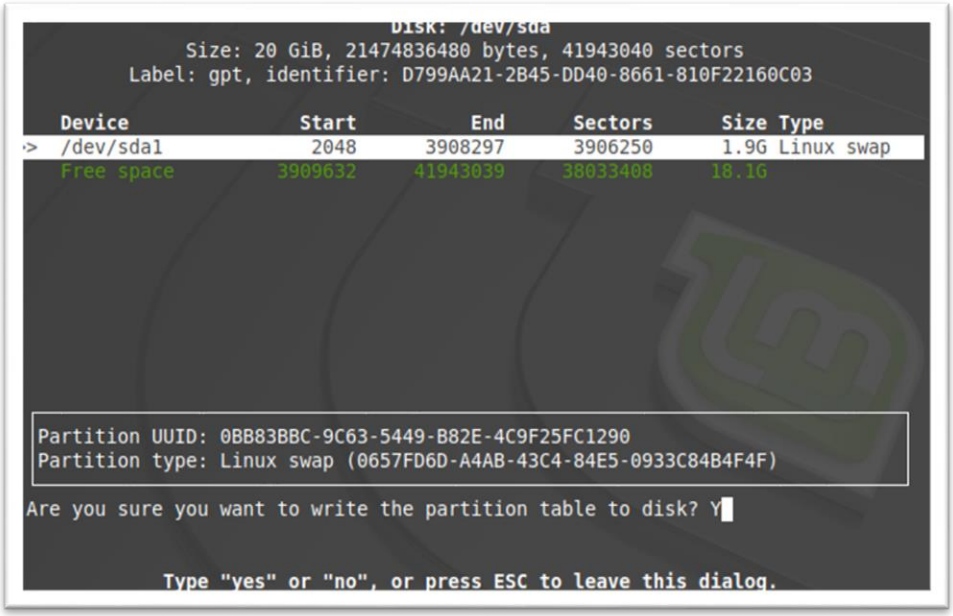


پس از اندازه گیری پارتیشن ، نوع مورد نظر خود را انتخاب کنید ، در مورد من ، من **Linux Swap را** انتخاب می کنم.

پس از انتخاب اندازه و نوع **نوشتن** روی دیسک:



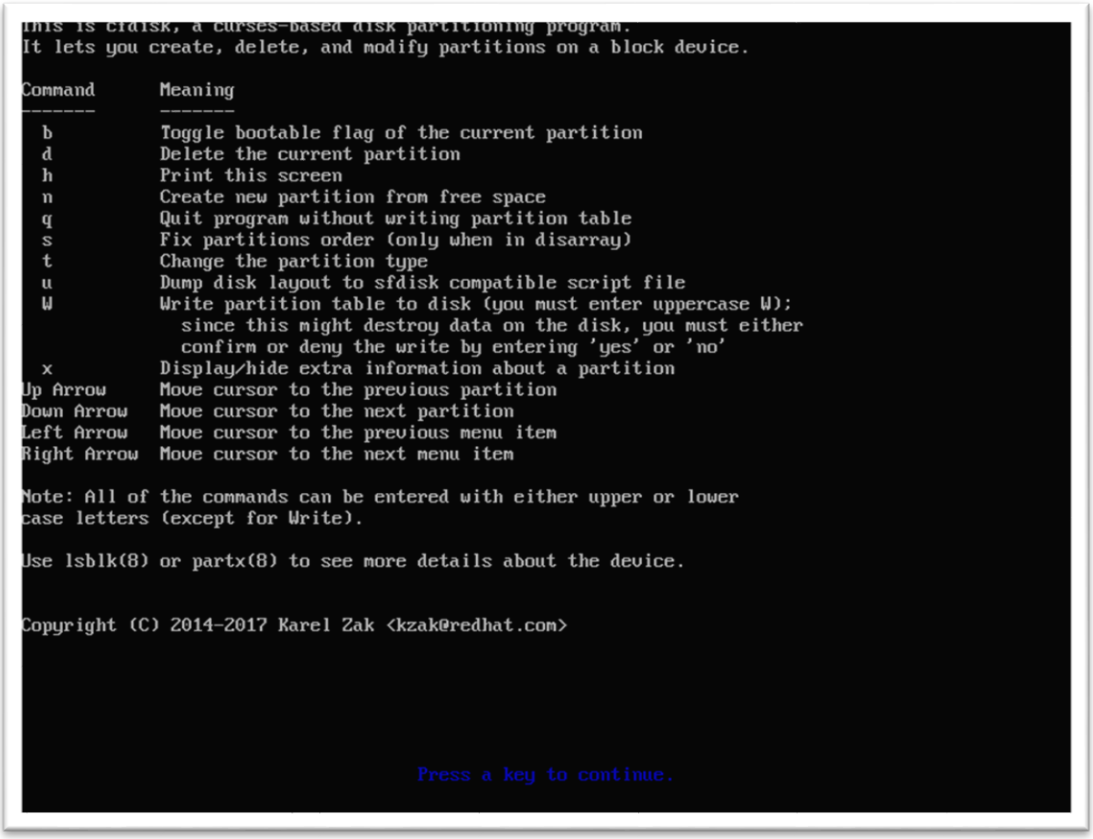
دستورالعملی مانند این را مشاهده خواهید کرد:



گزینه های دستور:

- **-h یا help–** متن راهنما و خروج را نمایش می دهد.
- **-L یا [when =] color–** خروجی را رنگ آمیزی کنید.اگر آرگومان **when** حذف شود ، به طور پیش فرض به صورت خودکار در می آید .رنگها می توانند غیرفعال شوند ، زیرا در حالت پیش فرض داخلی تعبیه شده است.
- **-V یا version:** اطلاعات نسخه را نمایش داده و از آن خارج شوید.
- **-z یا zero:** با یک جدول پارتیشن صفر در حافظه شروع کنید .این گزینه جدول پارتیشن بندی را روی دیسک صفر نمی کند .بلکه به راحتی برنامه را بدون خواندن جدول پارتیشن موجود شروع می کند .این گزینه به شما امکان می دهد یک جدول پارتیشن جدید از ابتدا یا از یک اسکریپت سازگار با sfdisk ایجاد کنید.
- **سایر دستورات خط فرمان :** در حین استفاده از **cfdisk** می توانید از دستورات ساده استفاده کنید دقیقاً مانند آنچه ما در ویرایشگر viبرای ذخیره ، درج و غیره استفاده می کنیم. لیست دستورات به شرح زیر است:

- **b** پرچم قابل بوت پارتیشن فعلی را تغییر دهید .به کاربر این امکان را می دهد که کدام قسمت را در درایو قابل بوت اصلی انتخاب کند .برای دیدن نتایج کافیست **b را** فشار دهید ، نیازی به استفاده از **ctrl نیست**.
- **d** این پارتیشن علامت گذاری شده فعلی را حذف می کند و فضای خالی برای پارتیشن جدید ایجاد میکند.
- **h** صفحه راهنما را چاپ می کند ، دستورات استفاده شده را نشان می دهد.
- **n** یک پارتیشن جدید از فضای خالی مشخص شده ایجاد می کند.
- **q** بدون نوشتن پارتیشن روی جدول ، برنامه را ترک می کند.
- **s** اگر پارتیشن ها در آرایه مناسب قرار بگیرند ، ترتیب آنها را برطرف می کنند.
- **t** به شما امکان می دهد نوع پارتیشن را تغییر دهید ، به شما امکان می دهد از لیست انتخاب کنید.
- **u** طرح دیسک را در یک نام فایل اسکریپت مشخص ریخته می شود.
- **W** به کاربر اجازه می دهد تا داده ها را بر روی دیسک بنویسد .با استفاده از ورودی ساده "بله" یا "نه" از کاربر سbe ال می شود که می خواهد بنویسد یا نه.
- **x** به کاربر اجازه می دهد تا تمام اطلاعات اضافی پارتیشن را پنهان یا نمایش دهد.
- **Up-Arrow** به کاربر اجازه می دهد تا مکان نما را به پارتیشن قبلی منتقل کند ، مانند حرکت به بالا در لیست جدول داده شده.
- **Down-Arrow** این گزینه به کاربر امکان می دهد تا مکان نما را به پارتیشن بعدی ، پارتیشن بعدی منتقل کند زیرا هر پارتیشن جدید بعد از پارتیشن قبلی قرار می گیرد.
- **Arrow-Arrow** این گزینه به کاربر اجازه می دهد تا آیتم قبلی منو را وارد کند.
- **Right-Arrow** این گزینه به کاربر اجازه می دهد تا وارد مورد منوی بعدی شود.



دستور :xfs_repair تعمیر فایل سیستمXFS

شرح دستور:

دستور xfs_repairیک ابزار خط فرمان است که برای بررسی و تعمیر فایل سیستم‌های XFS استفاده می‌شود .XFS یک نوع فایل سیستم journaling است که به دلیل کارایی بالا و پشتیبانی از حجم‌های بزرگ داده، به طور گسترده در سیستم‌های لینوکس استفاده می‌شود .هنگامی که فایل سیستم XFS به دلیل خرابی سخت‌افزار، خطاهای نرم‌افزاری یا قطع ناگهانی برق آسیب می‌بیند، xfs_repairمی‌تواند برای تشخیص و رفع مشکلات و بازیابی داده‌ها به کار رود.

ساختار دستور:

```
xfs_repair [options] device
```

- **options:**گزینه‌های مختلفی که رفتار دستور را تغییر می‌دهند.
- **device:**دستگاه بلوکی حاوی فایل سیستم XFS که می‌خواهید بررسی یا تعمیر کنید.

کاربردهای دستور:

- **بررسی یکپارچگی فایل سیستم :**دستور xfs_repairمی‌تواند فایل سیستم را برای یافتن هرگونه خطا یا ناسازگاری بررسی کند.
- **تعمیر فایل سیستم :**در صورتی که خطایی در فایل سیستم وجود داشته باشد، xfs_repairمی‌تواند تلاش کند تا آن را تعمیر کند.
- **بازیابی داده‌ها :**در برخی موارد، xfs_repairمی‌تواند داده‌های از دست رفته را بازیابی کند.

گزینه‌های مهم دستور:

- **n:-**تنها بررسی می‌کند و هیچ تغییری در فایل سیستم ایجاد نمی‌کند.
- **L:-**سطح خطا را تعیین می‌کند.
- **v:-**خروجی مفصل‌تری ارائه می‌دهد.
- **l:-**لیستی از همه گزینه‌ها را نمایش می‌دهد.
- **V:-**نسخه دستور را نمایش می‌دهد.

مثال‌های دستور:

- **بررسی یکپارچگی فایل سیستم :/dev/sda1**

```
sudo xfs_repair /dev/sda1
```

- **تعمیر فایل سیستم /dev/sdb2با خروجی مفصل:**

```
sudo xfs_repair -v /dev/sdb2
```

- **بررسی فایل سیستم بدون ایجاد تغییر:**

```
sudo xfs_repair -n /dev/sdc3
```

ملاحظات مهم:

- **بک‌آپ :**همیشه قبل از اجرای `xfs_repair` از داده‌های خود نسخه پشتیبان تهیه کنید.
- **دسترسی روت :**برای اجرای این دستور به دسترسی روت نیاز دارید.
- **خطاهای پیچیده :**در صورت بروز خطاهای پیچیده، ممکن است نیاز به کمک یک متخصص داشته باشید.
- **توقف سرویس‌ها :**قبل از اجرای `xfs_repair`، مطمئن شوید که هیچ فرآیندی از فایل سیستم آسیب دیده استفاده نمی‌کند.

مراحل عمومی تعمیر فایل سیستم XFS:

- **مونت کردن فایل سیستم :**اگر فایل سیستم مونت شده باشد، ابتدا آن را unmount کنید.
- **اجرای دستور :xfs_repair** دستور را با گزینه‌های مناسب اجرا کنید.
- **بررسی نتایج :**خروجی دستور را به دقت بررسی کنید تا مطمئن شوید که مشکل برطرف شده است.
- **مونت کردن مجدد فایل سیستم :**اگر همه چیز به درستی پیش رفته باشد، فایل سیستم را دوباره مونت کنید.

دستور cryptsetup

دستور cryptsetup در سیستم‌عامل‌های Unix/Linux برای ایجاد، مدیریت و استفاده از ابزارهای رمزنگاری (encryption) برای ایجاد و مدیریت encryption (encryptedاستفاده می‌شود. این دستور امکان ایجاد و مدیریت encryptionهای مبتنی بر dm-crypt در لینوکس را فراهم می‌کند.

مثال‌های دستور:

باز کردن یک encryption: این دستور برای باز کردن (unlockیک encryption قبلی با استفاده از رمزعبور اعمال شده استفاده می‌شود.

```
cryptsetup luksFormat /dev/sdb
```

نمایش اطلاعات encryption: این دستور برای نمایش اطلاعات مربوط به یک encryption (مانند وضعیت و نوع رمزنگاری) استفاده می‌شود.

```
cryptsetup luksOpen /dev/sdb my_encrypted_drive
```

تغییر رمزعبور encryption: این دستور برای تغییر رمزعبور یک encryption قبلی استفاده می‌شود.

```
cryptsetup luksDump /dev/sdb
```

بستن encryption: این دستور برای بستن یک encryption با استفاده از رمزعبور اعمال شده استفاده می‌شود.

```
cryptsetup luksChangeKey /dev/sdb
```

با استفاده از دستور cryptsetup و گزینه‌های مختلف آن، می‌توانید encryptionهای رمزنگاری شده ایجاد کرده، مدیریت کنید و از آن‌ها برای محافظت از داده‌های حساس خود در سیستم استفاده نمایید.

```
cryptsetup luksClose my_encrypted_drive
```

VeraCrypt:محافظی قدرتمند برای حریم خصوصی شما

VeraCryptیک نرم‌افزار متن‌باز و رایگان است که برای رمزنگاری درایوها، پارتیشن‌ها و کل دیسک‌ها استفاده می‌شود .این ابزار به شما اجازه می‌دهد تا داده‌های حساس خود را در برابر دسترسی‌های غیرمجاز محافظت کنید .VeraCrypt جایگزینی امن و قدرتمند برای نرم‌افزار TrueCrypt محسوب می‌شود.

ویژگی‌های کلیدی:VeraCrypt

- **رمزنگاری قدرتمند :**از الگوریتم‌های رمزنگاری قوی مانند AES و Twofish استفاده می‌کند.
- **حالت‌های مختلف کار :**امکان ایجاد درایوهای مجازی، رمزنگاری پارتیشن‌ها و رمزنگاری کل دیسک را فراهم می‌کند.
- **پشتیبانی از چندین سیستم‌عامل :**برای ویندوز، لینوکس و macOS قابل استفاده است.
- **پشتیبانی از هارد دیسک‌های خارجی :**امکان رمزنگاری هارد دیسک‌های خارجی را فراهم می‌کند.
- **رابط کاربری ساده :**استفاده از آن نسبتا آسان است.
- **امنیت بالا :**از ویژگی‌های امنیتی پیشرفته‌ای مانند هش کردن رمز عبور، نمک زدن و ساختارهای داده پنهان استفاده می‌کند.

نحوه نصب VeraCrypt در لینوکس

روش نصب با استفاده از بسته‌بندی‌های توزیع:

اکثر توزیع‌های لینوکس، VeraCryptرا در مخازن نرم‌افزاری خود دارند.

```
https://www.veracrypt.fr/en/Downloads.html
```

دستور quota

کار این دستور، استفاده از دیسک و محدودیت های نمایش سهمیه .نصب و راه اندازی،

ساختاردستور :

```
$  quota [ -u [ User ] ] [ -g [ Group ] ] [ -v | -q ]
```

مثال های دستور :

نمایش اطلاعات کاربران با حجم استفاده شده و مجاز:

```
$quota -u
```

در این مثال، با استفاده از گزینه u-، اطلاعات مربوط به فضای دیسک و سهم کاربران از آن نمایش داده می‌شود. این اطلاعات شامل حجم استفاده شده و مجاز برای هر کاربر است.

نمایش اطلاعات یک کاربر خاص:

```
$quota -u username
```

در این مثال، با استفاده از گزینه u-و پس از آن نام کاربری مورد نظر را مشخص می‌کنیم. این کار باعث نمایش اطلاعات مربوط به فضای دیسک و سهم کاربر مشخص شده می‌شود.

نمایش اطلاعات یک گروه خاص:

```
$quota -g groupname
```

در این مثال، با استفاده از گزینه g-و پس از آن نام گروه مورد نظر را مشخص می‌کنیم. این کار باعث نمایش اطلاعات مربوط به فضای دیسک و سهم گروه مشخص شده می‌شود.

نمایش اطلاعات به صورت جزئی:(verbose)

```
$quota -v
```

در این مثال، با استفاده از گزینه v-، اطلاعات مربوط به فضای دیسک و سهم کاربران و گروه‌ها به صورت جزئی و با جزئیات بیشتر نمایش داده می‌شود.

نمایش اطلاعات به صورت ساکت:(quiet)

```
$quota -q
```

در این مثال، با استفاده از گزینه q-، اطلاعات مربوط به فضای دیسک و سهم کاربران و گروه‌ها به صورت ساکت و بدون نمایش جزئیات نمایش داده می‌شود.

نحوه پاک کردن Swap Space

اگر میخواهید فاصله Swap را پاک کنید، دستور زیر را اجرا کنید.

```
# swapoff -a && swapon -a
```

دستور mkswap

شرح دستور

دستور mkswapدر سیستم‌های لینوکس برای ایجاد یک پارتیشن یا فایل swap استفاده می‌شود .پارتیشن یا فایل swap یک فضای ذخیره‌سازی مجازی است که سیستم عامل از آن برای ذخیره داده‌های موقت استفاده می‌کند، به خصوص زمانی که حافظه اصلی (RAM)پر شده است .این فرایند به عنوان swap یا paging شناخته می‌شود.

ساختار دستور:

```
sudo mkswap <device_or_file>
```

- **sudo:** برای اجرای دستور با دسترسی روت (superuser) استفاده می‌شود.
- **mkswap:** نام خود دستور است.
- **<device_or_file>:** پارتیشن یا فایلی که می‌خواهید به عنوان swap استفاده کنید .مثلاً `/dev/sdb1` برای یک پارتیشن یا `/swapfile` برای یک فایل.

کاربردهای دستور:

- **ایجاد پارتیشن swap:** زمانی که یک پارتیشن جداگانه برای swap در نظر گرفته شده باشد، از این دستور برای فرمت کردن آن به عنوان swap استفاده می‌شود.
- **ایجاد فایل swap:** اگر پارتیشن جداگانه‌ای برای swap وجود نداشته باشد، می‌توان یک فایل ایجاد کرده و آن را به عنوان swap فرمت کرد.
- **تغییر تنظیمات swap:** در برخی موارد، ممکن است نیاز به تغییر تنظیمات یک پارتیشن swap موجود باشد.

گزینه‌های دستور:

- **بدون گزینه:** دستور بدون هیچ گزینه‌ای، پارتیشن یا فایل مشخص شده را به عنوان swap فرمت می‌کند.
- **-v, --verbose:** اطلاعات بیشتری در مورد فرآیند فرمت کردن نمایش می‌دهد.
- **-L <label>, --label=<label>:** یک برچسب برای پارتیشن swap تعیین می‌کند.

مثال‌ها:

- **ایجاد پارتیشن swap بر روی /dev/sdb1:**

```
sudo mkswap /dev/sdb1
```

- **ایجاد فایل swap به نام /swapfile:**

```
sudo mkswap /swapfile
```

- **ایجاد پارتیشن swap با برچسب "myswap":**

```
sudo mkswap -L myswap /dev/sdb1
```

فعال کردن پارتیشن یا فایل swap:

پس از ایجاد پارتیشن یا فایل swap ، برای استفاده از آن باید آن را فعال کنید:

```
sudo swapon <device_or_file>
```

غیرفعال کردن پارتیشن یا فایل swap:

برای غیرفعال کردن یک پارتیشن یا فایل:swap

```
sudo swapoff <device_or_file>
```

اضافه کردن پارتیشن swap به فایل:fstab

برای اینکه پارتیشن swap به طور خودکار در هنگام بوت شدن سیستم فعال شود، باید آن را به فایل /etc/fstab اضافه کنید .
برای مثال:

```
/swapfile swap swap defaults 0 0
```

نکات مهم:

- **اندازه swap:** اندازه swap باید متناسب با میزان رم سیستم و نیازهای برنامه‌ها باشد .معمولاً دو برابر تا چهار برابر اندازه رم توصیه می‌شود.
- **پشتیبانی از swap:** همه سیستم‌های فایل از swap پشتیبانی نمی‌کنند .مطمئن شوید که سیستم فایل پارتیشن یا فایلی که می‌خواهید به عنوان swap استفاده کنید، از swap پشتیبانی می‌کند.
- **ایجاد فایل swap:** برای ایجاد فایلswap ، ابتدا باید یک فایل با اندازه دلخواه ایجاد کنید (مثلاً با استفاده از دستور dd) و سپس آن را با mkswapفرمت کنید.
- **فعال کردن swap در هنگام بوت :**برای فعال کردن خودکار swap در هنگام بوت، باید آن را به فایل /etc/fstab اضافه کنید.

دستور parted

دستور parted برای مدیریت پارتیشن های هارد دیسک در لینوکس استفاده می شود. می توان از آن برای افزودن، حذف، کوچک کردن و گسترش پارتیشن های دیسک به همراه فایل سیستم های واقع بر روی آنها استفاده کرد. برای اجرای دستورات جدا شده به دسترسی ریشه به سیستم نیاز دارید . توجه: Parted تغییرات را فوراً روی دیسک شما می‌نویسد، هنگام تغییر پارتیشن‌های دیسک مراقب باشید.

مثال های دستور :

۱.طرح پارتیشن تمام دستگاه های بلوک را نمایش می‌دهد:

```
$ sudo parted -l
```

۲. نمایش جدول پارتیشن یک دیسک خاص

```
$ sudo parted disk print
```

نمونه هایی از دیسک عبارتند از /dev/sda، dev/sdb/

۳. یک برچسب دیسک جدید از نوع برچسب برای یک دیسک خاص ایجاد میکند.

```
$ sudo parted mklabel disk label-type
```

label-type می تواند مقادیر "aix"، "amiga"، "bsd"، "dvh"، "gpt"، "loop"، "mac"، "msdos"، "pc98"، یا "sun" را بگیرد.

۴. یک پارتیشن جدید در یک دیسک خاص ایجاد میکند، سیستم فایل از نوع fs و اندازه Mb است.

```
$ sudo parted disk mkpart part-time fs-type 1 size
```

fs-type اختیاری است. می تواند مقادیر «btrfs»، «ext2»، «ext3»، «ext4»، «fat16»، «fat32»، «hfs»، «hfs+»، «linux-swap»، «ntfs»، «reiserfs» را بگیرد. "udf" یا اندازه "xfs" باید کمتر از اندازه کل دیسک مشخص شده باشد. برای ایجاد یک پارتیشن با اندازه ۵۰ مگابایت، مقدار ۵۰ خواهد بود.

۵. parted همچنین می تواند در قالب تعاملی اجرا شود. عملیات مدیریت پارتیشن های دیسک را می توان با وارد کردن دستورات مناسب در جلسه تعاملی انجام داد. دستور help در جلسه تعاملی فهرستی از تمام عملیات مدیریت دیسک ممکن را نشان می‌دهد که می توان انجام داد.

```
$ sudo parted
```

فرم های ساختار دستور:

```
$ parted [options] [device [command [options...]...]]
```

دستور Baobab

یک ابزار گرافیکی برای تجزیه و تحلیل فضای استفاده شده از دیسک است.

شرح دستور:

baobab قادر است پوشه های خاص یا کل سیستم فایل (محلی و راه دور) را اسکن کند تا به کاربر یک نمایش درختی گرافیکی شامل اندازه هر فهرست یا درصد در شاخه ارائه دهد. همچنین هر گونه تغییری را که در فهرست اصلی شما تا جایی که هر دستگاه نصب شده/بدون نصب شده است، در زمان واقعی شناسایی میکند. یک پنجره درختی گرافیکی نیز برای هر پوشه انتخابی ارائه میکند.

روش نصب

- debian:

```
$ sudo apt install baobab
```

- Fedora:

```
sudo dnf install baobab
```

- CentOS:

```
sudo yum install baobab
```

- Debian:

```
sudo apt-get install baobab
```

- openSUSE:

```
sudo zypper install baobab
```

دستور ddrescue

یک ابزار قدرتمند برای بازیابی داده ها است. این دستور داده ها را از یک فایل یا دستگاه به یک فایل دیگر کپی می کند، در حالی که سعی می کند خطاهای خواندن را حل کند. در زیر برخی از گزینه های این دستور و مثال هایی از چگونگی استفاده از آنها آورده شده است:

گزینه های دستور:

نمایش راهنمایی و خروج

```
ddrescue --help
```

خروجی اطلاعات نسخه و خروج

```
ddrescue --version
```

حداقل نرخ خواندن مناطق خوب به بایت/ثانیه

```
ddrescue --min-read-rate=1000 inputfile outputfile logfile
```

علامت گذاری بلوک های غیر تقسیم شده، غیر تراشیده به عنوان غیر تلاش شده

```
ddrescue --try-again inputfile outputfile logfile
```

اندازه بلوک دستگاه ورودی [پیش فرض ۵۱۲]

```
ddrescue --block-size=1024 inputfile outputfile logfile
```

نمایش ضرایب دودویی در اعداد [SI]

```
ddrescue --binary-prefixes inputfile outputfile logfile
```

بخش هایی که باید در یک زمان کپی شوند [پیش فرض ۱۲۸]

```
ddrescue --cluster-size=256 inputfile outputfile logfile
```

خواندن داده های جدید فراتر از محدوده های logfile را انجام ندهید

```
ddrescue --complete-only inputfile outputfile logfile
```

استفاده از دسترسی مستقیم به دیسک برای فایل ورودی

```
ddrescue --direct inputfile outputfile logfile
```

استفاده از نوشتن همزمان برای فایل خروجی

```
ddrescue --synchronous inputfile outputfile logfile
```

حداکثر تعداد مناطق خطا [جدید] مجاز

```
ddrescue --max-errors=10 inputfile outputfile logfile
```

حداکثر نرخ مجاز خطاهای خواندن در هر ثانیه

```
ddrescue --max-error-rate=500 inputfile outputfile logfile
```

بازنویسی دستگاه یا پارتیشن خروجی

```
ddrescue --force inputfile outputfile logfile
```

پر کردن بلوک های نوع داده شده با داده های infile (?*/-+)

```
ddrescue --fill=? inputfile outputfile logfile
```

تولید logfile تقریبی از کپی جزئی

```
ddrescue --generate-logfile inputfile outputfile logfile
```

موقعیت شروع در فایل ورودی [۰]

```
ddrescue --input-position=500 inputfile outputfile logfile
```

تأیید اندازه فایل ورودی با اندازه در logfile

```
ddrescue --verify-input-size inputfile outputfile logfile
```

اندازه اولیه برای جستجو در خطای خواندن [۶۴ KiB]

```
ddrescue --skip-size=1024 inputfile outputfile logfile
```

محدود کردن دامنه به بلوک های تمام شده در فایل

```
ddrescue --domain-logfile=domainlogfile inputfile outputfile logfile
```

علامت گذاری همه بلوک های ناموفق به عنوان غیر تراشیده

```
ddrescue --retrim inputfile outputfile logfile
```

تلاش نکنید برای تقسیم یا تلاش مجدد برای بلوک های ناموفق

```
ddrescue --no-split inputfile outputfile logfile
```

موقعیت شروع در فایل خروجی [ipos]

```
ddrescue --output-position=500 inputfile outputfile logfile
```

پیش تخصیص فضا روی دیسک برای فایل خروجی

```
ddrescue --preallocate inputfile outputfile logfile
```

عدم نمایش همه پیام ها

```
ddrescue --quiet inputfile outputfile logfile
```

خروج بعد از تلاش های داده شده (-۱=بی نهایت) [۰]

```
ddrescue --max-retries=5 inputfile outputfile logfile
```

معکوس کردن جهت عملیات کپی

```
ddrescue --reverse inputfile outputfile logfile
```

حداکثر اندازه داده های ورودی برای کپی شدن

```
ddrescue --max-size=1000 inputfile outputfile logfile
```

استفاده از نوشتن های خالی برای فایل خروجی

```
ddrescue --sparse inputfile outputfile logfile
```

کوتاه کردن فایل خروجی به اندازه صفر

```
ddrescue --truncate inputfile outputfile logfile
```

حداکثر زمان از آخرین خواندن موفق

```
ddrescue --timeout=5s inputfile outputfile logfile
```

باشید مفصل (یک nd -v۲ بیشتر می دهد)

```
ddrescue --verbose inputfile outputfile logfile
```

گسترش اندازه فایل خروجی برای حداقل این طول

```
ddrescue --extend-outfile=1000 inputfile outputfile logfile
```

فصل دهم

فصل دهم

دریافت اطلاعات سخت افزاری در linux

دستور Dmidecode

چگونه می‌توان از دستورDmidecode برای بازیابی اطلاعات سخت افزاری هر linux system استفاده کرد.فرض کنید ما می خواهیم اطلاعات یک system را که نیاز داریم جمع آوری کنیم.مانند memory ، بایوس[1] و پردازنده و غیره. با استفاده از دستور Dmidecode، ما می خواهیم جزئیات را بدون باز کردن system تشخیص دهیم.دستور Dmidecodeبرای RHEL / CentOS / Fedora / Ubuntu Linuxکار می کند.



به ابزار DmidecodeDMIبرخی می گوید (SMBIOS) که جدولی، برای جمع آوری داده ها و نمایش اطلاعات مفید system مانند اطلاعات سخت افزاری ، شماره سریال و نسخه BIOS، پردازنده و غیره در فرمت قابل خواندن برای انسان است. می توانید از دستور root برای اجرای دستور dmidecode استفاده کنید.

در زیر نمونه خروجی دستور دستور Demidecode است.

```
# dmidecode
```

خروجی دستور:

```
اصلاح نسخه.(SMBIOS (2.31 -> 2.3
SMBIOS 2.3 موجود است
45 سازه اشغال ۱۶۴۲ بایت.
جدول در x000E0010.۰
دسته x0000۰، نوع DMI 0 ، ۲۰بایت
اطلاعاتBIOS
فروشنده: فونیکس فن آوری های محدود
نسخه: ۶,۰۰
تاریخ انتشار: ۲۰۰۶/۰۶/۱۲
آدرس: xE78A0۰
```

[1] bios

```
حجم اجرا: ۱۰۰۱۹۲ بایت
ROM حجم: ۶۴ کیلوبایت
مشخصات:
ISA پشتیبانی می شود
PCI پشتیبانی می شود
PC Card (PCMCIA) پشتیبانی می شود
PNP پشتیبانی می شود
APM پشتیبانی می شود
بایوس قابل ارتقا است
سایه BIOS مجاز است
پشتیبانی ESCD در دسترس است
میراث USB پشتیبانی می شود
باتری هوشمند پشتیبانی می شود
مشخصات بوت BIOS پشتیبانی می شود
```

دریافت انواع DMI

اطلاعات سخت افزاری خاص system ارائه می دهد. Dmidecodeبا گزینه های ' t- ' یا ' type- ' و ' Id ' به ما اطلاعات دقیق ارائه می دهد. id اطلاعات ماژول memory را به ما نشان می دهد.

```
$ sudo dmidecode -t 6
```

```
اصلاح نسخه.(2.3 <- 2.31) SMBIOS
SMBIOS 2.3 موجود است
دسته ۰x0009، نوع DMI 6 ، ۱۲بایت
اطلاعات ماژول memory
تعیین سوکت: سوکت RAM ره ۰
اتصالات بانکی: ۱ ۰
سرعت فعلی: ناشناخته است
نوع EDO DIMM :
اندازه install شده: ۱۰۲۴ مگابایت (اتصال تک بانکی)
```

در زیر جزئیات نوع DMIوجود دارد. برای مثال ، برای به دست آوردن اطلاعات کش[۱] در system، می توانید دستور زیر را اجرا کنید.

```
$ sudo dmidecode -t cache
# dmidecode 2.11
```

دریافت اطلاعات memory

چگونه اطلاعات memory در system را دریافت کنیم وسیستم چقدر memory پشتیبانی می‌کند؟دستور زیر نشان می دهد که system می تواند حداکثر 4 گیگابایت RAMرا پشتیبانی کند.

```
$ sudo dmidecode -t 16
```

[۱] cache memory

دریافت اطلاعات BIOS

برای دریافت اطلاعات BIOSsystem، دستور زیر را با گزینه '-t' اجرا کنید.

```
$ sudo dmidecode -t BIOS
```

```
PNP پشتیبانی می شود
APM پشتیبانی می شود
بایوس قابل ارتقا است
سایه BIOS مجاز است
پشتیبانی ESCD در دسترس است
میراث USB پشتیبانی می شود
باتری هوشمند پشتیبانی می شود
مشخصات بوت BIOS پشتیبانی می شود
.5چگونه تولید کننده، مدل و ره سریال را دریافت کنم؟
```

برای دریافت اطلاعات مربوط به سازنده ، مدل و شماره سریال system، از دستور زیر استفاده کنید، همانطور که در زیر نشان داده شده است.

```
$ sudo dmidecode -t system
```

همیشه یک تجربه خوب برای دانستن اجزای سخت افزاری linux system در حال اجرا ، این کمک می کند تا با مشکلات سازگاری در هنگام نصب بسته ها، درایورهای system خود، مقابله کنید. بنابراین در این نکات و ترفندها، ما باید به برخی از دستورات مفید نگاه کنیم که می تواند به در استخراج اطلاعات در مورد linux system عامل و اجزای سخت افزاری کمک کند.

بدست آوردن اطلاعات سیستم

برای شناختن نام system فقط می توانید از دستور unameاستفاده کنید .

```
hosein @ hosein ~ $ uname
```

برای مشاهده نام host شبکه خود، از کلید "-n" با دستور uname استفاده کنید، همانطور که نشان داده شده است.

```
hosein @ hosein ~ $ uname -n
```

برای دریافت اطلاعات مربوط به هسته[1] نسخه، از کلید '-v' استفاده کنید.

```
hosein @ hosein ~ $ uname -v
# 64-ubuntu SMP Mon Sep 22 21:28:38 UTC 2014
```

برای دریافت اطلاعات در مورد انتشار هسته خود، از کلید '-r' استفاده کنید.

```
hosein @ hosein ~ $ uname -r
```

```
3.13.0-37 public
```

برای چاپ نام سخت افزار دستگاه خود، از کلید "-m" استفاده کنید:

```
hosein @ hosein ~ $ uname-m
```

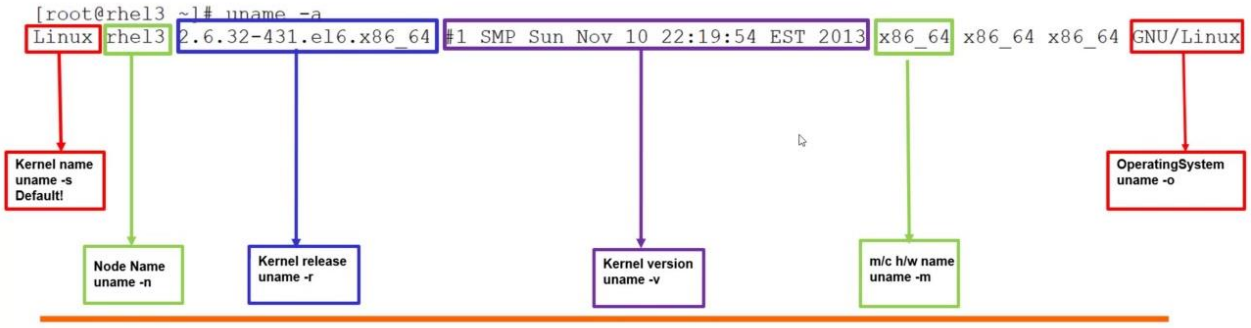
[1] kernel

x86_64

تمام این اطلاعات را می توان در یک بار با اجرای دستور uname -aبه صورت زیر نشان داد.

```
hosein @ hosein ~ $ uname -a
linux hosein .com 3.13.0-37- public ٦٤ # -ubuntu SMP monday ٢٢ november٢١:٢٨:٣٨ UTC
2014 x86_64 x86_64 x86_64 gnu / linux
```

به صورت کلی



بدست آوردن اطلاعات سخت افزاری

در اینجا می‌توانید از ابزارlshw برای جمع آوری اطلاعات وسیع در مورد اجزای سخت افزاری خود مانند پردازنده ، دیسک ، memory ، کنترل کننده هایUSB و غیره استفاده کنید. lshw یک ابزار نسبتا کوچک است و چندین گزینه وجود دارد که می توانید با آن در هنگام استخراج اطلاعات استفاده کنید.اطلاعات ارائه شده توسط lshw جمع آوری شده از فایل های مختلف proc /تشکیل شده است.

توجه داشته باشید :فراموش نکنید که دستور lshwتوسط (root) superuser یا sudoکاربر اجرا می شود. می توانید خلاصه ای از اطلاعات سخت افزار خود را با استفاده از گزینه short-چاپ کنید.

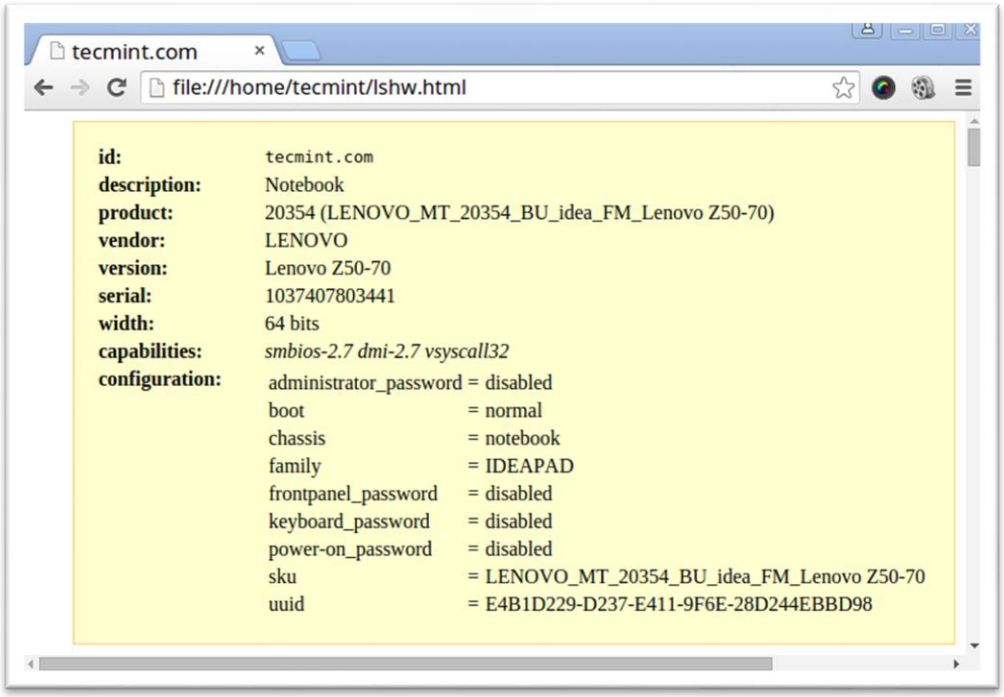
```
hosein @ hosein ~sudo lshw -short
```

بدست آوردن کلاس دستگاهH / W

```
system٢٠٣٥۴ (LENOVO_MT_20354_BU_idea_FM_Lenovo Z50-70)
/ 0 bus Lancer 5A5
/ 0/0 memory١٢٨ KiB BIOS
/ 0/4 پردازنده اینتل (R) هسته (TM) i5-4210U CPU @ 1.70GHz
/ 0/4 / b memory٣٢ KiB L1 memory cache
/ 0/4 / c memory٢٥۶ KiB L2 memory
/ 0/4 / d memory٣ MiB L3 memory cache
/ 0 / memory memory٣٢ KiB L1
/ 0/12 memory٨ GiB memory system
/ 0/12/0 memory DIMM [empty]
/ 0/12/1 memory DIMM [empty]
/ 0/12/2 memory٨ GiB SODIMM DDR3 ٠٫۶) همزمان ١۶٠٠ مگاهرتز ns)
/ 0/12/3 memory DIMM [empty]
/ 0/100 کنترل پلیر Haswell-ULT DRAM
/ 0/100/2 نمایشگرVGAمجتمع Haswell-ULT
/ 0/100/3 چند رسانه ای کنترل Haswell-ULT HD صوتی
```

اگر می خواهید خروجی را به عنوان یک فایل HTML ایجاد کنید، می توانید از گزینه html- استفاده کنید.

```
hosein @ hosein ~ $ sudo lshw -html> lshw.html
```



بدست آوردن اطلاعات CPU

برای مشاهده اطلاعات مربوط به CPU خود، از دستور lscpu استفاده کنید چون اطلاعات مربوط به architect CPU مانند تعداد CPU ها، هسته ها، مدل CPU family، memory های CPU ، موضوعات و غیره را از sysfs و proc / cpuinfo/ نشان می دهد.

```
hosein @ hosein ~ $ lscpu
```

```
architect: x86_64
bit ٦٤ ،protocolCPU: 32 bit
Order Byte: Little Endian
پردازنده (ها): ٤
```

بدست آوردن اطلاعات مربوط به بلوک ها

دستگاه های بلوکی دستگاه های ذخیره سازی مانند هارد دیسک، درایوهای فلش و غیره است .دستورlsblk برای گزارش اطلاعات مربوط به دستگاه های بلوک به شرح زیر استفاده می شود.

```
hosein @ hosein ~ $ lsblk
```

```
NAME MAJ: MIN MIN SIZE type RO MOUNTPOINT
sda 8: 0 0 931.5G 0
├─sda1 8: 1 0 1000M 0
├─sda2 8: 2 0 260M 0 part / boot / efi
├─sda3 8: 3 0 1000M 0
├─sda4 8: 4 0 128M 0
```

```
├─sda5 8: 5 0 557.1G 0
├─sda6 8: 6 0 25G 0
├─sda7 8: 7 0 14.7G 0
├─sda8 8: 8 0 1M 0
├─sda9 8: 9 0 324.5G 0   /
└─sda10 8:10 0 7.9G 0   [SWAP]
sr0  0 1 1024M 0 rom
```

اگر می خواهید تمام دستگاه های بلوک را در system خود مشاهده کنید ، گزینه a-را وارد کنید.

```
hosein @ hosein  ~ $ lsblk -a
```

```
NAME MAJ: MIN MIN SIZE نوع RO MOUNTPOINT
sda 8: 0 0 931.5G 0
├─sda1 8: 1 0 1000M 0
├─sda2 8: 2 0 260M 0 part / boot / efi
├─sda3 8: 3 0 1000M 0
├─sda4 8: 4 0 128M 0
├─sda5 8: 5 0 557.1G 0
├─sda6 8: 6 0 25G 0
├─sda7 8: 7 0 14.7G 0
├─sda8 8: 8 0 1M 0
├─sda9 8: 9 0 324.5G 0   /
```

بدست آوردن اطلاعات پردازنده

دریافت اطلاعات CPU با استفاده از دستورcat

می توانید به سادگی اطلاعات مربوط بهCPU system خود را با مشاهده محتویات فایل proc / cpuinfo /با کمک دستورcat به صورت زیر مشاهده کنید:

```
$  cat / proc / cpuinfo
```

هسته ها را نشان می دهد

```
$  cat / proc / cpuinfo | grep 'core id' #
```

دستور lscpuاطلاعات architect پردازنده را از sysfsو proc / cpuinfo /چاپ می کند همانطور که در زیر نشان داده شده است:

```
$  lscpu
```

معماری پردازنده[1] را نشان میدهد

```
cpuid
```

دستور cpuidاطلاعات کاملی از CPU جمع آوری شده از دستور CPUID را دریافت می‌کند و همچنین مدل دقیق پردازنده های x86از آن اطلاعات را نمایان می کند. اطمینان حاصل کنید قبل از اجرا آن را نصب کنید.

```
$  sudo apt   install cpuid
```

[1] cpu

```
$ sudo yum install cpuid
$ sudo dnf install cpuid
```

پس از install، cpuidرا اجرا کنید تا اطلاعات مربوط به پردازنده x86 را جمع آوری کنید.

```
$ cpuid
```

دستور cpufreq-info

دستور برای بازیابی اطلاعات هسته cpufreq

ساختار دستور:

```
$ cpufreq-info [options]
```

شرح دستور:

ابزار کوچکی که اطلاعات cpufreq را چاپ می‌کند که برای توسعه دهندگان و کاربران علاقه مند مفید است. دستور cpufreq-info یک ابزار در سیستم‌عامل‌های یونیکس است که اطلاعات مربوط به تنظیمات و ویژگی‌های فرکانس پردازنده را نمایش می‌دهد. این دستور معمولاً برای بررسی قابلیت‌های مدیریت فرکانس پردازنده در سیستم استفاده می‌شود. برای استفاده از دستور cpufreq-info، می‌توانید به صورت زیر عمل کنید:

```
$cpufreq-info
```

این دستور اطلاعاتی از جمله مدل و سرعت پردازنده، حداقل و حداکثر فرکانس قابل پشتیبانی، وضعیت فعلی فرکانس و سیاست مدیریت فرکانس را نمایش می‌دهد. برای نصب cpufreq-info در سیستم‌عامل‌های مبتنی بر یونیکس مانند لینوکس، معمولاً می‌توان از مدیر بسته‌ها (package manager) استفاده کرد. اگر سیستم عامل شما دارای مدیر بسته‌های مانند APT برای توزیع‌های مبتنی بر دبیان مانند اوبونتو( ) YUMبرای توزیع‌های مبتنی بر ردهت( ) Pacmanبرای توزیع‌های مبتنی بر آرچ لینوکس) یا Zypper)برای توزیع‌های مبتنی بر سوزه) است، می‌توانید از طریق خط فرمان دستورات زیر را اجرا کنید:

برای توزیع‌های مبتنی بر دبیان مانند اوبونتو:

```
$sudo apt-get update
$sudo apt-get install cpufrequtils
```

برای توزیع‌های مبتنی بر ردهت:

```
$sudo yum update
$sudo yum install cpufrequtils
```

برای توزیع‌های مبتنی بر آرچ لینوکس:

```
$sudo pacman -Sy cpufrequtils
```

برای توزیع‌های مبتنی بر سوزه:

```
$sudo zypper update
$sudo zypper install cpufrequtils
```

اگر سیستم عامل شما از مدیر بسته‌های متفاوتی استفاده می‌کند، می‌توانید نحوه نصب cpufreq-infoرا در منبع مرجع سیستم عامل خود جستجو کنید یا از راهنمای مربوطه استفاده کنید. لطفاً توجه داشته باشید که برای نصب برخی بسته‌ها، به دسترسی مدیر سیستم (sudo)نیاز است و بسته به تنظیمات سیستم شما، این دسترسی ممکن است مورد نیاز باشد.

گزینه های دستور:

```
$ cpufrequtils -c
$ cpufrequtils --cpu <CPU>
```

شماره <CPU> که اطلاعات باید در مورد آن نیازدارید، وارد نمایید.

```
$ cpufrequtils -e
$ cpufrequtils --debug
```

اطلاعات در حالت اشکال زدایی را چاپ می‌کند.

```
$ cpufrequtils –f
$ cpufrequtils --freq
```

با توجه به هسته ، فرکانسی را که CPU در حال حاضر روی آن کار می‌کند، دریافت کنید.

```
$ cpufrequtils -w
$ cpufrequtils --hwfreq
```

با کمک خواندن آن از روی سخت افزار (فقط برای روت در دسترس است)، فرکانسی را که CPU در حال حاضر در آن کار می‌کند، دریافت کنید.

```
$ cpufrequtils –l
$ cpufrequtils --hwlimits
```

حداقل و حداکثر فرکانس مجاز CPU را تعیین میکند.

```
$ cpufrequtils -d
$ cpufrequtils --driver
```

درایور کرنل cpufreq مورد استفاده را تعیین می‌کند.

```
$ cpufrequtils -p
$ cpufrequtils --policy
```

خط مشی cpufreq مورد استفاده فعلی را دریافت می‌کند.

```
$ cpufrequtils -g
$ cpufrequtils --governors
```

گاورنرهای cpufreq موجود را تعیین می‌کند.

```
$ cpufrequtils -a
$ cpufrequtils --related-cpus
```

تعیین می‌کند کدام CPU با فرکانس سخت افزاری مشابهی کار می‌کند.

```
$ cpufrequtils -a
$ cpufrequtils --afected-cpus
```

تعیین می‌کند که کدام CPU باید فرکانس آنها توسط نرم افزار هماهنگ شود.

```
$ cpufrequtils –s
$ cpufrequtils --stats
```

در صورت وجود، آمار cpufreq را نشان می‌دهد.

```
$ cpufrequtils -y
$ cpufrequtils --latency
```

حداکثر تأخیر در تغییرات فرکانس CPU را تعیین می‌کند.

```
$ cpufrequtils –o
$ cpufrequtils --proc
```

اطلاعاتی را که توسط رابط /proc/cpufreq/ در نسخه ۲,۴ ارائه شده است چاپ می‌کند

```
$ cpufrequtils –m
$ cpufrequtils --human
```

خروجی قابل خواندن توسط انسان برای پارامترهای f-، w-، s- و y-.

دستور lscpu

lscpu در لینوکس/یونیکس برای نمایش اطلاعات معماری CPU استفاده می شود. lscpu اطلاعات معماری CPU را از فایل های sysfs و /proc/cpuinfo/ جمع آوری می‌کند.

ساختاردستور

```
$ lscpu [-a|-b|-c] [-x] [-s dir] [-e [=list]|-p [=list]]
$ lscpu -h|-V
```

شرح دستور:

lscpu اطلاعات معماری CPU را از sysfs و /proc/cpuinfo/ جمع آوری می‌کند. خروجی فرمان را می توان برای تجزیه و تحلیل و یا برای خوانایی آسان تر برای کاربرتنظیم کرد.برای **مثال**، این اطلاعات شامل تعداد پردازنده‌ها، رشته‌ها، هسته‌ها، سوکت‌ها و گره‌های دسترسی به حافظه غیریکنواخت (NUMA) می‌شود. همچنین اطلاعاتی در مورد حافظه پنهان CPU و اشتراک گذاری حافظه پنهان، خانواده، مدل، bogoMIPS، ترتیب بایت و پله وجود دارد.

ستون های خروجی دستور:

- CPU
- شماره CPU منطقی یک CPU که توسط هسته لینوکس استفاده می شود.
- عدد هسته منطقی یک هسته می تواند شامل چندین CPU باشد.
- شماره سوکت منطقی یک سوکت می تواند شامل چندین هسته باشد.
- شماره کتاب منطقی یک کتاب می تواند شامل چندین سوکت باشد.
- شماره گره منطقی NUMA. یک گره ممکن است حاوی چندین کتاب باشد.
- اطلاعاتی درباره اشتراک کش ها بین CPU ها.
- آدرس فیزیکی یک CPU

برای استفاده از دستور lscpuبا همه گزینه های ذکر شده، می توانید از دستورات زیر استفاده کنید:

نمایش اطلاعات کامل درباره پردازنده:

```
$lscpu
```

نمایش اطلاعات مفصل درباره پردازنده:

```
$lscpu -a
```

نمایش اطلاعات ساختار پردازنده:

```
$lscpu -b
```

نمایش اطلاعات خاصی از پردازنده:

```
$lscpu -c
```

نمایش اطلاعات پردازنده به صورت:XML

```
$lscpu -x
```

ذخیره خروجی دستور lscpuدر یک پوشه خاص:

```
$lscpu -s [مسیر]
```

نمایش اطلاعات خاصی از پردازنده با استفاده از لیست مشخص:

```
$lscpu -e [=لیست]
```

نمایش اطلاعات پردازنده به صورت مرتب شده با استفاده از لیست مشخص:

```
$lscpu -p [=لیست]
```

لطفا توجه داشته باشید که برخی از گزینه ها ممکن است در برخی از سیستم ها و پلتفرم ها موجود نباشند. همچنین برای استفاده از دستور lscpu، شما باید دسترسی لازم را به سیستم مورد نظر داشته باشید.

دستور mpstat

دستور mpstat برای گزارش آمار مربوط به پردازنده استفاده می شود. این به طور دقیق آمار استفاده از CPU از سیستم و اطلاعات مربوط به استفاده و عملکرد CPU را نمایش می‌دهد.

ساختاردستور:

```
$  mpstat [options] [<interval> [<count>]]
```

نکته: اولین پردازنده را با CPU 0، دومی را با CPU 1 و غیره مقداردهی اولیه می‌کند.

گزینه های دستور:

- A – برای نمایش دقیق تمام آمار
- h – برای نمایش راهنمای mpstat
- I – برای نمایش آمار دقیق وقفه ها برای گزارش خلاصه آمار CPU بر اساس گره NUMA
- n – برای نشان دادن گره های NUMA که آمار باید برای آنها باشد
- N – برای نشان دادن پردازنده هایی که آمار باید برای آنها باشد گزارش شده است
- P – گزارش برای نمایش آمار در JSON (جاوا اسکریپت Object Notation)
- T – برای نمایش عناصر توپولوژی در گزارش CPU
- u – برای گزارش استفاده از CPU
- v – برای نمایش آمار استفاده در سطح پردازنده مجازی توضیحات گزینه
- V – برای نمایش نسخه mpstat
- ALL – برای نمایش آمار دقیق در مورد تمام پردازنده ها

مثال های دستور:

۱.برای نمایش آمار پردازنده و CPU:

```
$ mpstat
```

۲. برای نمایش شماره پردازنده همه CPU ها:

```
$ mpstat -P ALL
```

۳. برای دریافت تمام اطلاعاتی که ابزار ممکن است جمع آوری کند:

```
$ mpstat -A
```

۴. برای نمایش استفاده از CPU توسط یک پردازنده خاص:

```
$ mpstat -P 0
```

۵. برای نمایش میزان مصرف CPU با بازه زمانی:

```
$ mpstat 1 5
```

نکته: این دستور ۵ گزارش را با فاصله زمانی ۱ ثانیه چاپ می‌کند.

برای گزینه A-که آمار دقیق در مورد تمام پردازنده‌ها را نمایش می‌دهد:

```
$mpstat -A
```

در این مثال، دستور mpstat با استفاده از گزینه A- آمار دقیق در مورد تمام پردازنده‌ها را نمایش می‌دهد.

برای گزینه h- که راهنمای دستور mpstat را نمایش می‌دهد:

```
$mpstat -h
```

در این مثال، دستور mpstat با استفاده از گزینه h- راهنمای دستور را نمایش می‌دهد.

برای گزینه I- که آمار دقیق وقفه‌ها برای گزارش خلاصه آمار CPU بر اساس گره NUMA را نمایش می‌دهد:

```
$mpstat -I
```

در این مثال، دستور mpstat با استفاده از گزینه I- آمار دقیق وقفه‌ها بر اساس گره NUMA را نمایش می‌دهد.

برای گزینه n- که گره‌های NUMA را نمایش می‌دهد که آمار باید برای آنها باشد:

```
$mpstat -n
```

در این مثال، دستور mpstat با استفاده از گزینه n- گره‌های NUMA را نمایش می‌دهد که آمار باید برای آنها باشد.

برای گزینه N- که پردازنده‌هایی را نمایش می‌دهد که آمار باید برای آنها باشد گزارش شده است:

```
$mpstat -N
```

در این مثال، دستور mpstat با استفاده از گزینه N- پردازنده‌هایی را نمایش می‌دهد که آمار باید برای آنها باشد.

برای گزینه P- که گزارش آمار را در قالب JSON (Object Notation جاوا اسکریپت نمایش می‌دهد:

```
$mpstat -P
```

در این مثال، دستور mpstat با استفاده از گزینه P- گزارش آمار را در قالب JSON نمایش می‌دهد.

برای گزینه T- که عناصر توپولوژی را در گزارش CPU نمایش می‌دهد:

```
$mpstat -T
```

در این مثال، دستور mpstat با استفاده از گزینه T- عناصر توپولوژی را در گزارش CPU نمایش می‌دهد.

برای گزینه u- که گزارش استفاده از CPU را نمایش می‌دهد:

```
$mpstat -u
```

در این مثال، دستور mpstat با استفاده از گزینه u- گزارش استفاده از CPU را نمایش می‌دهد.

برای گزینه v- که آمار استفاده در سطح پردازنده مجازی را نمایش می‌دهد:

```
$mpstat -v
```

در این مثال، دستور mpstat با استفاده از گزینه v- آمار استفاده در سطح پردازنده مجازی را نمایش می‌دهد.

برای گزینه V- که نسخه mpstat را نمایش می‌دهد:

```
$mpstat -V
```

دستور Inxi

Inxi یک اسکریپت اطلاعاتی قدرتمند که در نظر گرفته شده برای کنسول و IRC (Internet Relay Relay) است. می توانید از آن برای بازیابی اطلاعات سخت افزاری استفاده کنید.

می توانید به صورت زیر آن را نصب کنید:

```
$ sudo apt install inxi
$ sudo yum install inxi
$ sudo dnf install inxi
```

برای نشان دادن اطلاعات کامل cpu، از جمله سرعت CPU و سرعت حداکثر CPU از سویچ C– استفاده کنید:

```
$ inxi -C
```

دستور lshw

lshw ابزار کمکی برای جمع آوری اطلاعات عمیق در تنظیمات سخت افزاری کامپیوتر است. می توانید از گزینه C-برای انتخاب کلاس سخت افزار، CPUدر این مورد استفاده کنید:

```
$ sudo lshw -C CPU
```

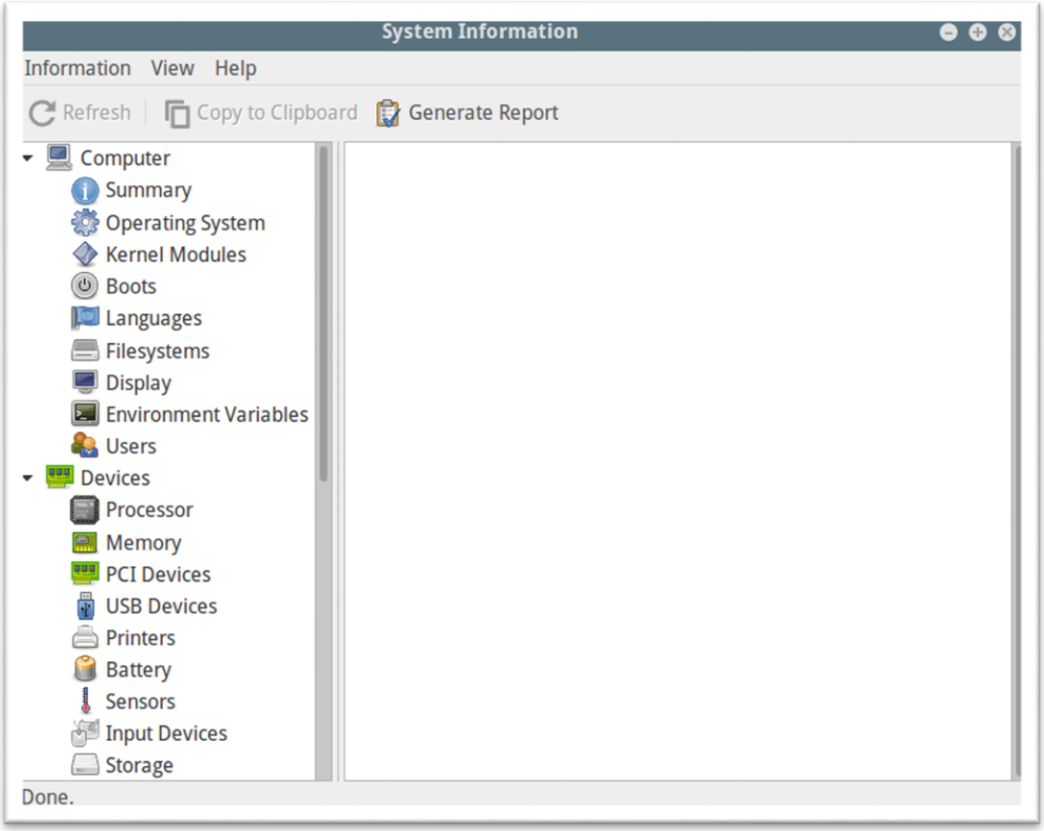
دستور hardinfo

hardinfoاطلاعات سخت افزاری را در یک پنجره GTK نمایش می دهد، می توانید آن را به صورت زیر نصب کنید:

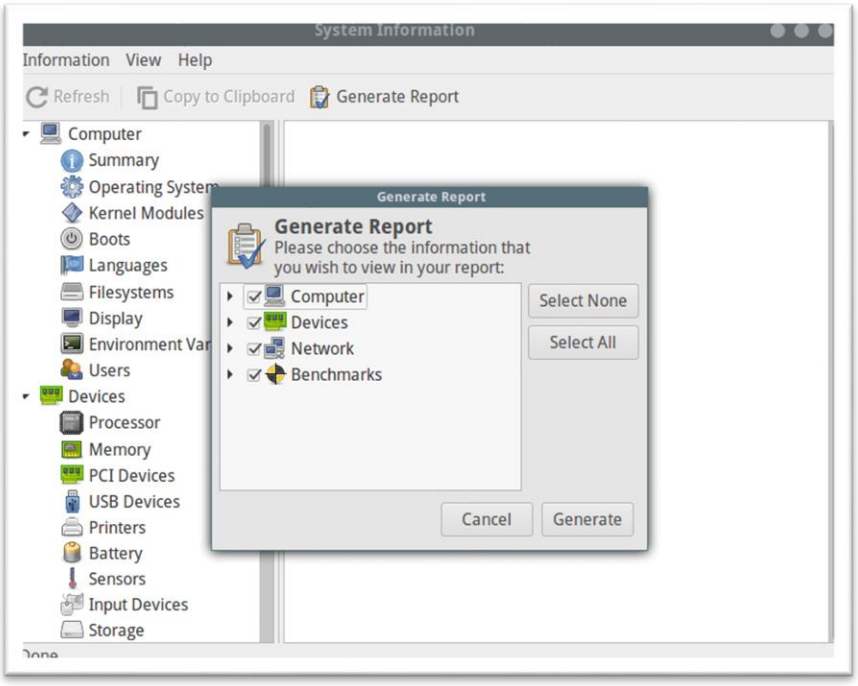
```
$ sudo apt  install hardinfo
$ sudo yum  install hardinfo
$ sudo dnf  install hardinfo
```

هنگامی که آن را نصب کرده اید، تایپ کنید:

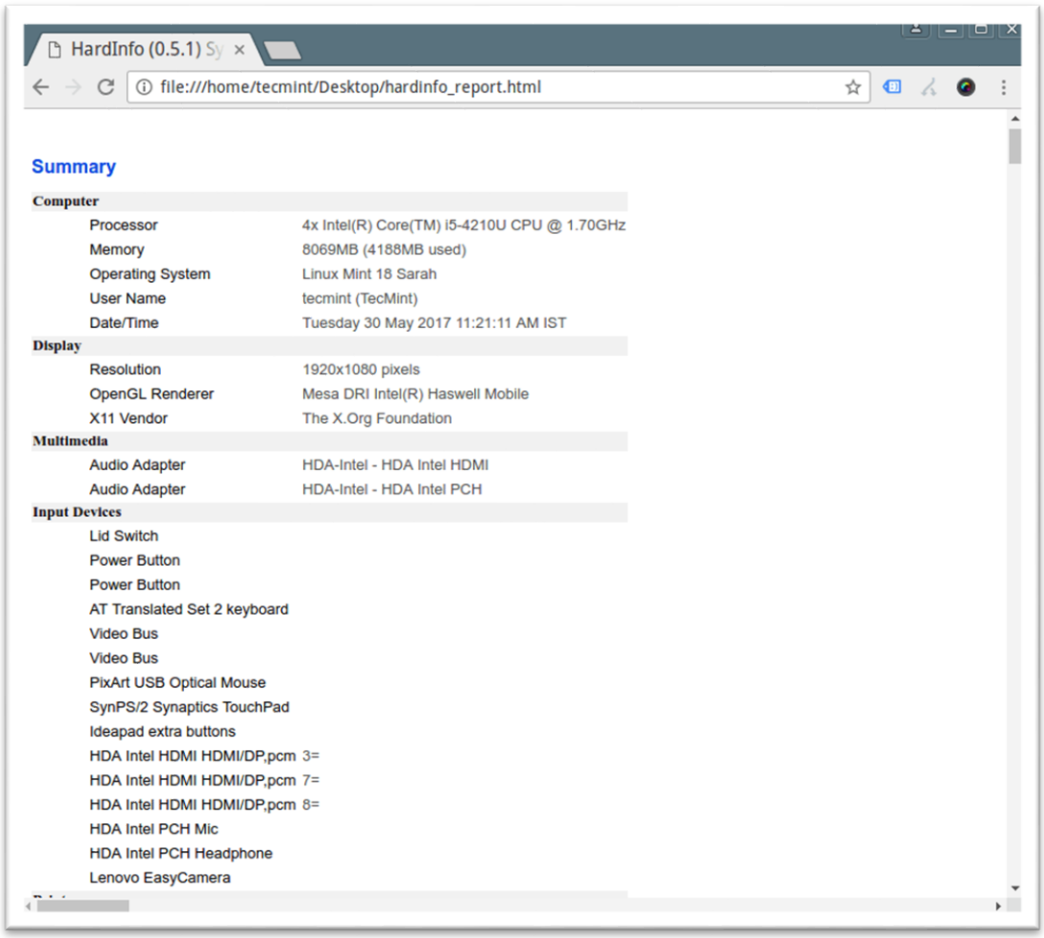
```
$ hardinfo
```



از گزینه " Generate "کلیک کنید تا گزارش ایجاد شود..توجه داشته باشید که می توانید اطلاعات سخت افزاری مورد نیاز را تولید کنید.



گزارش گزارش system را تولید کنید. هنگامی که این گزارش را در قالب HTML ایجاد می کنید، می توانید آن را از یک مرورگر[۱] وب مشاهده کنید که در زیر نشان داده شده است.



[۱] browser

دستور hwinfo

اطلاعات موجود در مورد سخت افزار را نشان می دهدhwinfo. برای استخراج اطلاعات در مورد سخت افزار موجود در یک system linux استفاده می شود.برای نمایش اطلاعات مربوط به CPU خود از cpu استفاده کنید

```
$ hwinfo - cpu
```

دستور lshal

لیست دستگاه های HAL

ساختاردستور:

```
$ lshal [options]
```

lshal ابزاری برای نمایش موارد در پایگاه داده دستگاه HAL است. برای اطلاعات بیشتر در مورد تصویر بزرگ و ویژگی های خاص HAL، به مشخصات HAL مراجعه کنید که بسته به توزیع در:

usr/share/doc/hal-0.5.14/spec/hal-spec.html/ یافت می شود.

```
-m, --monitor
```

تغییرات چاپی منتشر شده توسط hald daemon.

```
-s, --short
```

خروجی کوتاه

```
-t, --tree
```

نمایش دستگاه های سازمان یافته در یک درخت.

```
-u, --show
```

فقط UDI داده شده (شناسه دستگاه منحصر به فرد) را نشان میدهد.

دستور I-Nex

یک ابزار پیشرفته برای جمع آوری اطلاعات سخت افزار در linux. علاوه بر این ما همچنین می توانیم گزارش پیشرفته ای تولید کنیم و صفحه های خروجی را چاپ کنیم.ابزار I-Nexجزئیات را به صورت سازمان یافته و همچنین زمان واقعی CPU و استفاده از memory را نشان می دهد.

ویژگی هایI-Nex :

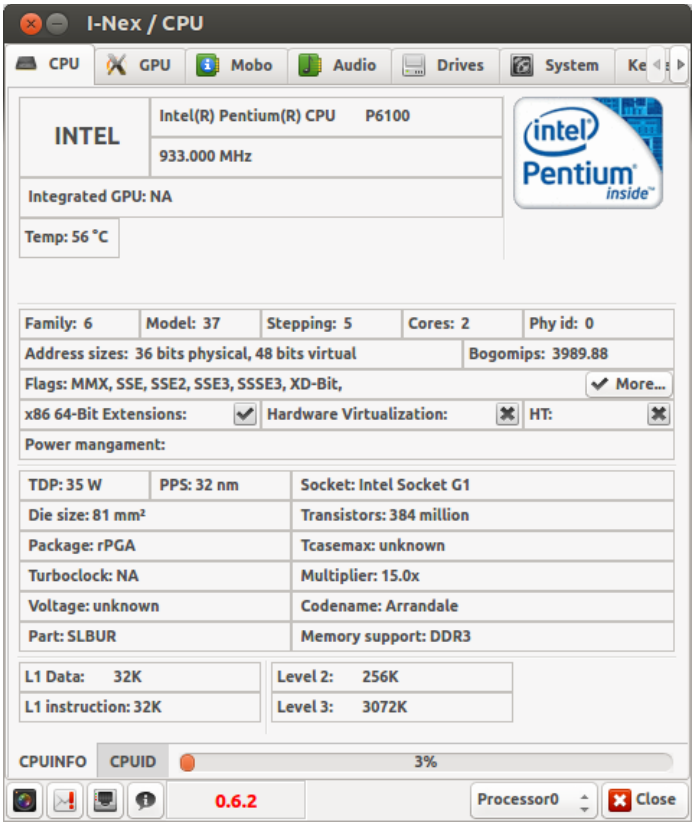
- اطلاعات پردازنده
- اطلاعات گرافیکی
- جزئیات مادربرد
- جزئیات memory
- جزئیات نسخه کرنل
- اطلاعات توزیع system linux عامل linux
- تولید گزارش
- نام hostsystem

نصب I-Nex

```
https://github.com/i-nex/I-Nex
```

نصب از مخزن

```
sudo apt-get install i-nex
dnf install i-nex i-nex-data
sudo pacman -S i-nex
```



دستور hdparm

" hdparm "به عنوان مثال پارامتر دیسک سخت یکی از برنامه های خط فرمان برای لینوکس است که برای اداره دستگاه های دیسک و دیسک های سخت استفاده می شود .با کمک این دستور می توانید آماری درباره هارد دیسک ، تغییر فواصل نوشتن ، مدیریت صوتی و تنظیمات DMA بدست آورید .همچنین می تواند پارامترهای مربوط به حافظه پنهان درایو ، حالت خواب ، مدیریت نیرو ، مدیریت صوتی و تنظیمات DMA را تنظیم کند.

ساختار دستور:

hdparm [دستگاه] [گزینه ها]

توجه:وقتی هیچ پرچمی مشخص نشود ، **acdgkmnru** پیش فرض می شود.

گزینه های دستور:

- **a**- این برای بدست آوردن/تنظیم شمارش بخشی از سیستم بازخوانی پرونده سیستم است که برای افزایش خواندن بدون وقفه پرونده هایی که اندازه آنها بسیار زیاد است، اجرا می شود.
- **A**- ویژگی خواندن به رو به جلو درایو IDE را که به طور پیش فرض معمولاً کار می کند غیرفعال و فعال میکند.
- **b**- برای دریافت / تنظیم وضعیت گذرگاه در جایی که (0 == Off, 1 == On, 2 == tristate) استفاده می شود.
- **B**- برای تنظیم مشخصات پیشرفته مدیریت انرژی (APM) استفاده می شود. .اگر مقدار آن کم باشد ، APMزیاد است و اگر مقدار آن زیاد باشد ، موفقیت عالی تری به دست می آورد .برای غیرفعال کردن APM باید مقدار ۲۵۵ را تنظیم کنید.

- **c-** پشتیبانی I / O 32 بیتی Query / enable (E). در اینجا ، ۳۲ بیتی اشاره به انتقال ورودی از طریق گذرگاه PCI یا VLBدارد.
- **C-** برای بررسی وضعیت حالت IDE در حال انجام استفاده می شود .از پرچم های S- ، Y- ، y-و Z- برای کنترل تکنیک های قدرت IDE به روشی ماهرانه استفاده می شود.
- **d-** پرچم استفاده شده توسط درایو "DMA" را غیرفعال یا فعال می کند .این از طریق ترکیب درایوها و PCI کار می کند.
- **D-** ویژگی مدیریت failer در درایو را فعال یا غیرفعال می کند.
- **E-** سرعت درایو SD / DVD را تنظیم می کند .برای اینکه بتوانید کار کنید ، باید یک عدد سرعت پس از گزینه اختصاص دهید .به طور کلی ، تعداد مورد استفاده دو یا چهار است.
- **f-** این مورد برای همگام سازی و پاک سازی حافظه پنهان دستگاه برای خروجی آن استفاده می شود .این عملکرد را می توان به عنوان بخشی از زمان بندی های t- و T- اجرا کرد.
- **g-** این مورد برای رونمایی از پیکربندی درایو ، جابجایی شروع دستگاه از نقطه شروع درایو استفاده می شود.
- **h-** پیام راهنما را نمایش می دهد و از آن خارج می شود.
- **i-** این اطلاعات شناسایی را که در زمان راه اندازی از درایو بدست آمده است ، نمایان می کند.
- **I-** این داده های شناسایی را مستقیماً از درایو جستجو می کند و ویژگی های بیشتری نسبت به پرچم i- نشان می دهد.
- **k-** این به دریافت / تنظیم پرچم **keep_settings_over_reset** برای درایو کمک می کند.
- **K-** این پرچم **Keep_features_over_reset** درایو را تنظیم می کند . این ویژگی توسط همه درایوها ارائه نشده است.
- **m-** برای بدست آوردن / تنظیم تعداد بخشها برای چندین بخش ورودی / خروجی درایو استفاده می شود .برای غیرفعال کردن این ویژگی باید مقدار را روی صفر تنظیم کنید.
- **M-** این به دریافت / تنظیمات مدیریت صوتی خودکار (AAM) کمک می کند .این ویژگی آزمایشی است و به طور کارآمدی آزمایش نشده است.
- **n-** برای گرفتن / تنظیم پرچم " نادیده گرفتن خطاهای نوشتن " استفاده میشود .بدون رمزگشایی کد منبع درایور ، نباید با این ویژگی بازی کرد.
- **r-** برای گرفتن / تنظیم پرچم فقط خواندنی برای دستگاه استفاده می شود .وقتی این گزینه تنظیم شد ، عملیات نوشتن روی دستگاه اعمال نمی شوند.
- **S-** زمان آماده به کار برای درایو را تنظیم میکند .وقتی مقدار روی صفر تنظیم شود ، خاموش است.
- **T-** زمان خواندن حافظه پنهان را برای اهداف استاندارد و متفاوت اجرا می کند .در صورت عدم دسترسی به دیسک ، از سرعت خواندن مستقیم از بافر حافظه پنهان لینوکس رونمایی می شود ، برای نتیجه بهتر ، باید این عملیات را دو تا سه بار انجام دهید.
- **t-** این هدف همان پرچم T- است با این تفاوت که از سرعت خواندن از طریق حافظه پنهان بر روی دیسک در صورت عدم وجود حافظه پنهان ورودی ورودی ، رونمایی می کند .اگر پرچم T- نیز در اینجا تعریف شده باشد ، اصلاحاتی که پس از T- انجام می شود شامل نتیجه ای است که در زمان کار t- ظاهر شده است.
- **u-** برای بدست آوردن / تنظیم پرچم قطع وقفه برای درایو هنگامی که مقدار روی یک تنظیم می شود ، استفاده می شود سپس درایور میتواند سایر ورودی هایی که در پردازش وقفه های دیسک رخ می دهد را از بین ببرد .این ویژگی می تواند باعث خرابی عظیم سیستم فایل شود ، بنابراین با مسئولیت خود استفاده کنید.
- **v-** این سوییچ همه تنظیمات را نشان می دهد ، به جز i-.
- **w-** در تنظیم مجدد دستگاه کمک می کند.
- **W-** ویژگی های نوشتن حافظه درایو IDE را غیرفعال می کند / فعال می کند.

- **y-** این یک درایو IDE را مجبور می کند بلافاصله وارد حالت آماده به کار کم مصرف شود و معمولاً باعث چرخش آن می شود .وضعیت حالت فعلی را می توان با استفاده از پرچم C- بررسی کرد.
- **Y-** این یک IDE را تحت فشار قرار می دهد تا در حالت خواب کم مصرف برق قطع شود ، تا کاملاً خاموش شود ، در اینجا می توان با کمک پرچم C- وضعیت حالت فعلی را تشخیص داد.
- **z-** این هسته ها را مجبور می کند که جدول پارتیشن را برای دستگاه تعیین شده دوباره بخواند.
- **Z-** این عملکرد صرفه جویی در مصرف اتوماتیک را غیرفعال می کند.

مثال های دستور:

دستور نمایش اطلاعات هارد دیسک :این یکی از مهمترین ویژگی هاست زیرا از جزئیات درایو دیسک سخت رونمایی می کند ، شما باید در اینجا از گزینه I- و درایو دیسک سخت استفاده کنید.

```
$ hdparm -I / dev / sda
```

```
/dev/sda:

ATA device, with non-removable media
        Model Number:       ST1000LM014-SSHD-8GB
        Serial Number:      W770LW8S
        Firmware Revision:  LVD4
        Transport:          Serial, ATA8-AST, SATA 1.0a, SATA II Extensions, SATA Rev 2.5, SATA Rev 2
0
Standards:
        Used: unknown (minor revision code 0x001f)
        Supported: 9 8 7 6 5
        Likely used: 9
Configuration:
        Logical         max     current
        cylinders       16383   16383
        heads           16      16
        sectors/track   63      63
        --
        CHS current addressable sectors:   16514064
        LBA    user addressable sectors:  268435455
        LBA48  user addressable sectors: 1953525168
        Logical  Sector size:                   512 bytes
        Physical Sector size:                  4096 bytes
        Logical Sector-0 offset:                  0 bytes
        device size with M = 1024*1024:      953869 MBytes
        device size with M = 1000*1000:     1000204 MBytes (1000 GB)
        cache/buffer size  = unknown
        Form Factor: 2.5 inch
        Nominal Media Rotation Rate: 5400
Capabilities:
        LBA, IORDY(can be disabled)
        Queue depth: 32
        Standby timer values: spec'd by Standard, no device specific minimum
        R/W multiple sector transfer: Max = 16  Current = ?
        Advanced power management level: 128
        Recommended acoustic management value: 254, current value: 0
        DMA: mdma0 mdma1 mdma2 udma0 udma1 udma2 udma3 udma4 udma5 *udma6
             Cycle time: min=120ns recommended=120ns
        PIO: pio0 pio1 pio2 pio3 pio4
             Cycle time: no flow control=120ns  IORDY flow control=120ns
Commands/features:
        Enabled Supported:
           *    SMART feature set
                Security Mode feature set
           *    Power Management feature set
           *    Write cache
           *    Look-ahead
           *    Host Protected Area feature set
           *    WRITE_BUFFER command
```

برای نمایش همه گزینه ها دستور دهید:

```
$ hdparm -h
```

```
hdparm - get/set hard disk parameters - version v9.48, by Mark Lord.

Usage:  hdparm  [options] [device ...]

Options:
 -a   Get/set fs readahead
 -A   Get/set the drive look-ahead flag (0/1)
 -b   Get/set bus state (0 == off, 1 == on, 2 == tristate)
 -B   Set Advanced Power Management setting (1-255)
 -c   Get/set IDE 32-bit IO setting
 -C   Check drive power mode status
 -d   Get/set using_dma flag
 -D   Enable/disable drive defect management
 -E   Set cd/dvd drive speed
 -f   Flush buffer cache for device on exit
 -F   Flush drive write cache
 -g   Display drive geometry
 -h   Display terse usage information
 -H   Read temperature from drive (Hitachi only)
 -i   Display drive identification
 -I   Detailed/current information directly from drive
 -J   Get/set Western DIgital "Idle3" timeout for a WDC "Green" drive (DANGEROUS)
 -k   Get/set keep_settings_over_reset flag (0/1)
 -K   Set drive keep_features_over_reset flag (0/1)
 -L   Set drive doorlock (0/1) (removable harddisks only)
 -m   Get/set multiple sector count
 -M   Get/set acoustic management (0-254, 128: quiet, 254: fast)
 -n   Get/set ignore-write-errors flag (0/1)
 -N   Get/set max visible number of sectors (HPA) (VERY DANGEROUS)
```

دستور تست سرعت درایو دیسک سخت:

```
$ hdparm -t / dev / vdb
```

```
/dev/vdb:
 Timing buffered disk reads: 172 MB in  3.02 seconds =  56.94 MB/sec
```

دستور اندازه گیری سرعت خواندن حافظه کش دیسک سخت:

```
$ hdparm -T / dev / vdb
```

خروجی دستور:

```
/dev/vdb:
 Timing cached reads:   10248 MB in  2.00 seconds = 5127.25 MB/sec
```

دستور فعال کردن پیش خواندن:

```
$ hdparm -A 1 / dev / sda
```

دستور تغییر درایو به پایین ترین درجه مدیریت نیرو:

```
$ hdparm -B 254 / dev / sda
```

برای دریافت تنظیمات فعلی دستور دهید:

```
$ hdparm -d / dev / sdX
```

دستور روشن کردن DMA برای یک دستگاه:

```
$ hdparm -d1 / dev / hda
```

دستور کاهش نویز ایجاد شده توسط برخی از دیسک های سخت کلاسیک با کاهش عملکرد دیسک:

```
$ hdparm -M 128 / dev / sda
```

دستور sdparm

دستور sdparm در سیستم‌عامل‌های Unix/Linux برای مدیریت پارامترها و تنظیمات دستگاه‌های SCSI (Small Computer System Interface) استفاده می‌شود. این دستور به شما امکان می‌دهد تا اطلاعات مربوط به دستگاه‌های SCSI را بخوانید، تنظیمات آن‌ها را تغییر دهید و کنترل کنید.

مثال های دستور:

نمایش وضعیت پارامترهای دستگاه SCSI: این دستور برای نمایش وضعیت پارامترهای یک دستگاه SCSI خاص استفاده می‌شود.

```
sdparm -g WCE /dev/sdb
```

تنظیم یک پارامتر برای دستگاه SCSI: این دستور برای تغییر یا تنظیم یک پارامتر خاص برای یک دستگاه SCSI استفاده می‌شود.

```
sdparm --set STANDBY=1 /dev/sdc
```

نمایش تمام پارامترهای دستگاه SCSI: این دستور برای نمایش تمام پارامترهای یک دستگاه SCSI مانند وضعیت حالت Standby, Write Cache و غیره استفاده می‌شود.

```
sdparm --get=all /dev/sdd
```

تنظیم وضعیت Write Cache برای دستگاه SCSI: این دستور برای فعال یا غیرفعال کردن وضعیت Write Cache برای یک دستگاه SCSI استفاده می‌شود.

```
sdparm --set=WCE /dev/sde
```

نمایش اطلاعات SMART برای دستگاه SCSI: این دستور برای نمایش اطلاعات SMART (Self-Monitoring, Analysis, and Reporting Technology) برای یک دستگاه SCSI استفاده می‌شود.

```
sdparm --get=smart /dev/sdf
```

دستور ftrim

شرح دستور

دستور `fstrim` ابزاری است که برای بهینه سازی عملکرد درایوهای حالت جامد (SSD) طراحی شده است. SSDها از مکانیزمی به نام TRIM استفاده می‌کنند که به سیستم عامل اجازه می‌دهد تا بلوک‌های داده‌ای که دیگر استفاده نمی‌شوند را علامت‌گذاری کند. این کار به SSD کمک می‌کند تا عملیات پاک‌سازی را به صورت پس‌زمینه انجام دهد و عملکرد کلی را بهبود بخشد.

ساختار دستور

```
fstrim [-o offset] [-l length] [-m minimum-free-extent] [-v] mountpoint
```

کاربردهای دستور

- **بهبود عملکرد SSD:** با حذف بلوک‌های داده‌ای بلااستفاده، عملکرد نوشتن و خواندن SSD بهبود می‌یابد.
- **افزایش طول عمر SSD:** با کاهش تعداد عملیات پاک‌سازی، طول عمر SSD افزایش می‌یابد.
- **بهینه‌سازی برای سیستم‌های فایل خاصی مانند ext4:** دستور `fstrim` به طور خاص برای سیستم فایل ext4 بهینه‌سازی شده است.

گزینه‌های دستور و مثال‌ها

- **-o offset:** آفست ابتدایی بلوک‌هایی که باید پاک شوند را مشخص می‌کند.
- **-l length:** طول بلوک‌هایی که باید پاک شوند را مشخص می‌کند.
- **-m minimum-free-extent:** حداقل اندازه یک بلوک آزاد برای پاک شدن را مشخص می‌کند.
- **-v:** حالت مفصل (Verbose) را فعال می‌کند و اطلاعات بیشتری را در مورد عملیات انجام شده نمایش می‌دهد.
- **mountpoint:** نقطه اتصال سیستم فایلی که باید TRIM شود.

مثال‌ها:

- **TRIM کردن کل سیستم فایل /home:**

```
sudo ftrim /home
```

- **TRIM کردن ۱۰ مگابایت از ابتدای سیستم فایل /var:**

```
sudo ftrim -o 0 -l 10M /var
```

- **TRIM کردن با حالت مفصل و حداقل اندازه بلوک آزاد ٤ مگابایت:**

```
sudo ftrim -v -m 4M /mnt
```

نکات مهم

- **اجرای خودکار:** بسیاری از سیستم‌های لینوکس به صورت خودکار `fstrim` را در فواصل زمانی مشخص اجرا می‌کنند.
- **سیستم فایل:** `fstrim` برای سیستم فایل‌های خاصی مانند ext4 ، Btrfs و F2FS طراحی شده است.
- **عملکرد:** اجرای `fstrim` به صورت دستی ممکن است باعث کند شدن موقت سیستم شود.
- **بک‌آپ:** همیشه قبل از انجام هرگونه تغییری در سیستم فایل، از داده‌های خود بک‌آپ تهیه کنید.

چه زمانی باید `fstrim` را اجرا کنیم؟

- **پس از نصب سیستم عامل جدید:** برای بهینه‌سازی عملکرد SSD پس از نصب سیستم عامل.
- **پس از حذف حجم زیادی از داده:** برای حذف بلوک‌های داده‌ای بلااستفاده.
- **به صورت دوره‌ای:** برای حفظ عملکرد بهینه SSD ، توصیه می‌شود `fstrim` را به صورت دوره‌ای اجرا کنید (مثلاً یک بار در هفته).

اسکریپت برای اجرای خودکار ftrim

برای اجرای خودکار `fstrim` می‌توانید از یک اسکریپت استفاده کنید .به عنوان مثال، یک اسکریپت ساده برای اجرای `fstrim` روی تمام پارتیشن‌های ext4:

```
#!/bin/bash
 findmnt -t ext4 | cut -d' ' -f1 | while read partition; do
  sudo ftrim "$partition"
done
```

این اسکریپت را با نام دلخواه ذخیره کنید (مثلاً `trim.sh`) و سپس آن را با استفاده از یک ابزار زمان‌بندی مانند `cron` اجرا کنید.

دستور iscsiadm

شرح دستور

`iscsiadm` یک ابزار خط فرمان برای مدیریت iSCSI Initiator است. این ابزار به شما اجازه می دهد تا iSCSI Targets را کشف، پیکربندی، مدیریت و نظارت کنید.

ساختار دستور

```
iscsiadm -m mode [options]
```

- `-m`: مشخص کننده حالت عملکرد (discovery, node, session, iface, fw, host, priority)
- `options`: گزینه های خاص برای هر حالت

کاربردهای دستور

- کشف iSCSI Targets
- مدیریت iSCSI Nodes (Targets)
- مدیریت iSCSI Sessions
- مدیریت iSCSI Interfaces
- نمایش اطلاعات Firmware
- مدیریت CHAP Authentication
- مدیریت اولویت iSCSI Connections

گزینه‌های کامل دستور

حالت discovery

- کشف iSCSI Targets
- `-t type`: نوع کشف (sendtargets, isns, fw)
- `-p ip:port`: آدرس IP و پورت سرور کشف
- `-I iface`: رابط شبکه برای کشف
- `-l`: لیست کردن Targets کشف شده
- `-D`: حذف تمام ورودی های کشف شده

حالت discoverydb

- مدیریت پایگاه داده کشف
- `-p ip:port`: آدرس IP و پورت سرور کشف
- `-t type`: نوع کشف
- `-o operation`: عملیات (discover, remove, list)
- `-n name`: نام ورودی کشف
- `-v value`: مقدار برای عملیات

حالت node

- مدیریت iSCSI Nodes (Targets)
- `-T targetname`: نام iSCSI Target
- `-p ip:port`: آدرس IP و پورت iSCSI Target
- `-I iface`: رابط شبکه برای اتصال
- `-l`: لیست اطلاعات درباره Node
- `-u`: حذف Node
- `-R`: اسکن مجدد Node
- `-s`: شروع یا توقف Node
- `-o operation`: عملیات دیگر (login, logout, modify, ...)
- `-n name`: نام گزینه
- `-v value`: مقدار برای گزینه

حالت session

- مدیریت iSCSI Sessions
- `-r sessionid`: ID Session
- `-R`: اسکن مجدد Session
- `-u`: حذف Session
- `-s`: شروع یا توقف Session
- `-o new`: ایجاد Session جدید

حالت iface

- مدیریت iSCSI Interfaces
- `-I ifacename`: نام رابط شبکه
- `-H hostno|MAC`: شماره میزبان یا آدرس MAC
- `-o operation`: عملیات (add, remove, modify)
- `-n name`: نام گزینه

- -v value: مقدار برای گزینه
- -C ping: انجام تست پینگ

حالت fw

- نمایش اطلاعات Firmware
- -l: لیست اطلاعات Firmware

حالت host

- مدیریت CHAP Authentication
- -P printlevel: سطح جزئیات خروجی
- -H hostno|MAC: شماره میزبان یا آدرس MAC
- -C chap: مدیریت تنظیمات CHAP
- -o operation: عملیات (add, remove, modify)
- -v chap_tbl_idx: ایندکس جدول CHAP

حالت priority

- مدیریت اولویت iSCSI Connections
- بدون گزینه

مثال‌ها

- کشف iSCSI Targets:

```
iscsiadm -m discovery -t sendtargets -p 192.168.1.100:3260
```

- ایجاد یک اتصال iSCSI:

```
iscsiadm -m node -T iqn.2007-10.com.example:target1 -p 192.168.1.100:3260 --login
```

- لیست کردن اطلاعات درباره یک Session:

```
iscsiadm -m session -r 1
```

توجه : استفاده از این دستور نیاز به دسترسی root دارد.

دستور nproc

دستور nproc برای نشان دادن تعداد پردازنده موجود در رایانه استفاده می شود. دستور nproc در لینوکس با گزینه‌های مختلفی قابل استفاده است. در زیر توضیح و مثال‌هایی برای هر یک از گزینه‌های nproc آورده شده است:

گزینه a- یا all--

این گزینه تعداد هسته‌های پردازنده مجازی ( )virtual CPU coresرا نشان می‌دهد.

```
nproc -a
```

گزینه i- یا ignore--

این گزینه تعداد هسته‌های پردازنده مجازی را نشان می‌دهد، اما اگر فرایندهای اجرایی محدودیت‌های مرتبط با تعداد هسته‌ها داشته باشند، این محدودیت‌ها را نادیده می‌گیرد.

```
nproc -i
```

گزینه N- یا allcores--

این گزینه تعداد همه هسته‌های پردازنده را نشان می‌دهد، حتی اگر همه آن‌ها فعال نباشند.

```
nproc -N
```

گزینه p- یا phys--

این گزینه تعداد هسته‌های پردازنده فیزیکی را نشان می‌دهد.

```
nproc -p
```

گزینه v- یا virtual--

این گزینه تعداد هسته‌های پردازنده مجازی را نشان می‌دهد.

```
nproc -v
```

نمایش همه هسته‌های پردازنده:

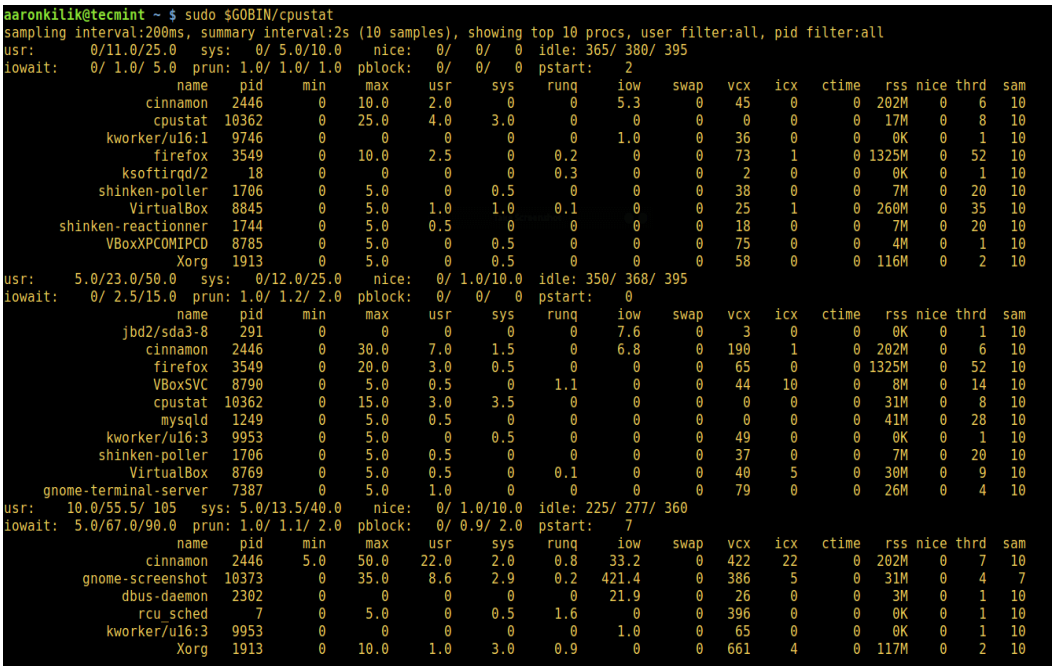
```
nproc -a
```

دستور Cpustat

استفاده از CPU را با فرآیندهای در حال اجرا

Cpustat یک برنامه قدرتمند برای اندازه گیری عملکرد system برای linux است که با استفاده از زبان برنامه نویسیGo نوشته شده است. داده های خروجی Cpustatرا به دو روش ممکن ارائه می دهد: یک لیستی از متن از به صورت خلاصه و داشبورد پیمایش رنگی برای هر نمونه. نحوه نصب Cpustat در linux به ادرس زیر مراجعه نمایید.

```
github.com/uber-common/cpustat
```



linux CPU نظارت بر استفاده از Cpustat

نحوه نصب CoreFreq

برای نصب CoreFreqابتدا باید پیش نیازها را برای کامپایل و ساخت برنامه از منبع تهیه کنید.

```
$ sudo dnf group install 'Tools Development'
# sudo apt install dkms git libpthread-stubs0-dev
```

نصب برنامه CoreFreq از سورس

توجه: کاربران Arch Linux می توانند corefreq-gitرااز AUR نصب کنند.

حالا دستورات زیر را اجرا کنید تا ماژول هسته linux را از فهرست محلی بجای دیمون[1] بارگذاری کنید:

```
$ sudo insmod corefreqk.ko
$ sudo ./corefreqd
```

سپس، برنامه را به عنوان یک کاربر شروع کنید.

```
$ ./corefreq-cli
```

نظارت بر CPU با CoreFreq

از رابط کاربری بالا می توانید از کلید های میانبر استفاده کنید:

۱. F2برای نمایش یک منو استفاده به عنوان در بخش بالا از صفحه نمایش دیده می شود.
۲. Rightو Leftفلش به حرکت در زبانه منو.
۳. فلش های Upو Downبرای انتخاب یک آیتم منو، سپس روی [Enter] کلیک کنید.
۴. F4برنامه را خواهد بست.
۵. یک مرجع سریع باز خواهد کرد.

برای مشاهده تمام گزینه های استفاده، دستور زیر را تایپ کنید:

```
$ ./corefreq-cli -h
```

[1] Daemon

برای چاپ اطلاعات در مورد هسته، اجرا کنید:

```
$ ./corefreq-cli -k
```

جزئیات شناسایی پردازنده:

```
$ ./corefreq-cli -u
```

همچنین می توانید دستورات CPU را در زمان واقعی نظارت کنید:

```
$ ./corefreq-cli -i
```

ردیابی به صورت زیر فعال کنید:

```
$ ./corefreq-cli -c
```

دستور info

دستور **info** مستندات را در قالب **info** می خواند . این info دقیق را برای یک فرمان در مقایسه با صفحه man به شما می دهد .این صفحات با استفاده از ابزار **texinfo** ساخته می شوند ، به همین دلیل می تواند با صفحات دیگر پیوند برقرار کند ، منوها و ناوبری آسان ایجاد کند. یکی دیگر از ابزارهای مفیدی که می توانید برای کسب اطلاعات بیشتر در مورد دستورات لینوکس از آن استفاده کنید، دستور اطلاعات است. info اسناد را در قالب اطلاعات می خواند (فرمت خاصی که معمولاً از منبع Texinfo تولید می شود). صفحات اطلاعات معمولاً اطلاعات دقیق تری را در مورد یک فرمان نسبت به صفحات مرد مربوطه آن ارائه می دهند. اطلاعات همچنین به پیمایش و پیوند بین صفحات اجازه می‌دهد.

ساختار دستور:

```
info فهرست منو [گزینه] ...
```

گزینه های دستور:

- **-a ،–all**: از همه راهنماهای منطبق استفاده می کند.
- **-k ،–apropos = STRING**: STRING را در تمام شاخص های همه راهنما جستجو می کند.
- **-d ،–directory = DIR**: این DIR را به INFOPATH اضافه می کند.
- **-f ،–file = MANUAL**: این دفترچه راهنما را برای بازدید مشخص می کند.
- **-h ،–help**: این راهنما را نمایش می دهد و از آن خارج می شود.
- **-n ،–node = NODENAME**: این گره ها را در پرونده info ی که برای اولین بار بازدید شده مشخص می کند.
- **-o ،–output = FILE**: گره های انتخاب شده را به FILE منتقل می کند.
- **-O ،–show-options ،–usage**: به گره گزینه های خط فرمان می رود.
- **-v ،–variable VAR = VALUE**: مقدار متغیری را به متغیر info ی VAR اختصاص می دهد.
- **–version** : info نسخه را نمایش می دهد و از آن خارج می شود.
- **-w ،– Where ،–location**: این مکان فیزیکی فایل info را چاپ می کند.

مثال ها:

-a: از همه راهنماهای منطبق استفاده می کند و آنها را برای یک دستور خاص نمایش می دهد.

```
info -a cvs
```

```
Node: File names matching 'cvs'

Info File Index
***************

File names that match 'cvs':

* Menu:

*     1: (/usr/share/info/cvs.info.gz).
*     2: (*manpages*)cvs.
```

-k این STRING را در تمام شاخص های همه راهنما جستجو می کند و سپس همان را نشان می دهد.

CV info -k

```
linux@ubuntu:~$ info -k cvs
"(cvs)Concurrency" -- #cvs.lock, removing
"(cvs)Locks" -- #cvs.lock, technical details
"(cvs)Locks" -- #cvs.pfl, technical details
"(cvs)Backing up" -- #cvs.rfl, and backups
"(cvs)Concurrency" -- #cvs.rfl, removing
"(cvs)Locks" -- #cvs.rfl, technical details
"(cvs)Locks" -- #cvs.tfl
"(cvs)Concurrency" -- #cvs.wfl, removing
"(cvs)Locks" -- #cvs.wfl, technical details
"(cvs)Specifying a repository" -- .bashrc, setting CVSROOT in
"(cvs)Specifying a repository" -- .cshrc, setting CVSROOT in
"(cvs)~/" -- .cvsrc file
"(cvs)Specifying a repository" -- .profile, setting CVSROOT in
"(cvs)Specifying a repository" -- .tcshrc, setting CVSROOT in
"(cvs)Repository" -- /usr/local/cvsroot, as example repository
"(cvs)Working directory storage" -- Base directory, in CVS directory
"(cvs)Working directory storage" -- Baserev file, in CVS directory
"(cvs)Working directory storage" -- Baserev.tmp file, in CVS directory
"(cvs)BUGS" -- Bugs in this manual or CVS
"(cvs)Compatibility" -- Compatibility, between CVS versions
"(cvs)config" -- config, in CVSROOT
"(cvs)What is CVS?" -- Contributors (CVS program)
"(cvs)What is CVS?" -- Credits (CVS program)
"(cvs)Watches Compatibility" -- CVS 1.6, and watches
"(cvs)Structure" -- CVS command structure
"(cvs)CVS in repository" -- CVS directory, in repository
"(cvs)Working directory storage" -- CVS directory, in working directory
"(cvs)Password authentication server" -- CVS passwd file
"(cvs)What is CVS?" -- CVS, history of
"(cvs)What is CVS?" -- CVS, introduction to
"(cvs)Compatibility" -- CVS, versions of
"(cvs)Working directory storage" -- CVS/Base directory
"(cvs)Working directory storage" -- CVS/Baserev file
"(cvs)Working directory storage" -- CVS/Baserev.tmp file
"(cvs)Working directory storage" -- CVS/Entries file
"(cvs)Working directory storage" -- CVS/Entries.Backup file
"(cvs)Working directory storage" -- CVS/Entries.Log file
```

-d :را به INFOPATH اضافه می کند و همچنین همان را نمایش می دهد.

info -d cvs

```
File: dir,      Node: Top,      This is the top of the INFO tree.

This is the Info main menu (aka directory node).
A few useful Info commands:

  'q' quits;
  'H' lists all Info commands;
  'h' starts the Info tutorial;
  'mTexinfo RET' visits the Texinfo manual, etc.

* Menu:

Basics
* Common options: (coreutils)Common options.
* Coreutils: (coreutils).       Core GNU (file, text, shell) utilities.
* Date input formats: (coreutils)Date input formats.
* Ed: (ed).                     The GNU line editor
* File permissions: (coreutils)File permissions.
                                Access modes.
* Finding files: (find).        Operating on files matching certain criteria.

C++ libraries
* autosprintf: (autosprintf).   Support for printf format strings in C++.

Compression
* Gzip: (gzip).                 General (de)compression of files (lzw).

Development
* SSIP: (ssip).                 Speech Synthesis Interface Protocol.
* Speech Dispatcher: (speech-dispatcher).
                                Speech Dispatcher.

DOS
* Mtools: (mtools).             Mtools: utilities to access DOS disks in Unix.

Editors
-----Info: (dir)Top, 264 lines --Top-----------------------------------------------------------------------------------------------------------
```

-O :برای دستور خاصی به گره گزینه های خط فرمان رفته و همان را نمایش می دهد.

info -O cvs



-w :محل فیزیکی فایل info را چاپ می کند.

```
info -w cvs
```

```
linux@ubuntu:~$ info -w cvs
/usr/share/info/cvs.info.gz
linux@ubuntu:~$
```

توجه داشته باشید:

برای بررسی صفحه دستی info ، از دستور زیر استفاده کنید:

```
man info
```

برای بررسی صفحه راهنمای دستور info ، از دستور زیر استفاده کنید:

```
info -h
```

بدست آوردن اطلاعات سخت افزاری سیستم

Hardware Info

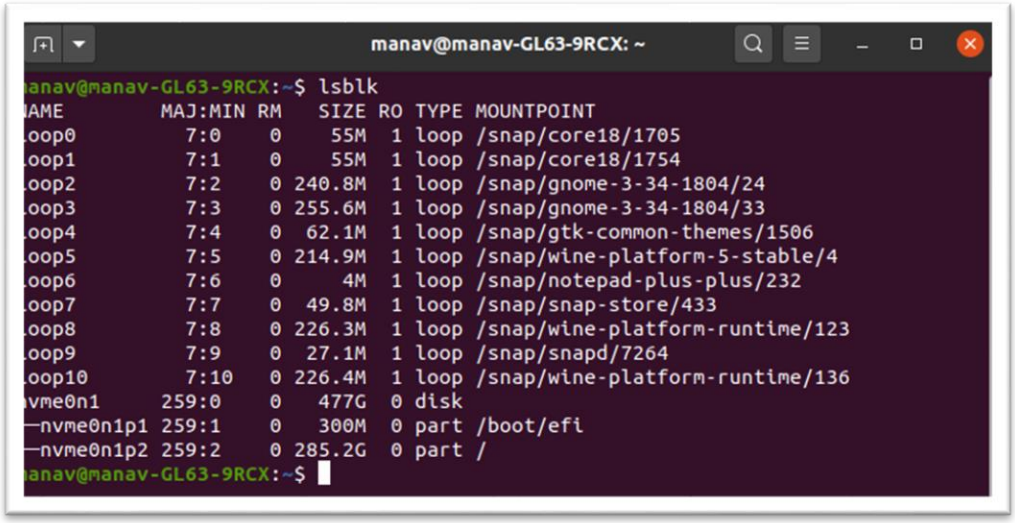
/usr/bin/lsdev	List devices and info on system hardware. Also IRQ's.(RPM package procinfo) Also cat /proc/devices
/sbin/lspci	list all PCI devices (result of probe) Also lspci -vvx and cat /proc/pci
cat /proc/interrupts	List IRQ's used by system and the device using the interrupt.
cat /proc/ioports	List I/O ports used by system.
cat /proc/dma	List DMA channels and device used by system.
cat /proc/cpuinfo	List info about CPU.

دستور lsblk

Lsblkبرای نمایش جزئیات در مورد دستگاههای بلوک استفاده می شود و این دستگاههای بلوک (به جز دیسک رم) اساساً آن فایلهایی هستند که نمایانگر دستگاههای متصل به رایانه هستند.این نمایش داده شد **SYS** / سیستم فایل مجازی و **از udev** به دست آوردن اطلاعات است که آن را نمایش می دهد .و اساساً خروجی را در یک ساختار درخت مانند نشان می دهد.

نمایش دستگاه های بلوک.

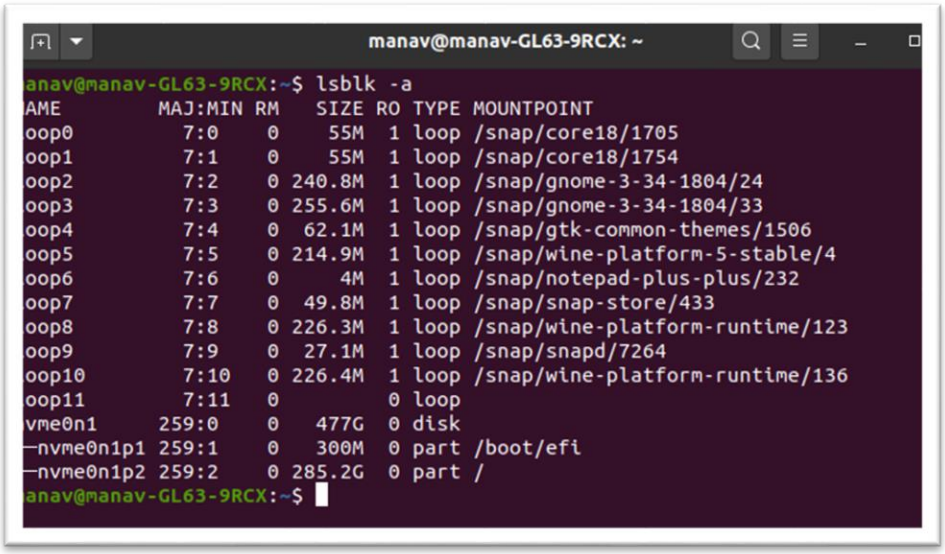
```
$ lsblk
```



این لیست دستگاه های بلوک شده در سیستم شما را نشان می دهد.

بدست آوردن دستگاه های بلوک خالی

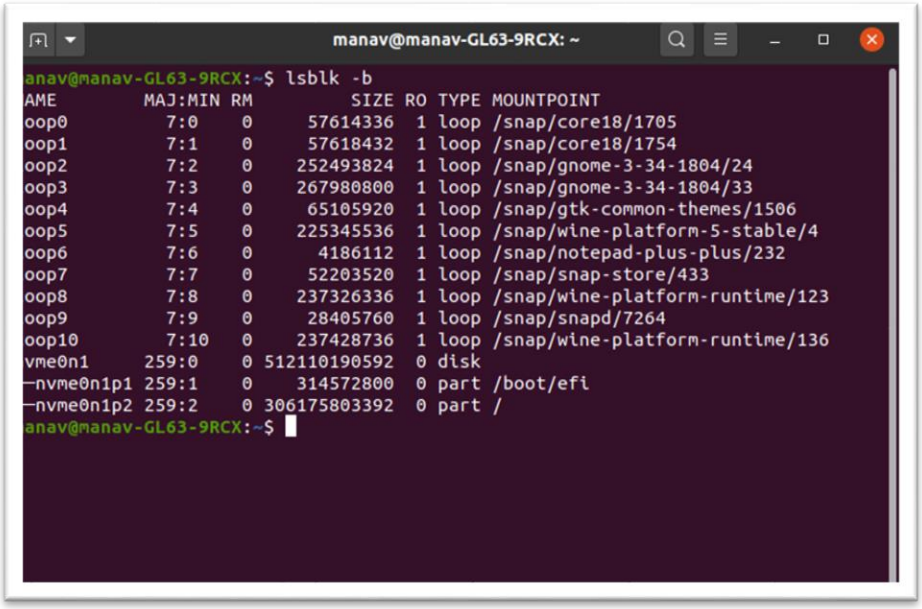
```
#lsblk -a
```



با این کار همه دستگاه های مسدود شده به همراه دستگاه های خالی نمایش داده می شوند.

برای چاپ اطلاعات اندازه در بایت

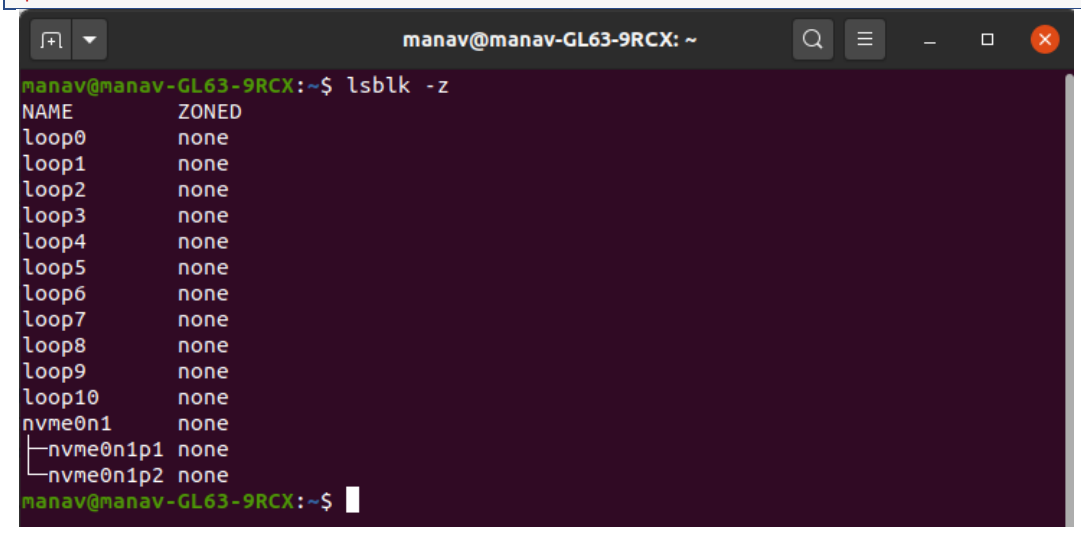
```
$ lsblk -b
```



همه دستگاه های بلوک را به همراه اندازه آنها در بایت نمایش می دهد.

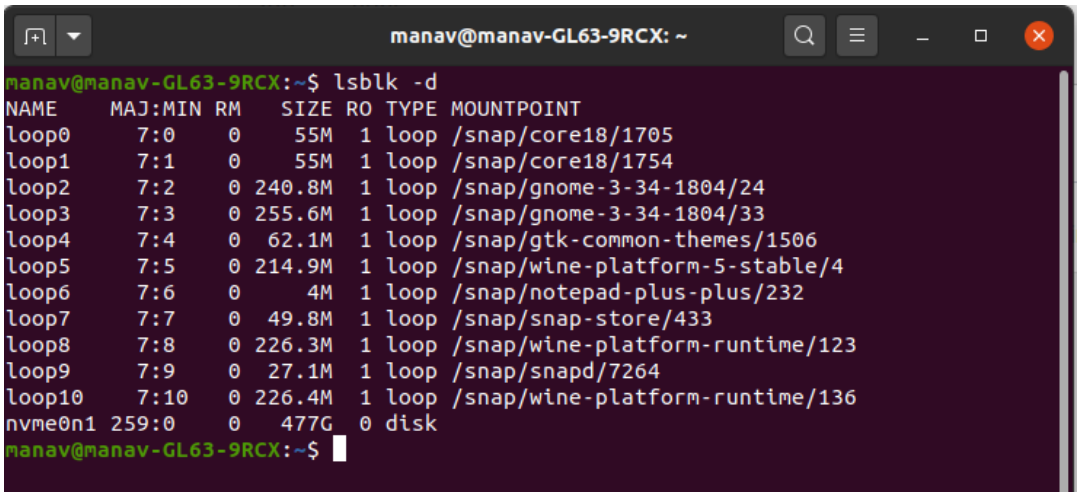
برای چاپ مدل منطقه برای دستگاه ها.

```
$ lsblk -z
```



برای جستجوی slave inputs

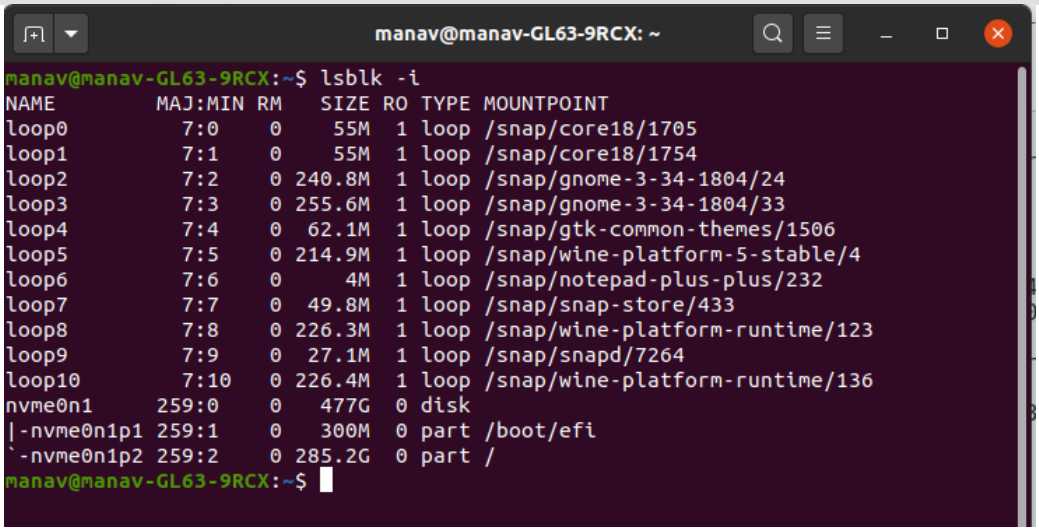
```
$ lsblk -d
```



با این کار از تمام slave های ورودی نادیده گرفته می شوند.

استفاده از نویسه های ASCII برای قالب بندی درخت.

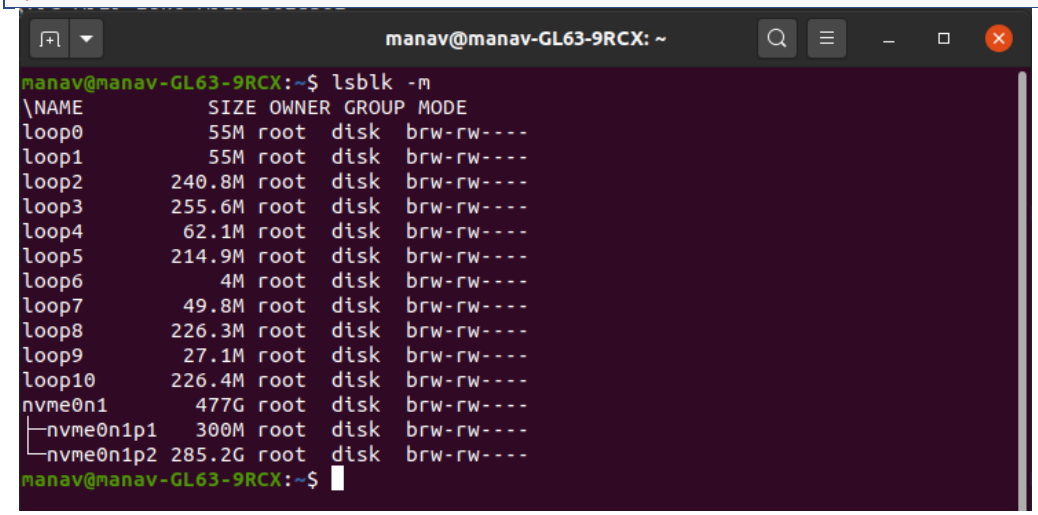
```
$ lsblk -i
```



این قالب بندی درخت را با مقادیر کاراکتر ASCII نمایش می دهد.

برای چاپ اطلاعات مربوط به مالک دستگاه ، گروه و حالت دستگاه های بلوک.

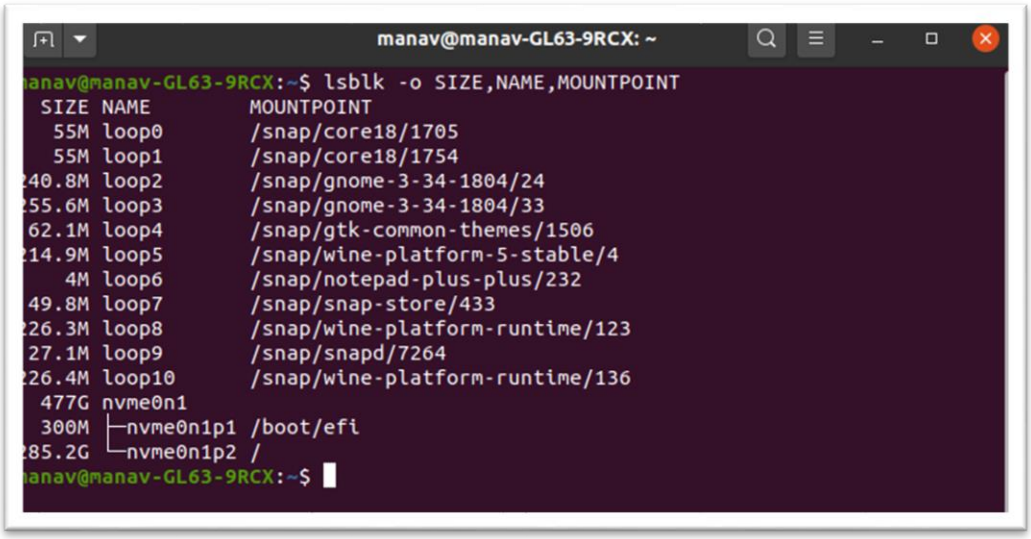
```
$ lsblk -m
```



با این کار تمام جزئیات مالک دستگاه ، گروه و حالت دستگاه های بلوک نمایش داده می شود.

چاپ دسته های انتخاب شده دستگاه های بلوک.

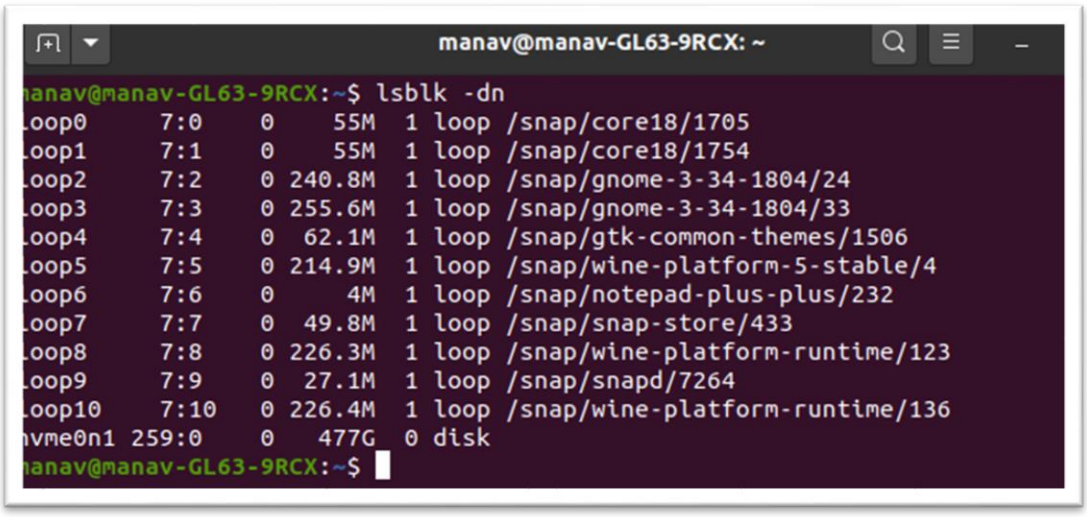
```
$ lsblk -o SIZE ،NAME ،MOUNTPOINT
```



این فقط ستون های مشخص شده را چاپ می کند.

مخفی کردن عناوین ستون ها

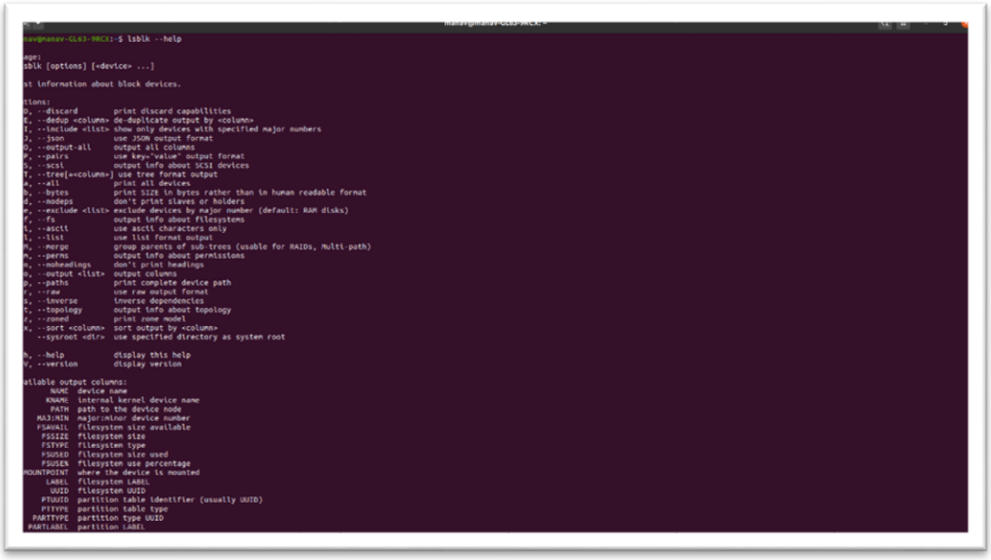
```
$ lsblk -dn
```

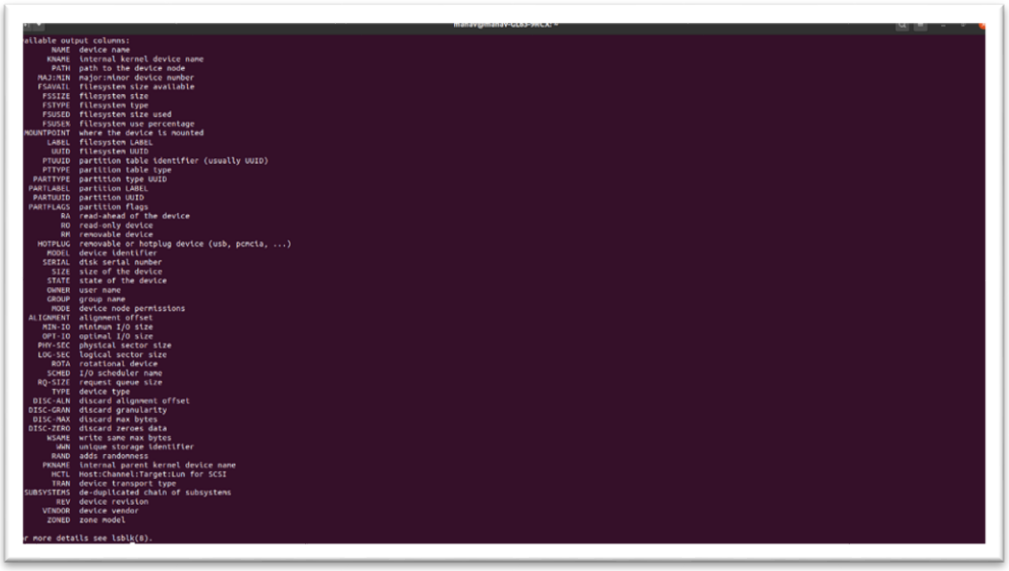


با این کار عناوین ستون پنهان می شوند.

نمایش بخش راهنما از دستور

```
$ lsblk - help
```





این بخش راهنمای دستور را نمایش می دهد.

دستور blkid

blkid یک ابزار خط فرمان در سیستم‌عامل‌های لینوکس است که برای شناسایی و نمایش اطلاعات مربوط به پارتیشن‌ها و دستگاه‌های بلوکی استفاده می‌شود .این اطلاعات شامل UUID ، نوع سیستم فایل، LABEL و سایر مشخصه‌های مهم است که برای مدیریت و مانت کردن دستگاه‌ها بسیار مفید است.

شرح دستور

دستور blkid با اسکن دستگاه‌های بلوکی موجود در سیستم، اطلاعات مربوط به هر دستگاه را جمع‌آوری کرده و به صورت قابل خواندن نمایش می‌دهد .این اطلاعات برای ابزارهای دیگر مانند fstab ، udev و سایر ابزارهای مدیریت دستگاه‌ها استفاده می‌شود.

ساختار دستور:

```
blkid [options] [DEVICE]
```

- **options:** گزینه‌های مختلفی که رفتار دستور را تغییر می‌دهند.
- **DEVICE:** دستگاه بلوکی خاصی که می‌خواهید اطلاعات آن را نمایش دهید .اگر این گزینه مشخص نشود، blkid تمام دستگاه‌های بلوکی را اسکن می‌کند.

کاربردهای دستور:

- **نمایش اطلاعات دستگاه‌های بلوکی :** با اجرای ساده دستور blkid ، اطلاعات کلی تمام دستگاه‌های بلوکی نمایش داده می‌شود.
- **یافتن UUID یک دستگاه :** UUID یک شناسه منحصر به فرد برای هر دستگاه است که در فایل fstab برای مانت کردن دستگاه‌ها استفاده می‌شود.
- **یافتن نوع سیستم فایل :** با استفاده از blkid می‌توانید نوع سیستم فایل هر دستگاه را به راحتی تشخیص دهید.
- **یافتن LABEL یک دستگاه :** LABEL نامی است که به یک دستگاه اختصاص داده می‌شود و برای شناسایی آن در محیط گرافیکی یا اسکریپت‌ها استفاده می‌شود.

گزینه‌های مهم دستور:

- **-o, --output:** خروجی را به یک قالب خاص (مانند JSON ، XML) تغییر می‌دهد.
- **-s, --show:** لیستی از ویژگی‌های خاصی را نمایش می‌دهد (مثلاً TYPE ، LABEL، UUID).
- **-L, --label:** فقط دستگاه‌هایی را نشان می‌دهد که دارای LABEL مشخصی هستند.
- **-U, --UUID:** فقط دستگاه‌هایی را نشان می‌دهد که دارای UUID مشخصی هستند.
- **-t, --type:** فقط دستگاه‌هایی را نشان می‌دهد که دارای نوع سیستم فایل مشخصی هستند.

مثال‌ها:

- **نمایش اطلاعات تمام دستگاه‌های بلوکی:**

```
blkid
```

- **نمایش UUID دستگاه /dev/sda1:**

```
blkid /dev/sda1
```

- **نمایش فقط نوع سیستم فایل تمام دستگاه‌ها:**

```
blkid -s TYPE
```

- **نمایش اطلاعات دستگاهی که دارای "LABEL "home است:**

```
blkid -L home
```

کاربردهای عملی:

- **ایجاد فایل fstab:** اطلاعاتی که توسط blkid ارائه می‌شود، برای ایجاد فایل fstab که برای مانت کردن خودکار دستگاه‌ها در هنگام بوت شدن استفاده می‌شود، بسیار مفید است.
- **اسکریپت‌نویسی blkid:** می‌تواند در اسکریپت‌های مختلف برای خودکارسازی وظایف مربوط به مدیریت دستگاه‌ها استفاده شود.
- **عیب‌یابی:** در مواقعی که مشکلی در ارتباط با دستگاه‌های بلوکی وجود دارد، blkidمی‌تواند برای تشخیص علت مشکل بسیار مفید باشد.

نکته: خروجی دستور blkid معمولاً به صورت زیر است:

```
/dev/sda1: UUID="e2c87a52-a1b8-471c-b69c-6b5758c34938" TYPE="ext4"
/dev/sdb1: LABEL="home" UUID="827db6e5-4b8c-4dba-8b04-2b9bf72a67c0"
TYPE="ext4"
```

در این مثال، UUID، TYPEو LABEL هر دستگاه به وضوح مشخص شده است.

بدست آوردن اطلاعات سیستم عامل

```
cat /etc/*-release
cat /etc/redhat-release
یا
cat /etc/issue
یا
cat /proc/version
یا
uname -a
```

بدست آوردن اطلاعات USB

دستور lsusb برای گزارش اطلاعات مربوط به کنترل کننده های USB و تمام دستگاه هایی که به آنها متصل است استفاده می شود.

```
hosein @ hosein ~ $ lsusb
```

```
bus ۰۰۱ device۰۰۲ : ID 8087: 8000 اینتل Corp.
bus ۰۰۱ device۰۰۱ : ID 1d6b: 0002 linux۲,۰ بنیاد hub root
bus ۰۰۳ device۰۰۱ : ID 1d6b: 0003 linux۳,۰ بنیاد hub root
Bus 002 device۰۰۵ : ID 0bda: b728 Realtek Semiconductor Corp.
bus ۰۰۲ device۰۰٤ : ID 5986: 0249 ایسر، وارز
bus ۰۰۲ device۰۰۲ : ID 045e: 00cb Microsoft Corp. Basic Optical Mouse v2.0
bus ۰۰۲ device۰۰۱ : ID 1d6b: 0002 linux پایه ۲,۰ root
```

می توانید از گزینه v-برای تولید اطلاعات دقیق در مورد هر دستگاه USB استفاده کنید.

```
hosein @ hosein ~ $ lsusb -v
```

بدست آوردن اطلاعات دستگاه هایPCI

دستگاه های PCI ممکن است شامل پورت هایUSB ، کارت گرافیک، آداپتورهای شبکه و غیره باشد .ابزارlspci برای تولید اطلاعات مربوط به تمامی کنترل کننده های PCI در system و همچنین دستگاه هایی که به آنها متصل است استفاده می شود.

برای چاپ اطلاعات در مورد دستگاه های PCI دستور زیر را اجرا کنید:

```
hosein @ hosein ~ $ lspci
```

از گزینه t-برای تولید خروجی در فرمت درخت استفاده کنید.

```
hosein @ hosein ~ $ lspci -t
```

```
- [0000: 00] - + - 00.0
+0.0.0
+0.0.0
+ -14.0
+ -16.0
```

از گزینه v-برای تولید اطلاعات دقیق در مورد هر دستگاه متصل استفاده کنید.

```
hosein @ hosein ~ $ lspci -v
```

```
00.0 :00: پل میزبان: کنترل کننده DRAM های Haswell-ULT شرکت(rev 0b) Intel
زیرsystem: دستگاه3978 Lenovo
سویچ ها bus master :، devsel fast، latency 0
```

بدست آوردن اطلاعات SCSI دستگاه

برای مشاهده تمام دستگاه های scsi / sata خود، از دستور lsscsiبه صورت زیر استفاده کنید.اگر ابزار lsscsiرا نصب نکرده اید ، دستور زیر را اجرا کنید تا آن را نصب کنید. متنهای داخل کروشه جز دستور نیست.

```
$ sudo apt install lsscsi          [در مشتقات دبیان]
# yum install lsscsi               [ بر روی سیستم های مبتنی بر ردهت]
# dnf install lsscsi               [در فدورا ۲۱ + بعد]
```

پس از install، دستور lsscsiرا اجرا کنید:

```
hosein @ hosein ~ $ lsscsi
```

```
[0: 0: 0: 0] disk ATA ST1000LM024 HN-M 2BA3 / dev / sda
[1: 0: 0: 0] cd/ dvd PLDS DVD-RW DA8A5SH RL61 / dev / sr0
[4: 0: 0: 0] disk Generic- xD / SD / MS 1.00 / dev / sdb
```

از گزینه s-برای نشان دادن اندازه دستگاه استفاده کنید.

```
hosein @ hosein ~ $ lsscsi -s
```

```
[0: 0: 0: 0] دیسک ATA ST1000LM024 HN-M 2BA3 / dev / sda 1.00TB
[1: 0: 0: 0] cd /دی وی دی PLDS DVD-RW DA8A5SH RL61 / dev / sr0 -
[4: 0: 0: 0] دیسک Generic- xD / SD / MS 1.00 / dev / sdb -
```

بدست آوردن اطلاعات در مورد دستگاه های SATA

با استفاده از ابزار hdparmمی توانید اطلاعاتی در مورد دستگاه های ساتا[۱] در system خود پیدا کنید.

```
hosein @ hosein ~ $ sudo hdparm / dev / sda1
```

بدست آوردن اطلاعات geometry

برای چاپ اطلاعات در مورد هندسه[۲] دستگاه از سیلندر، head، بخش، اندازه و offset شروع دستگاه، از گزینه g-استفاده کنید.

```
hosein @ hosein ~ $ sudo hdparm -g / dev / sda1
```

بدست آوردن اطلاعات سیستم فایل

برای جمع آوری اطلاعات در مورد پارتیشن های سیستم فایل، می‌توانید از دستور fdiskاستفاده کنید.اگر چه عملکرد اصلی دستور fdiskتغییر پارتیشن های سیستم فایل است، همچنین می‌توانید برای مشاهده اطلاعات مربوط به پارتیشن های مختلف در سیستم فایل خود استفاده کنید. می توانید اطلاعات پارتیشن را به صورت زیر چاپ کنید.فراموش نکنید که این دستور را به عنوان یک کاربر فوق کاربر اجرا کنید یا شاید هیچ خروجی را مشاهده نکنید.

```
hosein @ hosein ~ $ sudo fdisk -l
```

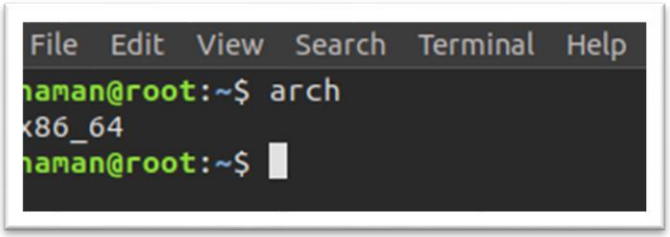
بدست آوردن معماری سیستم

از دستور arch برای چاپ معماری رایانه استفاده می شود .دستور Arch مواردی مانندi386" ، i486 ، i586 ، alpha ، arm، m68k، mips ، sparc ، x86_64و غیره را چاپ می کند.

ساختار دستور:

```
گزینه arch
```

مثال :

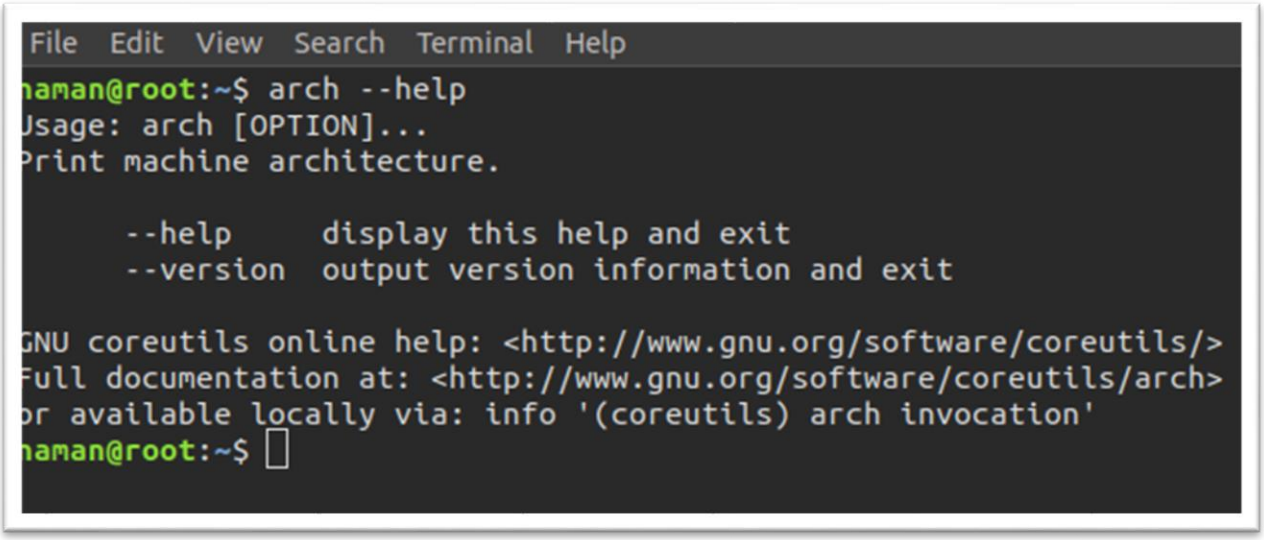


[۱] sata hard

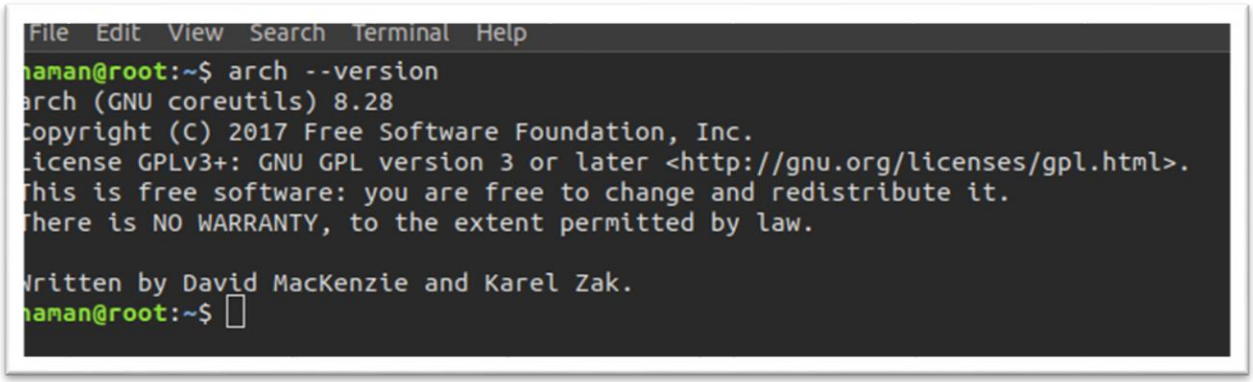
[۲] geometry

گزینه های دستور:

arch –help : اطلاعات راهنما را نمایش می دهد.



arch –version : اطلاعات نسخه را نمایش می دهد.



دستور acpid

مدیریت قدرت هوشمند بر روی یک سیستم.

ACPIپیکربندی پیشرفته و رابط قدرت به سیستم عامل این امکان را می دهد تا میزان نیرو را که برای هر دستگاه یا دستگاه جانبی متصل به سیستم فراهم می کند کنترل کند. این امر باعث می شود تا سیستم عامل هنگامی که دیگر از دستگاهها نظیر مانیتور استفاده نمی کنند ، خاموش باشد و این کار مدیریت پایدار و کارآمد را فراهم می کند acpid.بودن یک daemon به صورت پیش فرض به عنوان یک فرایند پس زمینه اجرا می شود و باید در هنگام بوت سیستم شروع شود .هر زمان که یک رویداد رخ می دهد برنامه هایی را برای مدیریت این رویداد اجرا می کند. وقتی یک رویداد دریافت می شود ، لیستی از قوانین را برای اجرای قانونی که با رویداد مطابقت دارد ، بررسی می کند. اگر یک پرونده قفل وجود داشته باشد (/ به طور پیش فرض var / lock / acpid) ، آن همه رویدادهای ورودی ACPI را نادیده می گیرد. از پرونده های پیکربندی ساده برای تعریف قوانین استفاده می شود. فهرست پیکربندی:در این مسیر / etc / acpi / events است.

دستور نصب:

```
sudo apt install acpid
```

از دستور **acpi** برای نمایش وضعیت باتری و سایر اطلاعات ACPI استفاده می شود .اطلاعات مربوط به سیستم پرونده */proc* / یا / *sys*مانند وضعیت باتری یا اطلاعات حرارتی را نمایش می دهد.

-b | –battery : اطلاعات باتری را نمایش می دهد.

مثال :

```
oindrila@LAPTOP-4L129038:/home/shubham$ acpi -b
Battery 0: Discharging, 0%, discharging at zero rate - will never fully discharge.
oindrila@LAPTOP-4L129038:/home/shubham$
```

-a | –ac-adapter: اطلاعات آداپتور ac را نمایش می دهد.

مثال :

```
oindrila@LAPTOP-4L129038:/home/shubham$ acpi -a
Adapter 0: off-line
oindrila@LAPTOP-4L129038:/home/shubham$
```

-t | –thermal: اطلاعات حرارتی را نشان می دهد.

مثال :

```
oindrila@LAPTOP-4L129038:/home/shubham$ acpi -t
No support for device type: thermal
oindrila@LAPTOP-4L129038:/home/shubham$
```

-c | –cooling: اطلاعات دستگاه خنک کننده را نمایش می دهد.

مثال :

```
oindrila@LAPTOP-4L129038:/home/shubham$ acpi -c
No support for device type: thermal
oindrila@LAPTOP-4L129038:/home/shubham$
```

-V | –everything: برای نشان دادن هر دستگاه استفاده می شود ، گزینه های فوق را لغو می کند.

مثال :

```
oindrila@LAPTOP-4L129038:/home/shubham$ acpi -v
acpi 1.7

Copyright (C) 2001 Grahame Bowland.
              2008-2012 Michael Meskes.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
oindrila@LAPTOP-4L129038:/home/shubham$
```

-s : دستگاههای غیر عملیاتی را نمایش می دهد.

مثال :

```
oindrila@LAPTOP-4L129O38:/home/shubham$ acpi -s
Battery 0: Discharging, 0%, discharging at zero rate - will never fully discharge.
oindrila@LAPTOP-4L129O38:/home/shubham$
```

-i | –details : جزئیات اضافی را در صورت وجود نمایش می دهد.

مثال :

```
oindrila@LAPTOP-4L129O38:/home/shubham$ acpi -i
Battery 0: Discharging, 0%, discharging at zero rate - will never fully discharge.
oindrila@LAPTOP-4L129O38:/home/shubham$
```

-f | –fahrenheit : از فارنهایت به عنوان واحد دما به جای درجه سانتیگراد پیش فرض استفاده می کند.

مثال :

```
oindrila@LAPTOP-4L129O38:/home/shubham$ acpi -f
Battery 0: Discharging, 0%, discharging at zero rate - will never fully discharge.
oindrila@LAPTOP-4L129O38:/home/shubham$
```

-k | –kelvin : از کلوین به عنوان واحد دما به جای درجه سانتیگراد پیش فرض استفاده می کند.

مثال :

```
oindrila@LAPTOP-4L129O38:/home/shubham$ acpi -k
Battery 0: Discharging, 0%, discharging at zero rate - will never fully discharge.
oindrila@LAPTOP-4L129O38:/home/shubham$
```

-p | –proc: از رابط قدیمی proc / استفاده می کند ، پیش فرض جدید sys / است.

-d | –directory : از مسیر دستیابی به اطلاعات ACPI (یا acpi / proc / یا class / sys) استفاده می کند

-h | –help: این راهنما و خروج را نمایش می دهد.

مثال :

```
oindrila@LAPTOP-4L129O38:/home/shubham$ acpi --help
Usage: acpi [OPTION]...
Shows information from the /proc filesystem, such as battery status or
thermal information.

  -b, --battery           battery information
  -i, --details           show additional details if available:
                            - battery capacity information
                            - temperature trip points
  -a, --ac-adapter        ac adapter information
  -t, --thermal           thermal information
  -c, --cooling           cooling information
  -V, --everything        show every device, overrides above options
  -s, --show-empty        show non-operational devices
  -f, --fahrenheit        use fahrenheit as the temperature unit
  -k, --kelvin            use kelvin as the temperature unit
  -d, --directory <dir>   path to ACPI info (/sys/class resp. /proc/acpi)
  -p, --proc              use old proc interface instead of new sys interface
  -h, --help              display this help and exit
  -v, --version           output version information and exit

By default, acpi displays information on installed system batteries.
Non-operational devices, for example empty battery slots are hidden.
The default unit of temperature is degrees celsius.

Report bugs to Michael Meskes <meskes@debian.org>.
oindrila@LAPTOP-4L129O38:/home/shubham$
```

version– | v- : اطلاعات نسخه خروجی و خروج را نشان می دهد.

مثال :

```
oindrila@LAPTOP-4L129038:/home/shubham$ acpi -version
acpi 1.7

Copyright (C) 2001 Grahame Bowland.
              2008-2012 Michael Meskes.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
oindrila@LAPTOP-4L129038:/home/shubham$
```

دستور acpi_available

دستور acpi_available در سیستم‌های لینوکس برای بررسی وجود پشتیبانی از ACPI (Advanced Configuration and Power Interface) در سیستم استفاده می‌شود. ACPI یک استاندارد برای مدیریت تنظیمات پیکربندی و انرژی سیستم است.

سه وضعیت را درخروجی نشان میدهد:

- **0** :این بدان معنی است که زیر سیستم ACPI در دسترس است.
- **1** :به این معنی است که زیر سیستم ACPI در دسترس نیست.
- **2** :این بدان معناست که برخی از خطاهای استفاده رخ داده است.

ساختار دستور:

```
$ acpi_available
```

گزینه‌های دستور :

بدون گزینه: نمایش وضعیت قابلیت دسترسی ACPI در سیستم.

```
acpi_available
```

خروجی دستور: ۱ (در صورتی که ACPI قابلیت دسترسی داشته باشد)

نمایش اطلاعات بیشتر درباره قابلیت دسترسی ACPI.

```
acpi_available -v
```

خروجی دستور: ACPI is available and supported

چک کردن قابلیت دسترسی ACPI و نمایش پیام مناسب.

```
acpi_available -c
```

خروجی دستور: ACPI is available on this system

نمایش راهنمای استفاده از دستور acpi_available.

```
acpi_available -h
```

خروجی دستور: راهنمای استفاده از دستور acpi_available

حالت سکوت و عدم نمایش هیچ پیامی در صورت عدم قابلیت دسترسی ACPI.

```
acpi_available -q
```

خروجی دستور: (بدون خروجی در صورت عدم قابلیت دسترسی ACPI)

نمایش پیام خطا اگر قابلیت دسترسی ACPI در سیستم موجود نباشد.

```
acpi_available -e
```

خروجی دستور:ACPI is not available on this system

مثال ۱ :در اینجا ،$? echo یک دستور است که برای دیدن آخرین وضعیت بازگشت استفاده می شود.

```
File  Edit  View  Search  Terminal  Help
naman@root:~$ acpi_available
naman@root:~$ echo $?
0
naman@root:~$
```

مثال ۲ :در اینجا ، ما یک خطای false ایجاد کرده و سپس وضعیت بازگشت را نمایش می دهیم.

```
~ acpid -help
Usage: acpid [OPTIONS]
 -c, --confdir      Set the configuration directory.
 -C, --clientmax    Set the limit on non-root socket connections.
 -d, --debug        Increase debugging level.
 -e, --eventfile    Use the specified file for events.
 -f, --foreground   Run in the foreground.
 -l, --logevents    Log all event activity.
 -g, --socketgroup  Set the group on the socket file.
 -m, --socketmode   Set the permissions on the socket file.
 -s, --socketfile   Use the specified socket file.
 -S, --nosocket     Do not listen on a UNIX socket (overrides -s).
 -p, --pidfile      Use the specified PID file.
 -L, --lockfile     Use the specified lockfile to stop processing.
 -n, --netlink      Force netlink/input layer mode. (overrides -e)
 -r, --dropaction   Define the pseudo-action to drop an event.
 -t, --tpmutefix    Fixup for ThinkPad mute-repeat behaviour.
 -v, --version      Print version information.
 -h, --help         Print this message.
```

کنترل سرویس acid

- برای کنترل سرویس acid از ابزار سرویس استفاده کنید:

```
service acid
```

```
~ service acpid
* Usage: /etc/init.d/acpid {start|stop|restart|reload|force-reload|status}
```

- **اجرا کردن سرویس:**

```
service acpid start
```

برای احراز هویت از شما خواسته می شود.

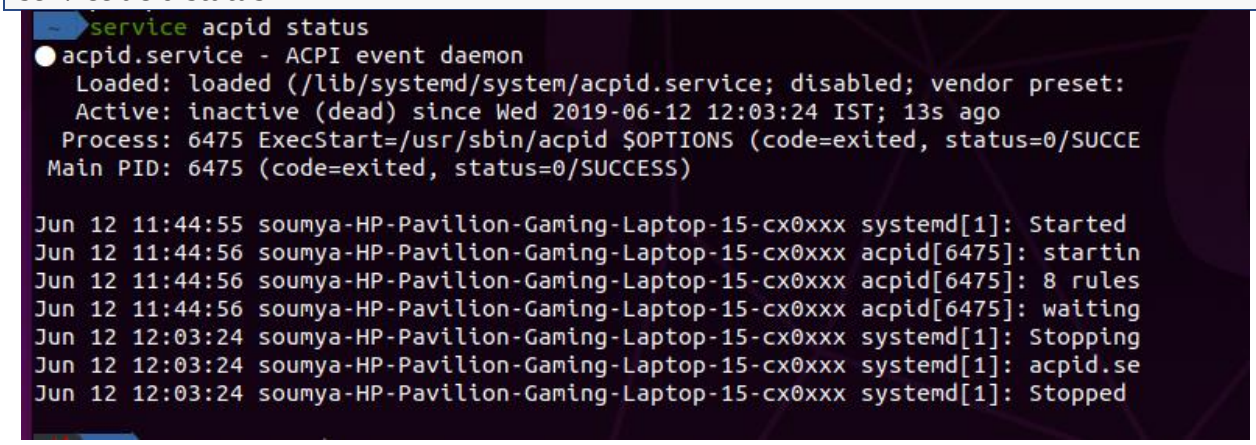
- **متوقف کردن سرویس:**

```
service acpid stop
```

```
✘ ~ service acpid stop
Warning: Stopping acpid.service, but it can still be activated by:
  acpid.path
```

- **بررسی سرویس که شروع یا متوقف شده است:**

```
service acid status
```



دستور dmesg

از دستور **dmesg** به عنوان "نمایش پیام" برای بررسی بافر حلقه هسته و چاپ بافر پیام هسته استفاده می شود .خروجی این دستور شامل پیام های تولید شده توسط درایورهای دستگاه است. وقتی کامپیوتر بوت می شود ، پیام های زیادی (ورود به سیستم) هنگام راه اندازی سیستم ایجاد می شود. بنابراین می توانید همه این پیام ها را با استفاده از دستور dmesg بخوانید .محتویات بافر حلقه هسته نیز در پرونده **var / log / dmesg** ذخیره می شود. دستور dmesg هنگامی که سیستم در هنگام راه اندازی با مشکلی روبرو می شود می تواند مفید باشد ، بنابراین با خواندن محتویات دستور dmesg می توانید واقعاً بفهمید که این مشکل در کجا رخ داده است (زیرا مراحل زیادی در توالی راه اندازی سیستم وجود دارد.

ساختار دستور:

```
dmesg [گزینه ها]
```

گزینه های دستور:

- -C – Clear :بافر حلقه را پاک میکند.
- -c –read-clear :بافر حلقه را پس از چاپ محتوای آن پاک میکند.
- -D –console-off :چاپ پیام برای کنسول را غیرفعال میکند.
- -E –console-on :چاپ پیام برای کنسول را فعال میکند.
- -F –file file :پیام های پرونده داده شده را میخواند.
- -h –help :نمایش متن راهنما.
- -k –kernel :پیام های هسته را چاپ میکند.
- -t –notime :مهر زمانهای هسته را چاپ نمیکند.
- -u –userspace :پیام های فضای کاربر را چاپ میکند.

از آنجا که خروجی **dmesg و** دستوربسیار بزرگ است، بنابراین برای بدست آوردن اطلاعات خاص در خروجی dmesg و است، بهتر است به دستور dmesg با less استفاده نمایید.

```
dmesg |less
```

یا

```
dmesg | grep " grep "text_to_search"
dmesg | grep "usb"
```

گزینه t-خروجی را با زمان مشخص می کند.

```
dmesg -t
```

مثال های دستور:

۱. گزینه C– یا clear:--

– این گزینه بافر حلقه (ring buffer)پیام‌های کرنل را پاک می‌کند.

```
dmesg -C
```

۲. گزینه c– یا read-clear:--

– این گزینه پیام‌های کرنل را چاپ می‌کند و سپس بافر حلقه را پاک می‌کند.

```
dmesg -c
```

۳. گزینه D– یا console-off:--

– این گزینه چاپ پیام‌ها برای کنسول را غیرفعال می‌کند.

```
dmesg -D
```

۴. گزینه E– یا console-on:--

– این گزینه چاپ پیام‌ها برای کنسول را فعال می‌کند.

```
dmesg -E
```

۵. گزینه F– یا file file:--

– این گزینه پیام‌های موجود در فایل مشخص شده را نمایش می‌دهد.

```
dmesg -F /var/log/dmesg.log
```

۶. گزینه h– یا help:--

– این گزینه راهنمای دستور dmesg را نمایش می‌دهد.

```
dmesg -h
```

۷. گزینه k– یا kernel:--

– این گزینه پیام‌های هسته را چاپ می‌کند.

```
dmesg -k
```

۸. گزینه t– یا notime:--

– این گزینه مهر زمان‌های هسته را در پیام‌ها چاپ نمی‌کند.

```
dmesg -t
```

۹. گزینه u– یا userspace:--

– این گزینه پیام‌های مربوط به فضای کاربر را چاپ می‌کند.

```
dmesg -u
```

دستور pmap

دستور pmap در سیستم‌عامل‌های Unix/Linux برای نمایش نقشه حافظه یک پردازه ((processاستفاده می‌شود. این دستور اطلاعات مربوط به استفاده از حافظه توسط یک پردازه را نمایش می‌دهد، از جمله نشان دادن محل بخش‌های مختلف حافظه که توسط پردازه مورد استفاده قرار می‌گیرد.

مثال‌های دستور:

نمایش نقشه حافظه یک پردازه با شناسه ((PIDخاص: این دستور برای نمایش نقشه حافظه یک پردازه با شناسه ((PIDمشخص استفاده می‌شود.

```
pmap 1234
```

نمایش نقشه حافظه به صورت خلاصه: این دستور برای نمایش نقشه حافظه یک پردازه به صورت خلاصه استفاده می‌شود.

```
pmap -q 1234
```

نمایش جزئیات بیشتر در مورد حافظه مصرفی: این دستور برای نمایش جزئیات بیشتر در مورد حافظه مصرفی توسط یک پردازه استفاده می‌شود.

```
pmap -X 1234
```

نمایش اطلاعات حافظه به صورت رقمی: این دستور برای نمایش اطلاعات حافظه یک پردازه به صورت رقمی (عددی) استفاده می‌شود.

```
pmap -n 1234
```

نمایش نقشه حافظه بدون نمایش اطلاعات اضافی: این دستور برای نمایش نقشه حافظه یک پردازه بدون نمایش اطلاعات اضافی استفاده می‌شود.

```
pmap -s 1234
```

با استفاده از دستور pmap می‌توانید نقشه حافظه یک پردازه را مشاهده کرده و اطلاعات مربوط به استفاده از حافظه توسط آن پردازه را دریافت کنید.

دستور addr2line

دستور addr2line در سیستم‌های لینوکس برای تبدیل آدرس حافظه ((memory addressبه اطلاعات خط و فایل مرتبط با آن آدرس در برنامه‌های اجرایی ((executablesاستفاده می‌شود. این دستور به توسعه‌دهندگان کمک می‌کند تا با استفاده از آدرس حافظه، خطاها و مشکلات مربوط به برنامه‌ها را ردیابی و رفع کنند.

گزینه‌های دستور:

بدون گزینه: تبدیل آدرس حافظه به اطلاعات خط و فایل.

```
addr2line 0x4005a1
```

خروجی دستور: main.c:25

مشخص کردن فایل اجرایی برای تبدیل آدرس حافظه.

```
addr2line -e my_program 0x7fffe0a7a1b4
```

خروجی دستور: function_name (source_file.c:123)

نمایش نام تابع (فانکشن) مرتبط با آدرس حافظه.

```
addr2line -f -e my_program 0x7fffe0a7a1b4
```

خروجی دستور: function_name

نمایش نام فایل منبع (source file)مرتبط با آدرس حافظه.

```
addr2line -s -e my_program 0x7fffe0a7a1b4
```

خروجی دستور: source_file.c

نمایش اطلاعات اضافی درباره آدرس حافظه.

```
addr2line -i -e my_program 0x7fffe0a7a1b4
```

خروجی دستور: main.c:25

نمایش نام توابع به صورت کامل با نشان دهنده‌های (demangled).

```
addr2line -C -e my_program 0x7fffe0a7a1b4
```

خروجی دستور: namespace::class::function_name

نمایش راهنمای استفاده از دستور addr2line.

```
addr2line -h
```

خروجی دستور: راهنمای استفاده از دستور addr2line

دستور free

میزان فضای استفاده شده، مبادله شده، ذخیره سازی، و غیره در system همچنین شامل فضای swap را نشان میدهد .از گزینه h-برای نمایش خروجی در فرمت قابل خواندن استفاده کنید.

```
$ free -h
```

در ادامه برخی از نمونه های مفیدی از دستورات "free" با گزینه ها را فراهم می کند که ممکن است مفید باشد برای استفاده بهتر از memory . هنگامی که هیچ گزینه ای استفاده نمی شود ، دستور free خروجی ستونی را مانند شکل بالا در ستون تولید می کند:

totalکل حافظه نصب شده را نمایش می دهد MemTotalو SwapTotal یعنی:

```
/ proc / meminfo
```

- **used**حافظه استفاده شده را نمایش می دهد.
- **آزاد** حافظه استفاده نشده را نمایش می دهد.
- **اشتراکی** حافظه مورد استفاده توسط tmpfs را نشان می دهد (Shmen یعنی در:
  - / proc / meminfo وجود دارد و درصورت عدم وجود صفر را نشان می دهد
- **buffers**حافظه مورد استفاده توسط بافرهای هسته را نمایش می دهد.
- **cached**حافظه مورد استفاده توسط حافظه پنهان صفحه و اسلب ها را نشان می دهد Cachedو Slab موجود در / proc / meminfo .

buffers / cache مجموع بافرها و حافظه پنهان را نمایش می دهد.

نمایش system memory

دستور free برای بررسی فضای مورد استفاده و در فضای دسترس از memory فیزیکی و swap memory درحالت KBاستفاده می شود.دستور زیر را در زیر ببینید.

```
# free
```

```
Mem: 1021628 912548 109080 0 120368 655548
بافر / کش: ۱۳۶۶۳۲ ۸۸۴۹۹۶ + / -
تعویض: ۴۱۹۴۲۹۶ ۰ ۴۱۹۴۲۹۶
```

نمایش memory درحالت Bytes

دستور free با گزینه b-نمایش اندازه memory در Bytes

```
# free -b
```

نمایش memory درKilo Bytes

دستور free با گزینه k-، اندازه memory را در KBکیلوبایت نشان می دهد.

```
# free-k
```

نمایش memory در مگابایت

برای مشاهده اندازه memory در مگابایت از گزینه به عنوان m-استفاده کنید.

```
# free -m
```

نمایش memory در گیگابایت

با استفاده از گزینه g-با دستور free، اندازه memory در GBگیگابایت نمایش داده می شود.

```
# free -g
```

نمایش مجموع حافظه

```
# free -t
```

غیرفعال کردن بافر

به طور پیش فرض، دستور free " buffer adjusted " را نشان میدهد، برای غیر فعال کردن این سویچ o- استفاده نمایید.

```
# free -o
```

نمایش وضعیت memory در زمان های مشخص

گزینه s– مورد استفاده برای بروزرسانی دستور free در فواصل زمانی منظم[1] است.به عنوان مثال ، دستور زیر دستور ree هر 5ثانیه را به روز می کند.

[1] period time

```
# free -s 5
```

شماره نسخه دستور free

گزینه V-، نمایش اطلاعات نسخه دستور free را نمایش می دهد.

```
# free -V
```

فصل یازدهم

فصل یازدهم

مانیتورینگ در لینوکس

دستور Vmstat

یک دستور مفید در سیستم‌های لینوکس است که اطلاعات مربوط به استفاده از منابع سیستم را نمایش می‌دهد. این دستور آمارهای مختلفی از سیستم شما ارائه می‌دهد که از جمله اطلاعات درباره حافظه فیزیکی، حافظهswap ، پردازنده و رودی/خروجی (I/O) می‌باشد. این اطلاعات می‌تواند به شما کمک کند تا عملکرد سیستم خود را بررسی کرده و منابع آن را بهینه‌تر مدیریت کنید.

توضیحاتی که vmstat ارائه می‌دهد شامل موارد زیر است:

- **Procs**نمایش آمارهای مربوط به پردازه‌ها از جمله تعداد پردازه‌های فعال، تعداد پردازه‌های در حال انتظار و تعداد پردازه‌های متوقف شده.
- **Memory** .نمایش آمارهای مربوط به حافظه فیزیکی و حافظه swap از جمله حافظه رزرو شده، حافظه آزاد، حافظه استفاده شده و تعداد صفحات swap
- **Swap** .نمایش آمارهای مربوط به فضای swap از جمله فضای swap آزاد، فضای swap استفاده شده و تعداد صفحات.swap
- **IO** .نمایش آمارهای ورودی/خروجی (I/O) از جمله تعداد خواندن و نوشتن به دیسک.
- **System** .نمایش آمارهای مربوط به سیستم از جمله تعداد تعاملات سیستمی (system) و تعداد تعاملات کاربری.(user)

نصب برنامه

ابتدا Sysstat را در install linux کنید

- vmstatخلاصه اطلاعات memory ، پردازش ، پیمایش و غیره
- iostatآمار پردازش مرکزی (CPU) و آمار ورودی / خروجی برای دستگاه ها و پارتیشن ها.

فهرست memory فعال و غیر فعال

```
$ sudo vmstat -a
```

اجرای vmstat برحسب تعداد و ثانیه

با استفاده از این دستور، vmstat هر دو ثانیه تا شش مرتبه اجرا می شود و پس از پایان طور خودکار متوقف می شود.

```
$ sudo vmstat 2 6
```

Vmstatبا برچسب های زمانی

دستور vmstatبا پارامتر t-نشانگرهای timestamps را نشان می دهد با هر خطی که در زیر نشان داده شده است.

```
[ hosein @ hosein ~] $ vmstat -t 1 5
```

آمار رویدادهای مختلف

دستور vmstatو s-سوئیچ نمایش خلاصه از رویدادهای رویداد های[1] مختلف و آمار memory است.

```
[ hosein @ hosein ~] $ vmstat -s
```

آمار دیسک

```
[ hosein @ hosein ~] $ vmstat -d
```

[1] events

نمایش آمار در حالت مگابایت

vmstatدر مگابایت با پارامترهای S–و m- نمایش داده می شود.به طور پیش فرض vmstatآمار در کیلوبایت را نمایش می دهد.

```
$ sudo vmstat -SM 1 5
```

دستور ss

شرح دستور:

دستور (socket statistics) ssابزاری برای نمایش اطلاعات مربوط به سوکت‌های شبکه در سیستم لینوکس است .این دستور جایگزین دستور netstatشده است و قابلیت‌های بیشتری را ارائه می‌دهد.

ساختار دستور:

```
ss [options]
```

- **options:**گزینه‌های مختلفی برای فیلتر کردن و نمایش اطلاعات سوکت‌ها.

کاربردهای دستور:

- نمایش اطلاعات کلی در مورد سوکت‌ها
- نمایش سوکت‌های در حال گوش‌دادن(listening sockets)
- نمایش اتصالات شبکه برقرار شده
- نمایش اطلاعات دقیق درباره سوکت‌ها) مانندUID ، inode، زمان‌سنج‌ها(
- نمایش سوکت‌های حافظه مشترک

گزینه‌های مهم دستور:

- **-h, --help:**نمایش کمک در مورد استفاده از دستور.
- **-V, --version:**نمایش نسخه دستور.
- **-H, --no-header:**حذف خط سرصفحه از خروجی.
- **-Q, --no-queues:**حذف ستون‌های صف ارسال و دریافت از خروجی.
- **-O, --oneline:**نمایش اطلاعات هر سوکت در یک خط.
- **-n, --numeric:**نمایش آدرس‌ها و پورت‌ها به صورت عددی بدون ترجمه به نام میزبان و سرویس.
- **-r, --resolve:**تلاش برای ترجمه آدرس‌های عددی به نام میزبان و سرویس.
- **-a, --all:**نمایش تمام سوکت‌ها (گوش‌دادن و برقرار شده).
- **-l, --listening:**نمایش فقط سوکت‌های در حال گوش‌دادن.
- **-B, --bound-inactive:**نمایش فقط سوکت‌های متصل شده اما غیرفعال (بدون گوش دادن یا اتصال).
- **-o, --options:**نمایش اطلاعات زمان‌سنج برای سوکت‌های.TCP
- **-e, --extended:**نمایش اطلاعات دقیق سوکت(UID ، inode، کوکی.)
- **-m, --memory:**نمایش اطلاعات حافظه استفاده شده توسط سوکت‌ها.

مثال‌ها:

مثال‌های کاربردی برای دستور ss

دستور ss ابزاری قدرتمند برای نمایش اطلاعات دقیق در مورد سوکت‌های شبکه است. در زیر، چندین مثال کاربردی برای گزینه‌های مختلف این دستور آورده شده است:

نمایش اطلاعات کلی

- **نمایش تمام سوکت‌ها:**

```
ss
```

- **نمایش نسخه دستور:**

```
ss -V
```

- **نمایش کمک:**

```
ss -h
```

فیلتر کردن بر اساس نوع سوکت

- **نمایش فقط سوکت‌های TCP:**

```
ss -t
```

- **نمایش فقط سوکت‌های UDP:**

```
ss -u
```

- **نمایش فقط سوکت‌های RAW:**

```
ss -w
```

نمایش وضعیت سوکت

- **نمایش سوکت‌های در حال گوش دادن:**

```
ss -l
```

- **نمایش تمام سوکت‌ها (گوش دادن و برقرار شده):**

```
ss -a
```

- **نمایش سوکت‌های متصل شده اما غیرفعال:**

```
ss -B
```

نمایش اطلاعات دقیق

- **نمایش اطلاعات دقیق سوکت‌ها (UID، inode، کوکی):**

```
ss -e
```

- **نمایش اطلاعات زمان‌سنج برای سوکت‌های TCP:**

```
ss -o
```

- **نمایش اطلاعات حافظه استفاده شده توسط سوکت‌ها:**

```
ss -m
```

قالب‌بندی خروجی

- **نمایش اطلاعات هر سوکت در یک خط:**

```
ss -O
```

- **حذف خط سرصفحه از خروجی:**

```
ss -H
```

- **حذف ستون‌های صف ارسال و دریافت:**

```
ss -Q
```

نمایش آدرس‌ها و پورت‌ها

- **نمایش آدرس‌ها و پورت‌ها به صورت عددی:**

```
ss -n
```

- **تلاش برای ترجمه آدرس‌های عددی به نام میزبان و سرویس:**

```
ss -r
```

مثال‌های ترکیبی

- **نمایش تمام سوکت‌های TCP و UDP به صورت عددی، همراه با اطلاعات دقیق و بدون سرصفحه:**

```
ss -tunpeH
```

- **نمایش سوکت‌های در حال گوش دادن روی پورت ۸۰، همراه با اطلاعات زمان‌سنج:**

```
ss -lt' sport = :80– 'o
```

نکات مهم:

- با ترکیب گزینه‌های مختلف، می‌توانید خروجی دستور ss را به طور دقیق تنظیم کنید.
- برای فیلتر کردن خروجی، می‌توانید از عبارات منظم استفاده کنید.
- دستور ss بسیار قدرتمند است و امکانات زیادی را برای تحلیل ترافیک شبکه فراهم می‌کند.
- برای اطلاعات بیشتر در مورد گزینه‌ها و خروجی دستور ss، می‌توانید از دستور man ss استفاده کنید.

مثال‌های پیشرفته‌تر:

- **یافتن اتصالات به یک سرور خاص:**

```
ss -tn' dst addr 192.168.1.100'
```

- **یافتن فرآیندهایی که از یک پورت خاص استفاده می‌کنند:**

```
ss -tulwnp' sport = :80'
```

- **بررسی اتصالات به یک سرویس خاص (مثلاً SSH):**

```
ss -tn' dport = :22'
```

با استفاده از این مثال‌ها و گزینه‌های مختلف دستور ss، می‌توانید اطلاعات جامعی در مورد وضعیت شبکه سیستم خود به دست آورید و مشکلات احتمالی را شناسایی و برطرف کنید.

دستور iptraf

شرح دستور:

iptrafیک ابزار خط فرمان است که برای نظارت بر ترافیک شبکه در سیستم‌های یونیکس‌مانند استفاده می‌شود .این ابزار با نمایش اطلاعاتی مانند میزان داده‌های ارسال شده و دریافت شده، پهنای باند مصرف شده و آدرس‌های IP مبدأ و مقصد، به شما کمک می‌کند تا ترافیک شبکه خود را تجزیه و تحلیل کنید.

ساختار دستور:

```
iptraf [options]
```

کاربردهای دستور:

- **نظارت بر ترافیک شبکه در زمان واقعی** iptraf **:**به شما امکان می‌دهد تا در حین انجام کارهای مختلف، میزان ترافیک شبکه را به صورت گرافیکی و عددی مشاهده کنید.
- **شناسایی برنامه‌ها و فرآیندهایی که بیشترین پهنای باند را مصرف می‌کنند :**با استفاده از iptraf می‌توانید به راحتی متوجه شوید که کدام برنامه یا سرویس در شبکه شما بیشترین ترافیک را ایجاد می‌کند.

- **تشخیص مشکلات شبکه** iptraf **:**به شما کمک می‌کند تا مشکلات شبکه مانند ترافیک زیاد، بسته‌های از دست رفته و ... را شناسایی کنید.
- **ایجاد گزارش‌های ترافیکی** iptraf **:**می‌تواند گزارش‌هایی از ترافیک شبکه را در قالب‌های مختلف ایجاد کند.

گزینه‌های مهم دستور:

- **-i interface:** مشخص کردن رابط شبکه برای نظارت) مثلاً eth0 ، .(wlan0
- **-d:** نمایش اطلاعات در قالب عددی.
- **-g:** نمایش اطلاعات به صورت گرافیکی.
- **-n:** نمایش آدرس‌های IP به صورت عددی.
- **-r:** نمایش آدرس‌های IP به صورت نام میزبان.
- **-w window:** تعیین اندازه پنجره نمایش داده‌ها (بر حسب ثانیه).
- **-c:** فعال کردن حالت شمارش بسته‌ها.
- **-s:** فعال کردن حالت ذخیره گزارش.
- **-h:** نمایش راهنمای استفاده.

مثال‌های کاربردی دستور iptraf

دستور iptraf ابزاری بسیار قدرتمند برای نظارت بر ترافیک شبکه است .با استفاده از گزینه‌های مختلف، می‌توانید اطلاعات دقیق و مفیدی درباره ترافیک شبکه خود به دست آورید .در ادامه، چندین مثال کاربردی برای گزینه‌های مختلف این دستور آورده شده است:

مثال‌های ساده

- **نمایش ترافیک کلی روی همه رابط‌های شبکه به صورت گرافیکی:**

```
iptraf -g
```

- **نمایش ترافیک روی رابط eth0 به صورت عددی:**

```
iptraf -i eth0 -d
```

- **نمایش ترافیک روی رابط wlan0 و ذخیره گزارش در فایل:traffic.log**

```
iptraf -i wlan0 -s traffic.log
```

مثال‌های پیشرفته‌تر

- **نمایش ترافیک روی همه رابط‌ها به صورت گرافیکی و با اندازه پنجره ۵ ثانیه:**

```
iptraf -g -w 5
```

- **نمایش ترافیک روی رابط eth0 و فیلتر کردن بر اساس پروتکل:TCP**

```
iptraf -i eth0 -f tcp
```

- **نمایش ترافیک روی رابط eth0 و ذخیره گزارش در فایل traffic.log با فرمت:CSV**

```
iptraf -i eth0 -s traffic.log -B -L traffic.csv
```

- **نمایش ترافیک روی همه رابط‌ها به صورت عددی و با نمایش آدرس‌های IP به صورت نام میزبان:**

```
iptraf -dr
```

- **نمایش ترافیک روی رابط eth0 و محاسبه آمار ترافیک برای هر ۱۰ ثانیه:**

```
iptraf -i eth0 -t 10
```

مثال‌های ترکیبی

- **نمایش ترافیک روی رابط eth0 و فیلتر کردن بر اساس پروتکل TCP و UDP:**

```
iptraf -i eth0 -f tcp,udp
```

- **نمایش ترافیک روی همه رابط‌ها به صورت گرافیکی و ذخیره گزارش در فایل traffic.log با فرمت CSV:**

```
iptraf -g -s traffic.log -B
```

نکات مهم

- **پروتکل‌ها iptraf :**از پروتکل‌های مختلفی مانندTCP ، UDP، ICMPو ... پشتیبانی می‌کند.
- **رابط‌های شبکه :**شما می‌توانید ترافیک روی هر یک از رابط‌های شبکه سیستم خود را نظارت کنید.
- **گزارش‌ها iptraf :**می‌تواند گزارش‌های مفیدی را در قالب‌های مختلف ایجاد کند که برای تجزیه و تحلیل ترافیک شبکه بسیار مفید است.
- **فیلترها :**با استفاده از گزینه‌های فیلتر کردن، می‌توانید ترافیک مورد نظر خود را به صورت دقیق‌تری بررسی کنید.

توجه :برای مشاهده لیست کامل گزینه‌ها و اطلاعات بیشتر، دستور `iptraf -h`را اجرا کنید.

دستور iostat

دستور iostat در لینوکس برای نظارت بر آمار ورودی/خروجی سیستم برای دستگاه ها و پارتیشن ها استفاده می شود. ورودی/خروجی سیستم را با مشاهده زمان فعال بودن دستگاه ها در رابطه با میانگین نرخ انتقال آنها نظارت میکند. گزارش های تولید iostat ممکن است برای تغییر پیکربندی سیستم به منظور بالا بردن تعادل ورودی/خروجی بین دیسک های فیزیکی استفاده شود. iostat در بسته sysstat قرارداده شده است.

ساختاردستور:

```
$ iostat [ -c ] [ -d ] [ -h ] [ -N ] [ -k | -m ] [ -t ] [ -V ] [ -x ] [ -z ] [ [ [ -T ] -g group_name ] { device [...] | ALL } ] [ -p [ device [,...] | ALL ] ] [ interval [ count ] ]
```

گزینه های دستور:

- **x – نمایش جزئیات بیشتر اطلاعات آمار.**
- c – فقط آمار cpu را نشان میدهد.
- d – فقط گزارش دستگاه را نمایش میدهد
- xd – آمار I/O توسعه یافته را فقط برای دستگاه نمایش میدهد.
- k – آمار را در کیلوبایت یا مگابایت ضبط میکند.
- k23 – نمایش آمار cpu و دستگاه با تاخیر.
- p – نمایش آمار برای دستگاه های بلوک.

-c :نمایش آمارهای متوسط مربوط به زمان بارگیری در طول زمان. مثال:

```
$ iostat -c
```

-d :نمایش آمارهای مربوط به دستگاه‌های ذخیره‌سازی. مثال:

```
$ iostat -d
```

-h :نمایش آمارها با خوانایی بهتر برای کاربر. مثال:

```
$ iostat -h
```

-N :نمایش نام دستگاه‌ها به جای شماره‌های مربوط به آن‌ها. مثال:

```
$ iostat -N
```

k-یا m– :نمایش آمارها با واحدهای کیلوبایت (KB) یا مگابایت .(MB) مثال:

```
$ iostat -k
```

t- :نمایش زمان در خروجی. مثال:

```
$ iostat -t
```

V- :نمایش نسخه iostat. مثال:

```
$ iostat -V
```

x-:نمایش آمارهای تفصیلی برای دستگاه‌های ذخیره‌سازی. مثال:

```
$ iostat -x
```

z- :نمایش آمارها برای دستگاه‌های غیرفعال. مثال:

```
$ iostat -z
```

T- نمایش زمان سیستم به جای زمان محلی. مثال:

```
$ iostat -T
```

g group_name- :نمایش آمارهای مربوط به گروه دستگاه‌ها. مثال:

```
$ iostat -g group_name
```

device [...] | ALL نمایش آمارها برای دستگاه‌های مشخص شده یا همه دستگاه‌ها. مثال:

```
$ iostat sda sdb
```

[device [,...] | ALL] p-نمایش آمارها برای دستگاه‌های ذخیره‌سازی و فرآیندهای مرتبط. مثال:

```
$ iostat -p sda
```

interval [count] مشخص کردن فاصله زمانی بین هر بروزرسانی و تعداد بروزرسانی‌ها. مثال:

```
$ iostat 5
```

دستور iw

دستور iw در لینوکس برای مدیریت رابط‌های شبکه بی‌سیم (Wi-Fi) استفاده می‌شود. این دستور به کاربر امکان می‌دهد تا اطلاعات مربوط به اتصالات Wi-Fi را مشاهده کرده، تنظیمات رابط‌های بی‌سیم را تغییر دهد و اطلاعات مربوط به شبکه‌های Wi-Fi موجود در دسترس را ببیند.

گزینه های دستور iw

```
iw [ OPTIONS ] { help [ command ] | OBJECT COMMAND }

OBJECT := { dev | phy | reg }

OPTIONS := { --version | --debug }
```

مثال های دستور iw

گزینه dev:

این گزینه برای مشخص کردن رابط شبکه بی‌سیم مورد نظر استفاده می‌شود.

```
iw dev wlan0 link
```

گزینه scan:

این گزینه برای اسکن شبکه‌های بی‌سیم در دسترس استفاده می‌شود.

```
iw dev wlan0 scan
```

گزینه connect:

این گزینه برای اتصال به یک شبکه بی‌سیم استفاده می‌شود.

```
iw dev wlan0 connect MyWiFiSSID
```

گزینه disconnect:

این گزینه برای قطع اتصال از یک شبکه بی‌سیم استفاده می‌شود.

```
iw dev wlan0 disconnect
```

گزینه info:

این گزینه اطلاعات کاملی از رابط بی‌سیم فعلی نمایش می‌دهد.

```
iw dev wlan0 info
```

گزینه event:

این گزینه برای نمایش رویدادهای مربوط به رابط بی‌سیم استفاده می‌شود.

```
iw dev wlan0 event
```

گزینه monitor:

این گزینه برای فعال کردن حالت مانیتور (monitor mode)برای رابط بی‌سیم استفاده می‌شود.

```
iw dev wlan0 set monitor control
```

گزینه frequency:

این گزینه برای تنظیم فرکانس بی‌سیم استفاده می‌شود.

```
iw dev wlan0 set freq 2462
```

گزینه txpower:

این گزینه برای تنظیم توان انتقال بی‌سیم استفاده می‌شود.

```
iw dev wlan0 set txpower fixed 20dBm
```

دستور iwlist

دستورiwlist نمایش اطلاعات بی سیم دقیق از یک رابط بی سیم را نشان می دهد.دستور زیر را قادر میسازد اطلاعات دقیق در مورد رابط wlp1s0دریافت کنید. دستور iwlist نیز مانند دستور iw برای مدیریت رابطهای شبکه بی‌سیم استفاده می‌شود. این دستور به کاربر امکان می‌دهد تا اطلاعات مختلفی از رابطهای بی‌سیم را مشاهده کند از جمله اسکن شبکه‌ها، فرکانس بی‌سیم، نرخ انتقال داده، کلیدهای امنیتی، قدرت انتقال، توان انتقال، تلاش مجدد برای ارتباط، رویدادها، احراز هویت، کلیدهای WPA، مودولاسیون و ...

گزینه های دستور:

```
iwlist [interface] scanning
iwlist [interface] frequency
iwlist [interface] rate
iwlist [interface] keys
iwlist [interface] power
iwlist [interface] txpower
iwlist [interface] retry
iwlist [interface] event
iwlist [interface] auth
iwlist [interface] wpakeys
iwlist [interface] genie
```

```
iwlist [interface] modulation
iwlist --help
iwlist –version
```

مثال های دستور iwlist

۱. گزینه scanning:

– این گزینه برای اسکن شبکه‌های بی‌سیم در دسترس استفاده می‌شود.

```
iwlist wlan0 scanning
```

۲. گزینه frequency:

– این گزینه برای نمایش فرکانس‌های بی‌سیم مورد حمایت توسط رابط استفاده می‌شود.

```
iwlist wlan0 frequency
```

۳. گزینه rate:

– این گزینه برای نمایش نرخ انتقال داده بی‌سیم استفاده می‌شود.

```
iwlist wlan0 rate
```

۴. گزینه keys:

– این گزینه برای نمایش کلیدهای امنیتی بی‌سیم استفاده می‌شود.

```
iwlist wlan0 keys
```

۵. گزینه power:

– این گزینه برای نمایش اطلاعات مربوط به قدرت انتقال بی‌سیم استفاده می‌شود.

```
iwlist wlan0 power
```

۶. گزینه txpower:

– این گزینه برای نمایش توان انتقال بی‌سیم استفاده می‌شود.

```
iwlist wlan0 txpower
```

۷. گزینه retry:

– این گزینه برای نمایش تلاش مجدد برای ارتباط بی‌سیم استفاده می‌شود.

```
iwlist wlan0 retry
```

۸. گزینه event:

– این گزینه برای نمایش رویدادهای مربوط به رابط بی‌سیم استفاده می‌شود.

```
iwlist wlan0 event
```

۹. گزینه auth:

– این گزینه برای نمایش اطلاعات مربوط به احراز هویت بی‌سیم استفاده می‌شود.

```
iwlist wlan0 auth
```

۱۰. گزینه wpakeys:

– این گزینه برای نمایش کلیدهای WPA بی‌سیم استفاده می‌شود.

```
iwlist wlan0 wpakeys
```

۱۱. گزینه genie:

– این گزینه برای نمایش اطلاعات Genie بی‌سیم استفاده می‌شود.

```
iwlist wlan0 genie
```

۱۲. گزینه modulation:

– این گزینه برای نمایش اطلاعات مربوط به مودولاسیون بی‌سیم استفاده می‌شود.

```
iwlist wlan0 modulation
```

۱۳. گزینه help:--

– این گزینه برای نمایش راهنمای دستور iwlist استفاده می‌شود.

```
iwlist --help
```

۱۴. گزینه version:--

– این گزینه برای نمایش نسخه دستور iwlist استفاده می‌شود.

```
iwlist --version
```

دستور iwconfig

برای تنظیمات و مدیریت رابط‌های شبکه بی‌سیم استفاده می‌شود. این دستور به کاربر امکان می‌دهد تا تنظیمات مختلفی را برای رابط بی‌سیم خود اعمال کند از جمله تنظیمESSID ، فرکانس، حالت، قدرت انتقال، رمزنگاری، قدرت انتقال، نرخ انتقال داده، و...

گزینه های دستور :

```
iwconfig [interface]
iwconfig interface [essid X] [nwid N] [mode M] [freq F]
[channel C][sens S ][ap A ][nick NN ]
[rate R] [rts RT] [frag FT] [txpower T]
[enc E] [key K] [power P] [retry R]
[modu M] [commit]
iwconfig --help
iwconfig --version
```

مثال های دستور:

گزینه essid X:

این گزینه برای تنظیم نام شبکه ESSIDبی‌سیم استفاده می‌شود.

```
iwconfig wlan0 essid MyNetwork
```

گزینه nwid N:

این گزینه برای تنظیم شناسه شبکه ( )NWIDبی‌سیم استفاده می‌شود.

```
iwconfig wlan0 nwid NetworkID
```

گزینه mode M:

این گزینه برای تنظیم حالت بی‌سیم ( )Modeاستفاده می‌شود.

```
iwconfig wlan0 mode Managed
```

گزینه freq F:

این گزینه برای تنظیم فرکانس بی‌سیم استفاده می‌شود.

```
iwconfig wlan0 freq 2.4G
```

گزینه channel C:

این گزینه برای تنظیم کانال بی‌سیم استفاده می‌شود.

```
iwconfig wlan0 channel 6
```

گزینه rate R:

این گزینه برای تنظیم نرخ انتقال داده بی‌سیم استفاده می‌شود.

```
iwconfig wlan0 rate 54M
```

گزینه txpower T:

این گزینه برای تنظیم توان انتقال بی‌سیم استفاده می‌شود.

```
iwconfig wlan0 txpower 20dBm
```

گزینه enc E:

این گزینه برای تنظیم رمزنگاری بی‌سیم استفاده می‌شود.

```
iwconfig wlan0 enc WPA2
```

گزینه key K:

این گزینه برای تنظیم کلید امنیتی بی‌سیم استفاده می‌شود.

```
iwconfig wlan0 key s:MyPassword
```

گزینه power P:

این گزینه برای تنظیم قدرت انتقال بی‌سیم استفاده می‌شود.

```
iwconfig wlan0 power on
```

گزینه retry R:

این گزینه برای تنظیم تلاش مجدد برای ارتباط بی‌سیم استفاده می‌شود.

```
iwconfig wlan0 retry 3
```

گزینه modu M:

این گزینه برای تنظیم مودولاسیون بی‌سیم استفاده می‌شود.

```
iwconfig wlan0 modu auto
```

گزینه commit:

این گزینه برای اعمال تنظیمات انجام شده استفاده می‌شود.

```
iwconfig wlan0 commit
```

گزینه help:--

این گزینه برای نمایش راهنمای دستور iwconfig استفاده می‌شود.

```
iwconfig --help
```

گزینه version:--

این گزینه برای نمایش نسخه دستور iwconfig استفاده می‌شود.

```
iwconfig --version
```

دستور iwevent

برای مشاهده و نمایش رویدادهای مربوط به رابط‌های شبکه بی‌سیم استفاده می‌شود. این دستور به کاربر این امکان را می‌دهد تا تغییرات و رویدادهایی که در ارتباط با رابط بی‌سیم رخ می‌دهد را مشاهده کند.

گزینه‌های دستور:

گزینه c- یا channel:--

این گزینه برای نمایش رویدادهای مربوط به تغییر کانال بی‌سیم استفاده می‌شود.

```
iwevent -c
```

گزینه r- یا remove:--

این گزینه برای نمایش رویدادهای مربوط به حذف رابط بی‌سیم استفاده می‌شود.

```
iwevent -r
```

گزینه a- یا all:--

این گزینه برای نمایش همه رویدادها، حتی رویدادهای کم اهمیت، استفاده می‌شود.

```
iwevent -a
```

گزینه v- یا verbose:--

این گزینه برای نمایش جزئیات بیشتر درباره رویدادها استفاده می‌شود.

```
iwevent -v
```

گزینه help:--

این گزینه برای نمایش راهنمای دستور iwevent استفاده می‌شود.

```
iwevent --help
```

گزینه version:--

این گزینه برای نمایش نسخه دستور iwevent استفاده می‌شود.

```
iwevent --version
```

دستور chrt

دستور chrt در لینوکس برای تغییر و یا دریافت ویژگی‌های زمان‌بندی واقعی (real-time scheduling attributes) یک فرآیند (processاستفاده می‌شود. این دستور به شما اجازه می‌دهد که سیاست زمان‌بندی (scheduling policyو اولویت (priorityیک فرآیند موجود را تنظیم یا بازیابی کنید، یا یک دستور را با ویژگی‌های داده شده اجرا کنید.

ساختار دستور:

```
$ chrt [options] priority command [argument ...]
$ chrt [options] -p [priority] pid
```

گزینه های دستور : (خط مش=قانون وسیاست)

- **b-، batch–** :برای تنظیم خط مشی روی SCHED_BATCH استفاده می شود.
- **d-، deadline–** :برای تنظیم خط مشی روی SCHED_DEADLINE استفاده می شود.
- **f-، fifo–** :برای تنظیم خط مشی روی SCHED_FIFO استفاده می شود.
- **i-، idle–** :برای تنظیم خط مشی روی SCHED_IDLE استفاده می شود.
- **o-، other–** :برای تنظیم خط مشی روی SCHED_OTHER استفاده می شود.
- **r-، rr–** :برای تنظیم خط مشی روی(SCHED_RR پیش فرض) استفاده می شود.

گزینه های برنامه ریزی:

- **SCHED_BATCH** :از الگوریتم زمانبندی فرآیندهای دسته ای استفاده میکند.
- **SCHED_FIFO**:از اولین الگوریتم زمان بندی اولین بار استفاده می کند .این روش برنامه ریزی در سیستم های دسته ای استفاده می شود ، غیرقابل پیش بینی است .این فقط یک صف اجرا می کند که وظایف را به ترتیب انجام آنها حفظ میکند.
- **SCHED_IDLE** :برای اجرای مشاغل پس زمینه با اولویت بسیار کم استفاده می شود.
- **SCHED_OTHER** :از الگوریتم زمان بندی پیش فرض برنامه ریزی زمانبندی لینوکس یا به سادگی از خط مشی استاندارد به اشتراک گذاری زمانRound Robin استفاده میکند.
- **SCHED_RR از** الگوریتم زمان بندی Round Robin استفاده می کند و اگر مشخص نشده باشد به عنوان الگوریتم پیش فرض استفاده می شود .این الگوریتمی است که برای برنامه ریزی پیش بینی شده استفاده می شود.

گزینه های دستور:

- **a-، all-Tasks–** :برای کارکردن همه وظایف (رشته ها) برای یک PID مشخص استفاده می شود.
- **m-، max–** :برای نشان دادن حداقل و حداکثر اولویت های معتبر استفاده می شود.
- **p-، pid–** :بر روی pid موجود داده عمل میکند.
- **v-، verbose–** :برای نشان دادن اطلاعات وضعیت استفاده می شود.
- **h-، help–** :برای نشان دادن پیام راهنما و خروج استفاده می شود.
- **v-، version–** :برای نشان دادن اطلاعات نسخه و خروج استفاده می شود.

مثال های دستور:

- **برای دیدن خط مشی برنامه ریزی فعلی** :ابتدا باید فرآیندی ایجاد کنیم .بیایید مثال ی بزنیم ، Firefoxدر حال اجرا است و برای یافتن pid خود دستور زیر را اجرا می کنیم:

```
$ pidof -s firefox
```

در مورد من PID 5794 است ، این تصویر است

```
linux@ubuntu:~$ pidof -s firefox
5794
linux@ubuntu:~$
```

اکنون برای بازیابی خط مشی برنامه ریزی فعلی و اولویت فرآیند firefox ، از chrt به روش زیر استفاده کنید:

```
$ chrt -p 5794
```

```
linux@ubuntu:~$ chrt -p 5794
pid 5794's current scheduling policy: SCHED_OTHER
pid 5794's current scheduling priority: 0
linux@ubuntu:~$
```

- **برای تغییر سیاست زمانبندی به SCHED_FIFO:** از مثال فوق ، سیاست زمانبندی فرآیند firefox به عنوان SCHED_OTHERتنظیم شده است .اکنون برای تغییر خط مشی به SCHED_FIFO می توانیم از دستور زیر استفاده کنیم:

```
$ sudo chrt -f -p 5794
```

```
linux@ubuntu:~$ sudo chrt -f -p 1 5794
linux@ubuntu:~$ chrt -p 5794
pid 5794's current scheduling policy: SCHED_FIFO
pid 5794's current scheduling priority: 1
linux@ubuntu:~$
```

- **برای تغییر سیاست برنامه ریزی SCHED_BATCH:** از مثال فوق ، سیاست زمانبندی فرآیند firefox به عنوان SCHED_FIFOتنظیم شده است .اکنون برای تغییر خط مشی به SCHED_BATCH می توانیم از دستور زیر استفاده کنیم:

```
$ sudo chrt -b -p 5794
```

```
linux@ubuntu:~$ sudo chrt -b -p 0 5794
linux@ubuntu:~$ chrt -p 5794
pid 5794's current scheduling policy: SCHED_BATCH
pid 5794's current scheduling priority: 0
linux@ubuntu:~$
```

- **برای دیدن حداکثر و حداقل اولویت های معتبر:** این کار را می توان با استفاده از گزینه خط فرمان m- ذکر شده در سیاست chrt انجام داد.

```
$ chrt -m
```

```
linux@ubuntu:~$ chrt -m
SCHED_OTHER min/max priority    : 0/0
SCHED_FIFO min/max priority     : 1/99
SCHED_RR min/max priority       : 1/99
SCHED_BATCH min/max priority    : 0/0
SCHED_IDLE min/max priority     : 0/0
SCHED_DEADLINE min/max priority : 0/0
linux@ubuntu:~$
```

گزینه b– یا batch:--

برای تنظیم سیاست به SCHED_BATCH.

```
chrt -b <PID>
```

گزینه -d یا --deadline:
برای تنظیم سیاست به SCHED_DEADLINE.

```
chrt -d <PID>
```

گزینه -f یا --fifo:
برای تنظیم سیاست به SCHED_FIFO.

```
chrt -f <PID>
```

گزینه -i یا --idle:
برای تنظیم سیاست به SCHED_IDLE.

```
chrt -i <PID>
```

گزینه -o یا --other:
برای تنظیم سیاست به SCHED_OTHER.

```
chrt -o <PID>
```

گزینه -r یا --rr:
برای تنظیم سیاست به (SCHED_RRپیش‌فرض).

```
chrt -r <PID>
```

گزینه -a یا --all-tasks:
برای عملکرد بر روی تمام وظایف ((threadsبرای یک PID داده شده.

```
chrt -a <PID>
```

گزینه -m یا --max:
برای نمایش حداقل و حداکثر اولویت‌های معتبر.

```
chrt -m
```

گزینه -p یا --pid:
عملکرد بر روی PID موجود.

```
chrt -p <PID>
```

گزینه -v یا --verbose:
برای نمایش اطلاعات وضعیت.

```
chrt -v
```

گزینه -h یا --help:

برای نمایش پیام راهنما و خروج.

```
chrt -h
```

گزینه v– یا version:--

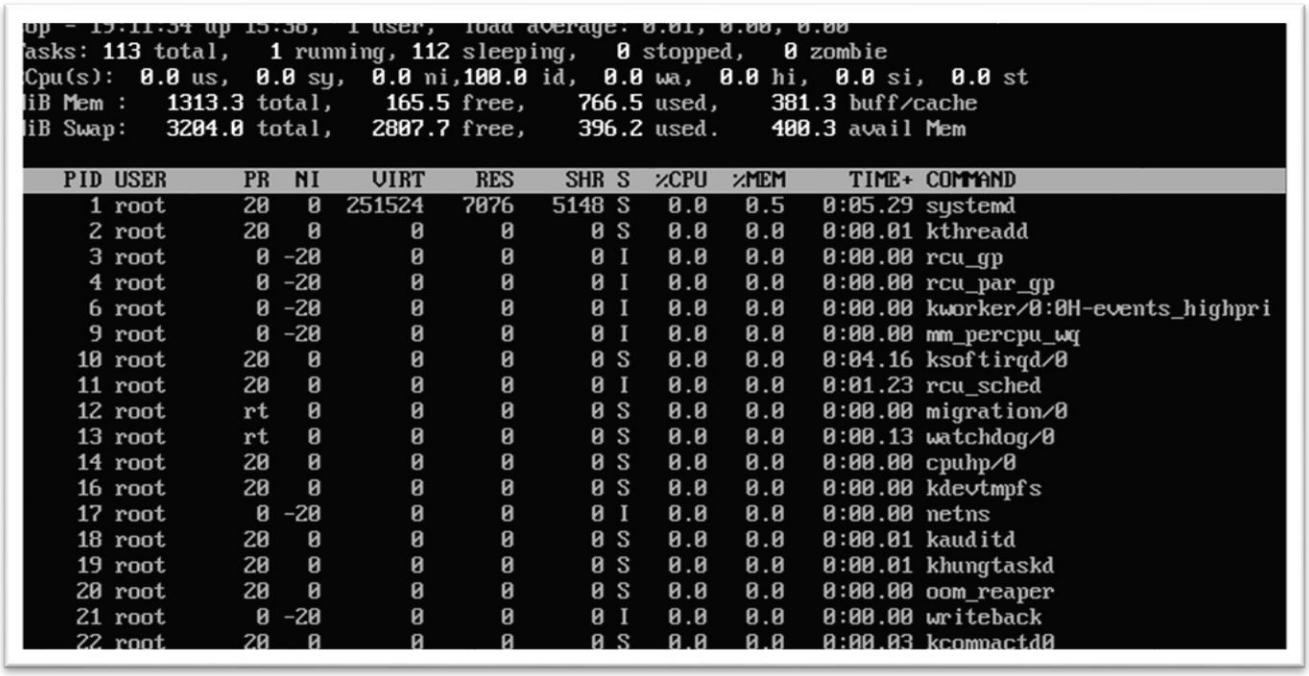
برای نمایش اطلاعات نسخه و خروج.

```
chrt -v
```

دستور top

برای فهرست کردن تمامی فرآیندهای لینوکس در حال اجرا، کافی است بالای خط فرمان را تایپ کنید تا اطلاعات مربوط به وظایف در حال اجرا ، حافظه ، cpuو swapرا دریافت کنید . برای خروج از پنجره q را فشار دهید.

```
# top
```



مشاهده فرآیندهای در حال اجرا لینوکس

مرتب سازی فرآیندهای لینوکس را بر اساس PID

برای مرتب کردن تمام فرآیندهای در حال اجرا لینوکس بر اساس شناسه فرآیند، فشار دهید Mو Tکلیدها را فشار دهید .

```
top - 19:23:26 up 15:50,  1 user,  load average: 0.00, 0.01, 0.00
Tasks: 113 total,   1 running, 112 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.3 hi,  0.0 si,  0.0 st
MiB Mem :   1313.3 total,    165.4 free,    766.6 used,    381.4 buff/cache
MiB Swap:   3204.0 total,   2807.7 free,    396.2 used.    400.3 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1681 mysql     20   0 1305848 132592   8964 S   0.0   9.9   8:02.56 mysqld
  13239 nginx     20   0  373980  41108   8020 S   0.0   3.1   0:30.91 php-fpm
  13236 nginx     20   0  376036  53704   8152 S   0.0   4.0   0:30.23 php-fpm
  13237 nginx     20   0  376492  61684   9416 S   0.0   4.6   0:29.39 php-fpm
  13238 nginx     20   0  378044  60392   8492 S   0.0   4.5   0:28.63 php-fpm
  13235 nginx     20   0  371776  47184   8008 S   0.0   3.5   0:27.62 php-fpm
  13241 nginx     20   0  372228  54304   8468 S   0.0   4.0   0:27.31 php-fpm
  13291 nginx     20   0  376272  44784   7440 S   0.0   3.3   0:26.05 php-fpm
  13376 nginx     20   0  373820  30100   7368 S   0.0   2.2   0:22.45 php-fpm
  13378 nginx     20   0  367724  32672   9636 S   0.0   2.4   0:21.57 php-fpm
  13379 nginx     20   0  374308  45656   8084 S   0.0   3.4   0:21.17 php-fpm
  13377 nginx     20   0  369804  42532   7412 S   0.0   3.2   0:20.78 php-fpm
     33 root       0 -20       0      0      0 I   0.0   0.0   0:19.87 kworker/0:1H-kblockd
  13374 nginx     20   0  373824  60032  10552 S   0.0   4.5   0:19.67 php-fpm
  13372 nginx     20   0  367704  35684   7880 S   0.0   2.7   0:19.44 php-fpm
    814 root      20   0  228112   7688   5868 S   0.0   0.6   0:10.96 sssd_be
    846 root      20   0  416596   6224   2632 S   0.0   0.5   0:08.79 tuned
  13228 nginx     20   0  163584   6348   3624 S   0.0   0.5   0:08.55 nginx
    818 root      20   0  229136  34540  33164 S   0.0   2.6   0:05.49 sssd_nss
      1 root      20   0  251524   7076   5148 S   0.0   0.5   0:05.34 systemd
    563 root      20   0       0      0      0 S   0.0   0.0   0:05.14 xfsaild/dm-0
     49 root      20   0       0      0      0 S   0.0   0.0   0:04.59 kswapd0
     10 root      20   0       0      0      0 S   0.0   0.0   0:04.19 ksoftirqd/0
    875 root      20   0  226712   3732   2312 S   0.0   0.3   0:04.12 rsyslogd
    442 root     -51   0       0      0      0 S   0.0   0.0   0:03.40 irq/18-vmwgfx
  13233 root      20   0  238448   5936   2856 S   0.0   0.4   0:03.31 php-fpm
     24 root      39  19       0      0      0 S   0.0   0.0   0:03.19 khugepaged
    802 dbus      20   0   76740   2960   2280 S   0.0   0.2   0:02.96 dbus-daemon
    830 root      20   0  396344   7156   5816 S   0.0   0.5   0:01.54 NetworkManager
  15285 root      20   0   65416   4456   3804 R   0.0   0.3   0:01.54 top
```

لیست فرآیندهای لینوکس بر اساس PID

مرتب سازی فرآیندهای لینوکس بر اساس حافظه و استفاده ازCPU

برای مرتب کردن تمام فرآیندهای در حال اجرا لینوکس بر اساس میزان مصرف حافظه، فشار دهید Mو Pکلیدها را فشار دهید .

```
top - 19:28:56 up 15:56,  1 user,  load average: 0.00, 0.00, 0.00
Tasks: 114 total,   3 running, 111 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.3 hi,  0.0 si,  0.0 st
MiB Mem :   1313.3 total,    165.4 free,    766.6 used,    381.4 buff/cache
MiB Swap:   3204.0 total,   2807.7 free,    396.2 used.    400.3 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  15291 root      20   0   65416   4500   3844 R   0.3   0.3   0:00.04 top
      1 root      20   0  251524   7076   5148 S   0.0   0.5   0:05.35 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
      9 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     10 root      20   0       0      0      0 S   0.0   0.0   0:04.20 ksoftirqd/0
     11 root      20   0       0      0      0 I   0.0   0.0   0:01.24 rcu_sched
     12 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/0
     13 root      rt   0       0      0      0 S   0.0   0.0   0:00.13 watchdog/0
     14 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
     17 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
     18 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kauditd
     19 root      20   0       0      0      0 S   0.0   0.0   0:00.01 khungtaskd
     20 root      20   0       0      0      0 S   0.0   0.0   0:00.00 oom_reaper
     21 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 writeback
     22 root      20   0       0      0      0 S   0.0   0.0   0:00.03 kcompactd0
     23 root      25   5       0      0      0 S   0.0   0.0   0:00.00 ksmd
     24 root      39  19       0      0      0 S   0.0   0.0   0:03.20 khugepaged
     25 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 crypto
     26 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kintegrityd
     27 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kblockd
     28 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 blkcg_punt_bio
     29 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 tpm_dev_wq
     30 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 md
     31 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 edac-poller
     32 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 watchdogd
     33 root       0 -20       0      0      0 I   0.0   0.0   0:19.87 kworker/0:1H-kblockd
```

مرتب سازی فرآیندهای لینوکس را بر اساس میزان استفاده از CPU حافظه

فرآیندهای لینوکس را بر اساس زمان اجرا مرتب کنید

برای مرتب کردن تمام فرآیندهای در حال اجرا لینوکس بر اساس زمان اجرا، فشار دهید Mو Tکلیدها را فشار دهید .

```
top - 19:31:31 up 15:58,  1 user,  load average: 0.00, 0.00, 0.00
Tasks: 114 total,   1 running, 113 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   1313.3 total,    165.4 free,    766.6 used,    381.4 buff/cache
MiB Swap:   3204.0 total,   2807.7 free,    396.2 used.    400.3 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1681 mysql     20   0 1305848 132592   8964 S   0.0   9.9   8:02.73 mysqld
  13239 nginx     20   0  373980  41108   8020 S   0.0   3.1   0:30.91 php-fpm
  13236 nginx     20   0  376036  53704   8152 S   0.0   4.0   0:30.23 php-fpm
  13237 nginx     20   0  376492  61684   9416 S   0.0   4.6   0:29.39 php-fpm
  13238 nginx     20   0  378044  60392   8492 S   0.0   4.5   0:28.63 php-fpm
  13235 nginx     20   0  371776  47184   8008 S   0.0   3.5   0:27.62 php-fpm
  13241 nginx     20   0  372228  54304   8468 S   0.0   4.0   0:27.31 php-fpm
  13291 nginx     20   0  376272  44784   7440 S   0.0   3.3   0:26.05 php-fpm
  13376 nginx     20   0  373820  30100   7368 S   0.0   2.2   0:22.45 php-fpm
  13378 nginx     20   0  367724  32672   9636 S   0.0   2.4   0:21.57 php-fpm
  13379 nginx     20   0  374308  45656   8084 S   0.0   3.4   0:21.17 php-fpm
  13377 nginx     20   0  369804  42532   7412 S   0.0   3.2   0:20.78 php-fpm
     33 root       0 -20       0      0      0 I   0.0   0.0   0:19.87 kworker/0:1H-kblockd
  13374 nginx     20   0  373824  60032  10552 S   0.0   4.5   0:19.67 php-fpm
  13372 nginx     20   0  367704  35684   7880 S   0.0   2.7   0:19.44 php-fpm
    814 root      20   0  228112   7688   5868 S   0.0   0.6   0:10.96 sssd_be
    846 root      20   0  416596   6224   2632 S   0.0   0.5   0:08.86 tuned
  13228 nginx     20   0  163584   6348   3624 S   0.0   0.5   0:08.55 nginx
    818 root      20   0  229136  34540  33164 S   0.0   2.6   0:05.49 sssd_nss
      1 root      20   0  251524   7076   5148 S   0.0   0.5   0:05.36 systemd
    563 root      20   0       0      0      0 S   0.0   0.0   0:05.16 xfsaild/dm-0
     49 root      20   0       0      0      0 S   0.0   0.0   0:04.59 kswapd0
     10 root      20   0       0      0      0 S   0.0   0.0   0:04.20 ksoftirqd/0
    875 root      20   0  226712   3732   2312 S   0.0   0.3   0:04.15 rsyslogd
    442 root     -51   0       0      0      0 S   0.0   0.0   0:03.40 irq/18-vmwgfx
  13233 root      20   0  238448   5936   2856 S   0.0   0.4   0:03.35 php-fpm
     24 root      39  19       0      0      0 S   0.0   0.0   0:03.20 khugepaged
    802 dbus      20   0   76740   2960   2280 S   0.0   0.2   0:02.98 dbus-daemon
    830 root      20   0  396344   7156   5816 S   0.0   0.5   0:01.55 NetworkManager
     11 root      20   0       0      0      0 I   0.0   0.0   0:01.24 rcu_sched
```

مرتب سازی فرآیندهای لینوکس بر اساس زمان اجرا

نمایش فرآیندهای لینوکس توسط کاربر خاص

برای نمایش تمام اطلاعات فرآیندهای در حال اجرا برای کاربر، از u-گزینه لیست جزئیات فرآیند کاربر خاص استفاده کنید.

```
# top –u predator-os
```

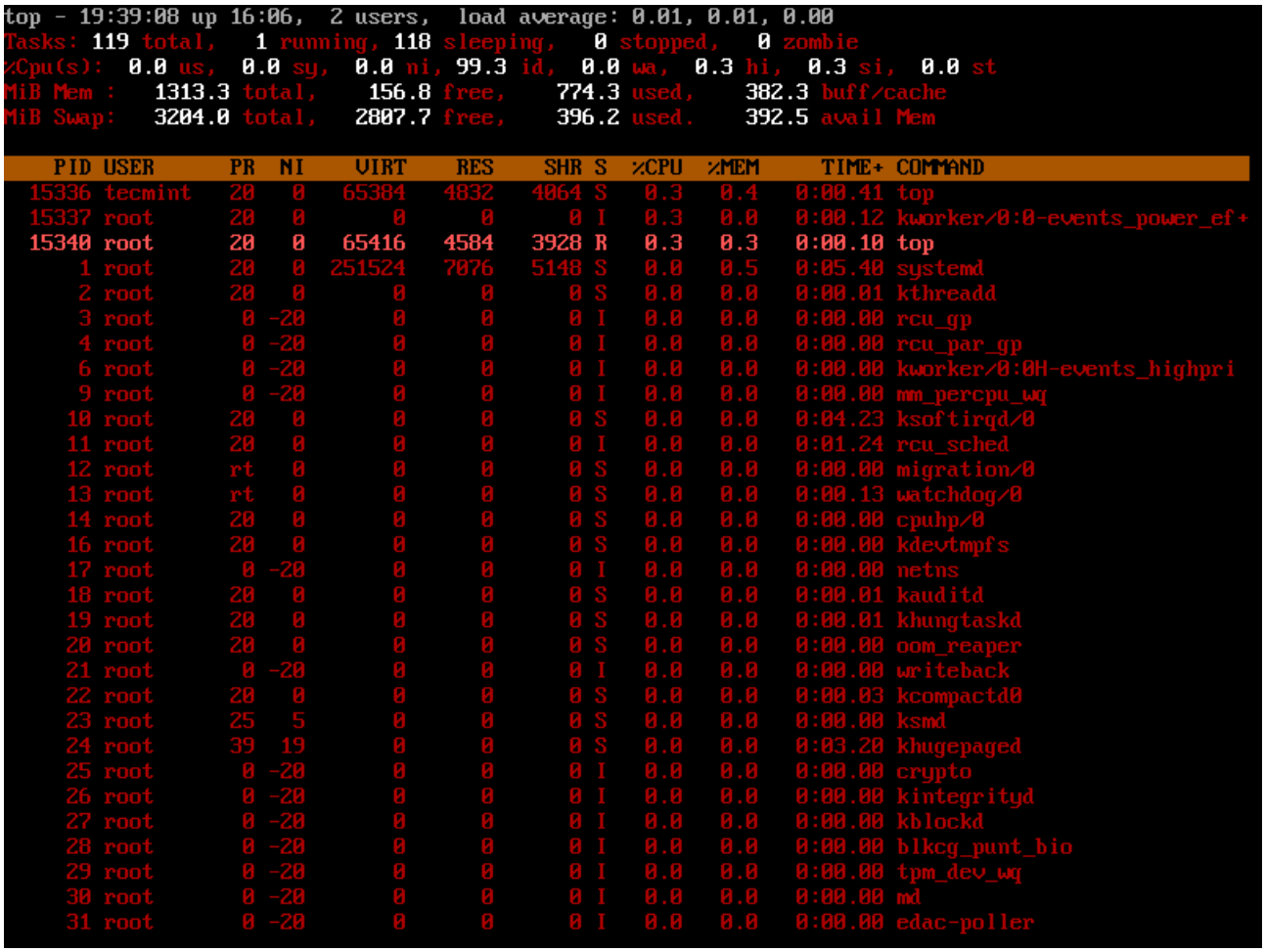
```
top - 19:36:05 up 16:03,  2 users,  load average: 0.00, 0.00, 0.00
Tasks: 119 total,   1 running, 118 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.3 hi,  0.3 si,  0.0 st
MiB Mem :   1313.3 total,    156.9 free,    774.2 used,    382.2 buff/cache
MiB Swap:   3204.0 total,   2807.7 free,    396.2 used.    392.5 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  15300 tecmint   20   0  100704  10168   8784 S   0.0   0.8   0:00.04 systemd
  15301 tecmint   20   0  321032   2360      4 S   0.0   0.2   0:00.00 (sd-pam)
  15308 tecmint   20   0  153400   5616   4364 S   0.0   0.4   0:00.00 sshd
  15309 tecmint   20   0   27020   4268   3692 S   0.0   0.3   0:00.01 bash
  15336 tecmint   20   0   65384   4832   4064 S   0.0   0.4   0:00.02 top
```

لیست فرآیندهای لینوکس بر اساس کاربر خاص

خروجی رنگی:

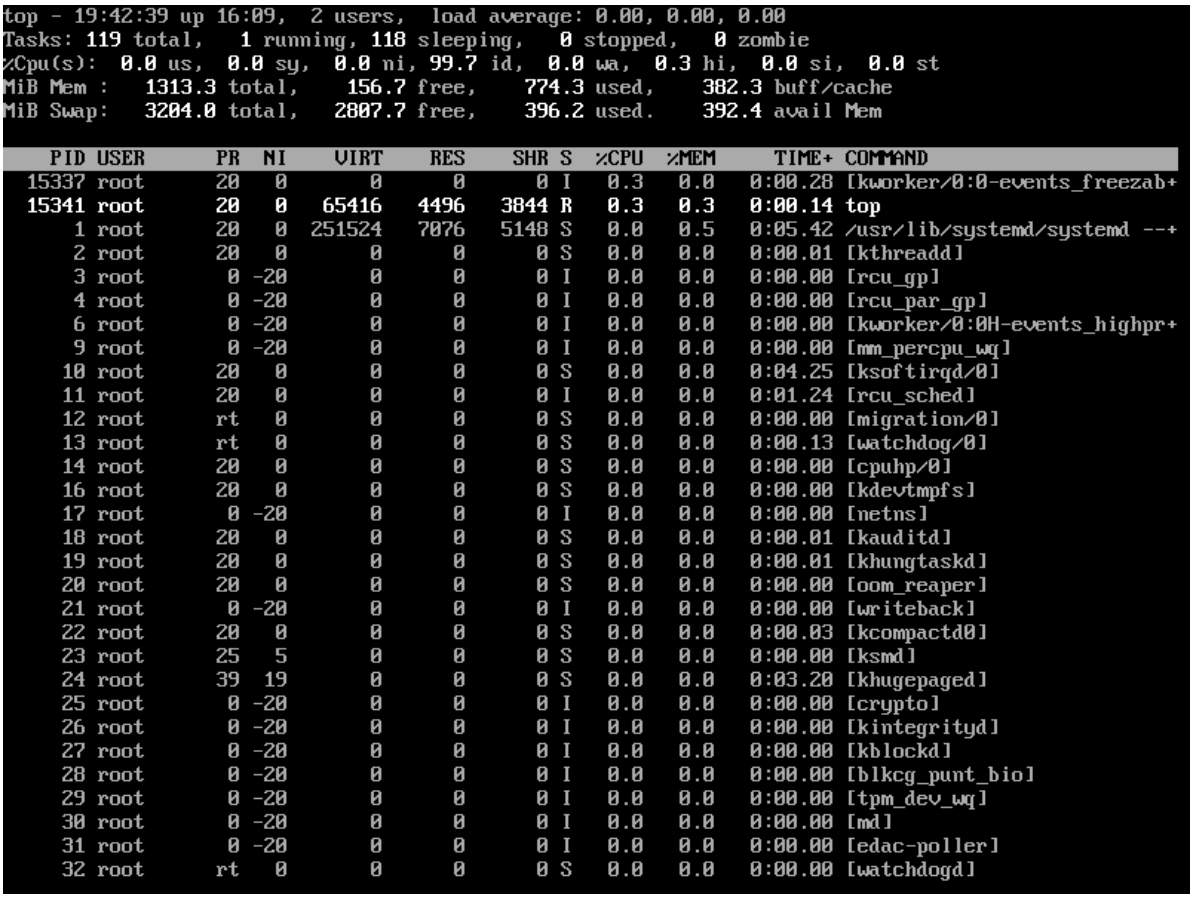
با فشار دادن Zگزینه " " روند در حال اجرا به صورت رنگی نمایش داده می شود که ممکن است به شما کمک کند تا فرآیند در حال اجرا را به راحتی شناسایی کنید.



برجسته کردن فرآیند های لینوکس

مسیر مطلق اجرای فرآیندها

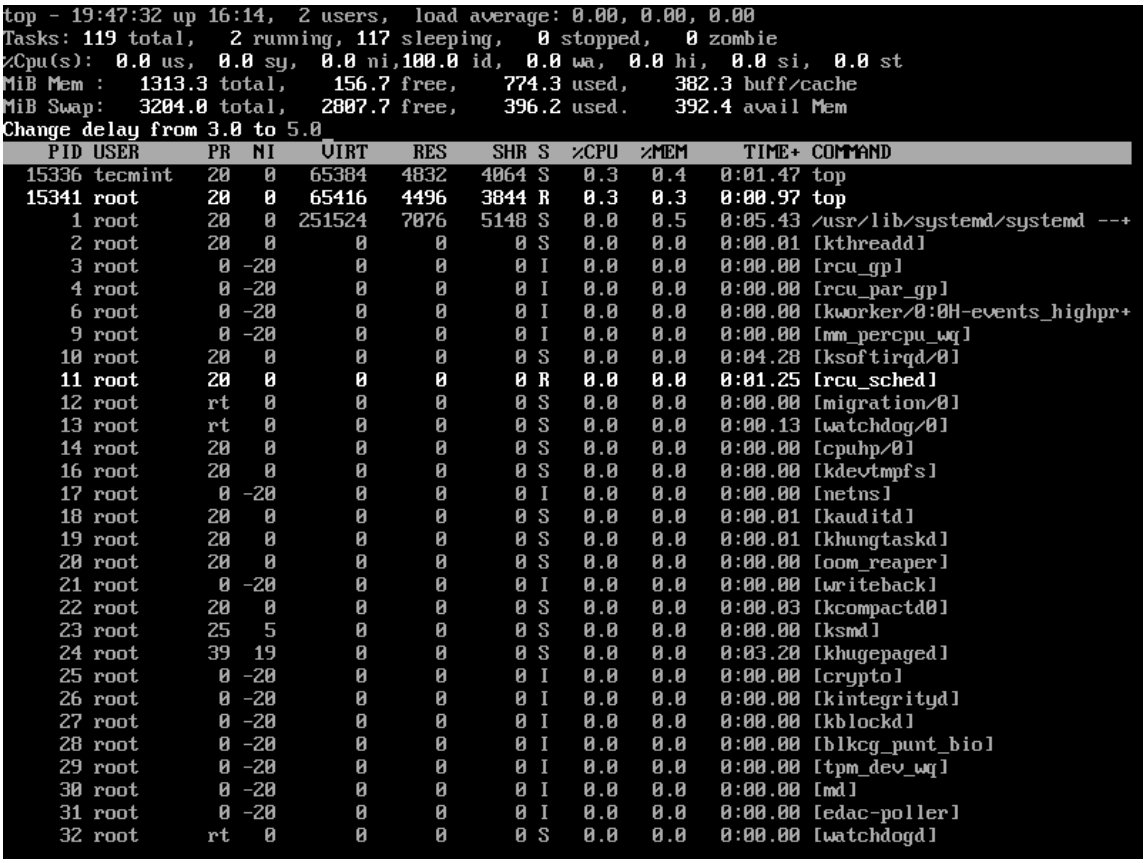
با فشار دادن گزینه ' C ' در دستور اجرای بالا مسیر مطلق فرآیند در حال اجرا نمایش داده می شود.



فهرست کردن فرآیندهای لینوکس را بر اساس مسیر

بازه تازه سازی را برای دستور top تنظیم کنید

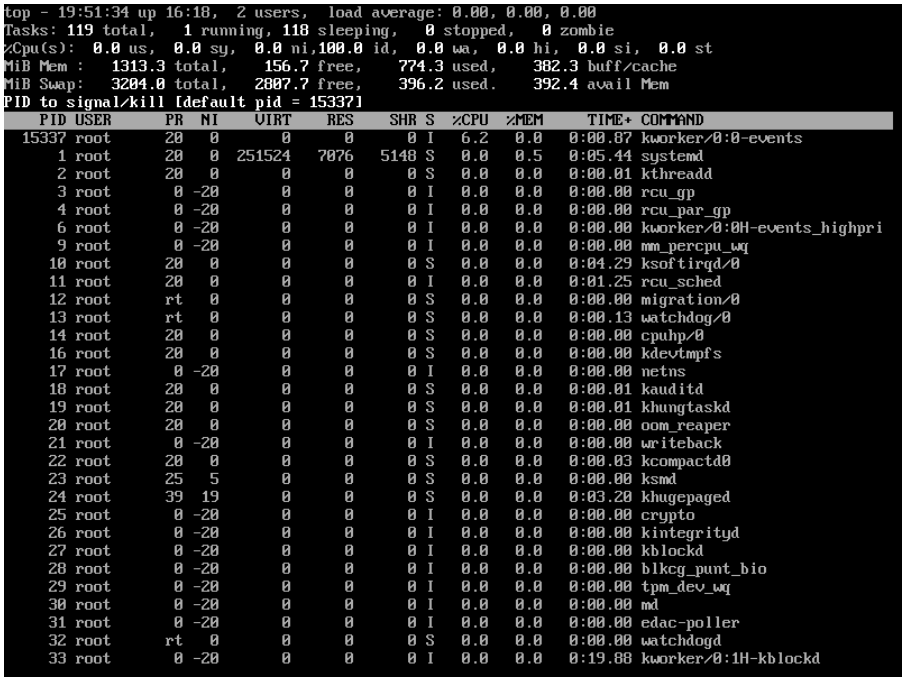
به‌طور پیش‌فرض فاصله بازخوانی صفحه روی 3.0ثانیه تنظیم شده است، همین مورد را می‌توان با فشار دادن گزینه d در اجرای فرمان بالا برای تنظیم زمان بازه دلخواه تغییر داد.



تنظیم کردن زمان بازخوانی فرمان top

توقف فرآیند لینوکس در حال اجرا

می‌توانید پس از یافتن PID فرآیند، با فشار دادن گزینه «k» در اجرای دستور top بدون بستن پنجره top، مانند شکل زیر، یک فرآیند را متوقف کنید.



توقف فرآیند لینوکس با استفاده از PID

مرتب سازی فرآیندهای لینوکس بر اساس استفاده ازCPU

برای مرتب کردن تمام فرآیندهای در حال اجرا بر اساس استفاده ازCPU ، به سادگی Shift+P کلید را فشار دهید.

```
top - 19:54:22 up 16:21,  2 users,  load average: 0.00, 0.00, 0.00
Tasks: 120 total,   2 running, 118 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.4 sy,  0.4 ni, 90.5 id,  8.4 wa,  0.4 hi,  0.0 si,  0.0 st
MiB Mem :   1313.3 total,    155.1 free,    774.6 used,    383.6 buff/cache
MiB Swap:   3204.0 total,   2808.0 free,    396.0 used.    392.0 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
      1 root      20   0  251524   7076   5148 S   0.3   0.5   0:05.46 systemd
  15336 tecmint   20   0   65384   4832   4064 S   0.3   0.4   0:02.36 top
  15350 root      20   0   65416   4524   3868 R   0.3   0.3   0:00.05 top
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
      9 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     10 root      20   0       0      0      0 S   0.0   0.0   0:04.30 ksoftirqd/0
     11 root      20   0       0      0      0 I   0.0   0.0   0:01.25 rcu_sched
     12 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/0
     13 root      rt   0       0      0      0 S   0.0   0.0   0:00.13 watchdog/0
     14 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
     17 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
     18 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kauditd
     19 root      20   0       0      0      0 S   0.0   0.0   0:00.01 khungtaskd
     20 root      20   0       0      0      0 S   0.0   0.0   0:00.00 oom_reaper
     21 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 writeback
     22 root      20   0       0      0      0 S   0.0   0.0   0:00.03 kcompactd0
     23 root      25   5       0      0      0 S   0.0   0.0   0:00.00 ksmd
     24 root      39  19       0      0      0 S   0.0   0.0   0:03.20 khugepaged
     25 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 crypto
     26 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kintegrityd
     27 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kblockd
     28 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 blkcg_punt_bio
     29 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 tpm_dev_wq
     30 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 md
     31 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 edac-poller
     32 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 watchdogd
```

مرتب سازی فرآیندهای لینوکس بر اساس استفاده ازCPU

یک فرآیند لینوکس را تقویت کنید

می توانید از گزینه ' r' برای تغییر اولویت فرآیند که Renice نیز نامیده می شود استفاده کنید .

```
top - 20:02:47 up 16:30,  2 users,  load average: 0.00, 0.00, 0.00
Tasks: 119 total,   1 running, 118 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   1313.3 total,    155.4 free,    774.3 used,    383.6 buff/cache
MiB Swap:   3204.0 total,   2808.0 free,    396.0 used.    392.4 avail Mem
PID to renice [default pid = 15368]
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  15368 root      20   0   65416   4348   3696 R   6.2   0.3   0:00.01 top
      1 root      20   0  251524   7076   5148 S   0.0   0.5   0:05.49 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
      9 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     10 root      20   0       0      0      0 S   0.0   0.0   0:04.34 ksoftirqd/0
     11 root      20   0       0      0      0 I   0.0   0.0   0:01.26 rcu_sched
     12 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/0
     13 root      rt   0       0      0      0 S   0.0   0.0   0:00.13 watchdog/0
     14 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
     17 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
     18 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kauditd
     19 root      20   0       0      0      0 S   0.0   0.0   0:00.01 khungtaskd
     20 root      20   0       0      0      0 S   0.0   0.0   0:00.00 oom_reaper
     21 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 writeback
     22 root      20   0       0      0      0 S   0.0   0.0   0:00.03 kcompactd0
     23 root      25   5       0      0      0 S   0.0   0.0   0:00.00 ksmd
     24 root      39  19       0      0      0 S   0.0   0.0   0:03.20 khugepaged
     25 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 crypto
     26 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kintegrityd
     27 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kblockd
     28 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 blkcg_punt_bio
     29 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 tpm_dev_wq
     30 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 md
     31 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 edac-poller
     32 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 watchdogd
     33 root       0 -20       0      0      0 I   0.0   0.0   0:19.89 kworker/0:1H-kblockd
```

فرآیند لینوکس Renice

هسته های CPU لینوکس را بررسی کنید

برای فهرست کردن اطلاعات بارگذاری هسته‌هایCPU ، کافیست 1 را برای فهرست کردن جزئیات هسته‌های CPU فشار دهید .

```
top - 12:59:37 up  3:56,  1 user,  load average: 1.30, 1.45, 1.42
Tasks: 253 total,   2 running, 251 sleeping,   0 stopped,   0 zombie
%Cpu0  : 10.9 us,  3.6 sy,  0.0 ni, 83.8 id,  0.0 wa,  0.0 hi,  1.7 si,  0.0 st
%Cpu1  :  9.4 us,  3.3 sy,  0.0 ni, 85.3 id,  1.0 wa,  0.0 hi,  1.0 si,  0.0 st
%Cpu2  :  9.5 us,  3.4 sy,  0.0 ni, 86.4 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
%Cpu3  :  9.2 us,  2.4 sy,  0.0 ni, 88.1 id,  0.0 wa,  0.0 hi,  0.3 si,  0.0 st
MiB Mem :   7853.0 total,    459.8 free,   5079.9 used,   2313.4 buff/cache
MiB Swap:  11264.0 total,  11245.0 free,     19.0 used.   1609.7 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   9023 tecmint   20   0   20.6g 364676 223996 S  14.3   4.5  33:05.61 chrome
    955 root      20   0 1326116 145660  99064 S   9.0   1.8  14:06.36 Xorg
  22603 tecmint   20   0   20.4g 152148  99176 R   6.3   1.9   0:09.99 chrome
   1390 tecmint   20   0  591352  50660  32280 S   4.3   0.6   6:36.71 xfwm4
   1627 tecmint   20   0   16.8g 493552 191464 S   3.3   6.1  22:48.73 chrome
  22992 tecmint   20   0  318692  37908  30688 S   3.3   0.5   0:00.62 xfce4-screen
  19049 tecmint   20   0 4080036   1.5g   1.4g S   3.0  19.9   2:06.27 VirtualBoxVM
   1677 tecmint   20   0   16.6g 348448 281080 S   2.7   4.3  24:54.32 chrome
  12062 tecmint   20   0   20.4g 262600 104324 S   2.3   3.3   4:27.55 chrome
  18001 tecmint   20   0   20.4g 165164 100716 S   1.3   2.1   1:03.83 chrome
  17231 tecmint   20   0  421960  46544  34324 S   1.0   0.6   0:10.19 xfce4-termin
```

هسته های CPU لینوکس را فهرست میکند

نتایج دستورات top را در فایل ذخیره کنید

برای ذخیره خروجی دستور بالا در حال اجرا در فایل /root/.toprc از دستور زیر استفاده کنید.

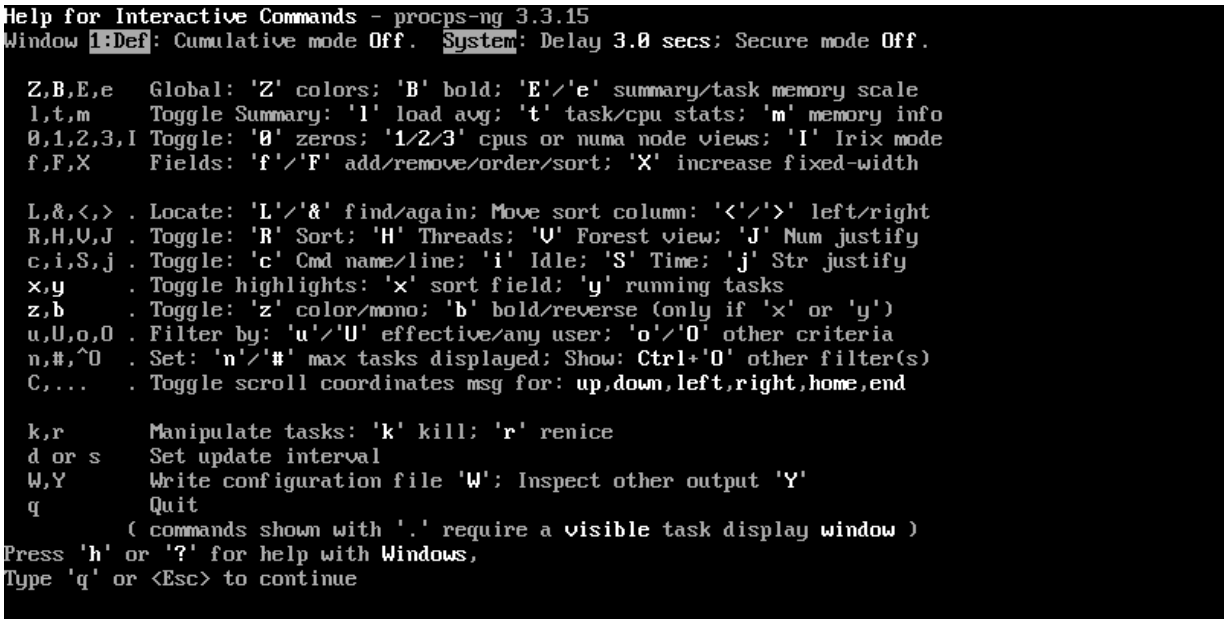
```
# top -n 1 -b > top-output.txt
```

لیست فرآیندهای idle لینوکس

کلید 'i' برای دریافت فرآیندهای خواب.

دریافت راهنمای دستور top

گزینه 'h' گزینه برای به دست آوردن راهنمای دستور top

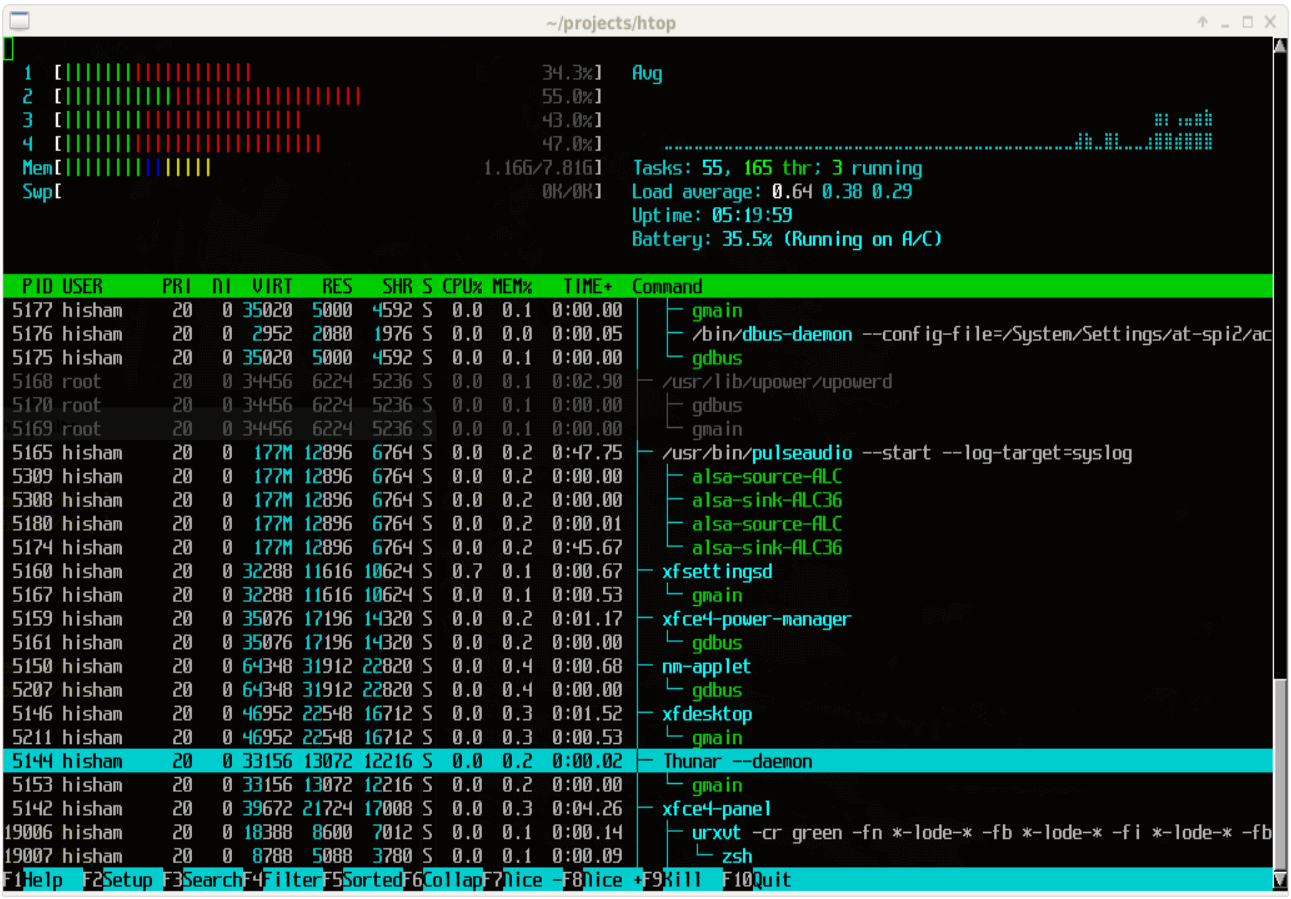


تکرار خاص در دستور top

خروجی فرمان بالا تا زمانی که ‘q’ را فشار دهید به روز می شود .با دستور زیر بعد از ۱۰ تکرار به صورت خودکار خارج می شود.

```
# top -n 10
```

تعدادی آرگومان برای دانستن بیشتر در مورد دستور topوجود دارد ، می توانید به صفحه man دستور topمراجعه کنید .



دستور htop

Htopیک برنامه کاربردی نظارت بر فرآیند بی‌درنگ برای سیستم‌های لینوکس/یونیکس و همچنین یک جایگزین مفید برای دستور topاست که یک ابزار نظارت بر فرآیند پیش‌فرض است که از پیش روی تمام سیستم‌عامل‌های لینوکس نصب شده است.

Htopدارای چندین ویژگی کاربر پسند دیگر است که تحت دستور top در دسترس نیستند و عبارتند از:

- در htop می توانید برای مشاهده لیست کامل فرآیندها به صورت عمودی اسکرول کنید و برای مشاهده خطوط فرمان کامل به صورت افقی حرکت کنید .
- در مقایسه با top بسیار سریع شروع می شود زیرا در هنگام راه اندازی منتظر نمی ماند تا داده ها را واکشی کند.
- درhtop ، شما می توانید بیش از یک فرآیند را در یک زمان بدون درج PID آنها بکشید.
- درhtop ، دیگر نیازی به وارد کردن شماره فرآیند یا مقدار اولویت برای بهبود مجدد یک فرآیند نیست.
- برای چاپ مجموعه ای از متغیرهای محیطی برای یک فرآیند، ”e”را فشار دهید .
- از ماوس برای انتخاب موارد لیست استفاده کنید.

نصب htop

این بسته عمدتا در تمام توزیع های مدرن لینوکس در دسترس است و می تواند با استفاده از مدیریت بسته پیشفرض از سیستم شما نصب شده است.

Htopرا روی دبیان نصب کنید

```
$ sudo apt install htop
```

Htopرا در اوبونتو نصب کنید

```
$ sudo apt install htop
```

Htopرا روی Linux Mint نصب کنید

```
$ sudo apt install htop
```

Htopرا روی فدورا نصب کنید

```
$ sudo dnf install htop
```

Htopرا روی CentOS 8/7 نصب کنید

```
$ sudo yum install epel-release
$ sudo yum install htop
```

Htopرا روی Rocky Linux/AlmaLinux نصب کنید

```
$ sudo yum install epel-release
$ sudo yum install htop
```

Htopرا روی جنتو نصب کنید

```
$ emerge sys-process/htop
```

Htopرا روی Arch Linux نصب کنید

```
$ pacman -S htop
```

Htopرا در OpenSUSE نصب کنید

```
$ sudo zypper install htop
```

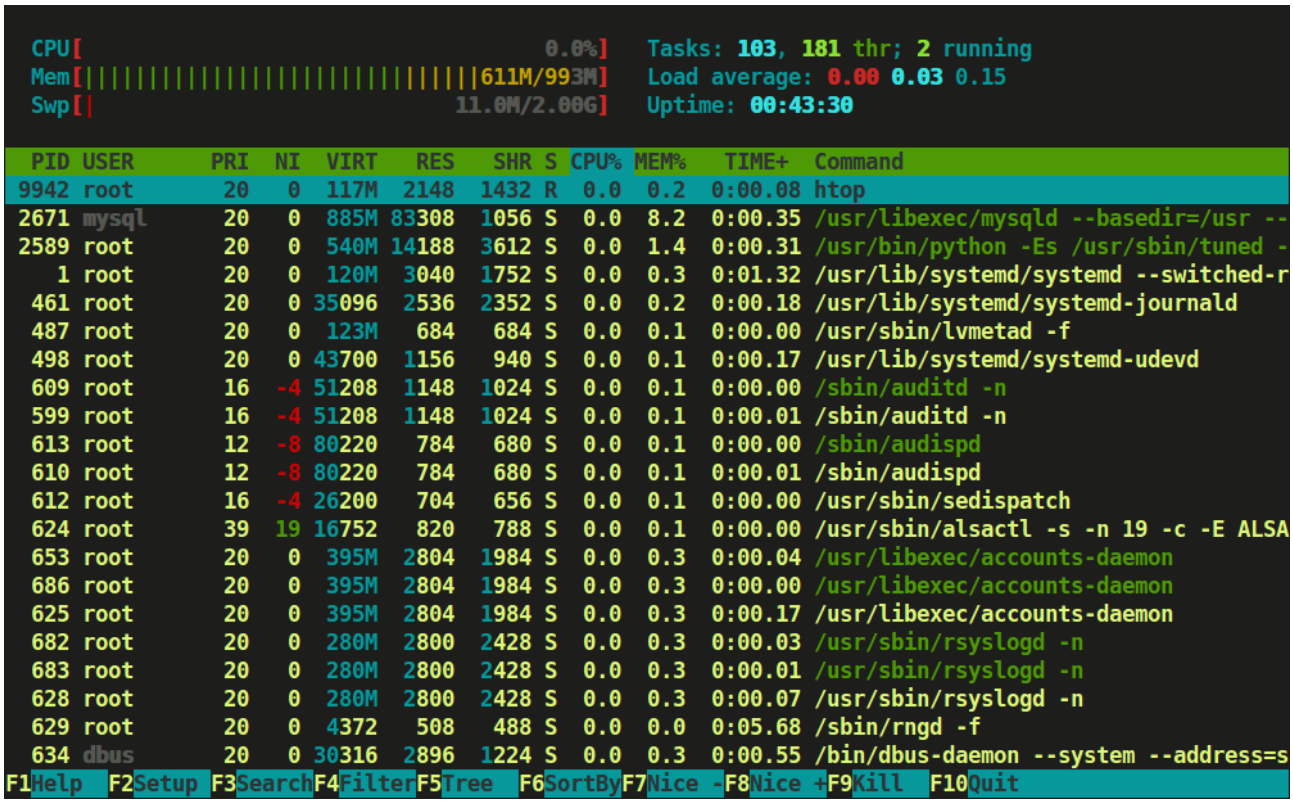
چگونه از htop استفاده کنم؟

اکنون با اجرای دستور زیر بر روی ترمینال، ابزار نظارت بر htop را اجرا کنید.

```
# htop
```

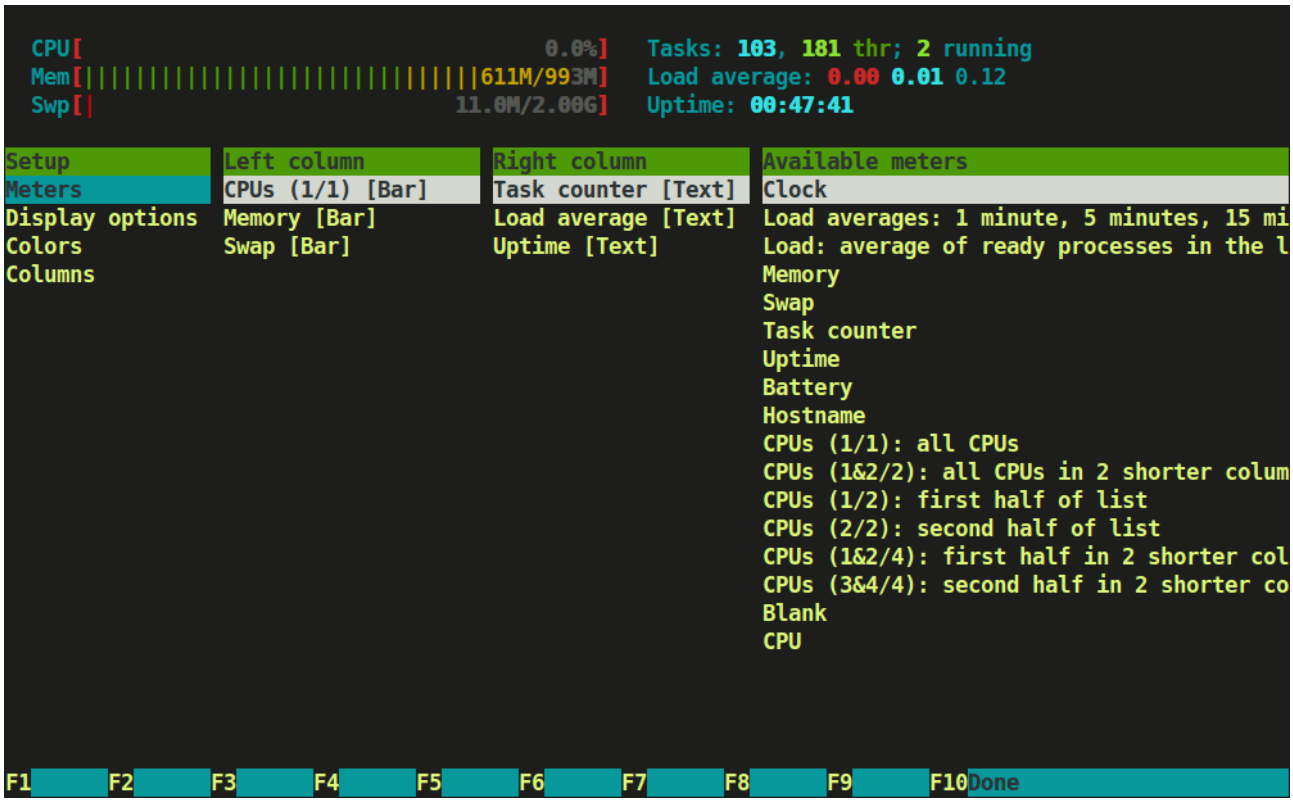
Htopدارای سه بخش است

headerجایی که می‌توانیم اطلاعاتی مانند CPU، Memory، Swapرا ببینیم و همچنین وظایف، میانگین بارگذاری و زمان زیاد را نشان می‌دهد .فهرستی از فرآیندها که بر اساس استفاده از CPUمرتب شده اند .

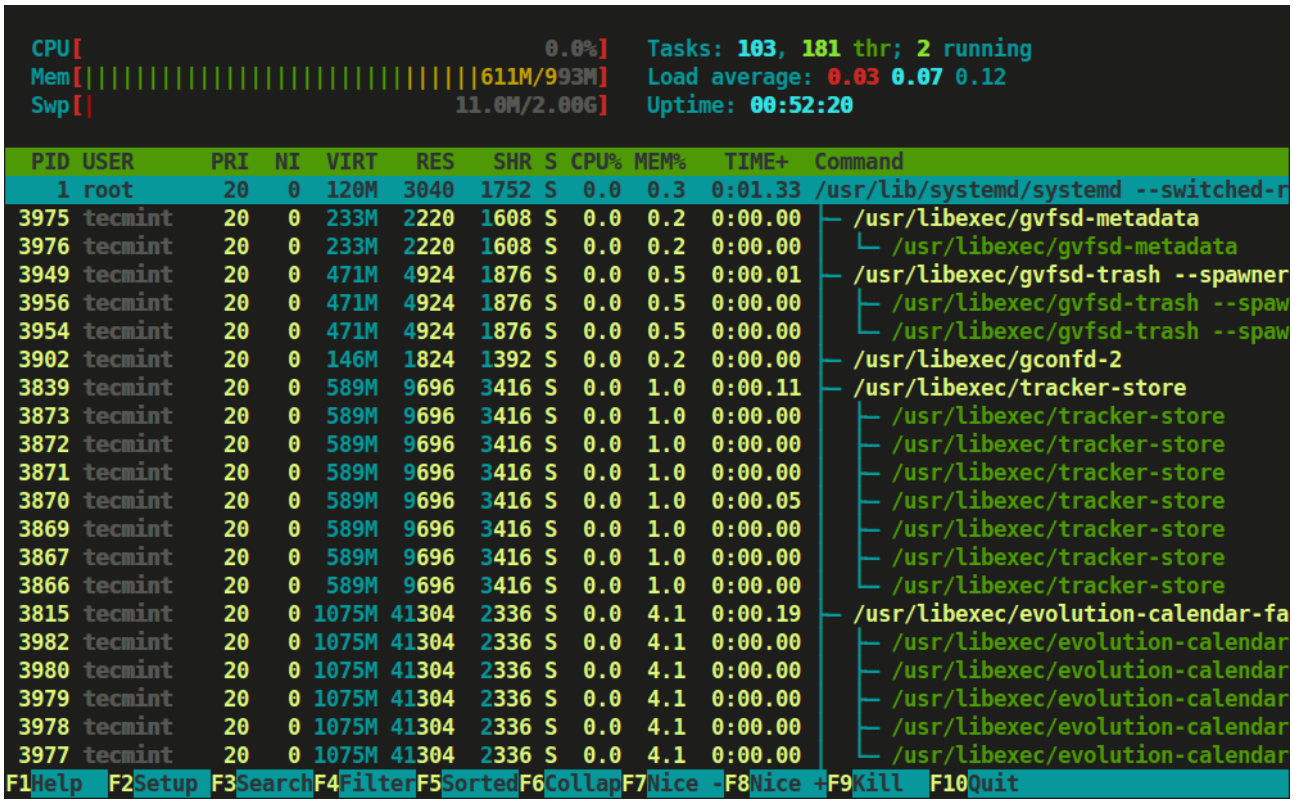


برای منوی تنظیمات، F2یا Sرا فشار دهید. چهار ستون وجود دارد، یعنی Setup، ستون چپ ، ستون راست و مترهای موجود .

در اینجا، می‌توانید مترهای چاپ شده در بالای پنجره را پیکربندی کنید، گزینه‌های مختلف نمایش را تنظیم کنید، از میان الگوهای رنگی انتخاب کنید و انتخاب کنید که کدام ستون‌ها به ترتیب چاپ شوند.



tree یا t را برای نمایش نمای درختی فرآیندها تایپ کنید.



برای استفاده از این برنامه کاربردی htop برای نظارت بر فرآیندهای در حال اجرا لینوکس، می توانید به کلیدهای عملکرد نمایش داده شده در پاورقی مراجعه کنید. با این حال، توصیه می کنیم از کلیدهای کاراکتر یا کلیدهای میانبر به جای کلیدهای عملکردی استفاده کنید زیرا ممکن است در طول اتصال ایمن با برخی عملکردهای دیگر نگاشت شده باشند.

کلیدهای میانبر و عملکردHtop

برخی از کلیدهای میانبر و عملکرد و عملکرد آنها برای تعامل با htop.

Description	Function Keys	Shortcut Keys
Help	F1	h
Setup	F2	s
Search Process	F3	/
Invert Sort Order	F4	I
Tree	F5	t
Sort By	F6	>
Nice - (change priority)	F7	[
Nice + (change priority)	F8	]
Kill a Process	F9	k

کلیدهای میانبر و دستورHtop

دستور monit

ابزاری برای نظارت بر خدمات در سیستم

ساختاردستور

```
$ monit [options] {arguments}
```

شرح دستور

monit ابزاری برای مدیریت و نظارت بر فرآیندها، فایل ها، فهرست ها و سیستم های فایل در یک سیستم یونیکس است. Monit تعمیر و نگهداری و تعمیر خودکار را انجام می‌دهد و می تواند اقدامات خوب و معنی داری را در موقعیت های خطا انجام دهد. به عنوان مثال. Monit می‌تواند یک فرآیند را در صورت اجرا نشدن راهاندازی کند، در صورت عدم پاسخگویی یک فرآیند را مجدداً راهاندازی می‌کند و اگر از منابع زیادی استفاده می‌کند فرآیندی را متوقف می‌کند. می‌توانید از Monit برای نظارت بر فایل‌ها، فهرست‌ها و سیستم‌های فایل برای تغییرات، مانند تغییرات مُهر زمانی، تغییرات چک‌سوم یا تغییرات اندازه استفاده کنید.دستور "monit" یک ابزار مدیریت و نظارت بر سیستم‌های لینوکس است که قابلیت‌های متنوعی دارد. این دستور توسط Monit نصب شده در سیستم قابل استفاده است. در ادامه، چند مثال از استفاده از گزینه‌های مختلف دستور "monit" را می‌توانید ببینید:

1.راه اندازی Monit:

```
$monit -I
```

این دستور Monit را راهاندازی می‌کند و به صورت تعاملی و با قابلیت نظارت بر روی سیستم اجرا می‌شود.

2.بررسی وضعیت فرآیندها:

```
$monit status
```

این دستور وضعیت فرآیندهایی که توسط Monit نظارت می‌شوند را نمایش می‌دهد.

3.اجرای یک فرآیند مشخص:

```
$monit start <process_name>
```

این دستور یک فرآیند مشخص را اجرا می‌کند. شما باید نام فرآیند مورد نظر را به جای <process_name> قرار دهید.

4.توقف یک فرآیند مشخص:

```
$monit stop <process_name>
```

این دستور یک فرآیند مشخص را متوقف می‌کند. شما باید نام فرآیند مورد نظر را به جای <process_name> قرار دهید.

5.راهاندازی مجدد یک فرآیند:

```
$monit restart <process_name>
```

این دستور یک فرآیند مشخص را مجدداً راهاندازی می‌کند. شما باید نام فرآیند مورد نظر را به جای <process_name> قرار دهید.

6.نظارت بر فایل‌ها و فهرست‌ها:

```
$monit monitor <file_or_directory_path>
```

دستور iotop

ساختاردستور:

```
$ iotop [OPTIONS]
```

شرح دستور:

iotop خروجی اطلاعات را نشان میدهد با استفاده از مانیتور I/O توسط هسته لینوکس. iotop ستون‌هایی را برای پهنای باند ورودی/خروجی که توسط هر فرآیند/رشته در طول دوره نمونه‌برداری خوانده و نوشته می‌شود، نمایش می‌دهد. همچنین درصد زمانی را که رشته/فرآیند صرف شده در حین تعویض و در حین انتظار در I/O نمایش می‌دهد. برای هر فرآیند، اولویت I/O آن (کلاس/سطح)

نشان داده شده است. علاوه بر این، کل پهنای باند ورودی/خروجی خوانده و نوشته شده در طول دوره نمونه برداری در بالای رابط نمایش داده می شود.از پیکان های چپ و راست برای تغییر مرتب سازی، r برای معکوس کردن ترتیب مرتب سازی، o برای جابجایی گزینه های دستور: ، p برای جابجایی گزینه processes، a برای جابجایی گزینه accumulated، q برای خروج یا i استفاده کنید..

گزینه های دستور:

```
$ iotop -o
$ iotop --only
```

به جای نمایش همه فرآیندها یا رشته ها، فقط فرآیندها یا رشته هایی را نشان میدهد که در واقع I/O را انجام می دهند. این را می توان به صورت پویا با فشار دادن o تغییر داد.

```
$ iotop -b
$ iotop --batch
```

حالت غیر تعاملی را روشن میکند. برای ثبت میزان استفاده از I/O در طول زمان مفید است.

```
$ iotop -n NUM
$ iotop --iter=NUM
```

تعداد تکرارها را قبل از خروج تنظیم میکند. این بیشتر در حالت غیر تعاملی مفید است.

```
$ iotop -d SEC
$ iotop --delay=SEC
```

تأخیر بین تکرارها را بر حسب ثانیه تنظیم میکند (به طور پیش فرض ۱ ثانیه). مقادیر غیر صحیح مانند ۱,۱ ثانیه را می پذیرد.

```
$ iotop -p PID
$ iotop --pid=PID
```

لیستی از فرآیندها/رشته ها برای نظارت (همه به طور پیش فرض).

```
$ iotop -u USER
 $ iotop --user=USER
```

لیستی از کاربران برای نظارت (همه به طور پیش فرض)

```
 $ iotop -P
$ iotop --processes
```

فقط فرآیندها را نشان می‌دهد. به طور معمول iotop همه موضوعات را نشان می‌دهد.

```
$ iotop -a
$ iotop --accumulated
```

نمایش ورودی/خروجی ،به جای پهنای باند. در این حالت، iotop میزان پردازش های I/O انجام شده از زمان شروع iotop را نشان می‌دهد.

```
$ iotop -k
$ iotop --kilobytes
```

iotop تمام اندازه ها را در کیلوبایت نمایش می‌دهد.

```
$ iotop -t
$ iotop --time
```

در هر خط یک timestamp اضافه میکند. برچسب زمانی

```
$ iotop -q
```

این گزینه را می توان تا سه بار برای حذف خطوط سرصفحه مشخص کرد.

دستور nice/renice

nice/renice برای تغییر اولویت برنامه برای اجرا استفاده می شود . محدوده اولویت بین ۲۰- تا ۱۹ است که ۱۹ کمترین اولویت را دارد.

مثال های دستور:

۱.اجرای دستور cc در پس زمینه با اولویت کمتر از پیش فرض (آهسته تر):

```
$ nice -n 15 cc -c *.c &
```

۲. اولویت را برای همه فرآیندهای متعلق به گروه "تست" افزایش دهید:

```
$ renice --20 -g test
```

ساختاردستور:

```
$ nice [ -Increment| -n Increment ] Command [ Argument ... ]
```

دستور ionice

دستور ionice برای تنظیم یا دریافت کلاس زمانبندی ورودی/خروجی فرآیند و اولویت استفاده می شود.

اگر هیچ آرگومانی داده نشود، ionice کلاس زمانبندی ورودی/خروجی فعلی و اولویت آن فرآیند را جویا می شود.

ساختاردستور:

```
$ ionice [options] -p <pid>
$ ionice [options] -P <pgid>
$ ionice [options] -u <uid>
$ ionice [options] <command
```

گزینه های دستور:

- c – ، نام کلاس یا تعداد کلاس برنامه ریزی، ۰: هیچ، ۱: بیدرنگ، ۲: بهترین تلاش، ۳: بیکار
- n – ، Priority classdata – (۷..۰) در کلاس زمانبندی مشخص شده، فقط برای کلاسهای بیدرنگ و بهترین تلاش
- p – ، pid-- ... روی این فرآیندهای در حال اجرا عمل می کنند
- P – pgid-- , ... بر روی فرآیندهای در حال اجرا در این گروه ها عمل می‌کند
- t– نادیده گرفتن شکست ها
- u – ، uid-- ... بر روی فرآیندهای در حال اجرا متعلق به این کاربران عمل می‌کند
- h – ، help-- نمایش این راهنما
- V – ، نمایش نسخه

نمونه استفاده از گزینه : -c

```
$ionice -c 2 command_name
```

این دستور، دستور command_name را با کلاس برنامه‌ریزی" I/O بهترین تلاش" اجرا می‌کند.

نمونه استفاده از گزینه : -n

```
$ionice -n 4 command_name
```

این دستور، دستور command_name را با کلاس برنامه‌ریزی" I/O بیدرنگ" و اولویت ۴ اجرا می‌کند.

نمونه استفاده از گزینه : -p

```
$ionice -p 1234
```

این دستور، اولویت I/O فرآیند با شناسه پردازه 1234 (PID) را تنظیم می‌کند.

نمونه استفاده از گزینه : P-

```
$ionice -P 5678
```

این دستور، اولویت I/O فرآیندهای در حال اجرا در گروه فرآیندی 5678 (PGID) را تنظیم می‌کند.

نمونه استفاده از گزینه : t-

```
$ionice -t command_name
```

این دستور، دستور command_name را اجرا می‌کند و شکست‌ها را نادیده می‌گیرد.

نمونه استفاده از گزینه : u-

```
$ionice -u username command_name
```

این دستور، دستور command_name را برای فرآیندهای در حال اجرا که متعلق به کاربر با نام کاربری username هستند، اجرا می‌کند.

نمونه استفاده از گزینه : h-

```
$ionice -h
```

این دستور، راهنمای دستور ionice را نمایش می‌دهد.

نمونه استفاده از گزینه : V-

```
$ionice -V
```

دستور strace

دستور strace در سیستم‌عامل‌های Unix/Linux برای پیگیری و نمایش تمامی فراخوانی‌های سیستمی (system calls)که یک فرآیند انجام می‌دهد، استفاده می‌شود. این دستور به کاربر این امکان را می‌دهد تا عملکرد و عملیات انجام شده توسط یک برنامه را بررسی کرده و از آن‌ها بازخورد دریافت کند.

مثال‌های دستور:

نمایش تمام فراخوانی‌های سیستمی یک برنامه: این دستور برای نمایش تمام فراخوانی‌های سیستمی یک برنامه استفاده می‌شود.

```
strace ./my_program
```

نمایش تنها فراخوانی‌های سیستمی از یک نوع خاص: این دستور برای نمایش تنها فراخوانی‌های سیستمی از یک نوع خاص مانند open استفاده می‌شود.

```
strace -e open ./my_program
```

ذخیره خروجی strace در یک فایل: این دستور برای ذخیره خروجی فراخوانی‌های strace در یک فایل استفاده می‌شود.

```
strace -o output.txt ./my_program
```

نمایش اطلاعات کامل فراخوانی‌های سیستمی: این دستور برای نمایش اطلاعات کامل و جزئی‌تر در مورد هر فراخوانی سیستمی استفاده می‌شود.

```
strace -v ./my_program
```

اجرای برنامه با strace و نمایش فراخوانی‌ها به صورت زمان‌بندی شده: این دستور برای اجرای برنامه و نمایش فراخوانی‌های سیستمی به صورت زمان‌بندی شده استفاده می‌شود.

```
strace -T ./my_program
```

دستور Wait

این دستوری است که منتظر تکمیل هر فرآیند در حال اجرا ID داده شده است. اگر شناسه پردازش داده نشود، منتظر می ماند تا تمام فرآیندهای فرزند فعلی تکمیل شوند.

wait: این دستور در یک اسکریپت منتظر اتمام تمامی فرآیندها است.

```
command1 &
command2 &
wait
echo “All commands have finished.”
```

wait <pid>: این دستور در یک اسکریپت منتظر اتمام فرآیند با شناسه پردازه مشخص (PID) است.

```
command1 &
command2 &
command3 &
wait <pid>
echo “Process with PID <pid> has finished.”
```

wait -n: این دستور در یک اسکریپت منتظر اتمام هر یک از فرآیندها است و پس از اتمام هر فرآیند، ادامه می‌دهد.

```
command1 &
command2 &
wait -n
echo “A command has finished.”
```

دستور kill

فرض کنید که یک برنامه داشته باشید که پاسخی نداشته باشد چگونه از آن خلاص شوید؟ در حالی که دستور kill برای فرآیندها مورد استفاده قرار می گیرد، هدف واقعی آن ارسال *سیگنال ها* به فرآیندها است. برنامه ها (اگر به درستی نوشته شده باشند) برای سیگنال ها به سیستم عامل گوش فرا می دهند و به آنها پاسخ می دهند، اغلب اجازه می دهد تا برخی از روش های منحصر به فرد، قطع شود. دستور kill می تواند سیگنال های مختلفی را برای فرآیندها ارسال کند. تایپ کنید :

```
kill -l
```

به یک لیست از سیگنال های آن را پشتیبانی می کند. اکثر آنها نسبتا مبهم هستند، اما چندین مورد مفید هستند :

سیگنال	نام	شرح
1	**Sighup**	قطع کردن سیگنال برنامه ها می توانند برای این سیگنال گوش دهند و بر آن عمل کنند. هنگامی که ترمینال را ببندید، این سیگنال به فرآیندهای در حال اجرا در یک ترمینال ارسال می شود.
2	**Sigint**	سیگنال وقفه این سیگنال به فرآیندها داده می شود تا آنها را قطع کنند. برنامه ها می توانند این سیگنال را پردازش کنند و بر آن عمل کنند . همچنین می توانید این سیگنال را به طور مستقیم با تایپ کردن Ctrl-c در پنجره ترمینال که در آن برنامه اجرا می شود را صادر کنید.
15	**Sigterm**	سیگنال خاتمه دادن این سیگنال به فرآیندها داده می شود تا آنها را خاتمه دهد. دوباره برنامه ها می توانند این سیگنال را پردازش کنند و بر آن عمل کنند. اگر سیگنال مشخص نشده باشد، این سیگنال به طور پیش فرض توسط دستور kill ارسال می شود.
9	**Sigkill**	سیگنال را بکشید این سیگنال منجر به توقف روند توسط هسته لینوکس می شود. برنامه ها نمی توانند برای این سیگنال گوش کنند.

ساختار دستور:

```
kill [-signal|-s signal|-p] [-q value] [-a] [--timeout
milliseconds signal] [--] pid|name...
```

```
kill -l [number] | -L
```

مثال های دستور:

گزینه s- یا:signal--

برای ارسال یک سیگنال خاص به پردازه‌ها.

```
kill -s SIGTERM <PID>
```

گزینه:p-

برای ارسال سیگنال به گروه پردازه‌ها.

```
kill -p <PGID>
```

گزینه:q-

برای تنظیم مقدار اولویت برای ارسال سیگنال.

```
kill -q 15 <PID>
```

گزینه:a-

برای ارسال سیگنال به همه پردازه‌های فرزند.

```
kill -a <PID>
```

گزینه:timeout--

برای تنظیم زمان انتظار قبل از ارسال سیگنال.

```
kill --timeout 5000 SIGKILL <PID>
```

گزینه:l-

برای نمایش لیست سیگنال‌های موجود.

```
kill -l
```

گزینه:L-

برای نمایش لیست نام‌های سیگنال‌ها.

```
kill -L
```

دستور nohup

هنگامی که یک پوسته خارج می شود (شاید در حین خروج از یک جلسه SSH)، سیگنال ('Hang up') HUP به همه پردازش های فرزند آن ارسال می شود و باعث خاتمه آنها می شود. اگر پس از خروج از پوسته به یک فرآیند طولانی مدت نیاز دارید، ازدستور nohup استفاده کنید . دستور nohup برنامه دیگری را که به عنوان آرگومان مشخص شده است اجرا می کند و تمام سیگنالهای

) SIGHUPقطع) را نادیده می گیرد. پیشوند هر دستور با nohup باعث می شود که دستور در برابر سیگنال های HUP مصون شود. علاوه بر این، STDIN نادیده گرفته می شود و تمام خروجی ها به فایل محلی ./nohup.out هدایت می شوند .

ساختار دستور:

```
$ nohup your_command
nohup OPTION
```

مثال های دستور:

۱. اعمال nohup در ارتقای زمان بر:

```
$ nohup apt-get -y upgrade
```

استفاده از nohup در پیش زمینه بسیار مفید نیست زیرا تا زمانی که دستور کامل نشود نمی توانید با پوسته تعامل کنید.

برای اجرای دستور در پس زمینه ، نماد & را در انتهای دستور ضمیمه کنید:

```
$ nohup mycommand &
```

گزینه :n-تغییر نام فایل خروجی.(output file)

```
$nohup command > output.txt
```

گزینه :p-نمایش شماره پردازه (process ID) پشت زمینه.

```
$nohup -p
```

گزینه :s-تغییر سیگنال (signal) برای پایان دادن به پردازه پشت زمینه.

```
$nohup command > output.txt 2>&1 &
$kill -s SIGTERM <PID>
```

گزینه :l-نمایش لیست پردازه‌های پشت زمینه.

```
$nohup -l
```

گزینه :c-تغییر سیگنال (signal) برای پایان دادن به پردازه پشت زمینه.

```
$nohup command > output.txt 2>&1 &
$kill -s SIGKILL <PID>
```

گزینه :h-نمایش راهنمای دستور nohup.

```
$nohup -h
```

دستور pstree

دستور pstree شبیه ps است، اما به جای فهرست کردن فرآیندهای در حال اجرا، آنها را به صورت درخت نشان می‌دهد. قالب درخت مانند گاهی اوقات راه مناسب تری برای نمایش سلسله مراتب فرآیندها است که راه بسیار ساده تری برای تجسم فرآیندهای در حال اجرا است. ریشه درخت یا init یا فرآیند با pid داده شده است.

ساختار دستور:

```
pstree [-a, --arguments] [-c, --compact-not] [-C, --color attr]
[-g, --show-pgids]
[-h, --highlight-all, -H pid, --highlight-pid pid] [-l, --long]
[-n, --numeric-sort] [-N, --ns-sort ns] [-p, --show-pids]
[-s, --show-parents] [-S, --ns-changes] [-t, --thread-names]
[-T, --hide-threads] [-u, --uid-changes] [-Z, --security-context]
[-A, --ascii, -G, --vt100, -U, --unicode] [pid, user]
```

```
pstree -V, --version
```

گزینه های دستور:

```
$ ps [OPTIONS] [USER or PID]
```

- a --arguments نمایش آرگومان های خط فرمان
- A --ascii از کاراکترهای رسم خط ASCII استفاده می‌کند
- c --compact زیردرختهای یکسان را فشرده نمی کند.
- h --highlight-all فرآیند فعلی و اجداد آن را برجسته میکند
- H - PID --highlight-pid=PID این فرآیند و اجداد آن را برجسته می‌کند
- g --show-pgids شناسه های گروه فرآیند را نشان می‌دهد. دلالت بر - cدارد
- G --vt100 از کاراکترهای ترسیم خط VT100 استفاده می‌کند
- l - --long خطوط طولانی را کوتاه نمیکند
- n --numeric-sort مرتب سازی خروجی بر اساس PID
- N --ns-sort=type– مرتب سازی بر اساس نوع فضای نام (cgroup, ipc,mnt، net، pid، user، uts)
- p --show-pids پردازش ها را نشان می‌دهد. دلالت بر –c دارد
- s --show-parents نمایش والدین فرآیند انتخاب شده
- S --ns-changes انتقال فضای نام را نشان می‌دهد
- t --thread-names نمایش نام های کامل رشته
- T --hide-threads موضوعات و جزئیات را پنهان میکند، فقط فرآیندها را نشان دهید
- u --uid-changes نمایش انتقال uid
- U --unicode از طراحی خطی (UTF-8یونیکد) استفاده کنید
- V --version نمایش اطلاعات نسخه
- Z - --security-context نمایش زمینه های امنیتی SELinux

مثال های دستور:

۱.برای نمایش ساختار درختی سلسله مراتبی از تمام فرآیندهای در حال اجرا:

```
$ pstree
```

۲. برای نمایش درختی با فرآیند داده شده به عنوان ریشه درخت:

```
$ pstree [pid]
```

۳. برای نمایش فقط آن دسته از فرآیندهایی که توسط یک کاربر شروع شده اند:

```
$ pstree [USER]
```

۴. برای نمایش فرآیندهای والد فرآیند داده شده:

```
$ pstree -s [PID]
```

گزینه :p-نمایش شماره پردازه (PID) هر پردازه.

```
$pstree -p
```

گزینه :a-نمایش آرگومان‌های روند (command-line arguments) هر پردازه.

```
$pstree -a
```

گزینه n-: مرتب‌سازی پردازه‌ها بر اساس شماره پردازه.(PID)

```
$pstree -n
```

گزینه c-: نمایش تعداد فرزندان (children) هر پردازه.

```
$pstree -c
```

گزینه h-: نمایش راهنمای دستور pstree.

```
$pstree -h
```

گزینه U-: نمایش نمادهای یونیکد (Unicode) برای نمایش پردازه‌ها.

```
$pstree -U
```

گزینه Z-: نمایش اطلاعات امنیتی (security information) هر پردازه.

```
$pstree -Z
```

دستور ps

شرح دستور:

دستور ps یکی از پرکاربردترین دستورات در سیستم‌عامل‌های یونیکس‌مانند و لینوکس است که برای مشاهده فرآیندهای در حال اجرا در سیستم استفاده می‌شود .با استفاده از این دستور می‌توانید اطلاعات مفیدی در مورد هر فرآیند مانند PID (شناسه فرآیند)، نام فرآیند، کاربر اجراکننده، استفاده از حافظه و CPU و ... به دست آورید.

ساختار دستور:

```
ps [ options ] [ argument ]
```

- **options:** گزینه‌هایی هستند که رفتار دستور را تغییر می‌دهند و نوع اطلاعات نمایش داده شده را کنترل می‌کنند.
- **argument:** آرگومانی است که برای انتخاب فرآیندهای خاص استفاده می‌شود) مثلاً PID فرآیند.(

کاربردهای دستور:

- **مشاهده لیستی از تمام فرآیندهای در حال اجرا :** با استفاده از گزینه‌های مختلف، می‌توانید لیستی از تمام فرآیندها یا فرآیندهای خاصی را مشاهده کنید.
- **یافتن PID یک فرآیند خاص :** با استفاده از نام فرآیند یا کاربر اجراکننده، می‌توانید PID فرآیند مورد نظر را پیدا کنید.
- **بررسی وضعیت و منابع مصرفی یک فرآیند :** با استفاده از گزینه‌های مختلف، می‌توانید اطلاعات دقیقی در مورد وضعیت و منابع مصرفی هر فرآیند به دست آورید.
- **پیدا کردن فرآیندهای مشکل‌ساز :** با استفاده از دستور ps می‌توانید فرآیندهایی که باعث کندی سیستم یا مشکلات دیگر می‌شوند را شناسایی کنید.

گزینه‌های مهم دستور:

- **a:** نمایش تمام فرآیندها، حتی فرآیندهای متعلق به کاربران دیگر.
- **c:** نمایش نام کامل دستور.(command name)
- **e:** نمایش اطلاعات گسترده در مورد فرآیندها.
- **g:** نمایش فرآیندهای گروهی.
- **n:** نمایش نام کاربری عددی به جای نام کاربری واقعی.
- **w:** نمایش خطوط طولانی‌تر.

- **x:** نمایش فرآیندهایی که ترمینال کنترلی ندارند (مانند دیمون‌ها).
- **l:** نمایش اطلاعات طولانی در مورد فرآیندها.
- **s:** نمایش اطلاعات خلاصه در مورد فرآیندها.
- **u:** نمایش اطلاعات مربوط به کاربر.
- **v:** نمایش اطلاعات مجازی در مورد فرآیندها.
- **t tty:** نمایش فرآیندهای مرتبط با ترمینال مشخص شده.
- **X:** نمایش اطلاعات اضافی.
- **ProcessNumber:** نمایش اطلاعات مربوط به فرآیند با شناسه مشخص شده.

مثال‌های کاربردی دستور ps با گزینه‌های مختلف

دستور psابزاری قدرتمند برای مشاهده فرآیندهای در حال اجرا در سیستم‌عامل‌های یونیکس‌مانند و لینوکس است .با استفاده از گزینه‌های مختلف، می‌توانید اطلاعات دقیق و مفیدی درباره فرآیندها به دست آورید .در ادامه، چندین مثال کاربردی برای گزینه‌های مختلف این دستور آورده شده است:

نمایش کلی فرآیندها

- **نمایش تمام فرآیندها، شامل فرآیندهای کاربران دیگر:**

```
ps aux
```

- **نمایش نام کامل دستور برای همه فرآیندها:**

```
ps auxc
```

- **نمایش اطلاعات گسترده برای همه فرآیندها:**

```
ps auxe
```

فیلتر کردن فرآیندها

- **نمایش فرآیندهای مربوط به کاربر علی:**

```
ps -u ali
```

- **نمایش فرآیندهای مرتبط با ترمینال:pts/0**

```
ps -t pts/0
```

- **نمایش فرآیندهای بدون ترمینال کنترلی (دیمون‌ها):**

```
ps aux | grep -v TTY
```

نمایش اطلاعات خاص

- **نمایش اطلاعات طولانی در مورد فرآیند با:PID 1234**

```
ps -l 1234
```

- **نمایش اطلاعات خلاصه در مورد همه فرآیندها:**

```
ps as
```

- **نمایش اطلاعات مربوط به کاربر برای همه فرآیندها:**

```
ps aux
```

نمایش اطلاعات پیشرفته

- **نمایش اطلاعات مجازی در مورد فرآیندها:**

```
ps auxv
```

- **نمایش اطلاعات اضافی برای همه فرآیندها:**

```
ps auxX
```

ترکیب گزینه‌ها

- **نمایش نام کامل دستور، اطلاعات گسترده و فرآیندهای گروهی:**

```
ps auxeg
```

- **نمایش نام کاربری عددی و فرآیندهای مرتبط با ترمینال:pts/1**

```
ps -nt pts/1
```

تفسیر خروجی

خروجی دستور ps معمولاً شامل ستون‌هایی مانند) PID شناسه فرآیند(،) PPIDشناسه فرآیند پدر(،) USERنام کاربر(، %CPU)درصد استفاده از(CPU ،) %MEMدرصد استفاده از حافظه(،) VSZاندازه فضای مجازی فرآیند(،) RSSاندازه مجموعه کاری مقیم فرآیند(،) TTYترمینال مرتبط با فرآیند(،) STATوضعیت فرآیند(،) STARTزمان شروع فرآیند(،) TIMEزمان کل CPUمصرف شده توسط فرآیند (و COMMAND (دستور اجرا شده برای شروع فرآیند) است.

نکته:

- با ترکیب گزینه‌های مختلف، می‌توانید خروجی دستور ps را به صورت دلخواه سفارشی کنید.
- برای فیلتر کردن خروجی، می‌توانید از دستورات دیگر مانند grep استفاده کنید.
- دستور psابزاری بسیار قدرتمند است و با یادگیری آن می‌توانید به راحتی فرآیندهای سیستم خود را مدیریت کنید.

مثال‌های بیشتر و پیچیده‌تر:

- **یافتن فرآیندهایی که بیشترین حافظه را مصرف می‌کنند:**

```
ps aux | sort -k 4 -r | head -n 10
```

- **یافتن فرآیندهایی که کلمه خاصی در نام آن‌ها وجود دارد:**

```
ps aux | grep httpd
```

دستور tree

دستور tree در لینوکس به صورت بازگشتی فهرست ها را به عنوان ساختارهای درختی فهرست می‌کند. هر فهرست بر اساس عمق آن نسبت به ریشه درخت تورفتگی دارد.

ساختاردستور:

```
$ tree [-acdfghilnpqrstuvxACDFQNSUX] [-L level [-R]] [-H baseHREF] [-T title] [-o filename] [--nolinks] [-P pattern] [-I pattern] [- -inodes] [--device] [--noreport] [--dirsfirst] [--version] [-- help] [--filelimit #] [--si] [--prune] [--du] [--timefmt format] [-- matchdirs] [--from-file] [--] [directory ...]
```

گزینه های دستور:

- a- چاپ همه فایل ها، از جمله موارد مخفی.
- d- فقط فهرست راهنماها.
- l- پیوندهای نمادین را به فهرست ها دنبال میکند.
- f- مسیر کامل هر فهرست را چاپ کنید، نه فقط نام اصلی آن.
- x- در میان فایل سیستم ها حرکت نمیکند.
- L- عمق بازگشت را به # محدود میکند.
- P REGEX- اما فقط فایل هایی را فهرست کنید که با REGEX مطابقت دارند.
- I REGE- فایل هایی را که با REGEX مطابقت دارند فهرست نکنید.
- --ignore-case- نادیده گرفتن حروف در حالت الگوی تطبیق.
- --prune- فهرست های خالی را از خروجی جدا میکند.
- --filelimit- فهرست هایی که حاوی # در فایل هستند را حذف میکند.
- o FILE- خروجی STDOUT را به FILE هدایت کنید.
- i- از تورفتگی در خروجی استفاده نمیکند.

مثال های دستور:

گزینه a-: نمایش همه فایل‌ها و فهرست‌ها، حتی آنهایی که با نقطه شروع می‌شوند) مخفف "all").

```
$tree -a
```

گزینه l-: نمایش عمق درخت مخفف "level".

```
$tree -l 3
```

گزینه h-: نمایش اندازه‌های فایل‌ها و فهرست‌ها به صورت قابل خواندن برای انسان) مخفف "human-readable").

```
$tree -h
```

گزینه p-: نمایش نام کامل فایل‌ها و فهرست‌ها) مخفف "print full path".

```
$tree -p
```

گزینه i-: نمایش نام فایل‌ها و فهرست‌ها به صورت ساده) مخفف "indent".

```
$tree -i
```

گزینه R-: نمایش درخت به صورت معکوس (از پایین به بالا)) مخفف "reverse".

```
$tree -R
```

گزینه N-: عدم نمایش پیوندهای سمبلیک) مخفف "no links").

```
$tree -N
```

گزینه L-: تعیین عمق درخت) مخفف "level".

```
$tree -L 2
```

گزینه R-: نمایش درخت به صورت معکوس (از پایین به بالا)) مخفف "reverse".

```
$tree -R
```

گزینه H-: تعیین آدرس پایه برای لینک‌ها در HTML خروجی مخفف "baseHREF".

```
$tree -H /var/www/html
```

گزینه T-: تعیین عنوان برای HTML خروجی) مخفف "title".

```
$tree -T "My Directory Tree"
```

گزینه :o-ذخیره خروجی در یک فایل (مخفف "output".

```
$tree -o output.txt
```

گزینه :P-فیلتر کردن فایل‌ها و فهرست‌ها بر اساس الگوی داده شده (مخفف "pattern".

```
$tree -P "*.txt"
```

دستور killall

killall یک سیگنال به تمام فرآیندهایی که هر یک از دستورات مشخص شده را اجرا می کنند ارسال می‌کند. اگر نام سیگنال مشخص نشده باشد، SIGTERM ارسال می شود. به طور کلی، دستور killall با دانستن نام فرآیند، تمام فرآیندها را از بین می برد. سیگنال ها را می توان با نام (مثلا HUP– یا SIGHUP–) یا با شماره (مثلا ۱–) یا با گزینه s– مشخص کرد. اگر نام دستور یک عبارت معمولی (گزینه r –) نباشد و دارای علامت اسلش (/) باشد، فرآیندهایی که آن فایل خاص را اجرا می کنند، مستقل از نامشان برای kill انتخاب می شوند. killall یک کد بازگشت صفر را برمی‌گرداند. اگر حداقل یک فرآیند برای هر دستور فهرست شده kill شده باشد، یا هیچ دستوری فهرست نشده باشد و حداقل یک فرآیند با معیارهای جستجو u– و Z– مطابقت داشته باشد. در غیر این صورت. killall غیر صفر برمی گرداند . برنامه killall هرگز خودش را kill (اما ممکن است سایر فرآیندهای killall را kill‌کند).

ساختار دستور:

```
$ sudo killall [OPTION]... NAME...
  $ killall -l, --list
$ killall -V, --version
```

گزینه های دستور:

- e – --exact نیاز به تطابق دقیق برای نام طولانی دارد.
- I– --ignore-case حساس به حروف کوچک و بزرگ مطابقت با نام فرآیند دارد.
- y – --younger-than برای kill کردن فرآیندهای جوانتر از TIME
- o – --older-than برای kill کردن فرآیندهای قدیمی‌تر از TIME
- i– --interactive قبل از kill درخواست تایید میگیرد.
- l– --list تمام نام های سیگنال شناخته شده را فهرست می‌کند
- q – quiet -- درحالت سکوت
- u – user USER -- فقط فرآیند(های) اجرا شده به عنوان کاربر USER متوقف میکند.
- v – اگر سیگنال با موفقیت ارسال شد، گزارش دقیقی ارائه میدهد
- w – منتظر می میاند،تا فرآیندها از بین بروند

مثال های دستور:

۱. تمام فرآیندهای منطبق با نام firefox را با SIGTERM ، killکنید :

```
$ killall firefox # OR killall -SIGTERM firefox # OR kilall -15 firefox
```

من توانستم برنامه‌های Wine (که فایل‌های exe ویندوزی هستند که روی لینوکس اجرا می‌شوند) را نیز از این طریق kill کنم.

```
$ killall gedit
```

۲. فهرست همه سیگنال های پشتیبانی شده:

```
$ killall -l
$  killall -l HUP INT QUIT ILL TRAP ABRT BUS FPE KILL USR1 SEGV USR2 PIPE
ALRM TERM STKFLT CHLD CONT STOP TSTP TTIN TTOU URG XCPU XFSZ
VTALRM PROF WINCH POLL PWR SYS
```

در مورد اعداد

```
$ for s in $(killall -l); do echo -n "$s " && kill -l $s; done
HUP 1
INT 2
QUIT 3
ILL 4
TRAP 5
ABRT 6
BUS 7
FPE 8
KILL 9
USR1 10
SEGV 11
USR2 12
PIPE 13
ALRM 14
TERM 15
STKFLT 16
CHLD 17
CONT 18
STOP 19
TSTP 20
TTIN 21
TTOU 22
URG 23
XCPU 24
XFSZ 25
VTALRM 26
PROF 27
WINCH 28
POLL 29
PWR 30
SYS 31
```

۳. برای جلوگیری از توقف ناخواسته، قبل از kill کردن سوال میپرسد:

```
$ killall -i conky
$ Kill conky(1685) ? (y/N)
```

۴. تمام فرآیندهای kill را منتظر میگذارد، تا فرآیندها از بین بروند.

```
$ killall -w conky
```

دستور login

دستور login به system برای ایجاد یک جلسه جدید با system استفاده می شود

ساختار دستور:

```
$ login [-p] [-h host] [-H] [-f username|username]
```

مثال های دستور :

اگر رمز عبور تعریف شده باشد، اعلان رمز عبور ظاهر می شود. رمز عبور خود را در این اعلان وارد کنید. برای استفاده از دستور login با همه گزینه های ذکر شده، می توانید از دستورات زیر استفاده کنید:

ورود با نام کاربری و رمز عبور معمولی:

```
$login
```

ورود با نام کاربری و رمز عبور و نمایش پیام راهنما:

```
$login -p
```

ورود به یک میزبان خاص با استفاده از نام کاربری و رمز عبور:

```
$login -h [نام میزبان]
```

ورود با نام کاربری و رمز عبور و ایجاد یک شل تمام عیار:(Full login shell)

```
$login -H
```

ورود به حساب کاربری مشخص شده و اجازه ورود به دیگر حساب های کاربری:

```
$login -f [نام کاربری|نام کاربری]
```

لطفا توجه داشته باشید که برای استفاده از دستور login، شما باید دسترسی لازم را به سرور یا سیستم مورد نظر داشته باشید و نام کاربری و رمز عبور معتبر را وارد کنید. همچنین برخی از گزینه ها ممکن است در برخی از سیستم ها و پلتفرم ها موجود نباشند.

دستور ulimit

شرح دستور

دستور ulimit یک ابزار خط فرمان در لینوکس است که برای تنظیم و نمایش محدودیت‌های منابع سیستم برای یک فرآیند یا کاربر استفاده می‌شود .این محدودیت‌ها می‌توانند شامل موارد زیر باشند:

- **حافظه :**حداکثر مقدار حافظه‌ای که یک فرآیند می‌تواند استفاده کند.
- **تعداد فایل‌های باز :**حداکثر تعداد فایل‌هایی که یک فرآیند می‌تواند به طور همزمان باز کند.
- **اندازه فایل هسته :**حداکثر اندازه فایلی که در صورت وقوع خطا ایجاد می‌شود.
- **تعداد پردازش‌ها :**حداکثر تعداد پردازش‌هایی که یک کاربر می‌تواند به طور همزمان اجرا کند.

با استفاده از این دستور، می‌توان از منابع سیستم به طور مؤثرتر استفاده کرد و از بروز مشکلاتی مانند اشغال بیش از حد حافظه یا ایجاد تعداد زیادی فرآیند جلوگیری کرد.

ساختار دستور:

```
ulimit [options] [limit]
```

- **options:**گزینه‌هایی برای تنظیم نوع محدودیت و مقدار آن.
- **limit:**مقدار عددی برای محدودیت.

کاربردهای دستور:

- **نمایش محدودیت‌های فعلی :**بدون هیچ گزینه‌ای، ulimit محدودیت‌های فعلی را نمایش می‌دهد.
- **تنظیم محدودیت‌ها :**با استفاده از گزینه‌های مختلف، می‌توان محدودیت‌های خاصی را تنظیم کرد.
- **تنظیم محدودیت‌های نرم و سخت :**محدودیت‌های نرم (soft limits) محدودیت‌هایی هستند که سیستم به طور پیش‌فرض اعمال می‌کند و می‌توان آن‌ها را به طور موقت افزایش داد .محدودیت‌های سخت (hard limits) محدودیت‌های بالاتری هستند که حتی با افزایش موقت نیز قابل تجاوز نیستند.

گزینه‌های مهم دستور:

- **a:**-تمام محدودیت‌ها را نمایش می‌دهد.
- **c:**-محدودیت اندازه فایل هسته (core file size) را تنظیم می‌کند.
- **d:**-محدودیت اندازه بخش داده (data segment size) را تنظیم می‌کند.
- **f:**-محدودیت اندازه فایل (file size) را تنظیم می‌کند.
- **n:**-حداکثر تعداد فایل‌های باز (open files) را تنظیم می‌کند.
- **m:**-حداکثر اندازه حافظه (memory size) را تنظیم می‌کند.
- **p:**-اندازه بافر لوله (pipe size) را تنظیم می‌کند.
- **s:**-اندازه پشته (stack size) را تنظیم می‌کند.
- **t:**-حداکثر زمان پردازنده (CPU time) را تنظیم می‌کند.
- **u:**-حداکثر تعداد پردازش‌ها (user processes) را تنظیم می‌کند.
- **v:**-حداکثر اندازه حافظه مجازی (virtual memory) را تنظیم می‌کند.
- **S:**-محدودیت نرم را تنظیم می‌کند.
- **H:**-محدودیت سخت را تنظیم می‌کند.

مثال‌های کاربردی

نمایش تمام محدودیت‌ها

```
ulimit -a
```

این دستور تمام محدودیت‌های فعلی سیستم را به صورت جدول‌وار نمایش می‌دهد.

تنظیم محدودیت اندازه فایل هسته

```
ulimit -c 1024
```

این دستور حداکثر اندازه فایلی که در صورت وقوع خطا ایجاد می‌شود را به ۱۰۲۴ بلوک تنظیم می‌کند.

تنظیم محدودیت اندازه بخش داده

```
ulimit -d unlimited
```

این دستور محدودیت اندازه بخش داده را نامحدود می‌کند.

تنظیم محدودیت اندازه فایل

```
ulimit -f 2048
```

این دستور حداکثر اندازه فایلی که یک فرآیند می‌تواند ایجاد کند را به ۲۰۴۸ بلوک تنظیم می‌کند.

تنظیم حداکثر تعداد فایل‌های باز

```
ulimit -n 4096
```

این دستور حداکثر تعداد فایل‌هایی که یک فرآیند می‌تواند به طور همزمان باز کند را به ۴۰۹۶ فایل تنظیم می‌کند.

تنظیم حداکثر اندازه حافظه

```
ulimit -m 2G
```

این دستور حداکثر اندازه حافظه‌ای که یک فرآیند می‌تواند استفاده کند را به ۲ گیگابایت تنظیم می‌کند.

تنظیم اندازه بافر لوله

```
ulimit -p 16384
```

این دستور اندازه بافر لوله را به ۱۶۳۸۴ بایت تنظیم می‌کند.

تنظیم اندازه پشته

```
ulimit -s 8192
```

این دستور اندازه پشته را به ۸۱۹۲ کیلوبایت تنظیم می‌کند.

تنظیم حداکثر زمان پردازنده

```
ulimit -t 60
```

این دستور حداکثر زمان پردازنده‌ای که یک فرآیند می‌تواند استفاده کند را به ۶۰ ثانیه تنظیم می‌کند.

تنظیم حداکثر تعداد پردازش‌ها

```
ulimit -u 256
```

این دستور حداکثر تعداد پردازش‌هایی که یک کاربر می‌تواند به طور همزمان اجرا کند را به ۲۵۶ پردازش تنظیم می‌کند.

تنظیم حداکثر اندازه حافظه مجازی

```
ulimit -v 4G
```

این دستور حداکثر اندازه حافظه مجازی که یک فرآیند می‌تواند استفاده کند را به ۴ گیگابایت تنظیم می‌کند.

تنظیم محدودیت نرم و سخت

تنظیم محدودیت نرم تعداد فایل‌های باز به ۱۰۲۴ و محدودیت سخت به ۲۰۴۸

```
ulimit -S -n 1024 -H -n 2048
```

این دستور محدودیت نرم تعداد فایل‌های باز را به ۱۰۲۴ فایل و محدودیت سخت را به ۲۰۴۸ فایل تنظیم می‌کند .این بدان معناست که به طور پیش‌فرض، یک فرآیند می‌تواند حداکثر ۱۰۲۴ فایل را باز کند، اما با استفاده از برخی مکانیزم‌ها می‌توان این محدودیت را تا ۲۰۴۸ فایل افزایش داد.

توجه:

- مقادیر عددی در مثال‌ها به عنوان نمونه آورده شده‌اند و شما می‌توانید آن‌ها را با توجه به نیاز خود تغییر دهید.
- برای تنظیم محدودیت‌ها به صورت دائمی، باید فایل‌های پیکربندی مربوطه (مانند `/etc/security/limits.conf`) را ویرایش کنید.
- تنظیم محدودیت‌های بسیار پایین می‌تواند باعث ایجاد مشکلاتی در اجرای برنامه‌ها شود.

مثال عملی :فرض کنید می‌خواهید حداکثر تعداد فایل‌هایی که یک کاربر می‌تواند به طور همزمان باز کند را به ۲۰۴۸ فایل محدود کنید و همچنین می‌خواهید مطمئن شوید که هیچ فرآیندی نتواند بیش از ۴ گیگابایت حافظه استفاده کند .برای این کار می‌توانید از دستورات زیر استفاده کنید:

```
ulimit -n 2048
ulimit -m 4G
```

با اجرای این دستورات، محدودیت‌های مورد نظر برای پوسته فعلی تنظیم می‌شوند .برای اعمال این محدودیت‌ها برای تمام کاربران، باید فایل‌های پیکربندی مربوطه را ویرایش کنید.

دستور lsof

Lsof به این معنی است که "فایلهای باز شده list Open file" استفاده می شود تا مشخص شود کدام پرونده چه پروسه ای باز است.همانطور که همه ما می دانیم linux / یونیکس همه چیز را به عنوان یک فایل در نظر می گیرند. اساساً ، این اطلاعات را برای یافتن فایل هایی که در هر فرآیند باز می شوند، می دهد .با یک بار مراجعه ، تمام فایل های باز موجود در کنسول خروجی را لیست می کند .این فقط نمی تواند فایل های معمولی را لیست کند اما می تواند یک فهرست ، یک فایل ویژه بلوک ، یک کتابخانه مشترک ، یک فایل خاص کاراکتر ، یک pipe معمولی ، یک pipe نامگذاری شده ، یک سوکت اینترنت ، یک سوکت دامنه UNIX و بسیاری دیگر را لیست کند .این می تواند با دستور grep ترکیب شود می تواند برای جستجوی پیشرفته و لیست استفاده کنید.

گزینه های دستور:

```
$ lsof -A A -c [c] +c [c] +|-d [d] +|-D [D] +|-e [s] +|-E +|-f [cfgGn] -F [f] -g [s] -i [i] -k [k] -K [k] +|-L [l]
+|-m [m] +|-M -o [o] -p [s] +|-r [t[m<fmt>]] -s [p:s] -S [t] -T [t] -u [s] +|-w -x [fl] -z [z] -Z [Z] -- [names]
```

مثال های دستور lsof

در مثال زیر لیستی از پرونده های باز را نشان می دهد که بعضی از آنها برای درک بهتر است که ستون هایی مانند دستور ، PID، USER، FD، TYPEو غیره را نمایش می دهد.

```
# lsof
```

فهرست فایل های باز شده مخصوص کاربر

دستور زیر لیست تمام فایل های باز شده کاربر hoseinرا نمایش می دهد.

```
# lsof -u  hosein
```

```
COMMAND PID USER FD TYPE DEVICE SIZE / OFF NAME NAME
sshd 1838  hosein  cwd DIR 253،0 4096 2 /
sshd 1838  hosein  rtd DIR 253،0 4096 2 /
sshd 1838  hosein  txt REG 253،0 532336 188129 / usr / sbin / sshd
sshd 1838  hosein  mem reg 253،0 19784 190237 /lib/libdl-2.12.so
```

فرآیندهای در حال اجرا در port خاص

برای پیداکردن تمام فرآیند در حال اجرا از پورت خاص، فقط از دستور زیر با گزینه i-استفاده کنید.مثال زیر تمام پروسه در حال اجرا پورت 22را لیست می کند.

```
# lsof -i TCP: 22
```

```
COMMAND PID USER FD TYPE DEVICE SIZE / OFF NAME NAME
sshd 1471 root 3u IPv4 12683 0t0 TCP *: ssh (listen)
sshd 1471 root 4u IPv6 12685 0t0 TCP *: ssh (listen)
```

لیست پورت TCP دامنه ۱-۱۰۲٤

برای لیست تمام روند در حال اجرا از فایل های باز از TCPپورت محدوده از 1024-1است.

```
# lsof -i TCP: 1-1024
```

```
COMMAND PID USER FD TYPE DEVICE SIZE / OFF NAME NAME
rpcbind 1203 rpc 11u IPv6 11336 0t0 TCP *: sunrpc (listen)
cupsd 1346 root 7u IPv4 12113 0t0 TCP localhost: ipp (listen)
sshd 1471 root 4u IPv6 12685 0t0 TCP *: ssh (listen)
master 1551 root 13u IPv6 12898 0t0 TCP localhost: smtp (listen)
sshd 1834 root 3r IPv4 15101 0t0 TCP 192.168.0.2:ssh-> 192.168.0.1:conclave-cpp (ESTABLISHED)
sshd 1838  hosein  3u IPv4 15101 0t0 TCP 192.168.0.2:ssh-> 192.168.0.1:conclave-cpp (ESTABLISHED)
sshd 1871 root 3r IPv4 15842 0t0 TCP 192.168.0.2:ssh-> 192.168.0.1:groove (ESTABLISHED)
httpd 1918 root 5u IPv6 15991 0t0 TCP *: http (LISTEN)
httpd 1918 root 7u IPv6 15995 0t0 TCP *: https (LISTEN)
```

کاربر را با علامت " رد کنید

در اینجا ما کاربر root را رد کردیم. می توانید یک کاربر خاص را با استفاده از '^'با دستور همانطور که در بالا نشان داده شد، حذف کنید.

```
# lsof -i -u ^ root
```

COMMAND PID USER FD TYPE DEVICE SIZE / OFF NAME NAME
rpcbind 1203 rpc 6u IPv4 11326 0t0 UDP *: sunrpc
rpcbind 1203 rpc 7u IPv4 11330 0t0 UDP *: 954
rpcbind 1203 rpc 8u IPv4 11331 0t0 TCP *: sunrpc (listen)

بدست آوردن افرادی که فایل ها و دستورات را دنبال می کنند

در زیر مثال نشان می دهد کاربر hosein از دستور مانند پینگ و etc /استفاده می کند.

```
# lsof -i -u  hosein
```

COMMAND PID USER FD TYPE DEVICE SIZE / OFF NAME NAME
bash 1839 hosein cwd DIR 253،0 12288 15 / etc

لیست تمام اتصالات شبکه

دستور زیر با گزینه '-i'لیست تمام اتصالات شبکه را 'LISTENING & ESTABLISHED' را نشان می دهد.

```
# lsof -i
```

جستجو توسط PID

مثال زیر PID1 را نشان می دهد .

```
# lsof -p 1
```

COMMAND PID USER FD TYPE DEVICE SIZE / OFF NAME NAME
init 1 root cwd DIR 253،0 4096 2 /
init 1 root rtd DIR 253،0 4096 2 /
init 1 root txt REG 253،0 145180 147164 / sbin / init

سایر گزینه ها

a – ، all-- شامل همه خطوط برای CPUهای آنلاین و آفلاین در خروجی است

. به عنوان **مثال:**

```
$  lsof -a
```

b – ، online-- خروجی را به CPUهای آنلاین محدود میکند .این گزینه ممکن است فقط همراه با گزینه e– یا p– مشخص شود.

به عنوان **مثال:**

```
$ lscpu -b
```

c – ، offline-- خروجی را به CPUهای آفلاین محدود میکند. این گزینه ممکن است فقط همراه با گزینه e– یا p– مشخص شود.

e – ، [=list] extended-- نمایش اطلاعات CPU در فرمت قابل خواندن توسط انسان. به عنوان **مثال:**

```
 $ lsof -e
```

A A: – نمایش همه مراجعه‌کننده‌های باز شده توسط هر فایل

```
lsof -A A
```

c [c]– :نمایش فایل‌های باز شده توسط فرآیند با نام c

```
$lsof -c نام فرآیند
```

+c [c]: نمایش فایل‌های باز شده توسط همه فرآیندهای با نام شامل c

```
$lsof +c نام فرآیند
```

+|-d [d]: نمایش فایل‌های باز شده با میزان تاخیر بیشتر یا کمتر از d

```
$lsof +d میزان تاخیر
$lsof -d میزان تاخیر
```

+|-D [D]: نمایش فایل‌های باز شده با میزان توسعه بیشتر یا کمتر از D

```
$lsof +D میزان توسعه
$lsof -D میزان توسعه
```

+|-e [s]: نمایش فایل‌های باز شده با پسوند s

```
$lsof +e پسوند
$lsof -e پسوند
```

+|-E: نمایش فایل‌های باز شده توسط همه فرآیندها بدون پرونده اضافی

```
$lsof +E
$lsof -E
```

+|-f [cfgGn]: نمایش فایل‌های باز شده با حالت‌های مشخص (c: character devices, f: regular files, g: NFS files, G: non-NFS files)

```
$lsof +f [cfgGn]
$lsof -f [cfgGn]
```

-F [f]: خروجی دستور lsof را به فرمت مورد نظر تغییر دهید

```
$lsof -F فرمت
```

-g [s]: نمایش فایل‌های باز شده توسط گروه با شماره s

```
$lsof -g [شماره گروه]
```

-i [i]: نمایش فایل‌های باز شده توسط پورت i

```
$lsof -i [شماره پورت]
```

-k [k]: نمایش فایل‌های باز شده توسط فایل‌های اسکریپتی با نام k

```
$lsof -k [نام فایل اسکریپت]
```

-K [k]: بستن تمامی فایل‌های باز شده توسط فایل‌های اسکریپتی با نام k

```
$lsof -K [نام فایل اسکریپت]
```

+|-L [l]: نمایش فایل‌های باز شده با سطح لاگ‌ها بیشتر یا کمتر از l

```
$lsof +L [سطح لاگ]
lsof -L [سطح لاگ]
```

+|-m [m]: نمایش فایل‌های باز شده با نوع موجودیت بیشتر یا کمتر از m

```
$lsof +m [نوع موجودیت]
```

```
$lsof -m [نوع موجودیت]
```

+|-M :نمایش فایل‌های باز شده توسط همه موجودیت‌ها بدون موجودیت اضافی

```
$lsof +M
$lsof -M
```

-o [o] :نمایش فایل‌های باز شده توسط فرآیند با شماره o

```
$lsof -o [شماره فرآیند]
```

-p [s] :نمایش فایل‌های باز شده توسط فرآیند با شماره s

```
$lsof -p [شماره فرآیند]
```

+|-r [t[m<fmt>]] :نمایش فایل‌های باز شده با زمان تغییر بیشتر یا کمتر از (t با قالب زمانی اختیاری (fmt

```
$lsof +r [زمان]m[فرمت زمان]
$lsof -r [زمان]m[فرمت زمان]
```

-s [p:s] :نمایش فایل‌های باز شده با وضعیت پروتوکل p و وضعیت سوکت s

```
$lsof -s [پروتوکل:وضعیت سوکت]
```

-S [t] :نمایش فایل‌های باز شده توسط فرآیند با شماره t

```
$lsof -S [شماره فرآیند]
```

-T [t] :نمایش فایل‌های باز شده با نوع t

```
$lsof -T [نوع]
```

-u [s] :نمایش فایل‌های باز شده توسط کاربر با نام s

```
$lsof -u [نام کاربر]
```

+|-w :نمایش فایل‌های باز شده توسط فرآیندهای قابل نوشتن

```
$lsof +w
$lsof -w
```

-x [fl] :نمایش فایل‌های باز شده توسط فایل‌های اطلاعات ترافیک شبکه (f) با شماره فایل (l)

```
$lsof -x [شماره فایل]
```

-z [z] :نمایش فایل‌های باز شده توسط نوع فشرده‌سازی z

```
$lsof -z [نوع فشرده‌سازی]
```

-Z [Z] :نمایش فایل‌های باز شده توسط لیبل امنیتی Z

```
$lsof -Z [لیبل امنیتی]
```

-- [names] :نمایش فایل‌های باز شده با نام‌های مشخص

نمایش فایل‌های باز شده با نام file1.txt و file2.txt :

```
$lsof -- file1.txt file2.txt
```

نمایش فایل‌های باز شده در فهرست /home/user/ :

```
$lsof -- /home/user/
```

نمایش فایل‌های باز شده با نام file1.txt در فهرست /home/user/ :

```
$lsof -- /home/user/file1.txt
```

نمایش فایل‌های باز شده با نام file1.txt و file2.txt در فهرست /home/user/ :

```
$lsof -- /home/user/file1.txt /home/user/file2.txt
```

دستور fuser

دستور **fuser** برای نمایش پردازه‌هایی که از یک فایل یا سوکت استفاده می‌کنند، استفاده می‌شود. این دستور می‌تواند برای پیدا کردن پردازه‌هایی که یک فایل را باز کرده‌اند و یا از یک سوکت استفاده می‌کنند، بسیار مفید باشد.

ساختار دستور:

```
fuser [-fuv] [-a|-s] [-4|-6] [-c|-m|-n space] [ -k [-i] [-M] [-w]
    [-SIGNAL] ] name ...
    fuser -l
    fuser -V
```

مثال‌های دستور:

۱. نمایش پردازه‌هایی که از یک فایل استفاده می‌کنند:

این دستور نمایش پردازه‌هایی که از یک فایل خاص استفاده می‌کنند را نمایش می‌دهد.

```
fuser /path/to/file
```

۲. نمایش پردازه‌هایی که از یک سوکت استفاده می‌کنند:

این دستور نمایش پردازه‌هایی که از یک سوکت خاص استفاده می‌کنند را نمایش می‌دهد.

```
fuser -v /path/to/socket
```

۳. نمایش پردازه‌هایی که از یک پورت خاص استفاده می‌کنند:

این دستور نمایش پردازه‌هایی که از یک پورت خاص استفاده می‌کنند را نمایش می‌دهد.

```
fuser -n tcp 80
```

۴. نمایش اطلاعات جزئی‌تر درباره پردازه‌ها:

این دستور اطلاعات بیشتری درباره پردازه‌هایی که از یک فایل یا سوکت استفاده می‌کنند را نمایش می‌دهد.

```
fuser -m /path/to/file
```

۵. کشف پردازه‌های خاص بر اساس نام کاربری:

این دستور پردازه‌هایی که توسط یک نام کاربری خاص ایجاد شده‌اند و از یک فایل یا سوکت استفاده می‌کنند را نمایش می‌دهد.

```
fuser -u username /path/to/file
```

۶. نمایش پردازه‌ها به صورت تعاملی:

این دستور به صورت تعاملی اطلاعات پردازه‌هایی که از یک فایل یا سوکت استفاده می‌کنند را نمایش می‌دهد.

```
fuser -i /path/to/file
```

دستور tuptime

Tuptime ابزاری است برای گزارش زمانی زمان آپدیت (تاریخی و آماری یک سیستم عاملLinux ، که آن را بین بازیابی نگه می دارد.خروجی پیشرفته تر را فراهم می کند.

این ابزار می تواند:

۱. ثبت، نام هسته مورد استفاده
۲. ثبت اولین بوت
۳. تعداد راه اندازی های system. (تعداد بوت شدن سیستم)
۴. از زمان اولین بوت شدن، درصد زمان آپدیت و خرابی را محاسبه میکند.
۵. محاسبه بزرگترین، کوتاه ترین و متوسط زمان بوت وخرابی.
۶. محاسبه زمان خرابی.
۷. چاپ زمان فعلی
۸. چاپ جدول فرمت شده یا فهرست با بیشتر مقادیر قبلی ذخیره شده است.

گزینه های دستور:

- p – ، نمایش uptime در قالب زیبا
- h – ، این متن راهنما را نمایش دهید
- s – ، سیستم از آن زمان، در قالب yyyy-mm-dd HH:MM:SS
- V – ، نمایش اطلاعات نسخه و خروج

مثال های دستور:

نصب ابزار tuptime

ابتدا باید با اجرای دستور زیرباید به clone کردن مخزن بپردازید:

```
git clone https://github.com/rfrail3/tuptime.git
```

```
tecmint@tecmint.com ~> git clone https://github.com/rfrail3/tuptime.git
Cloning into 'tuptime'...
remote: Counting objects: 1021, done.
remote: Compressing objects: 100% (61/61), done.
remote: Total 1021 (delta 15), reused 0 (delta 0), pack-reused 953
Receiving objects: 100% (1021/1021), 456.58 KiB | 88.00 KiB/s, done.
Resolving deltas: 100% (468/468), done.
Checking connectivity... done.
tecmint@tecmint.com ~>
```

سپس به آخرین فهرست داخل فهرست tuptimeبروید.بعد، اسکریپت tuptimeرا داخل آخرین فهرست به / usr / bin کپی کنید و مجوز اجرایی را همانطور که نشان داده شده تنظیم کنید.

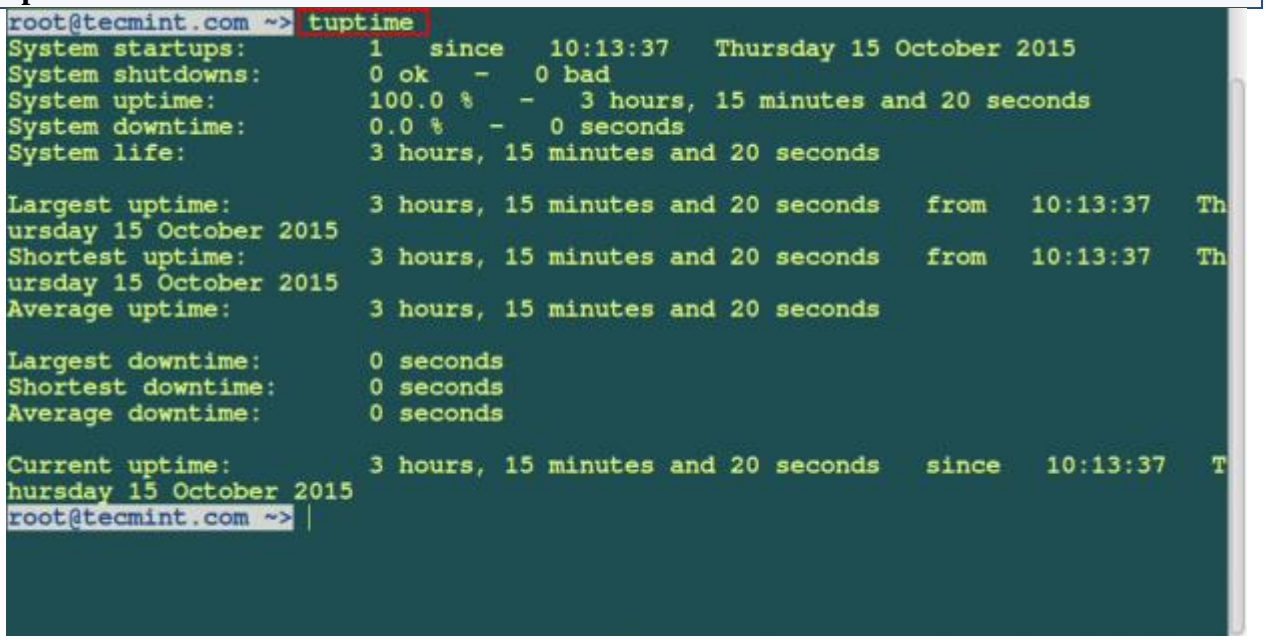
```
$ cd tuptime / last
$ sudo cp tuptime / usr/ bin/tuptime
$ sudo chmod ugo + x / usr / bin / tuptime
```

```
tecmint@tecmint.com ~> cd tuptime/latest/
tecmint@tecmint.com ~/tuptime/latest> sudo cp tuptime /usr/bin/tuptime
[sudo] password for tecmint:
tecmint@tecmint.com ~/tuptime/latest> sudo chmod ugo+x /usr/bin/tuptime
tecmint@tecmint.com ~/tuptime/latest>
```

چگونه از tuptime استفاده کنم؟

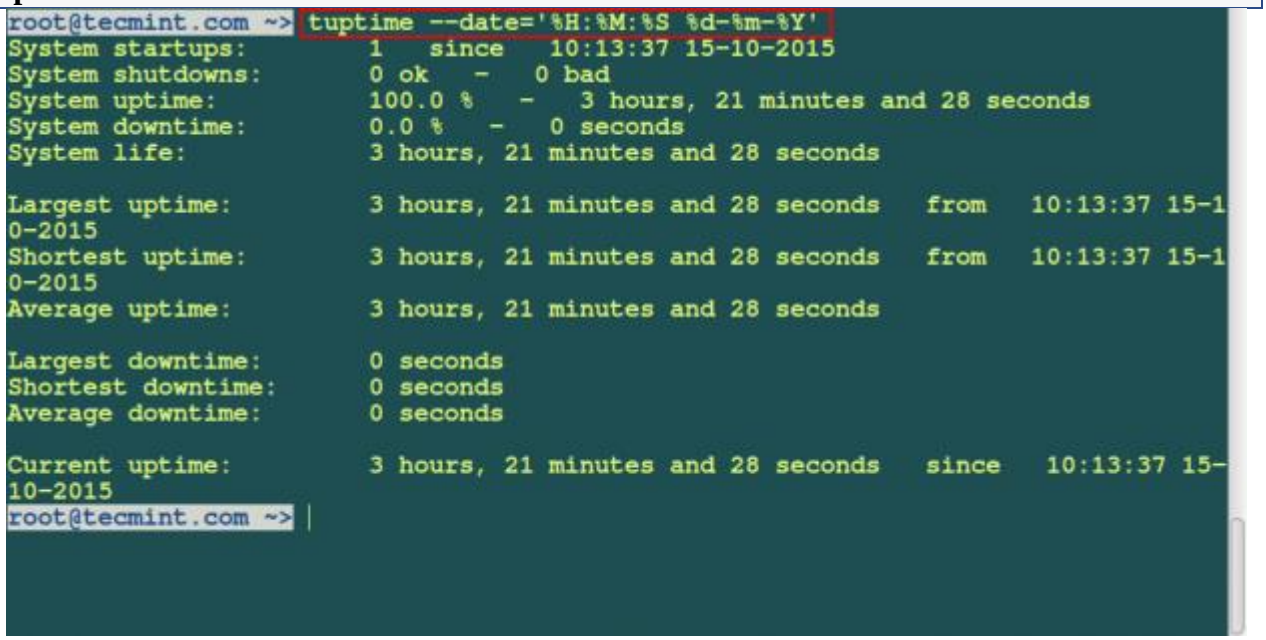
هنگامی که بدون هیچ گزینه ای tuptimeرا اجرا می کنید، یک صفحه نمایش شبیه به یکی از زیر دریافت می کنید.

 # tuptime



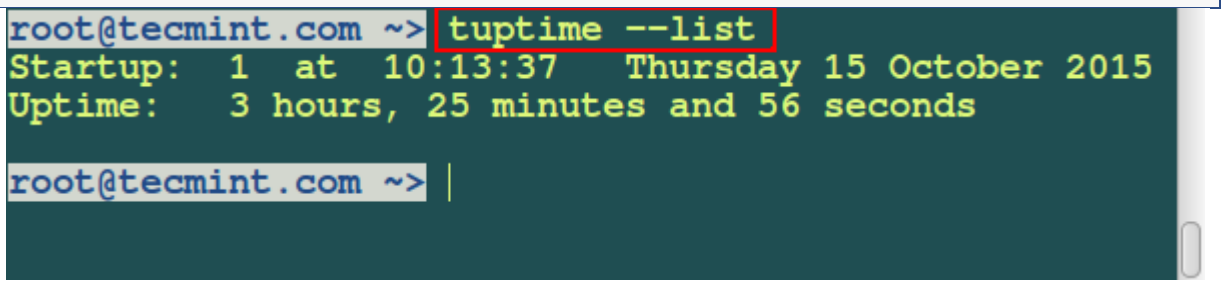
می توانید خروجی را با تاریخ و زمان به صورت زیر نمایش دهید.

 # tuptime - date = '٪H:٪M:٪S٪d-٪m-٪Y'



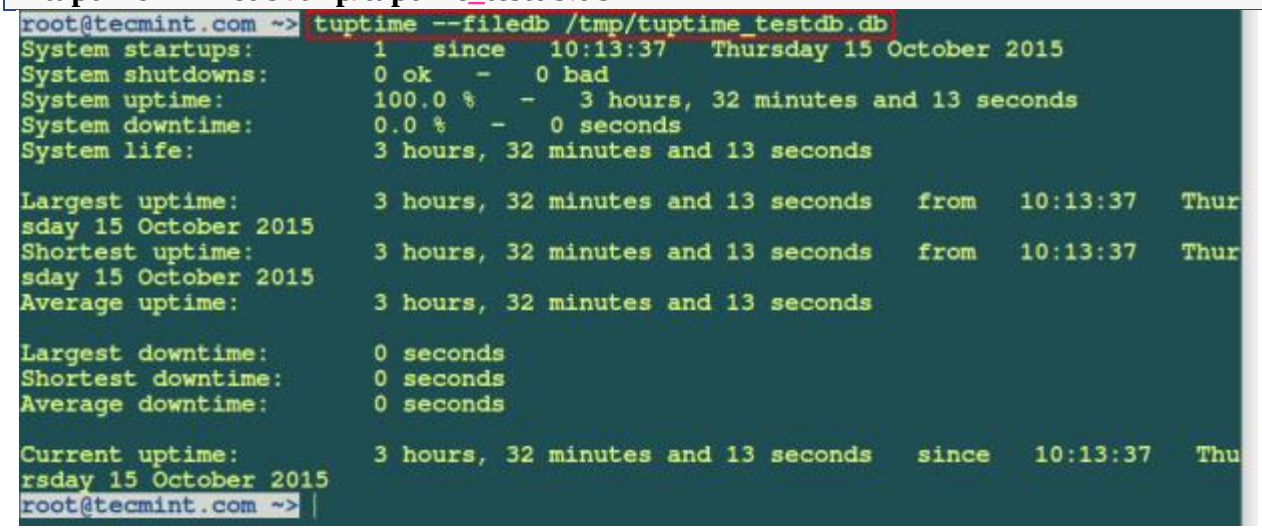
برای چاپ زمان فعال بودن live سیستم می توانید این دستور زیر را اجرا کنید:

```
# tuptime - list
```



می توانید یک فایل پایگاه داده جایگزین را به صورت زیر ایجاد کنید.پایگاه داده[۱] در فرمت SQLite ایجاد خواهد شد.

```
# tuptime --filedb /tmp/tuptime_testdb.db
```



دستور Watch

Watchیک دستور linux است که به امکان می دهد که یک دستور یا برنامه را به صورت دوره ای اجرا کنید و همچنین خروجی را روی صفحه نمایش نشان می دهد.این به این معنی است که می توانید خروجی برنامه را در زمان ببینید.به طور پیش فرض ساعت اجرا دستور / برنامه هر ۲ ثانیه.فاصله را می توان به راحتی تغییر داد تا نیازهای را برآورده کند.

گزینه های دستور:

- -b, --beep: اگر خروجی دستور دارای خروجی غیرصفر باشد، صدای بیپ (صدای هشدار) تولید می‌شود.
- -c, --color: ترجمه و نمایش توالی‌های رنگی و استایل ANSI.
- -C, --no-color: عدم ترجمه و نمایش توالی‌های رنگی و استایل ANSI.
- -d, --differences[=permanent]: نمایش تفاوت‌ها بین به‌روزرسانی‌های متوالی. اگر آرگومان دائمی اختیاری مشخص شود، watch تمام تغییرات از اولین اجرا را نشان می‌دهد.
- -e, --errexit: توقف به‌روزرسانی‌ها در صورت خطا در دستور و خروج پس از فشردن یک کلید. کد خروجی watch برابر با کد خروجی دستور است. اگر سیگنال n علت پایان دستور باشد، کد خروجی برابر با ۱۲۸ + n خواهد بود.
- -g, --chgexit: خروجی از دستور را زمانی که تغییر می‌کند، متوقف می‌کند.

[۱] database

- -n, --interval seconds: تعیین فاصله زمانی بین به‌روزرسانی‌ها. مقادیر کوچک‌تر از ۰,۱ و بزرگ‌تر از ۲۶۷۸۴۰۰ (۳۱ روز) به مقادیر مربوطه تبدیل می‌شوند.
- -p, --precise: اجرای دستور به‌صورت --interval ثانیه پس از شروع اجرای قبلی آن، به جای --interval ثانیه پس از اتمام اجرای قبلی آن. اگر زمان اجرای دستور بیش از --interval ثانیه طول بکشد، در هر دو حالت منتظر می‌ماند.
- -q, --equexit <cycles>: خروجی از دستور را زمانی که تغییر نمی‌کند، متوقف می‌کند.
- -r, --no-rerun: اجرای برنامه در تغییر اندازه ترمینال انجام نمی‌شود، و خروجی برنامه در زمان اجرای منظم بعدی دوباره ظاهر می‌شود.
- -s, --shotsdir: مسیر ذخیره تصاویر اسکرین‌شات.
- -t, --no-title: عدم نمایش عنوان.

مثال های دستور:

```
watch -b <command>
```

اگر خروجی دستور دارای خروجی غیرصفر باشد، صدای بیپ (صدای هشدار) تولید می‌شود.

```
watch -b ls -l
watch -c <command>
```

ترجمه و نمایش توالی‌های رنگی و استایل ANSI.

```
watch -c ps aux
watch -C <command>
```

عدم ترجمه و نمایش توالی‌های رنگی و استایل ANSI.

```
watch -C df -h
watch -d <command>
```

نمایش تفاوت‌ها بین به‌روزرسانی‌های متوالی.

```
watch -d free -m
watch -e <command>
```

توقف به‌روزرسانی‌ها در صورت خطا در دستور و خروج پس از فشردن یک کلید.

```
watch -e cat nonexistent_file
watch -n 5 <command>
```

تعیین فاصله زمانی بین به‌روزرسانی‌ها به ۵ ثانیه.

```
watch -n 5 date
watch -p <command>
```

اجرای دستور به‌صورت --interval ثانیه پس از شروع اجرای قبلی آن.

```
watch -p -n 10 ls
watch -q 3 <command>
```

خروجی از دستور را زمانی که تغییر نمی‌کند، متوقف می‌کند پس از ۳ بار تغییر.

```
watch -q 3 sensors
watch -r <command>
```

اجرای برنامه در تغییر اندازه ترمینال انجام نمی‌شود.

```
watch -r top
watch -s /path/to/directory <command>
```

مسیر ذخیره تصاویر اسکرین‌شات.

```
watch -s ~/Desktop screenshots.sh
```

```
watch -t <command>
```

عدم نمایش عنوان.

```
watch -t date
```

فصل دوازدهم

فصل دوازدهم

مدیریت سرویس ها و اجرای سطوح در لینوکس

دستور chkconfig

از دستور **chkconfig** برای لیست کردن تمام سرویس های موجود و مشاهده یا به روزرسانی تنظیمات سطح اجرای آنها استفاده می شود .به زبان ساده، از این لیست برای فهرست کردن اطلاعات راه اندازی فعلی سرویس ها یا هر سرویس خاص، به روزرسانی تنظیمات سطح پایین سرویس و افزودن یا حذف سرویس از مدیریت استفاده می شود.

گزینه های دستور:

```
Synopsis:
chkconfig --list [name]
chkconfig --add name
chkconfig --del name
chkconfig --override name
chkconfig [--level levels] name
chkconfig [--level levels] name
```

مثال های دستور:

- **برای لیست کردن وضعیت فعلی همه سرویس های سیستم.**

```
chkconfig --list
```

```
[root@localhost ~]# chkconfig --list

Note: This output shows SysV services only and does not include native
      systemd services. SysV configuration data might be overridden by native
      systemd configuration.

      If you want to list systemd services use 'systemctl list-unit-files'.
      To see services enabled on particular target use
      'systemctl list-dependencies [target]'.

netconsole      0:off   1:off   2:off   3:off   4:off   5:off   6:off
network         0:off   1:off   2:on    3:on    4:on    5:on    6:off
rhnsd           0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@localhost ~]#
```

- **برای مشاهده وضعیت فعلی یک سرویس خاص.**

```
chkconfig --list rhnsd
```

```
[root@localhost ~]# chkconfig --list rhnsd

Note: This output shows SysV services only and does not include native
      systemd services. SysV configuration data might be overridden by native
      systemd configuration.

      If you want to list systemd services use 'systemctl list-unit-files'.
      To see services enabled on particular target use
      'systemctl list-dependencies [target]'.

rhnsd           0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@localhost ~]#
```

- **برای غیرفعال کردن سرویس :**به طور پیش فرض ۲ ۳ ۴ ۵ سطح اجرا تحت تأثیر این دستور قرار می گیرد تا فقط برخی از سطوح اجرا را غیرفعال کنید ، ویژگی level– را اضافه کنید و سطح اجرا را اضافه کنید.

chkconfig rhnsd off

```
[root@localhost ~]# chkconfig rhnsd off
[root@localhost ~]# chkconfig --list rhnsd

Note: This output shows SysV services only and does not include native
      systemd services. SysV configuration data might be overridden by native
      systemd configuration.

      If you want to list systemd services use 'systemctl list-unit-files'.
      To see services enabled on particular target use
      'systemctl list-dependencies [target]'.

rhnsd           0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@localhost ~]#
```

- **فعال کردن سرویس :**به طور پیش فرض ۲ ۳ ۴ ۵ سطح اجرا تحت تأثیر این دستور قرار می گیرد تا فقط برخی از سطوح اجرا را فعال کند ، ویژگی level- را اضافه کنید و سطح اجرا را اضافه کنید.

chkconfig rhnsd on

```
[root@localhost ~]# chkconfig rhnsd on
[root@localhost ~]# chkconfig --list rhnsd

Note: This output shows SysV services only and does not include native
      systemd services. SysV configuration data might be overridden by nat
      systemd configuration.

      If you want to list systemd services use 'systemctl list-unit-files'
      To see services enabled on particular target use
      'systemctl list-dependencies [target]'.

rhnsd           0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@localhost ~]#
```

- **برای حذف یک سرویس**

chkconfig del rhnsd

```
[root@localhost ~]# chkconfig --del rhnsd
[root@localhost ~]# chkconfig --list

Note: This output shows SysV services only and does not include native
      systemd services. SysV configuration data might be overridden by native
      systemd configuration.

      If you want to list systemd services use 'systemctl list-unit-files'.
      To see services enabled on particular target use
      'systemctl list-dependencies [target]'.

netconsole      0:off   1:off   2:off   3:off   4:off   5:off   6:off
network         0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@localhost ~]#
```

- **برای افزودن سرویس**

chkconfig add rhnsd

```
[root@localhost ~]# chkconfig --add rhnsd
[root@localhost ~]# chkconfig --list

Note: This output shows SysV services only and does not include native
      systemd services. SysV configuration data might be overridden by native
      systemd configuration.

      If you want to list systemd services use 'systemctl list-unit-files'.
      To see services enabled on particular target use
      'systemctl list-dependencies [target]'.

netconsole      0:off   1:off   2:off   3:off   4:off   5:off   6:off
network         0:off   1:off   2:on    3:on    4:on    5:on    6:off
rhnsd           0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@localhost ~]#
```

مدیر سیستم systemd

نحوه استفاده از Systemctl برای مدیریت خدمات و واحدهای Systemd

systemdیک سیستم init و مدیر سیستم است که به طور گسترده به استاندارد جدیدی برای توزیع های لینوکس تبدیل شده است .به دلیل پذیرش زیاد آن، آشنایی با systemdآن به دردسر می ارزد، زیرا مدیریت سرورها را به میزان قابل توجهی آسان تر می کند .یادگیری و استفاده از ابزارها و شیاطین موجود systemdبه شما کمک می‌کند تا قدرت، انعطاف‌پذیری و قابلیت‌هایی را که ارائه می‌دهد بهتر درک کنید یا حداقل به شما کمک می‌کند تا کار خود را با حداقل دردسر انجام دهید. ابزار مدیریت مرکزی برای کنترل سیستم init است . لطفاً توجه داشته باشید که اگرچه systemdبه سیستم init پیش‌فرض برای بسیاری از توزیع‌های لینوکس تبدیل شده است، اما به طور جهانی در همه توزیع‌ها اجرا نمی‌شود .همانطور که این آموزش را دنبال می کنید، اگر ترمینال شما خطا bash: systemctl is not installedرا صادر کند، احتمالاً دستگاه شما یک سیستم init متفاوت نصب کرده است.

مدیریت سرویس ها

هدف اساسی یک سیستمinit ، مقداردهی اولیه اجزایی است که باید پس از بوت شدن هسته لینوکس راه اندازی شوند) که به طور سنتی به عنوان اجزای "Userland" شناخته می شود .(سیستم init همچنین برای مدیریت سرویس ها و دیمون ها برای سرور در هر نقطه ای که سیستم در حال اجرا است استفاده می شود .با در نظر گرفتن این موضوع، ما با برخی از عملیات مدیریت خدمات اولیه شروع خواهیم کرد. در systemd هدف اکثر اقدامات "واحدها" هستند که منابعی هستند که systemdمی دانند چگونه مدیریت کنند .واحدها بر اساس نوع منبعی که نشان می دهند دسته بندی می شوند و با فایل هایی که به عنوان فایل های واحد شناخته می شوند تعریف می شوند .نوع هر واحد را می توان از پسوند انتهای فایل استنباط کرد.برای وظایف مدیریت خدمات، واحد هدف واحدهای خدماتی خواهد بود که دارای فایل های واحد با پسوند service. با این حال، برای اکثر دستورات مدیریت سرویس، می‌توانید serviceپسوند را کنار بگذارید ، زیرا systemdبه اندازه کافی هوشمندانه است که بدانید احتمالاً می‌خواهید هنگام استفاده از دستورات مدیریت سرویس، روی یک سرویس کار کنید.

شروع و توقف سرویس

برای شروع یک systemdسرویس، اجرای دستورالعمل ها در فایل واحد سرویس، از startدستور استفاده کنید .اگر به عنوان یک کاربر غیر ریشه در حال اجرا هستید، باید از sudoآن استفاده کنید زیرا این روی وضعیت سیستم عامل تأثیر می گذارد:

```
sudo systemctl start application.service
```

همانطور که در بالا ذکر کردیم، systemdمی‌داند که به دنبال service.*فایل‌ها برای دستورات مدیریت سرویس می‌گردد، بنابراین این دستور به راحتی می‌تواند به صورت زیر تایپ شود:

```
sudo systemctl start application
```

اگرچه ممکن است از قالب فوق برای مدیریت عمومی استفاده کنید، اما برای وضوح، ما از .service پسوند برای بقیه دستورات استفاده می کنیم تا در مورد هدفی که روی آن کار می کنیم صریح باشیم.

برای متوقف کردن یک سرویس در حال اجرا، می توانید به جای آن از stopدستور استفاده کنید:

```
sudo systemctl stop application.service
```

راه اندازی مجدد و بارگذاری مجدد سرویس

برای راه اندازی مجدد یک سرویس در حال اجرا، می توانید از restartدستور زیر استفاده کنید:

```
sudo systemctl restart application.service
```

اگر برنامه مورد نظر بتواند فایل های پیکربندی خود را مجدداً بارگیری کند (بدون راه اندازی مجدد)، می توانید reloadدستور شروع آن فرآیند را صادر کنید:

```
sudo systemctl reload application.service
```

اگر مطمئن نیستید که سرویس قابلیت بارگذاری مجدد پیکربندی خود را دارد، می توانید:

reload-or-restartدستور را صادر کنید .در صورت موجود بودن، این پیکربندی را دوباره بارگیری می‌کند .در غیر این صورت، سرویس را دوباره راه اندازی می کند تا پیکربندی جدید انتخاب شود:

```
sudo systemctl reload-or-restart application.service
```

فعال و غیرفعال کردن سرویس

دستورات بالا برای شروع یا توقف خدمات در جلسه جاری مفید هستند .برای اطلاع systemdاز شروع خودکار سرویس ها در هنگام بوت، باید آنها را فعال کنید.

برای راه اندازی یک سرویس در هنگام بوت، از enableدستور زیر استفاده کنید:

```
sudo systemctl enable application.service
```

این یک پیوند نمادین از کپی سیستم از فایل سرویس (معمولاً در:

/lib/systemd/systemیا (/etc/systemd/systemبه مکانی روی دیسک که در آن systemdفایل‌های شروع خودکار را جستجو می‌کند ایجاد می‌کند (معمولاً .ما بعداً در این راهنما به این خواهیم پرداخت که هدف چیست./etc/systemd/system/some_target.target.wants

برای غیرفعال کردن شروع خودکار سرویس، می توانید تایپ کنید:

```
sudo systemctl disable application.service
```

با این کار پیوند نمادینی که نشان می دهد سرویس باید به طور خودکار شروع شود حذف می کند. به خاطر داشته باشید که فعال کردن یک سرویس آن را در جلسه فعلی شروع نمی کند .اگر می خواهید سرویس را راه اندازی کنید و همچنین آن را در هنگام بوت فعال کنید، باید هر دو دستور `start` و `enable`را صادر کنید.

بررسی وضعیت سرویس

برای بررسی وضعیت یک سرویس در سیستم خود، می توانید از `status`دستور زیر استفاده کنید :

```
systemctl status application.service
```

با این کار وضعیت سرویس، سلسله مراتب cgroup و چند خط ورود اولیه به شما ارائه می شود. به عنوان مثال ، هنگام بررسی وضعیت یک سرورNginx ، ممکن است خروجی مانند زیر را مشاهده کنید:

```
Output
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2015-01-27 19:41:23 EST; 22h ago
 Main PID: 495 (nginx)
   CGroup: /system.slice/nginx.service
           ├─495 nginx: master process /usr/bin/nginx -g pid /run/nginx.pid; error_log
stderr;
           └─496 nginx: worker process
Jan 27 19:41:23 desktop systemd[1]: Starting A high performance web server and a reverse
proxy server...
Jan 27 19:41:23 desktop systemd[1]: Started A high performance web server and a reverse
proxy server.
```

این به شما یک نمای کلی از وضعیت فعلی برنامه می دهد و شما را از هر گونه مشکل و هر اقدامی که ممکن است لازم باشد مطلع می کند. همچنین روش هایی برای بررسی وضعیت های خاص وجود دارد .به عنوان مثال ، برای بررسی اینکه آیا یک واحد در حال حاضر فعال است (در حال اجرا)، می توانید از `is-active`دستور استفاده کنید :

```
systemctl is-active application.service
```

این حالت واحد فعلی را که معمولاً `active`یا `inactive`. کد خروج در صورت فعال بودن "۰" خواهد بود و تجزیه نتیجه را در اسکریپت های پوسته ساده تر می کند. برای اینکه ببینید واحد فعال است، می توانید از `is-enabled`دستور زیر استفاده کنید :

```
systemctl is-enabled application.service
```

این خروجی را نشان می دهد که آیا سرویس است `enabled`یا `disabled`بسته به پاسخ به سؤال دستور، کد خروج را روی "۰" یا "۱" تنظیم می کند. بررسی سوم این است که آیا دستگاه در وضعیت خراب است یا خیر .این نشان می دهد که مشکلی در راه اندازی واحد مورد نظر وجود داشته است:

```
systemctl is-failed application.service
```

این بازگشت `active`آن است که اگر به درستی در حال اجرا و یا `failed`اگر یک خطا رخ داده است .اگر واحد عمدا متوقف شده باشد، ممکن است برگردد `unknown`یا `inactive`. وضعیت خروج "۰" نشان می دهد که یک خرابی رخ داده است و وضعیت خروج "۱" نشان دهنده هر وضعیت دیگری است.

نمای کلی وضعیت سیستم

برای بررسی وضعیت فعلی سیستم چندان مفید نیستند .تعدادی `systemctl`دستور وجود دارد که این اطلاعات را ارائه می دهد.

لیست واحدهای فعلی

برای دیدن لیستی از تمام واحدهای فعالی که systemdدر مورد آنها می دانند، می توانیم از list-unitsدستور استفاده کنیم :

```
systemctl list-units
```

با این کار لیستی از تمام واحدهایی که systemdدر حال حاضر روی سیستم فعال هستند را به شما نشان می دهد .خروجی چیزی شبیه به این خواهد بود:

Output

```
UNIT                                    LOAD   ACTIVE SUB     DESCRIPTION
atd.service                             loaded active running ATD daemon
avahi-daemon.service                    loaded active running Avahi mDNS/DNS-SD Stack
dbus.service                            loaded active running D-Bus System Message Bus
dcron.service                           loaded active running Periodic Command Scheduler
dkms.service                            loaded active exited  Dynamic Kernel Modules
System
getty@tty1.service                      loaded active running Getty on tty1
```

خروجی دارای ستون های زیر است:

UNIT:نام systemdواحد

LOAD:آیا پیکربندی واحد توسط تجزیه و تحلیل شده systemdاست یا خیر .پیکربندی واحدهای بارگذاری شده در حافظه نگهداری می شود.

ACTIVE:حالت خلاصه ای در مورد فعال بودن واحد .این معمولاً یک روش نسبتاً ابتدایی برای تشخیص اینکه آیا واحد با موفقیت شروع به کار کرده است یا خیر است.

SUB:این یک حالت سطح پایین تر است که اطلاعات دقیق تری را در مورد واحد نشان می دهد .این اغلب بر اساس نوع واحد، حالت، و روش واقعی که واحد در آن اجرا می شود، متفاوت است.

از آنجایی که list-unitsدستور به طور پیش فرض فقط واحدهای فعال را نشان می دهد، تمام ورودی های بالا loadedدر ستون LOADو activeدر ستون ACTIVE نشان داده می شوند .این نمایشگر در واقع رفتار پیش‌فرض systemctlهنگام فراخوانی بدون دستورات اضافی است، بنابراین اگر systemctlبدون آرگومان تماس بگیرید، همان چیزی را خواهید دید :

دستور systemctl

systemctl با اضافه کردن پرچم‌های اضافی، می‌توانیم بگوییم که اطلاعات مختلف را خروجی کنیم .به عنوان مثال ، برای مشاهده تمام واحدهایی که systemdبارگیری شده اند (یا تلاش کرده اند)، صرف نظر از اینکه در حال حاضر فعال هستند یا خیر، می توانید از -all-پرچم مانند زیر استفاده کنید:

```
systemctl list-units --all
```

این نشان می دهد هر واحدی که systemdبارگذاری شده یا تلاش کرده است، صرف نظر از وضعیت فعلی آن در سیستم .برخی از واحدها پس از اجرا غیرفعال می شوند و برخی واحدهایی که systemdسعی در بارگیری داشتند ممکن است روی دیسک پیدا نشده باشند. می توانید از پرچم های دیگر برای فیلتر کردن این نتایج استفاده کنید .برای مثال ، می‌توانیم از --state=پرچم برای نشان دادن حالت‌هایLOAD ، ACTIVEیا SUB که می‌خواهیم ببینیم استفاده کنیم .شما باید --allپرچم را نگه دارید تاsystemctl واحدهای غیرفعال نمایش داده شوند:

```
systemctl list-units --all --state=inactive
```

یکی دیگر از فیلترهای رایج فیلتر است `--type=`. ما می‌توانیم بگوییم `systemctl` که فقط واحدهایی از نوع مورد علاقه ما نمایش داده می‌شوند. به عنوان مثال ، برای مشاهده فقط واحدهای خدمات فعال، می‌توانیم از موارد زیر استفاده کنیم:

```
systemctl list-units --type=service
```

فهرست کردن همه فایل های واحد

این list-unitsدستور فقط واحدهایی را نشان می دهد که systemdسعی در تجزیه و بارگذاری در حافظه داشته اند. از آنجایی که systemdفقط واحدهایی را می خواند که فکر می کند به آن نیاز دارد، لزوماً همه واحدهای موجود در سیستم را شامل نمی شود. برای مشاهده هر فایل واحد موجود در systemdمسیرها، از جمله مواردی که systemdتلاشی برای بارگیری نکرده‌اند، می‌توانید به list-unit-filesجای آن از دستور استفاده کنید:

```
systemctl list-unit-files
```

واحدها بازنمایی منابعی هستند که `systemd` در مورد آن می دانند. از آنجایی `systemd` که لزوماً همه تعاریف واحد را در این نما نخوانده است، فقط اطلاعات مربوط به خود فایل ها را ارائه می دهد. خروجی دارای دو ستون است: فایل واحد و وضعیت.

Output

```
UNIT FILE                                 STATE
proc-sys-fs-binfmt_misc.automount         static
dev-hugepages.mount                       static
dev-mqueue.mount                          static
proc-fs-nfsd.mount                        static
proc-sys-fs-binfmt_misc.mount             static
sys-fs-fuse-connections.mount             static
sys-kernel-config.mount                   static
sys-kernel-debug.mount                    static
tmp.mount                                 static
var-lib-nfs-rpc_pipefs.mount              static
org.cups.cupsd.path                       enabled
```

. . .

حالت معمولاً `static`, `disabled`, `enabled` یا `masked` خواهد بود . در این زمینه، static به این معنی است که فایل واحد شامل یک `install` نیست که برای فعال کردن یک واحد استفاده می شود. به این ترتیب، این واحدها را نمی توان فعال کرد. معمولاً این بدان معنی است که واحد یک عمل یکباره انجام می دهد یا فقط به عنوان وابستگی واحد دیگری استفاده می شود و نباید به تنهایی اجرا شود.

مدیریت واحد

تا کنون با سرویس ها کار کرده ایم و اطلاعات مربوط به واحد و فایل های واحدی را که اطلاع دارد را نمایش `systemd` می دهیم. با این حال، ما می توانیم با استفاده از برخی دستورات اضافی، اطلاعات دقیق تری در مورد واحدها پیدا کنیم.

نمایش فایل واحد

برای نمایش فایل واحدی که `systemd` در سیستم آن بارگذاری شده است، می توانید از `cat` استفاده کنید (این در `systemd` نسخه ۲۰۹ اضافه شده است . به عنوان مثال ، برای دیدن فایل واحد `atd` شبح زمان‌بندی، می‌توانیم تایپ کنیم:

```
systemctl cat atd.service
```

Output

```
[Unit]
Description=ATD daemon
[Service]
Type=forking
```

```
ExecStart=/usr/bin/atd
[Install]
WantedBy=multi-user.target
```

خروجی فایل واحدی است که در systemdفرآیند در حال اجرا شناخته شده است .اگر اخیراً فایل‌های واحد را تغییر داده‌اید یا اگر گزینه‌های خاصی را در یک قطعه فایل واحد لغو کرده‌اید، می‌تواند مهم باشد.

نمایش وابستگی ها

برای دیدن درخت وابستگی یک واحد، می توانید از list-dependenciesدستور زیر استفاده کنید :

```
systemctl list-dependencies sshd.service
```

این یک نقشه سلسله مراتبی را نشان می دهد که وابستگی هایی که باید برای راه اندازی واحد مورد نظر با آنها برخورد کرد .وابستگی‌ها، در این زمینه، شامل آن دسته از واحدهایی می‌شوند که توسط واحدهای بالای آن مورد نیاز یا مورد نیاز هستند.

Output

```
sshd.service
├─system.slice
└─basic.target
  ├─microcode.service
  ├─rhel-autorelabel-mark.service
  ├─rhel-autorelabel.service
  ├─rhel-configure.service
  ├─rhel-dmesg.service
  ├─rhel-loadmodules.service
  ├─paths.target
  ├─slices.target
```

برای نشان دادن وابستگی های معکوس (واحدهایی که به واحد مشخص شده بستگی دارند)،

بررسی ویژگی های واحد

برای مشاهده ویژگی های سطح پایین یک واحد، می توانید از showدستور استفاده کنید .با این کار لیستی از ویژگی هایی که برای واحد مشخص شده با استفاده از key=valueفرمت تنظیم شده اند نمایش داده می شود:

```
systemctl show sshd.service
```

Output

```
Id=sshd.service
Names=sshd.service
Requires=basic.target
Wants=system.slice
WantedBy=multi-user.target
Conflicts=shutdown.target
Before=shutdown.target multi-user.target
After=syslog.target network.target auditd.service systemd-journald.socket basic.target
system.slice
Description=OpenSSH server daemon
```

اگر می خواهید یک ویژگی را نمایش دهید، می توانید p-پرچم را با نام ویژگی ارسال کنید .به عنوان مثال ، برای مشاهده تداخل هایی که sshd.serviceواحد دارد، می توانید تایپ کنید:

```
systemctl show sshd.service -p Conflicts
```

```
Conflicts=shutdown.target
```

ماسک کردن یک واحد

ما در بخش مدیریت خدمات دیدیم که چگونه یک سرویس را متوقف یا غیرفعال کنیم، اما systemdهمچنین این قابلیت را دارد که با پیوند دادن یک واحد به عنوان کاملاً شروع نشده، به صورت خودکار یا دستی، آن را علامت گذاری کند /dev/nullبه این حالت ماسک کردن واحد می گویند و با maskدستور زیر امکان پذیر است :

```
sudo systemctl mask nginx.service
```

این کار تا زمانی که سرویس Nginx ماسک است از راه اندازی خودکار یا دستی جلوگیری می کند.

اگر تیک بزنید list-unit-files، خواهید دید که سرویس اکنون به صورت ماسک شده فهرست شده است:

```
systemctl list-unit-files
```

Output

```
. . .
kmod-static-nodes.service              static
ldconfig.service                       static
mandb.service                          static
messagebus.service                     static
nginx.service                          masked
quotaon.service                        static
rc-local.service                       static
rdisc.service                          disabled
rescue.service                         static
```

اگر بخواهید سرویس را راه اندازی کنید، پیامی مانند زیر را مشاهده خواهید کرد:

```
sudo systemctl start nginx.service
```

Output

```
Failed to start nginx.service: Unit nginx.service is masked.
```

برای از بین بردن ماسک یک واحد و در دسترس قرار دادن مجدد آن برای استفاده، از unmaskدستور استفاده کنید :

```
sudo systemctl unmask nginx.service
```

این دستگاه را به حالت قبلی خود برمی گرداند و امکان راه اندازی یا فعال کردن آن را فراهم می کند.

ویرایش فایل های واحد

در حالی که فرمت خاص برای فایل های واحد خارج از محدوده این آموزش است، systemctlمکانیزم های داخلی را برای ویرایش و اصلاح فایل های واحد در صورت نیاز به انجام تنظیمات ارائه می دهد .این قابلیت در systemdنسخه ۲۱۸ اضافه شده است . این editدستور به طور پیش فرض یک قطعه فایل واحد را برای واحد مورد نظر باز می کند:

```
sudo systemctl edit nginx.service
```

این یک فایل خالی خواهد بود که می تواند برای لغو یا اضافه کردن دستورالعمل ها به تعریف واحد استفاده شود .یک فهرست در فهرست ایجاد می شود /etc/systemd/systemکه حاوی نام واحد با d.ضمیمه است .به عنوان مثال ، برای nginx.serviceیک فهرست به نام nginx.service.dایجاد خواهد شد. در این فهرست، یک قطعه به نام ایجاد می شود override.conf. هنگامی که واحد بارگیری systemdمی شود، در حافظه، قطعه لغو با فایل واحد کامل ادغام می شود .دستورالعمل های قطعه بر دستورالعمل های موجود

در فایل واحد اصلی ارجحیت دارند .اگر می خواهید به جای ایجاد یک قطعه، فایل واحد کامل را ویرایش کنید، می توانید full--پرچم را ارسال کنید:

```
sudo systemctl edit --full nginx.service
```

با این کار فایل واحد فعلی در ویرایشگر بارگیری می شود، جایی که می توان آن را تغییر داد .هنگامی که ویرایشگر خارج می شود، فایل تغییر یافته در نوشته می شود `/etc/systemd/system/`، که بر تعریف واحد سیستم (معمولاً در جایی یافت می شود) اولویت دارد `/lib/systemd/system`.

برای حذف هرگونه افزودنی که ایجاد کرده اید، `.d`فهرست پیکربندی واحد یا فایل سرویس تغییر یافته را از `/etc/systemd/system`.به عنوان مثال ، برای حذف یک قطعه، می توانیم تایپ کنیم:

```
sudo rm -r /etc/systemd/system/nginx.service.d
```

برای حذف یک فایل واحد اصلاح شده کامل، باید تایپ کنیم:

```
sudo rm /etc/systemd/system/nginx.service
```

پس از حذف فایل یا فهرست، باید `systemd`فرآیند را مجدداً بارگیری کنید تا دیگر تلاشی برای ارجاع به این فایل‌ها نداشته باشد و به استفاده از کپی‌های سیستم برگردد .می توانید این کار را با تایپ کردن انجام دهید:

```
sudo systemctl daemon-reload
```

تنظیم (Runlevel)با target

target فایل های واحد ویژه ای هستند که وضعیت سیستم یا نقطه همگام سازی را توصیف می کنند .مانند سایر واحدها، فایل هایی که هدف ها را تعریف می کنند را می توان با پسوند آنها شناسایی کرد که در این مورد می `.target`باشد .اهداف به خودی خود کار زیادی انجام نمی دهند، بلکه برای گروه بندی واحدهای دیگر با هم استفاده می شوند .این می تواند به منظور رساندن سیستم به حالت های خاص مورد استفاده قرار گیرد، درست مانند سایر سیستم های اولیه که از سطوح اجرا استفاده می کنند .آنها به عنوان مرجع برای زمانی که توابع خاصی در دسترس هستند استفاده می شوند و به شما امکان می دهند به جای واحدهای جداگانه مورد نیاز برای تولید آن حالت، حالت مورد نظر را مشخص کنید .به عنوان مثال ، علامتی وجود دارد `swap.target`که برای نشان دادن آماده بودن swap برای استفاده استفاده می شود .

دریافت و تنظیم target پیش فرض

این `systemd`فرآیند دارای یک هدف پیش فرض است که هنگام بوت کردن سیستم از آن استفاده می کند .ارضای آبشار وابستگی ها از آن هدف واحد، سیستم را به وضعیت مطلوب می رساند .برای پیدا کردن هدف پیش فرض برای سیستم خود، تایپ کنید:

```
systemctl get-default
```

```
Output
multi-user.target
```

اگر می خواهید یک هدف پیش فرض متفاوت تنظیم کنید، می توانید از `set-default`.به عنوان مثال ، اگر یک دسکتاپ گرافیکی نصب کرده اید و می خواهید سیستم به طور پیش فرض در آن بوت شود، می توانید هدف پیش فرض خود را بر این اساس تغییر دهید:

```
sudo systemctl set-default graphical.target
```

فهرست کردن target موجود

با تایپ کردن زیر می توانید لیستی از **target** موجود در سیستم خود را دریافت کنید:

```
systemctl list-unit-files --type=target
```

بر خلاف runlevel ها، چندین هدف می توانند در یک زمان فعال باشند .یک هدف فعال نشان می دهد که systemdسعی کرده است تمام واحدهای متصل به هدف را راه اندازی کند و دوباره سعی نکرده آنها را خراب کند .برای مشاهده همه اهداف فعال، تایپ کنید:

```
systemctl list-units --type=target
```

جداسازی target

می توان تمام واحدهای مرتبط با یک هدف را شروع کرد و تمام واحدهایی را که بخشی از درخت وابستگی نیستند متوقف کرد .دستوری که برای انجام این کار نیاز داریم، به طور مناسب، نامیده می شود isolate. این شبیه به تغییر سطح اجرا در سایر سیستم های init است .به عنوان مثال ، اگر در یک محیط گرافیکی با graphical.targetفعال کار می کنید، می توانید سیستم گرافیکی را خاموش کنید و سیستم را با ایزوله کردن در حالت خط دستور چند کاربره قرار دهید multi-user.target. آنجایی که graphical.targetبستگی دارد multi-user.target، اما نه برعکس، همه واحدهای گرافیکی متوقف خواهند شد. ممکن است بخواهید قبل از انجام این روش به وابستگی های هدفی که در حال جداسازی هستید نگاهی بیندازید تا مطمئن شوید که خدمات حیاتی را متوقف نمی کنید:

```
systemctl list-dependencies multi-user.target
```

هنگامی که از واحدهایی که زنده نگه داشته می شوند راضی هستید، می توانید با تایپ کردن هدف، هدف را ایزوله کنید:

```
sudo systemctl isolate multi-user.target
```

استفاده از کلیدهای میانبر برای رویدادهای مهم

اهدافی برای رویدادهای مهم مانند خاموش کردن یا راه اندازی مجدد تعریف شده است .با این حال، systemctlچند میانبر نیز دارد که کمی عملکرد اضافی را اضافه می کند.

به عنوان مثال ، برای قرار دادن سیستم در حالت نجات (تک کاربر)، فقط می توانید از rescueدستور به جای isolate rescue.target:

```
sudo systemctl rescue
```

دستورات زیر هشداری به همه کاربرانی که وارد سیستم شده اند را در مورد رویداد فراهم می کند.

برای توقف سیستم می توانید از haltدستور زیر استفاده کنید:

```
sudo systemctl halt
```

برای شروع خاموش کردن کامل، می توانید از poweroffدستور زیر استفاده کنید:

```
sudo systemctl poweroff
```

راه اندازی مجدد را می توان با rebootدستور زیر شروع کرد:

```
sudo systemctl reboot
```

همه اینها به کاربران هشدار می دهند که رویداد در حال رخ دادن است، چیزی که فقط اجرا یا جداسازی هدف انجام نمی دهد .توجه داشته باشید که اکثر ماشین‌ها دستورات کوتاه‌تر و متعارف‌تری را برای این عملیات به هم پیوند می‌دهند تا به درستی با systemd.

به عنوان مثال ، برای راه اندازی مجدد سیستم، معمولاً می توانید تایپ کنید:

```
sudo reboot
```

سطوح اجرایی

سطوح اجرایی، حالت آغازین و کل سیستم است که خدمات سیستم را فعال می کند و مشخص میکند .سطح اجرا با اعداد مشخص می شود. هر زمان که سیستم LINUX بوت می شود ، در مرحله اول فرایند **init**شروع میشود که در واقع وظیفه اجرای اسکریپت های دیگر را دارد که عمدتا شامل اولیه سازی سخت افزار شما ، ایجاد شبکه ، شروع رابط گرافیکی است. اکنون **init** ابتدا سطح اجرای پیش فرض سیستم را پیدا می کند تا بتواند اسکریپت های شروع مربوط به سطح اجرای پیش فرض را اجرا کند. به راحتی می توان یک **runlevel**را به عنوان حالتی که سیستم شما وارد می کند در نظر گرفت ، مانند این که اگر یک سیستم در حالت تک کاربر باشد ، دارای یک **runlevel 1**خواهد بود در حالی که اگر سیستم در حالت چند کاربره باشد ، دارای **runlevel 5**خواهد بود.

به عبارت دیگر ، یک سطح اجرا می تواند به عنوان یک عدد صحیح یک رقمی از پیش تعیین شده برای تعریف حالت عملیاتی سیستم عامل LINUX یا UNIX شما تعریف شود .هر سطح اجرا پیکربندی سیستم متفاوتی را تعیین می کند و امکان دسترسی به ترکیبات مختلف فرایندها را فراهم می کند . نکته مهمی که در اینجا باید به آن توجه شود این است که با توجه به سیستم عامل تفاوت هایی در دو سطح وجود دارد .هسته استاندارد LINUXاز این هفت سطح مختلف مختلف پشتیبانی می کند:

0 -سیستم متوقف می شود یعنی سیستم بدون هیچ گونه فعالیت با خیال راحت خاموش می شود.

1 -حالت کاربر تنها.

2 -حالت کاربر چندگانه بدون سیستم فایل NFSشبکه.

3 -حالت کاربر چندگانه تحت رابط خط فرمان و نه تحت رابط کاربر گرافیکی.

4 -قابل تعریف توسط کاربر.

5 -حالت کاربر چندگانه تحت GUI رابط کاربری گرافیکی و این سطح استاندارد برای اکثر سیستم های مبتنی بر LINUX است.

6 - راه اندازی مجدد سیستم که برای راه اندازی مجدد سیستم استفاده می شود.

به طور پیش فرض ، بیشتر سیستم های مبتنی بر LINUX به سطح ۳ یا سطح ۵ اجرا می شوند .علاوه بر سطح استاندارد ، کاربران می توانند سطح اجرا شده از پیش تعیین شده را اصلاح کرده یا حتی جدیدی را با توجه به نیاز ایجاد کنند 2 Runlevels و ۴ برای سطح کاربر تعریف شده و 0 runlevel و ۶ برای توقف و راه اندازی مجدد سیستم استفاده می شوند. بدیهی است که اسکریپت های شروع برای هر سطح اجرا با انجام کارهای مختلف متفاوت خواهند بود .این اسکریپت های شروع مربوط به هر سطح اجرا را می توان در پرونده های ویژه موجود در زیر فایل rc یافت .در فایل /etc/rc.dیا مجموعه ای از پرونده ها به نام rc.0 ، rc.1 ، rc.2 ، rc.3 ، rc.4 ، rc.5و rc.6وجود دارد یا مجموعه ای از فایل ها نام rc0.d، rc1.d، rc2.d، rc3.d، rc4.d، rc5.dو . rc6.d به عنوان مثال، سطح اجرای ۱ اسکریپت آغازین خود را در پرونده /etc/rc.d/rc.1 یا هر پرونده در فایل /etc/rc.d/rc1.d دارد.

تغییر سطح اجرا

init برنامه ای است که تغییر سطح اجرا را می توان با استفاده از دستور **telinit** فراخوانی کرد . به عنوان مثال ، برای تغییر سطح اجرا از ۳ به سطح ۵ که در واقع اجازه می دهد GUI در حالت چند کاربر شروع شود ، از دستور **telinit** می توان به عنوان زیر استفاده کرد:

سطح اجرا از ۳ تا ۵

```
telinit 5
```

توجه:تغییر سطح اجرا یک وظیفه برای کاربر ریشه است و نه کاربر عادی ، به همین دلیل لازم است که برای اجرای موفقیت آمیز دستور telinit فوق به عنوان کاربر فوق العاده وارد سیستم شوید یا می توانید از دستور **sudo**استفاده کنید :

استفاده از sudo برای اجرایtelinit

```
sudo telinit 5
```

دستور halt

این دستور در لینوکس برای آموزش سخت افزار برای متوقف کردن همه عملکردهای پردازنده استفاده می شود .اصولاً سیستم را ریبوت یا متوقف می کند .اگر سیستم در سطح 0 یا ۶ اجرا شده باشد یا از دستور with -force استفاده کند ، منجر به راه اندازی مجدد سیستم می شود در غیر این صورت منجر به خاموش شدن می شود.

ساختار دستور:

```
halt [OPTION]...
```

گزینه های دستور:

گزینه	شرح
-f, –force	خاموش کردن را فراخوانی نمی کند.
-w, –wtmp-only	این امر خاموش کردن یا مجدداً را راه اندازی نمی کند اما logهای مربوط به خاموش کردن را در فایل / var / log / wtmp می نویسد.
-p –poweroff	به عنوان خاموش کردن استفاده میشود.
–verbose	هنگام راه اندازی مجدد پیام های می دهد که به رفع اشکال در هنگام خاموش کردن کمک می کند.

فایل ها:

/ var / log / wtmp از یک رکورد جدید در سطح اجرا برای زمان خاموش کردن تشکیل شده است.

/ var / run / utmp هنگام خواندن سطح اجرای فعلی ، با ضبط زمان خاموش شدن به روز می شود.

مثال ۱ :توقف عملکرد CPU در سیستم.

```
$ halt
```

مثال ۲ :برای خاموش کردن سیستم با استفاده از دستور.stop

```
$halt -p
```

مثال ۳ :دستور stop با گزینه w- برای نوشتن رکورد خاموش کردن.

```
$halt  -w
```

دستورات متوقف ، خاموش و راه اندازی مجدد درLinux

در زیر موارد معمول استفاده از توقف ، خاموش و راه اندازی مجدد وجود دارد.

halt: به سخت افزار دستور می دهد عملکردهای CPU را متوقف کند.

poweroff به سیستم دستور می دهد تا خاموش شود.

reboot سیستم را مجدداً راه اندازی یا راه اندازی مجدد می کند.

دستورات فوق فقط توسط کاربر فوق العاده قابل اجرا هستند زیرا این اقدامات شامل توقف سخت افزار سیستم است .اگر کاربر در عنوان عضو وارد نشده اید کاربر فوق العاده پس از آن دستور sudo دستور را می توان مورد استفاده برای اجرای این دستورات . این به سخت افزار دستور می دهد تا تمام عملکردهای پردازنده را متوقف کند.

```
// syntax of halt command halt [OPTION]...
```

از توقف به راحتی می توان برای متوقف کردن، خاموش کردن سیستم و راه اندازی مجدد سیستم استفاده کرد:

برای متوقف کردن عملکردهای پردازنده به سخت افزار دستور می دهد.

```
halt
```

خاموش شدن سیستم

```
halt -p
```

ریستارت شدن سیستم

```
halt --reboot
```

با این کار کاربر می تواند سیستم را از خط فرمان خاموش کند .با استفاده از توقف تمام عملکردهای پردازنده متوقف می شود و سیستم به حالتی وارد می شود که کاربر بتواند سطح پایین نگهداری را انجام دهد.

poweroff سیگنال ACPI ارسال می کند که به سیستم دستور می دهد تا خاموش شود .در اینجا دستور دستور poweroff آمده است:

ساختار دستور:

```
poweroff [OPTION] ...
```

از Poweroff به راحتی می توان برای **متوقف کردن** ، **خاموش کردن** و **راه اندازی مجدد** سیستم استفاده کرد:

قدرت خاموش سیستم

```
poweroff
```

فصل سیزدهم

فصل سیزدهم

مدیریت زمان و تاریخ

دستور cal

دستور cal یک تقویم فرمت شده را در ترمینال نمایش می‌دهد. اگر هیچ گزینه ای مشخص نشده باشد، cal ماه جاری را با برجسته شدن روز جاری نمایش می‌دهد.

ساختار دستور:

```
$ cal [general options] [-jy] [[month] year]
```

گزینه های دستور:

- h– تاریخ امروز را نماش نمیدهد.
- m month– یک ماه برای نمایش مشخص کنید. تعیین کننده ماه یک نام کامل ماه (مثلاً فوریه)، مخفف ماه حداقل سه حرف (مثلاً فوریه)، یا یک عدد (مثلاً ۲) است. اگر شما. یک عدد را مشخص کنید و به دنبال آن حرف "f" یا "p"، ماه سال بعد یا سال قبل را به ترتیب نمایش دهید. به عنوان **مثال**، m 2f– فوریه سال آینده را نمایش می‌دهد. یک سال برای نمایش مشخص کنید. به عنوان **مثال**، y 2023- نمایش می‌دهد.
- A num- ماه و چهار ماه بعد از این یکی

مثال های دستور:

۱.تقویم این ماه را با برجسته شدن امروز نمایش دهید.

```
$ cal
```

۲. مانند دستور قبلی، اما امروز را نشان نمیدهد.

```
$ cal -h
```

۳. ماه گذشته، این ماه و ماه آینده نمایش داده شود.

```
$ cal -3
```

۴. نمایش تقویم کل این سال

```
$ cal -y
```

۵. نمایش کل تقویم سال ۲۰۲۲

```
$ cal -y  2023
```

۶. همانند دستور قبلی

```
$ cal  2023
```

۷. نمایش تقویم برای دسامبر سال جاری.

```
$ cal -m [December, Dec, or 12]
```

۱۰. تقویم دسامبر ۲۰۲۰۰ را نمایش دهید.

```
$ cal 12 2022
```

دستور hwclock

hwclockهمچنین ***Real Time Clock (RTC)*** نامیده می شود ، ابزاری برای دسترسی به ساعت سخت افزاری است .ساعت سخت افزاری مستقل از سیستم عامل (سیستم عامل) مورد استفاده شما است و حتی هنگام خاموش بودن دستگاه نیز کار می کند .ساعت سخت افزاری را ساعت BIOS نیز می نامند .همچنین کاربر می تواند تاریخ و ساعت ساعت سخت افزاری را از BIOS تغییر دهد .ساعت

سخت افزاری مقادیر دوم ، دقیقه ، ساعت ، روز ، ماه و سال را ذخیره می کند **hwclock** .ابزار موجب صرفه جویی در تنظیمات آن را در/etc/adjtime فایل، که ایجاد شده است که کاربر باعث تغییر اول.

ساختار دستور:

hwclock [گزینه] [تابع] ...]

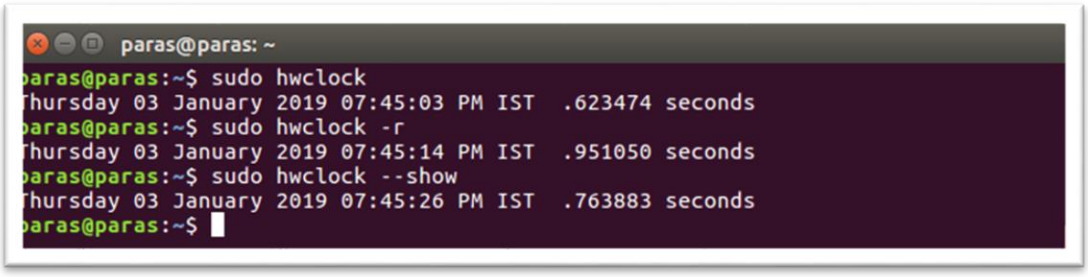
گزینه های دستور:

- **r- یا show–** زمان RTC را نشان می دهد.
- **get–** برای نمایش زمان RTC اصلاح شده با رانش استفاده می شود.
- **set–** برای تنظیم RTC با توجه به date– استفاده می شود.
- **s-یا hctosys–** زمان سیستم را از RTC تنظیم می کند.
- **w-یا systohoc–** این RTC را از زمان سیستم یعنی برعکس s– ، hctosys–تنظیم می کند.
- **systz–** برای ارسال پیکربندی های جدول زمانی به هسته استفاده می شود.
- **a- adjust–** RTC را تنظیم کنید تا رانش سیستماتیک را در نظر بگیرید.
- **predict–** زمان رانده شده RTC را با توجه به **date–** پیش بینی می کند.
- **u-یا utc–** نشان می دهد که مقیاس زمانی RTC UTC است.
- **l- یا localtime–**نشان می دهد که مقیاس زمانی RTC محلی است.
- **D-یا debug–**این برای نمایش اطلاعات اشکال زدایی استفاده می شود .اطلاعات با توجه به خواسته های دستور hwclock نشان داده می شوند.
- **V-یا version–** اطلاعات نسخه را نمایش داده و از آن خارج شوید.
- **h-یا help–** نمایش متن راهنما و خروج.

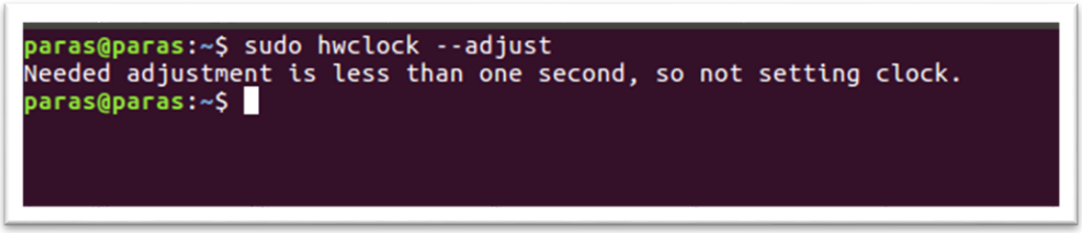
توجه :از **sudo** با دستور hwclock استفاده کنید.

مثال های دستور:

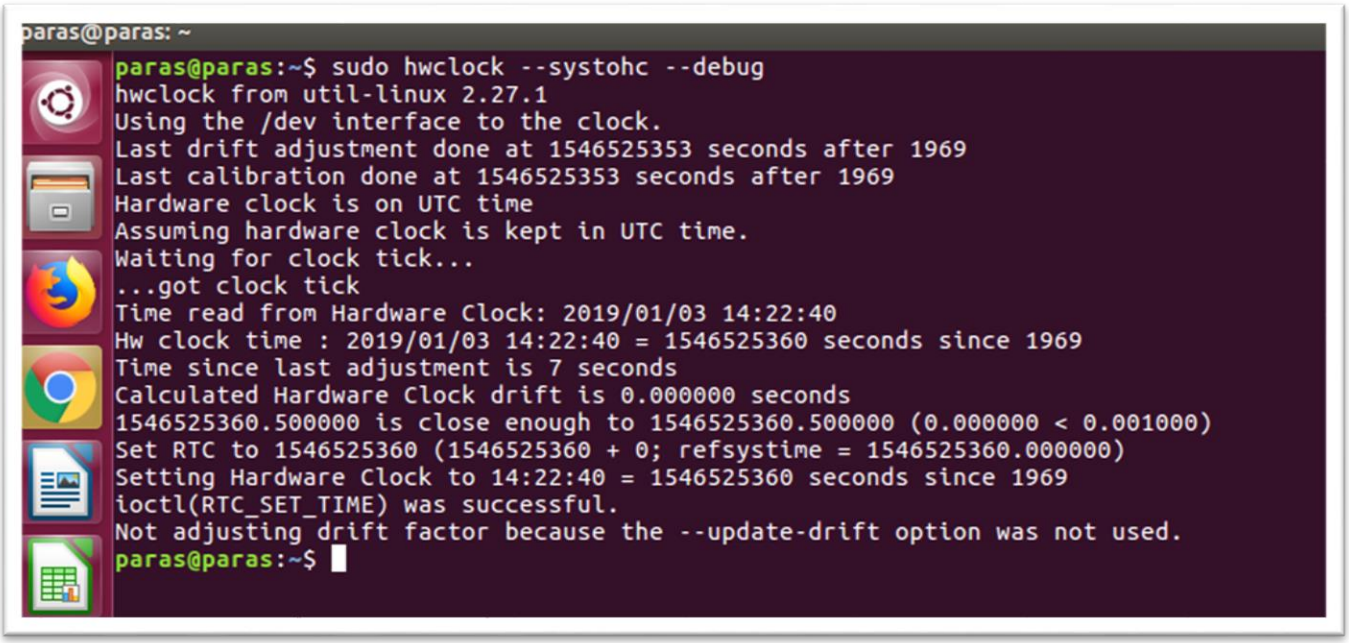
۱. برای نمایش ساعت و ساعت سخت افزار:



۲. با استفاده از عملکرد تنظیم:



٤. با استفاده از گزینه اشکال زدایی برای به دست آوردن اطلاعات اشکال زدایی:



مثال‌های بیشتر:

نمایش زمان RTC:

```
hwclock -r
```

این دستور زمان RTC را نشان می‌دهد.

نمایش زمان RTC اصلاح شده با رانش:

```
hwclock --get
```

این دستور زمان RTC اصلاح شده با در نظر گرفتن رانش را نمایش می‌دهد.

تنظیم زمان RTC با توجه به زمان مشخص شده:

```
hwclock --set --date="2022-10-01 10:30"
```

این دستور زمان RTC را با توجه به زمان مشخص شده (مانند "۲۰۲۲-۱۰-۰۱ ۱۰:۳۰") تنظیم می‌کند.

تنظیم زمان سیستم با استفاده از RTC:

```
hwclock -s
```

این دستور زمان سیستم را با استفاده از زمان RTC تنظیم می‌کند.

تنظیم RTC با زمان سیستم:

```
hwclock -w
```

این دستور زمان RTC را با زمان سیستم (برعکس دستور قبلی) تنظیم می‌کند.

ارسال پیکربندی‌های جدول زمانی به هسته:

```
hwclock --systz
```

این دستور برای ارسال پیکربندی‌های جدول زمانی به هسته استفاده می‌شود.

تنظیم RTC با در نظر گرفتن رانش سیستماتیک:

```
hwclock -a
```

این دستور RTC را تنظیم می‌کند تا رانش سیستماتیک را در نظر بگیرد.

پیش‌بینی زمان رانده شده RTC با توجه به زمان مشخص شده:

```
hwclock --predict --date="2022-10-01 10:30"
```

این دستور زمان رانده شده RTC را با توجه به زمان مشخص شده (مانند "۲۰۲۲-۱۰-۰۱ ۱۰:۳۰") پیش‌بینی می‌کند .سایر سویچ‌ها و گزینه‌های دستور hwclock نیز شامل مواردی مانند نشان دادن مقیاس زمانی UTC یا محلی، اشکال زدایی، نمایش اطلاعات نسخه و راهنما هستند. برای مثال، اگر می‌خواهید نسخه دستور hwclock را ببینید و از آن خارج شوید، از سویچ V- یا version-- استفاده کنید:

```
hwclock -V
```

همچنین، اگر نیاز دارید متن راهنما را مشاهده کنید و از برنامه خارج شوید، از سویچ h- یا help-- استفاده کنید:

```
hwclock -h
```

دستور tzselect

شرح دستور:

دستور tzselectیک ابزار خط فرمان در سیستم‌های لینوکسی است که به شما کمک می‌کند منطقه زمانی سیستم خود را به صورت تعاملی تنظیم کنید .این دستور با نمایش لیستی از مناطق زمانی مختلف و پرسیدن سوالاتی از شما، به شما امکان می‌دهد تا منطقه زمانی مورد نظر خود را انتخاب کنید.

ساختار دستور:

```
tzselect
```

این دستور به سادگی اجرا می‌شود و نیازی به هیچ آرگومان اضافی ندارد.

کاربردهای دستور:

- **تنظیم اولیه منطقه زمانی :**هنگام نصب یک سیستم لینوکس جدید یا هنگام تغییر مکان جغرافیایی، از این دستور برای تنظیم منطقه زمانی صحیح استفاده می‌شود.
- **تغییر منطقه زمانی موجود :**اگر منطقه زمانی سیستم شما به اشتباه تنظیم شده باشد، می‌توانید با استفاده از این دستور آن را اصلاح کنید.
- **یافتن کد منطقه زمانی :**اگر کد منطقه زمانی خاصی را می‌دانید، می‌توانید از این دستور برای جستجوی آن استفاده کنید.

گزینه‌های دستور:

دستور tzselectگزینه‌های بسیار کمی دارد و معمولاً بدون هیچ گزینه‌ای اجرا می‌شود .اما برخی از توزیع‌های لینوکس ممکن است از گزینه‌های اضافی پشتیبانی کنند .برای مشاهده گزینه‌های موجود در سیستم خود، می‌توانید از دستور man tzselect استفاده کنید.

مثال های دستور:

۱. **اجرای دستور:**

```
tzselect
```

پس از اجرای این دستور، یک فهرست تعاملی از قاره‌ها، اقیانوس‌ها و سایر مناطق جغرافیایی نمایش داده می‌شود .شما باید با وارد کردن شماره مربوط به هر گزینه، منطقه زمانی مورد نظر خود را انتخاب کنید.

۲. **یافتن منطقه زمانی خاص :**اگر کد منطقه زمانی خاصی را می‌دانید (مثلاً Asia/Tehranبرای تهران)، می‌توانید به جای استفاده از فهرست تعاملی، به طور مستقیم آن را وارد کنید .اما این روش در همه توزیع‌ها پشتیبانی نمی‌شود.

مراحل تنظیم منطقه زمانی با استفاده از tzselect:

۱. **اجرای دستور :**دستور tzselectرا در ترمینال اجرا کنید.

۲. **انتخاب قاره :**از فهرست قاره‌ها، قاره خود را انتخاب کنید.

۳. **انتخاب منطقه :**از فهرست مناطق موجود در قاره انتخاب شده، منطقه خود را انتخاب کنید.

۴. **انتخاب شهر :**از فهرست شهرهای موجود در منطقه انتخاب شده، شهر خود را انتخاب کنید.
۵. **تایید انتخاب :**پس از انتخاب شهر، دستور tzselectکد منطقه زمانی را نمایش می‌دهد و از شما می‌خواهد تا انتخاب خود را تایید کنید.

توجه :پس از انتخاب منطقه زمانی، سیستم شما به صورت خودکار تنظیم می‌شود و ساعت سیستم با منطقه زمانی جدید هماهنگ می‌شود.

نکته :پس از تنظیم منطقه زمانی با استفاده از tzselect، ممکن است نیاز به اجرای دستور hwclock --systohcداشته باشید تا ساعت سخت‌افزاری سیستم نیز با زمان سیستم هماهنگ شود.

مزایای استفاده از:tzselect

- **رابط کاربری ساده و تعاملی :**این دستور از یک رابط کاربری ساده و قابل درک استفاده می‌کند که حتی برای کاربران مبتدی نیز آسان است.
- **دقت بالا :**با استفاده از این دستور، می‌توانید منطقه زمانی خود را با دقت بسیار بالا تنظیم کنید.
- **پشتیبانی گسترده :**این دستور در اکثر توزیع‌های لینوکس موجود است.

دستور locale

دستور locale در سیستم‌عامل‌های Unix/Linux برای نمایش و تنظیم تنظیمات محلی ( )localeسیستم استفاده می‌شود. تنظیمات محلی شامل مواردی مانند زبان و فرمت زمان است که برای هر کاربر یا سیستم می‌تواند متفاوت باشد.

ساختار دستور:

```
locale [option]
locale [option] -a
locale [option] -m
locale [option] name...
```

مثال‌های دستور:

نمایش تنظیمات محلی فعلی:

```
locale
```

تنظیم تنظیمات محلی به زبان فرانسوی:

```
LANG=fr_FR.UTF-8 locale
```

۱. – a:نمایش تمامی تنظیمات محلی موجود.

```
locale -a
```

۲. – m:نمایش مپ‌های تنظیمات محلی.

```
locale -m
```

۳. – k:نمایش کلیدهای تنظیمات محلی.

```
locale -k
```

۴. – v: نمایش جزئیات بیشتر درباره تنظیمات محلی.

```
locale -v
```

۵. – c: نمایش تنظیمات محلی برای کدام کلیدی را اعمال کند.

```
locale -c LC_TIME
```

دستور timedatectl

دستور timedatectl در سیستم‌عامل‌های مبتنی بر Linux برای مدیریت تاریخ، زمان و تنظیمات ساعت سیستم استفاده می‌شود. این دستور به کاربر امکان می‌دهد تا تاریخ و زمان سیستم را تنظیم کند، منطقه زمانی را تغییر دهد و اطلاعات دقیق‌تری را در مورد ساعت سیستم دریافت کند.

ساختار دستور:

```
timedatectl [OPTIONS...] {COMMAND}
```

گزینه های دستور:

- **Status**: نمایش تنظیمات فعلی ساعت سیستم و (RTCساعت سیستمی داخلی)، از جمله وضعیت همگام‌سازی زمان شبکه. اگر دستور خاصی مشخص نشده باشد، به عنوان پیش‌فرض در نظر گرفته می‌شود. اضافه شده در نسخه ۱۹۵.
- **Show**: نمایش همان اطلاعاتی که در دستور status نمایش داده می‌شود، اما به صورت قابل خواندن برای ماشین. این دستور برای استفاده در مواردی که نیاز به خروجی قابل پردازش توسط کامپیوتر دارید، مناسب است. اگر به دنبال خروجی قابل خواندن برای انسان هستید، از دستور status استفاده کنید. به طور پیش‌فرض، ویژگی‌های خالی حذف می‌شوند. اگر می‌خواهید همه ویژگی‌ها نمایش داده شوند، از --all استفاده کنید. برای انتخاب ویژگی‌های خاص برای نمایش، از --property= استفاده کنید. اضافه شده در نسخه ۲۳۹.
- **set-time [TIME]**: تنظیم ساعت سیستم به زمان مشخص شده. این عمل باعث به‌روزرسانی زمان RTC نیز می‌شود. زمان می‌تواند به فرمت "۲۰۱۲-۱۰-۳۰ ۱۶:۱۷:۱۸" مشخص شود. اضافه شده در نسخه ۱۹۵.
- **set-timezone [TIMEZONE]**: تنظیم منطقه زمانی سیستم به مقدار مشخص شده. مناطق زمانی موجود می‌توانند با دستور list-timezones لیست شوند. اگر RTC به زمان محلی تنظیم شده باشد، این عمل همچنین زمان RTC را به‌روز می‌کند. این تماس نیز نمایه /etc/localtime را تغییر می‌دهد. برای اطلاعات بیشتر، به localtime(5)مراجعه کنید. اضافه شده در نسخه ۱۹۵.
- **list-timezones**: لیست مناطق زمانی موجود، هر کدام در یک خط. می‌توان با استفاده از دستور set-timezone یکی از موارد از این لیست را به عنوان منطقه زمانی سیستم تنظیم کرد. اضافه شده در نسخه ۱۹۵.
- **set-local-rtc [BOOL]**: این دستور یک آرگومان بولی می‌پذیرد. اگر "۰" باشد، سیستم به‌طور کلی به حالت زمان جهانی (UTC) تنظیم می‌شود. اگر "۱" باشد، RTC به‌جای زمان جهان
- **set-ntp [BOOL]**: این دستور یک آرگومان بولی می‌پذیرد. اگر مقدار آرگومان true باشد، همگام‌سازی زمان شبکه فعال و راه‌اندازی می‌شود. اگر مقدار آرگومان false باشد، همگام‌سازی شبکه غیرفعال می‌شود.
- **timesync-status**: نمایش وضعیت فعلی systemd-timesyncd.service. اگر --monitor مشخص شود، وضعیت به‌روزرسانی‌ها نیز نظارت می‌شود.
- **show-timesync**: نمایش همان اطلاعاتی که در timesync-status نمایش داده می‌شود، اما به صورت قابل خواندن برای ماشین. این دستور برای استفاده در مواردی که نیاز به خروجی قابل پردازش توسط کامپیوتر دارید، مناسب است. اگر به

دنبال خروجی قابل خواندن برای انسان هستید، از دستور timesync-status استفاده کنید. به طور پیش‌فرض، ویژگی‌های خالی حذف می‌شوند. اگر می‌خواهید همه ویژگی‌ها نمایش داده شوند، از --all استفاده کنید. برای انتخاب ویژگی‌های خاص برای نمایش، از --property= استفاده کنید.

- **ntp-servers INTERFACE SERVER…**: تنظیم سرورهای NTP مخصوص رابط (INTERFACE). این دستور فقط زمانی قابل استفاده است که رابط توسط systemd-networkd مدیریت می‌شود.
- **revert INTERFACE**: بازگرداندن سرورهای NTP مخصوص رابط (INTERFACE) به حالت اولیه. این دستور نیز فقط زمانی قابل استفاده است که رابط توسط systemd-networkd مدیریت می‌شود.
- **no-ask-password**: عدم درخواست رمز عبور از کاربر برای عملیات‌های اجازه‌دهنده.
- **adjust-system-clock**: اگر دستور set-local-rtc فراخوانی شود و این گزینه ارسال شود، ساعت سیستم از RTC به‌روزرسانی می‌شود و تنظیمات جدید در نظر گرفته می‌شود. در غیر این صورت، RTC از ساعت سیستم به‌روزرسانی می‌شود.
- **monitor** وقتی timesync-status با این گزینه فراخوانی می‌شود، timedatectl وضعیت systemd-timesyncd.service را نظارت می‌کند و خروجی‌ها را به‌روز می‌کند. شما می‌توانید با فشردن Ctrl+C نظارت را متوقف کنید.
- **a, --all**: وقتی ویژگی‌های systemd-timesyncd.service را نمایش می‌دهد، این گزینه تمام ویژگی‌ها را نمایش می‌دهد، بدون توجه به اینکه آیا تنظیم شده‌اند یا نه.
- **p, --property=**: نمایش ویژگی‌های خاصی از systemd-timesyncd.service را به مقدار مشخص شده در آرگومان محدود می‌کند. اگر مشخص نشود، تمام ویژگی‌های تنظیم شده نمایش داده می‌شوند. به عنوان مثال، می‌توانید از --property=ServerName برای نمایش تنها ویژگی ServerName استفاده کنید.
- **Value**: وقتی ویژگی‌ها را با show-timesync نمایش می‌دهد، فقط مقدار نمایش داده می‌شود و نام ویژگی و "=" حذف می‌شود.
- **H, --host=**: اجرای عملیات به‌صورت راه‌دور با مشخص کردن نام میزبان یا نام کاربری و میزبان با "@" جدا شده. همچنین می‌توانید شماره پورت (جدا شده با ":") و نام کانتینر (جدا شده با "/") را برای متصل شدن مستقیم به یک کانتینر خاص در میزبان مشخص‌شده با استفاده از SSH اضافه کنید. آدرس‌های IPv6 باید در براکت قرار گیرند.
- **M, --machine=**: اجرای عملیات در یک کانتینر محلی.

مثال‌های دستور:

۱. نمایش تاریخ و زمان فعلی سیستم:

```
timedatectl
```

۲. تنظیم تاریخ و زمان سیستم:

```
sudo timedatectl set-time 'YYYY-MM-DD HH:MM:SS'
```

۳. تغییر منطقه زمانی سیستم به مثلاً Asia/Tehran:

```
sudo timedatectl set-timezone Asia/Tehran
```

گزینه‌های دستور timedatectl:

۱. status نمایش وضعیت فعلی تاریخ، زمان و منطقه زمانی.

```
timedatectl status
```

۲. set-timezoneتغییر منطقه زمانی سیستم.

```
sudo timedatectl set-time 'YYYY-MM-DD HH:MM:SS'
sudo timedatectl set-timezone Asia/Tehran
```

۴. list-timezonesنمایش لیست مناطق زمانی موجود.

```
timedatectl list-timezones
```

۵. set-ntpفعال یا غیرفعال کردن سینک زمان با NTP.

```
sudo timedatectl set-ntp true
```

دستور date

دستور ``date`` در سیستم‌عامل‌های Unix/Linux برای نمایش تاریخ و زمان فعلی سیستم استفاده می‌شود. این دستور به کاربر امکان می‌دهد تا تاریخ و زمان را به صورت محلی یا بر اساس فرمت‌های مختلف نمایش دهد.

ساختار دستور:

```
date [OPTION]... [+FORMAT]
date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
```

گزینه های دستور:

- d یا date=STRING تاریخ و زمان را بر اساس STRING نمایش دهید (نه "الان").
- debug تاریخ تجزیه‌شده را توضیح دهید و در مورد استفاده مشکوک به stderr هشدار دهید.
- f یا file=DATEFILE مانند date؛ یک بار برای هر خط از DATEFILE.
- I[FMT] یا I[=FMT] iso8601 تاریخ/زمان را در فرمت ISO 8601 نمایش دهید. FMT='date' برای تاریخ فقط (پیش‌فرض)، 'hours'، 'minutes'، 'seconds' یا 'ns' برای تاریخ و زمان با دقت مشخص. مثال T0234560600.۲۰۰۶۰۸۱۴
- resolution دقت موجود برای نشان‌دادن زمان را نمایش دهید. مثال ۰٫۰۰۰۰۰۰۰۰۱.
- R یا rfcemail تاریخ و زمان را در فرمت RFC 5322 نمایش دهید. مثال Mon, 14 Aug 2006 023456 0600.
- rfc3339=FMT تاریخ/زمان را در فرمت RFC 3339 نمایش دهید. FMT='date'، 'seconds' یا 'ns' برای تاریخ و زمان با دقت مشخص. مثال ۲۰۰۶۰۸۱۴ ۰۲۳۴۵۶۰۶۰۰.
- r یا reference=FILE زمان آخرین تغییر فایل FILE را نمایش دهید.
- u یا utc یا universal زمان یا تاریخ را به زمان هماهنگ جهانی (UTC) تنظیم یا نمایش دهید.
- help این راهنما را نمایش دهید و خارج شوید.
- version اطلاعات نسخه را نمایش دهید و خارج شوید

مثال های دستور:

نمایش تاریخ و زمان فعلی به صورت پیش‌فرض:

```
date
```

نمایش تاریخ و زمان به صورت فرمت‌دار (مثلاً تاریخ و ساعت کامل):

```
date “+%Y-%m-%d %H:%M:%S”
```

۱. +%Y`: نمایش سال به فرمت چهار رقمی.

```
date “+%Y”
```

۲. +%m`: نمایش ماه به فرمت دو رقمی.

```
date “+%m”
```

۳. +%d`: نمایش روز ماه به فرمت دو رقمی.

```
date “+%d”
```

۴. +%H`: نمایش ساعت به فرمت دو رقمی.

```
date “+%H”
```

۵. +%M`: نمایش دقیقه به فرمت دو رقمی.

```
date “+%M”
```

۶. +%S`: نمایش ثانیه به فرمت دو رقمی.

```
date “+%S”
```

دستور ntpd

دستور ntpd در سیستم‌عامل‌های Unix/Linux برای مدیریت و همگام‌سازی زمان سیستم با استفاده از پروتکل شبکه NTP (Network Time Protocol) استفاده می‌شود. این دستور اجازه می‌دهد تا سیستم به‌طور خودکار زمان خود را با سرورهای زمان خارجی همگام‌سازی کند.

گزینه های دستور ntpd:

```
ntpd [ -46aAbdDgLmnNqx ] [ -c conffile ] [ -f driftfile ] [ -i jaildir ] [ -I iface ] [ -k keyfile] [ -
l logfile ] [ -p pidfile ] [ -P priority ] [ -r broadcastdelay ] [ -s statsdir] [ -t key ] [ -
u user[:group] ] [ -U interface_update_interval ] [ -v variable] [ -V variable ]
```

گزینه های دستور ntpd:

- ۴: استفاده از IPv4 برای ارتباط با سرورهای NTP.
- ۶: استفاده از IPv6 برای ارتباط با سرورهای NTP.
- a: استفاده از تمام سرورها به عنوان منابع زمانی،
- A: استفاده از تمام سرورها به عنوان منابع زمانی، b
- b: فورس کردن یک تنظیم ساعت با استفاده از سرور NTP، حتی اگر تنظیم ساعت به عنوان پیر شناخته شده باشد.
- d: حالت اشکال‌زدایی را فعال کنید و پیام‌های خطا را به stderr ارسال کنید.
- D: نمایش اطلاعات جزئی‌تر در مورد عملکرد ntpd.
- g: تنظیم ساعت به زمان سرور NTP، حتی اگر تنظیم ساعت به عنوان پیر شناخته شده باشد.

- l logfile: ذخیره‌سازی خروجی وقایع در یک فایل لاگ.
- p pidfile: ذخیره‌سازی شناسه فرآیند ( )PIDدر یک فایل.
- s statsdir: ذخیره‌سازی آمارهای مربوط به ntpd در یک دایرکتوری.
- u user[:group]: تنظیم کاربر و گروه برای اجرای ntpd.
- v variable: تنظیم متغیرهای داخلی ntpd.
- V variable: نمایش مقدار متغیرهای داخلی ntpd.
- x: استفاده از تنظیم ساعت با استفاده از سرور NTP، حتی اگر تنظیم ساعت به عنوان پیر شناخته شده باشد، و اجتناب از تنظیم‌های بزرگتر از ۶۰۰ ثانیه.

مثال های دستور:

۴: استفاده از IPv4 برای ارتباط با سرورهای NTP.

```
ntpd -4
```

۶: استفاده از IPv6 برای ارتباط با سرورهای NTP.

```
ntpd -6
```

a: استفاده از تمام سرورها به عنوان منابع زمانی

```
ntpd -a
```

A: استفاده از تمام سرورها به عنوان منابع زمانی

```
ntpd -A
```

b: فورس کردن یک تنظیم ساعت با استفاده از سرور NTP

```
ntpd -b
```

d: حالت اشکال‌زدایی را فعال کنید و پیام‌های خطا را به stderr ارسال کنید.

```
ntpd -d
```

D: نمایش اطلاعات جزئی‌تر در مورد عملکرد ntpd.

```
ntpd -D
```

g: تنظیم ساعت به زمان سرور NTP،

```
ntpd -g
```

l logfile: ذخیره‌سازی خروجی وقایع در یک فایل لاگ.

```
ntpd -l /var/log/ntpd.log
```

p pidfile: ذخیره‌سازی شناسه فرآیند ( )PIDدر یک فایل.

```
ntpd -p /var/run/ntpd.pid
```

s statsdir: ذخیره‌سازی آمارهای مربوط به ntpd در یک دایرکتوری.

```
ntpd -s /var/lib/ntpd/stats
```

دستور ntpdate در سیستم‌عامل‌های Unix/Linux برای دستیابی به زمان دقیق از یک سرور NTP (Network Time Protocol) خارجی استفاده می‌شود. این دستور به کاربر امکان می‌دهد تا زمان سیستم خود را با یک سرور NTP همگام‌سازی کند.

گزینه های دستور:

```
ntpdate [ -46bBdqsuv ] [ -a key ] [ -e authdelay] [ -k keyfile ] [ -o version ] [ -p samples ] [ -t timeout ] [ -U user_name] server [ ... ]
```

مثال های دستور

۴ استفاده از IPv4 برای ارتباط با سرورهای

```
ntp. ntpdate -4 pool.ntp.org
```

۶ استفاده از IPv6 برای ارتباط با سرورهای

```
ntp. ntpdate -6 pool.ntp.org
```

bاستفاده از برنامه‌ی ntpdc برای تنظیم ساعت.

```
ntpdate -b pool.ntp.org
```

Bاستفاده از برنامه‌ی ntpdate برای تنظیم ساعت.

```
ntpdate -b pool.ntp.org
```

dحالت اشکال‌زدایی را فعال کنید و اطلاعات بیشتری را نمایش دهید.

```
ntpdate -d pool.ntp.org
```

qتنها یکبار تنظیم ساعت را انجام دهید و خروجی را نمایش دهید.

```
ntpdate -q pool.ntp.org
```

sحالت سکوت را فعال کنید و تنها خروجی نهایی را نمایش دهید.

```
ntpdate -s pool.ntp.org
```

uاجازه دهید که کاربر عادی هم از ntpdate استفاده کند.

```
ntpdate -u pool.ntp.org
```

vحالت verbose را فعال کرده و جزئیات بیشتری را نمایش دهید.

```
ntpdate -v pool.ntp.org
```

a keyاستفاده از کلید احراز هویت برای اتصال به سرور

```
ntp. ntpdate -a mykey pool.ntp.org
```

e authdelayتنظیم تأخیر احراز هویت برای اتصال به سرور

```
ntp. ntpdate -e 1.5 pool.ntp.org
```

k keyfileاستفاده از یک فایل کلید برای احراز هویت.

```
ntpdate -k ntpkeyfile pool.ntp.org
```

o versionتنظیم نسخه پروتکل NTP برای ارتباط.

```
ntpdate -o 4 pool.ntp.org
```

p samplesتنظیم تعداد نمونه‌ها برای محاسبه زمان دقیق.

```
ntpdate -p 4 pool.ntp.org
```

t timeoutتنظیم زمان انتظار برای اتصال به سرور

```
ntp. ntpdate -t 10 pool.ntp.org
```

U user_nameتنظیم نام کاربر برای اجرای

```
ntpdate. ntpdate -u myuser pool.ntp.org
```

دستور fc-cache

در لینوکس اسکن فهرست فونت برای برنامه های کاربردی که از **fontconfig** برای دسترسی به انواع فونت های نصب شد، هستند، استفاده میشود. .هنگامی که دستوراجرا میشود، یک حافظه پنهان ایجاد می شود که شامل خصوصیات هر فونت و نام پرونده ایجاد شده و این حافظه پنهان بیشتر برای سرعت بخشیدن به برنامه هنگام استفاده از کتابخانه ***fontconfig*** هنگام بوت استفاده می شود.

ساختار نحوی دستور:

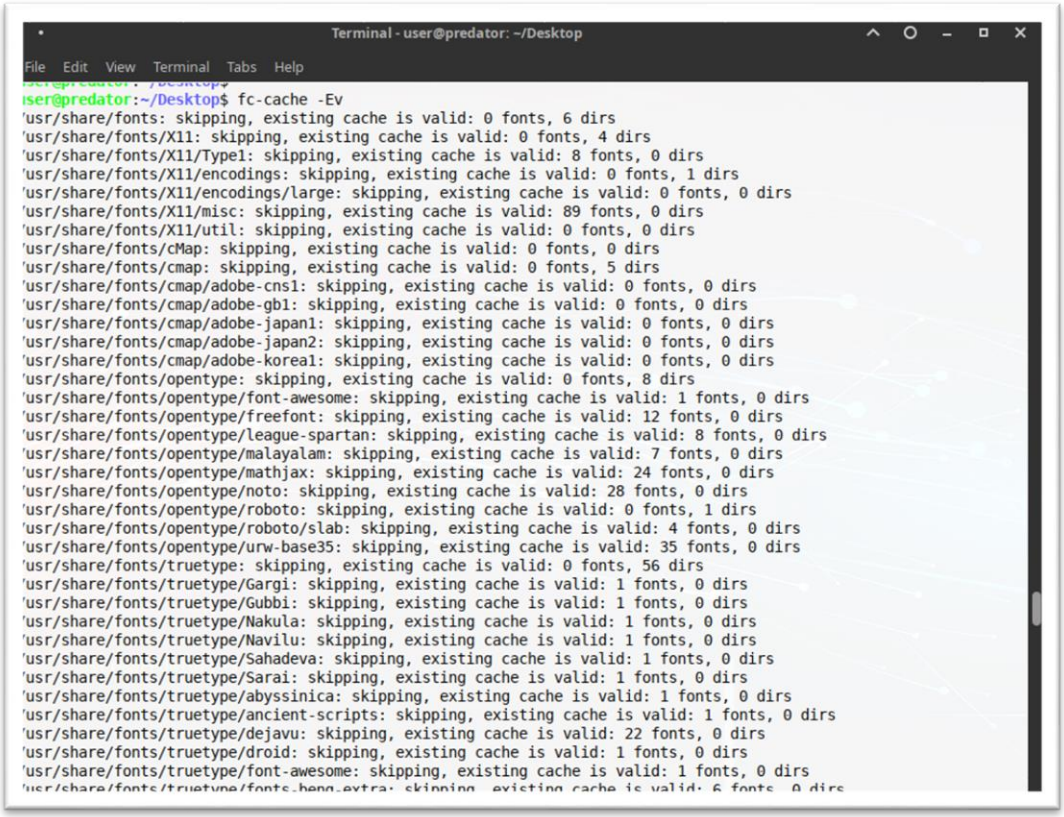
fc-cache [-EfrsvVh] [–error-on-no-fonts] [–force] [–really-force] [[-y dir] [–sysroot dir]] [–system-only] [–verbose] [–version] [–help] [dir…]

گزینه های دستور:

- E (–error-on-no-fonts) هنگامی که از این گزینه استفاده می شود اگر هیچ فونت موجود در dir یا فهرست ها در پیکربندی نباشد خطایی ایجاد می‌شود.
- f (–force) هنگامی که از این دستور استفاده می شود ، پرونده های حافظه پنهان به ظاهر به روز به زودی ایجاد می شود، این امر مهمترین زمان بررسی Timestamp است.
- r (–really-force) این گزینه تمام پرونده های حافظه پنهان موجود را پاک کرده و مجدداً اسکن می کند.
- s این دستور فقط فهرست های کل سیستم را اسکن می کند و مکان های واقع در /usr/home را حذف می کند.
- v (–verbose) در حین کار خروجی را نمایش میدهد.
- V (–version) نسخه را نمایش می دهد و از آن خارج می شود.

مثال های دستور:

fc-cache -Ev



توجه : اگر cache با موفقیت تولید شود و غیر صفر باشد ، **fc-cache** صفر را برمی‌گرداند

دستور fc-list

دستور **fc-list** بخشی از سیستم *fontconfig* است .برای لیست کردن فونت ها و سبک های فونت موجود استفاده می شود .با استفاده از گزینه format می توان لیست تمام فونت ها را فیلتر و مرتب کرد .

ساختار دستور:

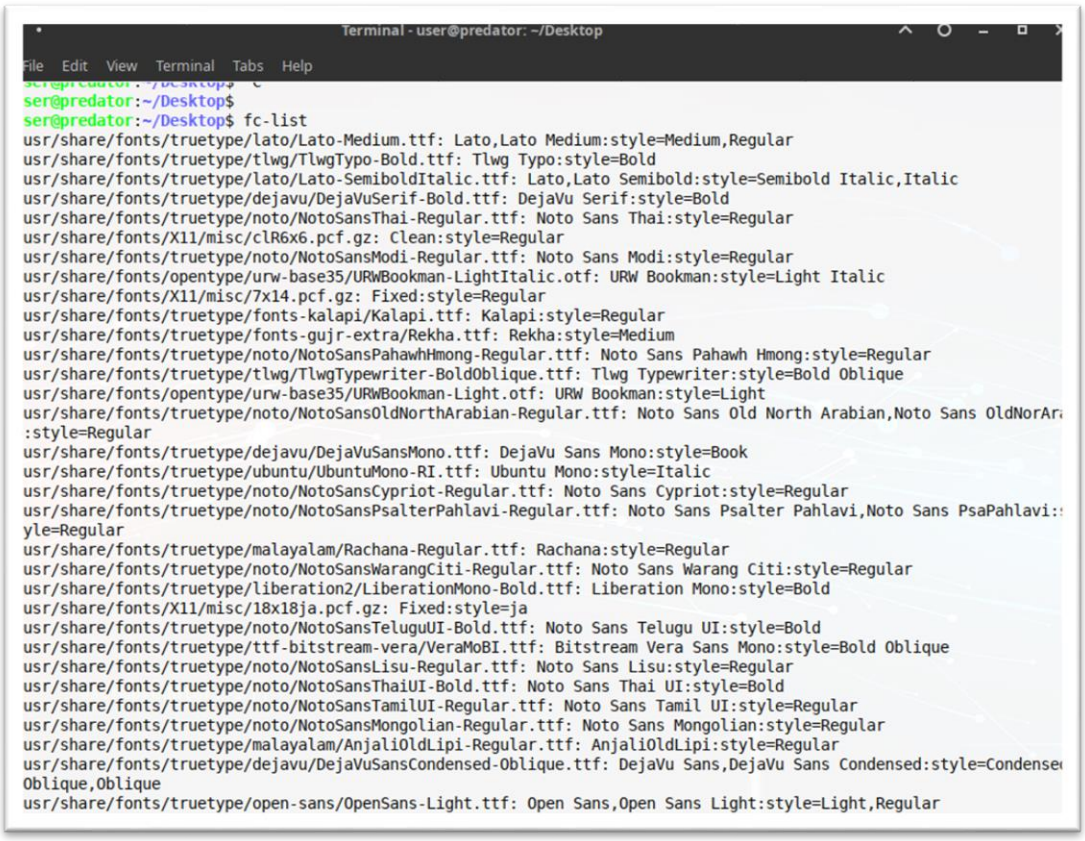
fc-list [-vqVh] [-f FORMAT] [–verbose] [–format=FORMAT] [–quiet] [–version] [–help] [pattern] {element …} List fonts matching [pattern]

مثال :همه مکان های پرونده های فونت موجود در سیستم را با نام فونت ، فاصله و نوع سبک آنها چاپ می کند.

تزینه های دستور:

- **-v –verbose:** برای نشان دادن کل الگوی فونت به صورت کلامی استفاده می شود.
- **-f –format = FORMAT:** برای استفاده از قالب خروجی داده شده.
- **-q –quiet:** همه خروجی های را متوقف می کند .
- **-V –version:** نسخه پیکربندی فونت را نشان میدهد و خارج میشود.
- **-h –help:** برای نشان دادن پیام راهنمای دستور است.

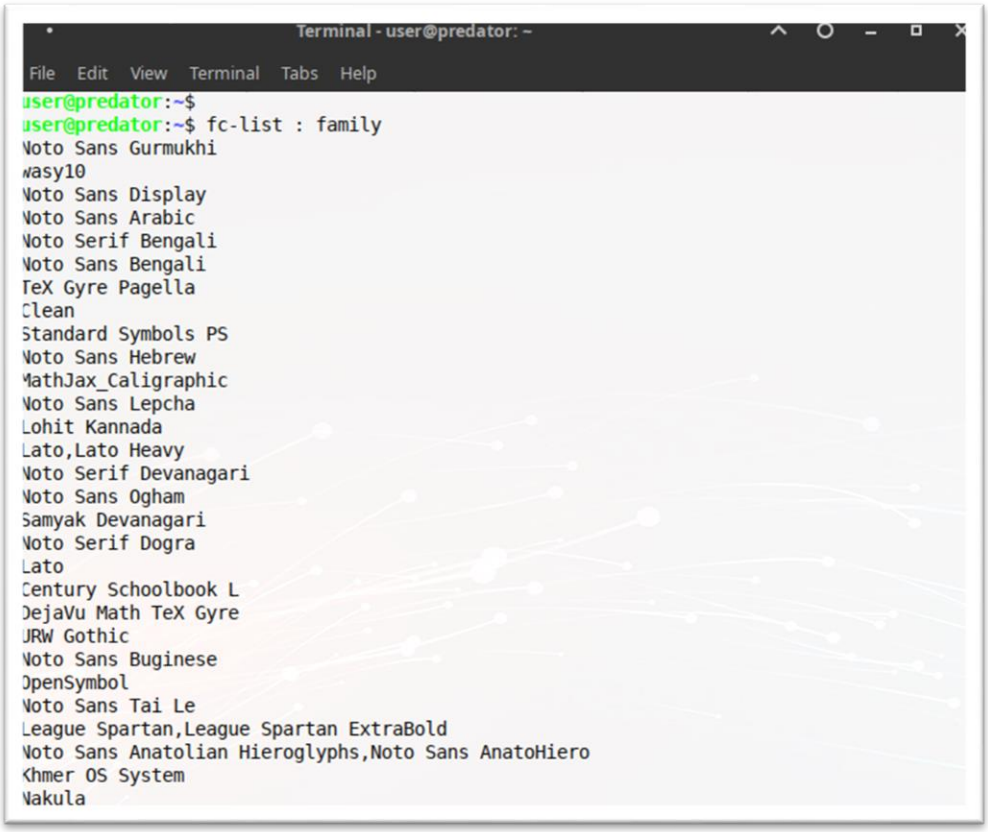
fc-list



مثال های دیگر:

در دستور زیر فقط نام خانواده های فونت را چاپ می کند ، بدون اینکه سایر جزئیات فوق را نشان دهد.

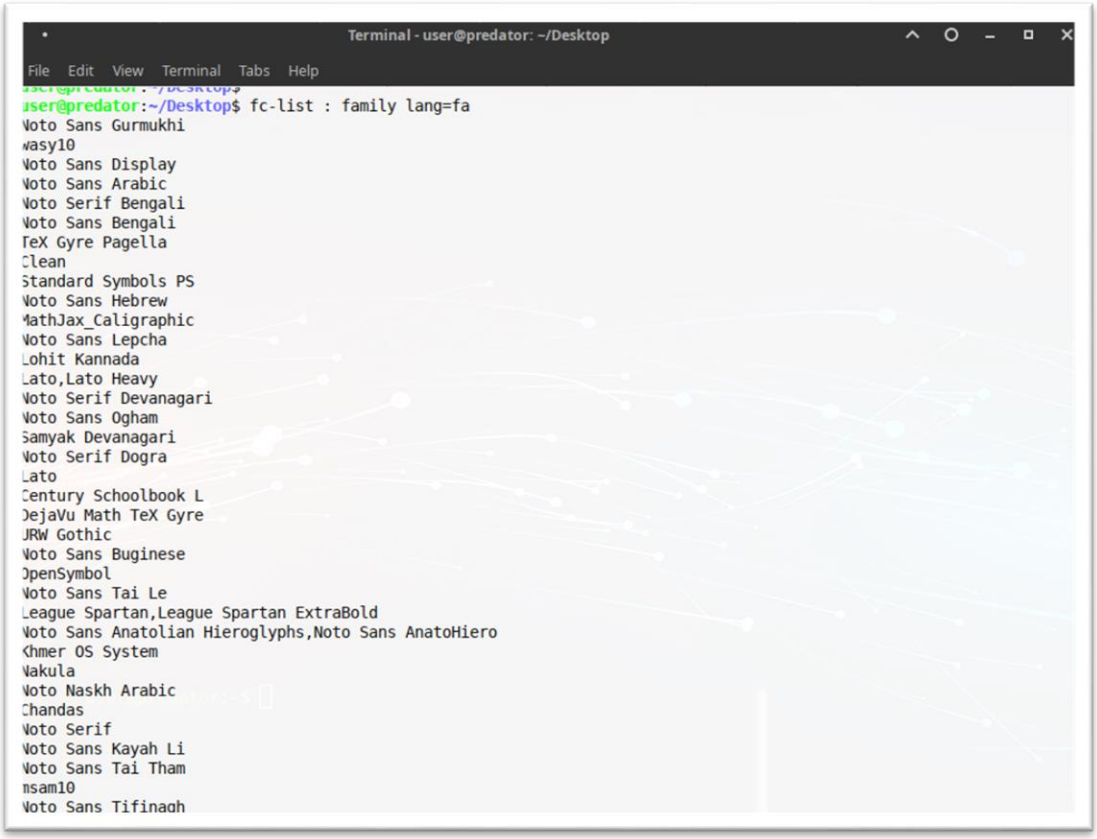
fc-list : family



- **fc-list with font family + Language Selector:** این فقط نام خانواده های فونت را که از کد زبان انتخاب شده پشتیبانی می کنند ، بدون نمایش سایر جزئیات فوق چاپ می کند.

توجه : کد زبان مورد استفاده در اینجا ***"ta" است*** که مخفف زبان **تامیل است**.

```
fc-list : family lang=ta
```

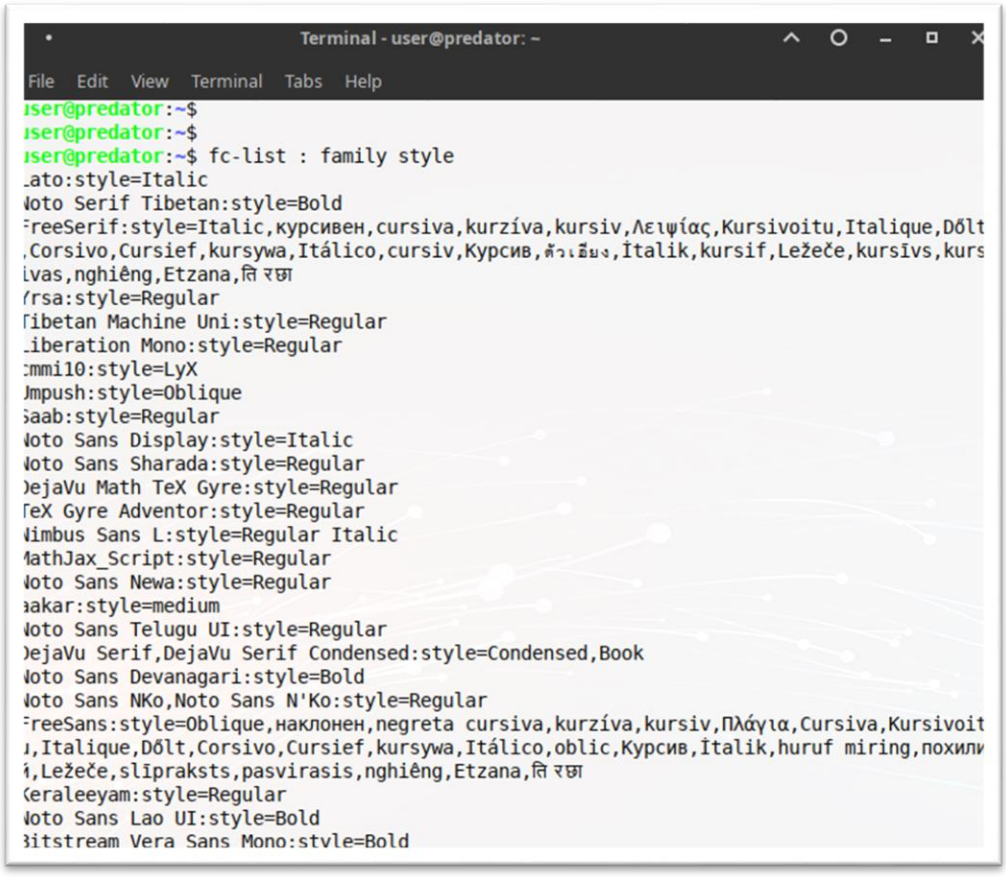


توجه :کد زبان مورد استفاده در اینجا"hi" است که مخفف زبان **هندی** است.

```
fc-list : family lang=hi
```

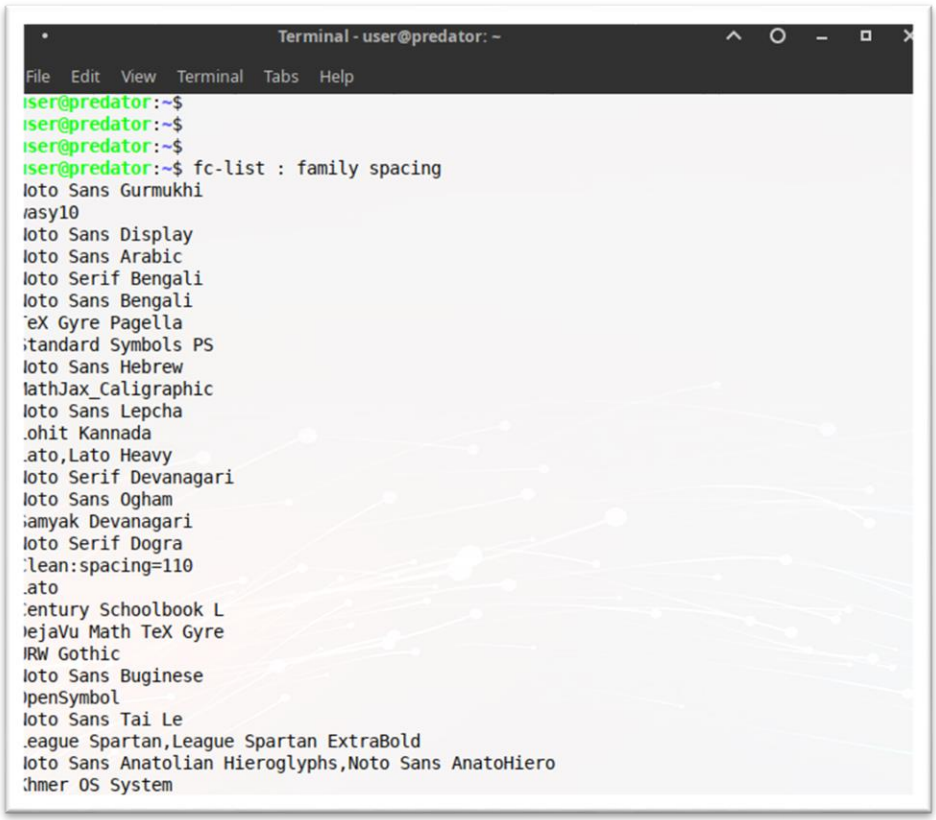
- **fc-list with other selectors:**مشابه انتخاب خانواده، ما همچنین می توانیم محل فایل ، فاصله یا / و سبک فونت های مورد نیاز را برای نمایش در صفحه انتخاب کنیم.

```
fc-list : family style
```



توجه: مرتب سازی و منحصر به فرد نیز می تواند همراه با این دستور ترکیب شود

```
fc-list : family spacing |
```

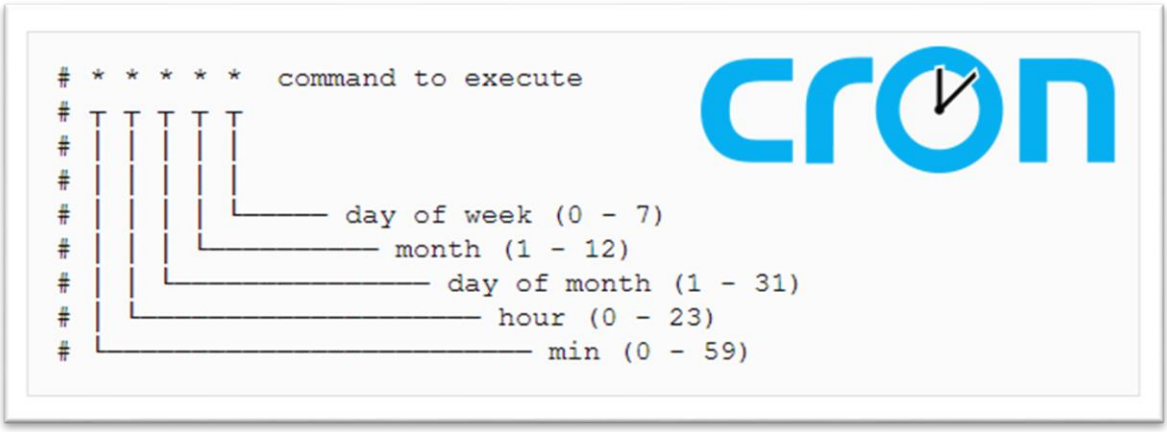


- **گزینه fc-list with format: از** این گزینه برای قالب بندی متن خروجی به الگوی مورد نیاز کاربر استفاده می شود .در این مثال ، از گزینه format برای بدست آوردن نام خانوادگی همه فونت ها ، مرتب و منحصر به فرد استفاده شده است.

```
fc-list --format="%{family[0]}\n"
```

دستور cron

cronرا یک ابزار نرم افزاری، ارائه شده توسط لینوکس مانند سیستم عامل که خودکار کار برنامه ریزی شده در یک زمان از پیش تعیین شده است .این یک فرآیند دایمون است ، که به عنوان یک فرایند پس زمینه اجرا می شود و عملیات مشخص شده را در زمان از پیش تعیین شده هنگامی که یک رویداد یا شرایط خاص بدون دخالت کاربر شروع می شود ، انجام می دهد. می توان چنین فرآیندهای به طور خودکار اجرا در پس زمینه در فاصله زمانی منظم از زمان با ایجاد یک لیست اجرا کرد .این امکان را برای کاربران فراهم می کند تا وظیفه برنامه ریزی شده را بصورت منظم مانند انجام پشتیبان گیری هر روز در نیمه شب ، برنامه ریزی به روز رسانی به صورت هفتگی و همگام سازی پرونده ها در یک بازه منظم انجام دهند .این ابزار به طور مرتب برای کار برنامه ریزی شده بررسی می کند و هنگامی که فیلدهای زمان برنامه ریزی شده با زمینه های زمان فعلی مطابقت دارند



ساختار دستور:

```
cron [-f] [-l] [-L loglevel]
```

گزینه های دستور:

- **-f**: برای ماندن در حالت پیش زمینه استفاده می شود و تقلب نمی کنید.
- **-l**: این نام های سازگار با LSB را برای/etc/cron.d پرونده ها فعال می کند.
- **-n**: برای افزودن FQDN در موضوع هنگام ارسال نامه های الکترونیکی استفاده می شود.
- **-L loglevel**: این گزینه به cron می گوید که چه کارهایی را باید با مقادیر زیر وارد کنید:
  - **1**: با شروع کارهای cron وارد سیستم می شود.
  - **2**: پایان تمام کارهای cron وارد می شود.
  - **4**: تمام jobهای ناموفق را ثبت می کند .در اینجا وضعیت خروجی برابر با صفر نخواهد بود.
  - **8**: تعداد فرآیند کار cron را ثبت می کند.

crontab لیستی از دستورات برای اجرای وظایف برنامه ریزی شده در زمان خاص است .به کاربر اجازه می دهد کارهای برنامه ریزی شده را اضافه ، حذف یا تغییر دهد .نحو دستور crontab دارای شش فیلد است که با فاصله از هم جدا شده اند ، جایی که پنج تای اول زمان اجرای کار را نشان می دهد و قسمت آخر مربوط به دستور است.

دقیقه (دارای مقداری بین ۰–۵۹ است)

ساعت (مقدار بین ۰–۲۳ را نگه می دارد)

روز ماه (مقدار بین ۱ تا ۳۱ را نگه می دارد)

ماه سال (مقداری بین ۱ تا ۱۲ یا ژانویه دارد ، سه حرف اول نام ماه استفاده می شود)

روز هفته (مقدار بین ۰–۶ یا Sun-Sat را نگه می دارد ، در اینجا همچنین از سه حرف اول روز استفاده می شود)

قواعد حاکم بر قالب قسمت تاریخ و زمان به شرح زیر است:

وقتی هر پنج فیلد اول روی ستاره تنظیم شوند (*) ، مخفف کل مقادیر فیلد است .به عنوان مثال ، برای اجرای یک دستور روزانه ، می توانیم یک ستاره (*) را در قسمت هفته قرار دهیم.

همچنین می توان از طیف وسیعی از اعداد استفاده کرد که با یک خط (–) در قسمت زمان و تاریخ از هم جدا شده و شامل بیش از یک مقدار مجاور است اما همه مقادیر فیلد را شامل نمی شود .به عنوان مثال ، ما می توانیم از ۷–۱۰ برای اجرای یک دستور از جولای تا اکتبر استفاده کنیم.

عملگر کاما (،) برای شامل لیستی از اعداد استفاده می شود که ممکن است پشت سر هم باشد یا نباشد .به عنوان مثال ، "۱ ، ۳ ، ۵" در قسمت هفته نشان دهنده اجرای یک دستور هر دوشنبه ، چهارشنبه و جمعه است.

یک کاراکتر اسلش (/) برای جست و خیز کردن تعدادی از مقادیر در نظر گرفته شده است .به عنوان مثال ، "* / ۴" در قسمت ساعت "هر ۴ ساعت" را مشخص می کند که معادل ۰ ، ۴ ، ۸ ، ۱۲ ، ۱۶ ، ۲۰ است.

اجازه دادن به کاربران برای اجرای cron jobs:

کاربر باید در این پرونده ذکر شود تا در صورت وجود پرونده بتواند cron jobs را اجرا کند.

/etc/cron.allow

اگر پرونده cron.allow وجود ندارد اما cron.deny پرونده وجود دارد ، بنابراین کاربر نمی تواند در این پرونده لیست نشده باشد تا بتواند کار cron را اجرا کند.

/etc/cron.deny

توجه :اگر هیچ یک از این پرونده ها وجود نداشته باشد ، فقط مدیر کاربر (مدیر سیستم) مجاز به استفاده از یک دستور داده شده است.

فرمت Crontab

```
MIN HOUR DOM MON DOW CMD
```

زمینه های Crontab و محدوده های مجاز

Field	Description	Allowed Value
MIN	Minute field	0 to 59
HOUR	Hour field	0 to 23
DOM	Day of Month	1-31
MON	Month field	1-12
DOW	Day Of Week	0-6
CMD	Command	Any command to be executed.

نمونه هایی از Cron jobs

```
30 08 10 06 * /home/maverick/full-backup
```

30 – 30th Minute
08 – 08 AM
10 – 10th Day
06 – 6th Month (June)
*** – Every day of the week**

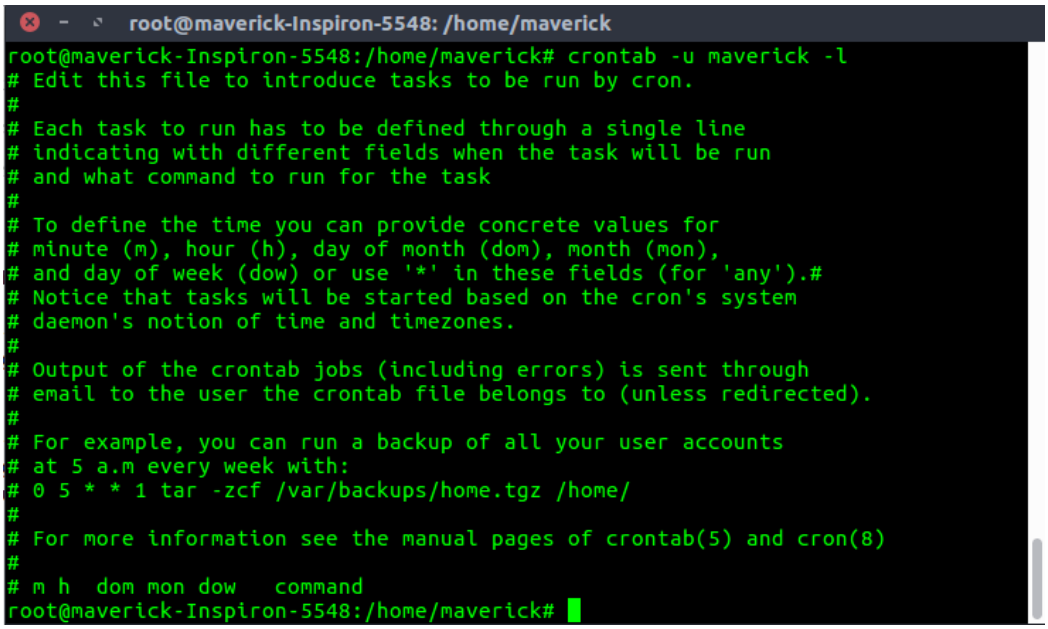
مشاهده ورودیهای Crontab کاربر ورود به سیستم فعلی: برای مشاهده مطالب crontab خود ، crontab -l را از حساب unix خود وارد کنید.

```
maverick@maverick-Inspiron-5548:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
maverick@maverick-Inspiron-5548:~$
```

مشاهده ورودی های ریشه :Crontab به عنوان کاربر ریشه وارد شوید (su - root) و crontab -l را انجام دهید.

```
root@maverick-Inspiron-5548: /home/maverick
root@maverick-Inspiron-5548:/home/maverick# crontab -l
no crontab for root
root@maverick-Inspiron-5548:/home/maverick#
```

برای مشاهده ورودی های crontab سایر کاربران لینوکس: برای root وارد شوید و از l- {username} u- استفاده کنید.



دستور anacron

دستور anacron در سیستم‌عامل‌های Unix/Linux برای اجرای وظایف دوره‌ای (cron jobs) استفاده می‌شود. این دستور به وظایفی که باید در زمان‌های مشخص اجرا شوند، اما به دلیل خاموشی سیستم در آن زمان قابل اجرا نبوده‌اند، کمک می‌کند.

ساختار دستور:

```
anacron [-s] [-f] [-n] [-d] [-q] [-t anacrontab] [-S spooldir] [job]
anacron [-S spooldir] -u [-t anacrontab] [job]
anacron [-V|-h]
anacron -T [-t anacrontab]
```

گزینه های دستور:

- f: اجرای اجباری کارها، بدون توجه به برچسب‌های زمانی.
- u: فقط به‌روزرسانی برچسب‌های زمانی کارها به تاریخ فعلی، بدون اجرای هیچ کاری.
- s: اجرای متوالی کارها. اناکرون کار جدیدی را شروع نمی‌کند تا کار قبلی به پایان برسد.
- n: اجرای کارها در حال حاضر. در این حالت، تأخیرهای مشخص شده در فایل /etc/anacrontab نادیده گرفته می‌شود. این گزینه به‌صورت ضمنی s را شامل می‌شود.
- d: اجرای در پس‌زمینه نشده. در این حالت، اناکرون پیام‌های اطلاعاتی را به خطای استاندارد و همچنین به سیستم لاگ می‌کند. خروجی کارها به طور معمول ارسال می‌شود.
- q: سرکوب پیام‌ها به خطای استاندارد. فقط با d قابل استفاده است.
- t some_anacrontab: استفاده از فایل anacrontab مشخص شده، به‌جای پیش‌فرض.
- T: آزمایش anacrontab. فایل پیکربندی برای صحت‌سنجی آزمایش می‌شود. اگر در فایل خطایی وجود داشته باشد، خطایی نمایش داده می‌شود و اناکرون بازگشتی به ۱ خواهد داد. anacrontabهای معتبر بازگشتی به ۰ خواهند داد.
- S spooldir: استفاده از دایرکتوری spooldir مشخص شده برای ذخیره برچسب‌های زمانی. این گزینه برای کاربرانی که می‌خواهند خودشان اناکرون را اجرا کنند، لازم است.
- V: چاپ اطلاعات نسخه و خروج.

h :چاپ پیام کوتاهی در مورد استفاده، و خروج.

مثال های دستور:

اجرای وظیفه دوره‌ای با فاصله زمانی مشخص: این دستور برای اجرای وظیفه دوره‌ای با فاصله زمانی مشخص استفاده می‌شود.

```
anacron -s -t 10 my_cron_job
```

اجرای وظیفه دوره‌ای بدون خروجی (silent mode): این دستور برای اجرای وظیفه دوره‌ای بدون نمایش خروجی استفاده می‌شود.

```
anacron -s -q my_cron_job
```

اجرای وظیفه دوره‌ای در حالت اصلاح (repair mode): این دستور برای اجرای وظیفه دوره‌ای در حالت اصلاح برای تعمیر وظایف اجرا نشده استفاده می‌شود.

```
anacron -r my_cron_job
```

نمایش لیست وظایف دوره‌ای تعریف شده: این دستور برای نمایش لیست وظایف دوره‌ای تعریف شده و زمان‌های اجرای آن‌ها استفاده می‌شود.

```
anacron -l
```

اجرای وظیفه دوره‌ای بر اساس فایل تنظیمات مشخص: این دستور برای اجرای وظیفه دوره‌ای بر اساس یک فایل تنظیمات خاص استفاده می‌شود.

```
anacron -t /path/to/config_file
```

دستور bg

از دستور bg در linux برای قرار دادن job پیش زمینه در پس زمینه استفاده می شود.

ساختار دستور:

```
bg [job_spec ...]
```

گزینه های دستور:

- bg [JOB_SPEC] :از این دستور برای قرار دادن کار ذکر شده در پس زمینه استفاده می شود .در تصویر زیر ، موارد زیر را انجام می دهیم:
  ۱. ما از دستور jobs برای لیست کردن همه job استفاده می کنیم
  ۲. ما با استفاده از دستور sleep یک فرایند ایجاد می کنیم ، شناسه آن را به صورت ۱ دریافت می کنیم.
  ۳. ما با ارائه شناسه آن به bg آن را در پس زمینه قرار می دهیم.

```
File  Edit  View  Search  Terminal  Help
naman@root:~$ jobs
naman@root:~$ sleep 500
^Z
[1]+  Stopped                 sleep 500
naman@root:~$ jobs
[1]+  Stopped                 sleep 500
naman@root:~$ bg %1
[1]+ sleep 500 &
naman@root:~$ jobs
[1]+  Running                 sleep 500 &
naman@root:~$
```

از "sleep 500" برای ایجاد job پیش زمینه ساختگی استفاده می شود.

- bg –help : این دستور اطلاعات راهنما را نمایش می دهد.

```
File  Edit  View  Search  Terminal  Help
naman@root:~$ bg --help
bg: bg [job_spec ...]
    Move jobs to the background.

    Place the jobs identified by each JOB_SPEC in the background, as if they
    had been started with `&'.  If JOB_SPEC is not present, the shell's notion
    of the current job is used.

    Exit Status:
    Returns success unless job control is not enabled or an error occurs.
naman@root:~$
```

دستور ATQ

دستور atq برای مشاهده **pending jobs** که توسط کاربر **scheduled** شده، استفاده می شود.

ساختار دستور :

```
at [-V] [-q queue] [-f file] [-mldbv] TIME
at [-V] [-q queue] [-f file] [-mldbv] -t time_arg
at -c job [job...]
atq [-V] [-q queue]
atrm [-V] job [job...]
batch
```

گزینه های دستور:

- **-V** :شماره نسخه را به خطای استاندارد نشان می دهد و با موفقیت خارج می شود.

```
kum@kMeghana:~$ atq -V
at version 3.1.18
```

- **-q** :کار صف مشخص را نشان می دهد .یک صف را می توان با هر حرف واحد از a به z و A به Z نشان داد. برای "در"، a صف پیش فرض است و برای "دسته ای"، b صف پیش فرض است .برای یک jobs در حال اجرا ، نام صف با "=" تعیین می شود .اگر یک jobs به یک صف مشخص شده با یک حرف بزرگ ارسال شود ، با این jobs رفتار می شود مثل اینکه در زمان انجام کار به دسته ای ارسال شده است.
  - atq بدون استدلال تمام jobs معلق کاربر را نمایش می دهد.

```
kum@kMeghana:~$ atq
15      Tue Feb 26 02:15:00 2019 m kum
17      Sun Mar  3 02:15:00 2019 m kum
13      Tue Feb 26 07:50:00 2019 a kum
12      Mon Mar  4 18:30:00 2019 a kum
14      Fri Mar  1 20:30:00 2019 m kum
```

  - atq بدون آرگومان اگر ریشه کاربر باشد همه jobs معلق همه کاربران را نمایش می دهد.

```
root@kMeghana:~# atq
16      Tue Feb 26 16:00:00 2019 m root
15      Tue Feb 26 02:15:00 2019 m kum
17      Sun Mar  3 02:15:00 2019 m kum
13      Tue Feb 26 07:50:00 2019 a kum
12      Mon Mar  4 18:30:00 2019 a kum
14      Fri Mar  1 20:30:00 2019 m kum
18      Tue Feb 26 07:00:00 2019 a root
19      Tue Feb 26 06:35:00 2019 a root
```

- برای نمایش jobsی که به یک صف خاص تعلق دارند:

```
atq -q Queue
```

- در زیر تصویر تمام jobs مربوط به صف "m" نشان داده شده است.

```
root@kMeghana:~# atq -q m
16      Tue Feb 26 16:00:00 2019 m root
15      Tue Feb 26 02:15:00 2019 m kum
17      Sun Mar  3 02:15:00 2019 m kum
14      Fri Mar  1 20:30:00 2019 m kum
```

دستور جایگزین برای:atq

```
at -l
```

- نمایش تمام jobs.

```
root@kMeghana:~# at -l
16      Tue Feb 26 16:00:00 2019 m root
15      Tue Feb 26 02:15:00 2019 m kum
17      Sun Mar  3 02:15:00 2019 m kum
13      Tue Feb 26 07:50:00 2019 a kum
12      Mon Mar  4 18:30:00 2019 a kum
14      Fri Mar  1 20:30:00 2019 m kum
18      Tue Feb 26 07:00:00 2019 a root
19      Tue Feb 26 06:35:00 2019 a root
```

- نمایش تمام jobs مربوط به صف."a"

```
root@kMeghana:~# at -l -q a
13      Tue Feb 26 07:50:00 2019 a kum
12      Mon Mar  4 18:30:00 2019 a kum
18      Tue Feb 26 07:00:00 2019 a root
19      Tue Feb 26 06:35:00 2019 a root
```

دستور atrm

دستور atrmبرای حذف / حذف jobs از صف دستور استفاده می شود.

از دستور **atrm** برای حذف job مشخص شده استفاده می شود .برای حذف یک کار ، شماره job آن در دستور ارسال می شود .یک کاربر فقط می تواند job مربوط به خود را حذف کند .فقط کاربر فوق العاده می تواند هر کاری را حتی اگر متعلق به کاربر دیگری باشد حذف کند.

ساختار دستور:

```
atrm [-V] job [job...]
```

گزینه های دستور:

- **V-:** برای چاپ شماره نسخه استفاده می شود

```
atrm -V
```

```
kum@kMeghana:~$ atrm -V
at version 3.1.18
```

- **job:** شماره job ی که باید حذف شود .برای دیدن لیست مشاغل معلق از موارد زیر استفاده کنید:

```
root@kMeghana:~# atq
22      Mon Mar  4 04:15:00 2019 a kum
21      Tue Mar  5 19:30:00 2019 a kum
26      Wed Feb 27 12:15:00 2019 a root
24      Sun Mar  3 08:10:00 2019 m kum
27      Wed Mar  6 14:15:00 2019 m root
25      Sat Mar  2 18:15:00 2019 m kum
23      Sun Mar  3 07:40:00 2019 a kum
```

مثال ۱: حذف job شماره ۲۲.

```
atrm 22
```

```
root@kMeghana:~# atrm 22
root@kMeghana:~# atq
21      Tue Mar  5 19:30:00 2019 a kum
26      Wed Feb 27 12:15:00 2019 a root
24      Sun Mar  3 08:10:00 2019 m kum
27      Wed Mar  6 14:15:00 2019 m root
25      Sat Mar  2 18:15:00 2019 m kum
23      Sun Mar  3 07:40:00 2019 a kum
```

مثال ۲: حذف چندین کار در دستور واحد atrm:

```
atrm 21 26
```

```
root@kMeghana:~# atrm 21 26
root@kMeghana:~# atq
24      Sun Mar  3 08:10:00 2019 m kum
27      Wed Mar  6 14:15:00 2019 m root
25      Sat Mar  2 18:15:00 2019 m kum
23      Sun Mar  3 07:40:00 2019 a kum
```

دستور جایگزین برای atrm:

ساختار دستور:

```
at -r Job
```

مثال ۱: حذف تک job:

```
at  -r 24
```

```
root@kMeghana:~# at -r 24
root@kMeghana:~# atq
27      Wed Mar  6 14:15:00 2019 m root
25      Sat Mar  2 18:15:00 2019 m kum
23      Sun Mar  3 07:40:00 2019 a kum
```

مثال ۲: حذف چندین job :

```
at  -r 25 27
```

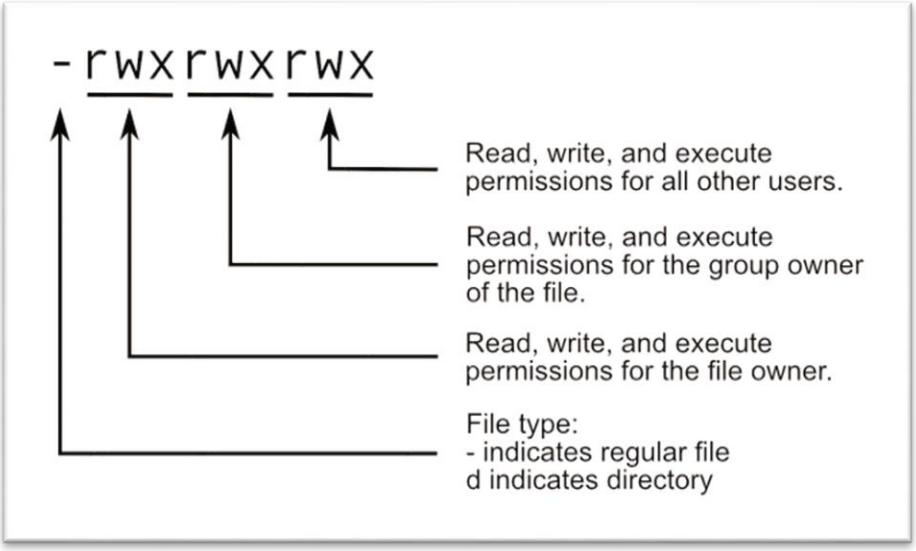
```
root@kMeghana:~# at -r 25 27
root@kMeghana:~# atq
23      Sun Mar  3 07:40:00 2019 a kum
```

فصل چهاردهم

فصل چهاردهم

مجوزها

سیستم عامل های مشابه یونیکس مانند لینوکس با سایر سیستم های کامپیوتری متفاوت است، زیرا آنها نه تنها چند وظیفه بلکه چند کاربره نیز هستند .این بدان معنی است که بیش از یک کاربر میتواند همزمان یک کامپیوتر را اداره کند .در حالی که کامپیوتر تنها دارای یک صفحه کلید و مانیتور است، هنوز هم می تواند توسط بیش از یک کاربر استفاده شود .به عنوان مثال ، اگر رایانه به یک شبکه یا اینترنت متصل شود،کاربران از راه دور می توانند از طریق (sshپوسته امن[1]) وارد سیستم شوید و کامپیوتر را اداره کنید .درواقع، کاربران از راه دور می توانند برنامه های گرافیکی را اجرا کنند و خروجی نمایش داده شده در یک کامپیوتر از راه دور .سیستم X Window از این پشتیبانی می کند .



دستور chmod

دستورchmodبرای تغییر مجوز یک فایل یا فهرست استفاده می شود .برای استفاده از آن، تنظیمات مجاز مورد نظر و فایل یا فایل هایی را که می خواهید تغییر دهید را تعیین می کنید. دو راه برای مشخص کردن مجوز وجود دارد. از تنظیمات مجوز به عنوان یک سری از بیت ها (که کامپیوترها در مورد آنها فکر می کنند) آسان است

ساختار دستور:

```
chmod [OPTION]... MODE[,MODE]... FILE...
chmod [OPTION]... OCTAL-MODE FILE...
chmod [OPTION]... --reference=RFILE FILE...
```

.در اینجا مشخص است که چگونه کار می کند :

```
rwx  rwx  rwx = 111 111 111
 rw- rw- rw- = 110 110 110
 rwx   ---   --- = 111 000 000

                                  یا

 rwx = 111  در دودویی = ۷
 rw- = 110  در دودویی = 6
```

[1] secure shell

rx = 101 در دودویی = 5
r -- = 100 در دودویی = 4

معادل عدد ۷ در مبنای دودویی ۱۱۱ است و همین طور به ترتیب در خط بالا توضیح داده شد.

در اینجا یک جدول اعداد است که تمام تنظیمات رایج را پوشش می دهد .

مقدار	توضیح
777	**(rwxrwxrwx)**هیچ محدودیتی در مجوزها وجود ندارد .هر کسی می تواند کاری انجام دهد .به طور کلی یک تنظیم مطلوب نیست
755	**(rwxr-xr-x)**صاحب پرونده ممکن است فایل را بخواند، نوشت و اجرا کند .همه دیگران ممکن است فایل را بخوانند و اجرا کنند .این تنظیم رایج برای برنامه هایی است که توسط همه کاربران مورد استفاده قرار می گیرد.
700	**(rwx -- -- --)**صاحب پرونده ممکن است فایل را بخواند، نوشتن و اجرا کند .هیچ کس هیچ حقوق ندارد این تنظیم برای برنامه هایی مفید است که تنها مالک می تواند از آن استفاده کند و باید از دیگران خصوصی باشد.
666	**(rw-rw-rw-)**همه کاربران ممکن است فایل را بخوانند و بنویسند.
644	**(rw-r-r --)**صاحب ممکن است فایل را بخواند و نوشت، در حالی که دیگران فقط فایل را می خوانند .یک تنظیم رایج برای فایل های داده ای که همه می توانند بخوانند، اما تنها مالک ممکن است تغییر کند.
600	**(rw -- -- ---)**مالک ممکن است فایل را بخواند و نوشت .همه دیگران حقوق ندارند .یک تنظیم رایج برای فایلهای داده ای که مالک می خواهد خصوصی نگه دارد.

گزینه های دستور:

- c, changes مانند verbose، اما فقط در صورتی که تغییری ایجاد شود، گزارش داده می‌شود.
- nopreserveroot: '/'را به‌صورت خاص نگیرید (پیش‌فرض).
- Preserveroot عملیات بازگشتی روی '/' انجام ندهید.
- f, silent, quiet بیشتر پیام‌های خطا را سرکوب کنید.
- v, verbose برای هر فایل پردازش شده، تشخیصی را خروجی دهید.
- reference=RFILE از حالت RFILE به جای مقادیر MODE استفاده کنید.
- R, recursive فایل‌ها و دایرکتوری‌ها را به صورت بازگشتی تغییر دهید.
- Help این راهنما را نمایش دهید و خارج شوید.
- Version اطلاعات نسخه را چاپ کنید و خارج شوید

مثال های دستور:

- c, changes:

```
chmod -c 644 file.txt
```

- nopreserveroot:

```
chmod --nopreserveroot 600 folder
```

- Preserveroot:

```
chmod --Preserveroot 755 file.txt
```

- f, silent, quiet:

```
chmod -f 777 file.txt
```

- v, verbose:

```
chmod -v 600 file.txt
```

- reference=RFILE:

```
chmod --reference=reference_file file.txt
```

- R, recursive:

```
chmod -R 755 folder
```

- Help:

```
chmod --help
```

- Version:

```
chmod --version
```

دستور chown

از دستور **chown** برای تغییر مالک یا گروه فایل استفاده می شود .هر زمان که می خواهید مالکیت خود را تغییر دهید می توانید از دستور chown استفاده کنید.

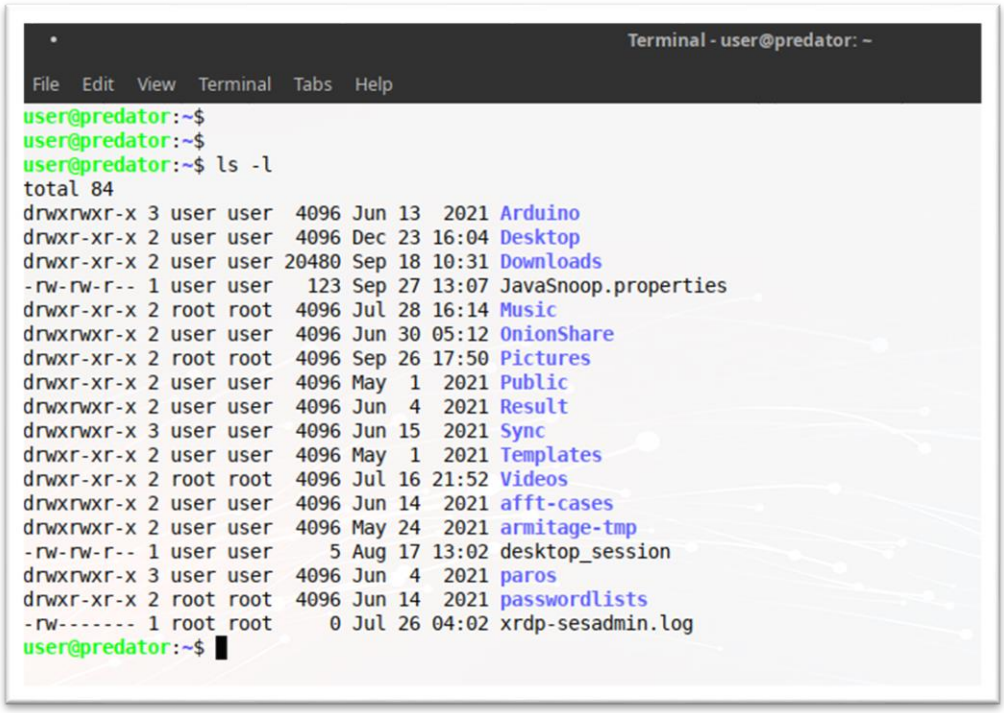
ساختار دستور:

```
chown [OPTION]… [OWNER][:[GROUP]] FILE…
chown [OPTION]… –reference = RFILE FILE
```

مثال :برای تغییر مالک فایل/:

```
chown owner_name file_name
```

در مورد ما فایل هایی به شرح زیر داریم:



اکنون اگر در فایل خود از **file1.txt** استفاده کنم ، برای تغییر مالکیت از **نحو** زیر استفاده خواهم کرد:

```
chown username file1.txt
```

که در آن کاربر دیگری در سیستم است .فرض کنید اگر شما کاربری به نام user1 هستید و می خواهید مالکیت را به root تغییر دهید (جایی که فهرست فعلی شما user1 است .قبل از **نحو** از "sudo" استفاده کنید.

```
sudo chown root file1.txt
```

گزینه های دستور:

- c, changes مشابه verbose، اما فقط در صورتی که تغییری ایجاد شود، گزارش داده می‌شود.
- Dereference: تأثیر بر روی مرجع هر لینک نمادی (پیش‌فرض)، نه خود لینک نمادی.
- h, nodereference: تأثیر بر روی هر لینک نمادی به جای هر فایل مرجع (فقط در سیستم‌هایی که می‌توانند مالکیت یک لینک نمادی را تغییر دهند، مفید است).
- from=CURRENT_OWNER:CURRENT_GROUP: تغییر مالک و/یا گروه هر فایل فقط اگر مالک و/یا گروه فعلی آن با مقادیر مشخص شده در اینجا مطابقت داشته باشد. هر دو می‌توانند حذف شوند، در این صورت برای ویژگی حذف شده نیاز به مطابقت نیست.
- Nopreserveroot '/': را به‌صورت خاص نگیرید (پیش‌فرض).
- Preserveroot: عملیات بازگشتی روی '/' انجام ندهید.
- f, silent, quiet: بیشتر پیام‌های خطا را سرکوب کنید.
- reference=RFILE: از مالک و گروه RFILE به جای مقادیر OWNER:GROUP استفاده کنید.
- R, recursive: فایل‌ها و دایرکتوری‌ها را به صورت بازگشتی تغییر دهید.
- v, verbose: برای هر فایل پردازش شده، تشخیصی را خروجی دهید.
- گزینه‌های زیر تغییراتی در نحوه پویش سلسله‌مراتبی را ایجاد می‌کنند هنگامی که گزینه R نیز مشخص شده است. اگر بیش از یک گزینه مشخص شود، تنها آخرین گزینه اعمال می‌شود:
- H: اگر آرگومان خط فرمان یک لینک نمادی به یک دایرکتوری باشد، آن را پویش کنید.

- L :همه لینک‌های نمادی به دایرکتوری‌ها را پویش کنید.
- P :هیچ لینک نمادی را پویش نکنید (پیش‌فرض) .

مثال های دستور:

- **-c:** هنگام ایجاد تغییر در فایل گزارش می شود.

مثال :

```
chown -c username file1.txt
```

- **-v:**برای نشان دادن اطلاعات پر سر و صدا برای هر فایل پردازش شده استفاده می شود.

مثال :

```
chown -v username file1.txt
```

-f :اکثر پیام های خطا را فشرده می کند .هنگامی که اجازه تغییر مجوزهای گروه را ندارید و خطا را نشان میدهد ، این گزینه مالکیت را با اجبار و بی صدا تغییر می دهد.

مثال ها:

برای تغییر مالکیت گروه در مورد ما من از group1 به عنوان یک گروه در سیستم استفاده می کنم .برای تغییر مالکیت ما استفاده خواهیم کرد

```
chown: group1 file1.txt
```

می بینید که مجوزهای گروه از root به group1 تغییر کرده است ، اگر از گزینه v- استفاده کنید آن را گزارش می دهد .برای تغییر گروه فقط باید ":" اضافه کنیم.

برای تغییر مالک و همچنین گروه

دوباره گرفتن username به عنوان کاربر و group1 به عنوان یک گروه در سیستم

```
chown username:group1 ali
```

تغییر مالک فقط از مالکیت خاص

فرض کنید می خواهیم مالکیت را از username به root تغییر دهیم در جایی که مالک فعلی فقط باید استاد باشد.

```
chown --from= username root ali
```

برای تغییر گروه از یک گروه خاص:

```
chown --from=:group1 root greek1
```

در اینجا ، گروه ali به root تغییر یافته است.

برای کپی مالکیت یک فایل در فایل دیگر:

```
chown --reference=ali  hossein
```

برای تغییر مالکیت چندین فایل:

```
chown username:group ali reza
```

دستور chattr

صفات دیگری وجود دارد که محدودیت های بیشتری را در عملیات مجاز در فایل ها فراهم می کند.برای مثال ، جلوگیری از تغییر نام، انتقال، حذف، یا حتی تغییر پرونده.

ساختار دستور :

```
chattr [ -RVf ] [ -v version ] [ mode ] files...
```

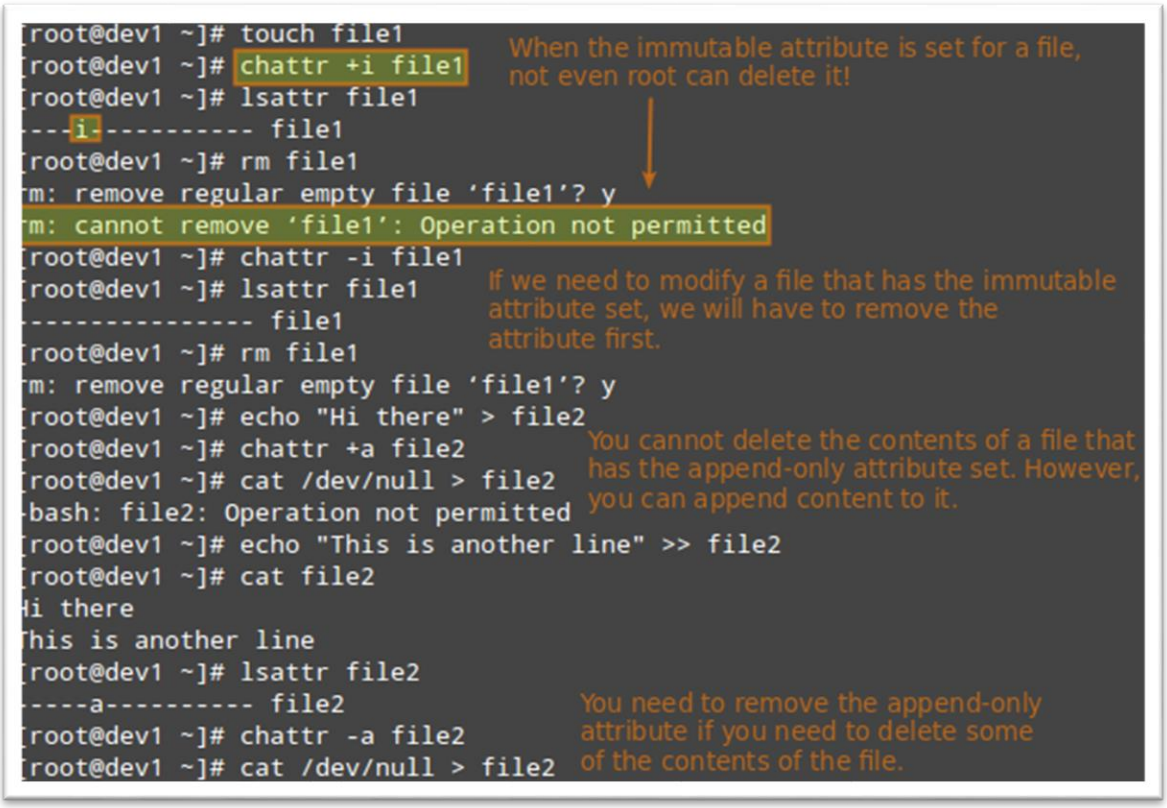
گزینه های دستور :

- **R**–به طور بازگشتی ویژگی‌های دایرکتوری‌ها و محتوای آن‌ها را تغییر دهید.
- V–، خروجی chattr را تفصیلی نمایش دهید و نسخه برنامه را چاپ کنید.
- **f** – ، اکثر پیام‌های خطا را سرکوب کنید.
- **v**–، شماره نسخه/تولید فایل را تنظیم کنید.
- **a-** برای لیست کردن تمام فایل های یک فهرست استفاده می شود که شامل نام آن با یک(’.‘) نیز می شود.
- **d-** این گزینه به جای فهرست کردن مطالب ، فهرست ها را به عنوان فایل های معمولی لیست می کند.

مثال های دستور :

```
# chattr +i file1
# chattr + file2
```

پس از اجرای این دو دستور، file1غیر قابل تغییر خواهد بود (به این معنی که نمی توان آن را جابجاکرد، تغییر نام، اصلاح یا حذف کرد) در حالی که file2تنها حالت add-only را وارد می کند (فقط می تواند در حالت آپلود برای نوشتن باز شود.



دستور chattr دستوردر لینوکس یک دستور فایل سیستم است که برای تغییر ویژگی های یک فایل در یک فهرست استفاده می شود .کاربرد اصلی این دستور این است که چندین فایل غیر از کاربر ریشه قادر به تغییر در چندین فایل نیستند .همانطور که می دانیم لینوکس یک سیستم عامل چند کاربره است ، این احتمال وجود دارد که کاربر بتواند فایلی را که باعث نگرانی کاربر دیگری است حذف کند .برای جلوگیری از چنین سناریوهایی ، لینوکس “ *chattr* ” را فراهم می کند . به طور خلاصه ، “chattr” می تواند یک فایل را تغییرناپذیر، غیرقابل حذف، فقط قابل پیوست و موارد دیگر کند.

chattr [-RVf] [-v version] [mode] files...

در ابتدای یک رشته حالت ، یکی از عملگرهای زیر باید ظاهر شود:

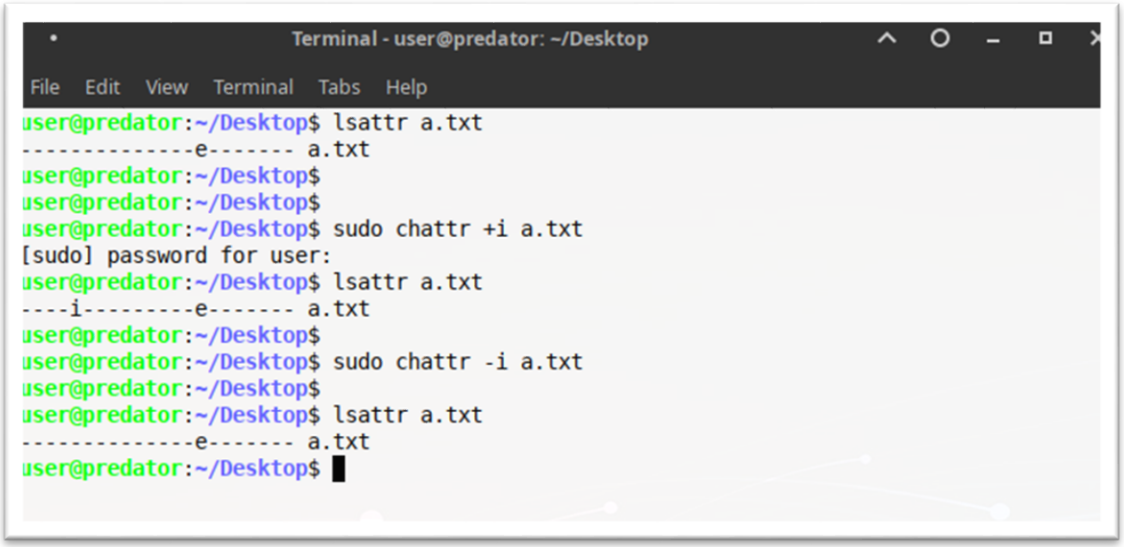
- ' + ' :افزودن ویژگی های انتخاب شده به ویژگی های موجود فایل ها.
- ' - ' :باعث می شود صفات انتخاب شده حذف شوند.
- ' = ' :باعث می شود صفات انتخاب شده تنها ویژگی هایی باشند که فایل ها دارند.

قالب حالت نمادین:

{+ | - | =} [aAcCdDeijsStTu]

استفاده از دستور chattr فرمان :دستور chattr را می توان مورد استفاده قرار گیرد برای حفظ برخی از فایل های سیستم که بسیار مهم هستند و باید در کامپیوتر میزبان بدون توجه به آنچه باقی می ماند .همچنین برای غیر قابل حذف یا غیر قابل تغییر کردن فهرست برای کاربران غیر از superuser ، این مورد ضروری است .استفاده معمول از "chattr" به شرح زیر است- :

۱. **تغییرناپذیر بودن فایل :**این دستور نام فایل file.txt را تغییرناپذیر می کند ، از این رو تا زمانی که ویژگی های فایل دوباره تغییر نکند ، دیگر هیچ عملیاتی روی این فایل امکان پذیر نیست.



۲. **باز کردن فایل فقط در حالت ضمیمه :**از پرچم ' a ' برای باز کردن فایل فقط در حالت ضمیمه استفاده می شود .در نتیجه ، فقط می توان آن را پیوست کرد و داده های قبلی را نمی توان اصلاح کرد.

```
tharva-kango@AKUbuntu:~/Desktop/Articles/temp$ ls
ile.txt
tharva-kango@AKUbuntu:~/Desktop/Articles/temp$ lsattr
------------e--- ./file.txt
tharva-kango@AKUbuntu:~/Desktop/Articles/temp$ sudo chattr +a file.txt
sudo] password for atharva-kango:
tharva-kango@AKUbuntu:~/Desktop/Articles/temp$ lsattr
----a--------e--- ./file.txt
tharva-kango@AKUbuntu:~/Desktop/Articles/temp$ cat file.txt
ello Geeks
tharva-kango@AKUbuntu:~/Desktop/Articles/temp$ echo "Hello World" > file.txt
ash: file.txt: Operation not permitted
tharva-kango@AKUbuntu:~/Desktop/Articles/temp$ echo "Hello World" >> file.txt
tharva-kango@AKUbuntu:~/Desktop/Articles/temp$ cat file.txt
ello Geeks
ello World
tharva-kango@AKUbuntu:~/Desktop/Articles/temp$
```

۳. **ایمن سازی فهرست ها** :از پرچم‘ i + می توان برای یک فهرست استفاده کرد (همانطور که در زیر نشان داده شده است) تا فهرست تغییر ناپذیر شود .همچنین ، از‘ flag -R در اینجا استفاده شده است ، که باعث می شود فراخوانی تماس گرفته شود و همه زیر فایل ها و فهرست ها نیز تغییرناپذیر شوند.

```
atharva-kango@AKUbuntu:~/Desktop/Articles/Test$ ls
temp
atharva-kango@AKUbuntu:~/Desktop/Articles/Test$ sudo chattr -R +i temp/
atharva-kango@AKUbuntu:~/Desktop/Articles/Test$ rm -rf temp/
rm: cannot remove 'temp/file.txt': Operation not permitted
atharva-kango@AKUbuntu:~/Desktop/Articles/Test$ lsattr -R  temp/
----i---------e--- temp/file.txt
atharva-kango@AKUbuntu:~/Desktop/Articles/Test$ sudo chattr -R -i temp/
atharva-kango@AKUbuntu:~/Desktop/Articles/Test$ lsattr -R  temp/
--------------e--- temp/file.txt
atharva-kango@AKUbuntu:~/Desktop/Articles/Test$
```

توجه: **از** lsattr دستور برای دیدن ویژگی های فایل ها در یک فهرست استفاده می شود .در اینجا ، لازم به ذکر است که پرچم **e** در فایل قبلا تنظیم شده است . موارد موجود وابسته به سیستم فایل هستند .به ندرت قابل جابجایی هستند.

دستور chgrp

دستور chgrp برای تغییر مالکیت گروه یک فایل استفاده می شود.

ساختار دستور:

chgrp [OPTION]… GROUP FILE…
chgrp [OPTION] re –reference = RFILE FILE

گزینه های دستور :

- c یا changes :این دستور، مشابه دستور verbose عمل می‌کند، اما فقط در صورتی که تغییری ایجاد شود، گزارش می‌دهد.
- Dereference :با استفاده از این دستور، تغییرات را بر روی فایل‌های مرتبط با لینک‌های نمادین (symbolic link) اعمال می‌کند (پیش‌فرض).

- h یا nodereference: با این دستور، تغییرات را بر روی خود لینک‌های نمادین اعمال می‌کند (مفید فقط در سیستم‌هایی که می‌توانند مالکیت یک لینک نمادین را تغییر دهند).
- Nopreserveroot: با این دستور، '/' را به صورت خاص نمی‌گیرد (پیش‌فرض).
- Preserveroot: با این دستور، عملیات را به صورت بازگشتی روی '/' انجام نمی‌دهد.
- f یا silent یا quiet: این دستور، پیام‌های خطا را کاهش می‌دهد.
- reference=RFILE: با استفاده از این دستور، گروه فایل‌ها را با گروه فایل مرجع (RFILE) تغییر می‌دهد.
- R یا recursive: این دستور، عملیات را به صورت بازگشتی روی فایل‌ها و دایرکتوری‌ها انجام می‌دهد.
- v یا verbose: این دستور، برای هر فایل پردازش شده، یک تشخیصی (diagnostic) نمایش می‌دهد.

همچنین، دستورات زیر نحوه عبور از سلسله مراتب را تغییر می‌دهند هنگامی که دستور R نیز مشخص شده است. اگر بیش از یک دستور مشخص شده باشد، تنها آخرین دستور اعمال می‌شود:

- H: اگر آرگومان خط فرمان یک لینک نمادین به یک دایرکتوری باشد، از آن عبور می‌کند.
- L: از هر لینک نمادین به دایرکتوری‌ها عبور می‌کند.
- P: از هیچ لینک نمادینی عبور نمی‌کند (پیش‌فرض).

مثال های دستور:

-c یا --changes:

```
chgrp -c group1 file1
```

--dereference:

```
chgrp --dereference group2 symlink1
```

-h یا --no-dereference:

```
chgrp -h group3 symlink2
```

--no-preserve-root:

```
chgrp --no-preserve-root group4 file2
```

--preserve-root:

```
chgrp --preserve-root group5 /
```

-f یا --silent یا --quiet:

```
chgrp -f group6 file3
```

--reference=RFILE:

```
chgrp --reference=file1 file2
```

-R یا --recursive:

```
chgrp -R group7 directory1
```

-v یا --verbose:

```
chgrp -v group8 file4
```

تغییرات در سلسله مراتب:

-H:

عبور از لینک نمادین به دایرکتوری:

```
chgrp -R -H group9 directory2
```

-L:

عبور از هر لینک نمادین به دایرکتوری:

```
chgrp -R -L group10 directory3
```

-P:

عبور از هیچ لینک نمادینی:

```
chgrp -R -P group11 directory4
```

دستور getfacl

دستور getfacl در سیستم‌عامل‌های Unix/Linux برای نمایش و مدیریت لیست کنترل دسترسی فایل‌ها و فهرست‌ها به صورت محیطی است.

ساختار دستور:

```
getfacl [-aceEsRLPtpndvh] file ...
getfacl [-aceEsRLPtpndvh] -
```

گزینه‌های دستور:

- a یا access: نمایش لیست کنترل دسترسی فایل.
- d یا default: نمایش لیست کنترل دسترسی پیش‌فرض.
- c یا omitheader: عدم نمایش سربرگ نظریه (سه خط اول خروجی هر فایل).
- e یا alleffective: نمایش تمام نظریه‌های حقوق مؤثر، حتی اگر با مقادیر تعریف شده توسط ورودی ACL یکسان باشد.
- E یا noeffective: عدم نمایش نظریه‌های حقوق مؤثر.
- s یا skipbase: رد کردن فایل‌هایی که فقط شامل ورودی‌های ACL پایه (مالک، گروه، دیگران) هستند.
- R یا recursive: لیست کردن ACLها برای تمام فایل‌ها و دایرکتوری‌ها به صورت بازگشتی.
- L یا logical: پیمایش منطقی، دنبال لینک‌های نمادین به دایرکتوری‌ها. رفتار پیش‌فرض دنبال کردن لینک‌های نمادین در آرگومان‌ها و رد کردن لینک‌های نمادین در زیردایرکتوری‌ها است. تنها در ترکیب با R مؤثر است.
- P یا physical: پیمایش فیزیکی، دنبال کردن لینک‌های نمادین به دایرکتوری‌ها نمی‌شود. این همچنین آرگومان‌های لینک‌های نمادین را نیز رد می‌کند. تنها در ترکیب با R مؤثر است.
- t یا tabular: استفاده از یک فرمت جدولی جایگزین. ACL و ACL پیش‌فرض به صورت کنار هم نمایش داده می‌شوند. مجوزهایی که به دلیل ورودی ACL ماسک نامآور نیستند، با حروف بزرگ نمایش داده می‌شوند. نام‌های برچسب ورودی‌های ACL_USER_OBJ و ACL_GROUP_OBJ نیز با حروف بزرگ نمایش داده می‌شوند که در شناسایی این ورودی‌ها کمک می‌کند.
- p یا absolutenames: حذف کاراکترهای اسلش اولی ('/') نمی‌شود. رفتار پیش‌فرض حذف کاراکترهای اسلش اولی است.
- n یا numeric: لیست کردن شناسه‌های عددی کاربر

مثال های دستور:

نمایش فقط دسترسی‌های ACL برای کاربران و گروه‌ها.

```
getfacl -a file.txt
```

خروجی دستور: نمایش فقط دسترسی‌های ACL برای کاربران و گروه‌ها برای فایل file.txt.

حذف سربرگ در خروجی.

```
getfacl -c file.txt
```

خروجی دستور: نمایش لیست کنترل دسترسی برای فایل file.txt بدون سربرگ.

تنظیم اندازه تب‌ها در خروجی.

```
getfacl -e 4 file.txt
```

خروجی دستور: نمایش لیست کنترل دسترسی برای فایل file.txt با اندازه تب ۴.

نادیده گرفتن ACL پیش‌فرض.

```
getfacl -E file.txt
```

خروجی دستور: نادیده گرفتن ACL پیش‌فرض برای فایل file.txt.

نادیده گرفتن خطاها در صورت عدم وجود ACL.

```
getfacl -s file.txt
```

خروجی دستور: نادیده گرفتن خطاها در صورت عدم وجود لیست کنترل دسترسی برای فایل file.txt.

نمایش ACL برای فایل‌ها و زیرفهرست‌ها به صورت بازگشتی.

```
getfacl -R directory
```

خروجی دستور: نمایش لیست کنترل دسترسی برای فهرست directory و تمام زیرفهرست‌ها به صورت بازگشتی.

نمایش ACL برای لینک‌های نمادین به جای فایل یا فهرست اصلی.

```
getfacl -L symbolic_link
```

خروجی دستور: نمایش لیست کنترل دسترسی برای لینک نمادین symbolic_link.

نمایش ACL برای فایل یا فهرست اصلی به جای لینک‌های نمادین.

```
getfacl -P file.txt
```

خروجی دستور: نمایش لیست کنترل دسترسی برای فایل file.txt به صورت فیزیکی.

نمایش ACL به صورت جدولی.

```
getfacl -t directory
```

خروجی دستور: نمایش لیست کنترل دسترسی برای فهرست directory به صورت جدولی.

نادیده گرفتن ACLهای پیش‌فرض.

```
getfacl -p file.txt
```

خروجی دستور: نادیده گرفتن ACLهای پیش‌فرض برای فایل file.txt.

نمایش شماره‌های کاربران و گروه‌ها به جای نام‌ها.

```
getfacl -n file.txt
```

خروجی دستور: نمایش لیست کنترل دسترسی برای فایل file.txt با شماره‌های کاربران و گروه‌ها.

نمایش ACL پیش‌فرض.

```
getfacl -d directory
```

خروجی دستور: نمایش ACL پیش‌فرض برای فهرست directory.

نمایش شماره نسخه getfacl.

```
getfacl -v
```

خروجی دستور: نمایش شماره نسخه getfacl.

دستور setfacl

دستور setfacl در سیستم‌عامل‌های Unix/Linux برای تنظیم و مدیریت لیست کنترل دسترسی فایل‌ها و فهرست‌ها استفاده می‌شود.

ساختار دستور:

```
setfacl [-bkndRLPvh] [{-m|-x} acl_spec] [{-M|-X} acl_file] file ...
setfacl --restore=file
```

گزینه‌های دستور:

- b یا removeall: حذف تمامی ورودی‌های ACL گسترده. ورودی‌های ACL پایه مربوط به مالک، گروه و دیگران حفظ می‌شوند.
- k یا removedefault: حذف ACL پیش‌فرض. اگر ACL پیش‌فرض وجود نداشته باشد، هیچ هشداری صادر نمی‌شود.
- n یا nomask: محاسبه مجدد ماسک حقوق اثرگذار نشود. رفتار پیش‌فرض setfacl این است که ماسک ACL مجدداً محاسبه شود، مگر اینکه ورودی‌ای برای ماسک صراحتاً داده شده باشد. ماسک به اتحاد تمام مجوزهای گروه مالکیت و تمام ورودی‌های کاربر و گروه نام‌دار تنظیم می‌شود. (این دقیقاً ورودی‌هایی هستند که توسط ماسک تأثیر می‌پذیرند).
- Mask: محاسبه مجدد ماسک حقوق اثرگذار باشد، حتی اگر ورودی‌ای برای ماسک ACL صراحتاً داده شده باشد. (جزئیات را در گزینه n ببینید).
- d یا default: تمام عملیات به ACL پیش‌فرض اعمال می‌شود. ورودی‌های ACL معمولی در مجموعه ورودی به ورودی‌های ACL پیش‌فرض ارتقاء می‌یابند. ورودی‌های ACL پیش‌فرض در مجموعه ورودی دور انداخته می‌شوند. (در صورت رخ دادن این موضوع، یک هشدار صادر می‌شود).
- restore=file: بازگردانی پشتیبانی از مجوز ایجاد شده توسط getfacl R’ ‘یا مشابه آن. تمام مجوزهای یک زیردرخت کامل دایرکتوری با استفاده از این مکانیزم بازیابی می‌شوند. اگر ورودی شامل نظرات مالک یا نظرات گروه باشد، setfacl تلاش می‌کند مالک و گروه مالکیت را بازیابی کند. اگر ورودی شامل نظرات پرچم‌ها (که بیت‌های setuid، setgid و sticky را تعریف می‌کنند) باشد،

setfacl این سه بیت را مطابقت داده و در غیر این صورت آن‌ها را پاک می‌کند. این گزینه نمی‌تواند با گزینه‌های دیگر به جز ‘–test’ ترکیب شود.

test: حالت آزمایشی. به جای تغییر ACL فایل‌ها، ACLهای نتیجه‌ی حاص

مثال های دستور:

تغییر دادن دسترسی‌های ACL برای یک فایل یا فهرست.

```
setfacl -m u:ali:rwx file.txt
```

خروجی دستور: تغییر دسترسی‌های ACL برای فایل file.txt برای کاربر ali به حالت خواندن، نوشتن و اجرا کردن.

حذف یک مجوز ACL از یک فایل یا فهرست.

```
setfacl -x u:bob file.txt
```

خروجی دستور: حذف مجوز ACL کاربر bob از فایل file.txt.

اعمال تغییرات بر روی فایل‌ها و زیرفهرست‌ها به صورت بازگشتی.

```
setfacl -R -m u:ali:rwx directory
```

خروجی دستور: تغییر دسترسی‌های ACL برای فهرست directory و تمام زیرفهرست‌ها به حالت خواندن، نوشتن و اجرا کردن برای کاربر ali.

تنظیم دسترسی‌های ACL به صورت کامل برای یک فایل یا فهرست.

```
setfacl -s acl_file.txt
```

خروجی دستور: تنظیم دسترسی‌های ACL برای فایل acl_file.txt بر اساس محتوای فایل ACL.

حذف تمام دسترسی‌های ACL از یک فایل یا فهرست.

```
setfacl -b file.txt
```

خروجی دستور: حذف تمام دسترسی‌های ACL از فایل file.txt.

تنظیم دسترسی‌های ACL به عنوان پیش‌فرض برای زیرفهرست‌ها.

```
setfacl -d -m u:alice:rwx directory
```

خروجی دستور: تنظیم دسترسی‌های ACL به حالت پیش‌فرض برای زیرفهرست‌ها برای کاربر alice.

دسترسی به حساب root

یکی از راه هایی که کاربران می توانند به حساب root دسترسی داشته باشند، تایپ کردن دستور زیر است.

```
$ su
```

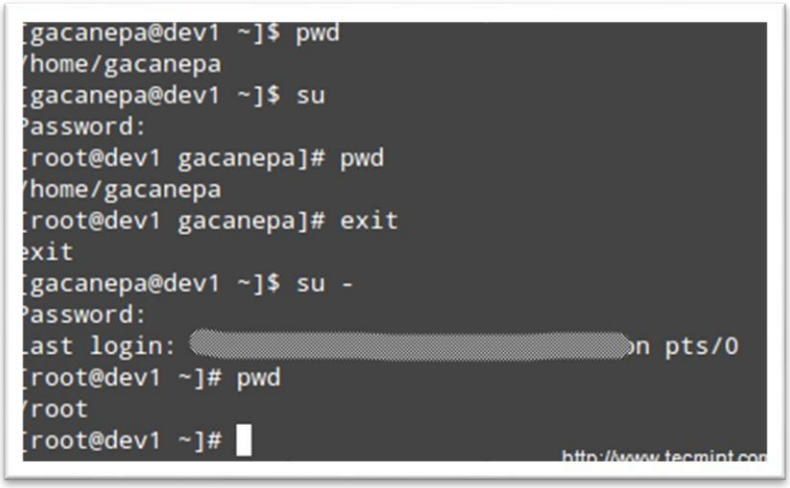
یا

```
$ su root
```

و سپس با رمز عبور root وارد شوید.اگر احراز هویت موفق باشد، به عنوان root با فهرست کاری فعلی به همان شیوه که قبلا بود، وارد system شوید.اگر بخواهید به جای آن در فهرست home از root قرار دهید، اجرا کنید.

```
$ su -
```

و سپس رمز عبور root را وارد کنید. دسترسی sudo به کاربران را فعال کنید



برای دسترسی به نوع دسترسی های sudo ، مدیر system باید فایل etc/sudoers / را ویرایش کند.توصیه می شود که این فایل با استفاده از دستور visudo[1] به جای باز کردن آن به طور مستقیم با یک ویرایشگر متن ویرایش شود.

```
# visudo
```

این فایل فایل /etc/sudoers را با استفاده از vim باز می کند.

```
secure_path = “/ usr / sbin: / usr / bin: / sbin”
root  ALL = (ALL) ALL
hossein  ALL = (ALL) ALL
 hosein  ALL = / bin / yum update
reza ALL = NOPASSWD: / bin / updatedb
%admin ALL = (ALL) ALL
```

بیایید به آنها نگاه کنیم.

```
secure_path = “/ usr / sbin: / usr / bin: / sbin: / usr / local / bin”
```

این خط اجازه می دهد فهرست هایی را که برای sudo استفاده می شوند مشخص کنید و برای جلوگیری از استفاده از فهرست های خاص که می تواند به system آسیب برساند استفاده می شود.خطوط بعدی برای تعیین مجوز استفاده می شوند.

```
root  ALL = (ALL) ALL
```

۱. اولین کلید ALL نشان دهنده این است که این قانون برای همه میزبان آنها اعمال می شود.
۲. دوم ALL نشان می دهد که کاربر در ستون اول میتواند دستورات را با امتیازات هر کاربر اجرا کند.
۳. دوم ALL به معنی هر دستور می تواند اجرا شود.

```
 hosein  ALL = / bin / yum update
```

اگر کاربر پس از علامت = (تساوی) مشخص نشده باشد، sudo پیش فرض کاربر root را دارد.در این حالت، کاربر hosein قادر خواهد بود به روز رسانی yum را به عنوان root اجرا کند.

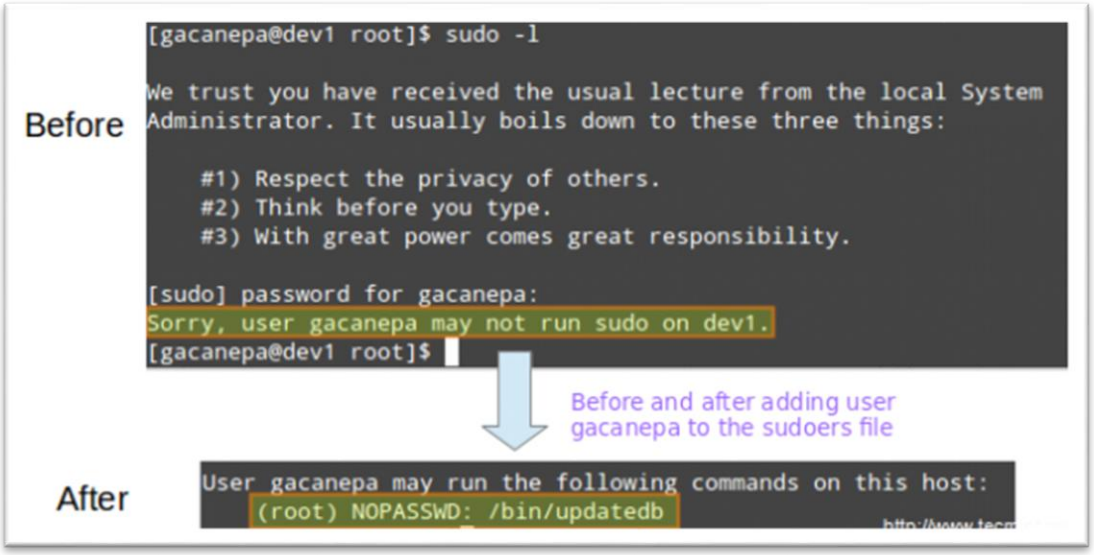
```
reza ALL = NOPASSWD: / bin / updatedb
```

دستور NOPASSWD به کاربر اجازه می دهد تا بدون نیاز به وارد کردن کلمه عبور خود:
دستور /bin/updatedb را اجرا کند.

[1] visual sudo

```
%admin ALL = (ALL) ALL
```

نماد[1] % نشان می دهد که این خط برای یک گروه به نام " admin " اعمال می شود.معنای بقیه خط همان یک کاربر معمولی است.این بدان معنی است که اعضای گروه " admin " می توانند تمام دستورات را به عنوان هر کاربر در تمام host اجرا کنند. برای دیدن اینکه چه امتیاز هایی توسط sudo ارائه می شوند، از گزینه " l- " برای لیست کردن آنها استفاده کنید.



دستور chage

از دستور **chage** برای مشاهده و تغییر اطلاعات منقضی شده رمز عبور کاربر استفاده می شود .این دستور هنگامی استفاده می شود که ورود به سیستم برای مدت زمان محدودی برای کاربر ارائه شود یا اینکه لازم است گذرواژه ورود به سیستم را به زمان دیگری تغییر دهید .با کمک این دستور می توانیم اطلاعات مربوط به پیری یک حساب ، تاریخ تغییر رمز عبور، تنظیم زمان تغییر رمز ورود ، قفل کردن حساب پس از مدت زمان مشخص و غیره را مشاهده کنیم.

دستور chage در زیر آورده شده است:

ساختار دستور:

```
chage [options] LOGIN
```

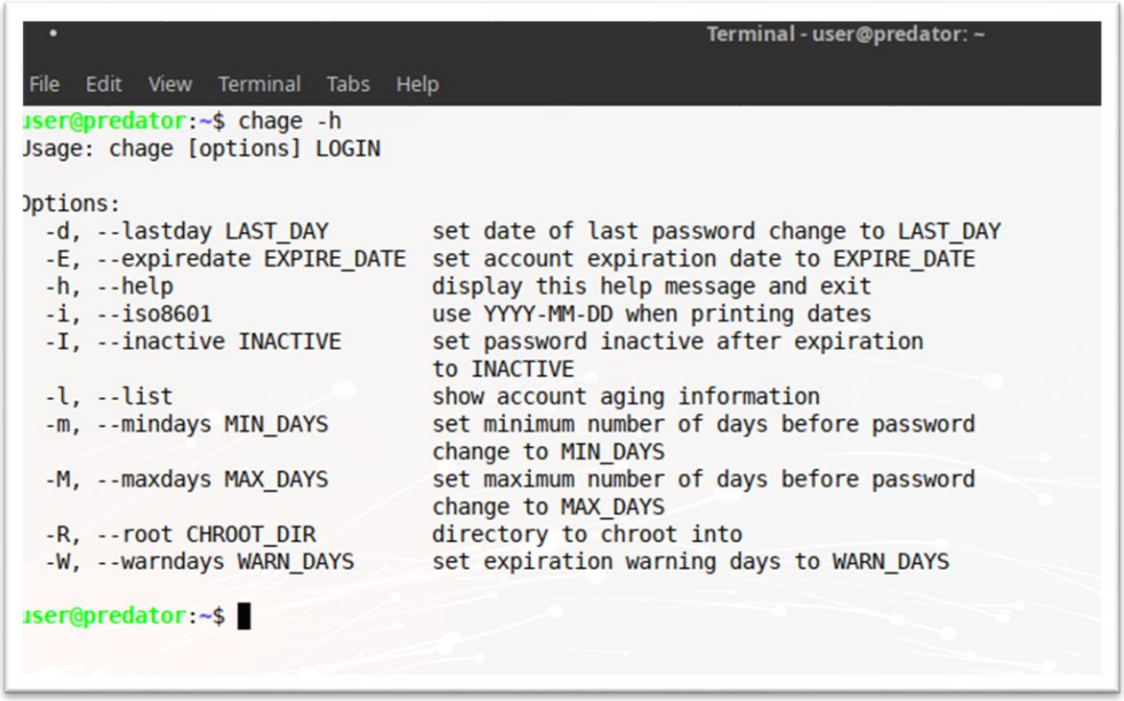
برای مشاهده لیست گزینه های قابل استفاده با دستور chage از گزینه help استفاده کنید

ورودی :

```
chage -h
```

[1] symbol

خروجی دستور:

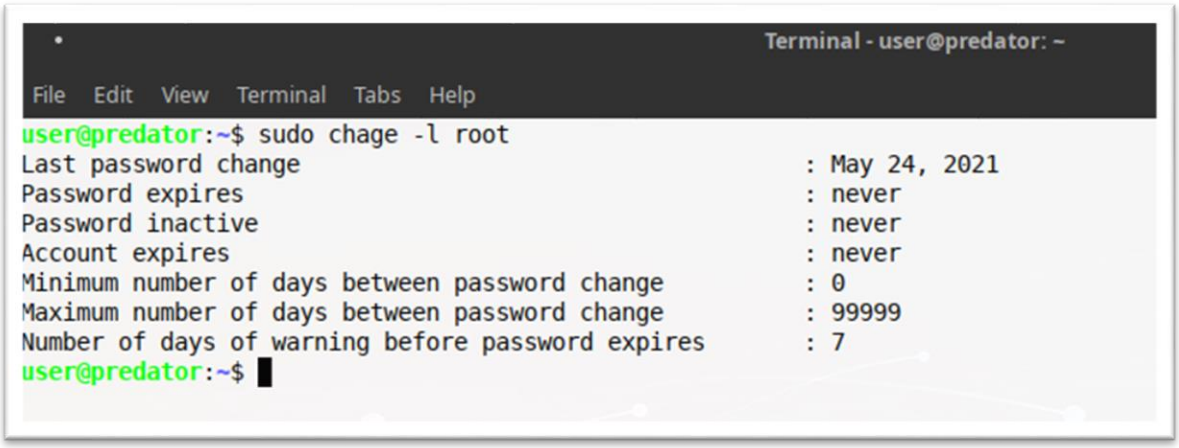


مثال های دستور:

گزینه-l برای مشاهده اطلاعات انقضا حساب از این گزینه استفاده کنید .برای مشاهده اطلاعات ریشه، من از sudoاستفاده می کنم:

```
sudo chage -l root
```

خروجی دستور:



گزینه -d: از این گزینه برای تنظیم آخرین تاریخ تغییر رمز عبور به تاریخ مشخص شده در دستور استفاده کنید .برای تغییر اطلاعات ریشه ، من از کلمه کلیدی "sudo" استفاده می کنم .بعلاوه من از گزینه -l برای مشاهده تاریخ تغییر یافته استفاده می کنم.

```
sudo chage -d 2018-12-01 root
```

گزینه -E: . برای تعیین تاریخ انقضا حساب باید از این گزینه استفاده کنید .به منظور تغییر اطلاعات ریشه ، من از کلمه کلیدی sudoاستفاده می کنم. بعلاوه من از گزینه -l برای مشاهده تاریخ تغییر یافته استفاده می کنم.

ورودی:

```
sudo chage -E root
```

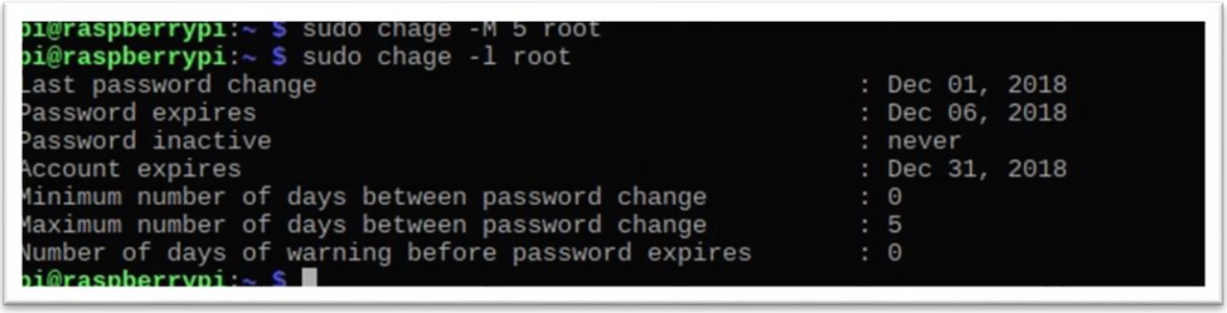
خروجی دستور:



گزینه M یا m: -از این گزینه برای تعیین حداکثر و حداقل تعداد روز بین تغییر رمز عبور استفاده کنید .برای تغییر اطلاعات ریشه ، من از کلمه کلیدی sudo استفاده می کنم .بعلاوه من از گزینه l- برای مشاهده دوره تغییر یافته استفاده می کنم.

```
sudo chage -M 5 root
```

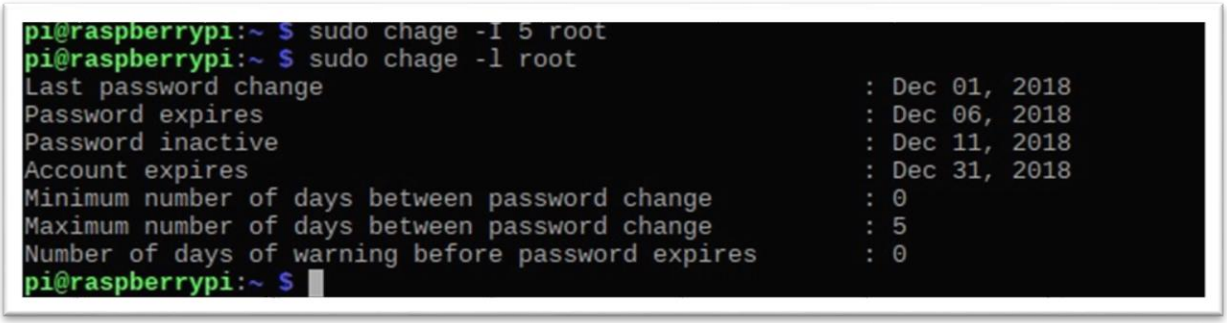
خروجی دستور:



گزینه I: از این گزینه برای تعیین تعداد روزهایی که حساب باید پس از انقضا غیرفعال باشد استفاده کنید .لازم است که کاربر پس از انقضا رمز ورود خود را تغییر دهد ، این دستور زمانی مفید است که کاربر پس از انقضا وارد سیستم نمی شود .حتی پس از این دوره عدم فعالیت، در صورت عدم تغییر رمز ورود، حساب قفل شده و کاربر باید برای باز کردن قفل آن به مدیر مراجعه کند .برای تغییر اطلاعات انقضا ریشه ، من از کلمه کلیدی sudo استفاده می کنم .بعلاوه من از گزینه l- برای مشاهده دوره عدم فعالیت استفاده کردم.

```
sudo chage -I 5 root
```

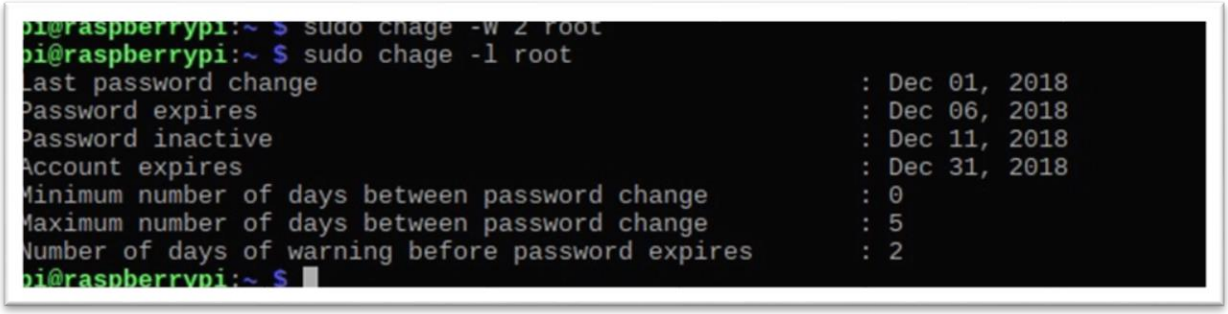
خروجی دستور:



گزینه W– :از این گزینه برای دادن اخطار قبلی قبل از انقضا رمز عبور استفاده کنید. ورودی داده شده در دستورتعداد روزهای قبل از تاریخ انقضا است که باید هشدار داده شود. برای تغییر اطلاعات منقضی شدن root من از کلمه کلیدی sudo استفاده میکنم. بعلاوه من از گزینه l- برای مشاهده دوره هشدار استفاده میکنم.

```
sudo chage -W 2 root
```

خروجی دستور:



دستور file

از دستور file برای تعیین نوع فایل استفاده می شود **.file** .نوع ممکن است برای انسان قابل خواندن باشد (مثلا متن (' ASCII و یا نوع MIME پرونده (به عنوان مثال ، متن / ساده؛ مجموعه کاراکتر = US-ASCII')). این دستور هر آرگومان را برای دسته بندی آن آزمایش می کند.

سه حالت تست دارد که به شرح زیر است:

filesystem test: این تست بر اساس نتیجه ای است که از فراخوان سیستم **stat** برمیگردد . این برنامه تأیید می کند که پرونده خالی است یا اینکه نوعی پرونده خاص است .این تست باعث چاپ نوع پرونده می شود.

Magic test: این تست ها برای بررسی پرونده های دارای داده در قالب های ثابت خاص استفاده می شوند.

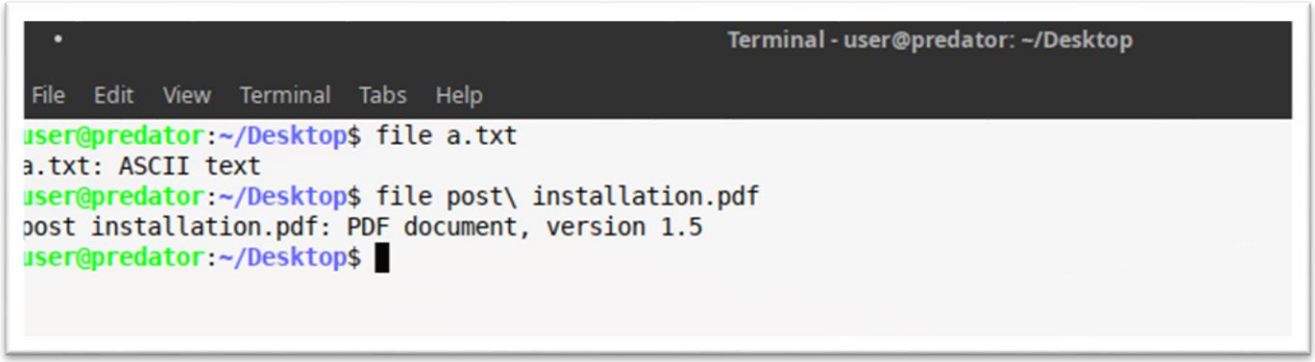
language test: این آزمون رشته های خاصی را جستجو می کند که می توانند در هر قسمت از چند بلوک اول پرونده ظاهر شوند.

ساختار دستور:

```
file [-bchikLNnprsvz0] [--apple] [--mime-encoding] [--mime-type] [-e testname] [-F separator]
[-f namefile] [-m magicfiles]
file ...
file -C [-m magicfiles]
```

مثال: نوع فایل را نمایش می دهد

```
file email.py
file name.jpeg
file Invoice.pdf
file exam.ods
file videosong.mp4
```



مثال های دستور:

گزینه b- یا brief--برای نمایش خلاصه نتیجه بدون توضیحات اضافی استفاده می‌شود. مثال:

```
$file -b myfile.txt
```

گزینه c- یا checking-printout--برای نمایش چک کردن پرینت استفاده می‌شود. مثال:

```
$file -c myfile.txt
```

گزینه h- یا no-dereference--برای عدم از بین بردن پیوندهای نمادین استفاده می‌شود. مثال:

```
$file -h myfile.txt
```

گزینه i- یا mime--برای نمایش نوع MIME فایل استفاده می‌شود. مثال:

```
$file -i myfile.txt
```

گزینه k- یا keep-going--برای ادامه تشخیص نوع فایل برای فایل‌های متعدد استفاده می‌شود. مثال:

```
$file -k myfile1.txt myfile2.txt
```

گزینه L- یا dereference--برای دنبال کردن پیوندهای نمادین استفاده می‌شود. مثال:

```
$file -L myfile.txt
```

گزینه N- یا no-pad--برای عدم پر کردن فاصله در خروجی استفاده می‌شود. مثال:

```
$file -N myfile.txt
```

گزینه n- یا no-buffer--برای عدم استفاده از بافر در خواندن فایل استفاده می‌شود. مثال:

```
$file -n myfile.txt
```

گزینه p- یا preserve-date--برای حفظ تاریخچه فایل استفاده می‌شود. مثال:

```
$file -p myfile.txt
```

گزینه r- یا raw--برای نمایش خروجی به صورت خام بدون تغییرات استفاده می‌شود. مثال:

```
$file -r myfile.txt
```

گزینه s- یا special-files--برای تشخیص نوع فایل‌های ویژه استفاده می‌شود. مثال:

```
$file -s myfile.txt
```

گزینه v- یا version--برای نمایش نسخه fileاستفاده می‌شود. مثال:

```
$file --version
```

گزینه z- یا uncompress--برای از حالت فشرده سازی خارج کردن فایل استفاده می‌شود. مثال:

```
$file -z myfile.txt.gz
```

گزینه 0-برای جدا کردن خروجی با صفر استفاده می‌شود. مثال:

```
$file -0 myfile.txt
```

گزینه apple--برای تشخیص نوع فایل برای سیستم عامل اپل استفاده می‌شود. مثال:

```
$file --apple myfile.txt
```

گزینه mime-encoding--برای نمایش رمزگذاری MIME فایل استفاده می‌شود. مثال:

```
$file --mime-encoding myfile.txt
```

گزینه mime-type--برای نمایش نوع MIME فایل استفاده می‌شود. مثال:

```
$file --mime-type myfile.txt
```

گزینه e testname-برای انجام تست مشخص بر روی فایل استفاده می‌شود. مثال:

```
$file -e elf myfile
```

گزینه F separator-برای استفاده از جداکننده مشخص بین خروجی‌ها استفاده می‌شود. مثال:

```
$file -F '|' myfile1.txt myfile2.txt
```

گزینه f namefile-برای استفاده از فایل مشخص برای خواندن نام فایل‌ها استفاده می‌شود. مثال:

```
$file -f filenames.txt
```

گزینه m magicfiles-برای استفاده از فایل‌های جادویی مشخص برای تشخیص نوع فایل استفاده می‌شود. مثال:

```
$file -m magic1.txt -m magic2.txt myfile.txt
```

ماژول های تأیید هویت

ماژول های تأیید هویت انطباق پذیر (PAM)انعطاف پذیری تنظیم یک طرح تأیید هویت خاص را بر اساس یک برنامه کاربردی و یا هر سرویس بر اساس ماژول ها ارائه می دهند.این ابزار موجود در تمام توزیع های مدرن linux مشکلی را که اغلب با توسعه دهندگان در روزهای اولیه linux مواجه شده است، برطرف می کند، در حالی که هر برنامه ای که احتیاج به احراز هویت داشت، باید به طور خاص برای بدست آوردن اطلاعات لازم بدست آورد. برای مثال ، باPAM ، مهم نیست که آیا گذر واژه در /etc/shadowذخیره می شود یا در یک سرور جداگانه در داخل شبکه به عنوان مثال ، هنگامی که برنامه ورود به system نیاز به تأیید اعتبار یک کاربردارد، PAM به صورت پویا آن را فراهم می کند که حاوی توابع برای طرح احراز هویت صحیح است.بنابراین، تغییر شیوه تأیید اعتبار برای login (و یا هر برنامه دیگر با استفاده از (PAM آسان است، زیرا فقط شامل ویرایش فایل پیکربندی است. اجازه دهید فایل پیکربندی PAM برای passwd را بررسی کنیم.، ابزار شناخته شده برای تغییر رمزهای عبور کاربر.این در /etc/pam.d/passwdقرار دارد:

```
# cat  / etc / pam.d/ passwd
```

```
:/# cat /etc/pam.d/system-auth  |less
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        sufficient    pam_fprintd.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 1000 quiet_success
auth        required      pam_deny.so
account     required      pam_unix.so
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 1000 quiet
account     required      pam_permit.so

password    requisite     pam_pwquality.so try_first_pass local_users_only retry=3 authtok_type=
password    sufficient    pam_unix.so sha512 shadow nullok try_first_pass use_authtok
password    required      pam_deny.so

session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
-session     optional      pam_systemd.so
session     [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session     required      pam_unix.so
(END)
```

پیکربندی فایل PAM برای linux رمز عبور

ستون اول نشان دهنده نوع اعتبارسنجی است که بایدبا module-path ستون سوم مورد استفاده قرار گیرد.وقتی یک خطا قبل از نوع نمایش داده می شود، PAMبه ورود به system وارد نمی شود .ماژول را نمی توان بارگذاری کرد، زیرا نمی تواند در system پیدا شود.

انواع احراز هویت زیر برای pam در دسترس هستند:

۱. Account: این نوع ماژول بررسی می کند که آیا کاربر یا سرویس اعتبار معتبر را تأیید کرده است.
۲. auth: این نوع ماژول تایید می کند چه کسی ادعا می کند و هر گونه امتیاز مورد نیاز را می دهد.
۳. Password: این نوع ماژول اجازه می دهد تا کاربر یا سرویس رمز عبور خود را به روز رسانی کنند.
۴. Session: این نوع ماژول نشان می دهد چه چیزی پس از احراز هویت موفق شود.

ستون دوم به نام control نشان می دهد چه اتفاقی می افتد اگر احراز هویت با این ماژول نتواند انجام شود:

۱. Requisite: اگر احراز هویت از طریق صورت نگیرد و این ماژول نتواند، احراز هویت کلی بلافاصله انکار خواهد شد.

۲. Required است شبیه به اجباربرای تایید هویت نامیده می شود.

۳. Sufficient اگر احراز هویت از طریق این ماژول نتواند انجام شود، PAMهمچنان احراز هویت را تأیید می کند، حتی اگر قبلا به عنوان مورد نیاز نتوانسته اید.

۴. Optional: اگر احراز هویت از طریق این ماژول نتواند یا موفق شود، هیچ اتفاقی نمی افتد، مگر اینکه این تنها واحد از نوع آن برای این سرویس باشد.

۵. Include این است که خطوط نوع داده شده، باید از یک فایل دیگر ضمیمه شود.

۶. Substack شبیه به include می باشد اما شکست های احراز هویت یا موفقیت ها باعث خروج از ماژول کامل نمی شود، بلکه فقط از زیر شاخه است.

سه خط اول در /etc/pam.d/passwd (در بالا نشان داده شده است)، ماژول system-auth را بارگذاری می کند تا بررسی کند که کاربر اعتبار معتبری را (حساب) ارائه کرده است.اگر چنین است، اجازه می دهد تا او را با اجازه دادن به استفاده از (passwd auth) تغییر رمز عبور را انجام دهد.

دستور setenforce

دستور setenforceبرای تغییر حالت SELinux در سیستم عامل لینوکس استفاده می‌شود SELinux .یک ماژول امنیتی هسته است که برای کنترل دسترسی و افزایش امنیت سیستم طراحی شده است.

Enforcing:این حالت پیش فرض SELinux است. در این حالت، SELinux تمام سیاست های امنیتی تعریف شده را به طور کامل اجرا می کند.

Permissive:در این حالت، SELinux به جای مسدود کردن عملیات غیرمجاز، آنها را به لاگ سیستم می‌نویسد.

Disabled:در این حالت، SELinux به طور کامل غیرفعال می شود.

ساختار دستور:

برای مشاهده حالت فعلی SELinux، از دستور getenforce استفاده کنید.

برای تغییر حالت SELinux، از دستور setenforce به همراه یکی از آرگومان های ۰، ۱ یا ۲ استفاده کنید.

گزینه های دستور:

- h- یا help-- :نمایش راهنمای دستور
- v- یا version-- :نمایش نسخه دستور

مثال های دستور:

مشاهده حالت فعلی SELinux:

```
getenforce
```

تغییر حالت SELinux به Permissive:

```
setenforce 0
```

تغییر حالت SELinux به Enforcing:

```
setenforce 1
```

غیرفعال کردن SELinux:

```
setenforce 2
```

مشاهده اطلاعات مربوط به حالت فعلی SELinux:

```
setenforce --help
```

تغییر حالت SELinux به Permissive و نمایش لاگ های مربوط به عملیات غیرمجاز:

```
setenforce 0 && tail -f /var/log/audit/audit.log
```

غیرفعال کردن SELinux و راه اندازی مجدد سیستم:

```
setenforce 2 && reboot
```

نکات مهم:

تغییر حالت SELinux می تواند بر عملکرد سیستم شما تأثیر بگذارد.

قبل از تغییر حالت SELinux، از اطلاعات سیستم خود نسخه پشتیبان تهیه کنید.

توصیه می شود SELinux را فقط در صورت نیاز و با احتیاط کامل غیرفعال کنید.

دستور chcon

دستور chcon برای تغییر برچسب‌های امنیتی SELinux (Security-Enhanced Linux) فایل‌ها و دایرکتوری‌ها استفاده می‌شود SELinux .یک ماژول امنیتی هسته‌ای است که برای کنترل دسترسی به منابع سیستم بر اساس برچسب‌های امنیتی و نقش‌های کاربران استفاده می‌شود.

موارد استفاده:

- ✓ تغییر برچسب امنیتی یک فایل یا دایرکتوری
- ✓ عیب‌یابی مشکلات مربوط به دسترسی به فایل
- ✓ تنظیمات امنیتی پیشرفته برای فایل‌ها و دایرکتوری‌ها

نحوه استفاده:

- ✓ بدون هیچ گونه آرگومانی، chcon اطلاعات مربوط به برچسب امنیتی فایل جاری را نمایش می‌دهد.
- ✓ برای تغییر برچسب امنیتی یک فایل یا دایرکتوری، نام آن را به عنوان آرگومان اول و برچسب امنیتی جدید را به عنوان آرگومان دوم به دستور بدهید.
- ✓ برای مشاهده لیست برچسب‌های امنیتی موجود، از گزینه l– استفاده کنید.
- ✓ برای تنظیمات پیشرفته، از گزینه‌های مختلف chcon استفاده کنید.

گزینه‌های دستور:

- l :نمایش لیست برچسب‌های امنیتی موجود
- R :تغییر برچسب امنیتی تمام فایل‌ها و دایرکتوری‌های یک دایرکتوری
- t :تنظیم برچسب امنیتی پیش‌فرض برای نوع فایل
- u :تنظیم مالک برچسب امنیتی
- r :تنظیم نقش برچسب امنیتی
- s :تنظیم سطح حساسیت برچسب امنیتی

نحوه استفاده:

بدون هیچ گونه آرگومانی، chcon اطلاعات مربوط به برچسب امنیتی فایل جاری را نمایش می‌دهد.

برای تغییر برچسب امنیتی یک فایل یا دایرکتوری، نام آن را به عنوان آرگومان اول و برچسب امنیتی جدید را به عنوان آرگومان دوم به دستور بدهید.

برای مشاهده لیست برچسب‌های امنیتی موجود، از گزینه l– استفاده کنید.

برای تنظیمات پیشرفته، از گزینه‌های مختلف chcon استفاده کنید.

مثال های دستور:

نمایش برچسب امنیتی فایل myfile.txt:

```
chcon myfile.txt
```

تغییر برچسب امنیتی فایل myfile.txt به user_u:object_r:myfile_t:s0:

```
chcon user_u:object_r:myfile_t:s0 myfile.txt
```

مشاهده لیست برچسب‌های امنیتی موجود:

```
chcon -l
```

تغییر برچسب امنیتی تمام فایل‌ها و دایرکتوری‌های دایرکتوری /home/user:

```
chcon -R user_u:object_r:home_dir_t:s0 /home/user
```

تنظیم برچسب امنیتی پیش‌فرض برای نوع فایل *.txt:

```
chcon -t user_u:object_r:text_t:s0 *.txt
```

تغییر برچسب امنیتی دایرکتوری /var/log به system_u:object_r:var_log_t:s0:

```
chcon -R system_u:object_r:var_log_t:s0 /var/log
```

تنظیم برچسب امنیتی پیش‌فرض برای نوع فایل *.sh:

```
chcon -t user_u:object_r:shell_script_t:s0 *.sh
```

تنظیم مالک برچسب امنیتی فایل myfile.txt به کاربر root:

```
chcon -u root myfile.txt
```

تنظیم نقش برچسب امنیتی فایل myfile.txt به sysadmin:

```
chcon -r sysadmin myfile.txt
```

تنظیم سطح حساسیت برچسب امنیتی فایل myfile.txt به topsecret:

```
chcon -s topsecret myfile.txt
```

فصل پانزدهم

فصل پانزدهم

مدیریت چندرسانه ای

دستور amixer

میکسر خط دستوربرای درایور کارت صدا یعنی ALSA (Advanced Linux Sound Architecture) است. می تواند از چندین کارت صدا پشتیبانی کند . بدون هیچ آرگومان تنظیمات فعلی میکسر را برای کارت صدا پیش فرض و همچنین دستگاه نمایش می دهد .این روش خوبی برای دیدن لیستی از کنترل های ساده میکسر است که می توانید استفاده کنید.

موارد استفاده:

- ✓ تنظیم سطح صدا
- ✓ تنظیم تعادل صدا بین کانال های مختلف
- ✓ فعال یا غیرفعال کردن میوت
- ✓ تنظیم تن صدا و بیس
- ✓ انتخاب منبع ورودی صدا
- ✓ انتخاب خروجی صدا
- ✓ مشاهده اطلاعات مربوط به کارت صدا

نحوه استفاده:

- ✓ بدون هیچ گونه آرگومانی، amixer اطلاعات مربوط به تمام کنترل های موجود را نمایش می‌دهد.
- ✓ برای مشاهده اطلاعات مربوط به یک کنترل خاص، نام آن را به عنوان آرگومان به دستور بدهید.
- ✓ برای تنظیم مقدار یک کنترل، از گزینه –s و نام کنترل و مقدار جدید استفاده کنید.
- ✓ برای افزایش یا کاهش مقدار یک کنترل، از گزینه های –i و –d استفاده کنید.

ساختار دستور:

```
amixer [-option] [دستور]
amixer [-option] [command]
```

گزینه های دستور:

- c – :انتخاب کارت صدا
- D – :نمایش اطلاعات به صورت دیباگ
- h – :نمایش راهنمای دستور
- i – :افزایش مقدار یک کنترل
- m – :میوت کردن یک کنترل
- s – :تنظیم مقدار یک کنترل
- u unmute – :کردن یک کنترل

info– :اطلاعات را نمایش می دهد و سپس خارج می شود.

```
nishanth@nishanth-VirtualBox:~$ amixer info
Card default 'I82801AAICH'/'Intel 82801AA-ICH with AD1980 at irq 21'
  Mixer name    : 'Analog Devices AD1980'
  Components    : 'AC97a:41445370'
  Controls      : 46
  Simple ctrls  : 32
nishanth@nishanth-VirtualBox:~$
```

scontols– :لیست کاملی از کنترل های ساده را نشان می دهد و سپس خارج می شود.

```
Simple mixer control 'Master',0
Simple mixer control 'Master Mono',0
Simple mixer control 'Headphone Jack Sense',0
Simple mixer control '3D Control - Switch',0
Simple mixer control 'PCM',0
Simple mixer control 'PCM Out Path & Mute',0
Simple mixer control 'Surround',0
Simple mixer control 'Surround Jack Mode',0
Simple mixer control 'Center',0
Simple mixer control 'LFE',0
Simple mixer control 'Line',0
Simple mixer control 'Line Jack Sense',0
Simple mixer control 'CD',0
Simple mixer control 'Mic',0
Simple mixer control 'Mic Boost (+20dB)',0
Simple mixer control 'Mic Select',0
Simple mixer control 'Video',0
Simple mixer control 'Phone',0
Simple mixer control 'Beep',0
Simple mixer control 'Aux',0
Simple mixer control 'Mono Output Select',0
```

scontents– :لیست کاملی از کنترل های ساده میکسر و همچنین محتوای آنها را نشان می دهد.

```
Simple mixer control 'Master',0
  Capabilities: pvolume pswitch
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 31
  Mono:
  Front Left: Playback 17 [55%] [-21.00dB] [on]
  Front Right: Playback 17 [55%] [-21.00dB] [on]
Simple mixer control 'Master Mono',0
  Capabilities: pvolume pswitch
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 63
  Mono:
  Front Left: Playback 0 [0%] [-94.50dB] [off]
  Front Right: Playback 0 [0%] [-94.50dB] [off]
Simple mixer control 'Headphone Jack Sense',0
  Capabilities: pswitch pswitch-joined
  Playback channels: Mono
  Mono: Playback [off]
Simple mixer control '3D Control - Switch',0
  Capabilities: pswitch pswitch-joined
  Playback channels: Mono
  Mono: Playback [off]
Simple mixer control 'PCM',0
  Capabilities: pvolume pswitch
```

set or sset :محتویات مخلوط کن ساده را تنظیم می کند.

```
Simple mixer control 'Master',0
  Capabilities: pvolume pswitch
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 31
  Mono:
  Front Left: Playback 4 [13%] [-40.50dB] [on]
  Front Right: Playback 4 [13%] [-40.50dB] [on]
nishanth@nishanth-VirtualBox:~$
```

get or sget: محتوای کنترلی ساده را نشان می دهد.

```
nishanth@nishanth-VirtualBox:~$ amixer get Master
Simple mixer control 'Master',0
  Capabilities: pvolume pswitch
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 31
  Mono:
  Front Left: Playback 4 [13%] [-40.50dB] [on]
  Front Right: Playback 4 [13%] [-40.50dB] [on]
```

controls: لیست کاملی از کنترل های کارت را نمایش می دهد.

```
nishanth@nishanth-VirtualBox:~$ amixer controls
numid=3,iface=MIXER,name='Master Mono Playback Switch'
numid=4,iface=MIXER,name='Master Mono Playback Volume'
numid=1,iface=MIXER,name='Master Playback Switch'
numid=2,iface=MIXER,name='Master Playback Volume'
numid=42,iface=MIXER,name='Headphone Jack Sense'
numid=32,iface=MIXER,name='3D Control - Switch'
numid=31,iface=MIXER,name='PCM Out Path & Mute'
numid=20,iface=MIXER,name='PCM Playback Switch'
numid=21,iface=MIXER,name='PCM Playback Volume'
numid=40,iface=MIXER,name='Surround Jack Mode'
numid=22,iface=MIXER,name='Surround Playback Switch'
numid=23,iface=MIXER,name='Surround Playback Volume'
```

content: لیست کاملی از کنترل های کارت را با محتوای آنها نیز نمایش می دهد.

```
numid=3,iface=MIXER,name='Master Mono Playback Switch'
  ; type=BOOLEAN,access=rw------,values=2
  : values=off,off
numid=4,iface=MIXER,name='Master Mono Playback Volume'
  ; type=INTEGER,access=rw---R--,values=2,min=0,max=63,step=0
  : values=0,0
  | dBscale-min=-94.50dB,step=1.50dB,mute=0
numid=1,iface=MIXER,name='Master Playback Switch'
  ; type=BOOLEAN,access=rw------,values=2
  : values=on,on
numid=2,iface=MIXER,name='Master Playback Volume'
  ; type=INTEGER,access=rw---R--,values=2,min=0,max=31,step=0
  : values=4,4
  | dBscale-min=-46.50dB,step=1.50dB,mute=0
numid=42,iface=MIXER,name='Headphone Jack Sense'
  ; type=BOOLEAN,access=rw------,values=1
  : values=off
numid=32,iface=MIXER,name='3D Control - Switch'
```

cset: برای تنظیم کنترلهای کارت استفاده می شود.

```
nishanth@nishanth-VirtualBox:~$ amixer cset name='Line Playback Volume' 66,77
numid=13,iface=MIXER,name='Line Playback Volume'
  ; type=INTEGER,access=rw---R--,values=2,min=0,max=63,step=0
  : values=63,63
  | dBscale-min=-34.50dB,step=1.50dB,mute=0
```

cget: این محتویات کنترل کارت را نشان می دهد. شناسه دارای همان نحو دستور cset است.

```
nishanth@nishanth-VirtualBox:~$ amixer cget name='Line Playback Volume'
numid=13,iface=MIXER,name='Line Playback Volume'
  ; type=INTEGER,access=rw---R--,values=2,min=0,max=63,step=0
  : values=63,63
  | dBscale-min=-34.50dB,step=1.50dB,mute=0
```

-c: شماره کارتی را که باید کنترل شود انتخاب می کند.

```
nishanth@nishanth-VirtualBox:~$ amixer -c I82801AAICH
Simple mixer control 'Master',0
  Capabilities: pvolume pswitch
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 31
  Mono:
  Front Left: Playback 4 [13%] [-40.50dB] [on]
  Front Right: Playback 4 [13%] [-40.50dB] [on]
Simple mixer control 'Master Mono',0
  Capabilities: pvolume pswitch
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 63
  Mono:
  Front Left: Playback 0 [0%] [-94.50dB] [off]
  Front Right: Playback 0 [0%] [-94.50dB] [off]
Simple mixer control 'Headphone Jack Sense',0
  Capabilities: pswitch pswitch-joined
  Playback channels: Mono
  Mono: Playback [off]
Simple mixer control '3D Control - Switch',0
```

-D: نام دستگاهی را که باید کنترل شود انتخاب می کند.

```
nishanth@nishanth-VirtualBox:~$ amixer -D default
Simple mixer control 'Master',0
  Capabilities: pvolume pswitch
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 31
  Mono:
  Front Left: Playback 4 [13%] [-40.50dB] [on]
  Front Right: Playback 4 [13%] [-40.50dB] [on]
Simple mixer control 'Master Mono',0
  Capabilities: pvolume pswitch
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 63
  Mono:
  Front Left: Playback 0 [0%] [-94.50dB] [off]
  Front Right: Playback 0 [0%] [-94.50dB] [off]
Simple mixer control 'Headphone Jack Sense',0
  Capabilities: pswitch pswitch-joined
  Playback channels: Mono
  Mono: Playback [off]
Simple mixer control '3D Control - Switch',0
  Capabilities: pswitch pswitch-joined
  Playback channels: Mono
```

-s یا stdin – **:** از stdin بخوانید و هر دستور را به ترتیب اجرا کنید.

```
nishanth@nishanth-VirtualBox:~$ amixer -s
^Z
[4]+  Stopped                 amixer -s
```

-q **:** نتیجه تغییرات را نشان نمی دهد.

```
nishanth@nishanth-VirtualBox:~$ amixer -q
Simple mixer control 'Master',0
  Capabilities: pvolume pswitch
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 31
  Mono:
  Front Left: Playback 4 [13%] [-40.50dB] [on]
  Front Right: Playback 4 [13%] [-40.50dB] [on]
Simple mixer control 'Master Mono',0
  Capabilities: pvolume pswitch
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 63
  Mono:
  Front Left: Playback 0 [0%] [-94.50dB] [off]
  Front Right: Playback 0 [0%] [-94.50dB] [off]
Simple mixer control 'Headphone Jack Sense',0
  Capabilities: pswitch pswitch-joined
  Playback channels: Mono
  Mono: Playback [off]
```

-R **:** از مقدار خام برای ارزیابی نمایندگی درصد که حالت پیش فرض نیز است استفاده می کند.

```
nishanth@nishanth-VirtualBox:~$ amixer -R
Simple mixer control 'Master',0
  Capabilities: pvolume pswitch
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 31
  Mono:
  Front Left: Playback 4 [13%] [-40.50dB] [on]
  Front Right: Playback 4 [13%] [-40.50dB] [on]
Simple mixer control 'Master Mono',0
  Capabilities: pvolume pswitch
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 63
  Mono:
  Front Left: Playback 0 [0%] [-94.50dB] [off]
  Front Right: Playback 0 [0%] [-94.50dB] [off]
Simple mixer control 'Headphone Jack Sense',0
  Capabilities: pswitch pswitch-joined
  Playback channels: Mono
  Mono: Playback [off]
Simple mixer control '3D Control - Switch',0
  Capabilities: pswitch pswitch-joined
  Playback channels: Mono
```

M- :از حجم نگاشت شده برای ارزیابی درصد نمایش مانند **alsamixer** استفاده می کند تا طبیعی تر برای گوش انسان باشد.

```
nishanth@nishanth-VirtualBox:~$ amixer -M
Simple mixer control 'Master',0
  Capabilities: pvolume pswitch
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 31
  Mono:
  Front Left: Playback 4 [5%] [-40.50dB] [on]
  Front Right: Playback 4 [5%] [-40.50dB] [on]
Simple mixer control 'Master Mono',0
  Capabilities: pvolume pswitch
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 63
  Mono:
  Front Left: Playback 0 [0%] [-94.50dB] [off]
  Front Right: Playback 0 [0%] [-94.50dB] [off]
Simple mixer control 'Headphone Jack Sense',0
  Capabilities: pswitch pswitch-joined
```

v- :نسخه را نمایش می دهد.

```
nishanth@nishanth-VirtualBox:~$ amixer -v
amixer version 1.1.3
```

دستور aplay

aplayیک پخش کننده صوتی خط فرمان برای درایورهای کارت صدا:

ALSA (Advanced Linux Sound Architecture) است .از چندین قالب فایل و چندین کارت صدا با چندین دستگاه پشتیبانی می کند .این اساساً برای پخش صدا روی رابط خط فرمان استفاده می شود . تقریباً همان آرکورد است که فقط به جای ضبط پخش می شود .برای قالب های صوتی پشتیبانی شده ، می توان بطور خودکار از سرآیند فایل صوتی میزان نمونه برداری ، عمق بیت و غیره را تعیین کردaplay . یک پخش کننده صوتی خط دستور برای درایورهای کارت صدای ALSA (Advanced Linux Sound Architecture) است. از چندین فرمت فایل و چندین کارت صدا با چندین دستگاه پشتیبانی می‌کند. اساساً برای پخش صدا در رابط خط دستور استفاده می شود. aplay تقریباً مشابه arecord است که به جای ضبط پخش می شود. برای قالب‌های فایل صوتی پشتیبانی‌شده، نرخ نمونه‌برداری، عمق بیت و غیره را می‌توان به‌طور خودکار از سربرگ فایل صوتی تعیین کرد.

ساختار دستور:

```
$ aplay [flags] [filename [filename]] ...
```

توجه: این دستور شامل گزینه های دستور:ی مختلف دیگری است که معمولاً به آنها نیاز نداریم. اگر می خواهید بیشتر بدانید می توانید به سادگی دستور زیر را در ترمینال خود اجرا کنید.

```
$  aplay --help
```

گزینه های دستور:

- h نمایش راهنمای دستور
- l لیست کردن دستگاه های صوتی موجود
- D انتخاب دستگاه صوتی مورد نظر برای پخش
- f تنظیم فرمت صوتی مورد نظر
- r تنظیم نرخ نمونه برداری
- c تنظیم تعداد کانال های صوتی
- t تنظیم طول بافر
- q تنظیم سطح صدا
- v تنظیم (verbosityسطح جزئیات)
- n تعداد دفعات پخش فایل صوتی
- i پخش از ورودی استاندارد
- s سکوت قبل از پخش
- p پیش نمایش چند ثانیه اول از فایل صوتی
- g تنظیم گین (gain)
- m تنظیم حالت پخش (mono، stereo، ...)
- o تنظیم (offsetجابجایی) در فایل صوتی

مثال های دستور:

برای پخش صدا فقط به مدت ۱۰ ثانیه با فرکانس ۲۵۰۰ هرتز.

```
$ aplay -d 10 -r 2500hz sample.mp3
```

فایل sample.mp3 را تنها به مدت ۱۰ ثانیه با فرکانس ۲۵۰۰ هرتز پخش می‌کند.

برای پخش کامل کلیپ صوتی با فرکانس ۲۵۰۰ هرتز.

```
$ aplay -r 2500hz sample.mp3
```

فایل sample.mp3 را با فرکانس ۲۵۰۰ هرتز پخش می‌کند.

برای نمایش اطلاعات نسخه

```
$ aplay --version
```

اطلاعات نسخه را نمایش می‌دهد. برای من aplay را نشان می‌دهد: vesrion 1.1.0

پخش یک فایل صوتی:

```
aplay my_file.wav
```

لیست کردن دستگاه های صوتی موجود:

```
aplay -l
```

انتخاب دستگاه صوتی “ ”hw:1,0برای پخش:

```
aplay -D hw:1,0 my_file.wav
```

تنظیم فرمت صوتی به "wav":

```
aplay -f wav my_file.wav
```

تنظیم نرخ نمونه برداری به ۴۴۱۰۰ هرتز:

```
aplay -r 44100 my_file.wav
```

تنظیم تعداد کانال های صوتی به ۲:

```
aplay -c 2 my_file.wav
```

تنظیم طول بافر به ۱۰۲۴:

```
aplay -t 1024 my_file.wav
```

تنظیم سطح صدا به ٪۵۰:

```
aplay -q 50 my_file.wav
```

پخش ۵ بار فایل صوتی:

```
aplay -n 5 my_file.wav
```

پخش از ورودی استاندارد:

```
aplay < <(cat my_file.wav)
```

سکوت ۵ ثانیه قبل از پخش:

```
aplay -s 5 my_file.wav
```

پیش نمایش ۵ ثانیه اول از فایل صوتی:

```
aplay -p 5 my_file.wav
```

تنظیم گین به ۱,۵:

```
aplay -g 1.5 my_file.wav
```

تنظیم حالت پخش به مونو:

```
aplay -m mono my_file.wav
```

تنظیم offset به ۱۰ ثانیه در فایل صوتی:

```
aplay -o 10 my_file.wav
```

دستور aplaymidi

از دستور aplaymidi در لینوکس برای پخش پرونده های استاندارد (MIDI رابط دیجیتال ابزار موسیقی) ، با ارسال محتوای یک فایل MIDIبه یک پورت (MIDI ALSA (Advanced Sound Architecture ، ارائه دهنده صدا مانند ترسو یا یک سخت افزار MIDIمورد نیاز است. پخش پرونده های MIDI.

ساختار دستور:

```
aplaymidi [گزینه ها] [MIDIfile]
```

گزینه های دستور:

- **h-یاhelp–** :این برای نشان دادن همه گزینه ها و نحو دستور استفاده می شود.

```
$ aplaymidi -h
```

```
ubuntu@ubuntu-VirtualBox:~$ aplaymidi -h
Usage: aplaymidi -p client:port[,...] [-d delay] midifile ...
-h, --help                  this help
-V, --version               print current version
-l, --list                  list all possible output ports
-p, --port=client:port,...  set port(s) to play to
-d, --delay=seconds         delay after song ends
ubuntu@ubuntu-VirtualBox:~$
```

- **-V or –version**: برای نمایش نسخه فعلی SW استفاده می شود.

```
$ aplaymidi -V
```

```
ubuntu@ubuntu-VirtualBox:/$ aplaymidi -V
aplaymidi version 1.0.25
```

- **-l یا –list**: برای نمایش لیستی از همه پورت های خروجی MIDI ممکن استفاده می شود.

```
$ aplaymidi -l
```

```
ubuntu@ubuntu-VirtualBox:/$ aplaymidi -l
 Port    Client name                      Port name
 14:0    Midi Through                     Midi Through Port-0
```

- **-p or –port = client: port**: پورت (های) مشخصی را برای پخش MIDI در آنها مشخص می کند.

```
aplaymidi $ -p 14: 0 test1.mid
```

```
ubuntu@ubuntu-VirtualBox:~/Downloads$ aplaymidi -p 14:0 test1.mid
ubuntu@ubuntu-VirtualBox:~/Downloads$ ls
test1.mid  testmidi.mid
```

- **-d یا –delay = seconds**: با پایان **یافتن** صدای MIDI به تأخیر می افتد.

```
aplaymidi $ -p 14: 0 -d 10 test1.mid
```

```
ubuntu@ubuntu-VirtualBox:~/Downloads$ aplaymidi -p 14:0 -d 10 test1.mid
ubuntu@ubuntu-VirtualBox:~/Downloads$
```

دستور ffmpeg

ffmpeg یک مبدل ویدیویی FFmpeg است. ffmpeg یک ابزار قدرتمند و انعطاف‌پذیر برای پردازش و تبدیل فایل‌های صوتی و تصویری در سیستم‌عامل‌های مختلف از جمله لینوکس است. این ابزار به شما امکان می‌دهد تا وظایف مختلفی مانند برش، ادغام، تبدیل فرمت، رمزگذاری، و استخراج صدا را انجام دهید.

ساختار دستور:

```
$ ffmpeg [[infile options][-i infile]]... {[outfile options] outfile}...
```

گزینه های دستور:

- ✓ i ورودی فایل. می‌توانید از این گزینه برای مشخص کردن فایل صوتی یا تصویری که می‌خواهید پردازش کنید استفاده کنید.
- ✓ خروجی فایل. این گزینه نام فایل خروجی را مشخص می‌کند.
- ✓ c کدک. این گزینه به شما امکان می‌دهد تا کدک مورد نظر خود را برای رمزگذاری و فشرده‌سازی فایل خروجی انتخاب کنید.
- ✓ b بیت‌ریت. این گزینه به شما امکان می‌دهد تا بیت‌ریت فایل خروجی را تنظیم کنید.
- ✓ s اندازه. این گزینه به شما امکان می‌دهد تا اندازه فریم ویدیو را تنظیم کنید.

- ✓ t زمان. این گزینه به شما امکان می‌دهد تا طول برش ویدیو را مشخص کنید.
- ✓ vf فیلترهای ویدیو. این گزینه به شما امکان می‌دهد تا فیلترهای مختلفی را برای اعمال به ویدیو مانند چرخش، برش، و تنظیم رنگ استفاده کنید.
- ✓ af فیلترهای صوتی. این گزینه به شما امکان می‌دهد تا فیلترهای مختلفی را برای اعمال به صدا مانند تنظیم ولوم، حذف نویز، و تنظیم اکولایزر استفاده کنید.
- ✓ به عنوان یک قاعده کلی، گزینه های دستور: برای فایل مشخص شده بعدی اعمال می شوند. بنابراین، ترتیب مهم است و می توانید چندین بار همان گزینه را در خط فرمان داشته باشید. سپس هر رخداد به فایل ورودی یا خروجی بعدی اعمال می شود.

❋ برای تنظیم میزان بیت ویدیوی فایل خروجی روی ۶۴ کیلوبیت بر ثانیه:

```
$ ffmpeg -i input.avi -b 64k output.avi
```

❋ برای وادار کردن نرخ فریم فایل خروجی به ۲۴ فریم در ثانیه:

```
$ ffmpeg -i input.avi -r 24 output.avi
```

❋ برای وادار کردن نرخ فریم فایل ورودی (معتبر برای فرمت های خام) به ۱ فریم در ثانیه و نرخ فریم فایل خروجی به ۲۴ فریم در ثانیه:

```
$ ffmpeg -r 1 -i input.m2v -r 24 output.avi
```

گزینه فرمت ممکن است برای فایل های ورودی خام مورد نیاز باشد. به‌طور پیش‌فرض، FFmpeg سعی می‌کند تا حد امکان بدون تلفات تبدیل کند: از همان پارامترهای صوتی و تصویری برای خروجی‌ها استفاده می‌کند که برای ورودی‌ها مشخص شده است.

تبدیل فرمت ویدیو:

```
ffmpeg -i input.mp4 -c:v libx264 -c:a aac -b:v 1000k -b:a 128k output.mp4
```

برش ویدیو:

```
ffmpeg -i input.mp4 -ss 00:00:10 -t 00:00:30 output.mp4
```

اضافه کردن واترمارک به ویدیو:

```
ffmpeg -i input.mp4 -vf "drawtext=text='My Watermark':fontfile=/path/to/font.ttf:fontsize=24:color=white:x=10:y=10" output.mp4
```

استخراج صدا از ویدیو:

```
ffmpeg -i input.mp4 -c:a copy output.mp3
```

ادغام دو فایل صوتی:

```
ffmpeg -i input1.mp3 -i input2.mp3 -filter_complex amerge -ac 2 output.mp3
```

دستور spd- say

spd-say درخواست خروجی متن به گفتار را به فرآیند پخش کننده گفتار می فرستد که آن را مدیریت می‌کند و در حالت ایده آل نتیجه را به سیستم صوتی ارسال می‌کند.

ساختاردستور:

```
$ spd-say [options] "some text"
```

گزینه های دستور:

r – ، rate- نرخ گفتار را تنظیم میکند (بین −۱۰۰ و +۱۰۰، پیش‌فرض : ۰)

p – ، pitch-- تنظیم زیر و بم صدا (بین −۱۰۰ و +۱۰۰، پیش فرض : ۰)

i – ، volume-- تنظیم حجم (شدت) صدا (بین −۱۰۰ و +۱۰۰، پیش‌فرض : ۰)

o ، – --output-module

ماژول خروجی را تنظیم میکند

- l – تنظیم زبان (کد iso)
- t – ، voice-type-- نوع صدای ترجیحی را تنظیم میکند
(male1، male2، male3، women1، women2، women3، child_male، child_female)
- m – حالت نقطه گذاری
- حالت نقطه گذاری را تنظیم میکند (هیچ، بعضی، همه)
- s – غلط گیری و بررسی املا
- P – ، priority-- اولویت پیام را تنظیم میکند (مهم، پیام، متن، اعلان، پیشرفت؛ پیش‌فرض : متن) –N، –نام برنامه
- نام برنامه مورد استفاده برای برقراری ارتباط را با مقدار رشته مشخص شده تنظیم میکند
- n – connection-name-- نام اتصال مورد استفاده برای برقراری اتصال را با مقدار رشته مشخص شده تنظیم میکند
- w – منتظر میماند .صبر میکند تا پیام گفته شود یا نادیده گرفته شود
- S – توقف
- C- لغو همه پیام‌ها در Dispatcher گفتار
- v– نمایش نسخه
- h – این اطلاعات راهنما را چاپ میکند

مثال ها:

۱. برای پخش متن داده شده به عنوان صدا.

```
$ spd-say “Hello”
```

“Hello” را به صورت صدا پخش می‌کند.

گزینه l-یا language--برای تنظیم زبان) کد (iso استفاده می‌شود. مثال:

```
$spd-say -l en “Hello, how are you?”
```

گزینه t-یا voice-type--برای تنظیم نوع صدای ترجیحی استفاده می‌شود. مثال:

```
$spd-say -t male2 “This is a sample text.”
```

گزینه m-برای تنظیم حالت نقطه‌گذاری استفاده می‌شود. مثال:

```
$spd-say -m some “This is a sample text.”
```

گزینه s-یا spell--برای غلط‌گیری و بررسی املا استفاده می‌شود. مثال:

```
$spd-say -s “Hello, how are you?”
```

گزینه P-یا priority--برای تنظیم اولویت پیام استفاده می‌شود. مثال:

```
$spd-say -P alert “This is an important message.”
```

گزینه N-یا program-name--برای تنظیم نام برنامه استفاده می‌شود. مثال:

```
$spd-say -N “MyApp” “This is a sample text.”
```

گزینه n-یا connection-name--برای تنظیم نام اتصال استفاده می‌شود. مثال:

```
$spd-say -n “MyConnection” “This is a sample text.”
```

گزینه w-برای صبر تا پیام گفته شود یا نادیده گرفته شود استفاده می‌شود. مثال:

```
$spd-say -w “Please wait for the message to be spoken.”
```

گزینه S-یا stop--برای توقف گفتار استفاده می‌شود. مثال:

```
$spd-say -S
```

گزینه C-یا flush--برای لغو همه پیام‌ها در Dispatcher گفتار استفاده می‌شود. مثال:

```
$spd-say -C
```

گزینه v-یا version--برای نمایش نسخه استفاده می‌شود. مثال:

```
$spd-say -v
```

گزینه h-یا help--برای نمایش اطلاعات راهنما استفاده می‌شود. مثال:

```
$spd-say -h
```

گفتن جمله "سلام دنیا" با نرخ گفتار دو برابر:

```
spd-say -r 100 "سلام دنیا"
```

گفتن جمله "امروز شنبه است" با نرخ گفتار نصف:

```
spd-say -r -50 "امروز شنبه است"
```

تنظیم زیر و بم صدا:

گفتن جمله "من یک ربات هستم" با صدای بم:

```
spd-say -p -50 "من یک ربات هستم"
```

گفتن جمله "به خانه خوش آمدید" با صدای زیر:

```
spd-say -p 50 "به خانه خوش آمدید"
```

تنظیم حجم صدا:

گفتن جمله "ساعت ۱۰ صبح است" با صدای بلند:

```
spd-say -i 100 "ساعت ۱۰ صبح است"
```

گفتن جمله "هوا سرد است" با صدای آهسته:

```
spd-say -i -50 "هوا سرد است"
```

تنظیم زبان:

گفتن جمله "من یک ربات هستم" به زبان انگلیسی:

```
spd-say -l en "I am a robot"
```

گفتن جمله " "你好به زبان چینی:

```
spd-say -l zh "你好"
```

تنظیم نوع صدا:

گفتن جمله "من یک ربات هستم" با صدای زن:

```
spd-say -t women1 "من یک ربات هستم"
```

گفتن جمله "به خانه خوش آمدید" با صدای مرد:

```
spd-say -t male2 به خانه خوش آمدید"
```

تنظیم حالت نقطه گذاری:

گفتن جمله "امروز، شنبه است" بدون نقطه گذاری:

```
spd-say -m none امروز، شنبه است"
```

گفتن جمله "ساعت ۱۰. صبح است" با نقطه گذاری کامل:

```
spd-say -m all ساعت ۱۰. صبح است"
```

غلط گیری و بررسی املا:

گفتن جمله "من یک ربات هستم" با بررسی املا:

```
spd-say -s من یک ربات هستم"
```

تنظیم اولویت پیام:

گفتن جمله "هشدار: خطر قریب الوقوع" با اولویت بالا:

```
spd-say -P important هشدار: خطر قریب الوقوع"
```

تنظیم نام برنامه:

گفتن جمله "این یک پیام از برنامه 'My Program' است":

```
spd-say -N "My Program"
```

تنظیم نام اتصال:

گفتن جمله "این یک پیام از اتصال 'My Connection' است":

```
spd-say -n "My Connection"
```

انتظار تا پایان گفتار:

گفتن جمله "لطفا منتظر بمانید" و صبر تا پایان گفتار:

```
spd-say -w لطفا منتظر بمانید"
```

لغو همه پیام ها:

```
spd-say -C
```

نمایش نسخه:

```
spd-say -v
```

نمایش اطلاعات راهنما:

```
spd-say -h
```

دستور cdrecord

دستور cdrecord در سیستم‌عامل‌های Unix/Linux برای رایت و کپی دیسک‌های CD یا DVD استفاده می‌شود. با استفاده از این دستور، می‌توانید اطلاعات را بر روی دیسک‌های نوری (optical discs) نوشته و آن‌ها را خوانده یا ایجاد کنید. همچنین دستور cdrecord برای نوشتن داده ها بر روی سی دی های صوتی و تصویری در سیستم عامل لینوکس استفاده می‌شود. این ابزار می‌تواند برای ضبط موسیقی، فیلم، و یا هر نوع داده دیگری بر روی سی دی استفاده شود.

گزینه های دستور:

- dev نام دستگاه سی دی رام را مشخص می‌کند.
- speed سرعت نوشتن را تنظیم می‌کند.
- buffer اندازه بافر را تنظیم می‌کند.
- data نوع داده‌ای که می‌خواهید بر روی سی دی بنویسید را مشخص می‌کند.
- audio برای نوشتن سی دی صوتی استفاده می‌شود.
- tao برای نوشتن سی دی تصویری با فرمت TAO استفاده می‌شود.
- sao برای نوشتن سی دی تصویری با فرمت SAO استفاده می‌شود.
- track برای نوشتن ترک های صوتی جداگانه استفاده می‌شود.
- pregap فاصله زمانی بین ترک های صوتی را تنظیم می‌کند.
- blank برای پاک کردن سی دی قابل بازنویسی استفاده می‌شود.
- eject بعد از اتمام فرآیند نوشتن، سی دی را از دستگاه خارج می‌کند.
- h راهنمای دستور را نمایش می‌دهد.

مثال های دستور:

نوشتن یک سی دی صوتی با سرعت x۴:

```
cdrecord -v -speed=4x dev=sr0 data=audio
```

نوشتن یک سی دی تصویری با فرمت TAO با سرعت x۸:

```
cdrecord -v -speed=8x dev=sr0 data=tao
```

نوشتن یک سی دی تصویری با فرمت SAO با سرعت x۱۲:

```
cdrecord -v -speed=12x dev=sr0 data=sao
```

نوشتن ترک اول یک آلبوم صوتی با فاصله ۲ ثانیه بین ترک ها:

```
cdrecord -v dev=sr0 track=1 pregap=2 data=audio
```

پاک کردن یک سی دی قابل بازنویسی:

```
cdrecord -v dev=sr0 blank=fast
```

نوشتن داده ها بر روی سی دی با نام "My CD":

```
cdrecord -v dev=sr0 data=my_data.iso
```

خارج کردن سی دی از دستگاه بعد از اتمام فرآیند نوشتن:

```
cdrecord -v dev=sr0 eject
```

رایت فایل ISO بر روی دیسک CD: این دستور برای رایت یک فایل ISO بر روی یک دیسک CD استفاده می‌شود.

```
cdrecord -v dev=/dev/cdrom mydata.iso
```

نمایش اطلاعات جزئی در حین رایت: این دستور برای نمایش اطلاعات جزئی در حین رایت فایل بر روی دیسک استفاده می‌شود.

```
cdrecord -v -s dev=/dev/cdrom mydata.iso
```

رایت فایل با سرعت خاص: این دستور برای رایت یک فایل بر روی دیسک با سرعت خاص مشخص شده استفاده می‌شود.

```
cdrecord -v speed=8 dev=/dev/cdrom mydata.iso
```

رایت فایل و پاکسازی دیسک قبل از رایت: این دستور برای رایت یک فایل بر روی دیسک و پاکسازی دیسک قبل از رایت استفاده می‌شود.

```
cdrecord -v blank=fast dev=/dev/cdrom mydata.iso
```

رایت فایل بر روی دیسک DVD: این دستور برای رایت یک فایل بر روی یک دیسک DVD استفاده می‌شود.

```
cdrecord -v dev=/dev/dvd mydata.iso
```

دستور رایت CD/DVD

ابزارها و روش های دستوری زیادی برای رایت CD و DVD در لینوکس وجود دارد. در ادامه به معرفی برنامه های گرافیکی و ترمینالی برای این امر میپردازیم:

رابط کاربری گرافیکی برای رایت سی دی:

```
X-CdRoast
 Gnome Nautilus
GNOME Brasero
 GNOME Toaster
Disc-O-Matic ۱
ECLipt Roaster
K3
SSCDRFE - Qt
TkBurn
```

خط فرمان برای رایت سی دی:

```
cdrecord
 mkisofs
mkisofs-md5
cdparanoia
 cdrdao
```

دستور brasero

برنامه ساده و آسان برای رایت CD/DVD برای دسکتاپ Gnome. البته در سایر میزهای دیگه هم قابل نصب و اجرا است. همچنین Braseroیک ابزار قدرتمند برای رایت دیسک های نوری (CD/DVD) در سیستم عامل لینوکس است. این برنامه به شما امکان می دهد تا انواع مختلفی از داده ها را بر روی دیسک های نوری رایت کنید، مانند فایل های صوتی، تصویری، ویدیویی، و داده های کامپیوتری.

ساختاردستور:

```
$ brasero [option] [uri] [URI]
```

rasero یک برنامه رایت دیسک است. طراحی آن ساده و استفاده از آن آسان است. این امکان ایجاد CD/DVD داده، CD صوتی، کپی CD/DVD و رایت تصاویر را فراهم می‌کند. دارای برخی از ویژگی های خوب، مانند پیش نمایش آهنگ، تصویر و ویدیو. همچنین می تواند فایل ها را جستجو کند و لیست پخش و محتویات آن را نمایش دهد.

گزینه های دستور:

- h نمایش راهنمای دستور
- v نمایش نسخه برنامه
- d مشخص کردن دستگاه رایت دیسک (پیش فرض /dev/sr0)
- s سرعت رایت (پیش فرض حداکثر سرعت)
- f نام فایل یا پوشه ای که می خواهید رایت کنید
- l لیست کردن محتویات دیسک نوری
- e پاک کردن دیسک نوری قابل بازنویسی
- b رایت بوت لودر
- t نوع دیسک نوری (پیش فرض CDROM)
- c ایجاد یک دیسک bootable
- i رایت یک image ISO
- نام فایل image ISO برای خروجی
- n تعداد کپی های مورد نظر برای رایت
- j رایت مستقیم از یک پوشه
- q فعال کردن حالت سکوت
- p پیش نمایش عملیات رایت قبل از اجرا

رایت یک تصویر ISO یا داده‌ها روی یک دیسک:

```
$brasero --burn /path/to/image.iso
```

کپی کردن یک دیسک:

```
$brasero --copy /dev/cdrom
```

رایت یک دیسک صوتی:

```
$brasero --audio /path/to/audio/files
```

پاک کردن محتوای یک دیسک قابل رایت:

```
$brasero --erase /dev/cdrom
```

جدا کردن دیسک از درایو:

```
$brasero --eject /dev/cdrom
```

رایت یک فایل صوتی به نام song.mp3 بر روی CD:

```
brasero -f song.mp3
```

رایت یک پوشه به نام data بر روی DVD:

```
brasero -f data -t DVD
```

لیست کردن محتویات یک دیسک CD:

```
brasero -l
```

پاک کردن یک دیسک DVD قابل بازنویسی:

```
brasero -e
```

رایت بوت لودر Grub بر روی دیسک CD:

```
brasero -b grub
```

ایجاد یک دیسک bootable از سیستم عامل Ubuntu:

```
brasero -i ubuntu.iso -c
```

رایت ۵ کپی از یک فایل image ISO به نام image.iso بر روی CD:

```
brasero -i image.iso -n 5
```

رایت مستقیم محتویات یک پوشه به نام my_project بر روی DVD:

```
brasero -j my_project -t DVD
```

رایت یک فایل صوتی به نام music.wav بر روی CD با سرعت x۴:

```
brasero -f music.wav -s 4x
```

رایت یک دیسک CD بدون نمایش هیچ گونه پیامی:

```
brasero -f data -q
```

لطفاً توجه داشته باشید که در هر مثال، /path/to/image.iso و /path/to/audio/files و /dev/cdrom به ترتیب باید با مسیر و فایل‌های واقعی که می‌خواهید از آن‌ها استفاده کنید جایگزین شوند. همچنین، برای اطلاعات دقیق‌تر و گزینه‌های بیشتر، می‌توانید از دستور brasero --help برای نمایش راهنمای استفاده از برنامه Brasero استفاده کنید.

دستور mkisofs

دستور mkisofs در سیستم عامل‌های Unix/Linux برای ایجاد یک تصویر ایزو (ISO image)از یک فهرست یا ساختار فایل‌ها استفاده می‌شود. این دستور به شما امکان می‌دهد تا یک تصویر ایزو از فایل‌ها و فهرست‌های خود ایجاد کنید که می‌تواند به عنوان یک CD یا DVD مجازی استفاده شود. همچنین دستور mkisofs برای ساخت تصاویر ISO از فایل ها و دایرکتوری ها در سیستم عامل لینوکس استفاده می‌شود. این ابزار می‌تواند برای ایجاد تصاویر قابل بوت برای سیستم عامل ها، دیسک های نوری داده، و یا هر نوع تصویر ISO دیگری استفاده شود.

گزینه های دستور:

- ن‌ام فایل خروجی ISO را مشخص می‌کند.
- r دایرکتوری ریشه ای که باید در تصویر ISO گنجانده شود را مشخص می‌کند.
- b بوت لودر را برای تصویر ISO مشخص می‌کند.
- c فایل بوت لودر را برای تصویر ISO مشخص می‌کند.
- m نقطه اتصال بوت لودر را در تصویر ISO مشخص می‌کند.
- l برچسب دیسک را برای تصویر ISO مشخص می‌کند.
- J سیستم فایل را برای تصویر ISO مشخص می‌کند.
- V نام و نسخه سیستم عامل را برای تصویر ISO مشخص می‌کند.
- A اطلاعات ناشر را برای تصویر ISO مشخص می‌کند.
- T نوع دیسک را برای تصویر ISO مشخص می‌کند.
- d اطلاعات بوت را برای تصویر ISO مشخص می‌کند.

- N نام دیسک را برای تصویر ISO مشخص می‌کند.
- P پسوند نام فایل را برای تصویر ISO مشخص می‌کند.
- f فایل های اضافی را برای گنجاندن در تصویر ISO مشخص می‌کند.
- x فایل ها و دایرکتوری های خاصی را از گنجاندن در تصویر ISO исключает.
- h راهنمای دستور را نمایش می‌دهد.

مثال های دستور:

ایجاد یک تصویر ISO ساده از دایرکتوری /home/user/data:

```
mkisofs -o my-data.iso -r /home/user/data
```

ایجاد یک تصویر ISO قابل بوت از دایرکتوری /home/user/os با بوت لودر grub:

```
mkisofs -o my-os.iso -r /home/user/os -b grub -c /boot/grub/grub.cfg -m /boot
```

ایجاد یک تصویر ISO از دایرکتوری /home/user/data با سیستم فایل ext4 و برچسب دیسک My Data:

```
mkisofs -o my-data.iso -r /home/user/data -J ext4 -l "My Data"
```

ایجاد یک تصویر ISO از دایرکتوری /home/user/data با نام و نسخه سیستم عامل My OS 1.0:

```
mkisofs -o my-os.iso -r /home/user/data -V "My OS 1.0"
```

ایجاد یک تصویر ISO از دایرکتوری /home/user/data با اطلاعات ناشر My Company:

```
mkisofs -o my-data.iso -r /home/user/data -A "My Company"
```

ایجاد یک تصویر ISO از دایرکتوری /home/user/data با نوع دیسک CD-ROM:

```
mkisofs -o my-data.iso -r /home/user/data -T CD-ROM
```

ایجاد یک تصویر ISO از دایرکتوری /home/user/data با اطلاعات بوت menu.lst:

```
mkisofs -o my-data.iso -r /home/user/data -d /boot/menu.lst
```

ایجاد یک تصویر ISO از دایرکتوری /home/user/data با نام دیسک My Data و پسوند iso:.

```
mkisofs -o My\ Data.iso -r /home/user/data -N "My Data" -P iso
```

ایجاد یک تصویر ISO از دایرکتوری /home/user/data با گنجاندن فایل my-file.txt:

```
mkisofs -o my-data.iso -r /home/user/data -f /home/user/my-file.txt
```

ایجاد یک تصویر ISO از دایرکتوری /home/user/data با استثنای دایرکتوری /home/user/data/temp:

```
mkisofs -o my-data.iso -r /home/user/data -x /home/user/data/temp
```

ایجاد یک تصویر ایزو از یک فهرست: این دستور برای ایجاد یک تصویر ایزو از یک فهرست مشخص استفاده می‌شود.

```
mkisofs -o output.iso /path/to/directory
```

تنظیم برچسب برای تصویر ایزو: این دستور برای تنظیم یک برچسب (label) برای تصویر ایزو ایجاد شده استفاده می‌شود.

```
mkisofs -V "MyISO" -o output.iso /path/to/directory
```

اضافه کردن فایل‌ها به تصویر ایزو: این دستور برای اضافه کردن فایل‌ها به تصویر ایزو در زمان ایجاد آن استفاده می‌شود.

```
mkisofs -o output.iso /path/to/directory file1 file2 file3
```

تعیین سطح دسترسی برای فایل‌ها در تصویر ایزو: این دستور برای تعیین سطح دسترسی (فایل‌ها در تصویر ایزو استفاده می‌شود.

```
mkisofs -o output.iso -r /path/to/directory
```

تصویر ایزو ایجاد شده استفاده می‌شود. (filesystem تنظیم فرمت فایل سیستم تصویر ایزو: این دستور برای تنظیم فرمت فایل سیستم

```
mkisofs -o output.iso -J /path/to/directory
```

با استفاده از دستور mkisofs و گزینه‌های مختلف آن، می‌توانید تصاویر ایزو با تنظیمات مختلف برای استفاده در CD یا DVDها ایجاد کنید و به راحتی اطلاعات خود را به صورت یک تصویر قابل حمل جابجا کنید.

دستور genisoimage

دستور genisoimage نیز مانند mkisofs برای ایجاد تصویر ایزو (ISO imageاز فایل‌ها و فهرست‌ها در سیستم عامل‌های Unix/Linux استفاده می‌شود. این دستور به شما امکان می‌دهد تا تصویر ایزویی از اطلاعات خود ایجاد کرده و برای ایجاد CD یا DVD مورد استفاده قرار دهید. همچین دستور genisoimage برای ایجاد تصاویر دیسک نوری ISO از فایل ها و دایرکتوری ها در سیستم عامل لینوکس استفاده می‌شود. این ابزار می‌تواند برای ساخت انواع مختلف دیسک‌های نوری مانندCD-ROM ، DVD، و Blu-ray استفاده شود.

گزینه های دستور:

- o÷ نام فایل تصویری ISO خروجی را مشخص می‌کند.
- r دایرکتوری که می خواهید از آن تصویر ISO بسازید را مشخص می‌کند.
- b بوت لودر را برای تصویر ISO مشخص می‌کند.
- V نام و نسخه سیستم عامل را برای تصویر ISO مشخص می‌کند.
- J سطح Rock Ridge را برای تصویر ISO مشخص می‌کند.
- joliet از فرمت Joliet برای نام فایل ها در تصویر ISO استفاده می‌کند.
- h راهنمای دستور را نمایش می‌دهد.

مثال های دستور:

ایجاد تصویر ایزو از یک فهرست: این دستور برای ایجاد یک تصویر ایزو از یک فهرست مشخص استفاده می‌شود.

```
genisoimage -o output.iso /path/to/directory
```

تنظیم برچسب برای تصویر ایزو: این دستور برای تنظیم یک برچسب (labelبرای تصویر ایزو ایجاد شده استفاده می‌شود.

```
genisoimage -V “MyISO” -o output.iso /path/to/directory
```

اضافه کردن فایل‌ها به تصویر ایزو: این دستور برای اضافه کردن فایل‌ها به تصویر ایزو در زمان ایجاد آن استفاده می‌شود.

```
genisoimage -o output.iso /path/to/directory file1 file2 file3
```

تعیین سطح دسترسی برای فایل‌ها در تصویر ایزو: این دستور برای تعیین سطح دسترسی (permissionsفایل‌ها در تصویر ایزو استفاده می‌شود.

```
genisoimage -o output.iso -r /path/to/directory
```

تنظیم فرمت فایل سیستم تصویر ایزو: این دستور برای تنظیم فرمت فایل سیستم (filesystemتصویر ایزو ایجاد شده استفاده می‌شود.

```
genisoimage -o output.iso -J /path/to/directory
```

ایجاد تصویر ISO از دایرکتوری /home/user/data:

```
genisoimage -o my-image.iso -r /home/user/data
```

ایجاد تصویر ISO از دایرکتوری /home/user/data با بوت لودر grub:

```
genisoimage -o my-image.iso -r /home/user/data -b /usr/lib/grub/i386-pc/boot.img
```

ایجاد تصویر ISO از دایرکتوری /home/user/data با نام و نسخه سیستم عامل “My OS 1.0”:

```
genisoimage -o my-image.iso -r /home/user/data -V “My OS 1.0”
```

ایجاد تصویر ISO از دایرکتوری /home/user/data/ با سطح Rock Ridge 2:

```
genisoimage -o my-image.iso -r /home/user/data -J 2
```

ایجاد تصویر ISO از دایرکتوری /home/user/data/ با استفاده از فرمت Joliet:

```
genisoimage -o my-image.iso -r /home/user/data -joliet
```

دستور eject

دستورeject برای از خارج کردن رسانه هایی مانند DVD / CD ROM یا فلاپی دیسک از system استفاده می شود. اجازه می دهد تا یک رسانه قابل جابجایی (معمولاً CD-ROM ، فلاپی دیسک ، نوار ، یا دیسک JAZ یا ZIP) را با استفاده از نرم افزار خارج کنید .همچنین می توانید برخی از تعویض های CD-ROM چند دیسک ، ویژگی خودکار خروج را کنترل کنید (توسط برخی دستگاه ها پشتیبانی می شود) و همچنین سینی دیسک برخی از درایوهای CD-ROM را ببندید .روش های بیرون انداختن به نوع دستگاه بستگی دارد ، چه CD-ROM باشد ، چه دستگاه SCSI ، فلاپی قابل جدا شدن یا نوار .عملکرد پیش فرض eject این است که هر چهار روش را به ترتیب امتحان می کند تا زمانی که به موفقیت برسد .اگر mount باشد ، قبل از بیرون انداختن دستگاه از آن خارج نمی شود.

```
$ eject /dev/cdrom
$ eject /mnt/cdrom /
$ eject /dev/sda
```

ساختار دستور:

```
eject [... OPTIONS]
```

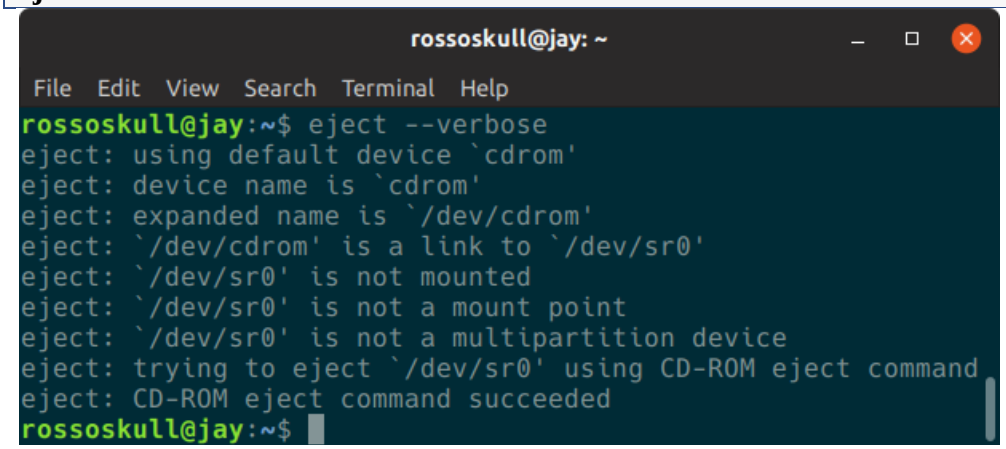
گزینه های دستور:

- f به طور پیش فرض، eject قبل از خارج کردن دیسک، از کاربر تأیید می‌خواهد. با استفاده از این گزینه، دیسک بدون تأیید کاربر خارج می‌شود.
- l لیست درایوهای نوری موجود را نمایش می‌دهد.
- d نام درایو نوری specific را برای خارج کردن دیسک از آن مشخص می‌کند.
- t نوع دیسک را برای خارج کردن مشخص می‌کند.
- m پیامی را قبل از خارج کردن دیسک نمایش می‌دهد.
- h راهنمای دستور را نمایش می‌دهد.

خروجی دستور:

- **v-یا verbose:** اطلاعات بیشتر در مورد اجرای دستور را از سیستم خارج کنید.

```
eject -v
```

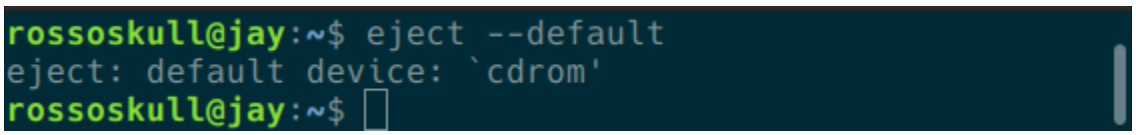


-d or --default نام دستگاه پیش فرض را لیست کنید.

ساختار دستور:

```
eject -d
```

خروجی دستور:



با این دستور حالت بیرون انداختن خودکار را که توسط برخی دستگاه ها پشتیبانی می شود کنترل کنید .هنگام بسته شدن دستگاه ، این کار را برای بیرون انداختن خودکار فعال کنید.

ساختار دستور:

```
eject -a on
```

- **-c or –changerslot :** یک اسلات CD را از یک تعویض CD-ROM ATAPI / IDE انتخاب کنید .برای استفاده از این ویژگی به Linux 2.0+ نیاز دارید .شما باید مطمئن شوید که اسلات CD از قبل استفاده نشده است .اولین شکاف موجود با عنوان صفر شناخته می شود.

ساختار دستور:

```
eject -c 0
```

-t or –trayclose : با استفاده از این دستور درایو را به یک دستور close می دهید.

توجه : پشتیبانی سخت افزاری از این ویژگی محدود است.

ساختار دستور:

```
eject -t
```

- **T-یا traytoggle–:** در صورت باز بودن شکاف ، دستور بستن سینی را می دهد یا اگر شکاف بسته است ، یک دستور eject می دهد.

توجه : پشتیبانی سخت افزاری برای این دستور محدود است، به عنوان آن استفاده می کند از بالا *t* -فرمان.

ساختار دستور:

```
eject -T
```

- **-x یا cdspeed–** :با این گزینه به CD-ROM دستور speed را به درایو بدهید .آرگومان یک عدد است که نشان دهنده سرعت مورد نظر است (به عنوان مثال ۳ برای سرعت 3X یا ۰ برای حداکثر سرعت داده .پشتیبانی دستگاه از این دستور محدود است و شما فقط می توانید سرعتی را که درایو توانایی آن را دارد مشخص کنید .با تغییر رسانه ، این گزینه پاک می شود .این گزینه با دستور c- یا t- نیز قابل استفاده است.

ساختار دستور:

```
eject -x 0
```

- **-X or –listspeed** :لیستی از سرعتهای موجود را از درایو CD-ROM با استفاده از این گزینه نمایش دهید .می توانید از سرعت های ذکر شده به عنوان آرگومان در گزینه x- استفاده کنید .برای کار با این گزینه به Linux 2.6.13+نیاز دارید .در نسخه های قدیمی ، فقط حداکثر سرعت را گزارش می دهد .همچنین ، برخی از درایوها ممکن است سرعت را به درستی گزارش نکنند ، بنابراین این گزینه با آنها کار نمی کند.

ساختار دستور:

```
eject -X
```

- **-n or –noop** :دستگاه انتخاب شده را نمایش دهید ، اما با استفاده از این گزینه هیچ عملی انجام **ندهید**.

ساختار دستور:

```
eject -n
```

خروجی دستور:

```
rossoskull@jay:~$ eject -n
eject: device is `/dev/sr0'
rossoskull@jay:~$
```

- **-r or –cdrom** :مشخص کنید که درایو باید با استفاده از دستور CDROM eject با استفاده از این گزینه خارج شود.

ساختار دستور:

```
eject -r
```

- **-s or –scsi** :مشخص کنید که درایو باید با استفاده از دستورات SCSI با استفاده از این گزینه خارج شود.

ساختار دستور:

```
eject -s
```

- **-f or –floppy** :مشخص کنید که درایو باید با استفاده از یک دستور فلاپی دیسک قابل جابجایی با استفاده از این گزینه خارج شود.

ساختار دستور:

```
eject -f
```

- **-q or –tape** :مشخص کنید که درایو باید با استفاده از یک دستور آفلاین درایو tape با استفاده از این گزینه خارج شود.

ساختار دستور:

```
eject -q
```

- **-p or –proc** :با این دستور از / proc / mounts به جای / etc / mtab استفاده کنید .همچنین از گزینه -n برای umount عبور می کند.

ساختار دستور:

```
eject -p
```

- **-m یا –no-unmount** :اجازه کار کردن با درایورهای دستگاه را که به طور خودکار رسانه های قابل جابجایی را نصب می کنند کار می کند و بنابراین باید همیشه نصب شود .این گزینه به eject می گوید که سعی نکنید دستگاه مورد نظر را پیاده کنید ، حتی اگر دستگاه آن مطابق با:

```
/ proc / mounts
```

یا

```
/ etc / mtab
```

ساختار دستور:

```
eject -m
```

- **-V or –version** :شماره نسخه را برای کنسول وارد کرده و از آن خارج شوید.

ساختار دستور:

```
eject -V
```

فصل شانزدهم

فصل شانزدهم

دستور find

نحوه استفاده از دستور "find" برای جستجو برای نام فایل های چندگانه (پسوند) در linux

چندین بار، ما در موقعیتی قرار داریم که در آن ما باید چندین فایل با پسوند های مختلف را جستجو کنیم، این احتمالا به چندین کاربر linux خصوصا در داخل ترمینال رخ داده است. چندین ابزار linux وجود دارد که ما می توانیم از آنها برای یافتن یا find فایل ها در سیستم فایل استفاده کنیم، اما find نام فایل های متعدد و یا فایل های با پسوند های مختلف گاهی اوقات می تواند پیچیده و نیاز به دستورات خاص. یکی از بسیاری از امکانات برای یافتن فایل ها در یک سیستم فایل linux ابزار findاست و در این راهنمای نحوه استفاده از چند نمونه از استفاده از findبرای کمک به ما در find چندین نام فایل در یک زمان وارد می کنیم.

گزینه های دستور:

- path مسیر دایرکتوری را برای جستجو مشخص می کند.
- name نام فایل را برای جستجو مشخص می کند.
- type نوع فایل را برای جستجو مشخص می کند.
- mtime فایل هایی را پیدا می کند که در مدت زمان مشخص شده تغییر کرده
- atime فایل هایی را پیدا می کند که در مدت زمان مشخص شده به آنها دسترسی
- ctime فایل هایی را پیدا می کند که در مدت زمان مشخص شده تغییر
- size فایل هایی را با اندازه مشخص شده پیدا می کند.
- perm فایل هایی را با مجوز مشخص شده پیدا می کند.
- exec دستور مشخص شده را برای هر فایل یافت شده اجرا می
- print مسیر هر فایل یافت شده را به صورت

مثال های دستور:

قسمت اول – دستورالعمل اساسی برای یافتن فایل با نام

تمام فایل هایی که نام آن hosein .txt را در یک فهرست کاری فعلی پیدا کنید پیدا کنید.

```
# find  .-name  hosein .txt
```

همه فایل ها را در زیر فهرست home /با نام hosein .txtپیدا میکند.

```
# find / home-name  hosein .txt
/home/ hosein .txt
```

تمام فایل هایی که اسم hosein .txtرا پیدا کرده که حاوی حروف بزرگ و کوچک در فهرست home /است.

```
# find / home-name  hosein .txt
./ hosein .txt
./ hosein .txt
```

یافتن راهنماها با استفاده از نام

یافتن همه فهرست هایی که نام آن hosein و در فهرست ریشه است.

```
# find / -type d -name  hosein
/  hosein
```

همه فایل های PHP با استفاده از نام

پیداکردن تمام فایل های phpکه نام آن hosein .phpدر یک فهرست کاری فعلی است.

```
# find .-type f -name  hosein .php
```

./ hosein .php

تمام فایل های PHP در فهرست

همه فایل های phpرا در یک فهرست پیدا می کند.

```
# find .-type f -name “* .php”
```

./ hosein .php
./login.php
./index.php

یافتن فایل ها براساس مجوزهای آنها

جستجو فایل ها با مجوز ۷۷۷

تمام فایل هایی که مجوزها 777هستند پیدا می کند.

```
# find  . –type  f  -perm 0777  -print
```

فایل های SGID با مجوز ۶۴۴

تمام فایلهای bit SGID که مجوزها آنها 644است تنظیم شده را پیدا می کند.

```
# find / -perm  2644
```

یافتن فایل های bit با ۵۵۱ مجوز

تمام فایل های مجموعه Sticky Bit را که اجازه دارند 551پیدا کنید پیدا می کند.

```
# find / -perm 1551
```

همه فایل های SUID را پیدا می کند.

```
# find / -perm / u = s
```

فایل هایSGID

تمام فایل های مجموعه SGID

```
# find / -perm / g = s
```

فقط فایل های قابل خواندن[۱] را پیدا می کند.

```
# find / -perm / u = r
```

یافتن فایل های اجرایی

همه فایلهای اجرایی را پیدا میکند.

```
# find / -perm / a = x
```

فایل ها را با ۷۷۷ مجوز و Chmod به ۶۴۴ پیدا میکند.

```
# find / -type f -perm 0777 -print -exec chmod 644 {} \;
```

یافتن راهنماها با مجوز ۷۷۷ و تغییر مجوز به ۷۵۵

```
# find / -type d -perm 777 -print  -exec chmod 755 {} \;
```

جستجو یک فایل به نام hosein .txtو حذف آن.

---

[۱] read only

```
# find .-type f -name “ hosein .txt” -exec rm -f {} \;
```

فایل چندگانه را پیدا کنید و حذف کنید.

برای find و حذف چندین فایل مانند mp3.یا txt.، سپس از آن استفاده کنید.

```
# find .-type f -name “* .txt” -exec rm -f {} \;
```

یا

```
# find .-type f -name “* .mp3” -exec rm -f {} \;
```

تمام فایل های empty را پیدا کنید.

برای جستجو همه فایلهای خالی[1] در مسیر خاص

```
# find / tmp -type f -type
```

همه راهنمایی های خالی را پیدا کنید.

```
# find / tmp -type d-empty
```

برای جستجو همه فایل های مخفی[2]، از دستور زیر استفاده کنید.

```
# find / tmp -type f -name “. *”
```

فایل های جستجو بر اساس مالکیت و گروه ها

برای جستجو همه یا یک فایل تک به نام hosein .txtزیر root/ با مالکیت root پیدا میکند.

```
# find / -user root -name  hosein .txt
```

جستجو همه فایل هایی که متعلق به hosein کاربر زیر فهرست home /هستند.

```
# find / home -user  hosein
```

برای جستجو همه فایل هایی که متعلق به گروه توسعه در زیر فهرست home /هستند.

```
# find / home -group developer
```

برای جستجو همه فایل های txt.از کاربر hosein زیر فهرست home. /

```
# find / home -user  hosein  -name “* .txt”
```

جستجو فایل ها و راهنماها بر اساس تاریخ و زمان

جستجو تمام فایل هایی که به مدت 50روز به آنها دسترسی پیدا شده است.

```
# find / -atme 50
```

جستجو تمام فایل هایی که بیش از 50روز قبل و کمتر از 100روز تغییر کرده اند.

```
# find / -mtime +50 -mtime -100
```

جستجو همه فایل هایی که در 1ساعت گذشته تغییر کرده اند.

```
# find / -cmin -60
```

جستجو همه فایل هایی که در 1ساعت گذشته اصلاح شده اند.

```
# find / -mmin -60
```

---

[1] empty file

[2] hidden file

جستجو همه فایل هایی که در 1ساعت گذشته دیده می شوند.

```
# find / -amin -60
```

فایل ها و راهنماها بر اساس اندازه

جستجو تمام فایل های 50مگابایتی

```
# find / -size 50M
```

جستجو همه فایل هایی که بیش از 50مگابایت و کمتر از 100MBهستند.

```
# find / -size + 50M-size -100M
```

جستجو همه فایل های 100MBو حذف آنها با استفاده از یک دستور تنها.

```
# find / -size + 100M -exec rm -rf {} \;
```

جستجو همه فایل های mp3.را با بیش از 10مگابایت پیدا میکند و آنها را با استفاده از دستور rm[1] حذف میکند.

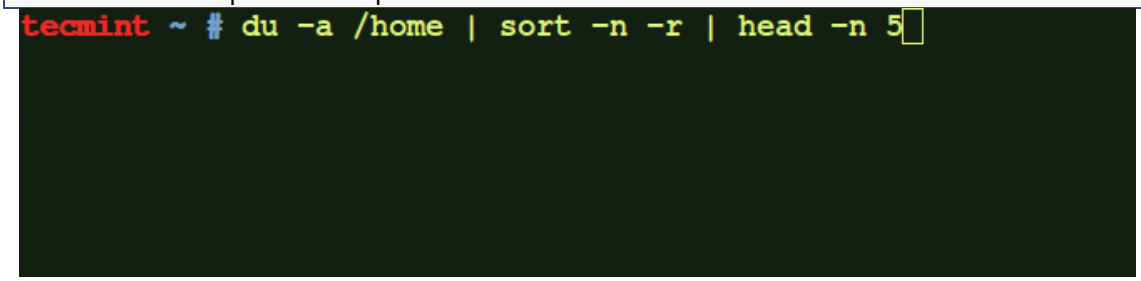
```
# find / -type f -name * .mp3 -size + 10M -exec rm {} \;
```

به عنوان یک مدیر linux، باید به صورت دوره ای بررسی کنید که چه فایل ها و فهرست ها فضای دیسک بیشتری را مصرف می کنند.بسیار مهم است که junks های غیر ضروری را پیدا کنید و آنها را از هارد دیسک خود آزاد کنید.

نحوه یافتن بزرگترین فایل ها و راهنماها در linux

دستور زیر را برای find بزرگترین بزرگترین فهرست زیر home/پارتیشن . home/

```
# du -a / home | sort -n -r | head -n 5
```

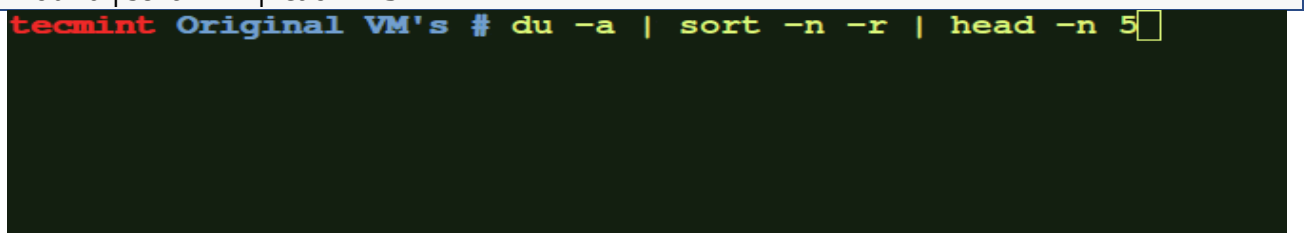


یافتن بزرگترین فهرست ها در linux

دستور بالا ۵ تا از بزرگترین فهرست پارتیشن home / را نمایش می دهد.

اگر می خواهید بزرگترین فهرست ها را در فهرست فعلی نمایش دهید، اجرا کنید:

```
# du -a | sort -n -r |head  -n 5
```




```
# du -hs * | sort -rh |head ۵
```

[1] remove

```
tecmint Original VM's # du -hs * | sort -rh | head -5
```

دستور بالا فهرست های بالا را نشان می دهد که فضای دیسک بیشتری دارند.اگر احساس می کنید که برخی از فهرست ها مهم نیستند، می توانید به سادگی چند زیر فهرست را حذف کنید یا کل فهرست را برای آزاد کردن فضای empty پاک کنید.

برای نمایش بزرگترین فهرست ها / فایل ها از جمله زیر شاخه ها، اجرا کنید:

```
# du -Sh | sort -rh |head ۵
```

```
tecmint tecmint # du -Sh | sort -rh | head -5
```

اگر می خواهید بزرگترین اندازه فایل را نمایش دهید، سپس دستور زیر را اجرا کنید:

```
# find-type f -exec du -Sh {} + | sort -rh |head  -n 5
```

```
tecmint tecmint # find -type f -exec du -Sh {} + | sort -rh | head -n 5
```

برای جستجو بزرگترین فایل ها در یک مکان خاص، فقط مسیر را در کنار دستور find :

```
# find / home /  hosein  / downloads / -type f -exec du -Sh {} + | sort -rh |سر -n 5
```

یا

```
# find / home /  hosein  / Downloads / -type f -printf “٪s٪p \ n” | sort-rn |سر -n 5
```

```
tecmint tecmint # find Downloads/ -type f -exec du -Sh {} + | sort -rh | head -n
5
```

دستور locate

دستوری برای جستجو فایل ها می باشد. دستور زیر یک فایل را با نام دقیق آن جستجو می کند:

```
$ locate -b '\domain-list.txt'
```

از دستور location در لینوکس برای locate فایل ها با نام استفاده می شود .دو نرم افزار پرکاربرد برای جستجوی فایل که برای کاربران قابل دسترسی است وجود دارد که به آنها "پیدا کردن و مکان یابی" می گویند . این ابزار بهتر و سریع تر کار می کند از find دستور همتای دیگر. چرا که به جای جستجو در فایل سیستم زمانی که یک جستجو فایل آغاز شده است، آن را از طریق یک پایگاه داده است .این پایگاه داده شامل بیت ها و قسمت هایی از فایل ها و مسیرهای مربوط به آنها در سیستم شما است .به طور پیش فرض ، دستور loc بررسی نمی کند که آیا فایل های موجود در پایگاه داده هنوز وجود دارند یا خیر و هرگز فایل های ایجاد شده پس از آخرین به روزرسانی پایگاه داده مربوطه را گزارش نمی کند.

ساختار دستور:

```
locate ... [گزینه] ...
```

دستور locate در سیستم‌عامل‌های Unix/Linux برای جستجوی سریع و پیدا کردن مسیر فایل‌ها و فهرست‌ها بر اساس نام آن‌ها استفاده می‌شود. این دستور از یک پایگاه داده فایل به نام updatedb برای ذخیره و به روزرسانی اطلاعات فایل‌ها استفاده می‌کند تا جستجوها به صورت سریع انجام شود.

گزینه های دستور:

- d پایگاه داده ی locate specific را برای جستجو مشخص می کند.
- f فقط نام فایل ها را بدون مسیر نمایش می دهد.
- i جستجو را به حروف کوچک و بزرگ حساس می کند.
- l تعداد خطوطی که نام فایل ها در آن ها نمایش داده می شوند را مشخص می کند.
- n تعداد نتایج جستجو را محدود می کند.
- q خروجی را فقط به نام فایل ها محدود می کند.
- r جستجو را به صورت بازگشتی در تمام زیرشاخه ها انجام می دهد.
- s اندازه فایل ها را بر حسب بایت نمایش می دهد.
- t نوع فایل ها را نمایش می دهد.
- u مالک فایل ها را نمایش می دهد.
- g گروه فایل ها را نمایش می دهد.
- v اطلاعات نسخه ی locate را نمایش می دهد.
- h راهنمای دستور را نمایش می دهد.

مثال های دستور:

جستجوی فایل myfile.txt در تمام سیستم:

```
locate myfile.txt
```

جستجوی فایل myfile.txt فقط در دایرکتوری /:home/user

```
locate -d /home/user myfile.txt
```

جستجوی فایل myfile.txt بدون نمایش مسیر:

```
locate -f myfile.txt
```

جستجوی فایل myfile.txt با در نظر گرفتن حروف کوچک و بزرگ:

```
locate -i myfile.txt
```

جستجوی ۱۰ مورد اول از فایل های *.txt:

```
locate -n 10 *.txt
```

جستجوی فایل myfile.txt و نمایش فقط نام فایل:

```
locate -q myfile.txt
```

جستجوی بازگشتی فایل myfile.txt در تمام زیرشاخه ها:

```
locate -r myfile.txt
```

جستجوی فایل myfile.txt و نمایش اندازه آن:

```
locate -s myfile.txt
```

جستجوی فایل myfile.txt و نمایش نوع آن:

```
locate -t myfile.txt
```

جستجوی فایل myfile.txt و نمایش مالک آن:

```
locate -u myfile.txt
```

جستجوی فایل myfile.txt و نمایش گروه آن:

```
locate -g myfile.txt
```

نمایش اطلاعات نسخه ی locate:

```
locate -v
```

نمایش راهنمای دستور:

```
locate -h
```

دستور updatedb

دستور updatedb در سیستم عامل‌های Unix/Linux برای به‌روزرسانی پایگاه داده جستجوی فایل‌ها توسط دستور locate استفاده می‌شود. این دستور با اسکن تمام فایل‌ها و فهرست‌های موجود در سیستم، اطلاعات مربوط به محل آن‌ها را در پایگاه داده بروزرسانی می‌کند. همچنین دستور updatedb پایگاه داده locate را که برای جستجوی سریع فایل ها در سیستم عامل لینوکس استفاده می شود، به روز می کند. این دستور به طور پیش فرض به صورت روزانه توسط cron اجرا می شود، اما شما می توانید آن را به صورت دستی نیز اجرا کنید.

گزینه های دستور:

- f به طور پیش فرض، updatedb فقط فایل های جدید و تغییر یافته را در پایگاه داده locate به روز می کند. با استفاده از این گزینه، تمام فایل ها در سیستم شما به روز می شوند.
- l فقط پایگاه داده locate را قفل می کند.
- u فقط پایگاه داده locate را باز می کند.
- t فقط پایگاه داده locate را تست می کند.
- S فقط پایگاه داده locate را به روز می کند.
- d مسیر دایرکتوری specific را برای به روز رسانی مشخص می کند.
- p فایل پیکربندی specific را برای استفاده مشخص می کند.
- h راهنمای دستور را نمایش می دهد.

مثال های دستور:

به روز رسانی پایگاه داده locate به صورت دستی:

```
updatedb
```

به روز رسانی تمام فایل ها در سیستم:

```
updatedb -f
```

قفل کردن پایگاه داده locate:

```
updatedb -l
```

باز کردن پایگاه داده locate:

```
updatedb -u
```

تست کردن پایگاه داده locate:

```
updatedb -t
```

به روز رسانی پایگاه داده locate بدون اسکن فایل ها:

```
updatedb -S
```

به روز رسانی پایگاه داده locate برای دایرکتوری /home/user:

```
updatedb -d /home/user
```

استفاده از فایل پیکربندی specific:

```
updatedb -p /etc/updatedb.conf
```

نمایش راهنمای دستور:

```
updatedb –h
```

به‌روزرسانی پایگاه داده با پیام نمایش دادن:

این دستور برای به‌روزرسانی پایگاه داده با نمایش پیام‌های مربوط به فرآیند استفاده می‌شود.

```
sudo updatedb
```

به‌روزرسانی پایگاه داده بدون نمایش پیام:

این دستور برای به‌روزرسانی پایگاه داده بدون نمایش پیام‌های مربوط به فرآیند استفاده می‌شود.

```
sudo updatedb -q
```

مشخص کردن مسیر برای ذخیره پایگاه داده:

این دستور برای ذخیره پایگاه داده در یک مسیر خاص استفاده می‌شود.

```
sudo updatedb -o /path/to/database
```

اجرای به‌روزرسانی به صورت خودکار:

این دستور برای اجرای به‌روزرسانی پایگاه داده به صورت خودکار در زمان‌های مشخص استفاده می‌شود.

```
sudo updatedb --incremental
```

تغییر فرکانس به‌روزرسانی:

این دستور برای تغییر فرکانس به‌روزرسانی پایگاه داده استفاده می‌شود.

```
sudo updatedb --frequency=weekly
```

با استفاده از دستور updatedb و گزینه‌های مختلف آن، می‌توانید پایگاه داده جستجوی فایل‌ها را به‌روز نگه داشته و به سرعت و دقت بیشتری به فایل‌های مورد نیاز دسترسی پیدا کنید.

فصل هفدهم

فصل هفدهم

مدیریت شبکه

دستور hostname

از این دستور استفاده می شود تا به راحتی دستگاه را در یک شبکه در فرمت قابل خواندن انسان تشخیص دهد.جای تعجب نیست، اما در linux system، نام host را می توان به راحتی با استفاده از دستور ساده به عنوان " نام میزبان "تغییر داد

گزینه های دستور:

- A تمام نام های میزبان ((aliasرا نمایش می‌دهد.
- d نام دامنه ((domain nameرا نمایش می‌دهد.
- f نام فایل حاوی نام های میزبان را مشخص می‌کند.
- i آدرس IP سیستم را نمایش می‌دهد.
- s نام میزبان را تنظیم می‌کند.
- v اطلاعات نسخه و جزئیات مربوط به دستور را نمایش می‌دهد.
- h راهنمای دستور را نمایش می‌دهد.

مثال های دستور:

نام hostخود را به بدون هیچ پارامتری، نام hostsystem عامل linux خود را به این صورت نشان می دهد:

```
$ hostname
hosein
```

اگر میخواهید نام linux hostsystem خود را تغییر دهید یا تنظیم کنید، به سادگی اجرا کنید:

```
$ hostname NEW_HOSTNAME
```

البته، باید "NEW_HOSTNAME"را با نام hostواقعی که می خواهید تنظیم کنید، جایگزین کنید.این نام hostsystem را بلافاصله تغییر می دهد، اما یک مشکل وجود دارد – نام اصلی اصلی بعد از restart دوباره بازیابی خواهد شد .راه دیگری برای تغییر نام hostsystem وجود دارد که به طور دائمی است. ممکن است قبلا آن را کشف کرده باشید که در برخی از فایل های تنظیمات نیاز به تغییر دارد.

نمایش نام میزبان:

```
hostname
```

نمایش تمام نام های میزبان:

```
hostname -A
```

نمایش نام دامنه:

```
hostname -d
```

تنظیم نام میزبان به "my-hostname":

```
hostname -s my-hostname
```

نمایش آدرس IP:

```
hostname -i
```

نمایش اطلاعات نسخه:

```
hostname -v
```

نمایش راهنمای دستور:

```
hostname -h
```

ایجاد نام هاست دائمی

نسخه جدیدتر توزیعهای linux مانند آخرین Ubuntu، Debian، CentOS، Fedora، RedHatو غیره با SystemD، یک system و مدیریت سرویس است که دستور commandnamectlرا برای مدیریت نام host در linux فراهم می کند.برای تنظیم نام hostsystem در توزیعهای مبتنی بر SystemD، از دستور hostnamectl به عنوان نشان داده شده استفاده خواهیم کرد:

```
$ sudo hostnamectl    host   NEW_HOSTNAME
```

برای توزیع های قدیمی تر linux که SysVinit را در init short استفاده می کنند، می توانند نام های hostخود را با تغییر دادن نام پرونده های host در سایت تغییر دهند.

```
# vi / etc / hostname
```

پس از آن باید یک رکورد دیگر برای نام hostرا وارد کنید:

```
# vi / etc / hosts
127.0.0.1  hosein
```

سپس باید اجرا کنید:

```
# /etc/init.d/hostname restart
```

در system های مبتنی بر RHEL / CentOSکه از initاستفاده می کنند، نام hostبا اصلاح تغییر می کند:

```
# vi / etc / sysconfig / network
```

در اینجا یک نمونه از آن فایل وجود دارد:

```
/ etc / sysconfig / network
NETWORKING = yes
HOSTNAME = “ hosein .com”
GATEWAY = “192.168.0.1”
GATEWAYDEV = “eth0”
FORWARD_IPV4 = “yes”
```

برای نگه داشتن نام hostدائمی، مقدار در کنار "HOSTNAME"به یکی از نام hostخود تغییر دهید.

دستور host

دستور **host** در سیستم لینوکس برای عملیات جستجوی DNS سیستم نام دامنه استفاده می شود .به عبارت ساده ، از این دستور برای یافتن آدرس IP یک نام دامنه خاص استفاده می شود یا اگر می خواهید از نام دامنه آدرس IP خاصی استفاده کنید ، دستور host مفید واقع می شود .با تعیین گزینه مربوطه به همراه نام دامنه ، میتوانید جزئیات دقیق تری از دامنه را نیز پیدا کنید.

ساختار دستور:

```
host  [-aCdlriTWV] [-c class] [-N ndots] [-t type] [-W time]
[-R number] [-m flag] hostname [server]
```

مثال های دستور:

این دستورالعمل عمومی دستور را به همراه گزینه های مختلفی که می توان با دستور host استفاده کرد چاپ می کند و همچنین توضیح مختصری درباره هر گزینه ارائه می دهد.

```
anshul@anshul-VirtualBox:~$ host
Usage: host [-aCdilrTvVw] [-c class] [-N ndots] [-t type] [-W time]
            [-R number] [-m flag] hostname [server]
       -a is equivalent to -v -t ANY
       -c specifies query class for non-IN data
       -C compares SOA records on authoritative nameservers
       -d is equivalent to -v
       -i IP6.INT reverse lookups
       -l lists all hosts in a domain, using AXFR
       -m set memory debugging flag (trace|record|usage)
       -N changes the number of dots allowed before root lookup is done
       -r disables recursive processing
       -R specifies number of retries for UDP packets
       -s a SERVFAIL response should stop query
       -t specifies the query type
       -T enables TCP/IP mode
       -v enables verbose output
       -V print version number and exit
       -w specifies to wait forever for a reply
       -W specifies how long to wait for a reply
       -4 use IPv4 query transport only
       -6 use IPv6 query transport only
anshul@anshul-VirtualBox:~$
```

host domain_name :با این کار جزئیات آدرس IP دامنه مشخص شده چاپ می شود.

مثال :

```
host  google.com
```

host IP_Address :با این کار جزئیات دامنه آدرس IP مشخص شده نمایش داده می شود.

مثال :

```
host۵۲,۲۵,۱۰۹,۲۳۰
```

```
anshul@anshul-VirtualBox:~$ host 52.25.109.230
230.109.25.52.in-addr.arpa domain name pointer ec2-52-25-109-230.us-west-2.compute.amazonaws.com.
anshul@anshul-VirtualBox:~$
```

-a or -v :برای مشخص کردن نوع پرس و جو استفاده می شود یا خروجی کلامی را فعال می کند.

مثال :

```
host  -a google.com
```

-t :برای تعیین نوع پرس و جو استفاده می شود.

مثال ۱:

```
host  -t ns google.com
```

مثال ۲ :برای چاپ رکوردSOA

```
host  -t SOA google.com
```

مثال ۳ :برای چاپ رکورد*txt*

```
host -t txt google.com
```

C-به منظور مقایسه سوابق SOA در سرورهای نام معتبر.
مثال :

```
host -C google.com
```

R- به منظور تعیین تعداد دوباره امتحانات ، می توانید در صورت عدم موفقیت در یکبار انجام دهید .اگر کسی موفق شود ، دستور متوقف می شود.
مثال :

```
host -R 3 google.com
```

l- برای لیست کردن همه host ها در یک دامنه. برای کار با این دستور شما باید یک مدیر یا یک سرور گره باشید.
مثال :

```
host -l google.com
```

دستور domainname

از دستور **domainname** در Linux برای بازگرداندن نام دامنه سیستم اطلاعات شبکه (NIS) میزبان استفاده می شود `hostname -d` .برای بدست آوردن نام دامنه میزبان می توانید از دستورات نیز استفاده کنید .اگر نام دامنه در میزبان شما تنظیم نشده باشد ، پاسخ "هیچ" است.در اصطلاحات شبکه ، نام دامنه نگاشت IP با نام است .در صورت وجود شبکه محلی ، نام دامنه در سرور DNS ثبت می شود .اگر سرور DNS وجود ندارد ، می توانید ورودی را در پرونده *etc / hosts* /*قرار* دهید تا آدرس IP را با نام دامنه ترسیم کند.

ساختار دستور:

```
domainname [options]
```

گزینه های دستور:

d-یا domain– نام دامنه DNS را نمایش می دهد.

f-، fqdn–، long– نام میزبان طولانی نام دامنه کاملاً واجد شرایط (FQDN).

F-، file– نام میزبان یا نام دامنه NIS را از فایل داده شده میخواند.

مثال های دستور:

domainname –h: برای نمایش همه گزینه ها و نحو دستور استفاده می شود.

```
domainname –h
```

domainname -a یاalias–: برای نمایش نام مستعار استفاده می شود .اگر نام مستعار تنظیم نشده باشد ، خط خالی را برمی گرداند.

```
[root@allinone ~]# domainname -A
allinone allinone
```

domainname -A یاall-fqdns– : برای نمایش تمام نام های دامنه کاملاً واجد شرایط (FQDN) استفاده می شود.

domainname -b یاboot–: برای تنظیم نام دامنه پیش فرض در صورت موجود نبودن استفاده می شود.

در مثال زیر ، می توانید در ابتدا نام دامنه "هیچکدام" برگشت داده شود اما پس از تنظیم دستور ، نام جدید را بر می گرداند.

```
[root@os ~]# domainname
(none)
[root@os ~]# domainname -b allinone
[root@os ~]# domainname
allinone
[root@os ~]#
```

domainname -s or –short: نام کوتاه میزبان را نشان می دهد.

```
[root@os ~]# domainname -s
os
```

domainname -I یا –all-ip-addresses: برای نمایش تمام آدرس های IP اختصاص داده شده به میزبان استفاده می شود ، برای بررسی آدرس های IP از دستور ip -a استفاده کنید.

```
[root@os ~]# domainname -I
10.164.228.233
```

domainname -i یا –ip-address: آدرس localhost را نمایش می دهد.

```
[root@os ~]# domainname -i
::1%1 127.0.0.1
```

domainname -y یا –nis یا –yp: نام دامنه سیستم اطلاعات شبکه (NIS) را نمایش می دهد.

در این مثال می بینید ، نام دامنه نمایش داده شده همان است که ما با استفاده از گزینه b- تنظیم کردیم.

تنظیمات شبکه

Linux TCP/IP Network Configuration Files:

File	شرح
/etc/resolv.conf	دارای سرورهای DNSرا برای نام دامنه اینترنتی
/etc/hosts	هاست هایی را فهرست می کند که باید به صورت محلی هستند
/etc/nsswitch.conf	فهرست ترتیب جستجوی نام میزبان .به طور معمول به فایل های محلی، سپس سرور NISو سپس سرور DNSنگاه کنید.
Red Hat/Fedora/CentOS: /etc/sysconfig/network	پیکربندی شبکه را مشخص میکند .به عنوان مثال IP .استاتیک ، DHCP، NISو غیره
RedHat/Fedora/CentOS: /etc/sysconfig/network-scripts/ifcfg-device	اطلاعات شبکه TCP را مشخص میکند.
Ubuntu/Debian: /etc/network/interfaces	پیکربندی شبکه و دستگاه ها را مشخص میکند .به عنوان مثال IP .استاتیک و اطلاعات، DHCPو غیره

دستور ifconfig

ifconfig: دستور شبکه فراموش شده که توسط دستورip جایگزین شده است. دستور ifconfig برای پیکربندی و نمایش اطلاعات مربوط به رابط های شبکه در سیستم عامل لینوکس استفاده می‌شود. این ابزار قدرتمندی است که می تواند برای تنظیم آدرسIP ، ماسک شبکه، دروازه پیش فرض، و سایر پارامترهای شبکه استفاده شود.

گزینه های های دستور:

- a اطلاعات مربوط به تمام رابط های شبکه موجود را نمایش می‌دهد.
- s خلاصه ای از اطلاعات مربوط به رابط های شبکه را نمایش می‌دهد.
- i< نام رابط> اطلاعات مربوط به رابط شبکه specific را نمایش می‌دهد.
- up رابط شبکه specific را فعال می‌کند.
- down رابط شبکه specific را غیرفعال می‌کند.
- addr< آدرسIP> آدرس IP رابط شبکه specific را تنظیم می‌کند.
- mask< ماسک شبکه> ماسک شبکه رابط شبکه specific را تنظیم می‌کند.
- gateway< دروازه پیش فرض> دروازه پیش فرض رابط شبکه specific را تنظیم می‌کند.
- mtu< اندازه واحد انتقال حداکثر> اندازه واحد انتقال حداکثر (MTU) رابط شبکه specific را تنظیم می‌کند.
- promisc رابط شبکه specific را در حالت promiscuous قرار می‌دهد.
- allmulti رابط شبکه specific را برای دریافت همه بسته های چندگانه فعال می‌کند.
- broadcast< آدرس پخش> آدرس پخش رابط شبکه specific را تنظیم می‌کند.
- h راهنمای دستور را نمایش می‌دهد.

مشاهده تمام تنظیمات شبکه

دستورifconfig بدون هیچ استدلالی تمام جزئیات اینترفیس[1] یا سخت افزار شبکه که فعال است، نمایش داده خواهد شد.دستور ifconfigهمچنین برای بررسی آدرس IP اختصاص داده شده سرور استفاده می شود.

```
$ sudo ifconfig
```

```
eth0 Link encap: اترنت HWaddr 00: 0B: CD: 1C: 18: 5A
inet addr: 172.16.25.126 Bcast: 172.16.25.63:۲۵۵,۲۵۵,۲۵۵,۲۲۴
inet6 addr: fe80 :: 20b: cdff: fe1c: 185a / 64
UP BROADCAST RUNNING MULTICAST MTU: 1500
```

اطلاعات تمام اینترفیس های شبکه را نمایش می دهد

دستورifconfig زیر با آرگومان a-اطلاعات تمام اینترفیس های شبکه فعال یا غیر فعال در سرور را نمایش می دهد.این نتایج را برای eth0، lo، sit0و tun0نمایش می دهد.

```
$ sudo ifconfig  -a
```

چگونگی فعال کردن یک رابط شبکه

سویچ " up " یا " ifup " با نام رابط (eth0) یک رابط شبکه را فعال می کند، اگر در حالت فعال نیست و اجازه ارسال و دریافت اطلاعات را می دهد.به عنوان مثال ، " ifconfig eth0 up "یا " ifup eth0 " رابط eth0را فعال کنید.

```
$ sudo ifconfig eth0
```

[1] Interface

یا

```
$ sudo ifup eth0
```

نحوه غیرفعال کردن یک رابط شبکه

سویچ " down " یا " ifdown " با نام رابط (eth0) رابط کاربری مشخص شده را غیرفعال[۱] می کند.به عنوان مثال ، اگر " ifconfig eth0 down " یا " ifdown eth0 " رابط eth0را غیرفعال کند، اگر در وضعیت فعال[۲] باشد.

```
$ sudo ifconfig eth0  down
```

یا

```
$ sudo ifdown eth0
```

نحوه اختصاص یک آدرس IP به اینترفیس شبکه

برای اختصاص یک آدرس IP به یک رابط خاص، از دستور زیر با نام رابط (eth0) و آدرس IP که می خواهید تنظیم کنید استفاده کنید.به عنوان مثال ، " ifconfig eth0 172.16.25.125 "آدرس IP را به رابط eth0تنظیم می کند.

چگونگی اختصاص یک ماسک شبکه به اینترفیس شبکه

با استفاده از دستورifconfig با بیان " netmask " و نام رابط به عنوان(eth0) ، می توانید یک mask را به یک رابط مشخص تعریف کنید.به عنوان مثال :

```
ifconfig eth0 netmask 255.255.255.224
```

ماسک شبکه را به یک رابط داده eth0تنظیم خواهد کرد.

نمایش اطلاعات مربوط به تمام رابط های شبکه:

```
ifconfig -a
```

نمایش خلاصه ای از اطلاعات مربوط به رابط های شبکه:

```
ifconfig -s
```

نمایش اطلاعات مربوط به رابط شبکه eth0:

```
ifconfig eth0
```

فعال کردن رابط شبکه eth1:

```
ifconfig eth1 up
```

غیرفعال کردن رابط شبکه eth2:

```
ifconfig eth2 down
```

تنظیم آدرس IP رابط شبکه eth3 به ۱۹۲,۱۶۸,۱,۱۰۰:

```
ifconfig eth3 192.168.1.100
```

تنظیم ماسک شبکه رابط شبکه eth4 به ۲۵۵,۲۵۵,۲۵۵,۰:

```
ifconfig eth4 netmask 255.255.255.0
```

---

[۱] disable

[۲] enable

تنظیم دروازه پیش فرض رابط شبکه eth5 به ۱۹۲,۱۶۸,۱,۱:

```
ifconfig eth5 gateway 192.168.1.1
```

تنظیم MTU رابط شبکه eth6 به ۱۵۰۰:

```
ifconfig eth6 mtu 1500
```

قرار دادن رابط شبکه eth7 در حالت promiscuous:

```
ifconfig eth7 promisc
```

فعال کردن دریافت همه بسته های چندگانه توسط رابط شبکه eth8:

```
ifconfig eth8 allmulti
```

تنظیم آدرس پخش رابط شبکه eth9 به ۲۵۵,۲۵۵,۲۵۵,۲۵۵:

```
ifconfig eth9 broadcast 255.255.255.255
```

نمایش راهنمای دستور:

```
ifconfig -h
```

دستور ethtool

دستور ethtool در سیستم‌عامل‌های Unix/Linux برای پیکربندی و نمایش اطلاعات مربوط به تنظیمات و ویژگی‌های کارت شبکه Ethernet استفاده می‌شود. این دستور به کاربر این امکان را می‌دهد تا وضعیت و تنظیمات کارت شبکه خود را بررسی و تنظیم کند. همچنین دستور ethtool ابزاری قدرتمند برای مدیریت و پیکربندی رابط‌های شبکه در سیستم‌عامل لینوکس است. این ابزار اطلاعات مختلفی راجع به کارت شبکه، مانند سرعت، نوع اتصال، و آمار مربوط به ترافیک را نمایش می‌دهد. همچنین می‌توان از ethtool برای تنظیم پارامترهای مختلف کارت شبکه، مانند سرعت، duplex، وoffload ، استفاده کرد.

گزینه های دستور:

- h نمایش راهنمای دستور.
- i نمایش اطلاعات کلی رابط شبکه.
- s نمایش تنظیمات رابط شبکه.
- a نمایش آمار رابط شبکه.
- g نمایش اطلاعات مربوط به گروه‌های بندی کانال (فقط برای رابط‌های ۳,۸۰۲.ad).
- x نمایش اطلاعات مربوط به تنظیمات فلوی کنترل (فقط برای رابط‌های ۳,۸۰۲.x).
- k نمایش اطلاعات مربوط به تنظیمات (offloadفقط برای رابط‌های ۱,۸۰۲.Q).
- r تنظیمات رابط شبکه را به مقادیر پیش‌فرض باز می‌گرداند.
- S تنظیمات رابط شبکه را از یک فایل مشخصه بارگذاری می‌کند.
- c تنظیمات رابط شبکه را به یک فایل مشخصه ذخیره می‌کند.
- p تنظیمات PHY رابط شبکه را نمایش می‌دهد.
- n تنظیمات PHY رابط شبکه را تنظیم می‌کند.
- t تست‌های مختلفی را برای رابط شبکه اجرا می‌کند.
- D اطلاعات مربوط به درایور رابط شبکه را نمایش می‌دهد.
- M اطلاعات مربوط به firmware رابط شبکه را نمایش می‌دهد.
- u اطلاعات مربوط به EEPROM رابط شبکه را نمایش می‌دهد.

- E اطلاعات مربوط به NVRAM رابط شبکه را نمایش می‌دهد.
- P اطلاعات مربوط به پروتکل‌های رابط شبکه را نمایش می‌دهد.
- V اطلاعات مربوط به نسخه ethtool را نمایش می‌دهد.

مثال های دستور:

نمایش اطلاعات کلی راجع به کارت شبکه eth0:

```
ethtool -i eth0
```

نمایش اطلاعات مربوط به تنظیمات فعلی کارت شبکه eth1:

```
ethtool -g eth1
```

تنظیم سرعت کارت شبکه eth2 به ۱۰۰۰ مگابیت بر ثانیه:

```
ethtool -s eth2 speed 1000
```

نمایش اطلاعات مربوط به آمار ترافیک کارت شبکه eth3:

```
ethtool -d eth3
```

فعال کردن offload TCP checksum برای کارت شبکه eth4:

```
ethtool -k eth4 tx offload tso on
```

نمایش اطلاعات EEPROM کارت شبکه eth5:

```
ethtool -x eth5
```

تست کارت شبکه eth6:

```
ethtool -p eth6
```

تنظیم timestamping برای کارت شبکه eth7:

```
ethtool -T eth7 rx on tx on
```

تنظیم RSS برای کارت شبکه eth8:

```
ethtool -r eth8 all on
```

تنظیم LRO برای کارت شبکه eth9:

```
ethtool -L eth9 on
```

تنظیم RX/TX checksum offload برای کارت شبکه eth10:

```
ethtool -U eth10 on
```

تنظیم MTU برای کارت شبکه eth11 به ۱۵۰۰ بایت:

```
ethtool -M eth11 1500
```

نمایش راهنمای دستور:

```
ethtool -h
```

دستور addresses

دستور addresses برای نمایش آدرس های IP یک رابط شبکه specific یا همه رابط های شبکه در سیستم عامل لینوکس استفاده می‌شود. این ابزار می‌تواند برای عیب یابی مشکلات شبکه، پیکربندی شبکه و یا مشاهده اطلاعات مربوط به اتصالات شبکه شما مفید باشد.

ساختار دستور:

```
$ addresses [-p <port>]
```

گزینه های دستور:

- a آدرس های IP همه رابط های شبکه را نمایش می‌دهد.
- i آدرس های IP رابط شبکه specific را با نام مشخص شده نمایش می‌دهد.
- f آدرس های IP را به صورت خلاصه نمایش می‌دهد.
- n آدرس های IP را به صورت عددی نمایش می‌دهد.
- ۴ فقط آدرس های IP نسخه ۴ را نمایش می‌دهد.
- ۶ فقط آدرس های IP نسخه ۶ را نمایش می‌دهد.
- h راهنمای دستور را نمایش می‌دهد.

مثال های دستور:

دستور addresses شما امکان می‌دهد تا آدرس‌های مرتبط با یک پورت خاص را ببینید. در زیر مثالی از استفاده از این دستور آورده شده است:

```
addresses -p 8080
```

در این مثال، دستور addressesبا استفاده از گزینه p و مقدار `۸۰۸۰` فقط آدرس‌هایی را نمایش خواهد داد که مرتبط با پورت ۸۰۸۰ هستند. شما می‌توانید پورت مورد نظر خود را با استفاده از این گزینه مشخص کنید. با کاراکتر '>' (بزرگتر از) هدایت میشود به یک فایل انتخابی در مسیرمشخص شما

```
$ addresses [-p <port>] > MyAddresses.txt
$ addresses [-p <port>]                    # print to STDOUT/the screen
```

نمایش آدرس های IP همه رابط های شبکه:

```
addresses -a
```

نمایش آدرس های IP رابط شبکه با نام eth0:

```
addresses -i eth0
```

نمایش آدرس های IP به صورت خلاصه:

```
addresses -f
```

نمایش آدرس های IP به صورت عددی:

```
addresses -n
```

نمایش فقط آدرس های IP نسخه ۴:

```
addresses -4
```

نمایش فقط آدرس های IP نسخه ۶:

```
addresses -6
```

نمایش راهنمای دستور:

```
addresses -h
```

دستور ip

دستور ip در net-tools وجود دارد که برای انجام چندین کار مدیریت شبکه استفاده می شود. IP مخفف Internet Protocol است. این دستور برای نمایش یا دستکاری مسیریابی، دستگاه ها و تونل ها استفاده می شود. می‌تواند کارهایی مانند پیکربندی و اصلاح

مسیریابی پیش‌فرض و ثابت، راه اندازی تونل روی IP، فهرست کردن آدرس‌های IP و اطلاعات دارایی، تغییر وضعیت رابط، تخصیص، حذف و تنظیم آدرس‌ها و مسیرهای IP را انجام دهد. همچنین دستور ip ابزاری قدرتمند در سیستم عامل لینوکس برای پیکربندی و مدیریت رابط های شبکه، آدرس های IP و روتینگ است. این دستور جایگزینی برای دستورات قدیمی تر مانند ifconfig و route است و امکانات و انعطاف پذیری بیشتری را ارائه می دهد.

ساختار دستور:

```
$ ip [ OPTIONS ] OBJECT { COMMAND | help }
```

گزینه های دستور:

تنظیمات رابط

- a نمایش اطلاعات مربوط به همه رابط های شبکه.
- l نمایش اطلاعات مربوط به رابط های شبکه به صورت لیست.
- s نمایش آمار مربوط به رابط های شبکه.
- u رابط شبکه را فعال می کند.
- d رابط شبکه را غیرفعال می کند.
- up رابط شبکه را بالا می آورد.
- down رابط شبکه را پایین می آورد.
- link set وضعیت لینک رابط شبکه را تنظیم می کند.
- address add آدرس IP را به رابط شبکه اضافه می کند.
- address del آدرس IP را از رابط شبکه حذف می کند.

تنظیمات روتینگ

- route add مسیر را به جدول روتینگ اضافه می کند.
- route del مسیر را از جدول روتینگ حذف می کند.
- rule add قانون را به جدول روتینگ اضافه می کند.
- rule del قانون را از جدول روتینگ حذف می کند.

سایر گزینه ها

- h نمایش راهنمای دستور.
- V نمایش نسخه دستور.

مثال های دستور:

نمایش نسخه ip گزینه : -V

```
$ ip -V
```

نمایش راهنمای دستور ip گزینه : -h

```
$ ip -h
```

نمایش آمار شبکه گزینه : s– x

```
$ ip -s
```

نمایش جزئیات بیشتر گزینه : -d

```
$ ip -d
```

رزولوشن آدرس‌ها به نام گزینه : -r

```
$ ip -r
```

استفاده از واحدهای IEC برای نمایش گزینه : -iec

```
$ ip -iec
```

استفاده از خانواده آدرس IPv4 گزینه :-f inet x

```
$ ip -f inet
```

استفاده از فرمت JSON برای خروجی گزینه : -j

```
$ ip -j
```

نمایش اطلاعات در یک خط گزینه : -o

```
$ ip -o
```

نمایش اطلاعات به صورت خلاصه گزینه : -br

```
$ ip -br
```

گزینه : -4 نمایش فقط آدرس‌هایIPv4

```
$ ip -4 address show
```

گزینه : -6 نمایش فقط آدرس‌هایIPv6

```
$ ip -6 address show
```

گزینه : -B نمایش جداول پهنای باند

```
$ ip -B link show
```

گزینه : -0 نمایش جداول پهنای باند در حالت صفر که نشان دهنده برداشتن پهنای باند است

```
$ ip -0 link show
```

گزینه : -l maximum-addr-flush-attempts تعیین تعداد بیشینه تلاش‌ها برای حذف آدرس‌ها در حالت oops

```
$ ip -l 3 addrflush
```

گزینه : -o نمایش خروجی در یک خط

```
$ ip -o link show
```

گزینه : -rcvbuf [size] تنظیم اندازه بافر ورودی جایگزین برای سوکت‌های rcvbuf

```
$ ip -rcvbuf 8192 link show
```

گزینه : -t نمایش زمان در خروجی

```
$ ip -t link show
```

گزینه : -ts نمایش زمان به صورت مختصر در خروجی

```
$ ip -ts link show
```

گزینه : -n name انتقال دستور به فضای نام مشخص شده

```
$ ip -n namespace link show
```

گزینه : -N نمایش آدرس‌ها به صورت عددی

```
$ ip -N address show
```

گزینه : -a نمایش همه موارد

```
$ ip -a link show
```

گزینه : -c استفاده از رنگ در خروجی

```
$ ip -c link show
```

گزینه : -br نمایش خروجی به صورت خلاصه

```
$ ip -br link show
```

گزینه : -j نمایش خروجی به صورت JSON

```
$ ip -j link show
```

گزینه : -p نمایش خروجی به صورت زیبا

```
$ ip -p link show
```

گزینه : link نمایش و مدیریت اطلاعات مربوط به لینک‌های شبکه

```
$ ip link show
$ ip link set dev eth0 up
```

گزینه : address نمایش و مدیریت اطلاعات آدرس‌های IP

```
$ ip address show
$ ip address add 192.168.1.10/24 dev eth0
```

گزینه : addrlabel نمایش و مدیریت برچسب‌های آدرس

```
$ ip addrlabel show
$ ip addrlabel add prefix 192.168.1.0/24 label 10
```

گزینه : route نمایش و مدیریت جداول مسیریابی

```
$ ip route show
$ ip route add default via 192.168.1.1 dev eth0
```

گزینه : rule نمایش و مدیریت قوانین مسیریابی

```
$ ip rule show
$ ip rule add from 192.168.1.0/24 table 10
```

گزینه : neigh نمایش و مدیریت جداول همسایگی (ARP/NDP)

```
$ ip neigh show
$ ip neigh add 192.168.1.1 lladdr 00:11:22:33:44:55 nud permanent dev eth0
```

گزینه : ntable نمایش و مدیریت جداول جاری شبکه

```
$ ip ntable show
$ ip ntable add table 10 type fib num 100
```

گزینه : tunnel نمایش و مدیریت تونل‌های شبکه

```
$ ip tunnel show
$ ip tunnel add tun0 mode gre remote 192.168.1.1 local 192.168.1.2
```

گزینه : tuntap نمایش و مدیریت دستگاه‌های TUN/TAP

```
$ ip tuntap show
$ ip tuntap add name tun0 mode tun
```

گزینه : maddress نمایش و مدیریت آدرس‌های چندگانه

```
$ ip maddress show
$ ip maddress add 239.0.0.1 dev eth0
```

گزینه : mroute نمایش و مدیریت مسیرهای چندگانه

```
$ ip mroute show
$ ip mroute add 239.0.0.1 dev eth0
```

گزینه : mrule نمایش و مدیریت قوانین چندگانه

```
$ ip mrule show
$ ip mrule add from 192.168.1.0/24 table 10
```

گزینه : monitor نمایش وضعیت مانیتورینگ

```
$ ip monitor
```

گزینه : xfrm نمایش و مدیریت امنیت و تونل‌ها

```
$ ip xfrm state show
$ ip xfrm policy add src 192.168.1.0/24 dst 10.0.0.0/24 dir out
```

گزینه : netns نمایش و مدیریت فضاهای نام شبکه

```
$ ip netns list
$ ip netns add mynamespace
```

گزینه : l2tp نمایش و مدیریت تونل‌های L2TP

```
$ ip l2tp show tunnel
$ ip l2tp add tunnel local 192.168.1.1 remote 192.168.2.1
```

گزینه : tcp_metrics نمایش و مدیریت معیارهای TCP

```
$ ip tcp_metrics show
$ ip tcp_metrics add 192.168.1.0/24 rtt 100ms
```

گزینه : token نمایش و مدیریت توکن‌ها

```
$ ip token show
$ ip token add dev eth0
```

گزینه : macsec نمایش و مدیریت امنیت MACsec

```
$ ip macsec show
$ ip macsec add dev eth0
```

گزینه : vrf نمایش و مدیریت - منابع مجازی شبکه

```
$ip vrf show
$ip vrf add myvrf
```

گزینه : mptcp نمایش و مدیریت - چندگانه پروتکل کنترل انتقال

```
$ip mptcp show
$ip mptcp add subflow dev eth0
```

گزینه : ioam نمایش و مدیریت –مانیتورینگ آگاه از مسیر

```
$ip ioam show
$ip ioam add hoplimit 64
```

گزینه : stats نمایش آمارهای شبکه

```
$ip -s link show
$ip -s route show
```

دستور fail2ban

شرح دستور

دستور fail2banابزاری قدرتمند برای افزایش امنیت سیستم‌های لینوکسی است .این ابزار با نظارت بر لاگ‌های سیستم، تلاش‌های ناموفق برای ورود به سیستم یا سرویس‌های دیگر را شناسایی کرده و آدرس‌های IP متخلف را به طور موقت مسدود می‌کند .به عبارت دیگر، fail2banیک سیستم پیشگیری از نفوذ مبتنی بر قوانین است که به صورت خودکار واکنش نشان می‌دهد.

ساختار دستور

```
fail2ban [options]
```

کاربردهای دستور

- **مسدود کردن آدرس‌های IP متخلف :**با شناسایی تلاش‌های ناموفق برای ورود به سیستم یا سرویس‌های دیگر، آدرس‌های IP متخلف را به طور موقت مسدود می‌کند.
- **افزایش امنیت سیستم :**با کاهش تعداد حملات موفق، امنیت کلی سیستم را افزایش می‌دهد.
- **انعطاف‌پذیری بالا :**با استفاده از فایل‌های پیکربندی، می‌توان قوانین مسدودسازی را به راحتی سفارشی‌سازی کرد.
- **پشتیبانی از پروتکل‌ها و سرویس‌های مختلف** fail2ban **:**از طیف گسترده‌ای از پروتکل‌ها و سرویس‌ها مانندSSH ، FTP، HTTPو ... پشتیبانی می‌کند.

گزینه‌های کامل دستور و مثال‌ها

- **b:-**دستور fail2ban را در پس‌زمینه اجرا می‌کند.

```
fail2ban -b
```

- **c FILE:-**فایل پیکربندی مشخص شده را برای بارگذاری تنظیمات استفاده می‌کند.

```
fail2ban -c /etc/fail2ban/jail.local
```

- **p FILE:-**فایل قفل PID را در مکان مشخص شده ایجاد می‌کند.

```
fail2ban -p /var/run/fail2ban.pid
```

- **h, --help:-**راهنمای استفاده از دستور را نمایش می‌دهد.

```
fail2ban -h
```

- **i IP:-**آدرس‌های IP مشخص شده را از مسدودسازی مستثنی می‌کند.

```
fail2ban -i 192.168.1.100
```

- **k:-**فرآیند fail2ban در حال اجرا را متوقف می‌کند.

```
fail2ban -k
```

- **r VALUE:-**حداکثر تعداد تلاش ناموفق برای ورود به سیستم را قبل از مسدودسازی تعیین می‌کند.

```
fail2ban -r 5
```

- **t TIME:-**مدت زمان مسدودسازی آدرس‌های IP متخلف را بر حسب ثانیه تعیین می‌کند.

```
fail2ban -t 3600
```

- **f TIME:-**مدت زمانی که یک تلاش ناموفق در نظر گرفته می‌شود را بر حسب ثانیه تعیین می‌کند.

```
fail2ban -f 10
```

- **v:-**حالت مفصل را فعال می‌کند .با استفاده از دو بار v-اطلاعات بیشتری نمایش داده می‌شود.

```
fail2ban -vv
```

- **V, --version:-**نسخه نرم‌افزار fail2ban را نمایش می‌دهد.

```
fail2ban -V
```

مثال های کاربردی

- **شروع fail2ban با استفاده از فایل پیکربندی سفارشی:**

```
fail2ban -c /etc/fail2ban/jail.local -b
```

- **مسدود کردن آدرس‌های IP متخلف برای مدت ۱ ساعت و اجازه ۳ تلاش ناموفق:**

```
fail2ban -t 3600 -r 3
```

- **نمایش اطلاعات مفصل در مورد اجرا:**

```
fail2ban -vv
```

نکته‌های مهم

- **فایل‌های پیکربندی :**فایل‌های پیکربندی fail2ban برای تعریف قوانین مسدودسازی بسیار مهم هستند.
- **جایگذاری :**معمولاً فایل‌های پیکربندی در دایرکتوری `/etc/fail2ban`قرار دارند.
- **پروتکل‌ها** fail2ban :از طیف گسترده‌ای از پروتکل‌ها و سرویس‌ها پشتیبانی می‌کند.
- **سفارشی‌سازی :**با ویرایش فایل‌های پیکربندی، می‌توان قوانین مسدودسازی را به طور کامل سفارشی‌سازی کرد.

دستور ufw

دستور ufw در سیستم‌عامل‌های Ubuntu و دیگر توزیع‌های Linux برای مدیریت فایروال ((firewallاستفاده می‌شود. این دستور به کاربر امکان می‌دهد تا قوانین و تنظیمات مربوط به فایروال را تعریف و مدیریت کند تا امنیت سیستم خود را بهبود دهد. همچنین دستور ufw ((مخففUncomplicated Firewall) برای پیکربندی فایروال در سیستم عامل های لینوکس مبتنی بر iptables به صورت آسان و کاربرپسند استفاده می‌شود. این ابزار به شما امکان می‌دهد تا قوانین فایروال را به صورت ساده و بدون نیاز به دانش عمیق iptablesایجاد و مدیریت کنید.

گزینه های دستور:

- h نمایش راهنمای دستور
- v نمایش نسخه ufw
- f فعال کردن ufw
- d غیرفعال کردن ufw
- s نمایش وضعیت فعلی ufw
- a اضافه کردن یک قانون جدید
- i اضافه کردن یک قانون جدید به ابتدای لیست قوانین
- d حذف یک قانون
- r جایگزینی یک قانون
- l نمایش لیست قوانین
- n نمایش شماره خط قوانین
- c پاک کردن تمام قوانین
- p پروتکل مورد نظر برای قانون (tcp, udp, icmp, etc.)
- m ماژول iptables مورد نظر برای قانون (state, limit, etc.)
- s آدرس IP مبدا
- d آدرس IP مقصد
- sport پورت مبدا

- dport پورت مقصد
- t نام جدول iptables برای قانون
- j عمل مورد نظر برای قانون (ACCEPT, DROP, REJECT, etc.)
- نام رابط شبکه برای قانون
- v تنظیم سطح verbosity

مثال های دستور:

فعال کردن ufw:

```
ufw enable
```

غیرفعال کردن ufw:

```
ufw disable
```

نمایش وضعیت فعلی ufw:

```
ufw status
```

اضافه کردن قانون جدید برای اجازه ورود به پورت ۲۲ SSH:

```
ufw allow 22
```

اضافه کردن قانون جدید به ابتدای لیست قوانین برای اجازه ورود به پورت ۸۰ HTTP:

```
ufw insert 1 allow 80
```

حذف قانون اجازه ورود به پورت ۲۵ SMTP:

```
ufw delete allow 25
```

جایگزینی قانون اجازه ورود به پورت ۸۰ HTTP با قانون جدیدی که فقط از آدرس IP 192.168.1.0/24 اجازه ورود می‌دهد:

```
ufw replace allow 80 from 192.168.1.0/24
```

نمایش لیست قوانین:

```
ufw list
```

نمایش لیست قوانین با شماره خط:

```
ufw list -n
```

پاک کردن تمام قوانین:

```
ufw reset
```

اجازه ورود به تمام ترافیک از رابط شبکه eth0:

```
ufw allow out on eth0
```

مسدود کردن تمام ترافیک به پورت ۲۵ SMTP:

```
ufw reject 25
```

نمایش راهنمای کامل:

```
ufw --help
```

دستور iptables

دستور iptables برای تنظیم و نگهداری جداول فایروال Netfilter برای IPv4، موجود در هسته لینوکس استفاده می شود. فایروال بسته ها را با قوانین تعریف شده در این جداول مطابقت می‌دهد و سپس اقدام مشخص شده را روی یک تطابق احتمالی انجام می‌دهد.

ساختار دستور:

```
$ iptables --table TABLE -A/-C/-D... CHAIN rule --jump Target
```

```
$ iptables [-t table] --append [chain] [parameters]
```

گزینه های دستور:

- h نمایش راهنمای دستور
- v نمایش نسخه iptables
- L نمایش لیست قوانین
- t نام جدول iptables برای اعمال قوانین
- A اضافه کردن یک قانون جدید
- I اضافه کردن یک قانون جدید به ابتدای لیست قوانین
- D حذف یک قانون
- R جایگزینی یک قانون
- C بررسی وجود یک قانون
- F پاک کردن تمام قوانین
- X حذف تمام زنجیره ها
- Z صفر کردن شمارنده بسته ها و بایت ها
- n نمایش آدرس های IP به صورت عددی
- p پروتکل مورد نظر برای قانون (tcp, udp, icmp, etc.)
- m ماژول iptables مورد نظر برای قانون (state, limit, etc.)
- s آدرس IP مبدا
- d آدرس IP مقصد
- sport پورت مبدا
- dport پورت مقصد
- j عمل مورد نظر برای قانون (ACCEPT, DROP, REJECT, etc.)
- i نام رابط ورودی برای قانون

مثال های دستور:

۱. نمایش لیست قوانین:

```
iptables -L
```

۲. نمایش لیست قوانین در جدول nat:

```
iptables -t nat -L
```

۳. اضافه کردن قانون جدید برای اجازه ورود به پورت ۲۲ SSH:

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

۴. اضافه کردن قانون جدید به ابتدای لیست قوانین برای اجازه ورود به پورت ۸۰ HTTP:

```
iptables -I INPUT 1 -p tcp --dport 80 -j ACCEPT
```

۵. حذف قانون اجازه ورود به پورت ۲۵ SMTP:

```
iptables -D INPUT -p tcp --dport 25 -j ACCEPT
```

۶. جایگزینی قانون اجازه ورود به پورت ۸۰ HTTP با قانون جدیدی که فقط از آدرس IP 192.168.1.0/24 اجازه ورود می‌دهد:

```
iptables -R INPUT 1 -p tcp --dport 80 -s 192.168.1.0/24 -j ACCEPT
```

۷. بررسی وجود قانون اجازه ورود به پورت ۸۰ HTTP:

```
iptables -C INPUT -p tcp --dport 80 -j ACCEPT
```

۸. پاک کردن تمام قوانین در جدول nat:

```
iptables -t nat -F
```

۹. حذف تمام زنجیره ها در جدول filter:

```
iptables -t filter -X
```

۱۰. صفر کردن شمارنده بسته ها و بایت ها در جدول mangle:

```
iptables -t mangle -Z
```

۱۱. اجازه ورود به تمام ترافیک از رابط شبکه eth0:

```
iptables -A INPUT -i eth0 -j ACCEPT
```

۱۲. مسدود کردن تمام ترافیک به پورت ۲۵ SMTP:

```
iptables -A INPUT -p tcp --dport 25 -j DROP
```

۱۳. نمایش جزئیات بیشتر در مورد قانون سوم:

```
iptables -L -n -v 3
```

۱۴. اضافه کردن قانون جدید برای اجازه ورود به پورت ۸۰ HTTP فقط از طریق رابط شبکه eth1:

```
iptables -A INPUT -i eth1 -p tcp --dport 80 -j ACCEPT
```

۱۵. اضافه کردن قانون جدید برای رد تمام ترافیک از آدرس IP 192.168.1.100:

```
iptables -A INPUT -s 192.168.1.100 -j REJECT
```

۱۶. محدود کردن تعداد اتصالات به پورت ۲۲ SSH به ۱۰ اتصال در هر ثانیه:

```
iptables -A INPUT -p tcp --dport 22 -m limit --limit 10/s -j ACCEPT
```

۱۷. ذخیره قوانین iptables در یک فایل:

```
iptables-save > my-iptables-rules.conf
```

۱۸. بارگیری قوانین iptables از یک فایل:

```
iptables-restore < my-iptables-rules.conf
```

۱۹. ایجاد یک زنجیره جدید به نام " "custom در جدول filter:

```
iptables -t filter -N custom
```

۲۰. اضافه کردن قانون جدید به زنجیره " "custom برای اجازه ورود به پورت ۸۰ HTTP:

```
iptables -t filter -A custom -p tcp --dport 80 -j ACCEPT
```

۲۱. پرش به زنجیره " "custom از زنجیره INPUT:

```
iptables -t filter -A INPUT -j custom
```

۲۲. رد تمام ترافیک ICMP:

```
iptables -A INPUT -p icmp -j DROP
```

۲۳. اجازه عبور ترافیک ICMP مربوط به echo request و echo reply:

```
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

۲۴. محدود کردن ترافیک ورودی به ۱ مگابیت بر ثانیه:

```
iptables -A INPUT -m limit --limit 1m -j ACCEPT
```

۲۵. رد ترافیک از آدرس IP 192.168.1.100 به پورت ۲۲ SSH:

```
iptables -A INPUT -s 192.168.1.100 -p tcp --dport 22 -j REJECT
```

۲۶. اجازه ورود به پورت ۸۰ HTTP فقط برای آدرس های IP در محدوده ۱۹۲,۱۶۸,۱,۰/۲۴:

```
iptables -A INPUT -p tcp --dport 80 -s 192.168.1.0/24 -j ACCEPT
```

۲۷. مسدود کردن تمام ترافیک به جز ترافیک مربوط به SSH و HTTP:

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -j DROP
```

دیگر ابزارهای شبکه

ابزارهای خط فرمان لینوکس برای مدیریت شبکه بسیار قدرتمند و انعطاف‌پذیر هستند .این ابزارها به شما اجازه می‌دهند تا ترافیک شبکه را تحلیل، مشکلات را عیب‌یابی و تغییرات پیکربندی را انجام دهید .در ادامه به بررسی برخی از این ابزارها و دستورهای پرکاربرد آن‌ها می‌پردازیم.

Tcmpdump - ابزار ضبط و تجزیه و تحلیل بسته دستور برای نظارت بر ترافیک شبکه است.

Netstat-ی‌ک ابزار نظارت بر شبکه خط دستور باز است که ترافیک بسته های شبکه های ورودی و خروجی را نظارت می کند.

Wireshark-یک تحلیلگر پروتکل پروتکل باز است که برای رفع مشکلات مربوط به شبکه استفاده می شود.

Munin-یک برنامه نظارت بر شبکه و system وب است که برای نمایش نتایج در نمودار با استفاده از rrdtool استفاده می شود.

Cacti-برنامه نظارت و برنامه نویسی کامل وب برای نظارت بر شبکه است.

تعریف ip ثابت

برای تعریف **statick ip address**شما باید فایل کانفیگ شبکه خود را آپدیت یا ویرایش کنید تا بتوانید یک ip static به سیستم assignکنید. برای این منظور شما باید super user باشید.

سیستم عامل های: RHEL /Fedora

کانفیگ فایل مورد نظر برای اینترفیس خود eth0 یا eth1 را با استفاده از ویرایشگر مورد علاقه تان باز کنید. برای مثال ما با دستور زیر میخواهیم به اینترفیس eth0 مان ip بدهیم:

کد

```
[root@linuxzone ~]# vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

خروجی دستور:

```
DEVICE="eth0"
BOOTPROTO=static
ONBOOT=yes
TYPE="Ethernet"
IPADDR=192.168.50.2
NAME="System eth0"
HWADDR=00:0C:29:28:FD:4C
GATEWAY=192.168.50.1
```

با خط های بالا که درون فایل وارد میکنیم میتوانیم ip و دیگر اطلاعات مورد نظرمان را به صورت static وارد کنیم.

سیستم عامل های: Ubuntu/Debian/Linux Mint

برای ip دادن به اینترفیس eth0 باید فایل کانفیگ آن را که در مسیر **/etc/network/interfaces** میباشد و ویرایش کنیم و پارامترهای آن را همانند مثال زیر تغییر دهیم:

کد

```
auto eth0 iface eth0 inet static address 192.168.50.2 netmask 255.255.255.0 gateway 192.168.50.1
```

اکنون سرویس network را بعد از وارد کردن همه جزئیات گفته شده با استفاده از دستور زیر restart میکنیم:

```
$ /etc/init.d/networking restart
```

کد

```
$ sudo /etc/init.d/networking restart
```

نحوه تخصیص یک ip address به اینترفیس مورد نظر:

با استفاده از دستور زیر میتوانیم ip address را به یک اینترفیس خاص (eth1) خیلی سریع اختصاص دهیم:

```
$ ip addr add 192.168.50.5 dev eth1
```

کد

```
$ sudo ip addr add 192.168.50.5 dev eth1
```

توجه : متاسفانه همه تنظیمات بالا بعد از یک بار restart سیستم از بین خواهد رفت و در حقیقت موقت میباشد.

نحوه چک کردن: ip address

برای دریافت اطلاعات کامل از اینترفیس های شبکه تان مانند جزییات ip address و mac address دستور زیر را استفاده میکنیم:

```
$ ip addr show
```

کد

```
$ sudo ip addr show
```

خروجی دستور:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen
1000
    link/ether 00:0c:29:28:fd:4c brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.2/24 brd 192.168.50.255 scope global eth0
    inet6 fe80::20c:29ff:fe28:fd4c/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen
1000
    link/ether 00:0c:29:28:fd:56 brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.5/24 scope global eth1
    inet6 fe80::20c:29ff:fe28:fd56/64 scope link
```

نحوه حذف یک: ip address

دستور زیر ip address داده شده به یک اینترفیس (eth1) را حذف خواهد کرد:

```
$ ip addr del 192.168.50.5/24 dev eth1
```

کد

```
$ sudo ip addr del 192.168.50.5/24 dev eth1
```

نحوه فعال کردن اینترفیس شبکه

آپشن up به همراه نام اینترفیس (eth1) اینترفیس شبکه را فعال میکند. برای مثال دستور زیر اینترفیس eth1 را فعال مینماید:

```
$ ip link set eth1 up
```

کد

```
$ sudo ip link set eth1 up
```

نحوه غیرفعال کردن اینترفیس شبکه

گزینه down به همراه نام اینترفیس (eth1) اینترفیس شبکه را غیرفعال میکند. برای مثال دستور زیر اینترفیس eth1 را غیرفعال مینماید:

```
$ ip link set eth1 down
```

کد

```
$ sudo ip link set eth1 down
```

چگونه route table را بررسی کنم؟

برای چک کردن اطلاعات routing table سیستم دستور زیر را اجرا کنید:

```
$ ip route show
```

کد

```
$ sudo ip route show
```

خروجی دستور:

```
10.10.20.0/24 via 192.168.50.100 dev eth0
192.168.160.0/24 dev eth1  proto kernel link  src 192.168.160.130  metric 1
192.168.50.0/24 dev eth0  proto kernel  scope link  src 192.168.50.2
169.254.0.0/16 dev eth0  scope link  metric 1002
default via 192.168.50.1 dev eth0  proto static
```

نحوه اضافه کردن: static route

دلیل اینکه چرا شما نیاز به اضافه کردن static route ها یا manual route یا route ها دستی به این خاطر است که ترافیک نباید از طریق default gateway عبور کند. ما باید static route اضافه کنیم تا ترافیک را از بهترین راه عبور دهیم تا به مقصد برسد:

```
$ ip route add 10.10.20.0/24 via 192.168.50.100 dev eth0
```

کد:

```
$ sudo ip route add 10.10.20.0/24 via 192.168.50.100 dev eth0
```

نحوه حذف: static route

برای حذف static route اختصاص داده شده میتوانید به اسانی با اجرای دستور زیر این کار را انجام دهید:

```
# ip route del 10.10.20.0/24
```

کد:

```
$ sudo ip route del 10.10.20.0/24
```

نحوه اضافه کردن static route های دائمی

همه روت های بالا بعد از یک بار restart سیستم از بین میرود. برای اضافه کردن static route به صورت دائمی فایل **etc/sysconfig/network-scripts/route-eth0**/ویرایش میکنیم (.ما static route مربوط به eth0 را جمع آوری کرده و خطوط زیر را اضافه و سپس save و خارج میشویم).به صورت پیش فرض فایل route-eth0 وجود ندارد و باید ایجاد گردد.

سیستم عامل های: RHEL/CentOS/Fedora

```
# vi /etc/sysconfig/network-scripts/route-eth0
10.10.20.0/24 via 192.168.50.100 dev eth0
```

سیستم عامل های: Ubuntu/Debian/Linux Mint

فایل **etc/network/interfaces**/را باز کنید و در انتهای آن route های static دائمی را اضافه کنید. در سیستم شما ip addressمتفاوت خواهد بود:

```
$ sudo vi /etc/network/interfaces
```

کد:

```
auto eth0
iface eth0 inet static
address 192.168.50.2
netmask 255.255.255.0
gateway 192.168.50.100
{Static Route}##
up ip route add 10.10.20.0/24 via 192.168.50.100 dev eth0
```

سپس سرویس network را بعد از وارد کردن جزییات بالا با استفاده از دستور زیر restart میکنیم:

```
# /etc/init.d/network restart
```

کد:

```
$ sudo /etc/init.d/network restart
```

نحوه اضافه کردن: default gateway

default gatewayمیتواند یا به صورت کلی و یا به در یک کانفیگ فایل اینترفیس خاص مشخص شود. مزایایی default gatewayدر اینجاست که اگر ما بیشتر از یک اینترفیس شبکه روی سیستم موجود داشته باشیم میتوانیم تنها برای یکی default gatewayتعریف کنیم. شما میتوانید سریعا و با دستور زیر default gateway تعریف کنید:

```
# ip route add default via 192.168.50.100
```

کد:

```
$ sudo ip route add default via 192.168.50.100
```

دستور arp

دستور arp حافظه پنهان سیستم را دستکاری می کند .همچنین امکان تخلیه کامل حافظه پنهان ARP را فراهم می کند . ARP مخفف Protocol Resolution Protocol است .عملکرد اصلی این پروتکل حل و فصل آدرس IP سیستم به آدرس Mac آن است و از این رو بین سطح ۲ (لایه پیوند داده) و سطح ۳ (لایه شبکه) کار می کند.

ساختار دستور:

```
arp [-v] [-i if] [-H type] -a [نام میزبان]
```

گزینه های دستور:

- h نمایش راهنمای دستور
- v نمایش نسخه arp
- a نمایش تمام ورودی های جدول ARP
- n نمایش آدرس های IP به صورت عددی
- d حذف یک ورودی از جدول ARP
- s اضافه کردن یک ورودی جدید به جدول ARP
- i نام رابط شبکه برای نمایش یا اضافه کردن ورودی
- f نام فایل برای ذخیره یا بازیابی اطلاعات ARP
- t نوع آدرس سخت افزاری (ethernet, etc.)
- p نوع پروتکل (ip, etc.)
- m نمایش اطلاعات MAC به صورت Manufacturer Address

مثال های دستور:

۱. نمایش تمام ورودی های جدول ARP:

```
arp -a
```

۲. نمایش آدرس IP 192.168.1.100 به صورت عددی:

```
arp -n 192.168.1.100
```

۳. حذف ورودی مربوط به آدرس IP 192.168.1.101:

```
arp -d 192.168.1.101
```

۴. اضافه کردن ورودی جدید برای آدرس IP 192.168.1.102 با آدرس MAC 00:11:22:33:44:55:

```
arp -s 192.168.1.102 00:11:22:33:44:55
```

۵. نمایش اطلاعات ARP برای رابط شبکه eth0:

```
arp -i eth0
```

۶. ذخیره اطلاعات ARP در فایل arp.txt:

```
arp -a > arp.txt
```

۷. بازیابی اطلاعات ARP از فایل arp.txt:

```
arp -f arp.txt
```

۸. نمایش فقط آدرس های MAC به صورت Manufacturer Address:

```
arp -a -m
```

۹. نمایش اطلاعات ARP برای آدرس IP 192.168.1.103 با نوع پروتکل IPv6:

```
arp -a -n -p ipv6 192.168.1.103
```

۱۰. اضافه کردن ورودی جدید برای آدرس IP 192.168.1.104 با آدرس MAC 00:11:22:33:44:55 و نوع آدرس سخت افزاری Infiniband:

```
arp -s 192.168.1.104 00:11:22:33:44:55 -t infiniband
```

مثالهای بیشتر:

۱. نمایش تمام ورودی های جدول ARP:

```
arp -a
```

۲. نمایش آدرس IP 192.168.1.100 به صورت عددی:

```
arp -n 192.168.1.100
```

۳. حذف ورودی مربوط به آدرس IP 192.168.1.101:

```
arp -d 192.168.1.101
```

۴. اضافه کردن ورودی جدید برای آدرس IP 192.168.1.102 با آدرس MAC 00:11:22:33:44:55:

```
arp -s 192.168.1.102 00:11:22:33:44:55
```

۵. نمایش اطلاعات ARP برای رابط شبکه eth0:

```
arp -i eth0
```

۶. ذخیره اطلاعات ARP در فایل arp.txt:

```
arp -a > arp.txt
```

۷. بازیابی اطلاعات ARP از فایل arp.txt:

```
arp -f arp.txt
```

۸. نمایش فقط آدرس های MAC به صورت Manufacturer Address:

```
arp -a -m
```

۹. نمایش اطلاعات ARP برای آدرس IP 192.168.1.103 با نوع پروتکل IPv6:

```
arp -a -n -p ipv6 192.168.1.103
```

۱۰. اضافه کردن ورودی جدید برای آدرس IP 192.168.1.104 با آدرس MAC 00:11:22:33:44:55 و نوع آدرس سخت افزاری Infiniband:

```
arp -s 192.168.1.104 00:11:22:33:44:55 -t infiniband
```

۱. بررسی وجود یک ورودی در جدول ARP:

```
arp -a | grep 192.168.1.100
```

۲. نمایش اطلاعات ARP برای تمام رابط های شبکه:

```
for i in $(ifconfig | grep -v lo | awk '{print $1}'); do arp -i $i; done
```

۳. یافتن آدرس IP مربوط به آدرس MAC 00:11:22:33:44:55:

```
arp -a | grep -E "00:11:22:33:44:55" | awk '{print $1}'
```

۴. حل مشکل عدم تطابق آدرس IP و MAC:

```
arp -d 192.168.1.100
arp -s 192.168.1.100 00:11:22:33:44:55
```

۵. اسکن شبکه برای یافتن تمام دستگاه های متصل:

```
arp -a | awk '{print $1}' | sort | uniq | xargs -I {} ping -c 1 {}
```

۶. رصد تغییرات در جدول ARP:

```
while true; do arp -a; sleep 1; done | grep -E "192.168.1.100|00:11:22:33:44:55"
```

۷. ارسال درخواست ARP برای یک آدرس IP خاص:

```
arp -i eth0 -p ipv6 192.168.1.103
```

۸. تنظیم آدرس MAC ثابت برای یک آدرس IP خاص:

```
echo “192.168.1.100 00:11:22:33:44:55” >> /etc/arp/aliases
```

۹. حذف تمام ورودی های جدول ARP:

```
arp -d -a
```

۱۰. فعال کردن حالت verbose:

```
arp -v
```

دستور arp-scan

هاست های هدف باید در خط فرمان مشخص شوند مگر اینکه گزینه file -- داده شود که در این حالت اهداف به جای آن از فایل مشخص شده خوانده می شوند یا از گزینه localnet -- استفاده می شود که در این صورت اهداف از شبکه تولید می شوند. آدرس IP رابط و ماسک شبکه. برای اجرای arp-scan باید root باشید یا arp-scan باید ریشه SUID باشد، زیرا توابعی که برای خواندن و نوشتن بسته ها استفاده می‌کند نیاز به دسترسی ریشه دارند. arp-scan بسته‌های ARP را به میزبان‌های شبکه محلی ارسال می‌کند و هر پاسخی را که دریافت می‌شود نمایش می‌دهد. رابط شبکه مورد استفاده را می توان با گزینه interface -- مشخص کرد. اگر این گزینه وجود نداشته باشد، arp-scan لیست رابط سیستم را برای کمترین شماره و پیکربندی شده (به استثنای Loopback) جستجو می‌کند. به طور پیش‌فرض، بسته‌های ARP به آدرس پخش اترنت، ff:ff:ff:ff:ff:ff ارسال می‌شوند، اما می‌توان آن را با گزینه destaddr-- تغییر داد. arp-scan از اترنت و شبکه های بی سیم ۸۰۲,۱۱ پشتیبانی می‌کند. همچنین می تواند از حلقه توکن و FDDI پشتیبانی کند، اما آنها آزمایش نشده اند. از پیوندهای سریالی مانند PPP یا SLIP پشتیبانی نمی‌کند، زیرا ARP روی آنها پشتیبانی نمی شود.

ساختار دستور:

```
$ arp-scan [options] [hosts...]
```

گزینه های دستور:

- h نمایش راهنمای دستور
- v نمایش نسخه arpscan
- i نام رابط شبکه برای اسکن
- r محدوده آدرس IP برای اسکن (مثال ۲۴/۱۹۲,۱۶۸,۱,۰)
- t نوع آدرس سخت افزاری (ethernet, etc.)
- m نمایش اطلاعات MAC به صورت Manufacturer Address
- n نمایش آدرس های IP به صورت عددی
- q خروجی فقط شامل آدرس های IP و MAC
- l ذخیره اطلاعات اسکن در فایل
- P پورت UDP برای ارسال درخواست) ARPپیش فرض ۶۷)
- b تعداد بایت های ارسال شده در هر درخواست) ARPپیش فرض ۲۸)
- T Timeout برای دریافت پاسخ) ARPپیش فرض ۱ ثانیه)
- c تعداد دفعات ارسال درخواست ARP برای هر آدرس) IPپیش فرض ۱)
- d فعال کردن حالت debug

مثال های دستور:

۱. اسکن تمام آدرس های IP در شبکه ۲۴/۱۹۲.۱۶۸.۱.۰:

```
arp-scan -i eth0 192.168.1.0/24
```

۲. اسکن آدرس های IP 192.168.1.1 تا ۱۹۲.۱۶۸.۱.۱۰۰:

```
arp-scan -i eth0 192.168.1.1-192.168.1.100
```

۳. نمایش فقط آدرس های IP و MAC:

```
arp-scan -i eth0 192.168.1.0/24 -q
```

۴. ذخیره اطلاعات اسکن در فایل arp.txt:

```
arp-scan -i eth0 192.168.1.0/24 -l arp.txt
```

۵. اسکن شبکه با استفاده از پورت UDP 53:

```
arp-scan -i eth0 192.168.1.0/24 -P 53
```

۶. ارسال ۶۴ بایت در هر درخواست ARP:

```
arp-scan -i eth0 192.168.1.0/24 -b 64
```

۷. افزایش Timeout به ۲ ثانیه:

```
arp-scan -i eth0 192.168.1.0/24 -T 2
```

۸. ارسال ۳ بار درخواست ARP برای هر آدرس IP:

```
arp-scan -i eth0 192.168.1.0/24 -c 3
```

۹. فعال کردن حالت debug:

```
arp-scan -i eth0 192.168.1.0/24 -d
```

۱۰. اسکن شبکه ۲۴/۱۹۲.۱۶۸.۱.۰ و نمایش اطلاعات MAC به صورت Manufacturer Address:

```
arp-scan -i eth0 192.168.1.0/24 -m
```

دستور traceroute

دستور traceroute یک ابزار است که در سیستم‌عامل‌های مختلف برای پیدا کردن مسیری که یک بسته داده برای رسیدن به یک مقصد خاص از طریق آن طی می‌کند، استفاده می‌شود. این دستور از پروتکل ICMP (Internet Control Message Protocol) یا UDP (User Datagram Protocol) برای ارسال بسته‌های داده به مقصد استفاده می‌کند و سپس مسیری که بسته‌ها در انتقال خود طی می‌کنند را نشان می‌دهد. همچنین دستور traceroute (برای ردیابی مسیر ارسال بسته ها از مبدا تا مقصد در شبکه استفاده می‌شود. این ابزار به شما امکان می‌دهد تا گام به گام مسیر عبور بسته ها را مشاهده کنید و اطلاعات مربوط به هر گام مانند آدرسIP ، نام هاست، و زمان رفت و برگشت را بدست آورید.

گزینه های دستور:

- h نمایش راهنمای دستور
- v نمایش نسخه traceroute
- d فعال کردن حالت debug
- n نمایش آدرس های IP به صورت عددی
- p پروتکل مورد نظر برای ردیابی (tcp, udp, icmp, etc.)
- m تعداد هاپ های حداکثر

- w زمان انتظار برای پاسخ (بر حسب ثانیه)
- f اولین هاپ برای شروع ردیابی
- r مسیر برگشت را نیز ردیابی
- t نام هاست یا آدرس IP مقصد
- I نام رابط شبکه برای ارسال بسته ها

مثال های دستور:

۱. استفاده از گزینه n:-

این گزینه باعث می‌شود که آدرس‌های IP به جای نام‌های DNS در نتایج نمایش داده شوند.

```
traceroute -n google.com
```

۲. استفاده از گزینه I:-

این گزینه برای ارسال بسته‌ها از طریق ICMP (پیام‌های کنترل اینترنت) استفاده می‌شود.

```
traceroute -I google.com
```

۳. استفاده از گزینه U:-

این گزینه برای ارسال بسته‌ها از طریق پروتکل UDP استفاده می‌شود.

```
traceroute -U google.com
```

۴. استفاده از گزینه q:-

این گزینه برای تعیین تعداد بسته‌های ارسالی در هر مرحله از تراسراوت استفاده می‌شود.

```
traceroute -q 3 google.com
```

۵. استفاده از گزینه m:-

این گزینه برای تعیین تعداد هاپ‌ها (هاپ: مراحل مختلفی که بسته‌ها در طول مسیر انتقال طی می‌کنند) استفاده می‌شود.

```
traceroute -m 15 google.com
```

۶. استفاده از گزینه w:-

این گزینه برای تعیین زمان انتظار برای دریافت پاسخ از هر هاپ استفاده می‌شود.

```
traceroute -w 2 google.com
```

۷. استفاده از گزینه T:-

این گزینه برای ارسال بسته‌ها از طریق TCP (کنترل انتقال پروتکل) استفاده می‌شود.

```
traceroute -T google.com
```

مثال های بیشتر:

۱. ردیابی مسیر به google.com:

```
traceroute google.com
```

۲. ردیابی مسیر به آدرس IP 192.168.1.100 با استفاده از پروتکل UDP:

```
traceroute -p udp 192.168.1.100
```

۳. ردیابی مسیر با حداکثر ۵ هاپ:

```
traceroute -m 5 google.com
```

۴. ردیابی مسیر با ۱۰ ثانیه زمان انتظار:

```
traceroute -w 10 google.com
```

۵. ردیابی مسیر از هاپ ۳ به بعد:

```
traceroute -f 3 google.com
```

۶. ردیابی مسیر رفت و برگشت:

```
traceroute -r google.com
```

۷. ردیابی مسیر به google.com با استفاده از رابط شبکه eth0:

```
traceroute -I eth0 google.com
```

۸. نمایش جزئیات بیشتر در مورد هر هاپ:

```
traceroute -d google.com
```

۹. ردیابی مسیر به یک آدرس IPv6:

```
traceroute -6 google.com
```

۱۰. ذخیره خروجی traceroute در یک فایل:

```
traceroute google.com > traceroute.txt
```

دستور tracepath

دستور tracepath نیز همانند traceroute برای پیدا کردن مسیری که یک بسته داده برای رسیدن به یک مقصد خاص از طریق آن طی می‌کند، استفاده می‌شود. اما با تفاوت اینکه tracepath برای ارسال بسته‌ها از طریق پروتکل ICMP (Internet Control Message Protocol) استفاده می‌کند و اطلاعات جزئی‌تری از هر هاپ را نشان می‌دهد.

گزینه های دستور:

- h نمایش راهنمای دستور
- v نمایش نسخه tracepath
- n نمایش آدرس های IP به صورت عددی
- p پروتکل مورد نظر برای ردیابی (tcp, udp, icmp, etc.)
- m ماژول iptables مورد نظر برای ردیابی (state, limit, etc.)
- s آدرس IP مبدا
- d آدرس IP مقصد
- sport پورت مبدا
- dport پورت مقصد
- t نام جدول iptables برای ردیابی
- j عمل مورد نظر برای ردیابی (ACCEPT, DROP, REJECT, etc.)
- نام رابط شبکه برای ردیابی
- f تعداد هاپ های اولیه
- l تعداد هاپ های نهایی
- M حداکثر تعداد هاپ ها
- i فاصله زمانی بین پروب ها
- T حداکثر زمان انتظار برای پاسخ
- w حداکثر تعداد تلاش ها برای هر هاپ
- U استفاده از پروتکل UDP

- S استفاده از پروتکل TCP
- I استفاده از پروتکل ICMP
- ۴ استفاده از پروتکل IPv4
- ۶ استفاده از پروتکل IPv6
- x نمایش اطلاعات TTL
- q عدم نمایش اطلاعات پیشرفت
- V نمایش اطلاعات مربوط به نسخه

مثال های دستور :

1. نمایش راهنمای دستور:

```
tracepath -h
```

2. نمایش نسخه tracepath:

```
tracepath -v
```

3. ردیابی مسیر به آدرس IP 192.168.1.100 و نمایش آدرس های IP به صورت عددی:

```
tracepath -n 192.168.1.100
```

4. ردیابی مسیر به آدرس IP 192.168.1.100 با استفاده از پروتکل TCP:

```
tracepath -p tcp 192.168.1.100
```

5. ردیابی مسیر به آدرس IP 192.168.1.100 با استفاده از ماژول iptables "state":

```
tracepath -m state 192.168.1.100
```

6. ردیابی مسیر از آدرس IP 10.0.0.1 به آدرس IP 192.168.1.100:

```
tracepath -s 10.0.0.1 -d 192.168.1.100
```

7. ردیابی مسیر به آدرس IP 192.168.1.100 با پورت مبدا ۸۰:

```
tracepath --sport 80 192.168.1.100
```

8. ردیابی مسیر به آدرس IP 192.168.1.100 با پورت مقصد ۲۲:

```
tracepath --dport 22 192.168.1.100
```

9. ردیابی مسیر به آدرس IP 192.168.1.100 در جدول iptables "nat":

```
tracepath -t nat 192.168.1.100
```

10. ردیابی مسیر به آدرس IP 192.168.1.100 با عمل "ACCEPT":

```
tracepath -j ACCEPT 192.168.1.100
```

11. ردیابی مسیر به آدرس IP 192.168.1.100 از طریق رابط شبکه eth0:

```
tracepath -o eth0 192.168.1.100
```

12. ردیابی مسیر به آدرس IP 192.168.1.100 با شروع از هاپ ۳:

```
tracepath -f 3 192.168.1.100
```

13. ردیابی مسیر به آدرس IP 192.168.1.100 فقط تا هاپ ۷:

```
tracepath -l 7 192.168.1.100
```

14. ردیابی مسیر به آدرس IP 192.168.1.100 با حداکثر ۱۰ هاپ:

```
tracepath -M 10 192.168.1.100
```

15. ردیابی مسیر به آدرس IP 192.168.1.100 با فاصله زمانی ۳ ثانیه بین پروب ها:

```
tracepath -i 3 192.168.1.100
```

16. ردیابی مسیر به آدرس IP 192.168.1.100 با حداکثر ۵ ثانیه انتظار برای پاسخ:

```
tracepath -T 5 192.168.1.100
```

17. ردیابی مسیر به آدرس IP 192.168.1.100 با حداکثر ۵ ثانیه انتظار برای پاسخ:

```
tracepath -T 5 192.168.1.100
```

18. ردیابی مسیر به آدرس IP 192.168.1.100 با حداکثر ۲ تلاش برای هر هاپ:

```
tracepath -w 2 192.168.1.100
```

19. ردیابی مسیر به آدرس IP 192.168.1.100 با استفاده از پروتکل UDP:

```
tracepath -U 192.168.1.100
```

20. ردیابی مسیر به آدرس IP 192.168.1.100 با استفاده از پروتکل TCP:

```
tracepath -S 192.168.1.100
```

21. ردیابی مسیر به آدرس IP 192.168.1.100 با استفاده از پروتکل ICMP:

```
tracepath -I 192.168.1.100
```

22. ردیابی مسیر به آدرس IP 192.168.1.100 با استفاده از پروتکل IPv4:

```
tracepath -4 192.168.1.100
```

23. ردیابی مسیر به آدرس IP 192.168.1.100 با استفاده از پروتکل IPv6:

```
tracepath -6 192.168.1.100
```

24. ردیابی مسیر به آدرس IP 192.168.1.100 و نمایش اطلاعات TTL:

```
tracepath -x 192.168.1.100
```

25. ردیابی مسیر به آدرس IP 192.168.1.100 و عدم نمایش اطلاعات پیشرفت:

```
tracepath -q 192.168.1.100
```

26. ردیابی مسیر به آدرس IP 192.168.1.100 و نمایش اطلاعات مربوط به نسخه:

```
tracepath -V 192.168.1.100
```

27. ردیابی مسیر به آدرس IP 192.168.1.100 با استفاده از رابط شبکه eth1 و پروتکل UDP:

```
tracepath -o eth1 -U 192.168.1.100
```

28. ردیابی مسیر به آدرس IP 192.168.1.100 با شروع از هاپ ۲ و حداکثر ۸ هاپ:

```
tracepath -f 2 -M 8 192.168.1.100
```

29. ردیابی مسیر به آدرس IP 192.168.1.100 با فاصله زمانی ۱ ثانیه بین پروب ها و حداکثر ۴ ثانیه انتظار برای پاسخ:

```
tracepath -i 1 -T 4 192.168.1.100
```

30. ردیابی مسیر به آدرس IP 192.168.1.100 با حداکثر ۳ تلاش برای هر هاپ و عدم نمایش اطلاعات پیشرفت:

```
tracepath -w 3 -q 192.168.1.100
```

دستور mtr

دستور mtr در سیستم‌عامل‌های Unix/Linux برای انجام تست پینگ و traceroute به صورت ترکیبی استفاده می‌شود. این دستور به شما این امکان را می‌دهد که همزمان پینگ pingرا بر روی یک مسیر خاص انجام دهید و همچنین مسیری که بسته‌ها برای رسیدن به مقصد طی می‌کنند را نیز نمایش دهید.

گزینه های دستور:

- h نمایش راهنمای دستور
- v نمایش نسخه mtr
- n نمایش آدرس های IP به صورت عددی
- p پروتکل مورد نظر برای ردیابی (tcp, udp, icmp, etc.)
- m ماژول iptables مورد نظر برای ردیابی (state, limit, etc.)
- s آدرس IP مبدا
- d آدرس IP مقصد
- sport پورت مبدا
- dport پورت مقصد
- t نام جدول iptables برای ردیابی
- j عمل مورد نظر برای ردیابی (ACCEPT, DROP, REJECT, etc.)
- ن‌ام رابط شبکه برای ردیابی
- f تعداد هاپ های اولیه
- l تعداد هاپ های نهایی
- M حداکثر تعداد هاپ ها
- i فاصله زمانی بین پروب ها
- T حداکثر زمان انتظار برای پاسخ
- w حداکثر تعداد تلاش ها برای هر هاپ
- U استفاده از پروتکل UDP
- S استفاده از پروتکل TCP
- I استفاده از پروتکل ICMP
- ۴ استفاده از پروتکل IPv4
- ۶ استفاده از پروتکل IPv6
- x نمایش اطلاعات TTL
- q عدم نمایش اطلاعات پیشرفت
- V نمایش اطلاعات مربوط به نسخه
- r گزارش دهی به صورت خلاصه
- C گزارش دهی به صورت csv
- P گزارش دهی به صورت pcap
- b گزارش دهی به صورت html
- g گزارش دهی به صورت graph
- F نام فایل برای ذخیره گزارش
- A اضافه کردن اطلاعات اضافی به گزارش
- L نمایش اطلاعات مربوط به رابط های شبکه
- H نمایش اطلاعات مربوط به هاست ها
- N نمایش اطلاعات مربوط به نام های دامنه
- Q نمایش اطلاعات مربوط به کیفیت لینک ها

- D نمایش اطلاعات مربوط به تاخیر
- R نمایش اطلاعات مربوط به jitter
- P نمایش اطلاعات مربوط به packet loss

مثال های دستور:

۱. ردیابی مسیر به آدرس IP 192.168.1.100 و نمایش اطلاعات به صورت خلاصه:

```
mtr -r 192.168.1.100
```

۲. ردیابی مسیر به آدرس IP 192.168.1.100 و ذخیره گزارش در فایل mtr.csv:

```
mtr -C 192.168.1.100 -F mtr.csv
```

۳. ردیابی مسیر به آدرس IP 192.168.1.100 و نمایش اطلاعات مربوط به رابط های شبکه:

```
mtr -L 192.168.1.100
```

۴. ردیابی مسیر به آدرس IP 192.168.1.100 و نمایش اطلاعات مربوط به هاست ها:

```
mtr -H 192.168.1.100
```

۵. ردیابی مسیر به آدرس IP 192.168.1.100 و نمایش اطلاعات مربوط به نام های دامنه:

```
mtr -N 192.168.1.100
```

۶. ردیابی مسیر به آدرس IP 192.168.1.100 و نمایش اطلاعات مربوط به کیفیت لینک ها:

```
mtr -Q 192.168.1.100
```

۷. ردیابی مسیر به آدرس IP 192.168.1.100 و نمایش اطلاعات مربوط به تاخیر:

```
mtr -D 192.168.1.100
```

۸. ردیابی مسیر به آدرس IP 192.168.1.100 و نمایش اطلاعات مربوط به jitter:

```
mtr -R 192.168.1.100
```

۹. ردیابی مسیر به آدرس IP 192.168.1.100 و نمایش اطلاعات مربوط به packet loss:

```
mtr -P 192.168.1.100
```

۱۰. ردیابی مسیر به آدرس IP 192.168.1.100 و اضافه کردن اطلاعات اضافی به گزارش:

```
mtr -A 192.168.1.100
```

۱۱. ردیابی مسیر به آدرس IP 192.168.1.100 با استفاده از رابط شبکه eth0:

```
mtr -o eth0 192.168.1.100
```

۱۲. ردیابی مسیر به آدرس IP 192.168.1.100 با شروع از هاپ ۳:

```
mtr -f 3 192.168.1.100
```

۱۳. ردیابی مسیر به آدرس IP 192.168.1.100 فقط تا هاپ ۷:

```
mtr -l 7 192.168.1.100
```

۱۴. ردیابی مسیر به آدرس IP 192.168.1.100 با حداکثر ۱۰ هاپ:

```
mtr -M 10 192.168.1.100
```

۱۵. ردیابی مسیر به آدرس IP 192.168.1.100 با فاصله زمانی ۳ ثانیه بین پروب ها:

```
mtr -i 3 192.168.1.100
```

۱۶. ردیابی مسیر به آدرس IP 192.168.1.100 با حداکثر ۵ ثانیه انتظار برای پاسخ:

```
mtr -T 5 192.168.1.100
```

۱۷. ردیابی مسیر به آدرس IP 192.168.1.100 با حداکثر ۲ تلاش برای هر هاپ:

```
mtr -w 2 192.168.1.100
```

۱۸. ردیابی مسیر به آدرس IP 192.168.1.100 با حداکثر ۳ تلاش برای هر هاپ و عدم نمایش اطلاعات پیشرفت:

```
mtr -w 3 -q 192.168.1.100
```

۱۹. ردیابی مسیر به آدرس IP 192.168.1.100 با استفاده از پروتکل UDP:

```
mtr -U 192.168.1.100
```

۲۰. ردیابی مسیر به آدرس IP 192.168.1.100 با استفاده از پروتکل TCP:

```
mtr -S 192.168.1.100
```

۲۱. ردیابی مسیر به آدرس IP 192.168.1.100 با استفاده از پروتکل ICMP:

```
mtr -I 192.168.1.100
```

۲۲. ردیابی مسیر به آدرس IP 192.168.1.100 با استفاده از پروتکل IPv4:

```
mtr -4 192.168.1.100
```

۲۳. ردیابی مسیر به آدرس IP 192.168.1.100 با استفاده از پروتکل IPv6:

```
mtr -6 192.168.1.100
```

۲۴. ردیابی مسیر به آدرس IP 192.168.1.100 و نمایش اطلاعات TTL:

```
mtr -x 192.168.1.100
```

۲۵. ردیابی مسیر به آدرس IP 192.168.1.100 و عدم نمایش اطلاعات پیشرفت:

```
mtr -q 192.168.1.100
```

۲۶. ردیابی مسیر به آدرس IP 192.168.1.100 و نمایش اطلاعات مربوط به نسخه:

```
mtr -V 192.168.1.100
```

۲۷. ردیابی مسیر به آدرس IP 192.168.1.100 با استفاده از رابط شبکه eth1 و پروتکل UDP:

```
mtr -o eth1 -U 192.168.1.100
```

۲۸. ردیابی مسیر به آدرس IP 192.168.1.100 با شروع از هاپ ۲ و حداکثر ۸ هاپ:

```
mtr -f 2 -M 8 192.168.1.100
```

۲۹. ردیابی مسیر به آدرس IP 192.168.1.100 با فاصله زمانی ۱ ثانیه بین پروب ها و حداکثر ۴ ثانیه انتظار برای پاسخ:

```
mtr -i 1 -T 4 192.168.1.100
```

دستور nmcli

دستور nmcli یک ابزار موجود در بسته NetworkManager است که به کاربران امکان مدیریت شبکه‌های سیمی و بی‌سیم را در محیط خط فرمان لینوکس فراهم می‌کند. این ابزار به شما امکان می‌دهد تا شبکه‌های WiFi و اترنت را مدیریت کنید، اطلاعات شبکه را نمایش دهید، اتصالات را پیکربندی کنید و بسیاری از وظایف مرتبط با شبکه را انجام دهید.

گزینه های دستور:

- h نمایش راهنمای دستور
- v نمایش نسخه nmcli
- g نمایش اطلاعات مربوط به اتصالات شبکه
- c نمایش اطلاعات مربوط به پیکربندی اتصالات شبکه
- a نمایش اطلاعات مربوط به آدرس های IP

- d نمایش اطلاعات مربوط به DNS
- r نمایش اطلاعات مربوط به مسیرها
- p نمایش اطلاعات مربوط به پروکسی ها
- m نمایش اطلاعات مربوط به VPN ها
- s نمایش اطلاعات مربوط به سرویس های شبکه
- t نمایش اطلاعات مربوط به تنظیمات زمان
- u نمایش اطلاعات مربوط به کاربران
- i نمایش اطلاعات مربوط به رابط های شبکه
- l نمایش اطلاعات مربوط به لینک ها
- w نمایش اطلاعات مربوط به بی سیم
- f نمایش اطلاعات مربوط به فایل های پیکربندی
- e ویرایش پیکربندی اتصالات شبکه
- C ایجاد اتصال شبکه جدید
- D حذف اتصال شبکه
- A فعال کردن اتصال شبکه
- d غیرفعال کردن اتصال شبکه

مثال های دستور:

1. نمایش اطلاعات مربوط به اتصالات شبکه:

```
nmcli connection show
```

2. نمایش اطلاعات مربوط به پیکربندی اتصالات شبکه:

```
nmcli connection show --config
```

3. نمایش اطلاعات مربوط به آدرس های IP:

```
nmcli connection show --addresses
```

4. نمایش اطلاعات مربوط به DNS:

```
nmcli connection show --dns
```

5. نمایش اطلاعات مربوط به مسیرها:

```
nmcli connection show --routes
```

6. نمایش اطلاعات مربوط به پروکسی ها:

```
nmcli connection show --proxy
```

7. نمایش اطلاعات مربوط به VPN ها:

```
nmcli connection show --vpn
```

8. نمایش اطلاعات مربوط به سرویس های شبکه:

```
nmcli service show
```

9. نمایش اطلاعات مربوط به تنظیمات زمان:

```
nmcli time show
```

10. نمایش اطلاعات مربوط به کاربران:

```
nmcli user show
```

11. نمایش اطلاعات مربوط به رابط های شبکه:

```
nmcli device show
```

12.نمایش اطلاعات مربوط به لینک ها:

```
nmcli link show
```

13.نمایش اطلاعات مربوط به بی سیم:

```
nmcli wifi show
```

14.نمایش اطلاعات مربوط به فایل های پیکربندی:

```
nmcli config show
```

15.ویرایش پیکربندی اتصالات شبکه:

```
nmcli connection edit <نام اتصال>
```

16.ایجاد اتصال شبکه جدید:

```
nmcli connection add <نام اتصال> type <نوع اتصال>
```

17.حذف اتصال شبکه:

```
nmcli connection delete <نام اتصال>
```

18.فعال کردن اتصال شبکه:

```
nmcli connection up <نام اتصال>
```

19.غیرفعال کردن اتصال شبکه:

```
nmcli connection down <نام اتصال>
```

دستور tcpdump

دستور tcpdump در لینوکس برای بررسی و ضبط ترافیک شبکه استفاده می‌شود. این دستور امکان نمایش و ذخیره بسته‌های شبکه را فراهم می‌کند تا بتوانید بررسی‌های مختلفی روی ترافیک شبکه انجام دهید.

گزینه‌های دستور:

- h نمایش راهنمای دستور
- v نمایش نسخه tcpdump
- i نام رابط شبکه مورد نظر برای ضبط ترافیک
- p فیلتر کردن بر اساس پروتکل (tcp, udp, icmp, etc.)
- s تعداد بایت های دریافتی برای هر بسته
- w ذخیره ترافیک ضبط شده در فایل
- r خواندن ترافیک ضبط شده از فایل
- n عدم نمایش نام های دامنه و آدرس های IP به صورت عددی
- A نمایش اطلاعات مربوط به تمام پروتکل ها
- t عدم نمایش timestamp در خروجی
- c تعداد بسته های دریافتی قبل از توقف
- F استفاده از فیلتر BPF
- X نمایش اطلاعات مربوط به لایه های مختلف پروتکل

- l عدم ارسال بسته ها به لایه های بالاتر
- q عدم نمایش اطلاعات اضافی
- d رمزگشایی پروتکل های مختلف
- e رمزگشایی فیلدهای خاص در پروتکل ها
- vvv نمایش اطلاعات مربوط به جزئیات

مثال های دستور: l

ضبط تمام ترافیک شبکه:

```
tcpdump
```

ضبط ترافیک شبکه فقط از رابط eth0:

```
tcpdump i eth0
```

ضبط ترافیک شبکه فقط از پروتکل TCP:

```
tcpdump p tcp
```

ضبط ترافیک شبکه فقط از پروتکل UDP:

```
tcpdump p udp
```

ضبط ترافیک شبکه فقط از پروتکل ICMP:

```
tcpdump p icmp
```

ضبط ۱۰۰ بایت اول هر بسته:

```
tcpdump s 100
```

ذخیره ترافیک ضبط شده در فایل tcpdump.pcap:

```
tcpdump w tcpdump.pcap
```

خواندن ترافیک ضبط شده از فایل tcpdump.pcap:

```
tcpdump r tcpdump.pcap
```

نمایش نام های دامنه و آدرس های IP به صورت عددی:

```
tcpdump n
```

نمایش اطلاعات مربوط به تمام پروتکل ها:

```
tcpdump A
```

عدم نمایش timestamp در خروجی:

```
tcpdump t
```

دریافت ۱۰۰ بسته و سپس توقف:

```
tcpdump c 100
```

ستفاده از فیلتر BPF برای فقط نمایش بسته های TCP با پورت ۸۰:

```
tcpdump F “tcp port 80”
```

نمایش اطلاعات مربوط به لایه های مختلف پروتکل:

```
tcpdump X
```

دم ارسال بسته ها به لایه های بالاتر:

```
tcpdump l
```

عدم نمایش اطلاعات اضافی:

```
tcpdump q
```

رمزگشایی پروتکل HTTP:

```
tcpdump d “tcp port 80”
```

رمزگشایی فیلد source IP در پروتکل IP:

```
tcpdump e “ip.src”
```

.نمایش اطلاعات مربوط به جزئیات

ضبط ترافیک شبکه فقط از رابط eth1 و فقط از پروتکل TCP با پورت ۸۰:

```
tcpdump i eth1 p tcp port 80
```

ضبط ترافیک شبکه فقط از رابط eth0 و فقط از پروتکل UDP با پورت ۵۳:

```
tcpdump i eth0 p udp port 53
```

ضبط ترافیک شبکه فقط از رابط eth1 و فقط از پروتکل ICMP:

```
tcpdump i eth1 p icmp
```

ضبط ۲۰۰ بایت اول هر بسته از پروتکل TCP با پورت ۸۰:

```
tcpdump s 200 p tcp port 80
```

نمایش تمام ترافیک در رابط شبکه eth0:

```
tcpdump i eth0
```

ذخیره سازی بسته‌ها در یک فایل برای آنالیز بعدی:

```
tcpdump w output.pcap
```

نمایش بسته‌های TCP با طول حداکثر ۱۰۰ بایت:

```
tcpdump ‘tcp and tcp[13] = 0 and tcp[14:2] <= 100’
```

نمایش تمام بسته‌ها به صورت ASCII:

```
tcpdump A
```

نمایش تمام ترافیک در رابط شبکه eth0:

```
tcpdump i eth0
```

ذخیره سازی بسته‌ها در یک فایل برای آنالیز بعدی:

```
tcpdump w output.pcap
```

نمایش بسته‌های TCP با طول حداکثر ۱۰۰ بایت:

```
tcpdump ‘tcp and tcp[13] = 0 and tcp[14:2] <= 100’
```

نمایش تمام بسته‌ها به صورت ASCII:

```
tcpdump A
```

Aنمایش پکت‌ها به صورت ASCII.

```
tcpdump A
```

b buffer_sizeتعیین اندازه بافر برای ذخیره سازی بسته‌ها.

```
tcpdump b 1024
```

c countتعیین تعداد بسته‌هایی که باید ضبط شوند.

```
tcpdump c 10
```

C file_sizeتقسیم فایل خروجی به اندازه‌های مشخص شده.

```
tcpdump C 100
```

E spi@ipaddr algosecretرمزگذاری برای ESP و AH.

```
tcpdump E 123@192.168.1.1 algo:secret
```

F fileخواندن فیلترها از یک فایل.

```
tcpdump F filter.txt
```

G rotate_secondsچرخش فایل خروجی هر تعداد ثانیه مشخص.

```
tcpdump G 60
```

i interfaceتعیین رابط شبکه برای گوش دادن.

```
tcpdump i eth0
```

immediatemodeفعال کردن حالت فوری برای شروع فوری ضبط بسته‌ها.

```
tcpdump immediatemode
```

j tstamp_typeتنظیم نوع برچسب زمانی برای بسته‌ها.

```
tcpdump j unix
```

m moduleاستفاده از ماژول برای تجزیه بسته‌ها.

```
tcpdump m module_name
```

M secretتنظیم کلید برای دیکریپت بسته‌ها.

```
tcpdump M my_secret
```

numberنمایش شماره بسته‌ها.

```
tcpdump number
```

printچاپ کردن بسته‌ها به صورت خوانا.

```
tcpdump print
```

Q in|out|inoutفیلتر کردن بسته‌ها بر اساس جهت.

```
tcpdump Q in
```

r fileپخش بسته‌ها از یک فایل ذخیره شده.

```
tcpdump r captured_packets.pcap
```

s snaplenتعیین حداکثر طول بسته‌ها برای ذخیره سازی.

```
tcpdump s 128
```

T typeتعیین نوع بسته‌ها برای تجزیه.

```
tcpdump T udp
```

versionنمایش نسخه tcpdump.

```
tcpdump version
```

V fileخواندن فیلترها از یک فایل.

```
tcpdump V filter.txt
```

w fileذخیره سازی بسته‌ها در یک فایل برای آنالیز بعدی.

```
tcpdump w captured_packets.pcap
```

W filecountتنظیم تعداد فایل‌های خروجی.

tcpdump W 5

yانتخاب نوع داده‌لینک برای تجزیه. datalinktype

tcpdump y EN10MB

zاجرای دستور پس از چرخش فایل‌ها. postrotatecommand

tcpdump z gzip

Z تنظیم کاربر برای اجرای tcpdump. user

tcpdump Z root

timestampprecision=tstamp_precision تعیین دقت برچسب زمانی.

tcpdump timestampprecision nano

microاستفاده از دقت میکروثانیه برای برچسب زمانی.

tcpdump micro

nanoاستفاده از دقت نانوثانیه برای برچسب زمانی.

tcpdump nano

اعمال فیلترها یا شرایط بر روی بسته‌ها.

tcpdump host 192.168.1.1

دستور Wget

قصد داریم ابزار wgetرا بررسی کنیم که فایل ها را از وب سایت جهانی (WWW) دانلود می کند. با استفاده از پروتکل های گسترده مانند HTTP، HTTPS و FTP . ابزار Wgetبسته آزادانه موجود است و مجوز تحت مجوز GNU GPL است.این ابزار می تواند هر system عامل یونیکس مانندWindows و MAC OS را install کند.این یک ابزار خط دستور غیر تعاملی است.ویژگی اصلی Wget آن استحکام است.این به گونه ای طراحی شده است که در اتصالات شبکه آهسته یا ناپایدار شبکه کار کند .به صورت خودکار شروع به download میکند و در صورت مشکل در شبکه، آن را لغو میکند . Wgetبارگیری کننده شبکه غیرتعاملی است که برای بارگیری پرونده ها از سرور استفاده می شود حتی اگر کاربر به سیستم وارد نشده باشد و بدون اینکه مانع روند فعلی شود می تواند در پس زمینه کار کند.

- wget GNUیک ابزار آزاد برای بارگیری غیر تعاملی پرونده ها از وب است .از پروتکل های HTTP ، HTTPSو FTPو همچنین بازیابی از طریق پروکسی های HTTP پشتیبانی می کند.
- wgetغیر تعاملی است ، به این معنی که می تواند در پس زمینه کار کند ، در حالی که کاربر به سیستم وارد نشده است .با این کار می توانید بازیابی را شروع کرده و از سیستم جدا شوید و به wget اجازه دهید کار را تمام کند .در مقابل ، بیشتر مرورگرهای وب به حضور مداوم کاربر نیاز دارند ، که می تواند مانع بزرگی در هنگام انتقال اطلاعات زیادی باشد.
- wgetمی تواند پیوندها را در صفحات HTML و XHTML دنبال کند و نسخه های محلی وب سایت های از راه دور را ایجاد کند و ساختار فهرست سایت اصلی را به طور کامل بازسازی کند .گاهی اوقات از این به عنوان بارگیری بازگشتی یاد می شود .در حالی که این کار را انجام می دهید ، wgetبه استاندارد حذف Robot (/robots.txt) احترام می گذارد .به wgetمی توان دستور داد که پیوندهای موجود در پرونده های HTML بارگیری شده را به پرونده های محلی برای مشاهده آفلاین تبدیل کند.
- wgetبرای مقاومت در برابر اتصالات شبکه کند یا ناپایدار طراحی شده است .اگر بارگیری به دلیل مشکلی در شبکه انجام نشود ، تا زمانی که کل پرونده بازیابی نشود ، دوباره در تلاش است .اگر سرور از سرگیری پشتیبانی کند ، به سرور دستور می دهد تا بارگیری را از جایی که متوقف شده است ادامه دهد.

گزینه های دستور:

- h نمایش راهنمای دستور
- V نمایش نسخه wget
- نام فایل برای ذخیره خروجی
- نام فایل برای ذخیره محتوای دانلود شده
- a اضافه کردن URL به لیست دانلود
- c ادامه دانلود فایل های نیمه کاره
- b محدود کردن پهنای باند دانلود
- t تعداد تلاش ها برای دانلود
- T حداکثر زمان انتظار برای دانلود
- spider بررسی وجود فایل بدون دانلود
- r دانلود بازگشتی صفحات وب
- l حداکثر عمق دانلود بازگشتی
- A لیست پسوندهای فایل برای دانلود
- R لیست پسوندهای فایل برای عدم دانلود
- U UserAgent برای درخواست HTTP
- e ارسال هدر HTTP
- p ذخیره اطلاعات مربوط به پروکسی
- noproxy عدم استفاده از پروکسی
- s عدم نمایش اطلاعات پیشرفت
- q عدم نمایش اطلاعات دانلود
- v نمایش اطلاعات جزئی
- nv عدم نمایش اطلاعات نسخه
- nc عدم دانلود فایل های موجود
- N عدم دانلود فایل های جدیدتر از فایل موجود
- i نام فایل لیست URL
- F استفاده از cookies ذخیره شده
- C ذخیره cookies در فایل
- nocookies عدم استفاده از cookies
- H نمایش هدر HTTP
- d ارسال داده های POST
- m ارسال روش HTTP
- Q استفاده از HTTP/2
- nocheckcertificate عدم بررسی گواهی SSL
- cacertificate گواهی SSL برای بررسی
- nohttps عدم استفاده از HTTPS
- ftpuser نام کاربری FTP
- ftppassword رمز عبور FTP
- passiveftp استفاده از حالت Passive FTP

- nopassiveftp عدم استفاده از حالت Passive FTP
- retrsymlinks دانلود لینک های نمادین
- noretrsymlinks عدم دانلود لینک های نمادین
- timestamping اضافه کردن timestamp به نام فایل ها
- notimestamping عدم اضافه کردن timestamp به نام فایل ها

ساختار دستور:

```
wget [گزینه] [URL]
```

مثال های دستور:

برای بارگیری ساده یک صفحه وب:

```
wget http://example.com/sample.php
```

برای بارگیری پرونده در پس زمینه

```
wget -b http://www.example.com/samplepage.php
```

برای بازنویسی log wile دستورwget

```
wget http://www.example.com/filename.txt -o /path/filename.txt
```

برای از سرگیری پرونده ای که بارگیری شده است

```
wget -c http://example.com/samplefile.tar.gz
```

تعداد دفعات مشخصی را امتحان کنید

```
wget --tries = 10 http://example.com/samplefile.tar.gz
```

گزینه های دستور:

-v / –version :این برای نمایش نسخه wget موجود در سیستم شما استفاده می شود.

```
$ wget -v
```

-h / –help :این برای چاپ یک پیام **راهنما** استفاده می شود که تمام گزینه های ممکن از دستور line را که با گزینه های خط فرمان wget در دسترس است، نشان می دهد.

```
$ wget -h [URL]
```

-o logfile :**از** این گزینه برای هدایت کلیه پیام های تولید شده توسط سیستم به سمت پرونده ای که توسط گزینه مشخص شده است هدایت می شود و هنگامی که روند کار به پایان می رسد ، تمام پیام های تولید شده در پرونده ورود به سیستم موجود است .اگر پرونده ورود به سیستم مشخص نشده باشد، پیام های خروجی به پرونده ورود به سیستم پیش فرض یعنی wget -log هدایت می شوند

```
$ wget -o logfile [URL]
```

-b / –background :این گزینه برای ارسال فرآیند به پس زمینه به محض شروع فرآیند استفاده می شود تا فرایندهای دیگر انجام شوند .اگر هیچ پرونده خروجی از طریق گزینه o- مشخص نشده باشد ، خروجی به طور پیش فرض به wget-log هدایت می شود.

```
$ wget -b [URL]
```

–a :**از** این گزینه برای ضمیمه کردن پیام های خروجی به پرونده ورودی خروجی فعلی بدون رونویسی پرونده به عنوان در گزینه o- ، پرونده ورودی خروجی رونویسی می شود اما با استفاده از این گزینه ، ثبت دستور قبلی ذخیره می شود و ثبت فعلی بعد از قبلی ثبت می شود.

```
$ wget - یک پرونده پرونده [URL]
```

i– :این گزینه برای خواندن URL ها از پرونده استفاده می شود .اگر i- به عنوان پرونده مشخص شده باشد ، URLها از ورودی استاندارد خوانده می شوند. اگر از این عملکرد استفاده شده باشد ، نیازی به URL در خط فرمان نیست .اگر URL هایی هم در خط فرمان و هم در یک فایل ورودی وجود داشته باشد ، آدرس های موجود در خطوط فرمان اولین کسانی هستند که بازیابی می شوند .اگر URLها فقط به صورت متوالی لیست شده باشند ، لازم نیست که این پرونده یک سند HTML باشد.

```
$ wget -i inputfile
$ wget -i inputfile [URL]
```

-t number / –tries = number :از این گزینه برای تنظیم تعداد تکرارها به تعداد مشخصی از بارها استفاده می شود 0 .یا infرا برای امتحان مجدد بی نهایت مشخص کنید .پیش فرض این است که ۲۰ بار امتحان کنید ، به استثنای خطاهای مهلکی مانند اتصال رد شد یا پیوند پیدا نشد ، که با بروز خطا دوباره امتحان نمی شوند.

```
$ wget -t number [URL]
```

c– :در صورت مثبت بودن قابلیت ادامه دانلود ، از این گزینه برای از سرگیری یک پرونده بارگیری شده استفاده می شود در غیر این صورت اگر قابلیت دانلود فایل داده نشده باشد یا مشخص نشده باشد ، بارگیری پرونده نمی تواند از سر گرفته شود.

```
$ wget -c [URL]
```

w– از این گزینه برای تنظیم سیستم استفاده می شود تا بین بازیابی ها تعداد ثانیه های مشخص شده منتظر بماند .استفاده از این گزینه توصیه می شود ، زیرا با کم شدن درخواست ها بار سرور را کاهش می دهد .به جای چند ثانیه ، می توان زمان را با استفاده از پسوند mدر دقیقه ، ساعت با پسوند h یا در روز با پسوند d تعیین کرد .تعیین مقدار زیادی برای این گزینه در صورت از کار افتادن شبکه یا میزبان مفید است ، بنابراین wget می تواند به اندازه کافی صبر کند تا منطقی انتظار داشته باشد که خطای شبکه قبل از دوباره امتحان شود.

```
$ wget -w number in seconds [URL]
```

r– :این گزینه برای روشن کردن بازیابی برگشتی پیوند مشخص شده در صورت بروز خطاهای مهلک نیز استفاده می شود .این گزینه یک تماس بازگشتی به لینک داده شده در خط فرمان است.

```
$ wget -r [URL]
```

نصب در ردهت بیس ها:

```
# yum - install wget
```

نصب در دبیان بیس ها:

```
#sudo apt install wget
```

گزینه y-که در اینجا استفاده می شود، قبل از نصب هر بسته،به عنوان تأیید است .

دانلود فایل از آدرس داده شده

```
# wget  http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz
```

فایل را با نام متفاوت download کنید.

با استفاده از گزینه O-بزرگ، دانلود فایل با نام فایل های مختلف انجام دهید. در اینجا نام فایل wget.zipرا به عنوان نشان داده شده در زیر آورده ایم.

```
# wget  -O  wget.zip http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz
```

فایل چندگانه را با پروتکل http وftp download کنید

در اینجا می بینیم که چگونه فایل های چندگانه را با استفاده از پروتکل HTTPو FTPبا دستورwget در آن ها بارگیری کنیم.

```
# wget http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz  ftp://ftp.gnu.org/gnu/wget/wget-1.10.1.tar.gz.sig  http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz
```

می توانید تعدادی URLرا در فایل متنی ذخیره کرده و آنها را با گزینه idownload- کنید.در مثال زیر ما یک فایل متنی به نام tmp.txtرا در فهرست wget ایجاد کردیم که در آنجا مجموعه ای ازURLها را برای download قرار می دهیم. تا از این فایل مسیر فایلهای دانلود را خواهنده و آنها را دانلود نماید.

```
# wget -i / wget / tmp.txt
```

بازخوانی دانلود ناتمام

در صورت download فایل بزرگ، ممکن است بعضی وقت ها برای جلوگیری از download در این مورد اتفاق بیفتد، ما می توانیم فایل مشابهی را که در آن با گزینه c-کنار گذاشته شد، download کنید.اما هنگامی که شروع به download فایل بدون مشخص کردن c-گزینه wgetاضافه خواهد شد 1.پسوند در پایان فایل، با توجه به عنوان download تازه.بنابراین، تمرین خوبی برای اضافه کردن c-هنگام بارگیری فایلهای بزرگ است.

```
# wget -c http://mirrors.hns.net.in/centos/6.3/isos/x86_64/CentOS-6.3-x86_64-LiveDVD.iso
```

فایل ها را در پس زمینه download کنید. با گزینه b-میتوانید اینکار را انجام دهید.

```
# wget -b /wget/log.txt ftp://ftp.iinet.net.au/debian/debian-cd/6.0.5/i386/iso-dvd/debian-6.0.5-i386-DVD-1. iso
```

محدودیت سرعت download را محدود کنید

باOption -limit-rate = 100k ، محدودیت سرعت download به ۱۰۰ kمحدود می شود

```
# wget -c  --limit-rate = 100k  /wget/log.txt  ftp://ftp.iinet.net.au/debian/debian-cd/6.0.5/i386/iso-dvd/debian-6.0.5 -i386-DVD-1.iso
```

مشاهده نسخه دستور

با گزینه ها version-و help- می توانید نسخه و کمک را در صورت نیاز مشاهده کنید.

```
# wget  --version
```

```
# wget –help
```

دستور ping

ping یا Packet Internet Groper برای بررسی اتصال شبکه بین میزبان و سرور/میزبان استفاده می شود. این دستور آدرس IP یا URL را به عنوان ورودی می گیرد و یک بسته داده را با پیام PING به آدرس مشخص شده ارسال می‌کند و پاسخی از سرور/میزبان دریافت می‌کند که این بار ثبت می شود که به آن تاخیر می گویند. پینگ از ICMP (پروتکل پیام کنترل اینترنت) برای ارسال پیام اکو ICMP به میزبان مشخص شده در صورت موجود بودن آن میزبان استفاده می‌کند، سپس پیام پاسخ ICMP را ارسال می‌کند. پینگ معمولاً در میلی ثانیه اندازه گیری می شود که هر سیستم عامل مدرنی این پینگ را از قبل نصب کرده باشد.

ساختاردستور:

دستور پینگ، به دنبال نام میزبان، نام یک وب سایت یا آدرس IP دقیق است.

```
$ ping [option] [hostname] or [IP address]
```

گزینه های دستور:

- h نمایش راهنمای دستور
- v نمایش نسخه ping
- n تعداد بسته های ارسالی
- c تعداد دفعات ارسال بسته ها
- i فاصله زمانی بین ارسال بسته ها (بر حسب ثانیه)
- t ارسال مداوم بسته ها تا زمانی که کاربر با Ctrl+C متوقف کند
- s اندازه بسته های ارسالی (بر حسب بایت)
- f عدم تکه تکه شدن بسته ها
- p مشخص کردن پروتکل مورد نظر برای ارسال بسته ها (tcp, udp, icmp, etc.)
- m مشخص کردن TTL بسته های ارسالی
- Q فعال کردن timestamp در بسته های ارسالی
- l فعال کردن timestamp در پاسخ های دریافتی
- A فعال کردن timestamp در هر دو بسته های ارسالی و دریافتی
- T فعال کردن timestamp دقیق در پاسخ های دریافتی
- ۴ استفاده از پروتکل IPv4
- ۶ استفاده از پروتکل IPv6
- r فعال کردن مسیریابی معکوس
- D فعال کردن Do Not Fragment (DF) bit در بسته های ارسالی
- M فعال کردن Record Route (RR) option در بسته های ارسالی
- S فعال کردن Source Specific Routing (SSR) option در بسته های ارسالی
- فعال کردن Timestamp option در بسته های ارسالی
- t فعال کردن Traceroute option در بسته های ارسالی
- a فعال کردن IP Address Resolution (ARP)
- b فعال کردن Broadcast ping
- g فعال کردن Loose Source Routing (LSR) option در بسته های ارسالی
- I مشخص کردن رابط شبکه مورد نظر برای ارسال بسته ها
- w حداکثر زمان انتظار برای پاسخ (بر حسب ثانیه)
- Q فعال کردن Quiet mode
- v فعال کردن Verbose mode

مثال های دستور:

۱. نمایش راهنمای دستور:

```
ping -h
```

۲. نمایش نسخه ping:

```
ping -v
```

۳. ارسال ۵ بسته ping به آدرس IP 192.168.1.100:

```
ping 192.168.1.100 -n 5
```

۴. ارسال ۱۰ بسته ping به آدرس IP 192.168.1.100 با فاصله زمانی ۲ ثانیه بین هر بسته:

```
ping 192.168.1.100 -c 10 -i 2
```

۵. ارسال مداوم بسته های ping به آدرس IP 192.168.1.100 تا زمانی که کاربر با Ctrl+C متوقف کند:

```
ping 192.168.1.100 -t
```

۶. ارسال بسته های ping به آدرس IP 192.168.1.100 با اندازه ۱۰۲۴ بایت:

```
ping 192.168.1.100 -s 1024
```

۷. عدم تکه تکه شدن بسته های ارسالی:

```
ping 192.168.1.100 -f
```

۸. ارسال بسته های ping به آدرس IP 192.168.1.100 با استفاده از پروتکل UDP:

```
ping 192.168.1.100 -p udp
```

۹. ارسال بسته های ping به آدرس IP 192.168.1.100 با TTL 5:

```
ping 192.168.1.100 -m 5
```

۱۰. فعال کردن timestamp در بسته های ارسالی:

```
ping 192.168.1.100 -Q
```

۱۱. فعال کردن timestamp در پاسخ های دریافتی:

```
ping 192.168.1.100 -l
```

۱۲. فعال کردن timestamp در هر دو بسته های ارسالی و دریافتی:

```
ping 192.168.1.100 -A
```

۱۳. فعال کردن timestamp دقیق در پاسخ های دریافتی:

```
ping 192.168.1.100 -T
```

۱۴. استفاده از پروتکل IPv4 برای ارسال بسته های ping:

```
ping 192.168.1.100 -4
```

۱۵. استفاده از پروتکل IPv6 برای ارسال بسته های ping:

```
ping 192.168.1.100 -6
```

۱۶. فعال کردن مسیریابی معکوس:

```
ping 192.168.1.100 -r
```

۱۷. فعال کردن Do Not Fragment (DF) bit در بسته های ارسالی:

```
ping 192.168.1.100 -D
```

۱۸. فعال کردن Record Route (RR) option در بسته های ارسالی:

```
ping 192.168.1.100 -M
```

۱۹. فعال کردن Source Specific Routing (SSR) option در بسته های ارسالی:

```
ping 192.168.1.100 -S
```

۲۰. فعال کردن Timestamp option در بسته های ارسالی:

```
ping 192.168.1.100 -O
```

۲۱. فعال کردن Traceroute option در بسته های ارسالی:

```
ping 192.168.1.100 -t
```

۲۲. فعال کردن IP Address Resolution (ARP):

```
ping 192.168.1.100 -a
```

۲۳. فعال کردن Broadcast ping:

```
ping 192.168.1.255 -b
```

۲۴. فعال کردن Loose Source Routing (LSR) option در بسته های ارسالی:

```
ping 192.168.1.100
```

Curl و wget نقاط قوت و ضعف خود را دارند. Curl پشتیبانی از پروتکل های بیشتر از wget است . curl همچنین می تواند داده ها را ارسال کند، در حالی که wget می تواند فقط داده ها را دریافت کند. wget می تواند فایل ها را به طور بازگشتی download کند.

```
wget http://releases.ubuntu.com/18.10/ubuntu-18.10-desktop-amd64.iso
```

```
curl  http://releases.ubuntu.com/18.10/ubuntu-18.10-desktop-amd64.iso
```

در لینوکس، curl ابزاری برای انتقال داده ها از یا به سرور با استفاده از یکی از پروتکل های پشتیبانی شده (DICT، FILE، FTP، FTPS، GOPHER، HTTP، HTTPS، IMAP، IMAPS، LDAP، LDAPS، POP3، POP3S، RTMP، RTSP ، SCP، SFTP، SMB، SMBS، SMTP، SMTPS، TELNET و TFTP).

مثال های دستور :

```
$  curl example.com
```

این دستور کد منبع صفحه اصلی example.com را در پنجره ترمینال چاپ می‌کند.

سینتکس دستور curl به صورت زیر است:

```
$ curl [options...]
```

گزینه های دستور :

گزینه های دستور: با یک یا دو خط تیره شروع می شوند. بسیاری از گزینه های دستور: به یک مقدار اضافی در کنار آنها نیاز دارند.شکل کوتاه "تک خط تیره" گزینه های دستور:، به عنوان مثال –d ، ممکن است با یا بدون فاصله بین آن و مقدار آن استفاده شود، اگرچه فاصله یک جداکننده توصیه شده است. به عنوان مثال، فرم طولانی "double-dash" –d، –data ، نیاز به فاصله بین آن و مقدار آن دارد. گزینه های دستور:یی که به مقادیر اضافی نیاز ندارند می توانند بلافاصله در کنار یکدیگر استفاده شوند، مثلاً می توانید همه گزینه های دستور:ی – O، L- و –v را همزمان به عنوان –OLv مشخص کنید. به طور کلی، همه گزینه های دستوری حالت بولین هستند که با option –– فعال می شوند و دوباره با no-option –– غیرفعال می شوند. یعنی دقیقاً از همان نام گزینه استفاده کنید اما پیشوند آن را با no– قرار دهید .

نصب و راه اندازی :

دستور curl با اکثر توزیع های لینوکس ارائه می شود. اما، اگر سیستم به طور پیش فرض کرل نصب نباشد. شما باید آن را به صورت دستی نصب کنید. برای نصب curl دستورات زیر را اجرا کنید:

با اجرای دستورات زیر سیستم را به روز کنید:

اکنون با اجرای دستور زیر برنامه curl را نصب کنید:

Ubuntu / Debian:

```
$sudo apt update
$sudo apt install curl
```

این دستورات ابتدا بسته‌های سیستم را بروزرسانی می‌کنند و سپس دستور curl را نصب می‌کنند.

Fedora / CentOS:

```
$sudo dnf install curl
```

یا

```
$sudo yum install curl
```

این دستورات بسته‌های curl را از مخازن مربوطه نصب می‌کنند.

Arch Linux:

```
$sudo pacman -S curl
```

با اجرای دستور زیر نصب را تایید کنید:

```
 $ curl -version
```

دستور بالا نسخه نصب شده دستور curl را نمایش می‌دهد.

برای گزینه : -I

```
$curl -I https://example.com
```

این دستور سرآیند (header) های درخواست HTTP را برای URL مشخص شده دریافت می‌کند.

برای گزینه : -L

```
$curl -L https://example.com
```

این دستور به صورت تعقیبی (follow) به URL مشخص شده درخواست ارسال می‌کند و در صورت وجود تغییر مکان(redirect) ، به مکان جدید هدایت می‌شود.

برای گزینه : -o

```
$curl -o output.txt https://example.com/file.txt
```

این دستور فایل موجود در URL مشخص شده را دانلود کرده و با نام output.txt ذخیره می‌کند.

برای گزینه : -u

```
$curl -u username:password https://example.com
```

این دستور با استفاده از نام کاربری و رمز عبور مشخص شده، احراز هویت را برای درخواست HTTP انجام می‌دهد.

برای گزینه : -X

```
$curl -X POST -d “param1=value1&param2=value2” https://example.com/api
```

این دستور یک درخواست POST به URL مشخص شده ارسال می‌کند و پارامترهای فرم را به عنوان بدنه درخواست ارسال می‌کند.

برای گزینه : -H

```
$curl -H “Content-Type: application/json” -d ‘{“param1”: “value1”, “param2”: “value2”}’ https://example.com/api
```

این دستور با استفاده از سرآیند (header) مشخص شده، درخواست HTTP را ارسال می‌کند. در این مثال، سرآیند Content-Type به عنوان application/json تنظیم شده است.

برای گزینه : -s

```
$curl -s https://example.com
```

دستور aria2c

ابزار دانلود فوق العاده سریع

ساختار دستور:

```
$ aria2c [<OPTIONS>] [<URI>|<MAGNET>|<TORRENT_FILE>|<METALINK_FILE>]
```

گزینه های دستور:

عمومی

- h نمایش راهنمای دستور
- v نمایش نسخه aria2c
- d مسیر ذخیره فایل دانلود شده
- نام فایل دانلود شده
- s حداکثر سرعت دانلود (بر حسب بایت بر ثانیه)
- x حداکثر تعداد اتصالات
- k فعال کردن KeepAlive
- U UserAgent دلخواه
- H ارسال هدرهای HTTP دلخواه
- C فعال کردن کوکی ها
- P فعال کردن احراز هویت پروکسی
- t تعداد رشته های دانلود
- m حداکثر تعداد فایل های دانلود شده به طور همزمان
- I فعال کردن دانلود جزئی
- r از سرگیری دانلودهای متوقف شده
- S فعال کردن حالت امن
- X فعال کردن حالت خصوصی
- L فعال کردن لاگ
- V فعال کردن verbosity

BitTorrent

- T تورنت مورد نظر برای دانلود
- b بیت تورنت را از URL دانلود کنید
- M حداکثر تعداد متصل به هر تورنت
- K فعال کردن DHT
- l فعال کردن Local Peer Discovery (LPD)
- u فعال کردن uTP
- p پورت مورد استفاده برای BitTorrent

Metalink

- M Metalink مورد نظر برای دانلود
- w لیست منابع Metalink

شرح دستور:

aria2 ابزاری برای دانلود فایل تحت ترمینال است. پروتکل های پشتیبانی شده برای این ابزار عبارت اند از HTTP(S)، FTP، BitTorrent و Metalink. aria2 می تواند یک فایل را از چندین منبع/پروتکل دانلود کند و سعی می‌کند از حداکثر پهنای باند دانلود شما استفاده کند. از دانلود همزمان یک فایل از HTTP(S)/FTP و BitTorrent پشتیبانی می‌کند، با استفاده از checksum ، aria2 به‌طور خودکار تکه‌های داده را در حین دانلود فایلی مانند BitTorrent تأیید می‌کند. با استفاده از آپشن ها میتونید امکانات بیشتری نیز از این دستور داشته باشید. برای **مثال** عبارت زیر به راحتی می‌توان ابزار دانلود aria2 را در کلیه توزیع‌های لینوکس مانند Debian، Ubuntu، Mint، RHEL، CentOS، suse Fedora، openSUSE، Arch Linux، Manjaro، Mageia و غیره نصب نمود. بدین منظور تنها باید دستور زیر را اجرا کرد. دقت کنید که برای سیستم‌های CentOS و RHEL باید مخزن uget یا RPMForge را فعال کنیم.

نصب برنامه

در سیستم Fedora ، از دستور DNF برای نصب aria2 استفاده کنید.

```
$ sudo dnf install aria2
```

در سیستم‌های Debian/Ubuntu، از دستور APT-GET یا دستور APT برای نصب aria2 استفاده نمایید.

```
$ sudo apt install aria2
```

در سیستم‌های مبتنی بر Arch Linux، از دستور Pacman برای نصب aria2 استفاده کنید.

```
$ sudo pacman -S aria2
```

در سیستم‌های RHEL / CentOS، از دستور YUM برای نصب aria2 استفاده نمایید.

```
$ sudo yum install aria2
```

در سیستم OpenSUSE Leap، از دستور Zypper برای نصب aria2 استفاده کنید.

```
$ sudo zypper install aria2
```

مثال های دستور:

```
$ aria2c –s 16  url
```

باعث میشه فایل به ۱۶ قسمت تقسیم شود و دانلود شود.به این معنی با ۱۶ connection فایل را دانلود میکند.

```
$ aria2c –x  8  url
```

با سویچ x حداکثر تعداد کانکشن رو تعیین میکنه. مقدار حداکثر ۱۶ است.

```
$ aria2c  url1  url2
```

دانلود همزمان دولینک

```
$ aria2c -k1M
```

اگه بخواین فایل به تکه هایی بر اساس حجم تقسیم کنید میتونید از فرمان بالا استفاده کنید.

در **مثال** بالا اگه یک فایل ۱۰ مگی رو دانلود میکندو نرم افزار اونارو به ۱۰ تیکه تقسیم میکند.

```
$ aria2c  -d  /home/alireza url1
```

مسیر دانلود رو میتونید با –d تعیین کنید

اگه بخواهید تعداد لینک بیشتری رو دانلود کنید، میتونید آنها را در یک فایل متنی بنویسید و به این صورت دانلود کنید.

```
$ aria2c -i   all-links.txt
```

اگه بخواهید فایل ها به ترتیب و یا دوتا دوتا دانلود بشوند میتونید از سویچ –j استفاده کنید

```
$ aria2c –j 1 url1 url2
```

برای متوقف کردن دانلود از (ctrl + c) استفاده کنید و برای ادامه دانلود متوقف شده از سویچ –c استفاده کنید

```
$ aria2c -c url
```

برای محدود کردن سرعت دانلود میتونید از فرمان زیر استفاده کنید(حرف k بزرگ است)

```
$ aria2c --max-download-limit=5K url
```

دانلود از بیت تورنت

```
$ aria2c url/xxx.torrent
```

دانلود از آدرس مگنت

```
$ aria2c "magnet:?xt=urn:btih:248D0A1C284299DE78D5C1ED359BB46717D8C&dn=aria2"
```

مثال های بیشتر:

۱. دانلود یک فایل از طریق HTTP:

```
aria2c https://example.com/file.txt
```

۲. دانلود یک فایل از طریق FTP:

```
aria2c ftp://user:password@example.com/file.txt
```

۳. دانلود یک فایل از طریق BitTorrent:

```
aria2c -T file.torrent
```

۴. دانلود یک فایل از طریق Metalink:

```
aria2c -M metalink.xml
```

۵. دانلود یک فایل با حداکثر سرعت ۱ مگابایت بر ثانیه:

```
aria2c -s 1M https://example.com/file.txt
```

۶. دانلود یک فایل با ۱۰ اتصال همزمان:

```
aria2c -x 10 https://example.com/file.txt
```

۷. از سرگیری دانلود یک فایل متوقف شده:

```
aria2c -r https://example.com/file.txt
```

۸. دانلود جزئی یک فایل:

```
aria2c -I https://example.com/file.txt
```

۹. دانلود یک فایل با استفاده از Keep-Alive:

```
aria2c -k https://example.com/file.txt
```

۱۰. دانلود یک فایل با ارسال هدرهای HTTP دلخواه:

```
aria2c -H "Accept: application/json" https://example.com/file.txt
```

دستور ss

دستور ss در سیستم عامل‌های Unix/Linux برای نمایش اطلاعات مربوط به اتصالات شبکه (مانند پورت‌های باز، اتصالات فعال و ...) استفاده می‌شود. این دستور اطلاعات مفیدی را درباره وضعیت شبکه سیستم شما نمایش می‌دهد. همچنین دستور ss ابزاری برای نمایش اطلاعات مربوط به سوکت های شبکه است. این ابزار اطلاعات کاملی از جمله نوع سوکت، پروتکل مورد استفاده، آدرس IP و پورت های مبدا و مقصد، وضعیت سوکت و آمار مربوط به ترافیک را ارائه می دهد.

گزینه های دستور:

- h نمایش راهنمای دستور
- V نمایش نسخه ss
- n نمایش آدرس های IP به صورت عددی
- a نمایش تمام سوکت ها، چه فعال و چه غیرفعال
- l نمایش سوکت های در حال گوش دادن (listening)
- e نمایش سوکت های متصل (established)
- t نمایش سوکت های TCP
- u نمایش سوکت های UDP
- d نمایش سوکت های DCCP
- r نمایش سوکت های RAW
- x نمایش سوکت های Unix
- p نمایش شناسه فرآیند ( )PIDمربوط به هر سوکت
- i نمایش رابط شبکه مربوط به هر سوکت
- s نمایش وضعیت سوکت (LISTEN, ESTABLISHED, etc.)
- m نمایش آمار مربوط به ترافیک هر سوکت
- ۴ نمایش اطلاعات مربوط به سوکت های IPv4
- ۶ نمایش اطلاعات مربوط به سوکت های IPv6
- F نمایش اطلاعات به صورت خلاصه
- c نمایش اطلاعات به صورت csv
- P نمایش اطلاعات به صورت pcap
- b نمایش اطلاعات به صورت html
- g نمایش اطلاعات به صورت graph
- f فیلتر کردن اطلاعات بر اساس معیارهای مختلف
- R فیلتر کردن اطلاعات بر اساس نام رابط شبکه
- L فیلتر کردن اطلاعات بر اساس نام هاست
- H فیلتر کردن اطلاعات بر اساس شناسه فرآیند (PID)
- N فیلتر کردن اطلاعات بر اساس نام سوکت
- Q فیلتر کردن اطلاعات بر اساس وضعیت سوکت
- D فیلتر کردن اطلاعات بر اساس پورت مبدا
- d فیلتر کردن اطلاعات بر اساس پورت مقصد
- A فیلتر کردن اطلاعات بر اساس آدرس IP مبدا
- a فیلتر کردن اطلاعات بر اساس آدرس IP مقصد
- T فیلتر کردن اطلاعات بر اساس پروتکل مورد استفاده
- M فیلتر کردن اطلاعات بر اساس TTL
- I فیلتر کردن اطلاعات بر اساس timestamp
- U فیلتر کردن اطلاعات بر اساس شناسه کاربر (UID)

G فیلتر کردن اطلاعات بر اساس شناسه گروه ((GID

مثال های دستور:

۱. نمایش تمام سوکت های فعال:

```
ss -a
```

۲. نمایش سوکت های TCP در حال گوش دادن:

```
ss -lt
```

۳. نمایش سوکت های UDP متصل:

```
ss -ue
```

۴. نمایش سوکت های DCCP با PID مربوطه:

```
ss -dp
```

۵. نمایش سوکت های RAW با رابط شبکه مربوطه:

```
ss -ri
```

۶. نمایش سوکت های Unix با وضعیت آنها:

```
ss -xs
```

۷. نمایش آمار مربوط به ترافیک سوکت های TCP:

```
ss -tm
```

۸. نمایش اطلاعات مربوط به سوکت های IPv4 به صورت خلاصه:

```
ss -4F
```

۹. نمایش اطلاعات مربوط به سوکت های IPv6

```
ss -6o
```

۱۰. فیلتر کردن سوکت ها بر اساس پورت مبدا ۸۰:

```
ss -f :80
```

۱۱. فیلتر کردن سوکت ها بر اساس آدرس IP مقصد ۱۹۲,۱۶۸,۱,۱۰۰:

```
ss -a 192.168.1.100
```

۱۲. فیلتر کردن سوکت ها بر اساس پروتکل UDP:

```
ss -T udp
```

۱۳. فیلتر کردن سوکت ها بر اساس نام رابط شبکه eth0:

```
ss -R eth0
```

۱۴. فیلتر کردن سوکت ها بر اساس نام هاست google.com:

```
ss -L google.com
```

۱۵. ترکیب چندین معیار برای فیلتر کردن سوکت ها:

```
ss -l -t -p 1234 -a 192.168.1.100
```

۱۶. فیلتر کردن سوکت ها بر اساس شناسه فرآیند 1234 (PID):

```
ss -H 1234
```

۱۷. فیلتر کردن سوکت ها بر اساس نام سوکت "mysocket":

```
ss -N mysocket
```

۱۸. فیلتر کردن سوکت ها بر اساس وضعیت "LISTEN":

```
ss -Q LISTEN
```

۱۹. فیلتر کردن سوکت ها بر اساس پورت مبدا ۸۰ و پورت مقصد ۴۴۳:

```
ss -f :80,:443
```

۲۰. فیلتر کردن سوکت ها بر اساس آدرس IP مبدا ۲۴/۱۹۲,۱۶۸,۱,۰ و آدرس IP مقصد ۲۴/۱۰,۰,۰,۰:

```
ss -A 192.168.1.0/24 -a 10.0.0.0/24
```

۲۱. فیلتر کردن سوکت ها بر اساس پروتکل TCP و TTL 64:

```
ss -T tcp -M 64
```

۲۲. فیلتر کردن سوکت ها بر اساس timestamp 1646940800:

```
ss -I 1646940800
```

۲۳. فیلتر کردن سوکت ها بر اساس شناسه کاربر 1000 (UID):

```
ss -U 1000
```

۲۴. فیلتر کردن سوکت ها بر اساس شناسه گروه 100 (GID):

```
ss -G 100
```

لیست تمام پورت های متصال TCP وUDP

لیست تمام پورت ها (هر دو TCP و UDP) با استفاده از گزینه netstat -a.

```
# netstat-a |more
```

لیست اتصالات پورت (Transmission Control Protocol) TCPرا با استفاده از netstat -at .

```
# netstat -at
```

لیست اتصالاتUDP Ports

فقط پروتکل (UDP پروتکل دیتاگرام کاربر (را با استفاده از netstat –au فهرست کنید.

```
# netstat –au
```

لیست تمام ارتباطات listen

لیست تمام اتصالات پورت فعال listen با netstat -l .

```
# netstat -l
```

لیست تمام پورت های listen TCP

لیست تمام پورتTCP listen فعال با استفاده از گزینه netstat -lt .

```
# netstat -lt
```

```
tcp 0 0 *: dctp *: * listen
tcp 0 0 *: mysql *: * LISTEN
tcp 0 0 *: sunrpc *: * listen
tcp 0 0 *: munin *: * listen
tcp 0 0 *: ftp *: * listen
tcp 0 0 localhost.localdomain: ipp *: * listen
```

```
tcp 0 0 localhost.localdomain: smtp *: * listen
tcp 0 0 *: http *: * listen
tcp 0 0 *: ssh *: * listen
tcp 0 0 *: https *: * listen
```

لیست تمام UDP listen پورت ها

لیست تمام پورت های UDP listen فعال با استفاده از گزینه . netstat -lu

```
# netstat -lu
```

لیست تمام یونیکس های listen

لیست تمام پورت های listen یونیکس فعال با استفاده از

```
netstat –lx
```

نمایش آمار توسط پروتکل

نمایش آمار توسط پروتکل.به طور پیش فرض، آمار برای TCP ، UDP، ICMP، و پروتکل های IP نمایش داده می شود.پارامتر-sمی تواند برای تعیین مجموعه ای از پروتکل ها استفاده شود.

```
# netstat -s
```

نمایش آمار توسط پروتکلTCP

نمایش آمار فقط پروتکل TCP با استفاده از گزینه . netstat -st

```
# netstat -st
```

نمایش آمار توسط پروتکلUDP

```
# netstat -su
```

نمایش سرویس باPID

نمایش نام سرویس با شماره PID خود، با استفاده از netstat -tpگزینه / PID“ نام برنامه” را نمایش می دهد.

```
# netstat -tp
```

نمایش حالتPromiscuous

نمایش حالت مشکوک با سویچ -AC netstatچاپ اطلاعات انتخاب شده یا صفحه نمایش تازه کردن هر پنج ثانیه.بازخوانی پیش فرض صفحه در هر ثانیه.

```
# netstat -ac 5 |grep TCP
```

نمایش مسیر کرنلIP

نمایش جدول کرنل IP با دستور netstat و.route

```
# netstat -r
```

نمایش تراکنش های اینترفیس شبکه

نمایش بسته های شبکه ای از جمله انتقال و دریافت بسته با اندازه.MTU

```
# netstat –i
```

نمایش جدول رابط هسته

نمایش جدول رابطkernel ، مشابه دستور ifconfig است.

```
# netstat -ie
```

نمایش اطلاعات IPv4 وIPv6

اطلاعات عضویت گروه چندپخشی را برای هر دو IPv4 و IPv6 نمایش می دهد.

```
# netstat -g
```

چاپ اطلاعات Netstat به طور مداوم

برای دریافت اطلاعات netstat هر چند ثانیه، سپس از دستور زیر استفاده کنید، اطلاعات netstat را بصورت مداوم چاپ می کند، هر چند ثانیه می گویند.

```
# netstat -c
```

یافتن برنامه های listen

یافتن چند برنامه listen که در یک پورت اجرا می شوند.

```
# netstat -ap | grep http
```

دستور sar

دستور sar در سیستم عامل‌های Unix/Linux برای نمایش و ثبت اطلاعات سیستم و عملکرد سرور استفاده می‌شود. این دستور به شما اجازه می‌دهد تا اطلاعات مختلفی مانند بار CPU، مصرف حافظه، ورودی/خروجی دیسک و ... را بررسی کنید.

مثال های دستور:

۱. نمایش راهنمای دستور:

```
sar -h
```

۲. نمایش نسخه sar:

```
sar -V
```

۳. جمع آوری و نمایش آمار CPU به مدت ۵ بار:

```
sar -u 5
```

۴. نمایش ۳ بار اول آمار جمع آوری شده CPU:

```
sar -c 3
```

۵. جمع آوری و نمایش آمار CPU با فاصله زمانی ۲ ثانیه:

```
sar -i 2
```

۶. جمع آوری و نمایش مداوم آمار CPU:

```
sar -t
```

۷. نمایش خلاصه آمار CPU:

```
sar -s
```

۸. نمایش جزئیات آمار CPU:

```
sar -o
```

۹. ذخیره آمار CPU در فایل csv:

```
sar -f
```

۱۰. ذخیره آمار CPU در فایل pcap:

```
sar -P
```

۱۱. نمایش آمار CPU در قالب html:

```
sar -b
```

۱۲. نمایش آمار CPU در قالب نمودار:

```
sar -g
```

۱۳. فیلتر کردن آمار CPU بر اساس نام رابط شبکه eth0:

```
sar -R eth0
```

۱۴. فیلتر کردن آمار CPU بر اساس نام هاست google.com:

```
sar -L google.com
```

۱۵. فیلتر کردن آمار CPU بر اساس شناسه فرآیند (PID) 1234:

```
sar -H 1234
```

۱۶. فیلتر کردن آمار CPU بر اساس پورت مبدا ۸۰:

```
sar -D 80
```

۱۷. فیلتر کردن آمار CPU بر اساس آدرس IP مقصد ۱۹۲,۱۶۸,۱,۱۰۰:

```
sar -a 192.168.1.100
```

۱۸. فیلتر کردن آمار CPU بر اساس پروتکل TCP:

```
sar -T tcp
```

۱۹. فیلتر کردن آمار CPU بر اساس TTL 64:

```
sar -M 64
```

۲۰. فیلتر کردن آمار CPU بر اساس timestamp 1646940800:

```
sar -I 1646940800
```

۲۱. فیلتر کردن آمار CPU بر اساس شناسه کاربر (UID) 1000:

```
sar -U 1000
```

۲۲. فیلتر کردن آمار CPU بر اساس شناسه گروه (GID) 100:

```
sar -G 100
```

۲۳. ترکیب چند معیار برای فیلتر کردن آمار CPU:

```
sar -u 5 -R eth0 -T tcp
```

۲۴. ذخیره آمار CPU به صورت csv با فیلتر بر اساس پورت مقصد ۸۰:

```
sar -f -d 80
```

۲۵. نمایش نمودار آمار CPU با فیلتر بر اساس نام هاست google.com:

```
sar -g -L google.com
```

دستور dstat

دستور dstat در سیستم‌عامل‌های Unix/Linux برای نمایش اطلاعات سیستمی و عملکرد سیستم به صورت زنده و آمارهای مختلف مانند CPU، حافظه، دیسک و شبکه استفاده می‌شود.

گزینه‌های دستور :

گزینه های کلی

- a نمایش میانگین معیارها در هر ثانیه

- c نمایش معیارها به صورت نمودار
- g نمایش معیارها به صورت گرافیکی
- h نمایش راهنمای دستور
- n تعداد دفعات جمع آوری معیارها
- p pid فرآیندی که می خواهید معیارهای آن را مشاهده کنید
- t فاصله زمانی بین جمع آوری معیارها (بر حسب ثانیه)
- v نمایش جزئیات بیشتر

گزینه های مربوط به معیارها

- cpu نمایش معیارهای مربوط به CPU
- disk نمایش معیارهای مربوط به دیسک
- mem نمایش معیارهای مربوط به حافظه
- net نمایش معیارهای مربوط به شبکه
- proc نمایش معیارهای مربوط به فرآیندها

مثال های دستور dstat

گزینه های کلی:

- a: نمایش میانگین معیارها در هر ثانیه

```
dstat -a -n 10 -t 1 cpu,mem,net
```

این دستور ۱۰ بار معیارهای مربوط به CPU، حافظه و شبکه را با فاصله زمانی ۱ ثانیه جمع آوری و میانگین آنها را در هر ثانیه نمایش می دهد.

- c: نمایش معیارها به صورت نمودار

```
dstat -c -t 2 cpu,mem
```

این دستور معیارهای مربوط به CPU و حافظه را به صورت نمودار با فاصله زمانی ۲ ثانیه نمایش می دهد.

- g: نمایش معیارها به صورت گرافیکی

```
dstat -g -t 3 cpu,disk
```

این دستور معیارهای مربوط به CPU و دیسک را به صورت گرافیکی با فاصله زمانی ۳ ثانیه نمایش می دهد.

- h: نمایش راهنمای دستور

```
dstat -h
```

این دستور راهنمای کامل دستور dstat را نمایش می دهد.

- n: تعداد دفعات جمع آوری معیارها

```
dstat -n 5 cpu
```

این دستور ۵ بار معیارهای مربوط به CPU را جمع آوری و نمایش می دهد.

- p: pid فرآیندی که می خواهید معیارهای آن را مشاهده کنید

```
dstat -p 1234
```

این دستور معیارهای مربوط به فرآیندی با pid 1234 را نمایش می دهد.

- t: فاصله زمانی بین جمع آوری معیارها (بر حسب ثانیه)

```
dstat -t 5 mem,net
```

این دستور معیارهای مربوط به حافظه و شبکه را با فاصله زمانی ۵ ثانیه جمع آوری و نمایش می دهد

- v: نمایش جزئیات بیشتر

```
dstat -v -t 1 cpu,disk
```

این دستور معیارهای مربوط به CPU و دیسک را با جزئیات بیشتر و با فاصله زمانی ۱ ثانیه نمایش می دهد.

گزینه های مربوط به معیارها:

- cpu: نمایش معیارهای مربوط به CPU

```
dstat -cpu
```

این دستور تمام معیارهای مربوط به CPU را نمایش می دهد.

- disk: نمایش معیارهای مربوط به دیسک

```
dstat -disk -t 2
```

این دستور معیارهای مربوط به دیسک را با فاصله زمانی ۲ ثانیه نمایش می دهد.

- mem: نمایش معیارهای مربوط به حافظه

```
dstat -mem -v
```

این دستور معیارهای مربوط به حافظه را با جزئیات بیشتر نمایش می دهد.

- net: نمایش معیارهای مربوط به شبکه

```
dstat -net -p 1234
```

این دستور معیارهای مربوط به شبکه را برای فرآیندی با pid 1234 نمایش می دهد.

- proc: نمایش معیارهای مربوط به فرآیندها

```
dstat -proc -t 3
```

این دستور معیارهای مربوط به تمام فرآیندها را با فاصله زمانی ۳ ثانیه نمایش می دهد.

نکته:

می توانید از چندین گزینه به طور همزمان استفاده کنید. برای مثال:

```
dstat -a -c -g -t 2 cpu,mem,net
```

این دستور ۱۰ بار معیارهای مربوط به CPU، حافظه و شبکه را با فاصله زمانی ۲ ثانیه جمع آوری و به صورت نمودار، گراف و میانگین در هر ثانیه نمایش می دهد.

دستور nmap

دستور nmap یک ابزار اسکن شبکه قدرتمند است که در سیستم‌عامل‌های مختلف مانند لینوکس و ویندوز قابل استفاده است. این ابزار به شما کمک می‌کند تا اطلاعات مربوط به دستگاه‌ها، پورت‌ها، سرویس‌ها و تنظیمات امنیتی شبکه را بررسی کنید.

گزینه های دستور:

گزینه های اسکن:

- a نمایش میانگین معیارها در هر ثانیه
- dstat a n 10 t 1 cpu,mem,net
- این دستور ۱۰ بار معیارهای مربوط به CPU، حافظه و شبکه را با فاصله زمانی ۱ ثانیه جمع آوری و میانگین آنها را در هر ثانیه نمایش می دهد.
- c نمایش معیارها به صورت نمودار
- dstat c t 2 cpu,mem
- این دستور معیارهای مربوط به CPU و حافظه را به صورت نمودار با فاصله زمانی ۲ ثانیه نمایش می دهد.

- g نمایش معیارها به صورت گرافیکی
- dstat g t 3 cpu,disk
- این دستور معیارهای مربوط به CPU و دیسگزینه های اسکن
- sT اسکن) TCP SYNنیمه باز)
- sS اسکن TCP SYN مخفی
- sU اسکن UDP
- sF اسکن TCP FIN
- sX اسکن TCP Xmas
- sN اسکن TCP Null
- A اسکن جامع (شامل تمام اسکن های)TCP

گزینه های هدف

- h آدرس IP یا نام هاست
- n اسکن بدون نامگذاری (فقط آدرس)IP
- p پورت یا محدوده پورت برای اسکن
- exclude آدرس IP یا نام هاست برای حذف از اسکن
- range محدوده آدرس IP برای اسکن

گزینه های خروجی

- v افزایش جزئیات خروجی
- vv افزایش بیشتر جزئیات خروجی
- oX ذخیره خروجی در فرمت XML
- oG ذخیره خروجی در فرمت Greppable
- oA ذخیره خروجی در تمام فرمت های پیش فرض

گزینه های پیشرفته

- T زمان انتظار برای پاسخ (بر حسب ثانیه)
- r تعداد دفعات ارسال درخواست
- version نمایش نسخه Nmap
- h نمایش راهنمای دستور

مثال های دستور

گزینه های اسکن

sT اسکن TCP SYNنیمه باز)

```
nmap -sT 192.168.1.100
```

این دستور هاست ۱۹۲,۱۶۸,۱,۱۰۰ را با اسکن) TCP SYNنیمه باز) اسکن می کند.

sS اسکن TCP SYN مخفی

```
nmap -sS example.com
```

این دستور هاست example.com را با اسکن TCP SYN مخفی اسکن می کند.

sU اسکن UDP

```
nmap -sU -p 53 192.168.1.0/24
```

این دستور تمام هاست های موجود در شبکه ۲۴/۱۹۲,۱۶۸,۱,۰ را برای پورت (DNS) 53 UDP اسکن می کند.

sF اسکن TCP FIN

```
nmap -sF -p 80,443 10.0.0.1
```

این دستور پورت های ۸۰ و ۴۴۳ را در هاست ۱۰,۰,۰,۱ با اسکن TCP FIN اسکن می کند.

sX اسکن TCP Xmas

```
nmap -sX 192.168.1.1-10
```

این دستور هاست های ۱۹۲,۱۶۸,۱,۱ تا ۱۹۲,۱۶۸,۱,۱۰ را با اسکن TCP Xmas اسکن می کند.

sN اسکن TCP Null

```
nmap -sN example.com -p 22
```

این دستور پورت ۲۲ را در هاست example.com با اسکن TCP Null اسکن می کند.

A اسکن جامع (شامل تمام اسکن های TCP)

```
nmap -A 192.168.1.100
```

این دستور هاست ۱۹۲,۱۶۸,۱,۱۰۰ را با اسکن جامع (شامل تمام اسکن های TCP) اسکن می کند.

گزینه های هدف

h آدرس IP یا نام هاست

```
nmap -h 192.168.1.100
```

این دستور اطلاعات مربوط به هاست ۱۹۲,۱۶۸,۱,۱۰۰ را نمایش می دهد.

n اسکن بدون نامگذاری (فقط آدرس IP)

```
nmap -n 10.0.0.1-254
```

این دستور تمام هاست های موجود در محدوده ۱۰,۰,۰,۱ تا ۱۰,۰,۰,۲۵۴ را بدون نامگذاری اسکن می کند.

p پورت یا محدوده پورت برای اسکن

```
nmap -p 80-85 example.com
```

این دستور پورت های ۸۰ تا ۸۵ را در هاست example.com اسکن می کند.

exclude آدرس IP یا نام هاست برای حذف از اسکن

```
nmap -A 192.168.1.0/24 --exclude 192.168.1.100
```

این دستور تمام هاست های موجود در شبکه ۲۴/۱۹۲,۱۶۸,۱,۰ را به جز هاست ۱۹۲,۱۶۸,۱,۱۰۰ با اسکن جامع اسکن می کند.

range محدوده آدرس IP برای اسکن

```
nmap -sS --range 192.168.1.1-10
```

این دستور هاست های ۱۹۲,۱۶۸,۱,۱ تا ۱۹۲,۱۶۸,۱,۱۰ را با اسکن TCP SYN مخفی اسکن می کند.

گزینه های خروجی

v افزایش جزئیات خروجی

```
nmap -v example
```

دستور snort

دستور snort یک سیستم تشخیص نفوذ (Intrusion Detection System) متن‌باز است که برای مانیتورینگ و تشخیص حملات شبکه استفاده می‌شود. snort قادر است به طور زنده ترافیک شبکه را بررسی کرده و الگوهای مشخص حملات را شناسایی کند.

گزینه های دستور:

گزینه های عمومی

- h نمایش راهنمای دستور
- V نمایش نسخه Snort
- c مسیر فایل پیکربندی Snort
- i رابط شبکه برای نظارت
- l سطح گزارش دهی (۰۹)
- A فعال کردن حالت پیشرفته
- T فعال کردن حالت تست
- u کاربر برای اجرای Snort
- g گروه برای اجرای Snort

گزینه های مربوط به قوانین

- r مسیر فایل قوانین Snort
- s فعال کردن قوانین پیش فرض Snort
- p فعال کردن قوانین مربوط به یک پروتکل خاص
- D غیرفعال کردن قوانین مربوط به یک پروتکل خاص
- e فعال کردن موتور قاعده گذاری عمیق (Deep Packet Inspection)

گزینه های مربوط به هشدار

- a فعال کردن هشدار صوتی
- b فعال کردن هشدار بصری
- w ارسال هشدار به یک آدرس ایمیل
- m ارسال هشدار به یک syslog

مثال های دستور:

۱. شروع snort بدون ذخیره لاگ:

این دستور برای شروع اجرای snort بدون ذخیره سازی لاگ‌ها استفاده می‌شود.

```
snort -q -c /etc/snort/snort.conf
```

۲. شروع snort با ذخیره لاگ در یک فایل:

این دستور برای شروع اجرای snort و ذخیره سازی لاگ‌ها در یک فایل خاص استفاده می‌شود.

```
snort -q -c /etc/snort/snort.conf -l /var/log/snort/
```

۳. استفاده از حالت verbose برای نمایش جزئیات:

این دستور برای اجرای snort با حالت verbose و نمایش جزئیات بیشتر استفاده می‌شود.

```
snort -q -c /etc/snort/snort.conf -v
```

۴. فیلتر کردن ترافیک بر اساس آدرس IP:

این دستور برای فیلتر کردن ترافیک بر اساس آدرس IP مشخص استفاده می‌شود.

```
snort -q -c /etc/snort/snort.conf host 192.168.1.1
```

۵. استفاده از قوانین سفارشی:

این دستور برای اجرای snort با استفاده از قوانین سفارشی که در فایل خاصی ذخیره شده‌اند استفاده می‌شود.

```
snort -q -c custom_rules.conf
```

۶. نمایش آمارهای تشخیص نفوذ:

این دستور برای نمایش آمارها و اطلاعات مربوط به تشخیص حملات توسط snort استفاده می‌شود.

```
snort -q -c /etc/snort/snort.conf -s
```

مثال های بیشتر:

hنمایش راهنمای دستور

```
snort -h
```

این دستور راهنمای کامل دستور Snort را نمایش می دهد.

Vنمایش نسخه Snort

```
snort -V
```

این دستور نسخه Snort را نمایش می دهد.

cمسیر فایل پیکربندی Snort

```
snort -c /etc/snort/snort.conf
```

این دستور Snort را با استفاده از فایل پیکربندی /etc/snort/snort.conf اجرا می کند.

iرابط شبکه برای نظارت

```
snort -i eth0
```

این دستور Snort را برای نظارت بر رابط شبکه eth0 اجرا می کند.

lسطح گزارش دهی (۰-۹)

```
snort -l 7
```

این دستور Snort را با سطح گزارش دهی ۷ اجرا می کند.

Aفعال کردن حالت پیشرفته

```
snort -A
```

این دستور Snort را با فعال کردن حالت پیشرفته اجرا می کند.

Tفعال کردن حالت تست

```
snort -T
```

این دستور Snort را با فعال کردن حالت تست اجرا می کند.

uکاربر برای اجرای Snort

```
snort -u snortuser
```

این دستور Snort را با استفاده از کاربر snortuser اجرا می کند.

gگروه برای اجرای Snort

```
snort -g snortgroup
```

این دستور Snort را با استفاده از گروه snortgroup اجرا می کند.

گزینه های مربوط به قوانین

rمسیر فایل قوانین Snort

```
snort -r /etc/snort/rules/local.rules
```

این دستور Snort را با استفاده از قوانین محلی در /etc/snort/rules/local.rules اجرا می کند.

sفعال کردن قوانین پیش فرض Snort

```
snort -s
```

این دستور Snort را با فعال کردن قوانین پیش فرض Snort اجرا می کند.

pفعال کردن قوانین مربوط به یک پروتکل خاص

```
snort -p tcp
```

این دستور Snort را با فعال کردن قوانین مربوط به پروتکل TCP اجرا می کند.

Dغیرفعال کردن قوانین مربوط به یک پروتکل خاص

```
snort -D icmp
```

این دستور Snort را با غیرفعال کردن قوانین مربوط به پروتکل ICMP اجرا می کند.

eفعال کردن موتور قاعده گذاری عمیق (Deep Packet Inspection)

```
snort -e
```

این دستور Snort را با فعال کردن موتور قاعده گذاری عمیق (Deep Packet Inspection) اجرا می کند.

گزینه های مربوط به هشدار

aفعال کردن هشدار صوتی

```
snort -a
```

این دستور Snort را با فعال کردن هشدار صوتی اجرا می کند.

bفعال کردن هشدار بصری

```
snort -b
```

این دستور Snort را با فعال کردن هشدار بصری اجرا می کند.

wارسال هشدار به یک آدرس ایمیل

```
snort -w user@example.com
```

این دستور Snort را با فعال کردن ارسال هشدار به آدرس ایمیل user@example.com اجرا می کند.

mارسال هشدار به یک syslog

```
snort -m
```

دستور nc

دستور Netcatیا nc یک ابزار خط فرمان است که داده ها را از طریق اتصالات شبکه با استفاده از پروتکل های TCP یا UDP می خواند و می نویسد. این یکی از قدرتمندترین ابزارها در شبکه و سیستم است و به عنوان یک همه کاره محسوب می شود. Netcat چند پلتفرمی است و برای لینوکس ، macOS ، ویندوز و BSD در دسترس است. می توانید از Netcat برای اشکال زدایی و نظارت بر اتصالات شبکه ، اسکن پورت های باز ، انتقال داده ها ، به عنوان پروکسی و موارد دیگر استفاده کنید. بسته Netcat از قبل روی macOS و توزیع های محبوب لینوکس مانند اوبونتو ، دبیان یا CentOS نصب شده است.

گزینه های دستور:

- h نمایش راهنمای دستور
- V نمایش نسخه Netcat
- v افزایش جزئیات خروجی
- q غیرفعال کردن خروجی
- l گوش دادن به اتصالات ورودی
- p شماره پورت برای اتصال یا گوش دادن
- i آدرس IP برای اتصال یا گوش دادن
- u استفاده از پروتکل UDP
- t فعال کردن حالت Timeout
- w زمان انتظار برای اتصال (بر حسب ثانیه)
- گزینه های مربوط به اتصال
- c دستور برای اجرا پس از برقراری اتصال
- e ارسال داده های اولیه به سرور
- ذخیره خروجی در فایل
- x استفاده از پروکسی SOCKS

ساختار دستور:

```
$ nc [options] [ip] [port]
```

مثال های دستور:

۱.یک اتصال TCP به پورت ۸۰ میزبان، با استفاده از پورت ۱۳۳۷ assource پورت با وقفه زمانی ۵ ثانیه باز میکند:

```
$ nc -p 1337 -w 5 host.ip 80
```

۲. یک اتصال UDP به پورت ۸۰ روی هاست باز میکند:

```
$ nc -u host.ip 80
```

۳. ایجاد و گوش دادن در سوکت جریان دامنه یونیکس:

```
$ nc -lU /var/tmp/dsocket
```

٤. یک مدل پایه سرور/کلینت ایجاد کنید:

این یک اتصال ایجاد می‌کند، بدون هیچ سمت سرور/کلینت خاصی با توجه به nc، پس از برقراری ارتباط.

```
$ nc -l 1234 # in one console
$ nc 127.0.0.1 1234 # in another console
```

۵. یک مدل پایه انتقال داده میسازد:

پس از انتقال فایل، به صورت متوالی، اتصال به طور خودکار بسته می شود

```
$ nc -l 1234 > filename.out # to start listening in one console and collect data $ nc host.ip 1234 < filename.in
```

٦. با سرورها ارتباط برقرارکنید:

مثال اصلی بازیابی صفحه اصلی هاست به همراه هدرها.

```
$ printf “GET / HTTP/1.0\r\n\r\n” | nc host.ip 80
```

۷. اسکن پورت:

بررسی اینکه کدام پورت ها باز هستند و سرویس ها را در ماشین های مورد نظر اجرا می کنند. دستورات پرچم z – برای اطلاع رسانی در مورد آنها به جای شروع یک اتصال.

```
$ nc -zv host.ip 20-2000 # range of ports to check for
```

دستور iwconfig

دستور iwconfig در سیستم‌عامل‌های Unix/Linux برای تنظیم و مدیریت تنظیمات شبکه بی‌سیم (Wi-Fi) استفاده می‌شود. این دستور به شما امکان می‌دهد تا اطلاعات مربوط به رابط‌های شبکه بی‌سیم را مشاهده کنید، تنظیمات آن‌ها را تغییر دهید و ارتباط با شبکه‌های بی‌سیم را مدیریت کنید.

گزینه های دستور:

- h نمایش راهنمای دستور
- V نمایش نسخه iwconfig
- a نمایش اطلاعات مربوط به تمام رابط های بی سیم
- i نام رابط بی سیم برای پیکربندی

گزینه های مربوط به تنظیمات

- essid تنظیم (SSIDنام شبکه)
- channel تنظیم کانال
- mode تنظیم مد (مانند Adhoc یا Infrastructure)
- rate تنظیم سرعت بیت
- power تنظیم قدرت سیگنال
- ap تنظیم آدرس MAC نقطه دسترسی
- key تنظیم کلید رمزگذاری

گزینه های مربوط به آمار

- stats نمایش آمار مربوط به رابط بی سیم
- link نمایش اطلاعات مربوط به لینک بی سیم
- txpower نمایش قدرت سیگنال ارسال شده
- rxpower نمایش قدرت سیگنال دریافت شده

مثال های دستور:

```
iwconfig wlan0 essid “MyNetwork”
```

این دستور SSID رابط بی سیم wlan0 را به “ ”MyNetworkتنظیم می کند.

مثال:

```
iwconfig wlan0 channel 6
```

این دستور کانال رابط بی سیم wlan0 را به ۶ تنظیم می کند.

مثال:

```
iwconfig wlan0 mode ad-hoc
```

این دستور مد رابط بی سیم wlan0 را به Ad-hoc تنظیم می کند.

مثال:

iwconfig wlan0 rate 54M

این دستور سرعت بیت رابط بی سیم wlan0 را به ۵۴ مگابیت بر ثانیه تنظیم می کند.

مثال:

iwconfig wlan0 power 15

این دستور قدرت سیگنال رابط بی سیم wlan0 را به ۱۵ تنظیم می کند.

مثال:

iwconfig wlan0 ap 00:11:22:33:44:55

این دستور آدرس MAC نقطه دسترسی رابط بی سیم wlan0 را به ۰۰:۱۱:۲۲:۳۳:۴۴:۵۵ تنظیم می کند.

مثال:

iwconfig wlan0 encryption WPA2

این

مثال:

iwconfig wlan0 key “mypassword”

این دستور کلید رمزگذاری رابط بی سیم wlan0 را به “ ”mypasswordتنظیم می کند.

مثال:

iwconfig wlan0 stats

این دستور آمار مربوط به رابط بی سیم wlan0 را نمایش می دهد.

مثال:

iwconfig wlan0 link

این دستور اطلاعات مربوط به لینک بی سیم رابط wlan0 را نمایش می دهد.

مثال:

iwconfig wlan0 txpower

این دستور قدرت سیگنال ارسال شده توسط رابط بی سیم wlan0 را نمایش می دهد.

مثال:

iwconfig wlan0 rxpower

این دستور قدرت سیگنال دریافت شده توسط رابط بی سیم wlan0 را نمایش می دهد.

دستور iwlist

دستور iwlist نیز مانند iwconfig برای مدیریت شبکه بی‌سیم در سیستم‌عامل‌های Unix/Linux استفاده می‌شود. با استفاده از این دستور، می‌توانید اطلاعات جزئی‌تری در مورد شبکه‌های بی‌سیم در دسترس را مشاهده کنید، اطلاعاتی مانند لیست شبکه‌ها، کیفیت سیگنال، فرکانس و غیره را به شما نمایش می‌دهد.

گزینه های دستور:

گزینه های عمومی

- h نمایش راهنمای دستور
- v افزایش جزئیات خروجی
- i رابط وایرلس مورد نظر
- گزینه های مربوط به اسکن

- scan اسکن شبکه های وای فای
- channel اسکن کانال های وای فای
- essid اسکن SSID های وای فای
- bitrate اسکن بیت ریت های وای فای

گزینه های مربوط به اطلاعات

- link نمایش اطلاعات مربوط به اتصال فعلی
- stats نمایش آمار مربوط به اتصال فعلی
- power نمایش اطلاعات مربوط به قدرت سیگنال
- rate نمایش اطلاعات مربوط به سرعت اتصال

گزینه های مربوط به تنظیمات

- freq تنظیم فرکانس رابط وایرلس
- channel تنظیم کانال رابط وایرلس
- essid تنظیم SSID رابط وایرلس
- mode تنظیم مد رابط وایرلس (مانند adhoc یا infrastructure)

گزینه های مربوط به عیب یابی

- txpower بررسی قدرت ارسال سیگنال
- rxpower بررسی قدرت دریافت سیگنال
- noise بررسی میزان نویز
- retry بررسی تعداد تلاش های مجدد برای ارسال بسته ها

مثال های دستور :

مثال:

```
iwlist wlan0 scan
```

این دستور شبکه های وای فای موجود را اسکن می کند و اطلاعات مربوط به آنها را نمایش می دهد.

مثال:

```
iwlist wlan0 link
```

این دستور اطلاعات مربوط به اتصال فعلی وای فای را نمایش می دهد.

مثال:

```
iwlist wlan0 essid "SSID مورد نظر"
```

این دستور اطلاعات مربوط به SSID مورد نظر را نمایش می دهد.

مثال:

```
iwlist wlan0 power
```

این دستور اطلاعات مربوط به قدرت سیگنال وای فای را نمایش می دهد.

مثال:

```
iwlist wlan0 freq 2462
```

این دستور فرکانس رابط وایرلس را روی ۲۴۶۲ مگاهرتز تنظیم می کند.

مثال:

```
iwlist wlan0 channel 11
```

این دستور کانال رابط وایرلس را روی ۱۱ تنظیم می کند.

مثال:

```
iwlist wlan0 mode ad-hoc
```

این دستور مد رابط وایرلس را روی ad-hoc تنظیم می کند.

مثال:

```
iwlist wlan0 txpower
```

این دستور قدرت ارسال سیگنال رابط وایرلس را بررسی می کند.

برای مشاهده لیست کامل گزینه ها و توضیحات مربوط به هر یک از آنها می توانید از دستور زیر استفاده کنید:

```
iwlist -h
```

دستور RFKILL

بسیاری از سیستم‌های کامپیوتری حاوی فرستنده‌های رادیویی از جمله Wi-Fi ، بلوتوث و دستگاه‌های 3Gهستند .این دستگاه ها انرژی مصرف می کنند که در صورت عدم استفاده از دستگاه هدر می رود. RFKillیک زیر سیستم در هسته لینوکس است که رابطی را فراهم می کند که از طریق آن فرستنده های رادیویی در یک سیستم کامپیوتری می توانند جستجو، فعال و غیرفعال شوند .هنگامی که فرستنده ها غیرفعال می شوند، می توان آنها را در حالتی قرار داد که نرم افزار بتواند آنها را واکنش نشان دهد (یک بلوک نرم (یا جایی که نرم افزار نتواند آنها را فعال کند (یک بلوک سخت)

هسته RFKill API را برای زیرسیستم فراهم می کند .درایورهای کرنل که برای پشتیبانی از RFkill طراحی شده اند، از این API برای ثبت در هسته استفاده می کنند و شامل روش هایی برای فعال و غیرفعال کردن دستگاه می شود .علاوه بر این، هسته RFKill اعلان‌هایی را ارائه می‌کند که برنامه‌های کاربردی کاربر می‌توانند آن‌ها را تفسیر کنند و روش‌هایی را برای برنامه‌های کاربر برای درخواست وضعیت فرستنده ارائه می‌دهد. رابط RFKill در آن قرار دارد /dev/rfkill که شامل وضعیت فعلی تمام فرستنده های رادیویی روی سیستم است .هر دستگاه دارای وضعیت RFKill فعلی خود است که در sysfs. علاوه بر این، RFKillبرای هر تغییر حالت در یک دستگاه دارایRFKill ، رویدادهایی را صادر می کند. rfkillیک ابزار خط فرمان است که با آن می توانید دستگاه های دارای RFKillرا در سیستم پرس و جو و تغییر دهید .برای به دست آوردن ابزار،بسته rfkillرا نصب کنید . از دستور rfkill listبرای به دست آوردن لیستی از دستگاه ها استفاده کنید که هر کدام دارای یک شماره فهرست مرتبط با آن هستند که از 0. می‌توانید از این شماره فهرست استفاده کنید تا به rfkillبگویید دستگاهی را مسدود یا رفع انسداد کند.

گزینه های دستور:

- h نمایش راهنمای دستور
- l لیست تمام رابط‌های بی‌سیم موجود را نمایش می‌دهد.
- i شناسه رابط بی‌سیم مورد نظر را برای فعال یا غیرفعال کردن آن مشخص می‌کند.
- d رابط بی‌سیم را غیرفعال می‌کند.
- u رابط بی‌سیم را فعال می‌کند.
- t نوع رابط بی‌سیم مورد نظر را برای لیست کردن مشخص می‌کند.
- s وضعیت فعلی رابط بی‌سیم را نمایش می‌دهد.

مثال: مسدود کردن یک دستگاه دارایRFKill

دستور زیر را برای مسدود کردن اولین دستگاه دارای RFKill در سیستم اجرا کنید:

```
rfkill block 0
```

همچنین می توانید از **rfkill**برای مسدود کردن دسته خاصی از دستگاه ها استفاده کنید .این دستور تمام دستگاه های Wi-Fi سیستم را مسدود می کند:

```
rfkill block wifi
```

برای مسدود کردن همه دستگاه‌های دارایRFKill ، اجرا کنید:

```
rfkill block all
```

برای رفع انسداد دستگاه‌های دارایRFKill ، اجرا کنید:

```
rfkill unblock
```

برای به دست آوردن یک لیست کامل از دسته های دستگاه هایی که rfkillمی تواند مسدود کند، اجرا کنید:

```
rfkill help
```

برای اطلاعات بیشتر در مورد rkfill، به صفحه man rfkill(1) مراجعه کنید.

دستورdig

Dig ابزاری انعطاف پذیر برای بازجویی از سرورهای نام DNS است. جستجوهای DNS را انجام می‌دهد و پاسخ‌هایی را که از سرور(های) نامی که درخواست شده است، بازگردانده می‌شود.

کاربردهای dig

- عیب یابی مشکلات مربوط به DNS
- جستجوی رکوردهای DNS خاص برای یک نام دامنه
- بررسی اطلاعات مربوط به سرورهای نام (Name Servers)
- مشاهده اطلاعات مربوط به رکوردهای MX (Mail Exchanger)
- تست و بررسی پیکربندی DNS

گزینه های دستور:

- h نمایش راهنمای دستور
- v افزایش جزئیات خروجی
- x تبدیل آدرس IP به نام دامنه
- t نوع رکورد DNS مورد نظر برای جستجو (مانند A، MX، NS)
- f جستجو برای رکوردهای DNS در یک فایل خاص
- q نام دامنه مورد نظر برای جستجو

گزینه های پیشرفته

- ۴ استفاده از DNS IPv4
- ۶ استفاده از DNS IPv6
- p پورت سرور DNS
- y نوع پرس و جو (مانند ANY، AAAA)
- k کلید TSIG برای احراز هویت

مثال:

```
dig google.com
```

این دستور اطلاعات کلی مربوط به نام دامنه google.com را نمایش می دهد.

مثال:

```
dig -t A google.com
```

این دستور فقط رکوردهای A آدرس IP برای google.com را نمایش می دهد.

مثال:

```
dig -t MX google.com
```

این دستور فقط رکوردهای MX (Mail Exchanger) برای google.com را نمایش می دهد.

مثال:

```
dig -x 8.8.8.8
```

این دستور نام دامنه مربوط به آدرس IP 8.8.8.8 را نمایش می دهد.

مثال:

```
dig -f /etc/resolv.conf google.com
```

این دستور اطلاعات مربوط به google.com را با استفاده از فایل پیکربندی /etc/resolv.conf DNS جستجو می کند.

مثال:

```
dig -4 google.com
```

این دستور اطلاعات مربوط به google.com را با استفاده از DNS IPv4 جستجو می کند.

مثال:

```
dig -6 google.com
```

این دستور اطلاعات مربوط به google.com را با استفاده از DNS IPv6 جستجو می کند.

مثال:

```
dig -p 53 google.com
```

این دستور اطلاعات مربوط به google.com را با استفاده از پورت ۵۳ پورت پیش فرض (DNS) جستجو می کند.

مثال:

```
dig -y ANY google.com
```

این دستور تمام رکوردهای DNS موجود برای google.com را نمایش می دهد.

دستور drill

دستور drill یک ابزار کامندلاین برای انجام پرس و جو DNS (Domain Name System) است. این ابزار امکان بررسی رکوردهای DNS، انجام پرس و جوهای مختلف و تست نام‌های دامنه را فراهم می‌کند.همچنین Drill ابزاری قدرتمند و انعطاف‌پذیر برای آزمایش و عیب‌یابی سرورهای DNS و حل نام‌ها در شبکه است. این ابزار طیف گسترده‌ای از پرس و جوهای DNS را پشتیبانی می‌کند و می‌تواند برای تست عملکرد، امنیت و پیکربندی سرورهای DNS مورد استفاده قرار گیرد.

گزینه‌های دستور:

- h نمایش راهنمای دستور
- v نمایش نسخه Drill
- c فایل پیکربندی Drill
- q نوع پرس و جو DNS (A, AAAA, MX, CNAME, etc.)
- n نام دامنه یا آدرس IP برای پرس و جو

- t نوع رکورد DNS (SOA, NS, PTR, etc.)
- u سرور DNS برای ارسال پرس و جو
- p پورت سرور DNS
- f فرمت خروجی ((text, json, etc.
- گزینه‌های پیشرفته
- ۴ استفاده از IPv4
- ۶ استفاده از IPv6
- d فعال کردن دیباگ
- i تعداد تکرار پرس و جو
- T زمان انتظار برای پاسخ (بر حسب ثانیه)
- r فعال کردن دنبال کردن CNAME

مثال های دستور:

نمایش اطلاعات مربوط به رکورد PTR برای یک آدرس IP:

```
drill -x 8.8.8.8
```

تعیین نوع پرس و جو برای DNS:

```
drill example.com -t A
```

تعیین کلاس پرس و جو برای DNS:

```
drill example.com -c IN
```

تعیین پورت برای پرس و جو:

```
drill example.com -p 53
```

تنظیم بایند ادرس برای پرس و جو:

```
drill example.com -b 192.168.1.1
```

استفاده از IPv4 برای پرس و جو:

```
drill example.com -4
```

استفاده از IPv6 برای پرس و جو:

```
drill example.com -6
```

نمایش اطلاعات بیشتر و حالت verbose:

```
drill example.com -v
```

نمایش جزئیات بیشتر پرس و جو

```
drill example.com -D
```

مثال:

```
drill -q A google.com
```

این دستور یک پرس و جو DNS از نوع A برای نام دامنه google.com ارسال می‌کند.

مثال:

```
drill -t MX example.com
```

این دستور رکوردهای MX را برای نام دامنه example.com نمایش می‌دهد.

مثال:

```
drill -u 192.168.1.1 -q AAAA www.example.com
```

این دستور یک پرس و جو DNS از نوع AAAA برای نام دامنه www.example.com به سرور DNS با آدرس IP 192.168.1.1 ارسال می‌کند.

مثال:

```
drill -f json -q CNAME wikipedia.org
```

این دستور رکوردهای CNAME را برای نام دامنه wikipedia.org در قالب JSON نمایش می‌دهد.

برای مشاهده لیست کامل گزینه‌ها و توضیحات مربوط به هر یک از آنها می‌توانید از دستور زیر استفاده کنید:

```
drill -h
```

دستور xdg-open

برای بازکردن یک url تحت ترمینال از دستور زیر استفاده نمایید.همچنین xdg-openابزاری در سیستم‌عامل‌های لینوکس است که برای باز کردن فایل‌ها وURL ها با برنامه‌های پیش‌فرض مرتبط با آنها استفاده می‌شود. این ابزار به شما امکان می‌دهد تا بدون نیاز به دانستن نام برنامه مربوطه، فایل‌ها را به طور مستقیم باز کنید.

گزینه های دستور:

- h نمایش راهنمای دستور
- a مشخص کردن برنامه‌ای برای باز کردن فایل
- e باز کردن فایل با استفاده از یک ویرایشگر متن
- r باز کردن فایل با استفاده از یک برنامه ترمینال
- t باز کردن URL با استفاده از یک مرورگر وب
- f باز کردن فایل بدون بررسی نوع آن
- version نمایش نسخه xdgopen

مثال های دستور:

مثال:

```
xdg-open file.txt
```

این دستور فایل file.txt را با برنامه پیش‌فرض مرتبط با آن باز می‌کند.

مثال:

```
xdg-open -a firefox https://www.google.com
```

این دستور URL https://www.google.com را با استفاده از مرورگر Firefox باز می‌کند.

مثال:

```
xdg-open -e file.txt
```

این دستور فایل file.txt را با استفاده از یک ویرایشگر متن باز می‌کند.

مثال:

```
xdg-open -r file.txt
```

این دستور فایل file.txt را با استفاده از یک برنامه ترمینال باز می‌کند.

مثال:

```
xdg-open -t https://www.google.com
```

این دستور URL https://www.google.com را با استفاده از یک مرورگر وب باز می‌کند.

مثال:

```
xdg-open -f file.txt
```

این دستور فایل file.txt را بدون بررسی نوع آن باز می‌کند.

دستور whois

دستور whois در لینوکس برای یافتن اطلاعات مربوط به یک دامنه، مانند مالک دامنه، اطلاعات تماس مالک و سرورهای نامی که دامنه از آنها استفاده می‌کند.

ساختار دستور:

```
$ whois [ -h HOST ] [ -p PORT ] [ -aCFHlLMmrRSVx ] [ -g SOURCE:FIRST-LAST ] [ -i ATTR ] [ -S SOURCE ] [ -T TYPE ] object
whois -t TYPE
whois -v TYPE
$ whois -q keyword
```

گزینه های دستور:

- h HOST ، –host HOST به HOST متصل میشود..
- H – سلب مسئولیت های قانونی را که برخی از رجیستری ها دوست دارند به شما نشان دهند نمایش نمیدهد.
- p – ، port PORT به PORT متصل میشود.
- help –– نمایش راهنمای.
- version –– نمایش اطلاعات نسخه
- a – تمام پایگاه داده های mirror شده را جستجو میکند.
- b – بازگرداندن محدوده آدرس IP مختصر با تماس سوء استفاده.
- B – فیلتر کردن شی را غیرفعال میکند
- l – شیء یک سطح کمتر خاص را میگرداند.
- L – تمام سطوح اشیاء کمتر خاص را میگرداند
- m – تمام اشیاء خاص تر را در یک سطح میگرداند
- M – تمام سطوح اشیاء خاص تر را میگرداند
- r – غیرفعال کردن جستجوی بازگشتی برای اطلاعات تماس.
- R – ارجاعات زیر و نمایش نیرو را غیرفعال کنید

مثال های دستور:

۱.یک پرس و جو whois برای نام دامنه انجام می‌دهد:

```
$ whois {Domain_name}
```

۲. گزینه –H سلب مسئولیت طولانی مدت قانونی را که بسیاری از ثبت دامنه همراه با اطلاعات دامنه ارائه می کنند حذف می‌کند.

```
$ whois -H {Domain_name}
```

نمونه استفاده از گزینه :h-

```
$whois -h whois.example.com example.com
```

این دستور، اطلاعات whois برای دامنه example.com را از سرور whois.example.com دریافت می‌کند.

نمونه استفاده از گزینه : -p

```
$whois -p 43 example.com
```

این دستور، اطلاعات whois برای دامنه example.com را از سرور whois با استفاده از پورت ۴۳ دریافت می‌کند.

نمونه استفاده از گزینه : -a

```
$whois -a example.com
```

این دستور، اطلاعات whois برای دامنه example.com را با جزئیات بیشتر نمایش می‌دهد.

نمونه استفاده از گزینه : -C

```
$whois -C example.com
```

این دستور، نماینده معتبر whois برای دامنه example.com را نمایش می‌دهد.

نمونه استفاده از گزینه : -F

```
$whois -F example.com
```

این دستور، اطلاعات whois برای دامنه example.comرا به صورت قالب‌بندی شده نمایش می‌دهد.

نمونه استفاده از گزینه : -H

```
$whois -H example.com
```

این دستور، همه اطلاعات whois برای دامنه example.com را بدون هیچ عنوان اضافی نمایش می‌دهد.

نمونه استفاده از گزینه : -l

```
$whois -l example.com
```

این دستور، اطلاعات whois برای دامنه example.com را به صورت خطوط تکی نمایش می‌دهد.

نمونه استفاده از گزینه : -L

```
$whois -L example.com
```

این دستور، اطلاعات whois برای دامنه example.com را به صورت یک لیست نمایش می‌دهد.

نمونه استفاده از گزینه : -m

```
$whois -m example.com
```

این دستور، اطلاعات whois برای دامنه example.com را به صورت خام نمایش می‌دهد.

نمونه استفاده از گزینه : -r

```
$whois -r example.com
```

این دستور، اطلاعات whois برای دامنه example.com را به صورت بازگشتی نمایش می‌دهد.

نمونه استفاده از گزینه : -R

```
$whois -R example.com
```

این دستور، اطلاعات whois برای دامنه example.com را به صورت درختی نمایش می‌دهد.

نمونه استفاده از گزینه : -S

```
$whois -S example.com
```

این دستور، منبع whois برای دامنه example.com را نمایش می‌دهد.

نمونه استفاده از گزینه : -T

```
$whois -T domain example.com
```

این دستور، اطلاعات whois برای دامنه example.com را با نوع مشخص domain نمایش می‌دهد.

نمونه استفاده از گزینه : t-

```
$whois -t domain example.com
```

این دستور، اطلاعات whois برای دامنه example.com را با نوع مشخص domain نمایش می‌دهد و هر سطر را به صورت قالب‌بندی شده نمایش می‌دهد.

نمونه استفاده از گزینه : v-

```
$whois -v domain example.com
```

این دستور، اطلاعات whois برای دامنه example.com را با نوع مشخص domain نمایش می‌دهد و اطلاعات اضافی مربوط به whois را نمایش می‌دهد.

دستور ssh

دستور ssh در لینوکس مخفف "Secure Shell" است. این یک پروتکل است که برای اتصال ایمن به سرور/سیستم راه دور استفاده می شود. ssh امن تر است به این معنا که داده ها را به صورت رمزگذاری شده بین میزبان و مشتری منتقل می‌کند. ssh در پورت TCP/IP 22 اجرا می شود.

ساختار دستور:

```
$ ssh user_name@host(IP/Domain_Name)
$ ssh -i private.key user_name@host
```

اجزای دستور

- options گزینه های مختلف SSH
- user نام کاربری برای ورود به سیستم
- hostname آدرس IP یا نام هاست کامپیوتر مقصد
- command دستوری که می خواهید در کامپیوتر مقصد اجرا کنید (اختیاری)

گزینه های دستور:

- p پورت SSHطپیش فرض
- i مسیر فایل کلید خصوصی SSH
- v افزایش سطح گزارش دهی
- q کاهش سطح گزارش دهی
- C فعال کردن فشرده سازی داده ها
- X ارسال X11 forwarding
- A فعال کردن agent forwarding
- t اتصال به صورت tty
- l نام کاربری برای ورود به سیستم (پیش فرض نام کاربری فعلی)
- g گروه های کاربری برای ورود به سیستم
- تنظیمات سفارشی SSH

مثال های دستور:

۱.از یک شماره پورت متفاوت برای اتصال SSH استفاده کنید:

```
$ ssh test.server.com -p 3322
```

۲. -i ssh به سرور راه دور با استفاده از کلید خصوصی

```
$ ssh -i private.key user_name@host
```

۳. -l ssh نام کاربری دیگری را مشخص می‌کند

```
$ ssh -l alternative-username sample.ssh.com
```

مثال های بیشتر:

گزینه -4: استفاده از IPv4 برای اتصال.

```
$ssh -4 remote_host
```

گزینه -6: استفاده از IPv6 برای اتصال.

```
$ssh -6 remote_host
```

گزینه -A: فعال کردن فراهم کردن هویت.(Agent Forwarding)

```
$ssh -A remote_host
```

گزینه -a: غیرفعال کردن فراهم کردن هویت.(Agent Forwarding)

```
$ssh -a remote_host
```

گزینه -C: فشرده‌سازی ترافیک.

```
$ssh -C remote_host
```

گزینه -f: اجرای ssh به صورت پس‌زمینه.(background)

```
$ssh -f remote_host
```

گزینه -G: چاپ تنظیمات پیکربندی متصل شده بدون برقراری اتصال.

```
$ssh -G remote_host
```

گزینه -g: فعال کردن فروادینگ ترافیک.(Traffic forwarding)

```
$ssh -g remote_host
```

گزینه -K: فعال کردن مدیریت کلید (GSSAPI Key Exchange) با تمام پشتیبانی ها.

```
$ssh -K remote_host
```

گزینه -k: غیرفعال کردن مدیریت کلید.(GSSAPI Key Exchange)

```
$ssh -k remote_host
```

گزینه -M: فعال کردن حالت تقاطعی چند جلسه.(Multiplex Master Mode)

```
$ssh -M remote_host
```

گزینه -N: عدم اجرای دستور پس از برقراری اتصال.

```
$ssh -N remote_host
```

گزینه -n: عدم خواندن ورودی از STDIN.

```
$ssh -n remote_host
```

گزینه -q: حالت سکوت.(Quiet mode)

```
$ssh -q remote_host
```

گزینه -s: فعال کردن ترمینال محلی برای اتصال.

```
$ssh -s remote_host
```

گزینه -T: غیرفعال کردن ترمینال پس از برقراری اتصال.

```
$ssh -T remote_host
```

گزینه -V: چاپ نسخه ssh.

```
$ssh -V
```

گزینه v:-حالت) verbose تفصیلی.

```
$ssh -v remote_host
```

دستور sftp

دستور sftp یک برنامه مشتق از SSH است که اجازه انتقال امن فایل بین دو سیستم را از طریق پروتکل SSH فراهم می‌کند. این دستور اجازه می‌دهد تا فایل‌ها و فهرست‌ها بین دو سیستم به صورت امن انتقال یابند. همچنین SFTPمخفف عبارت Secure File Transfer Protocol است. این پروتکل برای انتقال امن فایل بین دو سیستم از طریق SSH استفاده می شود SFTP .جایگزینی امن برای FTP (File Transfer Protocol)است که به صورت رمزنگاری نشده می باشد.

ساختار دستور:

برای استفاده از SFTP می توانید از دستور زیر استفاده کنید:

```
sftp [username]@[host]
```

- username نام کاربری شما در سیستم مقصد
- host آدرس IP یا نام هاست سیستم مقصد

مثال

```
sftp user@example.com
```

این دستور شما را به سیستم example.com با نام کاربری user متصل می کند.

گزینه های SFTP

- h نمایش راهنمای دستور
- V نمایش نسخه SFTP
- P پورت SSH
- i مسیر فایل کلید SSH
- l سطح گزارش دهی
- v فعال کردن حالت verbose
- c فعال کردن فشرده سازی
- b فعال کردن حالت batch
- r دانلود بازگشتی (recursive)
- u آپلود بازگشتی (recursive)

مثال های دستور:

اتصال به سرور از راه دور: در این مثال، به یک سرور از راه دور با استفاده از sftp متصل می‌شویم.

```
sftp username@remote_server
```

دریافت یک فایل از سرور از راه دور: در این مثال، یک فایل به نام file.txt از سرور از راه دور به سیستم محلی دریافت می‌شود.

```
get file.txt /local/path/to/save
```

ارسال یک فایل به سرور از راه دور: در این مثال، یک فایل به نام file.txt از سیستم محلی به سرور از راه دور ارسال می‌شود.

```
put /local/path/to/file.txt remote/path/to/save
```

لیست کردن فایل‌ها و پوشه‌ها در محل فعلی: با استفاده از دستور ls می‌توانید فایل‌ها و پوشه‌های موجود در محل فعلی را مشاهده کنید.

```
ls
```

انتقال فایل با استفاده از پورت خاص: در این مثال، از پورت ۲۲۲۲ برای ارسال یک فایل به سرور از راه دور استفاده می‌شود.

```
sftp -P 2222 username@remote_server
```

دستور sftp امکانات مختلفی برای مدیریت فایل‌ها و فهرست‌ها از راه دور ارائه می‌دهد و به شما امکان ارسال و دریافت فایل‌ها با امنیت بالا را می‌دهد.

دستور rcp

دستور rcp یک دستور موجود در سیستم‌های Unix است که برای انتقال فایل بین دو میزبان از طریق شبکه استفاده می‌شود. این دستور مشابه دستور cp بر روی سیستم فعلی عمل می‌کند، اما برای انتقال فایل بین دو سیستم متفاوت استفاده می‌شود. همچنین RCPمخفف عبارت Remote Copy Protocolاست. این دستور برای کپی کردن فایل بین دو سیستم از طریق SSH استفاده می شود RCP .جایگزینی امن برای FTP (File Transfer Protocol)است که به صورت رمزنگاری نشده می باشد.

استفاده از RCP:

برای استفاده از RCP می توانید از دستور زیر استفاده کنید:

```
rcp [options] [source] [destination]
```

گزینه های RCP

- sourceمسیر فایل یا دایرکتوری در سیستم مبدا
- destinationمسیر ذخیره سازی فایل یا دایرکتوری در سیستم مقصد

گزینه های دستور:

- h نمایش راهنمای دستور
- V نمایش نسخه RCP
- p حفظ timestamp و permission فایل
- r کپی کردن دایرکتوری به صورت بازگشتی (recursive)
- v فعال کردن حالت verbose
- u رمز عبور
- P پورت SSH
- i مسیر فایل کلید SSH

مثال های دستور:

گزینه p – این گزینه باعث می‌شود اطلاعات مرتبط با فایل، مانند سلسله مراتب و دسترسی‌ها، همراه با فایل انتقال یابند.

```
rcp -p file.txt user@remote_host:/path/to/destination
```

گزینه x – این گزینه از انتقال امن فایل با استفاده از رمزنگاری در طول انتقال جلوگیری می‌کند.

```
rcp -x file.txt user@remote_host:/path/to/destination
```

گزینه r – این گزینه به شما امکان می‌دهد یک فهرست به صورت بازگشتی انتقال دهید.

```
rcp -r directory user@remote_host:/path/to/destination
```

دستور rlogin

دستور rlogin یک دستور است که برای ورود به یک سیستم از راه دور از طریق شبکه با استفاده از پروتکل rlogin استفاده می‌شود. این دستور به کاربر امکان می‌دهد تا به سیستم دیگری وصل شود و به عنوان کاربری مشخص شده وارد شود. همچنین Rlogin مخفف عبارت Remote Login است. این دستور برای ورود به سیستم از راه دور از طریق شبکه استفاده می شود. Rlogin از پروتکل TCP برای برقراری ارتباط بین دو سیستم استفاده می کند.

برای استفاده از Rlogin می توانید از دستور زیر استفاده کنید:

```
rlogin [username]@[host]
```

- Username :نام کاربری شما در سیستم مقصد
- Host :آدرس IP یا نام هاست سیستم مقصد

گزینه های دستور:

- h نمایش راهنمای دستور
- V نمایش نسخه Rlogin
- l نام کاربری برای ورود به سیستم مقصد
- p پورت TCP
- e ارسال دستورات به صورت خودکار
- K فعال کردن Kerberos
- x فعال کردن X Window System

مثال های دستور:

گزینه – l :username این گزینه به شما امکان می‌دهد نام کاربری مورد نظر را برای ورود به سیستم از راه دور مشخص کنید.

```
rlogin -l remote_user remote_host
```

گزینه e :char این گزینه به شما امکان می‌دهد یک کاراکتر خاص را به عنوان نشانگر پایان خط فرمان مشخص کنید.

```
rlogin -e '#' remote_host
```

گزینه ۸: این گزینه باعث می‌شود اطلاعات ارسال شده و دریافتی به صورت ۸ بیتی (بجای ۷ بیتی) در نظر گرفته شود.

```
rlogin -8 remote_host
```

گزینه K: این گزینه به شما امکان می‌دهد تا از ارسال کلیدهای تشخیصی برای احراز هویت استفاده کنید.

```
rlogin -K remote_host
```

در این مثال‌ها، rlogin برای ورود به یک سیستم از راه دور از طریق شبکه استفاده می‌شود. انواع گزینه‌ها مختلف این دستور به کاربر امکانات مختلفی را برای ورود و ارتباط با سیستم مورد نظر ارائه می‌دهند.

مثال:

```
rlogin -l user -p 23 example.com
```

این دستور شما را به سیستم example.com با نام کاربری user و پورت TCP 23 متصل می کند.

مثال:

```
rlogin -e "ls -l" example.com
```

این دستور دستور ls -l را در سیستم example.com اجرا می کند.

دستور agetty

agettyیک نسخه لینوکس getty است. "getty" یک برنامه یونیکس است که روی یک کامپیوتر میزبان اجرا می شود و با مدیریت ترمینال های فیزیکی یا مجازی امکان دسترسی چند کاربر را فراهم می کند .لینوکس ترمینال مجازی (tty) را ارائه می دهد که شبیه ترمینال معمولی لینوکس است .دستور agetty یک ترمینال مجازی (پورت tty را باز می کند ، نام ورود را وارد می کند و دستور **bin / login /** را فراخوانی می کند.همچنین Agettyمخفف عبارت Automatic getty است. این برنامه برای مدیریت خطوط سریال و ارائه خدمات احراز هویت و ورود به سیستم برای کاربران از طریق ترمینال استفاده می شود . Agettyمعمولاً در هنگام بوت شدن سیستم اجرا می شود و منتظر می ماند تا کاربری از طریق خط سریال به سیستم متصل شود.

ساختار دستور:

```
agetty [options] port [baud_rate...] [term]
```

گزینه های دستور:

- **port** : به پوشه **dev/** اشاره دارد. اگر یک "-" مشخص شده باشد ، این دستور در نظر دارد که ورودی استاندارد آن از قبل به یک پورت **tty** متصل شده است و قبلاً اتصال به یک کاربر از راه دور برقرار شده است.
- **baud_rate**:این یک لیست جدا شده با کاما از یک یا چند نرخ baud است .باید به ترتیب نزولی مشخص شود.
- **term**:مقداری است که برای متغیر محیط TERM استفاده می شود.
- **8-** ، **8–بیت**:فرض کنید ۸ بیتی tty.
- **a-**، **–autologin**:ورود خودکار برای کاربر مشخص شده است.
- **c-**، **–noreset**:حالت کنترل را **مجدداً** تنظیم نکند.
- **E-**، **–remote**:**به** طور معمول به دستور login (1)نام میزبان از راه دور داده می شود که توسط چیزی مانند telnetd (8)فراخوانی شود .این گزینه به agetty اجازه می دهد آنچه را که برای نام میزبان استفاده می کند ، برای ورود (۱) برای استفاده در utmp (5) استفاده کند.
- **h-**، **–flow-control**:دست دادن CTS / RTS را کنترل می کند (کنترل جریان).
- **i-**، **–noissue**:پرونده را نمایش ندهید.
- **J-** **–noclear**:قبل از اعلان صفحه را پاک نکنید.
- **m-**، **–extract-baud**:در حین اتصال از میزان baud استخراج استفاده کنید.
- **n-**، **–skip-login**:برای ورود سریع وارد نشوید.
- **p-**، **–login-pause**:منتظر بمانید تا کاربر هر کلید را قبل از درخواست ورود فشار دهد.
- **R-**، **–hangup**:برای انجام قطع ارتباط مجازی از ترمینال مشخص شده ، با **vhangup ()** تماس بگیرید.
- **s-**، **–keep-baud**:سعی کنید میزان baud قبلاً استفاده شده را حفظ کنید.
- **t-**، **–timeout**:اگر هیچ نام کاربری در مدت زمان ثانیه خوانده نشود ، جلسه ورود به سیستم را خاتمه میدهد.
- **U-**، **–detect-case**:این برای روشن کردن پشتیبانی برای شناسایی ترمینال فقط بزرگ استفاده می شود.

برای جزئیات بیشتر در مورد گزینه ها می توانید دستور زیر را در ترمینال اجرا کنید:

```
[root@os ~]# agetty --help

Usage:
 agetty [options] <line> [<baud_rate>,...] [<termtype>]
 agetty [options] <baud_rate>,... <line> [<termtype>]

Open a terminal and set its mode.

Options:
 -8, --8bits                assume 8-bit tty
 -a, --autologin <user>     login the specified user automatically
 -c, --noreset              do not reset control mode
 -E, --remote               use -r <hostname> for login(1)
 -f, --issue-file <file>    display issue file
 -h, --flow-control         enable hardware flow control
 -H, --host <hostname>      specify login host
 -i, --noissue              do not display issue file
 -I, --init-string <string> set init string
 -J  --noclear              do not clear the screen before prompt
 -l, --login-program <file> specify login program
 -L, --local-line[=<mode>]  control the local line flag
 -m, --extract-baud         extract baud rate during connect
 -n, --skip-login           do not prompt for login
 -N  --nonewline            do not print a newline before issue
 -o, --login-options <opts> options that are passed to login
 -p, --login-pause          wait for any key before the login
 -r, --chroot <dir>         change root to the directory
 -R, --hangup               do virtually hangup on the tty
 -s, --keep-baud            try to keep baud rate after break
 -t, --timeout <number>     login process timeout
 -U, --detect-case          detect uppercase terminal
 -w, --wait-cr              wait carriage-return
     --nohints              do not print hints
     --nohostname           no hostname at all will be shown
     --long-hostname        show full qualified hostname
     --erase-chars <string> additional backspace chars
     --kill-chars <string>  additional kill chars
     --chdir <directory>    chdir before the login
     --delay <number>       sleep seconds before prompt
     --nice <number>        run login with this priority
     --reload               reload prompts on running agetty instances
     --list-speeds          display supported baud rates
     --help                 display this help
     --version              display version
```

مثال های دستور:

1)agetty -8 - linux

- 8-گزینه برای 8 tty بیتی.
- '-'برای مشخص می کند که ورودی استاندارد از قبل به یک پورت tty متصل شده است.
- نرخ baud اختیاری است بنابراین در اینجا استفاده نمی شود.
- 'linux'مقدار متغیر محیط TERM است.

```
[root@os ~]# agetty -8 - linux

CentOS Linux 7 (Core)
Kernel 3.10.0-957.12.1.el7.x86_64 on an x86_64

os login:
```

2) agetty -8 -t 5 - linux

- t 5-مهلت زمانی روند ورود به سیستم است.

```
[root@os ~]# agetty -8 -t 5 - linux

CentOS Linux 7 (Core)
Kernel 3.10.0-957.12.1.el7.x86_64 on an x86_64

os login: Alarm clock
[root@os ~]#
```

3) agetty -h -t 60 tty 9600 vtxxx

- ttyبه دستگاه **dev/tty / اشاره دارد.**
- 9600نرخ بیت بیت در ثانیه است.
- **vtxxx**متغیر محیط TERM برای نشان دادن اتصال ترمینال **VTxxx** است ، در مثال قبلی از **"linux"** به عنوان TERM envاستفاده شده است.
- **h-**را فعال **CTS/RTS** دست دادن (کنترل جریان)
- **t 60-**اجازه می دهد ۶۰ ثانیه قبل از قطع شدن مودم ، کسی وارد سیستم شود.
- a autologin-را مشخص می کند.

4) agetty -a -h -t 60 tty 9600 vt102

5) agetty –version

فصل هجدهم

فصل هجدهم

مدیریت پرینتر

CUPS یا سیستم مشترک چاپ یونیکس، روشی است که به طور گسترده‌ای برای مدیریت چاپ و خدمات چاپ در اوبونتو استفاده می‌شود. این سیستم چاپ آزاد به عنوان استاندارد چاپ در اکثر توزیع‌های لینوکس شناخته شده است و از پروتکل چاپ اینترنتی ((IPP استفاده می‌کند. CUPS مدیریت کارهای چاپ و صف‌ها را انجام می‌دهد و پشتیبانی از مجموعه‌ای گسترده از پرینترها را از جمله پرینترهای نقطه‌ای تا لیزر و بسیاری دیگر فراهم می‌کند. همچنین، CUPS از توصیف‌نامه پرینتر (PPD) PostScript و تشخیص خودکار پرینترهای شبکه پشتیبانی می‌کند و دارای یک ابزار پیکربندی و مدیریت ساده مبتنی بر وب است.

CUPS با انواع مختلف پرینترها سازگاری دارد. این سیستم مدیریت چاپ، پشتیبانی از پرینترهای متنوعی از جمله پرینترهای نقطه‌ای، لیزر، جوهرافشان و دیگر انواع را فراهم می‌کند. برای استفاده از CUPS با پرینتر خود، مراحل زیر را دنبال کنید:

نصب CUPS: ابتدا مطمئن شوید که CUPS در سیستم شما نصب شده است. در اوبونتو و توزیع‌های مشابه، می‌توانید از دستور زیر برای نصب آن استفاده کنید:

```
sudo apt-get install cups
```

راه‌اندازی پرینتر: پرینتر خود را به سیستم متصل کنید و روشن کنید. سپس به آدرس زیر در مرورگر وب خود بروید:

```
http://localhost:631
```

اضافه کردن پرینتر: در صفحه مدیریت CUPS ، به قسمت "Administration" بروید و گزینه "Add Printer" را انتخاب کنید. دستورالعمل‌های مربوط به اضافه کردن پرینتر را دنبال کنید. معمولاً CUPS پرینترها را به صورت خودکار تشخیص می‌دهد و اطلاعات مربوط به آن‌ها را نمایش می‌دهد.

تنظیمات پرینتر: پس از اضافه کردن پرینتر، می‌توانید تنظیمات مربوط به آن را در CUPS تغییر دهید. این شامل تنظیمات مربوط به کیفیت چاپ، نوع کاغذ، و دیگر تنظیمات است.

آزمایش چاپ: برای اطمینان از سازگاری پرینتر با CUPS ، یک صفحه آزمایشی چاپ کنید.

با این مراحل، شما می‌توانید پرینتر خود را با CUPS سازگار کنید و از این سیستم مدیریت چاپ برای مدیریت کارهای چاپ خود استفاده کنید.

دستور lpinfo

دستور lpinfo در سیستم‌های Unix/Linux برای نمایش اطلاعات مربوط به سیستم چاپ‌گرها و چاپ‌سرورها استفاده می‌شود. این دستور به کاربران این امکان را می‌دهد تا اطلاعاتی مانند لیست چاپ‌گرهای موجود، ویژگی‌های آن‌ها، و تنظیمات مربوط به چاپ را مشاهده کنند.

گزینه های دستور:

استفاده از lpinfo

برای استفاده از lpinfo می توانید از دستور زیر استفاده کنید

```
lpinfo [options]
```

- h نمایش راهنمای دستور
- v نمایش اطلاعات
- l نمایش لیست چاپگرها
- p نمایش اطلاعات مربوط به یک چاپگر خاص
- c نمایش اطلاعات مربوط به یک کار چاپ خاص
- u نمایش اطلاعات مربوط به کارهای چاپ یک کاربر خاص

مثال های دستور:

نمایش لیست تمام چاپگرهای موجود: این دستور برای نمایش لیست تمام چاپگرهای موجود در سیستم استفاده می‌شود.

```
lpinfo -v
```

نمایش جزئیات یک چاپگر خاص: این دستور برای نمایش جزئیات و ویژگی‌های یک چاپگر خاص بر اساس شناسه آن استفاده می‌شود.

```
lpinfo -v -l -m
```

نمایش تنظیمات مربوط به چاپ: این دستور برای نمایش تنظیمات مربوط به چاپ و چاپگرها استفاده می‌شود.

```
lpinfo -u
```

دستور lpstat

دستور lpstat در سیستم‌های Unix/Linux برای نمایش اطلاعات مربوط به وضعیت سیستم چاپگرها و چاپسرورها استفاده می‌شود. این دستور به کاربران این امکان را می‌دهد تا وضعیت چاپگرها، صف چاپ، و سایر اطلاعات مربوط به چاپ را مشاهده کنند.

استفاده از lpstat

برای استفاده از lpstat می توانید از دستور زیر استفاده کنید

```
lpstat [options]
```

گزینه های lpstat

- h نمایش راهنمای دستور
- V نمایش نسخه lpstat
- a نمایش اطلاعات مربوط به تمام چاپگرها
- p نمایش اطلاعات مربوط به یک چاپگر خاص
- u نمایش اطلاعات مربوط به job های چاپ کاربر
- t نمایش اطلاعات مربوط به job های چاپ با شناسه خاص
- W نمایش اطلاعات مربوط به job های چاپ در حال انتظار
- c نمایش اطلاعات مربوط به job های چاپ لغو شده

مثال های دستور:

نمایش وضعیت تمام چاپگرها: این دستور برای نمایش وضعیت تمام چاپگرهای موجود در سیستم استفاده می‌شود.

```
lpstat -p
```

نمایش صف چاپ: این دستور برای نمایش صف چاپ، یعنی فایل‌هایی که در حال انتظار برای چاپ هستند، استفاده می‌شود.

```
lpstat -o
```

نمایش جزئیات یک چاپگر خاص: این دستور برای نمایش جزئیات و وضعیت یک چاپگر خاص بر اساس شناسه آن استفاده می‌شود.

```
lpstat -l -p printer1
```

دستور lpstat به کاربران این امکان را می‌دهد تا اطلاعات و وضعیت مربوط به سیستم چاپگرها و چاپسرورها را به صورت دقیق و جامع مشاهده کنند، از جمله وضعیت چاپگرها، صف چاپ، و سایر اطلاعات مربوط به چاپ.

مثال:

```
lpstat -u user1
```

این دستور اطلاعات مربوط به job های چاپ کاربر user1 را نمایش می دهد.

مثال:

```
lpstat -t 1234
```

این دستور اطلاعات مربوط به job چاپ با شناسه ۱۲۳۴ را نمایش می دهد.

مثال:

```
lpstat -W
```

این دستور اطلاعات مربوط به job های چاپ در حال انتظار را نمایش می دهد.

مثال:

```
lpstat -c
```

این دستور اطلاعات مربوط به job های چاپ لغو شده را نمایش می دهد.

دستور lpc

دستور lpc در سیستم‌های Unix/Linux برای مدیریت و کنترل چاپ‌گرها و صف چاپ استفاده می‌شود. با استفاده از این دستور، کاربران می‌توانند چاپ‌گرها را فعال یا غیرفعال کنند، صف چاپ را مدیریت کنند و تنظیمات مربوط به چاپ را تغییر دهند.

استفاده از دستور:

برای استفاده از LPC می توانید از دستور زیر استفاده کنید

```
lpc [options]
```

گزینه های دستور:

- h نمایش راهنمای دستور
- V نمایش نسخه LPC
- l لیست چاپگرهای موجود
- s تنظیم چاپگر پیش فرض
- c لغو چاپ یک job
- m ارسال یک پیام به چاپگر
- i اطلاعات مربوط به یک job چاپ را نمایش می دهد
- q وضعیت صف چاپ را نمایش می دهد
- p تنظیم اولویت چاپ یک job
- u تنظیم نام کاربری برای چاپ
- g تنظیم گروه برای چاپ

مثال های دستور:

مثال:

```
lpc -l
```

این دستور لیست چاپگرهای موجود را نمایش می دهد.

مثال:

```
lpc -s laserjet
```

این دستور چاپگر laserjet را به عنوان چاپگر پیش فرض تنظیم می کند.

مثال:

```
lpc -c 1234
```

این دستور job چاپ با شناسه ۱۲۳۴ را لغو می کند.

مثال:

```
lpc -m “This is a test message.”
```

این دستور پیام “This is a test message.” را به چاپگر ارسال می کند.

مثال:

```
lpc -i 1234
```

این دستور اطلاعات مربوط به job چاپ با شناسه ۱۲۳۴ را نمایش می دهد.

مثال:

```
lpc -q
```

این دستور وضعیت صف چاپ را نمایش می دهد.

مثال:

```
lpc -p 5 1234
```

این دستور اولویت job چاپ با شناسه ۱۲۳۴ را به ۵ تنظیم می کند.

مثال:

```
lpc -u user
```

این دستور نام کاربری user را برای چاپ تنظیم می کند.

مثال:

```
lpc -g group
```

این دستور گروه group را برای چاپ تنظیم می کند.

دستور lpadmin

دستور lpadmin در سیستم‌های Unix/Linux برای اضافه کردن، حذف کردن، یا تغییر دادن تنظیمات یک چاپگر در CUPS مورد استفاده قرار می‌گیرد. با استفاده از این دستور، می‌توانید چاپگرها را مدیریت کرده و تنظیمات آن‌ها را تغییر دهید.

استفاده از دستور:

برای استفاده از Lpadmin می توانید از دستور زیر استفاده کنید

```
lpadmin [options]
```

گزینه های دستور:

- h نمایش راهنمای دستور
- V نمایش نسخه Lpadmin
- p نام چاپگر
- d توضیحات چاپگر
- L لیست چاپگرهای موجود
- m مدل چاپگر
- u آدرس IP یا نام هاست چاپگر
- E فعال کردن چاپ از طریق LPD
- P پورت LPD
- تنظیمات چاپگر

مثال های دستور:

مثال:

```
lpadmin -p my_printer -d “My Printer”
```

این دستور یک چاپگر با نام my_printer و توضیحات “My Printer” اضافه می کند.

مثال:

```
lpadmin -L
```

این دستور لیست چاپگرهای موجود را نمایش می دهد.

مثال:

```
lpadmin -p my_printer -m “HP LaserJet 1020”
```

این دستور مدل چاپگر my_printer را به HP LaserJet 1020 تنظیم می کند.

مثال:

```
lpadmin -p my_printer -u 192.168.1.100
```

این دستور آدرس IP چاپگر my_printer را به ۱۹۲,۱۶۸,۱,۱۰۰ تنظیم می کند.

مثال:

```
lpadmin -p my_printer -E
```

این دستور چاپ از طریق LPD را برای چاپگر my_printer فعال می کند.

مثال:

```
lpadmin -p my_printer -P 515
```

این دستور پورت LPD را برای چاپگر my_printer به ۵۱۵ تنظیم می کند.

مثال:

```
lpadmin -p my_printer -o “collate=yes”
```

این دستور چاپگر my_printer را طوری تنظیم می کند که قبل از چاپ، صفحات را مرتب کند.

دستور lp

دستور lp در سیستم‌های Unix/Linux برای ارسال فایل‌های قابل چاپ به یک چاپگر CUPS جهت چاپ استفاده می‌شود. این دستور امکان ارسال فایل‌ها برای چاپ به یک چاپگر خاص را فراهم می‌کند و می‌تواند با استفاده از گزینه‌ها و پارامترها تنظیمات مختلفی برای چاپ فایل‌ها اعمال کند.

استفاده از دستور:

برای ارسال فایل به چاپگر با استفاده از LP می‌توانید از دستور زیر استفاده کنید:

```
lp [options] file
```

- options: گزینه‌های LP
- file: نام فایل برای چاپ

گزینه های دستور:

- h نمایش راهنمای دستور
- V نمایش نسخه LP
- d نام چاپگر

- n تعداد کپی
- گزینه‌های چاپگر
- t عنوان چاپ
- m ارسال ایمیل پس از اتمام چاپ
- c کلاس چاپ
- p اولویت چاپ
- u نام کاربری
- w زمان انتظار

مثال‌های دستور:

چاپ یک فایل با تعداد کپی‌های مشخص: این دستور برای چاپ فایل example.txt با ۳ کپی استفاده می‌شود.

```
lp -n 3 example.txt
```

تنظیم عنوان برای چاپ: این دستور برای تنظیم یک عنوان ((titleبرای چاپ استفاده می‌شود.

```
lp -t “My Document” file.pdf
```

انتخاب چاپگر خاص برای چاپ: این دستور برای انتخاب یک چاپگر خاص به نام printer1 برای چاپ فایل استفاده می‌شود.

```
lp -d printer1 file.txt
```

تنظیم یک گزینه خاص برای چاپ: این دستور برای تنظیم یک گزینه خاص (مانند رنگ، سایز، و ...) برای چاپ استفاده می‌شود.

```
lp -o landscape file.docx
```

دستور lp به کاربران این امکان را می‌دهد تا فایل‌های خود را به یک چاپگر مشخص ارسال کرده و تنظیمات مختلفی برای چاپ اعمال کنند. این دستور بسیار مفید است برای ارسال فایل‌های قابل چاپ به یک چاپگر از خط فرمان. دستور LP در برخی از توزیع‌های Linux ممکن است منسوخ شده باشد. در این صورت می‌توانید از دستور lpr به جای آن استفاده کنید.

دستور LPR

دستور LPR مشابه LP است و برای ارسال فایل به چاپگر استفاده می‌شود.

استفاده از دستور:

برای ارسال فایل به چاپگر با استفاده از LPR می‌توانید از دستور زیر استفاده کنید:

```
lpr [options] file
```

- options: گزینه‌های LPR
- file: نام فایل برای چاپ

گزینه‌های دستور:

- hنمایش راهنمای دستور
- V نمایش نسخه LPR
- P نام چاپگر
- d تعداد کپی
- n چاپ شماره صفحه
- گزینه‌های چاپگر
- s چاپ در حالت silent
- m ارسال ایمیل پس از اتمام چاپ

مثال:

```
lpr -P printer1 file.txt
```

این دستور فایل file.txt را به چاپگر printer1 ارسال می‌کند.

مثال:

```
lpr -d 2 file.txt
```

این دستور دو نسخه از file.txt را چاپ می‌کند.

مثال:

```
lpr -o landscape file.txt
```

این دستور file.txt را در حالت افقی چاپ می‌کند.

دستور lpstat

دستور lpstat در سیستم‌عامل‌های مبتنی بر Unix و Linux برای نمایش اطلاعات مربوط به سیستم چاپگرها و وظایف چاپ در صف استفاده می‌شود. این دستور اطلاعاتی مانند وضعیت چاپگرها، وظایف چاپ در صف، چاپگرهای موجود و ... را نمایش می‌دهد.

استفاده از دستور:

برای مشاهده وضعیت چاپگر می توانید از دستور زیر استفاده کنید

```
lpstat
```

این دستور وضعیت چاپگر پیش فرض را نمایش می دهد.

گزینه های دستور:

- h نمایش راهنمای دستور
- V نمایش نسخه lpstat
- a نمایش اطلاعات تمام چاپگرها
- d نام چاپگر
- p شناسه job چاپ
- u نام کاربری

مثال های دستور:

نمایش وضعیت چاپگرها: این دستور برای نمایش وضعیت تمام چاپگرهای موجود در سیستم استفاده می‌شود.

```
lpstat -p
```

نمایش وظایف چاپ در صف: این دستور برای نمایش وظایف چاپی که در صف انتظار برای چاپ هستند استفاده می‌شود.

```
lpstat -o
```

نمایش وضعیت چاپگرها به صورت جزئی: این دستور برای نمایش جزئی‌تر وضعیت چاپگرها، اعمال تنظیمات خاص و ... استفاده می‌شود.

```
lpstat -l
```

نمایش اطلاعات مربوط به کاربران: این دستور برای نمایش اطلاعات مربوط به کاربران متصل به چاپگرها مانند وظایف چاپی و ... استفاده می‌شود.

```
lpstat -u
```

دستور lpstat به کاربر این امکان را می‌دهد تا اطلاعات مربوط به چاپگرها، وظایف چاپ در صف، کاربران متصل و ... را از خط فرمان به راحتی مشاهده کند. این دستور مفید است برای مدیریت و نظارت بر وظایف چاپ در سیستم.

دستور lprm

دستور lprm در سیستم‌عامل‌های مبتنی بر Unix و Linux برای حذف وظایف چاپ از صف چاپ استفاده می‌شود. با استفاده از این دستور، می‌توانید وظایف چاپی که در حالت انتظار برای چاپ هستند را از صف چاپ حذف کنید.

استفاده از دستور:

برای حذف فایل چاپ از صف چاپ می توانید از دستور زیر استفاده کنید

```
lprm [options] jobid
```

- options گزینه های LPRM
- jobid شناسه job چاپ

گزینه های دستور:

- h نمایش راهنمای دستور
- V نمایش نسخه LPRM
- a حذف تمام فایل های چاپ برای کاربر فعلی
- d نام چاپگر
- u نام کاربری

مثال های دستور:

حذف یک وظیفه چاپ با شماره شناسه وظیفه: این دستور برای حذف یک وظیفه چاپ با شماره شناسه مشخص از صف چاپ استفاده می‌شود.

```
lprm 123
```

حذف تمام وظایف چاپ مربوط به کاربر خاص: این دستور برای حذف تمام وظایف چاپ مربوط به یک کاربر خاص از صف چاپ استفاده می‌شود.

```
lprm -U username
```

حذف تمام وظایف چاپ مربوط به یک چاپگر خاص: این دستور برای حذف تمام وظایف چاپ مربوط به یک چاپگر خاص از صف چاپ استفاده می‌شود.

```
lprm -P printer_name
```

حذف تمام وظایف چاپ موجود در صف چاپ: این دستور برای حذف تمام وظایف چاپ موجود در صف چاپ استفاده می‌شود.

```
lprm -
```

دستور lpq

دستور lpq در سیستم‌های Unix/Linux برای نمایش صف چاپ و وضعیت فعلی فایل‌های در حال انتظار برای چاپ استفاده می‌شود. این دستور امکانات مختلفی برای نمایش اطلاعات صف چاپ و وضعیت فایل‌ها فراهم می‌کند.

ساختار دستور

ساختار کلی دستور lpq به شرح زیر است

```
lpq [options] [user] [printer]
```

گزینه‌های دستور lpq

a نمایش تمام jobها، چه فعال و چه غیرفعال.

c نمایش تعداد jobها در صف.

l نمایش اطلاعات برای هر job.

P نمایش نام چاپگر پیش‌فرض.

q نمایش صف چاپ مشخص‌شده.

u نمایش jobهای کاربر مشخص‌شده.

مثال‌های دستور lpq

نمایش صف چاپ بدون انتظار: این دستور برای نمایش صف چاپ بدون انتظار استفاده می‌شود.

```
lpq
```

نمایش صف چاپ برای یک کاربر خاص: این دستور برای نمایش صف چاپ مربوط به یک کاربر خاص استفاده می‌شود.

```
lpq -U username
```

نمایش صف چاپ برای یک چاپگر خاص: این دستور برای نمایش صف چاپ مربوط به یک چاپگر خاص استفاده می‌شود.

```
lpq -P printer1
```

نمایش جزئیات صف چاپ: این دستور برای نمایش جزئیات صف چاپ، شامل تعداد صفحات هر فایل و وضعیت چاپ، استفاده می‌شود.

```
lpq -l
```

نمایش صف چاپ با فاصله زمانی: این دستور برای نمایش صف چاپ با فاصله زمانی مشخص بین نمایش‌ها استفاده می‌شود.

```
lpq +5
```

نمایش تعداد jobها در صف چاپ:

```
lpq -c
```

نمایش اطلاعات تر برای هر job:

```
lpq -l
```

نمایش نام چاپگر پیش‌فرض:

```
lpq -P
```

نمایش jobهای کاربر ali:

```
lpq -u ali
```

نمایش jobها در صف چاپ lp2:

```
lpq -q lp2
```

دستور cupsctl

دستور cupsctl در سیستم‌های Unix/Linux برای تنظیم و مدیریت تنظیمات سیستم چاپگر CUPS استفاده می‌شود. این دستور به کاربران این امکان را می‌دهد تا تنظیمات مختلفی مانند لاگ‌گیری، دسترسی از راه دور، اشتراک چاپگرها و... را تغییر دهند. همچنین

اشتراک گذاری چاپگر

احراز هویت

رمزنگاری

ثبت وقایع

ساختار دستور:

ساختار کلی دستور cupsctl به صورت زیر است

```
cupsctl [options] [name=value]
```

گزینه های دستور:

- E رمزنگاری را در اتصال به زمان‌بندی فعال می‌کند.
- U username نام کاربری دیگری را برای احراز هویت با زمان‌بندی مشخص می‌کند.
- h server[port] آدرس سرور را مشخص می‌کند.
- [no]debuglogging ثبت وقایع اشکال‌زدایی را فعال (غیرفعال) می‌کند.
- [no]remoteadmin مدیریت از راه دور را فعال (غیرفعال) می‌کند.
- [no]remoteany چاپ از هر آدرسی، مانند اینترنت را فعال (غیرفعال) می‌کند.
- [no]shareprinters اشتراک گذاری چاپگرهای محلی با سایر رایانه‌ها را فعال (غیرفعال) می‌کند.
- [no]usercancelany به کاربران اجازه می‌دهد (جلوگیری می‌کند) تا کارهای متعلق به دیگران را لغو کنند.

نام=مقدار

شما می‌توانید از name=value برای تنظیم مقادیر خاص در فایل cupsd.conf استفاده کنید.

مثال های دستور:

مشاهده تنظیمات فعلی:

```
cupsctl
```

فعال کردن ثبت وقایع اشکال‌زدایی:

```
cupsctl --debug-logging
```

دریافت وضعیت فعلی ثبت وقایع اشکال‌زدایی:

```
cupsctl | grep '^_debug_logging' | awk -F= '{print $2}'
```

غیرفعال کردن اشتراک گذاری چاپگر:

```
cupsctl --no-share-printers
```

تنظیم زمان بیکاری پیش فرض برای چاپگرها:

```
cupsctl DefaultIdleTimeout=300
```

فعال کردن احراز هویت Kerberos:

```
cupsctl AuthType=Kerberos
```

محدود کردن دسترسی به چاپگرها به کاربران محلی:

```
cupsctl RemoteAdmin=deny
```

فعال‌سازی لاگ‌گیری اشکال‌زدایی: این دستور برای فعال‌سازی لاگ‌گیری اشکال‌زدایی در CUPS استفاده می‌شود.

```
cupsctl --debug-logging
```

غیرفعال‌سازی دسترسی از راه دور برای مدیریت: این دستور برای غیرفعال‌سازی دسترسی از راه دور برای مدیریت CUPS استفاده می‌شود.

```
cupsctl --no-remote-admin
```

فعال‌سازی اشتراک چاپگرها: این دستور برای فعال‌سازی اشتراک چاپگرها در CUPS استفاده می‌شود.

```
cupsctl --share-printers
```

تنظیم یک مقدار خاص: این دستور برای تنظیم یک مقدار خاص در تنظیمات CUPS استفاده می‌شود.

```
cupsctl PageSize=A4
```

دستور cupsenable

دستور cupsenable در سیستم‌عامل‌های مبتنی بر Unix و Linux برای فعال‌سازی یک چاپگر در سیستم CUPS (سیستم مدیریت چاپ) استفاده می‌شود. با استفاده از این دستور، می‌توانید یک چاپگر که قبلاً غیرفعال شده بود را دوباره فعال کرده و امکان چاپ از آن را فراهم کنید.

ساختار دستور

ساختار کلی دستور cupsenable به صورت زیر است

```
cupsenable [options] printer_name
```

options اختیارات مختلفی هستند که می توانید برای پیکربندی نحوه فعال شدن چاپگر از آنها استفاده کنید.

printer_name نام چاپگری است که می خواهید فعال کنید.

گزینه های دستور:

- h نمایش راهنمای دستور
- l لیست کردن تمام چاپگرهای موجود
- d فعال کردن چاپگر به صورت پیش فرض
- p فعال کردن چاپگر به صورت اشتراکی
- m مشخص کردن مدل چاپگر
- u مشخص کردن نام کاربری برای چاپ از طریق شبکه
- g مشخص کردن گروهی که کاربران برای چاپ از طریق شبکه باید عضو آن باشند.

مثال های دستور:

فعال‌سازی یک چاپگر خاص: در این مثال، یک چاپگر با نام " "printer_nameفعال‌سازی می‌شود.

```
cupsenable printer_name
```

فعال‌سازی چاپگر با نام " "printer_nameبر روی سرور خاص: در این مثال، چاپگر با نام " "printer_nameبر روی سرور با آدرس " "server_addressو پورت " "port_numberفعال‌سازی می‌شود.

```
cupsenable -h server_address:port_number printer_name
```

فعال‌سازی تمام چاپگرهای موجود در سیستم: این دستور برای فعال‌سازی تمام چاپگرهای موجود در سیستم استفاده می‌شود.

```
cupsenable --all
```

فصل نوزدهم

فصل نوزدهم

پاک کردن داده ها در linux

دستور shred

دستور shred برای حذف امن فایل‌ها در سیستم‌عامل‌های یونیکس و شبه یونیکس مانند لینوکس و macOS استفاده می‌شود. این دستور با داده‌های فایل با الگوهای تصادفی، بازیابی اطلاعات را دشوار یا غیرممکن می‌کند.

ساختار دستور:

```
shred [options] file...
```

گزینه‌های دستور:

- f اجبار به حذف فایل، حتی اگر فقط خواندنی باشد.
- n تعداد دفعات überschreiben داده‌ها را مشخص می‌کند. پیش‌فرض ۳ بار است.
- u پس از حذف فایل، فضای دیسک را آزاد می‌کند.
- v اطلاعات مربوط به فرآیند حذف را نمایش می‌دهد.
- z قبل از überschreiben، فایل را با صفر پر می‌کند.
- الگوی überschreiben را مشخص می‌کند. پیش‌فرض الگوی DoD 5220.22M است.

مثال‌های دستور:

حذف امن یک فایل:

```
shred file.txt
```

حذف امن یک فایل و آزاد کردن فضای دیسک:

```
shred -u file.txt
```

حذف امن یک فایل با ۷ بار überschreiben:

```
shred -n 7 file.txt
```

حذف امن یک فایل با الگوی Gutmann:

```
shred -o gutmann file.txt
```

حذف امن تمام فایل‌های یک پوشه:

```
shred -v /path/to/folder/*
```

نکته:

استفاده از دستور shred می‌تواند زمان‌بر باشد، به خصوص برای فایل‌های بزرگ.

قبل از استفاده از دستور shred، از اینکه واقعاً می‌خواهید فایل را حذف کنید، مطمئن شوید.

دستور wipe

دستور wipe برای پاک کردن امن اطلاعات از روی دیسک‌های سخت، درایوهای فلش، و سایر دستگاه‌های ذخیره‌سازی استفاده می‌شود. این دستور با überschreiben کردن داده‌ها، بازیابی آنها را دشوار یا غیرممکن می‌کند.

ساختار دستور:

```
wipe [options] <device>
```

options گزینه‌های مختلفی برای دستور wipe وجود دارد که می‌توانید از آنها برای سفارشی کردن فرآیند پاک کردن استفاده کنید.

Device نام یا مسیر دستگاهی که می‌خواهید اطلاعات آن را پاک کنید.

گزینه‌های دستور:

- a- تمام فضای خالی دیسک را پاک می‌کند.
- c- فقط فضای اشغال شده توسط فایل‌ها را پاک می‌کند.
- f- قبل از پاک کردن، از شما تأیید نمی‌گیرد.
- i- اطلاعات را به صورت تصادفی überschreiben می‌کند.
- n- تعداد دفعات überschreiben اطلاعات را مشخص می‌کند.
- p- یک الگوی خاص را برای überschreiben اطلاعات استفاده می‌کند.
- r- تمام زیرشاخه‌های یک دایرکتوری را پاک می‌کند.
- s- اطلاعات را به صورت امن پاک می‌کند.
- v- اطلاعات مربوط به فرآیند پاک کردن را نمایش می‌دهد.

مثال‌های دستور:

پاک کردن تمام فضای خالی یک دیسک

```
wipe -a /dev/sda
```

پاک کردن فقط فضای اشغال شده توسط فایل‌ها در یک دایرکتوری

```
wipe -c /home/user/data
```

پاک کردن امن اطلاعات یک درایو فلش

```
wipe -s /dev/sdb
```

پاک کردن اطلاعات یک دیسک با überschreiben تصادفی ۱۰ بار

```
wipe -i 10 /dev/sdc
```

پاک کردن اطلاعات یک دیسک با استفاده از الگوی خاص

```
wipe -p “0xdeadbeef” /dev/sdd
```

نکته قبل از استفاده از دستور wipe، حتماً از اطلاعات خود نسخه پشتیبان تهیه کنید. این دستور می‌تواند تمام اطلاعات موجود در دستگاه شما را پاک کند.

اول ان را نصب کنید و سپس دستور زیر را اجرا کنید:

```
$ sudo apt install wipe
$ sudo yum install wipe
```

دستور زیر همه چیز را در فهرست private پاک خواهد کرد.

```
$ wipe -rfi private / *
```

```
aaronkilik@tecmint ~ $ wipe -rfi private/*
Renaming      private/configurations-keys ->      private/T3C6
                        File private/configurations-keys (11 bytes) wiped
Renaming            private/passwords.doc ->            privat
                        File private/passwords.doc (62031 bytes) wiped
Renaming          private/serial-keys.doc ->          private/
                        File private/serial-keys.doc (59826 bytes) wiped
Renaming private/usernames-and-passwds.od -> private/6Ww1WorJ32M
                            File private/usernames-and-passwds.od (10967 bytes) wiped
Operation finished.
4 files wiped and 0 special files ignored in 0 directories, 0 symlinks removed but not followed,
0 errors occured.
aaronkilik@tecmint ~ $
```

دستور srm

Secure-deleteمجموعه ای از ابزار حذف فایل امن است که حاوی ابزار srm (secure_deletion)است که برای پاک کردن فایل ها به صورت ایمن استفاده می شود. ابتدا باید آن را با استفاده از دستور مربوطه نصب کنید:

```
$ sudo apt install secure-delete
$ sudo yum install secure-delete
```

پس از نصب می توانید از ابزار SRM برای حذف فایل ها یا فهرست ها به صورت ایمن بر روی یک linux system به صورت زیر استفاده کنید.

ساختار دستور:

```
srm [OPTIONS] FILE...
```

گزینه‌های دستور:

- f بدون پرسیدن تأیید، فایل‌ها را حذف می‌کند.
- i قبل از حذف، نام فایل‌ها را به کاربر نشان می‌دهد.
- l نام فایل‌ها را به جای حذف، در یک لیست ذخیره می‌کند.
- n فقط شبیه‌سازی عملیات حذف را انجام می‌دهد.
- r به طور بازگشتی، تمام محتویات یک دایرکتوری را حذف می‌کند.
- v جزئیات عملیات حذف را به کاربر نشان می‌دهد.
- z پس از حذف، فضای خالی دیسک را با صفرها پر می‌کند.

مثال های دستور:

حذف یک فایل بدون تأیید:

```
srm -f file.txt
```

نمایش نام فایل‌ها قبل از حذف:

```
srm -i file1.txt file2.txt
```

حذف بازگشتی یک دایرکتوری:

```
srm -r directory
```

شبیه‌سازی عملیات حذف:

```
srm -n file.txt
```

حذف امن یک فایل و پر کردن فضای خالی دیسک با صفرها:

```
srm -vz file.txt
```

نکات دستور:

دستور srm می‌تواند برای حذف امن فایل‌های حساس، مانند اطلاعات بانکی یا رمز عبور، استفاده شود.

استفاده از گزینه z– می‌تواند زمان حذف فایل‌ها را به طور قابل‌توجهی افزایش دهد.

قبل از استفاده از دستور srm، حتماً از اینکه فایل‌ها را به درستی انتخاب کرده‌اید، مطمئن شوید.

دستور Sfill-Secure

sfill بخشی از جعبه ابزار امن حذف است، یک دیسک امن و دیسک پاک کننده فضایی است و فایل ها را بر روی فضای دیسک آزاد در یک روش امن حذف می کند Sfill.فضای آزاد در پارتیشن مشخص شده را بررسی می کند و آن را با داده های تصادفی از/ dev / urandom پر می کند. دستور زیر در قسمت پارتیشن root من sfillرا اجرا می کند، با این که سوئیچ v-فعال کردن حالت متنی:

```
$ sudo sfill -v /home/hossein/tmp/
```

با فرض اینکه یک پارتیشن جداگانه ایجاد کرده اید، برای نگهداری فهرستهای خانگی کاربر system عادی، می توانید فهرست را در آن پارتیشن مشخص کنید تا درخواست آن را انجام دهید:

```
$ sudo sfill -v / home / username
```

چند محدودیتی که می توانید در صفحه شخصی بخوانید، محدودیت هایی است که همچنین می توانید سویچ های اضافی و دستورالعمل ها را پیدا کنید:

```
$ man spfill
```

دستور smem

smemپاک کننده memory امن است، طراحی شده است که داده ها موجود در memory (RAM) را به صورت امن محافظت می کند. ابتدا آن را به نام smem، اما به دلیل system های Debain، یک بسته دیگر به نام مصرف memory گزارش شده در هر فرایند و بر اساس کاربر وجود دارد، توسعه دهنده تصمیم گرفت آن را تغییر نام دهد.همچنین دستور sdmem برای مدیریت حافظه در سیستم‌های لینوکس استفاده می‌شود. این دستور به شما امکان می‌دهد تا حافظه را به فرآیندها اختصاص داده و از آنها آزاد کنید، و همچنین می‌توانید از آن برای مشاهده اطلاعات مربوط به استفاده از حافظه استفاده کنید.

ساختار دستور:

ساختار کلی دستور sdmem به شرح زیر است:

```
sdmem [options] command [arguments]
```

options :گزینه‌های مختلفی هستند که می‌توانید برای تغییر رفتار دستور sdmem از آنها استفاده کنید.

Command :دستوری است که می‌خواهید اجرا کنید.

Arguments :استدلال‌هایی هستند که به دستور می‌دهید.

گزینه‌های دستور

- h– این گزینه راهنمای دستور را نمایش می‌دهد.
- l– این گزینه لیست تمام فرآیندها را به همراه اطلاعات مربوط به استفاده از حافظه آنها نمایش می‌دهد.
- p– این گزینه اطلاعات مربوط به استفاده از حافظه یک فرآیند خاص را نمایش می‌دهد.
- a– این گزینه به شما امکان می‌دهد تا حافظه را به یک فرآیند اختصاص دهید.

- f- این گزینه به شما امکان می‌دهد تا حافظه را از یک فرآیند آزاد کنید.

نمایش لیست تمام فرآیندها به همراه اطلاعات مربوط به استفاده از حافظه آنها:

```
sdmem -l
```

نمایش اطلاعات مربوط به استفاده از حافظه یک فرآیند خاص:

```
sdmem -p 1234
```

اختصاص ۱۰ مگابایت حافظه به فرآیند با شناسه ۱۲۳۴:

```
sdmem -a 10m 1234
```

آزاد کردن ۵ مگابایت حافظه از فرآیند با شناسه ۱۲۳۴:

```
sdmem -f 5m 1234
```

برای اطلاعات بیشتر در مورد دستور sdmem می‌توانید به صفحه man آن مراجعه کنید:

```
man sdmem
```

دستور exiftool

exiftool خواندن و نوشتن اطلاعات متا در فایل ها. دستور exiftool یک ابزار قدرتمند برای خواندن، نوشتن و ویرایش متادیتاها (اطلاعات داخلی) در فایل‌های تصویری و ویدئویی است. این ابزار قابلیت آنالیز و تغییر اطلاعات مانند اطلاعات دوربین، تاریخ گرفتن عکس، مکان جغرافیایی و سایر اطلاعات مربوط به فایل را داراست. دستور exiftool دارای گزینه‌های مختلفی است که به شما امکان می‌دهد اطلاعات متادیتا را با استفاده از این ابزار به صورت دلخواه خوانده و ویرایش کنید.

گزینه های دستور:

```
$ exiftool [ OPTIONS ] [- TAG ...] [-- TAG ...] FILE ...
$ exiftool [ OPTIONS ] - TAG [+-<]=[ VALUE ]... FILE ...
$ exiftool [ OPTIONS ] tagsFromFile SRCFILE [- SRCTAG [> DSTTAG ]...] FILE
$ exiftool [ -ver | -list[w|f|wf|g[ NUM ]|d|x] ]
For specific examples, see the EXAMPLES sections below.
```

گزینه -ver نمایش نسخه exiftool

```
$ exiftool -ver
```

گزینه -listw نمایش تگ‌های قابل نوشتن

```
$ exiftool -listw
```

گزینه -listf نمایش فرمت‌های قابل پشتیبانی

```
$ exiftool -listf
```

گزینه -listwf نمایش فرمت‌های قابل نوشتن

```
$ exiftool -listwf
```

گزینه -listg نمایش گروه‌های متادیتا

```
$ exiftool -listg
```

گزینه -list NUM نمایش تگ‌های گروه متادیتا مشخص شده

```
$ exiftool -list1
```

گزینه -listd نمایش تگ‌های متادیتا مورد استفاده برای تاریخ

```
$ exiftool -listd
```

گزینه -listx نمایش تگ‌های متادیتا مورد استفاده برای موارد خاص

```
$ exiftool -listx
```

این بخش شامل گزینه‌های دیگری است که می‌توانید با exiftoolاستفاده کنید. به عنوان مثال

```
$exiftool -overwrite_original -r -P DIRECTORY
```

TAG-با استفاده از این آرگومان، می‌توانید تگ یا تگ‌های متادیتا مشخصی را در فایل‌ها بخوانید. به عنوان مثال

```
$exiftool -DateTimeOriginal FILE.jpg
```

TAG--این آرگومان مشابه آرگومان قبلی است، اما با استفاده از خط تیره ("--") می‌توانید تگ‌هایی که با حروف فارسی یا کاراکترهای خاص شروع می‌شوند را نیز بخوانید. به عنوان مثال

```
$exiftool --XMP-عنوان FILE.jpg
```

FILEبا استفاده از این آرگومان، می‌توانید فایل یا فایل‌هایی که می‌خواهید اطلاعات متادیتا آن‌ها را بخوانید یا ویرایش کنید، مشخص کنید. به عنوان مثال

```
$exiftool image.jpg
```

فصل بیستم

فصل بیستم

مدیریت فایلهای آرشیو و پشتیبانی

مدیریت فایلهای آرشیو دستور tar

برای آرشیو است که توسط بسیاری از مدیران linux system / یونیکس برای تهیه نسخه پشتیبان استفاده می شود.دستور tar به منظور جمع آوری فایلها و فهرستها در فایل آرشیو فشرده با نام تجاری tarballیا tar، gzip وbzip درlinux استفاده می شود. tar .به طور گسترده ای مورد استفاده قرار می گیرد برای ایجاد فایل های آرشیو فشرده و می تواند به راحتی از یک دیسک به دیسک دیگر یا ماشین به ماشین منتقل می شود. در این کتاب ما قصد بررسی و بحث در مورد نمونه های مختلف دستور tar را از جمله نحوه ایجاد فایل های بایگانی با استفاده ازtar ، tar.gzو tar.bz2فشرده سازی، چگونگی استخراج فایل بایگانی، استخراج یک فایل، مشاهده محتوای فایل ، فایل را تأیید کنید .

ایجاد آرشیو فایلtar

دستور زیر مثال زیر یک فایل آرشیو tarرا برای hosein -14-09-12.tar برای یک فهرست hosein / home/ در فهرست کاری فعلی ایجاد می کند.

ساختار دستور:

```
tar [options] [archive] [files or directories]
```

اجزای دستور:

- options گزینه‌هایی برای تعیین نحوه عملکرد دستور
- archive نام فایل آرشیو
- files or directories لیست فایل‌ها یا دایرکتوری‌هایی که می‌خواهید آرشیو یا استخراج شوند

گزینه‌های دستور:

- c ایجاد یک آرشیو جدید
- x استخراج محتویات آرشیو
- t نمایش لیست محتویات آرشیو
- v نمایش جزئیات عملیات
- f تعیین نام فایل آرشیو
- z فشرده‌سازی آرشیو با gzip
- j فشرده‌سازی آرشیو با bzip2
- l نمایش پیوندهای نمادین به جای دنبال کردن آنها
- p حفظ مجوزها و مالکیت فایل‌ها
- m حفظ زمان‌های اصلاح فایل‌ها

مثال های دستور :

ایجاد فایل آرشیو tar.gz

برای ایجاد فایل آرشیو فشرده gzipما از این گزینه به عنوان z استفاده می کنیم.به عنوان مثال دستور زیر یک فایل MyImages-14-09-12.tar.gzفشرده برای فهرست home / MyImages/ایجاد می کند.

```
# tar cvzf  MyImages-14-09-12.tar.gz / home / MyImages
```

یا

```
# tar cvzf  MyImages-14-09-12.tgz / home / MyImages
```

/ home / MyImages /
/home/MyImages/Sara-Khan-and-model-Priyanka-Shah.jpg
/home/MyImages/RobertKristenviolent101201.jpg
/home/MyImages/Justintimerlake101125.jpg
/home/MyImages/Mileyphoto101203.jpg
/home/MyImages/JenniferRobert101130.jpg
/home/MyImages/katrinabarbiedoll231110.jpg
/home/MyImages/the-japanese-wife-press-conference.jpg
/home/MyImages/ReesewitherspoonCIA101202.jpg
/home/MyImages/yanaguptabaresf231110.jpg

ایجاد فایل آرشیو tar.bz2

ویژگی bz2فشرده شده و فایل بایگانی کمتر از اندازه gzipرا ایجاد می کند.فشرده سازی bz2زمان بیشتری را برای فشرده سازی و فشرده سازی فایلها در مقایسه با gzipمی گیرد که زمان کمتری را می گیرد.برای ایجاد یک فایل tar کاملا فشرده ما از گزینه jاستفاده می کنیم.دستور زیر مثال زیر فایل phpfiles-org.tar.bz2را برای یک فهرست / home / php ایجاد می کند.

```
# tar cvfj Phpfiles-org.tar.bz2 / home / php
```

یا

```
# tar cvfj Phpfiles-org.tar.tbz / home / php
```

یا

```
# tar cvfj Phpfiles-org.tar.tb2 / home / php
```

/ home / php /
/home/php/iframe_ew.php
/home/php/videos_all.php
/home/php/rss.php
/home/php/index.php
/home/php/vendor.php
/home/php/video_title.php
/home/php/report.php
/home/php/object.html
/home/php/video.php

ایجاد فایل آرشیو Untar وtar

برای جدا کردن یا استخراج یک فایلtar ، فقط دستور زیر را با استفاده از سویچ x[1] صادر کنید.به عنوان مثال دستور زیر فایل public_html-14-09-12.tarرا در فهرست فعلی باز خواهد کرد.

```
# tar -xvf thumbnails-14-09-12.tar.gz
```

/ home / public_html / videos / thumbnails /
/home/public_html/videos/thumbnails/katdeepika231110.jpg

---

[1] extract

```
/home/public_html/videos/thumbnails/katrinabarbiedoll231110.jpg
/home/public_html/videos/thumbnails/onceuponatime101125.jpg
/home/public_html/videos/thumbnails/playbutton.png
/home/public_html/videos/thumbnails/ReesewitherspoonCIA101202.jpg
/home/public_html/videos/thumbnails/snagItNarration.jpg
/home/public_html/videos/thumbnails/Minissha-Lamba.jpg
/home/public_html/videos/thumbnails/Lindsaydance101201.jpg
/home/public_html/videos/thumbnails/Mileyphoto101203.jpg
```

برای غیرفعال کردن فایل tar.bz2بسیار فشرده، فقط از دستور زیر استفاده کنید.دستور زیر مثال زیر تمام فایل های flv.را از فایل آرشیو جدا می کند.

```
# tar -xvf videos-14-09-12.tar.bz2
```

```
/home/public_html/videos/flv/katrinabarbiedoll231110.flv
/home/public_html/videos/flv/BrookmuellerCIA101125.flv
/home/public_html/videos/flv/dollybackinbb4101125.flv
/home/public_html/videos/flv/JenniferRobert101130.flv
/home/public_html/videos/flv/JustinAwardmovie101125.flv
/home/public_html/videos/flv/Lakme-Fashion-Week.flv
/home/public_html/videos/flv/Mileyphoto101203.flv
/home/public_html/videos/flv/Minissha-Lamba.flv
```

فهرست محتوای فایل tar

برای فهرست محتویات فایل آرشیوtar ، فقط دستور زیر را با سویچ t لیست محتوا اجرا کنید.دستور زیر محتوی فایل uploadprogress.tarرا فهرست می کند.

```
# tar -tvf uploadprogress.tar
```

```
-rw-r - r   -- chregu / staff 2276 2011-08-15 18:51:10 package2.xml
-rw-r - r   -- chregu / ۱۸:۵۱:۱۰ ۱۵-۰۸-۲۰۱۱ ۷۸۷۷ uploadprogress / examples / index.php
-rw-r - r   -- chregu / staff 1685 2011-08-15 18:51:10 uploadprogress / examples / server.php
-rw-r - r   -- chregu / staff 1697 2011-08-15 18:51:10 uploadprogress / examples / info.php
-rw-r - r   -- chregu / ۱۸:۵۱:۱۰ ۱۵-۰۸-۲۰۱۱ ۳۶۷ uploadprogress / config.m4
-rw-r - r   -- chregu / ۱۸:۵۱:۱۰ ۱۵-۰۸-۲۰۱۱ ۳۰۳ uploadprogress / config.w32
-rw-r - r   -- chregu / ۱۸:۵۱:۱۰ ۱۵-۰۸-۲۰۱۱ ۳۵۶۳ uploadprogress / php_uploadprogress.h
-rw-r - r   -- chregu / staff 15433 2011-08-15 18:51:10 uploadprogress / uploadprogress.c
-rw-r - r   -- chregu / staff 1433 2011-08-15 18:51:10 package.xml
```

فهرست محتوای فایل tar.gz

دستورالعمل زیر را برای لیست محتویات فایل tar.gzاستفاده کنید.

```
# tar -tvf staging. hosein .com.tar.gz
```

```
-rw-r - r   -- root / root 0 2012-08-30 04:03:57 staging. hosein .com-access_log
-rw-r - r   -- root / root 587 2012-08-29 18:35:12 staging. hosein .com-access_log.1
-rw-r - r   -- root / root 156 2012-01-21 07:17:56 staging. hosein .com-access_log.2
-rw-r - r   -- root / root 156 2011-12-21 30:56 staging. hosein .com-access_log.3
-rw-r - r   -- root / root 156 2011-11-20 17:28:24 staging. hosein .com-access_log.4
-rw-r - r   -- root / root 0 2012-08-30 04:03:57 staging. hosein .com-error_log
-rw-r - r   -- root / root 3981 2012-08-29 18:35:12 staging. hosein .com-error_log.1
-rw-r - r   -- root / root 211 2012-01-21 07:17:56 staging. hosein .com-error_log.2
-rw-r - r   -- root / root 211 2011-12-21 30:56 staging. hosein .com-error_log.3
-rw-r - r   -- root / root 211 2011-11-20 17:28:24 staging. hosein .com-error_log.4
```

فهرست محتوای فایل tar.bz2

برای فهرست محتویات فایل tar.bz2، دستور زیر را صادر کنید.

```
# tar -tvf   Phpfiles-org.tar.bz2
drwxr-xr-x root / root 0 2012-09-15 03:06:08 / home / php /
-rw-r - r   -- root / root 1751 2012-09-15 03:06:08 /home/php/iframe_ew.php
-rw-r - r   -- root / root 11220 2012-09-15 03:06:08 /home/php/videos_all.php
-rw-r - r   -- root / root 2152 2012-09-15 03:06:08 /home/php/rss.php
-rw-r - r   -- root / root 3021 2012-09-15 03:06:08 /home/php/index.php
-rw-r - r   -- root / root 2554 2012-09-15 03:06:08 /home/php/vendor.php
-rw-r - r   -- root / root 406 2012-09-15 03:06:08 /home/php/video_title.php
```

برای استخراج یک فایل به نام cleanfiles.shازیک فایل cleanfiles.sh.tar از دستور زیر استفاده کنید.

```
# tar  -xvf   cleanfiles.sh.tar  cleanfiles.sh
```

یا

```
# tar    --extract    --file = cleanfiles.sh.tar cleanfiles.sh
```

برای استخراج یک فایل hosein backup.xmlاز فایل آرشیو hosein backup.tar.gz، از دستور زیر استفاده کنید.

```
# tar -zxvf  hosein backup.tar.gz  hosein backup.xml
```

یا

```
# tar    --extract    --file =  hosein backup.tar.gz  hosein backup.xml
```

برای استخراج یک فایل به نام index.phpاز فایل Phpfiles-org.tar.bz2از گزینه زیر استفاده کنید.

```
# tar -jxvf Phpfiles-org.tar.bz2 home / php / index.php
```

یا

```
# tar    --extract    --file = Phpfiles-org.tar.bz2 /home/php/index.php
/home/php/index.php
```

برای استخراج یا جدا کردن چندین فایل از فایل آرشیو tar، tar.gzو . tar.bz2به عنوان مثال :
دستور زیر "file1" " file 2 "را از فایل های بایگانی استخراج[1] می کند.

```
# tar -xvf  hosein -14-09-12.tar " file 1 " " file 2 "
```

یا

[1] extract

```
# tar -zxvf MyImages-14-09-12.tar.gz “ file 1” “ file2 “
```

یا

```
# tar -jxvf Phpfiles-org.tar.bz2 “ file1 “ “ file2”
```

برای استخراج گروهی از فایلهای ما از استخراج بر اساس کلمات استفاده می‌کنیم.برای مثال ، برای استخراج گروهی از تمام فایل هایی که الگوی آنها با phpآغاز می شود از یک فایل آرشیو tar، tar.gzو tar.bz2 .

```
# tar -xvf Phpfiles-org.tar - wildcards ‘* .php’
```

```
# tar -zxvf Phpfiles-org.tar.gz - wildcards ‘* .php’
```

```
# tar -jxvf Phpfiles-org.tar.bz2 - wildcards ‘* .php’
```

```
/home/php/iframe_ew.php
/home/php/videos_all.php
/home/php/rss.php
/home/php/index.php
/home/php/vendor.php
/home/php/video_title.php
/home/php/report.php
/home/php/video.php
```

برای اضافه کردن فایل ها یا فهرست ها به فایل های آرشیو شده موجود، از گزینه (append) rاستفاده می کنیم.برای مثال : فایل xyz.txtو فهرست phpرا به فایل آرشیو hosein -14-09-12.tarاضافه میکنیم.

```
# tar -rvf  hosein - 14-09-12.tar xyz.txt
```

```
# tar -rvf  hosein -14-09-12.tar php
```

بررسی اندازه فایل های آرشیو

برای بررسی حجم فایل tar، tar.gzو tar.bz2از دستور زیر استفاده کنید.به عنوان مثال دستور زیر حجم فایل بایگانی را در کیلوبایت (KB) نمایش می دهد.

```
# tar -czf -  hosein -14-09-12.tar |WC-C
12820480
# tar -czf - MyImages-14-09-12.tar.gz |WC-C
112640
# tar -czf - Phpfiles-org.tar.bz2 |WC-C
20480
```

دستور gzip

دستور gzip در لینوکس/یونیکس برای فشرده سازی/فشرده سازی داده ها استفاده می شود. دستور gzip در لینوکس یک ابزار فشرده‌سازی فایل‌ها است که برای کاهش حجم فایل‌ها و صرفه‌جویی در فضای ذخیره سازی استفاده می‌شود. این ابزار از الگوریتم فشرده‌سازی DEFLATE استفاده می‌کند که در بسیاری از موارد منجر به کاهش قابل توجه حجم فایل‌ها می‌شود.

دستور gzip معمولاً به صورت زیر استفاده می‌شود:

```
$gzip [options] [file]
```

گزینه های دستور:

- d– یا –decompress: برای بازگرداندن فایل‌های فشرده شده به حالت اولیه استفاده می‌شود.
- c– یا –stdout: خروجی را به صورت استاندارد (stdout) نمایش می‌دهد، بدون ایجاد فایل فشرده شده جدید.
- k– یا –keep: فایل اصلی را حفظ می‌کند و فایل فشرده شده را با پسوند .gz ایجاد می‌کند.
- r– یا –recursive: عملیات فشرده‌سازی را به صورت بازگشتی بر روی فهرست‌ها و زیرمجموعه‌ها انجام می‌دهد.
- v– یا –verbose: نمایش جزئیات بیشتر در هنگام اجرای دستور.
- File: فایلی که می‌خواهید فشرده شود. می‌توانید یک یا چند فایل را به gzip ارائه دهید. اگر از گزینه r– استفاده کنید، می‌توانید یک فهرست را نیز به عنوان ورودی به gzip بدهید.

مثال های دستور:

یک فایل را فشرده کنید

دستور:

```
$ zip file_name
```

یک فایل را از حالت فشرده خارج کنید

دستور:

```
$ gzip -d archive_01.gz
```

فشرده سازی چندین فایل:

دستور:

```
$ gzip file_name_01 file_name_02 file_name_03
```

چند فایل را از حالت فشرده خارج کنید:

دستور:

```
$ gzip -d archive_01.gz archive_02.gz archive_03.gz
```

فشرده سازی یک فهرست:

دستور:

```
$ gzip -r directory_name
```

یک فهرست را از حالت فشرده خارج کنید:

دستور:

```
$ gzip -dr directory_name
```

خروجی کامل (جزئیات) در حین فشرده سازی:

دستور:

```
$ gzip -v file_name
```

دستور zcat

دستور zcat در سیستم عامل لینوکس یک ابزار است که برای خواندن و نمایش محتوای فایل‌های فشرده با فرمت gzip استفاده می‌شود. این دستور عملکرد مشابه دستور cat را دارد، با این تفاوت که می‌تواند فایل‌های فشرده gzip را مستقیماً باز کرده و محتوای آن‌ها را به صورت خوانا نمایش دهد.

ساختار دستور:

```
zcat [OPTIONS] FILE
```

گزینه‌های دستور:

- f تلاش برای فشرده سازی فایل، حتی اگر به نظر نمی‌رسد فشرده شده باشد.
- h نمایش راهنمای دستور.
- q عدم نمایش کردن پیام‌های خطا.
- v نمایش اطلاعات نسخه.
- z سطح فشرده‌سازی را به صورت عددی بین ۰ تا ۹ (۰ کمترین و ۹ بیشترین) تعیین می‌کند.

نمایش محتوای یک فایل فشرده gzip:

```
$zcat file.txt.gz
```

ذخیره محتوای یک فایل فشرده gzip در یک فایل معمولی:

```
$zcat file.txt.gz > output.txt
```

استفاده از pipeکشی (Piping) بین zcatو دستور دیگری:

```
$zcat file.txt.gz | grep “keyword”
```

در این مثال، ما محتوای فایل فشرده gzip را با zcatباز می‌کنیم و سپس خروجی را به grepمنتقل می‌کنیم تا رشته “keyword” را در آن جستجو کنیم.

استفاده از یک الگوی دسته‌بندی در یک فایل فشرده gzip:

```
$zcat file.txt.gz | awk ‘{print $1}’ | sort | uniq
```

در این مثال، ما محتوای فایل فشرده gzip را با zcatباز می‌کنیم و سپس با استفاده از awk، ستون اول هر خط را چاپ کرده و سپس با استفاده از sortو uniq، این ستون‌ها را مرتب و بدون تکرار چاپ می‌کنیم.

نمایش محتوای یک فایل gzip:

```
zcat file.gz
```

نمایش محتوای یک فایل gzip به صورت صفحه بندی شده:

```
zcat file.gz | less
```

نمایش محتوای ۱۰ خط اول یک فایل gzip:

```
head -n 10 file.gz | zcat
```

بررسی اینکه آیا یک فایل gzip است یا نه:

```
zcat -f file
```

فشرده سازی یک فایل با سطح فشرده‌سازی ۵:

```
zcat -z 5 file > output.gz
```

نمایش اطلاعات نسخه:

```
zcat -v
```

دستور ar

از دستور **ar** برای ایجاد ، اصلاح و استخراج پرونده ها از بایگانی استفاده می شود .بایگانی مجموعه ای از پرونده های دیگر است که ساختار خاصی دارند و می توان پرونده های جداگانه را از آنها استخراج کرد .پرونده های فردی به عنوان اعضای بایگانی نامیده می شوند. برنامه ar آرشیوها را ایجاد، اصلاح و استخراج می‌کند. آرشیو یک فایل منفرد است که مجموعه ای از فایل های دیگر را در

ساختاری نگهداری می‌کند که شامل محتویات، حالت (مجوزها)، مُهر زمانی، مالک و گروه فایل های اصلی است.همه آنها در هنگام استخراج قابل بازیابی هستند. ar می‌تواند آرشیوهایی را نگهداری کند که نام‌هایی با هر طولی داشته باشند. محدودیت اغلب ۱۵ کاراکتر است (فرمت های معمولی مربوط به a.out) یا ۱۶ کاراکتر (از قالب های معمولی مربوط به coff). ar به عنوان یک ابزار باینری در نظر گرفته می شود.

ساختار دستور:

```
ar [OPTIONS] archive_name member_files
```

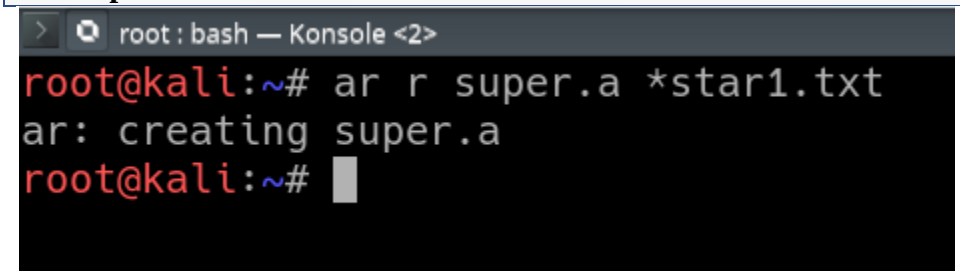
```
$ ar [ -c ] [ -l ] [ -g | -o ] [ -s ] [ -v ] [ -C ] [ -T ] [ -z ] { -m | -r [ -u ] } [ { -a | -b | -i } PositionName ] [ -X {32|64|32_64|d64|any}] ArchiveFile File ...
```

گزینه های دستور:

r :این برای ایجاد بایگانی، قرار دادن پرونده ها در بایگانی استفاده می شود .این با **q** متفاوت است زیرا اعضای قبلی را حذف می کند .اگر نام پرونده هر عضو وجود نداشته باشد خطایی ایجاد می کند .به طور پیش فرض ، یک عضو جدید در انتهای پرونده اضافه می کند.

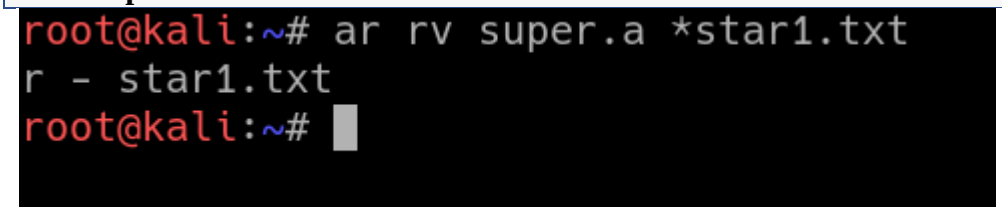
مثال :فرض کنید فایلی به نام star1 وجود دارد و می خواهید بایگانی با نام **"super"** بدون اصلاح کننده ایجاد کنید ، سپس می توانید از دستور زیر استفاده کنید:

```
ar r super.a * star1
```



این یک بایگانی با عضو *star1* ایجاد می کند . با اصلاح کننده به نظر می رسد به صورت زیر است:

```
ar rv super.a * star1.txt
```

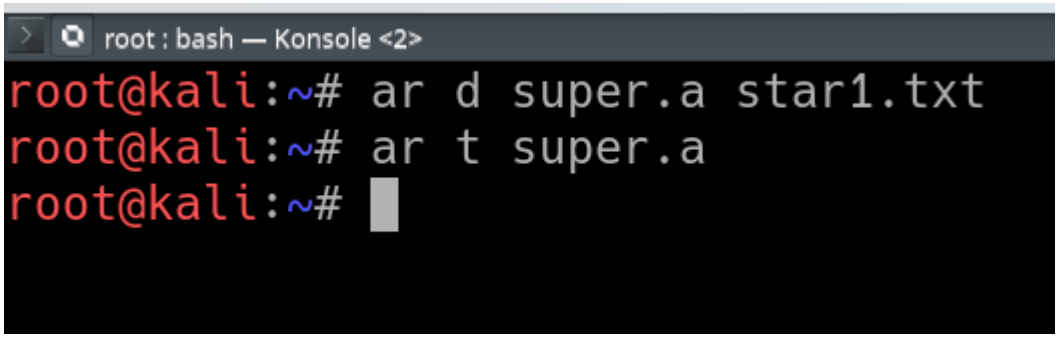


توجه :اصلاح کننده *v* یک خط یا خروجی با حرف *a* یا *r* می دهد که نشان می دهد فایل پیوست شده است یا خیر.

d :ماژول ها را از بایگانی حذف می کند .نام ماژول ها را به عنوان عضو مشخص کنید؛ هنگامی که modifier v را اضافه می کنید ، arهر یک از ماژول ها را لیست می کند زیرا حذف شده است.

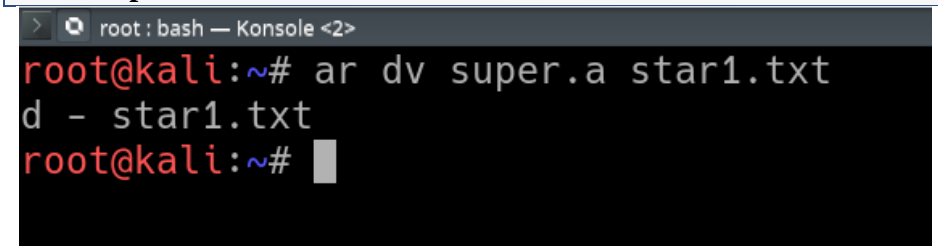
```
ar d super.a star1.txt
```

در مورد قبلی ما ، از star1.txt در بایگانی *super.a* *استفاده کردیم* ، اکنون آن پرونده را از آنجا حذف خواهیم کرد.



از این رو *star.txt* بایگانی *super.a* را ذخیره کرده *بودیم* ، پس از استفاده از *d* بدون اصلاح کننده ، فقط پرونده را حذف کرد .اکنون می توانید همان مثال را با modifier *v* *مشاهده کنید*

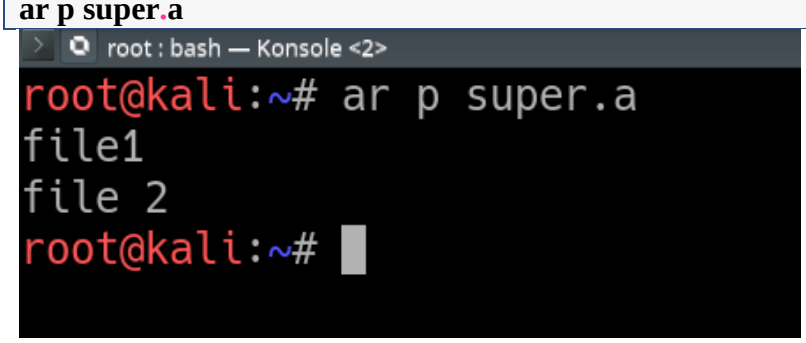
ar dv super.a star1.txt



با استفاده از modifier *v* ماژول را لیست کرد زیرا آنها حذف می شوند.

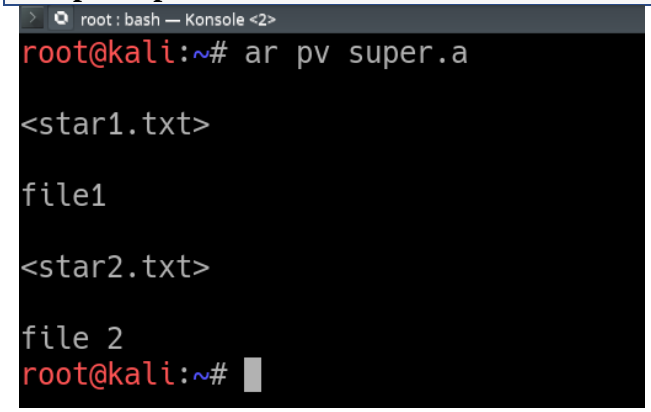
p :این گزینه برای چاپ اعضای مشخص شده بایگانی در یک پرونده خروجی استاندارد استفاده می شود ، اگر از modifierاستفاده نکنید ، عضو را چاپ می کند زیرا یک فایل خروجی است در حالی که اگر از modifier *v* استفاده می کنید ، نام آن را قبل از آن نشان می دهد در پرونده خروجی کپی شد.

ar p super.a



در اینجا خروجی را که در پرونده های ما *stat1.txt* و *star2.txt* نوشته شده بود ، داد ، اکنون اجازه دهید بررسی کنیم چه اتفاقی می افتد که ما از modifier *v* در آن استفاده می کنیم.

ar pv super.a



با استفاده از اصلاح کننده ، ابتدا نام فایل و سپس محتوای نوشته شده در داخل آن را دریافت کنید.

t :این محتویات بایگانی را به روشی ذکر شده نمایش می دهد ، معمولاً محتویات بایگانی را نشان می دهد اما اگر از تعدیل کننده O استفاده کنیم آنگاه جابجایی مربوط به هر یک از اعضا را نیز نشان می دهد.

```
root@kali:~# ar t super.a
star1.txt
star2.txt
star3.txt
root@kali:~#
```

در اینجا t نمایش محتوای آرشیو است.

x :اگر عضوی که استخراج می شود را نام نبرید ، هر عضو نامگذاری شده را از بایگانی استخراج می کند ، کل بایگانی را استخراج می کند .ما می توانیم از **v** modifier برای لیست کردن نام هر عضو استخراج شده استفاده کنیم.

```
root : bash — Konsole <2>
root@kali:~# ar xv super.a
x - star1.txt
x - star2.txt
x - star3.txt
root@kali:~#
```

سویچ v هر فایلی را که استخراج می شود نمایش می دهد و همچنین ما نامی را مشخص نکردیم بنابراین اگر اسامی عضو را مشخص کنیم ، فقط کل آن را استخراج می کند. این دستور مانند پرونده ما فایلهای نازکی را استخراج نمی کند ، فقط نشان می دهد که بایگانی را استخراج کرده است.

```
root@kali:~# ar xv super.a star1.txt
x - star1.txt
root@kali:~#
```

تصویر بالا استخراج را براساس نام عضو مشخص شده نشان می دهد و فایل لیست اصلاح کننده **v** استخراج می شود.

فشرده سازی فایل های bz2.با استفاده از ابزار bzip2 در linux خواهیم پرداخت.Bzip2یک ابزار فشرده سازی شناخته شده است و در بیشتر در همه توزیع های linux است. روش نصب ابزار:

```
$ sudo apt install bzip2
$ sudo yum install bzip2
$ sudo dnf install bzip2
```

دستور bzip2

می توانید فایل زیر را فشرده کنید، که از سویچ z– برای فشرده سازی فایل استفاده میشود. همچنین bzip2ابزاری قدرتمند برای فشرده سازی فایل ها در سیستم های لینوکس و Unix است. این ابزار از الگوریتم بُورُز–وِیلِر (Burrows-Wheeler) و کدگذاری هافمن (Huffman coding) برای فشرده سازی داده ها استفاده می کند و به طور قابل توجهی فایل ها را کوچکتر می کند.

ساختار دستور

```
bzip2 [options] <file(s)>
```

گزینه های دستور

- c فشرده سازی و چاپ خروجی به صورت استاندارد (stdout)
- d از حالت فشرده خارج کردن فایل
- f اجبار به فشرده سازی حتی اگر فایل قبلاً فشرده شده باشد
- k حفظ فایل اصلی پس از فشرده سازی
- l نمایش اطلاعات مربوط به فایل فشرده
- n تنظیم سطح فشرده سازی (۱ تا ۹، ۹ بالاترین)
- q فعال کردن حالت سکوت
- s فشرده سازی به صورت تک رشته ای
- t آزمایش صحت فایل فشرده
- V نمایش نسخه برنامه

مثال های دستور:

```
$ bzip2 filename
```

یا

```
$ bzip2 -z myfoldername
```

برای فشرده سازی یک فایل با پسوند tar. با استفاده از دستور:

```
$ bzip2 -z backup.tar
```

مهم :به طور پیش فرض، bzip2 فایلهای ورودی را در هنگام فشرده سازی یا پس فشرده سازی حذف می کند، برای حفظ فایل های ورودی، از گزینه k-یا keep -- استفاده نمایید.

علاوه بر این، سویچ f-یا force force-- را مجبور به بازنویسی یک فایل خروجی موجود می کند.

```
$ bzip2 -zk myfile
$ bzip2 -zk backup.tar
```

برای فشرده کردن یک فایل از سویچ decompress-- یا سویچ d- --decompressمانند زیر استفاده کنید:

```
$ bzip2 -d filename.bz2
```

برای مشاهده صفحه راهنمای راهنما bzip2و صفحهman ، دستور زیر را تایپ کنید:

```
$ bzip2 -h
$ man bzip2
```

مثال‌ها:

فشرده سازی یک فایل:

```
bzip2 myfile.txt
```

فشرده سازی چندین فایل:

```
bzip2 *.txt
```

فشرده سازی با سطح ۹:

```
bzip2 -n 9 myfile.txt
```

از حالت فشرده خارج کردن یک فایل:

```
bzip2 -d myfile.bz2
```

نمایش اطلاعات مربوط به فایل فشرده:

```
bzip2 -l myfile.bz2
```

حفظ فایل اصلی پس از فشرده سازی:

```
bzip2 -k myfile.txt
```

فشرده سازی و چاپ خروجی به صورت استاندارد:

```
bzip2 -c myfile.txt > myfile.bz2
```

فعال کردن حالت سکوت:

```
bzip2 -q myfile.txt
```

آزمایش صحت فایل فشرده:

```
bzip2 -t myfile.bz2
```

دستور bzless

دستور bzless در سیستم‌عامل‌های مبتنی بر Unix/Linux برای مشاهده محتوای یک فایل فشرده با فرمت Bzip2 به صورت صفحه به صفحه استفاده می‌شود. همچنین bzlessیک ابزار برای مشاهده فایل‌های فشرده شده با bzip2 است. این ابزار به شما امکان می‌دهد فایل‌های فشرده را به صورت صفحه‌ای مشاهده کنید، به طوری که نیازی به decompressing کل فایل قبل از مشاهده آن نیست.

ساختار دستور

```
bzless [OPTIONS] [FILE]...
```

گزینه‌های دستور:

- h, help نمایش راهنمای دستور
- V, version نمایش نسخه برنامه
- l, less از ابزار less برای نمایش فایل استفاده می‌کند
- m, more از ابزار more برای نمایش فایل استفاده می‌کند
- s, screenful تعداد خطوط نمایش داده شده در هر صفحه را مشخص می‌کند
- n, number شماره خطوط را در ابتدای هر خط نمایش می‌دهد
- q, quiet پیام‌های خطا را نمایش نمی‌دهد
- f, force حتی اگر فایل فشرده نباشد، آن را به عنوان یک فایل فشرده bzip2 در نظر می‌گیرد

مثال های دستور:

بدون گزینه: نمایش محتوای فایل فشرده Bzip2 به صورت صفحه به صفحه.

```
bzless file.bz2
```

خروجی دستور: نمایش محتوای فایل فشرده به صورت صفحه به صفحه

- c, --force: نمایش محتوای فایل Bzip2 حتی اگر به عنوان متن تصویری شناخته شود.

```
bzless -c image.bz2
```

خروجی دستور: نمایش محتوای فایل تصویری Bzip2

- N, --line-numbers: نمایش شماره خطوط در صفحات مختلف.

```
bzless -N text.bz2
```

خروجی دستور: نمایش محتوا با شماره خطوط در صفحات

- h, --help: نمایش راهنمای استفاده از دستور bzless.

```
bzless -h
```

خروجی دستور: نمایش راهنمای استفاده از دستور bzless

- V, --version: نمایش نسخه فعلی دستور bzless.

```
bzless -V
```

خروجی دستور: نمایش نسخه فعلی دستور bzless

دستور ark

این دستور یک ابزار بایگانی در محیط KDE است. برنامه ای برای مدیریت فرمتهای مختلف آرشیو در محیط KDE است. فالیهای فشرده را می توان از داخل Ark مشاهده، استخراج، ایجاد و اصلاح کرد. این برنامه می تواند فرمت های مختلفی مانند tar، gzip، bzip2، zip، rar و lha (در صورت نصب برنامه های خط فرمان مناسب) را مدیریت کند. اگر افزونه Konqueror Integration موجود در بسته kdeaddons را نصب کنید، Ark می‌تواند با Konqueror در محیط KDE برای مدیریت بایگانی‌ها کار کند.

ساختار دستور:

```
$ ark [--extract--extract-to--add--add-to--guess-name] [folder] [files] [archive] [KDE Generic Options] [Qt Generic Options]
```

مثال های دستور:

```
$ ark --extract-to --guess-name . An-Archive.tar.bz2
```

An-Archive.tar.bz2 را در پوشه ای به نام An-Archive در فهرست فعلی استخراج می‌کند.

```
$ ark --add-to *.jpg pictures.tar.bz2
```

همه فایل‌هایی که به jpg. ختم می‌شوند را به آرشیوی به نام pictures.tar.bz2 اضافه کنید، اگر از قبل وجود نداشته باشد، آن را ایجاد کنید.

دستور unrar

نحوه باز کردن، استخراج و ایجاد فایل های RAR

RARمحبوب ترین ابزار برای ایجاد و استخراج (rar.)فایل های آرشیو فشرده است .هنگامی که یک فایل آرشیو را از وب دانلود می کنیم، برای استخراج آنها به یک ابزار rarنیاز داریم RAR .به صورت آزاد در سیستم عامل های ویندوز برای مدیریت فایل های فشرده در دسترس نیست، اما متاسفانه، ابزار rarدر سیستم های لینوکس از قبل نصب نشده است. نحوه نصب ابزارهای خط فرمان unrarو rarرا با استفاده از فایل رسمی باینری tar در سیستم‌های لینوکس برای باز کردن، استخراج، از حالت فشرده‌سازی یا unrarکردن یک فایل بایگانی توضیح می‌دهد.

ساختار دستور

```
unrar [options] <archive_name> [destination]
```

گزینه‌های دستور:

- x استخراج فایل‌ها از آرشیو.
- e مشابه x است.
- t تست صحت آرشیو بدون استخراج.
- l لیست کردن محتویات آرشیو.
- v نمایش اطلاعات مربوط به آرشیو.
- c استخراج فایل‌ها به پوشه فعلی.
- d استخراج فایل‌ها به پوشه مشخص شده.
- p رمز عبور آرشیو را مشخص می‌کند.
- r استخراج فایل‌های recursively.
- o+ overwrite فایل‌های موجود در مقصد.
- overwrite نکردن فایل‌های موجود در مقصد.
- y پاسخ بله به تمام سوالات.
- n پاسخ نه به تمام سوالات.
- q غیرفعال کردن پیام‌های.

مثال های دستور:

مرحله ۱: نحوه نصب Unrar در لینوکس

در توزیع های مبتنی بر دبیان و اوبونتو ، می توانید به راحتی بسته unrarرا با استفاده از برنامه apt-getیا aptنصب کنید .

```
$ sudo apt-get install unrar
یا
$ sudo apt install unrar
```

اگر از توزیع فدورا استفاده می کنید، می توانید از دستور dnf برای نصب آن استفاده کنید.

```
$ sudp dnf install unrar
```

اگر از توزیع CentOS / RHEL استفاده می کنید، باید آخرین فایل unrar/rar را دانلود کرده و با استفاده از دستورات زیر نصب کنید.

```
# cd / tmp
# wget https://www.rarlab.com/rar/rarlinux-x64-5.6.0.tar.gz
# tar -zxvf rarlinux-x64-5.6.0.tar.gz
# cd rar
# sudo cp -v rar unrar /usr/local/bin/
```

```
----------روی ۳۲ بیت----------
# cd / tmp
# wget https://www.rarlab.com/rar/rarlinux-5.6.0.tar.gz
# tar -zxvf rarlinux-5.6.0.tar.gz
# cd rar
# sudo cp -v rar unrar /usr/local/bin/
```

مرحله ۲: نحوه باز کردن / استخراج یک فایل RAR در لینوکس

برای باز کردن/استخراج یک فایل RARدر پوشه کاری فعلی، کافیست از دستور زیر با گزینه unrar e استفاده کنید.

```
# unrar  e predator-os .rar
```

برای باز کردن/استخراج کردن یک فایل RARدر مسیر یا فهرست مقصد خاص، فقط از گزینه unrar eاستفاده کنید، تمام فایل‌های موجود در فهرست مقصد مشخص شده را استخراج می‌کند.

```
# unrar e predator-os .rar /home
```

برای باز کردن/استخراج یک فایل RARبا ساختار فهرست اصلی .فقط دستور زیر را با گزینه unrar xصادر کنید .مطابق با ساختار پوشه آنها خروجی دستور زیر را ببینید.

```
# unrar x predator-os .rar
```

مرحله ۳: نحوه فهرست کردن یک فایل RAR در لینوکس

برای فهرست کردن یک فایل در داخل یک فایل آرشیو از گزینه unrar lاستفاده کنید .لیستی از فایل ها را با اندازه ، تاریخ ، زمان و مجوزهای آنها نمایش می دهد .

```
unrar l predator-os .rar
```

مرحله ٤: چگونه یک فایل RAR را در لینوکس آزمایش کنیم

برای آزمایش یکپارچگی یک فایل آرشیو، از گزینه unrar tاستفاده کنید .دستور زیر یک بررسی کامل برای هر فایل انجام می دهد و وضعیت فایل را نمایش می دهد.

```
unrar t predator-os .rar
```

unrarدستور به تنها عصاره، لیست و یا آرشیو آزمون فایل استفاده می شود .هیچ گزینه ای برای ایجاد فایل های RARدر لینوکس ندارد .بنابراین، در اینجا باید ابزار خط فرمان RARرا برای ایجاد فایل‌های آرشیو نصب کنیم.

مثال های بیشتر: !

استخراج تمام فایل‌ها از آرشیو:

```
unrar my_archive.rar
```

استخراج فایل‌ها به پوشه مشخص شده:

```
unrar my_archive.rar /home/user/Desktop
```

لیست کردن محتویات آرشیو:

```
unrar l my_archive.rar
```

تست صحت آرشیو:

```
unrar t my_archive.rar
```

استخراج فایل‌ها به صورت رمزنگاری شده:

```
unrar -p my_password my_archive.rar
```

استخراج فایل‌ها به صورت recursively:

```
unrar -r my_archive.rar
```

overwrite فایل‌های موجود در مقصد:

```
unrar -o+ my_archive.rar
```

عدم overwrite فایل‌های موجود در مقصد:

```
unrar -o- my_archive.rar
```

پاسخ بله به تمام سوالات:

```
unrar -y my_archive.rar
```

پاسخ نه به تمام سوالات:

```
unrar -n my_archive.rar
```

غیرفعال کردن پیام‌ها:

```
unrar -q my_archive.rar
```

استخراج فایل war

با استفاده ازunzip

```
unzip -c whatever.war META-INF/MANIFEST.MF
```

خروجی را در ترمینال چاپ می کند.

دستور unzip

unzip همه فایل‌ها را از آرشیو ZIP مشخص شده استخراج می‌کند

ساختاردستور:

```
$ unzip zipFileName [OPTION] [PARAMS]
```

گزینه های ممکن:

- d path/to/directory/ – آرشیو را در فهرست دیگری از حالت فشرده خارج کنید.
- x – آرشیو را استخراج میکند، اما فایل های مشخص شده را استخراج نمیکند.
- j – در صورتی که آرشیو فشرده حاوی ساختار پوشه باشد، بدون ایجاد پوشه‌های جدید، از حالت فشرده خارج میکند.
- l – محتویات یک فایل آرشیو را بدون استخراج آن فهرست می‌کند.
- n – فایل های موجود را بازنویسی نمیکند. به جای آن یک نام فایل جایگزین ارائه میدهد.
- o– بازنویسی فایل ها
- P – یک رمز عبور برای باز کردن فایل بایگانی محافظت شده ارائه می‌دهد.
- q – بدون نوشتن پیام های وضعیت در خروجی استاندارد از حالت فشرده خارج می شود.

- t – بررسی می‌کند که آیا یک فایل آرشیو معتبر است یا خیر.
- v – اطلاعات دقیق (مختلف) را در مورد آرشیو بدون استخراج آن نمایش می‌دهد.

مثال های دستور:

برای استخراج فایل ها به صورت زیر خواهد بود:

```
$ unzip myZipFile.zip
```

برای باز کردن یک فایل ZIP در فهرست متفاوت از فهرست فعلی، فراموش نکنید که پرچم –d را اضافه کنید:

```
$ unzip myZipFile.zip -d /path/to/directory
```

برای از حالت فشرده خارج کردن یک فایل ZIP و حذف فایل یا فایل ها یا فهرست های خاص از استخراج، فراموش نکنید که پرچم –x را اضافه کنید:

```
$ unzip myZipFile.zip -x file1.txt file2.txt
```

دستور backintime

دستوری برای پشتیبان گیری از سیستم

ساختار دستور:

```
$ backintime [ --backup | --backup-job | --snapshots-path | --snapshots-list | --snapshots-list-path | --last-snapshot | --last-snapshot-path | --help | --version | --license ]
```

Back In Time یک ابزار ساده پشتیبان گیری برای لینوکس است. پشتیبان گیری با گرفتن عکس های فوری از یک مجموعه مشخص از پوشه ها انجام می شود. تنها کاری که باید انجام دهید این است که پیکربندی کنید: کجا عکس های فوری ذخیره شود، از چه پوشه هایی نسخه پشتیبان تهیه شود. به عنوان یک ابزار پشتیبان "حالت کاربر" عمل می‌کند. این بدان معناست که شما می‌توانید تنها پوشه‌هایی را که دسترسی نوشتن به آنها دارید پشتیبان‌گیری/بازیابی کنید (در واقع می‌توانید از پوشه‌های فقط خواندنی نسخه پشتیبان تهیه کنید، اما نمی‌توانید آنها را بازیابی کنید). اگر می خواهید آن را به صورت روت اجرا کنید باید از 'su' استفاده کنید. برای پشتیبان‌گیری خودکار از «cron» استفاده کنید، بنابراین نیازی به دیمون نیست، اما «cron» باید در حال اجرا باشد.

گزینه های دستور:

- h– یا help-- نمایش راهنمای دستور
- v– یا version-- نمایش نسخه برنامه
- c– یا config-- تنظیم فایل پیکربندی
- d– یا dry-run-- شبیه سازی عملیات بدون انجام
- f– یا force-- اجبار به اجرای عملیات
- l– یا log-- ذخیره سازی خروجی در فایل log

گزینه های دستوری:

- backup تهیه نسخه پشتیبان
- restore بازیابی نسخه پشتیبان
- list نمایش لیست نسخه های پشتیبان
- delete حذف نسخه پشتیبان
- prune حذف نسخه های پشتیبان قدیمی
- init ایجاد تنظیمات اولیه

مثال های دستور:

انجام یک پشتیبان‌گیری:

```
$backintime --backup
```

این دستور یک پشتیبان‌گیری اجرا می‌کند و فایل‌ها و پوشه‌های تغییر یافته را به پوشه‌ی پشتیبان‌ها اضافه می‌کند.

اجرای یک کار پشتیبان‌گیری خاص:

```
$backintime --backup-job MyBackupJob
```

در این مثال، دستور backintime برای اجرای کار پشتیبان‌گیری با نام MyBackupJob استفاده می‌شود. این کار پیکربندی شده و قبلاً توسط شما تعریف شده است.

نمایش مسیر پوشه‌ی اسنپ‌شات‌ها:

```
$backintime --snapshots-path
```

در این مثال، دستور backintime برای نمایش مسیر پوشه‌ی اسنپ‌شات‌ها استفاده می‌شود. این مسیر محل ذخیره‌سازی اسنپ‌شات‌های ایجاد شده توسط backintime است.

لیست کردن اسنپ‌شات‌ها:

```
$backintime --snapshots-list
```

در این مثال، دستور backintime برای لیست کردن اسنپ‌شات‌های موجود استفاده می‌شود. این دستور لیستی از اسنپ‌شات‌های موجود را به همراه جزئیات آنها نمایش می‌دهد.

نمایش مسیر لیست اسنپ‌شات‌ها:

```
$backintime --snapshots-list-path
```

در این مثال، دستور backintime برای نمایش مسیر فایل لیست اسنپ‌شات‌ها استفاده می‌شود. این فایل حاوی لیست اسنپ‌شات‌های ایجاد شده توسط backintime است.

بازیابی از آخرین اسنپ‌شات:

```
$backintime --last-snapshot
```

در این مثال، دستور backintime برای بازیابی از آخرین اسنپ‌شات استفاده می‌شود. این دستور فایل‌ها و پوشه‌های لازم را از آخرین اسنپ‌شات ایجاد شده توسط backintime بازیابی می‌کند.

دستور djpeg

دستور djpeg در لینوکس برای رمزگشایی تصاویر با فرمت JPEG و نمایش آن‌ها استفاده می‌شود. این دستور از کتابخانه libjpeg تحت عنوان (IJG) Independent JPEG Group استفاده می‌کند و قابلیت پردازش تصاویر با فرمت JPEG را فراهم می‌کند.

گزینه‌های دستور:

- dct int: انتخاب نوع تبدیل DCT (Discrete Cosine Transform).
- ppm: خروجی را به فرمت PPM تبدیل کرده و نمایش می‌دهد.
- bmp: خروجی را به فرمت BMP تبدیل کرده و نمایش می‌دهد.
- gif: خروجی را به فرمت GIF تبدیل کرده و نمایش می‌دهد.
- targa: خروجی را به فرمت Targa تبدیل کرده و نمایش می‌دهد.
- rgb: خروجی را به فرمت RGB تبدیل کرده و نمایش می‌دهد.
- raw: خروجی را به فرمت raw تبدیل کرده و نمایش می‌دهد.
- grayscale: نمایش تصویر به صورت تک رنگ (سیاه و سفید).

مثال های دستور:

نمایش تصویر با فرمت JPEG به صورت PPM:

```
djpeg -ppm image.jpg > output.ppm
```

نمایش تصویر در فرمت BMP:

```
djpeg -bmp image.jpg > output.bmp
```

تبدیل تصویر JPEG به GIF:

```
djpeg -gif image.jpg > output.gif
```

نمایش تصویر به صورت تک رنگ (سیاه و سفید):

```
djpeg -grayscale image.jpg
```

دستور xz

شرح دستور

دستور xzیک ابزار خط فرمان است که برای فشرده‌سازی و استخراج فایل‌ها با استفاده از الگوریتم فشرده‌سازی XZ استفاده می‌شود . این الگوریتم به دلیل نسبت فشرده‌سازی بسیار بالا و سرعت مناسب، یکی از محبوب‌ترین روش‌های فشرده‌سازی در سیستم‌های لینوکس است.

ساختار دستور

```
xz [options] [file ...]
```

- **options:**گزینه‌هایی هستند که رفتار دستور را تغییر می‌دهند، مانند نوع فشرده‌سازی، سطح فشرده‌سازی و...
- **file:**نام فایل یا فایل‌هایی است که می‌خواهید فشرده یا استخراج کنید.

کاربردهای دستور

- **فشرده‌سازی فایل‌ها :**با استفاده از دستور xzمی‌توانید فایل‌های خود را با فرمت xz . فشرده کنید تا حجم آن‌ها کاهش یابد.
- **استخراج فایل‌های فشرده :**فایل‌های فشرده شده با فرمت xz . را می‌توانید با استفاده از دستور xzاستخراج کنید.
- **ایجاد آرشیو :**با ترکیب دستور xzبا دستور tarمی‌توانید آرشیوهای فشرده ایجاد کنید.

گزینه‌های مهم دستور

- **-d:**استخراج فایل فشرده شده
- **-z:**فشرده‌سازی فایل
- **-v:**نمایش اطلاعات جزئی در حین عملیات
- **-f:**اجبار در عملیات (مثلاً بازنویسی فایل خروجی)
- **-k:**حفظ فایل اصلی پس از فشرده‌سازی
- **-c:**ایجاد خروجی به صورت جریان(stream)
- **-T:**نمایش نسخهxz
- **-l:**نمایش اطلاعات در مورد فایل فشرده شده (مانند اندازه فشرده شده و اصلی)
- **-b:**تنظیم اندازه بلوک برای فشرده‌سازی
- **-m:**تنظیم حافظه استفاده شده برای فشرده‌سازی

مثال های کاربردی دستور xz با گزینه‌های مختلف

دستور xzیک ابزار بسیار قدرتمند برای فشرده‌سازی و استخراج فایل‌ها است .با استفاده از گزینه‌های مختلف این دستور، می‌توانید عملیات فشرده‌سازی را دقیقاً مطابق نیاز خود تنظیم کنید .در ادامه، چندین مثال کاربردی برای گزینه‌های مختلف این دستور آورده شده است:

فشرده‌سازی فایل ها

- **فشرده‌سازی ساده یک فایل:**

```
xz file.txt
```

این دستور فایل file.txtرا با فرمت xz فشرده کرده و فایل خروجی file.txt.xzرا ایجاد می‌کند.

- **فشرده‌سازی با نمایش اطلاعات جزئی:**

```
xz -v file.txt
```

در این مثال، علاوه بر فشرده‌سازی فایل، اطلاعاتی مانند اندازه فایل قبل و بعد از فشرده‌سازی و زمان صرف شده برای عملیات نمایش داده می‌شود.

- **فشرده‌سازی با حفظ فایل اصلی:**

```
xz -k file.txt
```

این دستور فایل اصلی file.txtرا پس از فشرده‌سازی حفظ می‌کند.

- **فشرده‌سازی با تنظیم اندازه بلوک:**

```
xz -b 1M file.txt
```

این دستور اندازه بلوک برای فشرده‌سازی را به ۱ مگابایت تنظیم می‌کند .این گزینه می‌تواند روی سرعت و میزان فشرده‌سازی تأثیر بگذارد.

استخراج فایل های فشرده

- **استخراج فایل فشرده شده:**

```
xz -d file.txt.xz
```

این دستور فایل فشرده شده file.txt.xzرا استخراج کرده و فایل اصلی file.txtرا ایجاد می‌کند.

- **استخراج و بازنویسی فایل اصلی:**

```
xz -df file.txt.xz
```

این دستور فایل فشرده شده را استخراج کرده و فایل اصلی را بازنویسی می‌کند .حتی اگر فایلی با همین نام در دایرکتوری وجود داشته باشد.

ایجاد آرشیو فشرده

- **ایجاد آرشیو:tar.xz**

```
tar czvf archive.tar.xz directory/*
```

این دستور یک آرشیو tar.xz به نام archive.tar.xz ایجاد می‌کند که شامل تمام فایل‌های موجود در دایرکتوری directoryاست.

نمایش اطلاعات

- **نمایش نسخه:xz**

```
xz -T
```

این دستور نسخه نصب شده xz را نمایش می‌دهد.

- **نمایش اطلاعات در مورد فایل فشرده شده:**

```
xz -l file.txt.xz
```

این دستور اطلاعاتی مانند اندازه فشرده شده و اصلی، تاریخ ایجاد و ... را در مورد فایل `file.txt.xz` نمایش می‌دهد.

مثال‌های پیشرفته‌تر

- **فشرده‌سازی چندین فایل به صورت همزمان:**

```
xz file1.txt file2.txt file3.txt
```

- **فشرده‌سازی همه فایل‌های یک دایرکتوری:**

```
xz *
```

- **فشرده‌سازی با بالاترین سطح فشرده‌سازی:**

```
xz -9 file.txt
```

توجه داشته باشید که سطح فشرده‌سازی از ۰ تا ۹ متغیر است و هر چه این عدد بزرگتر باشد، زمان فشرده‌سازی بیشتر و میزان فشرده‌سازی بالاتر خواهد بود.

فصل بیستم‌ویکم

فصل بیست و یکم

برنامه نویسی

دستور make

دستور makeیک ابزار متداول در سیستم‌عامل‌های لینوکس و UNIX است که برای اتوماسیون فرآیند ساخت (build process) و تولید برنامه‌ها استفاده می‌شود. با استفاده از فایل Makefileکه شامل قوانین و دستورات ساخت است، makeفرآیند ساخت را اجرا و برنامه‌ها را به صورت اتوماتیک تولید می‌کند.

برای استفاده از make، عموماً مراحل زیر را دنبال می‌کنید:

ایجاد فایل : Makefileاین فایل حاوی قوانین ساخت و دستورات مربوط به هر قانون است. فایل Makefileباید در فهرست پروژه شما قرار داشته باشد.

تعریف قوانین ساخت: در فایل Makefile، شما می‌توانید قوانینی را تعریف کنید که مشخص می‌کنند چگونه برنامه را ساخت، کد منبع را کامپایل و لینک کنند. هر قانون شامل یک هدف(target) ، وابستگی‌های (dependencies) و دستورات ساخت (build commands)است.

اجرای :makeپس از تعریف قوانین ساخت در فایل Makefile، می‌توانید دستور makeرا اجرا کنید make .بر اساس وابستگی‌ها و قوانین ساخت، فرآیند ساخت را اجرا کرده و برنامه‌های مورد نظر را تولید می‌کند.

مثال: فرض کنید یک پروژه C++ دارید که شامل فایل‌های سورس main.cppو helper.cppو هدف نهایی آن برنامه اجرایی myprogramاست. یک مثال ساده از فایل Makefileبرای این پروژه می‌تواند به صورت زیر باشد:

```
makefile
myprogram: main.o helper.o
   g++ main.o helper.o -o myprogram

main.o: main.cpp
   g++ -c main.cpp

helper.o: helper.cpp
   g++ -c helper.cpp

clean:
   rm *.o myprogram
```

در این مثال، هدف myprogramوابستگی به فایل‌های main.oو helper.oدارد و دستور ساخت آن شامل لینک کردن فایل‌های o.است. همچنین، قوانین main.oو helper.oنیز وابستگی به فایل‌های cpp.مربوطه دارند و دستور ساخت آن‌ها شامل کامپایل کردن فایل‌های cpp.به فایل‌های o.است. با اجرای دستور makeدر فهرست پروژه، makeفرآیند ساخت را شروع می‌کند و برنامه اجرایی myprogramرا تولید می‌کند. وظایف و عملکردهای خاص خود را دارند. در زیر، برای هر یک از گزینه‌ها، یک مثال از دستور makeرا ارائه می‌دهم:

-b, -m : اجبار makeبه ساخت مجدد targetsمشخص شده.

```
$make -b myprogram
```

-B, --always-make : همیشه ساخت مجدد targetsبه همراه عدم بررسی تاریخچه.

```
$make -B myprogram
```

-C dir, --directory=dir : تنظیم فهرست کاری برای اجرای makeتنظیم .

```
$make -C /path/to/project
```

-d, --debug[=FLAGS] : نمایش اطلاعات جزئی در حین اجرای make .

```
$make -d myprogram
```

-e, --environment-overrides : اجازه تغییر متغیرهای محیطی توسط متغیرهای make .

```
$make -e myprogram
```

-f file, --file=file, --makefile=FILE : استفاده از فایل Makefile مشخص شده به جای فایل پیشفرض.

```
$make -f custom_makefile myprogram
```

-i, --ignore-errors : ادامه ساخت حتی در صورت بروز خطا.

```
$make -i myprogram
```

-I dir, --include-dir=dir : اضافه کردن فهرست به مسیرهای جستجوی make .

```
$make -I /path/to/include_dir myprogram
```

-j [jobs], --jobs[=jobs] : اجرای همزمان برای make با تعداد جابه‌جایی مشخص شده (معمولاً تعداد هسته‌های پردازنده).

```
$make -j4 myprogram
```

-k, --keep-going : ادامه ساخت حتی در صورت بروز خطا، اما ساخت موارد دیگر را ادامه می‌دهد.

```
$make -k myprogram
```

-l [load], --load-average[=load] : محدود کردن تعداد فرآیندهای ساخت بر اساس میانگین بار سیستم.

```
$make -l2 myprogram
```

-L, --check-symlink-times : بررسی بروزرسانی زمانهای symlink ها در هنگام بررسی وابستگی‌ها.

```
$make -L myprogram
```

-n, --just-print, --dry-run, --recon : نمایش دستورات make بدون اجرا (حالت آزمایشی).

```
$make -n myprogram
```

-o file, --old-file=file, --assume-old=file : تعیین فایل‌های قدیمی (بدون وابستگی) برای اجرای دستور ساخت.

```
$make -o old_file myprogram
```

-p, --print-data-base : نمایش پایگاه داده make برای قوانین ساخت.

```
$make -p
```

-q, --question : بررسی کردن وضعیت هدف برای تشخیص ساخت مجدد.

```
$make -q myprogram
```

-r, --no-builtin-rules : غیرفعال‌سازی قوانین ساخت پیش‌فرض make .

```
$make -r myprogram
```

-R, --no-builtin-variables : غیرفعال‌سازی متغیرهای پیش‌فرض make .

```
$make -R myprogram
```

-s, --silent, --quiet : حالت سکوت در اجرای make .

```
$make -s myprogram
```

-S, --no-keep-going, --stop : متوقف کردن ساخت در صورت بروز خطا.

```
$make -S myprogram
```

-t, --touch : لمس کردن فایل‌های هدف بدون اجرای دستور ساخت.

```
$make -t myprogram
```

-v, --version : نمایش نسخه make .

```
$make -v
```

: -w, --print-directory نمایش فهرست کاری در حین اجرای make .

```
$make -w myprogram
```

: --no-print-directory عدم نمایش فهرست کاری در حین اجرای make .

```
$make --no-print-directory myprogram
```

: -W file, --what-if=file, --new-file=file, --assume-new=file تعیین فایل جدید برای ساخت مجدد.

```
$make -W new_file myprogram
```

: --warn-undefined-variables اخطار در صورت استفاده از متغیرهای تعریف نشده.

```
$make --warn-undefined-variables myprogram
```

کامپایل کرنل لینوکس

دستورات `make` که در اینجا ذکر شده‌اند، برای ساخت و پیکربندی کرنل لینوکس استفاده می‌شوند .هر یک از این دستورات یک هدف مشخص را در فرآیند کامپایل دنبال می‌کنند.

```
make target
```

- **make:** خود دستور است که مسئول اجرای دستورالعمل‌های موجود در فایل `Makefile` است.
- **target:** هدفی که می‌خواهیم اجرا شود .این اهداف همان دستوراتی هستند که در پرسش شما آمده‌اند.

توضیح هر یک از اهداف

اهداف پیکربندی

- **all:** تمام مراحل کامپایل از جمله ساخت کرنل، ماژول‌ها و سایر اجزا را انجام می‌دهد.
- **config:** یک فایل پیکربندی پیش‌فرض (.config) ایجاد می‌کند.
- **xconfig:** یک رابط گرافیکی برای پیکربندی کرنل باز می‌کند) معمولا.(Xconfigurator
- **menuconfig:** یک رابط متنی منو محور برای پیکربندی کرنل باز می‌کند.
- **gconfig:** (در صورت موجود بودن) یک رابط گرافیکی مبتنی بر GTK برای پیکربندی کرنل باز می‌کند.
- **oldconfig:** از یک فایل پیکربندی موجود (.config) به عنوان پایه استفاده می‌کند و تغییرات جدید را درخواست می‌کند.
- **mrproper:** تمام فایل‌های تولید شده و فایل‌های شیء را حذف می‌کند و به حالت اولیه باز می‌گردد.

اهداف ساخت کرنل و ماژول‌ها

- **zImage:** یک تصویر فشرده از کرنل را برای سیستم‌هایی با حافظه کمتر از ۴ مگابایت ایجاد می‌کند.
- **bzImage:** یک تصویر فشرده از کرنل را برای سیستم‌هایی با حافظه بیشتر از ۴ مگابایت ایجاد می‌کند) از فشرده‌سازی bzip2 استفاده می‌کند.(
- **modules:** تمام ماژول‌های کرنل را بر اساس پیکربندی فعلی می‌سازد.
- **modules_install:** ماژول‌های ساخته شده را در دایرکتوری مناسب نصب می‌کند.

اهداف ساخت بسته

- **rpm-pkg:** یک بسته RPM حاوی کرنل و ماژول‌ها ایجاد می‌کند.
- **binrpm-pkg:** یک بسته RPM باینری (بدون کد منبع) ایجاد می‌کند.
- **deb-pkg:** یک بسته Debian حاوی کرنل و ماژول‌ها ایجاد می‌کند.

مثال‌های کاربردی دستورات `make` در کامپایل کرنل لینوکس

دستورات `make` ابزار قدرتمندی برای ساخت و مدیریت پروژه‌های نرم‌افزاری، به ویژه در محیط لینوکس هستند .در زمینه کامپایل کرنل لینوکس، این دستورات نقش بسیار مهمی ایفا می‌کنند .در ادامه، مثال‌هایی برای هر یک از اهداف (targets) ذکر شده ارائه می‌شود:

اهداف پیکربندی(Configuration Targets)

- **make config:**
  - یک فایل پیکربندی پیش‌فرض به نام `.config` ایجاد می‌کند .این فایل حاوی گزینه‌های مختلفی است که رفتار کرنل را تعیین می‌کنند.
  - **مثال:**

```
make config
```

- **make menuconfig:**
  - یک رابط متنی منو محور را باز می‌کند که به شما اجازه می‌دهد تا گزینه‌های پیکربندی را به صورت تعاملی انتخاب کنید.
  - **مثال:**

```
make menuconfig
```

- **make xconfig:**
  - (در صورت موجود بودن) یک رابط گرافیکی برای پیکربندی کرنل باز می‌کند.
  - **مثال:**

```
make xconfig
```

- **make oldconfig:**
  - از یک فایل پیکربندی موجود (.config) به عنوان پایه استفاده می‌کند و فقط گزینه‌های جدید یا تغییر یافته را نمایش می‌دهد.
  - **مثال:**

```
make oldconfig
```

اهداف ساخت کرنل و ماژول‌ها(Kernel and Module Targets)

- **make zImage:**
  - یک تصویر فشرده از کرنل را برای سیستم‌هایی با حافظه کمتر از ۴ مگابایت ایجاد می‌کند.
  - **مثال:**

```
make zImage
```

```
make bzImage:
```

  - یک تصویر فشرده از کرنل را برای سیستم‌هایی با حافظه بیشتر از ۴ مگابایت ایجاد می‌کند.
  - **مثال:**

```
make bzImage
```

- **make modules:**
  - تمام ماژول‌های کرنل را بر اساس پیکربندی فعلی می‌سازد.
  - **مثال:**

```
make modules
```

- **make modules_install:**
  - ماژول‌های ساخته شده را در دایرکتوری مناسب نصب می‌کند.
  - **مثال:**

```
make modules_install
```

اهداف ساخت بسته(Package Targets)

- **make rpm-pkg:**

- یک بسته RPM حاوی کرنل و ماژول‌ها ایجاد می‌کند.
- **مثال:**

```
make rpm-pkg
```

- **`make deb-pkg`:**
  - یک بسته Debian حاوی کرنل و ماژول‌ها ایجاد می‌کند.
  - **مثال:**

```
make deb-pkg
```

هدف تمیزکاری(Cleaning Target)

- **`make mrproper`:**
  - تمام فایل‌های تولید شده و فایل‌های شیء را حذف می‌کند و به حالت اولیه باز می‌گردد.
  - **مثال:**

```
make mrproper
```

مثال کامل برای ساخت یک کرنل سفارشی و نصب آن

#پیکربندی کرنل

```
make menuconfig
```

#ساخت تصویر کرنل و ماژول‌ها

```
make zImage modules
```

#نصب ماژول‌ها

```
make modules_install
```

#ایجاد بستهRPM

```
make rpm-pkg
```

نکته :ترتیب اجرای دستورات بسیار مهم است .ابتدا باید کرنل را پیکربندی کنید، سپس آن را بسازید و در نهایت ماژول‌ها را نصب کنید .همچنین، بسته به توزیع لینوکس و نسخه کرنل، ممکن است برخی از گزینه‌ها و دستورات کمی متفاوت باشند.

دستور cc

دستور cc در سیستم‌عامل‌های Unix/Linux برای کامپایل و اجرای برنامه‌های نوشته شده به زبان C استفاده می‌شود. همچنین cc یک کامپایلر است که برای کامپایل کد C به زبان ماشین استفاده می‌شود. این دستور جزو ابزارهای GNU Compiler Collection (GCC) است که مجموعه‌ای از کامپایلرها برای زبان‌های برنامه‌نویسی مختلف را شامل می‌شود.

ساختار دستور

```
cc [options] file(s) [-o output]
```

گزینه های ساختار دستور:

- options گزینه‌هایی برای کنترل نحوه کامپایل کد
- file(s) نام فایل‌های کد C که می‌خواهید کامپایل شوند
- o output– نام فایل خروجی که کد کامپایل شده در آن ذخیره می‌شود

گزینه‌های دستور:

- c– فقط کد را کامپایل می‌کند و آن را به زبان ماشین تبدیل می‌کند، اما آن را اجرا نمی‌کند.
- o– نام فایل خروجی را مشخص می‌کند.
- g– اطلاعات اشکال‌زدایی را در کد کامپایل شده ذخیره می‌کند.
- Wall– تمام هشدارهای مربوط به کد را نمایش می‌دهد.
- Werror– تمام هشدارها را به عنوان خطا در نظر می‌گیرد.
- std=c11– از استاندارد C11 برای کامپایل کد استفاده می‌کند.
- pedantic– کامپایلر را در مورد قوانین زبان C سختگیرتر می‌کند.

مثال های دستور:

بدون گزینه: کامپایل و ساخت فایل اجرایی از یک فایل برنامه به زبان C.

```
cc program.c
```

خروجی دستور: ساخت فایل اجرایی از برنامه program.c

o <output_file>, --output <output_file>: – تعیین نام فایل اجرایی خروجی.

```
cc program.c -o program
```

خروجی دستور: ساخت فایل اجرایی با نام program

c: – تنها انجام مرحله کامپایل و تولید فایل اشیاء بدون اجرای لینکر.

```
cc -c program.c
```

خروجی دستور: تولید فایل اشیاء بدون اجرای لینکر برای program.c

g: – اضافه کردن اطلاعات اجرایی debug به فایل اجرایی.

```
cc -g program.c -o program_debug
```

خروجی دستور: ساخت فایل اجرایی با اطلاعات debug برای برنامه program.c

Wall: نمایش تمام اخطارها و هشدارهای کامپایلر.

```
cc -Wall program.c -o program
```

خروجی دستور: نمایش تمام اخطارها و هشدارهای کامپایلر برای program.c

I <include_path>, --include <include_path>: اضافه کردن مسیرهای include برای جستجوی فایل‌های هدر.

```
cc -I /usr/local/include program.c -o program
```

خروجی دستور: جستجوی فایل‌های هدر در /usr/local/include برای program.c

با استفاده از این مثال‌ها، می‌توانید با گزینه‌های مختلف دستور cc، برنامه‌های نوشته شده به زبان C را کامپایل و اجرا کنید.

دستور autoconf

دستور autoconf در سیستم‌های لینوکس برای تولید اسکریپت‌های پیکربندی (configure scripts) از فایل‌های configure.ac یا configure.in به منظور تهیه و تنظیم پیکربندی برنامه‌ها و پروژه‌های نرم‌افزاری استفاده می‌شود. این دستور به توسعه‌دهندگان کمک می‌کند تا به صورت خودکار فایل‌های پیکربندی برنامه‌ها را ایجاد کنند.

ساختار دستور:

```
autoconf [OPTIONS] [SCRIPT]
```

گزینه‌های دستور :

- V- نمایش نسخه autoconf
- h- نمایش راهنمای autoconf
- o FILE- نام فایل خروجی را مشخص می‌کند
- I DIR- به دنبال فایل‌های configure.ac در DIR می‌گردد
- include FILE- فایل FILE را به عنوان ورودی اضافی در نظر می‌گیرد
- force- حتی اگر فایل‌های configure.ac و configure قدیمی‌تر از فایل‌های منبع نباشند، آنها را دوباره اجرا می‌کند
- clean- فایل‌های موقت ایجاد شده توسط autoconf را حذف می‌کند

بدون گزینه: اجرای دستور autoconf بدون هیچ گزینه اضافی.

```
autoconf
```

خروجی دستور

- h, --help: نمایش راهنمای استفاده از دستور autoconf.

```
autoconf -h
```

خروجی دستور: راهنمای استفاده از دستور autoconf

- o <output_file>: مشخص کردن فایل خروجی برای ذخیره اسکریپت پیکربندی.

```
autoconf -o configure_script
```

خروجی دستور: ایجاد فایل configure_script

- f, --force: اجبار اجرای دستور autoconf حتی اگر فایل configure.ac تغییر نکرده باشد.

```
autoconf -f
```

خروجی دستور: اجرای autoconf با اعمال اجباری

- i, --include: اضافه کردن فهرست‌های جستجو برای فایل‌های m4.

```
autoconf -i m4_include_directory
```

خروجی دستور: اضافه کردن m4_include_directory به فهرست‌های جستجو

- v, --version: نمایش نسخه فعلی autoconf.

```
autoconf -v
```

خروجی دستور: نمایش نسخه autoconf

- W <warning>, --warnings=<warning>: فعال کردن یک هشدار خاص.

```
autoconf -W all
```

خروجی دستور: فعالسازی تمامی هشدارها

این مثال‌ها به شما کمک می‌کنند تا با استفاده از دستور autoconf و گزینه‌های مختلف آن، بتوانید اسکریپت‌های پیکربندی برنامه‌ها را به صورت خودکار ایجاد و تنظیم کنید.

دستور automake

automakeابزاری است که برای تولید خودکار فایلهای **Makefile.in** سازگار با مجموعه استانداردهای کدگذاری GNU استفاده می شود **autoconf** .برای استفاده از خودکار لازم است .کتابچه راهنمای خودکار را می توان بصورت آنلاین خواند یا در

قالب PDF بارگیری کرد .قالب های بیشتری نیز برای بارگیری یا خواندن آنلاین ارائه شده است .این *Makefile.in* را برای پیکربندی از پرونده ***Makefile.am*** تولید می کند.

ساختار دستور:

```
automake [OPTION] .... [Makefile]
```

گزینه های دستور:

گزینه های دستور:

- i-یاignore-deps– :کد پیگیری وابستگی را غیرفعال می کند.
- include-deps– :کد رهگیری وابستگی را فعال می کند.
- foreign– :سختگیری را برای خارجی تعیین می کند.
- gnits– :سختگیری را بر روی gnits تنظیم می کند.
- gnu– :سختگیری را برای gnu تنظیم می کند.

پرونده های کتابخانه:

- **a or –add-lost-** :پرونده های استاندارد گمشده را به بسته اضافه می کند.
- **–libdir** = DIRECTORY : فهرست ذخیره کننده پرونده های کتابخانه را تنظیم می کند.
- **c-یاcopy–** :با a- ، پرونده های از دست رفته را کپی می کند.
- **f or –force-lost-** :بروزرسانی پرونده های استاندارد را مجبور – کند.

مثال های دستور:

help– : راهنما نشان میدهد و سپس خارج می شود.

```
nishanth@nishanth-VirtualBox:~/nis$ automake --help
Usage: /usr/bin/automake [OPTION]... [Makefile]...

Generate Makefile.in for configure from Makefile.am.

Operation modes:
      --help               print this help, then exit
      --version            print version number, then exit
  -v, --verbose            verbosely list files processed
      --no-force           only update Makefile.in's that are out of date
  -W, --warnings=CATEGORY  report the warnings falling in CATEGORY

Dependency tracking:
  -i, --ignore-deps      disable dependency tracking code
      --include-deps     enable dependency tracking code

Flavors:
      --foreign          set strictness to foreign
      --gnits            set strictness to gnits
      --gnu              set strictness to gnu
```

version– :اطلاعات نسخه را چاپ کرده و سپس خارج می شود.

```
nishanth@nishanth-VirtualBox:~/nis$ automake --version
automake (GNU automake) 1.15.1
Copyright (C) 2017 Free Software Foundation, Inc.
License GPLv2+: GNU GPL version 2 or later <http://gnu.org/licenses/gpl-2.0.htm
l>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by Tom Tromey <tromey@redhat.com>
       and Alexandre Duret-Lutz <adl@gnu.org>.
nishanth@nishanth-VirtualBox:~/nis$
```

v-یا verbose– : لیستی از فرایندها را چاپ می کند.

```
nishanth@nishanth-VirtualBox:~/nis$ automake --verbose
automake: thread 0: reading autoconf --trace=AM_INIT_AUTOMAKE:\$f:\$l::\$d::\$n
::\${::}% --trace=AC_CANONICAL_BUILD:\$f:\$l::\$d::\$n::\${::}% --trace=AC_INIT
:\$f:\$l::\$d::\$n::\${::}% --trace=AC_CONFIG_LIBOBJ_DIR:\$f:\$l::\$d::\$n::\${
::}% --trace=AC_CANONICAL_TARGET:\$f:\$l::\$d::\$n::\${::}% --trace=m4_include:
\$f:\$l::\$d::\$n::\${::}% --trace=AC_REQUIRE_AUX_FILE:\$f:\$l::\$d::\$n::\${::
}% --trace=_AM_COND_ELSE:\$f:\$l::\$d::\$n::\${::}% --trace=AM_MAINTAINER_MODE:
\$f:\$l::\$d::\$n::\${::}% --trace=AM_PROG_MKDIR_P:\$f:\$l::\$d::\$n::\${::}% -
-trace=AC_LIBSOURCE:\$f:\$l::\$d::\$n::\${::}% --trace=AC_CONFIG_AUX_DIR:\$f:\$
l::\$d::\$n::\${::}% --trace=AC_CONFIG_FILES:\$f:\$l::\$d::\$n::\${::}% --trace
=m4_sinclude:\$f:\$l::\$d::\$n::\${::}% --trace=LT_SUPPORTED_TAG:\$f:\$l::\$d::
\$n::\${::}% --trace=_AM_COND_IF:\$f:\$l::\$d::\$n::\${::}% --trace=AM_AUTOMAKE
_VERSION:\$f:\$l::\$d::\$n::\${::}% --trace=_LT_AC_TAGCONFIG:\$f:\$l::\$d::\$n:
:\${::}% --trace=AM_GNU_GETTEXT:\$f:\$l::\$d::\$n::\${::}% --trace=AC_CANONICAL
_HOST:\$f:\$l::\$d::\$n::\${::}% --trace=_AM_SUBST_NOTMAKE:\$f:\$l::\$d::\$n::\
${::}% --trace=AM_PROG_AR:\$f:\$l::\$d::\$n::\${::}% --trace=sinclude:\$f:\$l::
\$d::\$n::\${::}% --trace=AC_CONFIG_LINKS:\$f:\$l::\$d::\$n::\${::}% --trace=AM
_CONDITIONAL:\$f:\$l::\$d::\$n::\${::}% --trace=AC_CONFIG_HEADERS:\$f:\$l::\$d:
:\$n::\${::}% --trace=AC_SUBST_TRACE:\$f:\$l::\$d::\$n::\${::}% --trace=AM_EXTR
A_RECURSIVE_TARGETS:\$f:\$l::\$d::\$n::\${::}% --trace=AM_GNU_GETTEXT_INTL_SUBD
IR:\$f:\$l::\$d::\$n::\${::}% --trace=_AM_COND_ENDIF:\$f:\$l::\$d::\$n::\${::}%
 --trace=AC_FC_SRCEXT:\$f:\$l::\$d::\$n::\${::}%
automake: thread 0: reading /usr/share/automake-1.15/am/header-vars.am
automake: thread 0: reading makefile.am
```

no-force– : این فقط *Makefile.in* را که قدیمی است به روز می کند.

W-یا warnings = Category– : اخطارهای مربوط به آن دسته خاص را گزارش می کند.

```
nishanth@nishanth-VirtualBox:~/nis$ automake --warnings=obsolete
nishanth@nishanth-VirtualBox:~/nis$
```

دسته اخطارهای مختلف عبارتند از:

```
Warning categories include:
  gnu                GNU coding standards (default in gnu and gnits modes)
  obsolete           obsolete features or constructions
  override           user redefinitions of Automake rules or variables
  portability        portability issues (default in gnu and gnits modes)
  extra-portability  extra portability issues related to obscure tools
  syntax             dubious syntactic constructs (default)
  unsupported        unsupported or incomplete features (default)
  all                all the warnings
  no-CATEGORY        turn off warnings in CATEGORY
  none               turn off all the warnings
  error              treat warnings as errors
```

دستور cc

دستور **cc** مخفف **C Compiler** است ، معمولاً یک دستور مستعار به **gcc** یا **clang** است . همانطور که از نامش پیداست ، با اجرای دستور cc معمولاً **gcc** در سیستم های لینوکس فراخوانی می شود . برای کامپایل کدهای زبان C و ایجاد موارد اجرایی مورد استفاده قرار می گیرد .تعداد گزینه های موجود برای دستور cc بسیار زیاد است .مهمترین گزینه ها در زیر با مثال توضیح داده شده است.

ساختار دستور:

```
cc [options] files
```

به عنوان مثال :در زیر دستور کامپایل *source_file.c* فایل، و ایجاد به طور پیش فرض فایل خروجی اجرایی، *فایلهای دو دویی a.out* .

```
cc example.c
```

```
mukkesh@mex:~/Desktop$ cc example.c
mukkesh@mex:~/Desktop$ ls
a.out  example.c  Stuff
mukkesh@mex:~/Desktop$
```

گزینه های دستور:

دستور cc با گزینه :o- این دستور فایل *source_file.c* را کامپایل می کند و یک فایل خروجی اجرایی با نام مشخص ایجاد می کند.

```
cc example.c -o examp_out
```

```
mukkesh@mex:~/Desktop$ cc example.c -o examp_out
mukkesh@mex:~/Desktop$ ls
a.out  example.c  examp_out  Stuff
mukkesh@mex:~/Desktop$
```

- **دستور cc با گزینه :Wall-** این دستور فایل *source_file.c* را کامپایل می کند و تمام خطاها و هشدارهای موجود در برنامه را بررسی می کند.

```
cc example.c -Wall -o examp_out
```

```
mukkesh@mex:~/Desktop$ cc example.c -Wall -o examp_out
mukkesh@mex:~/Desktop$ ls
a.out  example.c  examp_out  Stuff
mukkesh@mex:~/Desktop$
```

- **دستور cc با گزینه :w-** این دستور فایل *source_file.c* را کامپایل می کند ، اما همه هشدارها را سرکوب می کند.

```
cc example.c -w
```

```
mukkesh@mex:~/Desktop$ cc example.c -w
mukkesh@mex:~/Desktop$ ls
a.out  example.c  example.o  examp_out  examp_out_debug  Stuff
mukkesh@mex:~/Desktop$
```

دستور cc با گزینه :g- این دستور فایل *source_file.c* را کامپایل می کند و یک نسخه اشکال زدایی از فایل خروجی اجرایی ایجاد می کند .این فایل خروجی می تواند توسط اشکال ساز مورد استفاده قرار گیرد و اندازه آن معمولاً بسیار بزرگتر از پرونده های خروجی عادی است.

```
cc example.c -g -o examp_out_debug
```

```
mukkesh@mex:~/Desktop$ cc example.c -g -o examp_out_debug
mukkesh@mex:~/Desktop$ ls
a.out  example.c  examp_out  examp_out_debug  Stuff
mukkesh@mex:~/Desktop$
```

دستور cc با گزینه :c- این دستور فایل *source_file.c* را کامپایل می کند و یک پرونده شی را ایجاد می کند *source_file.o* ، که بعداً می تواند برای ایجاد یک فایل خروجی اجرایی پیوند داده شود.

cc example.c -c

```
mukkesh@mex:~/Desktop$ cc example.c -c
mukkesh@mex:~/Desktop$ ls
a.out       example.o  examp_out_debug
example.c   examp_out  Stuff
mukkesh@mex:~/Desktop$
```

- **دستور cc با گزینه : *dir* L-** این دستور ، در حالی که فایل *source_file.c* را کامپایل می کند ، در فهرست مشخص شده برای پرونده های هدر نیز جستجو خواهد کرد.

cc example.c -L / home / ali /Desktop

```
mukkesh@mex:~/Desktop$ cc example.c -L /home/mukkesh/Desktop
mukkesh@mex:~/Desktop$ ls
a.out  example.c  example.o  examp_out  examp_out_debug  Stuff
mukkesh@mex:~/Desktop$
```

- **دستور سی سی با گزینه :ansi-** این دستور کامپایل *source_file.c* فایل، و مطمئن می سازد که کد زیر استانداردهای ANSI سخت است .با این کار مطمئن می شوید که کد قابل حمل است تا در هر سیستم دیگری کامپایل شود.

cc -ansi example.c

```
mukkesh@mex:~/Desktop$ cc example.c -ansi
mukkesh@mex:~/Desktop$ ls
a.out  example.c  example.o  examp_out  examp_out_debug  Stuff
mukkesh@mex:~/Desktop$
```

- **دستور - cc with گزینه های : *dump*** این دستورات مشخص شده در زیر نسخه دستور CC ، دستگاه هدف پیش فرض و رشته های مشخصات داخلی را چاپ می کنند.

cc -dumpversion

```
mukkesh@mex:~/Desktop$ cc -dumpversion
7
mukkesh@mex:~/Desktop$
```

cc -dumpmachine

```
mukkesh@mex:~/Desktop$ cc -dumpmachine
x86_64-linux-gnu
```

cc -dumpspecs

```
mukkesh@mex:~/Desktop$ cc -dumpspecs
*asm:
%{m16|m32:--32}  %{m16|m32|mx32:;:--64}  %{mx32:--x32}  %{!mno-sse2avx:%{mavx:-msse
2avx}} %{msse2avx:%{!mavx:-msse2avx}}

*asm_debug:
%{%:debug-level-gt(0):%{gstabs*:--gstabs}%{!gstabs*:%{g*:--gdwarf2}}} %{fdebug-pref
ix-map=*:--debug-prefix-map %*}

*asm_final:
%{gsplit-dwarf:
       objcopy --extract-dwo     %{c:%{o*:%*}%{!o*:%b%O}}%{!c:%U%O}     %{c:%{o*:%:
replace-extension(%{o*:%*} .dwo)}%{!o*:%b.dwo}}%{!c:%b.dwo}
       objcopy --strip-dwo       %{c:%{o*:%*}%{!o*:%b%O}}%{!c:%U%O}     }

*asm_options:
%{-target-help:%:print-asm-header()} %{v} %{w:-W} %{I*}  %{gz|gz=zlib:--compress-de
bug-sections=zlib} %{gz=none:--compress-debug-sections=none} %{gz=zlib-gnu:--compre
ss-debug-sections=zlib-gnu} %a %Y %{c:%W{o*}%{!o*:-o %w%b%O}}%{!c:-o %d%w%u%O}

*invoke_as:
%{!fwpa*:   %{fcompare-debug=*|fdump-final-insns=*:%:compare-debug-dump-opt()}   %{
!S:-o %|.s |
 as %(asm_options) %m.s %A }  }

*cpp:
%{posix:-D_POSIX_SOURCE} %{pthread:-D_REENTRANT}

*cpp_options:
%(cpp_unique_options) %1 %{m*} %{std*&ansi&trigraphs} %{W*&pedantic*} %{w} %{f*} %{
g*:%{%:debug-level-gt(0):%{g*} %{!fno-working-directory:-fworking-directory}}} %{O*
} %{undef} %{save-temps*:-fpch-preprocess} %(ssp_default)

*cpp_debug_options:
%{d*}

*cpp_unique_options:
%{!Q:-quiet} %{nostdinc*} %{C} %{CC} %{v} %{I*&F*} %{P} %I %{MD:-MD %{!o:%b.d}%{o*:
%.d%*}} %{MMD:-MMD %{!o:%b.d}%{o*:%.d%*}} %{M} %{MM} %{MF*} %{MG} %{MP} %{MQ*} %{MT
*} %{!E:%{!M:%{!MM:%{!MT:%{!MQ:%{MD|MMD:%{o*:-MQ %*}}}}}}} %{remap} %{g3|ggdb3|gsta
bs3|gcoff3|gxcoff3|gvms3:-dD} %{!iplugindir*:%{fplugin*:%:find-plugindir()}} %{H} %
c %{D*&U*&A*} %{i*} %Z %i %{E|M|MM:%W{o*}}
```

- **دستور cc با گزینه -v:** این دستور فایل *source_file.c* را کامپایل می کند و یک خروجی verboseمی دهد.

```
cc example.c -v
```

```
mukkesh@mex:~/Desktop$ cc example.c -v
Using built-in specs.
COLLECT_GCC=cc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/7/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 7.3.0-27ubuntu1~18.0
4' --with-bugurl=file:///usr/share/doc/gcc-7/README.Bugs --enable-languages=c,ada,c
++,go,brig,d,fortran,objc,obj-c++ --prefix=/usr --with-gcc-major-version-only --pro
gram-suffix=-7 --program-prefix=x86_64-linux-gnu- --enable-shared --enable-linker-b
uild-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --l
ibdir=/usr/lib --enable-nls --with-sysroot=/ --enable-clocale=gnu --enable-libstdcx
x-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-abi=new --enable-gnu-u
nique-object --disable-vtable-verify --enable-libmpx --enable-plugin --enable-defau
lt-pie --with-system-zlib --with-target-system-zlib --enable-objc-gc=auto --enable-
multiarch --disable-werror --with-arch-32=i686 --with-abi=m64 --with-multilib-list=
m32,m64,mx32 --enable-multilib --with-tune=generic --enable-offload-targets=nvptx-n
one --without-cuda-driver --enable-checking=release --build=x86_64-linux-gnu --host
=x86_64-linux-gnu --target=x86_64-linux-gnu
Thread model: posix
gcc version 7.3.0 (Ubuntu 7.3.0-27ubuntu1~18.04)
COLLECT_GCC_OPTIONS='-v' '-mtune=generic' '-march=x86-64'
 /usr/lib/gcc/x86_64-linux-gnu/7/cc1 -quiet -v -imultiarch x86_64-linux-gnu example
.c -quiet -dumpbase example.c -mtune=generic -march=x86-64 -auxbase example -versio
n -fstack-protector-strong -Wformat -Wformat-security -o /tmp/ccyOtN1M.s
GNU C11 (Ubuntu 7.3.0-27ubuntu1~18.04) version 7.3.0 (x86_64-linux-gnu)
        compiled by GNU C version 7.3.0, GMP version 6.1.2, MPFR version 4.0.1, MPC
 version 1.1.0, isl version isl-0.19-GMP

GGC heuristics: --param ggc-min-expand=100 --param ggc-min-heapsize=131072
ignoring nonexistent directory "/usr/local/include/x86_64-linux-gnu"
ignoring nonexistent directory "/usr/lib/gcc/x86_64-linux-gnu/7/../../../../x86_64-
linux-gnu/include"
#include "..." search starts here:
#include <...> search starts here:
 /usr/lib/gcc/x86_64-linux-gnu/7/include
 /usr/local/include
 /usr/lib/gcc/x86_64-linux-gnu/7/include-fixed
 /usr/include/x86_64-linux-gnu
```

دستور g ++

دستور **g ++** یک دستور فراخوانی کامپایلر **G + C** است که برای پیش پردازش ، تدوین ، مونتاژ و پیوند دادن کد منبع برای تولید یک فایل اجرایی استفاده می شود" .گزینه های" مختلف دستور g ++ به ما امکان می دهد تا این مرحله را در مرحله میانی متوقف کنیم.

ساختار دستور

```
g++ [options] source_file(s) -o output_file
```

اجزای دستور

- options گزینه‌هایی هستند که نحوه کامپایل کد را مشخص می‌کنند.
- source_file(s) فایل‌های منبع C++که می‌خواهید کامپایل شوند.
- -o گزینه خروجی که نام فایل قابل اجرا را مشخص می‌کند.

گزینه‌های دستور :

- -c فقط کد را کامپایل می‌کند و فایل قابل اجرا ایجاد نمی‌کند.
- -g اطلاعات اشکال‌زدایی را به کد اضافه می‌کند.
- -O کد را برای افزایش سرعت بهینه‌سازی می‌کند.
- -Wall تمام هشدارهای کامپایلر را فعال می‌کند.
- -std=c++XX نسخه استاندارد زبان C++را که می‌خواهید از آن استفاده کنید را مشخص می‌کند (مثلاً -std=c++17).

مثال های دستور :

بررسی اطلاعات نسخه کامپایلر g ++

```
g++ --version
```

```
ak@ubuntu:~$ g++ --version
g++ (Ubuntu 6.3.0-12ubuntu2) 6.3.0 20170406
Copyright (C) 2016 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

برای تولید فایل هدف اجرایی ، یک فایل CPP کامپایل کنید: از دستور **g ++ file_name** برای کامپایل و ایجاد یک فایل اجرایی) **a.out** نام پیش فرض هدف) استفاده می شود.

مثال : با یک برنامه ساده برای چاپ "Hello Geek" در خروجی استاندارد با نام فایل **hello.cpp**

```
// hello.cpp file
#include <iostream>
int  main()
{
   std::cout << "Hello Geek\n";
   return  0;
}
```

```
g ++ hello.cpp
```

```
ak@ubuntu:~$ g++ hello.cpp
```

این **hello.cpp** را کامپایل و پیوند می **دهد** تا یک فایل اجرایی پیش فرض **a.out** در فهرست کار فعلی تولید کند .برای اجرای این برنامه ، **./a.out** جایی که **./** نشان دهنده پوشه فعلی است را تایپ کنید و **a.out** فایل هدف اجرایی است.

./a.out

```
ak@ubuntu:~$ ./a.out
Hello Geek
```

g ++ -S file_nameفقط برای کامپایل **نام_ فایل** و **نه** مونتاژ یا پیوند دادن استفاده می شود .این یک فایل منبع اسمبلی **file_name.s** ایجاد می کند.

مثال :

```
g ++ -S hello.cpp
```

g ++ -c file_nameفقط برای کامپایل و جمع آوری **file_name** و **عدم** پیوند کد شی برای تولید فایل اجرایی استفاده می شود .این یک فایل کد شی **file_name.o** را در پوشه موجود ایجاد می کند.

مثال :

```
g ++ -c hello.cpp
```

G ++ -o TARGET_NAME FILE_NAME کامپایل و لینک **FILE_NAME** و تولید فایل مورد نظر اجرایی با) **TARGET_NAME** یا فایلهای دو دویی a.out به طور پیش فرض

مثال :

```
g ++ -o main.exe hello.cpp
```

```
ak@ubuntu:~$ g++ -o main.exe hello.cpp
ak@ubuntu:~$ ./main.exe
Hello Geek
```

چند فایل بایکدیگر برای کامپایل

```
g ++ -c helloWorld.cpp hello.cpp
```

این کد به ترتیب برای پرونده های helloWorld.cpp و hello.cpp به helloWorld.o و hello.o کامپایل و ایجاد می کند.

```
g ++ -o main.exe hello helloWorld.o hello
```

این کدهای شی helloWorld.o و hello.o را برای ایجاد یک فایل اجرایی main.exe پیوند می دهد

./main.exe

این فایل اجرایی main.exe را اجرا می کند

```
ak@ubuntu:~$ g++ -c helloWorld.cpp hello.cpp
ak@ubuntu:~$ g++ -o main.exe helloWorld.o hello.o
ak@ubuntu:~$ ./main.exe
Hello Geek
Hello World
```

g ++ -Wall file_name این پیام های هشدار دهنده ای را که در هنگام جمع آوری **نام_ پرونده** ایجاد می شوند چاپ می کند.

دستور nasm

دستور NASM یا Netwide Assembler یک ابزار متن‌باز است که برای ترجمه و آماده‌سازی کد ماشین برای پردازنده‌های x86و x86-64 استفاده می‌شود. این ابزار معمولاً در برنامه نویسی سطح پایین و توسعه نرم‌افزارهای سیستمی استفاده می‌شود.

ساختار کلی دستور :

```
nasm [options] source_file [-o output_file]
```

اجزای دستور:

- nasm: نام خود اسمبلر
- options: گزینه‌های مختلف اسمبلر
- source_file: نام فایل حاوی کد اسمبلی
- -o: سوئیچ برای تعیین نام فایل خروجی
- output_file: نام فایل خروجی (اختیاری)

گزینه‌های دستور:

- f- فرمت فایل خروجی را مشخص می‌کند. مقادیر رایج برای این گزینه عبارتند از
  - elf فرمت ELF برای لینوکس و macOS
  - win32 فرمت PE برای ویندوز
  - macho فرمت Mach-O برای macOS
- g- اطلاعات اشکال‌زدایی را در فایل خروجی قرار می‌دهد.
- O- سطح بهینه‌سازی را مشخص می‌کند. مقادیر رایج برای این گزینه عبارتند از
  - ۰ بدون بهینه‌سازی
  - ۱ بهینه‌سازی سطح ۱
  - ۲ بهینه‌سازی سطح ۲
- s- نام نمادهای عمومی را مشخص می‌کند.
- l- کتابخانه‌های مورد نیاز برای پیوند را مشخص می‌کند.

مثال های دستور :

ترجمه و آماده‌سازی برنامه با استفاده از NASM:

```
$nasm -f elf32 program.asm -o program.o
```

این دستور برنامهای با نام "program.asm" را به عنوان ورودی دریافت کرده و با استفاده از NASM آن را به یک فایل شی با فرمت ELF32 ترجمه می‌کند و در فایل با نام "program.o" ذخیره می‌کند.

ترجمه و آماده‌سازی برنامه با استفاده از NASM برای پردازنده x86-64:

```
$nasm -f elf64 program.asm -o program.o
```

این دستور برنامهای با نام "program.asm" را به عنوان ورودی دریافت کرده و با استفاده از NASM آن را به یک فایل شی با فرمت ELF64 ترجمه می‌کند و در فایل با نام "program.o" ذخیره می‌کند.

ترجمه و آماده‌سازی برنامه با استفاده از NASM و لینک کردن با یک لینکر خارجی:

```
$nasm -f elf32 program.asm -o program.o
ld -m elf_i386 program.o -o program
```

این دستور ابتدا برنامهای با نام "program.asm" را به عنوان ورودی دریافت کرده و با استفاده از NASM آن را به یک فایل شی با فرمت ELF32 ترجمه می‌کند و در فایل با نام "program.o" ذخیره می‌کند. سپس با استفاده از لینکر ld و فلگ "-m elf_i386"، فایل شی را با لینکر خارجی لینک می‌کند و فایل قابل اجرا با نام "program" را تولید می‌کند.

مثال ۱: اسمبل کردن یک فایل اسمبلی به نام hello.asm با فرمت ELF و اطلاعات اشکال‌زدایی:

```
nasm -f elf -g hello.asm
```

مثال ۲: اسمبل کردن یک فایل اسمبلی به نام main.asm با فرمت PE و بهینه‌سازی سطح ۲:

```
nasm -f win32 -O2 main.asm -o main.exe
```

مثال ۳: اسمبل کردن یک فایل اسمبلی به نام factorial.asm با استفاده از کتابخانه math:

```
nasm -f elf -l math factorial.asm -o factorial.o
```

دستور java

دستور java در سیستم‌های که JDK (Java Development Kit) نصب شده است برای اجرای و مدیریت برنامه‌های نوشته شده به زبان Java استفاده می‌شود. این دستور به شما این امکان را می‌دهد که برنامه‌های Java را اجرا کنید و پارامترهای مورد نیاز آن‌ها را تنظیم کنید.

ساختار دستور

```
java [options] <class-name> [arguments]
```

- options: گزینه های مختلفی هستند که می توانید برای تغییر نحوه اجرای برنامه از آنها استفاده کنید.
- class-name: نام کلاس اصلی برنامه شما است.
- arguments: پارامترهایی هستند که می توانید به برنامه خود ارسال کنید.

گزینه های دستور

در ادامه به برخی از رایج ترین گزینه های دستور java اشاره می کنیم

- version- نسخه فعلی Java را نمایش می دهد.
- classpath- مسیر کلاس ها و کتابخانه های مورد نیاز برنامه را مشخص می کند.
- D- متغیرهای سیستمی را برای برنامه شما تعریف می کند.
- Xmx- حداکثر حافظه ای که برنامه می تواند استفاده کند را مشخص می کند.

- jar- یک فایل JAR را اجرا می کند.

مثال های دستور:

اجرای یک فایل class: این دستور برای اجرای یک فایل کامپایل شده Java با فرمت .class استفاده می‌شود.

```
java MyClass
```

اجرای یک برنامه .jar: این دستور برای اجرای یک فایل JAR (Java Archive) استفاده می‌شود.

```
java -jar myprogram.jar
```

تنظیم حداقل و حداکثر حافظه Heap: این دستور برای تنظیم حداقل و حداکثر حافظه Heap که برنامه Java می‌تواند استفاده کند، استفاده می‌شود.

```
java -Xms512m -Xmx1024m MyClass
```

تنظیم متغیرهای محیطی: این دستور برای تنظیم متغیرهای محیطی (Environment Variables) برای برنامه Java استفاده می‌شود.

```
java -Dmyproperty=value MyClass
```

نمایش اطلاعات نسخه Java: این دستور برای نمایش اطلاعات مربوط به نسخه Java که در سیستم شما نصب شده استفاده می‌شود.

```
java -version
```

با استفاده از دستور java می‌توانید برنامه‌های Java خود را اجرا کرده و پیکربندی‌های مختلفی را برای اجرای آن‌ها تنظیم کنید.

اجرای یک برنامه ساده:

```
java HelloWorld
```

اجرای یک برنامه با استفاده از classpath:

```
java -classpath .:lib/mylibrary.jar MyProgram
```

تخصیص حافظه بیشتر به برنامه:

```
java -Xmx512m MyProgram
```

اجرای یک فایل JAR:

```
java -jar myprogram.jar
```

تعریف یک متغیر سیستمی:

```
java -Dmy.variable=value MyProgram
```

دستور python

دستور python در سیستم‌هایی که Python نصب شده است برای اجرای اسکریپت‌ها و کدهای نوشته شده به زبان Python استفاده می‌شود. این دستور به شما این امکان را می‌دهد که اسکریپت‌های Python خود را اجرا کرده و پارامترهای مورد نیاز آن‌ها را تنظیم کنید.

ساختار دستور

```
python [options] [script.py] [arguments]
```

گزینه‌های دستور:

- V- نمایش نسخه پایتون
- h- نمایش راهنمای دستور
- i- ورود به حالت تعاملی مفسر
- c- اجرای کد پایتون به صورت خط فرمان
- m- اجرای یک ماژول پایتون

- O- بهینه‌سازی کد پایتون برای افزایش سرعت
- W- فعال کردن هشدارهای مربوط به خطاهای احتمالی
- q- غیرفعال کردن چاپ پیغام‌های خوش‌آمد و خداحافظی
- B- فعال کردن حالت batch
- u- فعال کردن Unicode
- E- فعال کردن shebang line

مثال‌های دستور:

اجرای یک اسکریپت Python: این دستور برای اجرای یک فایل Python با پسوند py. استفاده می‌شود.

```
python myscript.py
```

اجرای کد Python در حالت تعاملی (Interactive Mode): این دستور برای ورود به محیط تعاملی Python به صورت تعاملی و اجرای دستورات Python به صورت تک تک استفاده می‌شود.

```
python
```

نصب پکیج‌های Python با استفاده از pip: این دستور برای نصب یک پکیج Python از Python Package Index (PyPI) با استفاده از pip استفاده می‌شود.

```
python -m pip install requests
```

اجرای کد Python به صورت Optimized: این دستور برای اجرای یک اسکریپت Python به صورت Optimized (با بهینه‌سازی) استفاده می‌شود.

```
python -O myscript.py
```

نمایش راهنمای استفاده از دستور Python: این دستور برای نمایش راهنمای استفاده از دستور python و گزینه‌های موجود آن استفاده می‌شود.

```
python --help
```

با استفاده از دستور python می‌توانید اسکریپت‌ها و کدهای Python خود را اجرا کرده و پیکربندی‌های مختلفی را برای اجرای آن‌ها تنظیم کنید.

مثال پیشرفته:

نمایش نسخه پایتون:

```
python -V
```

ورود به حالت تعاملی مفسر:

```
python -i
```

اجرای کد پایتون به صورت خط فرمان:

```
python -c "print('Hello, world!')"
```

اجرای یک ماژول پایتون:

```
python -m math
```

بهینه‌سازی کد پایتون برای افزایش سرعت:

```
python -O script.py
```

فعال کردن هشدارهای مربوط به خطاهای احتمالی:

```
python -W all script.py
```

غیرفعال کردن چاپ پیغام‌های خوش‌آمد و خداحافظی:

```
python -q script.py
```

فعال کردن حالت batch:

```
python -B script.py
```

فعال کردن Unicode:

```
python -u script.py
```

فعال کردن shebang line:

```
python -E script.py
```

دستور pip

pipابزاری قدرتمند برای مدیریت بسته‌های پایتون در سیستم‌های لینوکس است. این ابزار به شما امکان می‌دهد تا کتابخانه‌ها و چارچوب‌های پایتون را نصب، حذف، به‌روزرسانی و مدیریت کنید.

ساختار دستور:

```
pip [options] <command> [arguments]
```

گزینه های دستور:

- -h, --help

راهنمای برنامه را نشان میدهد.

- --debug

اجازه میدهد استثناء های بدون دستکاری خارج از زیر برنامه اصلی منتشر شوند، به جای ثبت آنها در stderr.

- --isolated

pip را در یک حالت جدا اجرا میکند، متغیرهای محیطی و پیکربندی کاربر را نادیده بگیرید.

- --require-virtualenv

اجازه میدهد pip فقط در یک محیط مجازی اجرا شود؛ در غیر این صورت با خطا خارج شوید.

- --python <python>

pip را با تفسیرگر Python مشخص شده اجرا میکند.

- -v, --verbose

بیشتر خروجی میدهد. گزینه افزودنی است و می تواند تا ۳ بار استفاده شود.

- -V, --version

نسخه را نشان میدهد و خارج میشود.

- -q, --quiet

کمتر خروجی نشان میدهد.

- --log <path>

توضیحات بیشتر.

- --no-input

غیرفعال کردن درخواست برای ورودی.

- --keyring-provider <keyring_provider>

اگر ورودی کاربر مجاز است، جستجوی اعتبارنامه را از طریق کتابخانه keyring فعال میکند.

- --proxy <proxy>

یک پروکسی را با فرم scheme://[user:passwd@]proxy.server:port مشخص میکند.

- --retries <retries>

حداکثر تعداد تلاش هایی که هر اتصال باید انجام دهد (پیش فرض ۵ بار).

- --timeout <sec>

تنظیم زمان انقضای سوکت (پیش فرض ۱۵ ثانیه).

- --exists-action <action>

(s)witch, (i)gnore, (w)ipe, (b)ackup, (a)bort. :عمل پیش فرض وقتی یک مسیر از قبل وجود دارد

- --trusted-host <hostname>

این میزبان یا جفت میزبان:port را به عنوان مورد اعتماد علامت گذاری میکند، حتی اگر HTTPS معتبر یا هیچ HTTPS نداشته باشد.

- --cert <path>

مسیر به بسته گواهینامه CA با کدگذاری PEM. اگر ارائه شود، پیش فرض را نادیده می گیرد. برای اطلاعات بیشتر ' SSL Certificate Verification'را در مستندات pip ببینید.

- --client-cert <path>

مسیر به گواهینامه مشتری SSL، یک فایل تکی شامل کلید خصوصی و گواهینامه در فرمت PEM.

- --cache-dir <dir>

داده های کش را در dir ذخیره میکند.

- --no-cache-dir

کش را غیرفعال میکند.

- --disable-pip-version-check

به صورت دوره ای PyPI را برای تعیین اینکه آیا نسخه جدیدی از pip برای دانلود موجود است، چک نمیکند.

- --no-color

خروجی رنگی را غیرفعال میکند.

- --no-python-version-warning

عدم هشدار برای از رده خارج شدن نسخه های Python های در بروزرسانی های بعدی.

- --use-feature <feature>

قابلیت جدید را فعال میکند، که ممکن است با نسخه های قبلی ناسازگار باشد.

- --use-deprecated <feature>

قابلیت منسوخ شده را فعال میکنید،

مثال های دستور:

فرض کنید می خواهید راهنمای کامل دستور pip install را مشاهده کنید. برای این کار:

```
pip install --help
```

این دستور خروجی مفصل تمام گزینه‌ها و آرگومان‌های قابل استفاده با pip install را نمایش می‌دهد.

-- debug

این گزینه معمولا برای توسعه دهندگان pip و اشکال زدایی آن کاربرد دارد. در استفاده روزمره نیازی به این گزینه نخواهید داشت.

--isolated

فرض کنید در حال کار بر روی یک پروژه خاص هستید و نمی‌خواهید تنظیمات و محیط کلی سیستم شما بر نصب بسته‌ها تاثیر بگذارد. با استفاده از این گزینه، pip به صورت ایزوله اجرا شده و تنظیمات محیطی را نادیده می‌گیرد.

```
pip install --isolated some-package
```

--require-virtualenv

این گزینه pip را مجبور می‌کند تا تنها در یک محیط مجازی ((virtual environmentاجرا شود. در صورتی که خارج از محیط مجازی باشید، با خطا مواجه خواهید شد. این گزینه برای اطمینان از اینکه بسته‌ها در یک محیط ایزوله نصب می‌شوند، مفید است.

```
pip install --require-virtualenv some-package
```

--python <python interpreter path>

فرض کنید چندین نسخه از پایتون در سیستم خود دارید و می‌خواهید pip از یک نسخه خاص استفاده کند. با این گزینه می‌توانید مسیر آن نسخه را مشخص کنید.

```
pip install --python /usr/bin/python3.9 some-package
```

-v (یا verbose --)

این گزینه میزان جزئیات خروجی pip را افزایش می‌دهد. با استفاده چندباره از این گزینه (تا سقف ۳ بار)، خروجی با جزئیات بیشتری همراه خواهد بود.

```
pip install -v some-package  # خروجی با جزئیات کم
pip install -vv some-package # خروجی با جزئیات متوسط
pip install -vvv some-package # خروجی با جزئیات زیاد
```

keyring-provider

این گزینه به شما امکان می‌دهد تا از طریق کتابخانه keyring، اطلاعات احراز هویت را بازیابی کنید. این موضوع در شرایطی که نیاز به رمز عبور برای دسترسی به یک منبع دارید، مفید خواهد بود.

```
pip install --keyring-provider=subprocess some-package
```

proxy

با استفاده از این گزینه می‌توانید یک پروکسی برای اتصال به اینترنت در pip مشخص کنید.

```
pip install --proxy=http://user:pass@proxy.example.com:8080 some-package
```

retries

این گزینه حداکثر تعداد تلاش‌های pip برای اتصال به یک منبع را تعیین می‌کند.

```
pip install --retries=10 some-package
```

timeout

با این گزینه می‌توانید زمان انتظار pip برای اتصال به یک منبع را مشخص کنید.

```
pip install --timeout=30 some-package
```

exists-action

این گزینه نحوه برخورد pip با وجود یک مسیر از قبل موجود را تعیین می‌کند.

```
pip install --exists-action=i some-package  # Ignore existing path
pip install --exists-action=w some-package  # Wipe existing path
```

trusted-host

این گزینه به شما امکان می‌دهد تا یک میزبان (یا میزبانپورت) را به عنوان قابل اعتماد علامت گذاری کنید، حتی اگر گواهینامه HTTPS معتبر نداشته باشد.

```
pip install --trusted-host=example.com some-package
```

cert

با این گزینه می‌توانید مسیر یک گواهی (Certificate Authority) CA را برای بررسی گواهینامه HTTPS مشخص کنید.

```
pip install --cert=/path/to/ca.pem some-package
```

client-cert

این گزینه مسیر یک گواهی (Secure Sockets Layer) SSL را برای احراز هویت کلاینت در pip مشخص می‌کند.

```
pip install --client-cert=/path/to/client.pem some-package
```

cache-dir

با این گزینه می‌توانید مسیر ذخیره‌سازی حافظه پنهان pip را تغییر دهید.

```
pip install --cache-dir=/path/to/cache some-package
```

دستور dialog

دستور dialog در لینوکس برای نمایش جعبه‌های دیالوگ از اسکریپت‌های شل ((shell scriptsاستفاده می‌شود. این ابزار به کاربران این امکان را می‌دهد تا از طریق خط فرمان و اسکریپت‌های خود، با استفاده از جعبه‌های دیالوگ، اطلاعات را به صورت تعاملی نمایش دهند و از کاربر ورودی دریافت کنند.

گزینه‌های دستور:

- Msgbox: نمایش یک پیام در یک جعبه دیالوگ
- Inputbox: نمایش یک باکس ورودی برای ورود اطلاعات
- Textbox: نمایش محتوای یک فایل متنی در یک باکس دیالوگ
- Menu: نمایش یک منو انتخاب از گزینه‌های مختلف
- Yesno: نمایش یک سوال با پاسخ‌های بله و خیر
- Calendar: نمایش یک تقویم برای انتخاب تاریخ
- Progressbox: نمایش یک جعبه دیالوگ پیشرفت
- Radiolist: نمایش یک لیست انتخابی با دکمه‌های رادیویی
- Checklist: نمایش یک لیست انتخابی با دکمه‌های چک‌باکس

مثال‌های دستور:

نمایش پیام در یک جعبه دیالوگ:

```
dialog --msgbox "Welcome to my script!" 10 30
```

نمایش باکس ورودی برای دریافت نام کاربر:

```
dialog --inputbox "Enter your name:" 10 30 2> output.txt
```

نمایش محتوای یک فایل متنی در یک باکس دیالوگ:

```
dialog --textbox file.txt 20 50
```

نمایش منو انتخاب از گزینه‌های مختلف:

```
dialog --menu "Choose an option:" 20 50 4 1 "Option 1" 2 "Option 2" 3 "Option 3" 4 "Option 4"
```

نمایش سوال با پاسخ‌های بله و خیر:

```
dialog --yesno "Are you sure you want to proceed?" 10 30
```

نمایش تقویم برای انتخاب تاریخ:

```
dialog --calendar "Select a date:" 0 0
```

نمایش جعبه دیالوگ پیشرفت:

```
dialog --progressbox "Please wait..." 10 30
```

نمایش لیست انتخابی با دکمه‌های رادیویی:

```
dialog --radiolist "Choose one:" 20 50 4 1 "Option A" on 2 "Option B" off 3 "Option C" off 4 "Option D" off
```

نمایش لیست انتخابی با دکمه‌های چک‌باکس:

```
dialog --checklist "Choose multiple options:" 20 50 4 1 "Option X" off 2 "Option Y" on 3 "Option Z" off
```

دستور expr

دستور expr در لینوکس برای انجام عملیات ریاضی، مقایسه‌ها و اعمال منطقی استفاده می‌شود. این دستور امکان انجام انواع عملیات حسابی و منطقی را فراهم می‌کند.

گزینه‌های دستور:

- length STRING: بازگرداندن طول یک رشته
- index STRING CHARS: پیدا کردن موقعیت اولین کاراکتر از CHARS در STRING
- substr STRING POS LENGTH: استخراج زیررشته از STRING با شروع از POS و با طول LENGTH
- match REGEX STRING: بررسی تطابق یک رشته با یک عبارت منظقی (regex)
- :: انجام عملیات جمع
- :* انجام عملیات ضرب
- :/ انجام عملیات تقسیم
- :% باقی‌مانده تقسیم
- :+ انجام عملیات جمع
- :- انجام عملیات تفریق
- :< عملیات مقایسه بزرگتر بودن
- :> عملیات مقایسه کوچکتر بودن
- := عملیات مقایسه برابری
- :!= عملیات مقایسه نابرابری

مثال‌های دستور:

بازگرداندن طول یک رشته:

```
expr length "Hello, World!"
```

پیدا کردن موقعیت اولین کاراکتر از CHARS در STRING:

```
expr index "Hello, World!" o
```

استخراج زیررشته از STRING:

```
expr substr "Hello, World!" 7 5
```

بررسی تطابق یک رشته با یک عبارت منظقی:

```
expr match "Hello, World!" '.*World.*'
```

انجام عملیات جمع:

```
expr 5 + 3
```

انجام عملیات ضرب:

```
expr 5 \* 3
```

انجام عملیات تقسیم:

```
expr 15 / 3
```

باقی‌مانده تقسیم:

```
expr 15 % 4
```

مقایسه بزرگتر بودن:

```
expr 5 \> 3
```

مقایسه کوچکتر بودن:

```
expr 5 \< 3
```

مقایسه برابری:

```
expr 5 = 5
```

مقایسه نابرابری:

```
expr 5 != 3
```

دستور factor

دستور factor در لینوکس برای تجزیه عدد به عوامل اول از نظر ریاضی استفاده می‌شود. این دستور به کاربر امکان می‌دهد تا عوامل اول یک عدد را بررسی کرده و نمایش دهد.

گزینه‌های دستور:

- v, --verbose: نمایش اطلاعات بیشتر درباره هر عامل
- q, --quiet: حالت ساکت بدون نمایش پیام‌های اضافی
- a, --all: نمایش همه عوامل حتی اعداد تکراری
- s, --sort: مرتب کردن عوامل به صورت صعودی
- h, --help: نمایش راهنما و خروج

مثال‌های دستور:

نمایش عوامل اول عدد ۳۶:

```
factor 36
```

نمایش اطلاعات بیشتر درباره عوامل عدد ۱۰۰:

```
factor -v 100
```

نمایش همه عوامل حتی اعداد تکراری عدد ۲۴:

```
factor -a 24
```

مرتب کردن عوامل به صورت صعودی برای عدد ۵۰:

```
factor -s 50
```

نمایش راهنما و خروج:

```
factor -h
```

ادغام چند فایل pdf

دستور convert

دستور convert ابزاری قدرتمند برای تبدیل انواع مختلف فایل‌ها در سیستم‌عامل‌های لینوکس است. این ابزار می‌تواند تصاویر، اسناد، فایل‌های صوتی و تصویری و ... را به فرمت‌های مختلف تبدیل کند. به منظور ادغام چند فایل pdf و تبدیل کردن آنها به یک فایل از دستور زیر استفاده نمایید.

ساختار دستور

```
convert [options] input-file output-file
```

گزینه‌های دستور :

- h- یا help-- نمایش راهنمای دستور
- version- نمایش نسخه convert
- verbose- نمایش جزئیات بیشتر در خروجی
- quiet- خروجی را به حداقل می‌رساند
- rotate- چرخاندن تصویر
- resize- تغییر اندازه تصویر
- crop- برش تصویر
- flip- برعکس کردن تصویر
- flop- چرخاندن تصویر به صورت افقی
- format- تعیین فرمت خروجی
- quality- تنظیم کیفیت تصویر خروجی
- delay- تنظیم تاخیر بین صفحات در تصاویر متحرک

مثال های دستور :

تبدیل یک تصویر PNG به JPG:

```
convert image.png image.jpg
```

تغییر اندازه یک تصویر:

```
convert -resize 500x500 image.jpg resized.jpg
```

برش یک تصویر:

```
convert -crop 100x100+100+100 image.jpg cropped.jpg
```

چرخاندن یک تصویر:

```
convert -rotate 90 image.jpg rotated.jpg
```

تبدیل یک فایل PDF به PNG:

```
convert -density 300 input.pdf output.png
```

تبدیل یک فایل صوتی MP3 به WAV:

```
convert input.mp3 output.wav
```

دستور pdftk

pdftkابزاری قدرتمند برای دستکاری فایل‌های PDF در سیستم‌های لینوکس است. این ابزار به شما امکان می‌دهد تا فایل‌های PDFرا ادغام، تقسیم، رمزگذاری، رمزگشایی، واترمارک، فرم‌ها را پر کنید و بسیاری از کارهای دیگر را انجام دهید.

ساختار دستور:

```
pdftk [options] <input_file> [output_file] [command] [arguments]
```

گزینه‌های دستور:

- h- یا help--: نمایش راهنمای دستور
- V- یا version--: نمایش نسخه pdftk
- q- یا quiet--: خروجی را به حداقل می‌رساند
- v- یا verbose--: جزئیات بیشتری را در خروجی نمایش می‌دهد
- p- یا password--: رمز عبور فایل PDF را مشخص می‌کند
- o- یا output--: نام فایل خروجی را مشخص می‌کند

گزینه های دستوری:

- cat: ادغام چندین فایل PDF
- split: تقسیم یک فایل PDF به چندین فایل
- encrypt: رمزگذاری یک فایل PDF
- decrypt: رمزگشایی یک فایل PDF
- watermark: اضافه کردن واترمارک به یک فایل PDF
- fill_form: پر کردن فرم‌های PDF
- stamp: اضافه کردن مهر به یک فایل PDF
- background: اضافه کردن تصویر به عنوان پس‌زمینه یک فایل PDF
- rotate: چرخاندن صفحات یک فایل PDF
- delete: حذف صفحات از یک فایل PDF
- reorder: ترتیب صفحات یک فایل PDF را تغییر می‌دهد

مثال های دستور:

ادغام دو فایل PDF:

```
pdftk file1.pdf file2.pdf cat output.pdf
```

تقسیم یک فایل PDF به دو فایل:

```
pdftk file.pdf burst output
```

رمزگذاری یک فایل PDF:

```
pdftk file.pdf encrypt output.pdf -p password
```

رمزگشایی یک فایل PDF:

```
pdftk file.pdf decrypt output.pdf -p password
```

اضافه کردن واترمارک به یک فایل PDF:

```
pdftk file.pdf background water.pdf output.pdf
```

پر کردن فرم PDF:

```
pdftk file.pdf fill_form data.fdf output.pdf
```

دستور basename

دستور basename ابزاری در خط فرمان لینوکس است که برای استخراج نام پایه یک مسیر فایل یا دایرکتوری به کار می‌رود. به عبارت دیگر، این دستور نام فایل را بدون پسوند و نام دایرکتوری را بدون مسیر والدین آن نمایش می‌دهد.

ساختار دستور:

```
basename [OPTION]... NAME... [SUFFIX]
```

گزینه‌های دستور:

- h– یا help– :نمایش راهنمای دستور
- s– یا suffix– :حذف پسوند مشخص شده از نام فایل
- z– یا zero– :جداکننده null به جای اسلش (/) در خروجی استفاده می‌کند
- a– یا all– :پسوندهای متداول را از نام فایل حذف می‌کند

```
$ basename bin /findhosts.sh
```

basenameاطلاعات پوشه و پسوندها را از نام پرونده ها جدا می کند ، *به عنوان مثال* نام پرونده NAME **را** با حذف هر یک از اجزای اصلی فهرست چاپ می کند .از دستور basename می توان به راحتی با نوشتن basename و به دنبال آن نام پرونده یا نام کامل نام ، استفاده کرد.

ساختار دستور :

```
$basename NAME [SUFFIX]
```

یا

```
$ basename OPTION NAME
```

که در آن ، **NAME** به نام پرونده یا نام کامل پرونده و **OPTION** به گزینه های سازگار با دستور **basename اشاره می کند** و در صورت وجود ، اگر می خواهید پسوند یک پرونده را حذف کنید ، می توانید نام فایل را به همراه نام SUFFIX مورد نظر خود قرار دهید .

مثال :

```
/* basename removes directory path and
   returns the file name, kt */
$basename /usr/local/bin/kt
kt
```

در حالت اول ، دستور **basename** نام فهرست ها را از نام مسیر کامل داده شده از پرونده kt حذف می کند و در حالت دوم **basename** پسوند html. را از kt.html حذف می کند.

گزینه های دستور:

- **a-، -multiple option** - :این گزینه به شما امکان می دهد از چندین آرگومان پشتیبانی کنید و هر یک را به عنوان یک **NAME در نظر** بگیرید ، یعنی با استفاده از گزینه **a-** می توانید چندین نام فایل یا نام مسیر کامل بدهید.

```
/ * برای چندین ورودی استفاده می شود a- * /
$ basename -a / usr / local / bin / kt /usr/local/bin/kt.html
kt
kt.html
```

- **s-، -suffix = گزینهSUFFIX-** :این گزینه پسوند عقب SUFFIX ، مانند پسوند پرونده را حذف می کند.

```
/*.html suffix is removed from kt.html
  when followed by -s option */
```

```
$basename -s .html kt.html
kt
```

- **گزینهz–** :این خروجی را به جای خط جدید با NULL جدا می کند.

```
/* The outputs of the two arguments passed
   are not separated by a newline*/
$basename -az /usr/local/bin/kt kt.html
ktkt
```

- **help option- -** :این اطلاعات راهنما و خروج را نمایش می دهد.
- **-گزینهversion–** :این اطلاعات نسخه و خروج را نمایش می دهد.

مثال بیشتر:

مثال‌ها:

استخراج نام پایه یک فایل:

```
basename file.txt
```

خروجی:

file

حذف پسوند .txt از نام فایل:

```
basename file.txt -s .txt
```

خروجی:

file

استخراج نام دایرکتوری:

```
basename /path/to/directory
```

خروجی:

directory

استفاده از جداکننده null:

```
basename /path/to/directory -z
```

خروجی:

path\0to\0directory

حذف پسوندهای متداول:

```
basename file.tar.gz -a
```

خروجی:

file

دستور dirname

دستور dirname در لینوکس برای استخراج نام دایرکتوری از یک مسیر فایل یا دایرکتوری استفاده می‌شود. دستور dirname در لینوکس توضیح بده.ساختار دستور، و همه گزینه های دستوررا بنویس و برای همه گزینه های دستور dirname تای جایی ممکن است مثال بنویس

ساختار دستور:

```
dirname [OPTION]... PATH
```

گزینه‌های دستور:

- z: – حذف / انتهایی از نام دایرکتوری
- P: – دنبال کردن پیوندهای نمادین
- l: – اگر PATH یک پیوند نمادین است، نام دایرکتوری پیوند را برمی‌گرداند
- f: – اگر PATH یک پیوند نمادین شکسته باشد، نام دایرکتوری پیوند را برمی‌گرداند

مثال های دستور:

- dirnameیک دستور در Linux است که برای حذف "/" اسلایس های عقب جلو از **NAME استفاده می شود** و قسمت باقیمانده را چاپ می کند .اگر آرگومان **NAME** حاوی اسلش فوروارد "/" نباشد ، به راحتی "." را چاپ می‌کند.

```
File  Edit  View  Search  Terminal  Help
naman@root:~$ dirname /Desktop/root/bash.sh
/Desktop/root
naman@root:~$
```

```
File  Edit  View  Search  Terminal  Help
naman@root:~$ dirname bash.sh
.
naman@root:~$
```

گزینه های دستور:

- dirname -zero NAME or dirname --z NAME: این گزینه راهی برای پایان دادن به خط توسط یک دستور nullو نه یک خط جدید فراهم می کند .با استفاده از این دستور بعدی به جای استفاده از خط بعدی در همان خط اجرا می شود

```
File  Edit  View  Search  Terminal  Help
naman@root:~$ dirname -z /Desktop/root/bash.sh
/Desktop/rootnaman@root:~$
```

- dirname –help: اطلاعات راهنما را نمایش می دهد

```
File  Edit  View  Search  Terminal  Help
naman@root:~$ dirname --help
Usage: dirname [OPTION] NAME...
Output each NAME with its last non-slash component and trailing slashes
removed; if NAME contains no /'s, output '.' (meaning the current directory).

  -z, --zero     end each output line with NUL, not newline
      --help     display this help and exit
      --version  output version information and exit

Examples:
  dirname /usr/bin/          -> "/usr"
  dirname dir1/str dir2/str  -> "dir1" followed by "dir2"
  dirname stdio.h            -> "."

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Full documentation at: <http://www.gnu.org/software/coreutils/dirname>
or available locally via: info '(coreutils) dirname invocation'
naman@root:~$
```

- dirname –version: اطلاعات نسخه را نمایش می دهد.

فصل بیست و دوم

فصل بیست و دوم

مدیریت کاربران و گروه ها

مشاهده کاربران در سیستم

دستور users

دستور users در لینوکس برای نمایش لیست کاربران متصل به سیستم استفاده می‌شود. این دستور اطلاعات مختصری از جمله نام کاربری، نام میزبان، زمان ورود و وضعیت هر کاربر را نمایش می‌دهد.

ساختار دستور:

```
users [options]
```

گزینه‌های دستور:

- a- یا --all: نمایش تمام کاربران، حتی کاربرانی که در حال حاضر غیرفعال هستند.
- h- یا --help: نمایش راهنمای دستور
- l- یا --long: نمایش اطلاعات بیشتر از هر کاربر، مانند شناسه کاربر (UID) ، شناسه گروه (GID) و زمان بیکاری
- i- یا --idle: نمایش لیست کاربران بیکار
- n- یا --numeric: نمایش شناسه کاربری (UID) و شناسه گروه (GID) به جای نام کاربری
- p- یا --processes: نمایش تعداد پردازش‌هایی که هر کاربر اجرا می‌کند
- s- یا --sort: مرتب‌سازی لیست کاربران بر اساس معیارهای مختلف، مانند نام کاربری، زمان ورود یا زمان بیکاری
- t- یا --tty: نمایش ترمینالی که هر کاربر از آن استفاده می‌کند

مثال های دستور:

نمایش لیست مختصری از کاربران:

```
users
```

نمایش اطلاعات کامل از تمام کاربران:

```
users -a -l
```

نمایش لیست کاربران بیکار:

```
users -i
```

نمایش شناسه کاربری (UID) و شناسه گروه (GID) به جای نام کاربری:

```
users -n
```

نمایش تعداد پردازش‌هایی که هر کاربر اجرا می‌کند:

```
users -p
```

مرتب‌سازی لیست کاربران بر اساس زمان ورود:

```
users -s login
```

نمایش ترمینالی که هر کاربر از آن استفاده می‌کند:

```
users -t
```

نکات های دستور:

دستور users فقط کاربران متصل به سیستم را نمایش می‌دهد. برای مشاهده لیست تمام کاربران تعریف شده در سیستم، از دستور cat /etc/passwd استفاده کنید.

برای اطلاعات بیشتر در مورد دستورusers ، می‌توانید به مستندات مربوطه در صفحات man مراجعه کنید:

```
man users
```

دستور whomai

دستور whoami در لینوکس برای نمایش اطلاعات مربوط به کاربر فعلی سیستم استفاده می‌شود. این اطلاعات شامل نام کاربری، نام میزبان و شناسه کاربری ( )UIDاست.

ساختار دستور:

```
whoami [options]
```

گزینه‌های دستور:

- h- یا --help: نمایش راهنمای دستور
- u- یا --uid: نمایش شناسه کاربری (UID)
- l- یا --long: نمایش اطلاعات بیشتر شامل نام کامل و گروه کاربری

مثال های دستور:

نمایش نام کاربری:

```
whoami
```

نمایش شناسه کاربری:

```
whoami -u
```

نمایش اطلاعات کامل:

```
whoami -l
```

- دستور whoami به طور پیش فرض نام کاربری را نمایش می‌دهد.
- برای استفاده از این دستور نیازی به دسترسی root ندارید.
- می‌توانید از خروجی whoami در اسکریپت‌های bash برای

گزینه‌های اضافی:

- U- یا --username: نمایش نام کاربری کامل
- g- یا --gid: نمایش شناسه گروه (GID)
- G- یا --groups: نمایش لیست گروه‌های کاربری
- p- یا --pid: نمایش شناسه فرآیند (PID)
- r- یا --real: نمایش شناسه کاربری واقعی (RUID)
- e- یا --effective: نمایش شناسه کاربری موثر (EUID)
- s- یا --session: نمایش شناسه

مثال های دستور:

نمایش نام کاربری کامل:

```
whoami -U
```

نمایش شناسه گروه:

```
whoami -g
```

نمایش لیست گروه‌های کاربری:

```
whoami -G
```

نمایش شناسه فرآیند:

```
whoami -p
```

نمایش شناسه کاربری واقعی:

```
whoami -r
```

نمایش شناسه کاربری موثر:

```
whoami -e
```

دستور who

دستور **$who** برای یافتن اطلاعات زیر استفاده می شود:

۱. زمان آخرین بوت شدن سیستم

۲. سطح اجرای فعلی سیستم

۳. لیست کاربران وارد شده در سیستم و موارد دیگر.

ساختار دستور:

```
$who [options] [نام پرونده]
```

مثال های دستور:

1. دستور $who در صورت عدم ارائه گزینه ، اطلاعات زیر را برای هر کاربر که در حال حاضر وارد سیستم شده است نمایش می دهد:

۱. نام ورود کاربران

۲. شماره خط ترمینال

۳. زمان ورود کاربران به سیستم

۴. نام میزبان از راه دور کاربر



2. برای نمایش نام میزبان و کاربر مرتبط با ورودی استاندارد مانند صفحه کلید

```
$who -m -H
```

3. برای نشان دادن تمام فرایندهای فعال که توسط فرآیند INIT ایجاد می شوند

```
$who -p -H
```

4. برای نمایش وضعیت پیام کاربران به صورت + ، – یا؟

```
$ who -T -H
```

5. برای نشان دادن لیست کاربران وارد شده به سیستم

```
$ $who -u
```

6. برای نشان دادن زمان سیستم هنگام آخرین بوت شدن سیستم

```
$who -b -H
```

7. برای نشان دادن جزئیات تمام فرایندهای مرده

```
$who -d -H
```

8. برای نشان دادن جزئیات روند ورود به سیستم

```
$who -l -H
```

9. برای شمارش تعداد کاربران وارد شده به سیستم

```
$who -q -H
```

10. برای نمایش سطح اجرای فعلی سیستم

```
$who -r
```

11. برای نمایش تمام جزئیات کاربر وارد شده در حال حاضر

```
$who -a
```

12. برای نمایش نام کاربری سیستم

```
$whoami
```

دستور logname

دستور logname در لینوکس برای نمایش نام کاربری (username) متصل شده به جلسه (sessionفعلی استفاده می‌شود.

ساختار دستور:

```
logname
```

گزینه‌های دستور:

- f: این گزینه نام کاربری کامل (full usernameرا نمایش می‌دهد. به طور پیش فرض، فقط نام کاربری کوتاه (short usernameنمایش داده می‌شود.
- l: مشابه گزینه –f عمل می‌کند.
- h: راهنمای دستور را نمایش می‌دهد.

مثال‌های دستور:

نمایش نام کاربری کوتاه

```
logname
```

نمایش نام کاربری کامل

```
logname -f
```

نمایش راهنمای دستور

```
logname -h
```

دستور w

دستورw در سیستم عامل‌های Unix و مشتقات آن مانند Linux برای نمایش اطلاعات جامع در مورد کاربرانی که وارد سیستم شده‌اند، وضعیت سیستم و زمان سیستم استفاده می‌شود. این دستور اطلاعاتی از جمله نام کاربری، ترمینال، زمان ورود، زمان فعلی، مدت زمانی که سیستم روشن بوده و بار سیستم را نمایش می‌دهد.

ساختار دستور:

```
w [options]
```

گزینه‌های دستور:

- -h یا --help: نمایش راهنمای دستور
- -u یا --users: فقط لیست کاربران را نمایش می‌دهد
- -s یا --short: فقط اطلاعات خلاصه را نمایش می‌دهد
- -l یا --long: اطلاعات کامل را نمایش می‌دهد
- -o یا --output: فرمت خروجی را مشخص می‌کند
- -f یا --full-path: مسیر کامل فایل‌ها را نمایش می‌دهد
- -t یا --tty: نام ترمینال کاربر را نمایش می‌دهد
- -i یا --idle: زمان بیکاری کاربر را نمایش می‌دهد
- -p یا --pid: شناسه فرآیند (PID) کاربر را نمایش می‌دهد
- -c یا --cpu: درصد استفاده از CPU توسط کاربر را نمایش می‌دهد
- -m یا --memory: درصد استفاده از حافظه توسط کاربر را نمایش می‌دهد
- -q یا --quiet: فقط تعداد کاربران را نمایش می‌دهد

مثال‌های دستور:

نمایش لیست کاربران:

```
w
```

نمایش اطلاعات خلاصه:

```
w -s
```

نمایش اطلاعات کامل:

```
w -l
```

نمایش فقط نام کاربری و زمان ورود:

```
w -o "%U %W"
```

نمایش مسیر کامل فایل‌ها:

```
w -f
```

نمایش نام ترمینال کاربر:

```
w -t
```

نمایش زمان بیکاری کاربر:

```
w -i
```

نمایش شناسه فرآیند ( )PIDکاربر:

```
w -p
```

نمایش درصد استفاده از CPU توسط کاربر:

```
w -c
```

نمایش درصد استفاده از حافظه توسط کاربر:

```
w -m
```

نمایش فقط تعداد کاربران:

```
w -q
```

نکات:

خروجی دستور w به طور پیش‌فرض شامل اطلاعات مربوط به همه کاربران متصل به سیستم است.

برای اطلاعات بیشتر در مورد هر گزینه، می‌توانید به مستندات صفحات (man pages) man مراجعه کنید.

دستور last

آخرین دستور لیستی از آخرین کاربران وارد شده را نمایش می دهد.

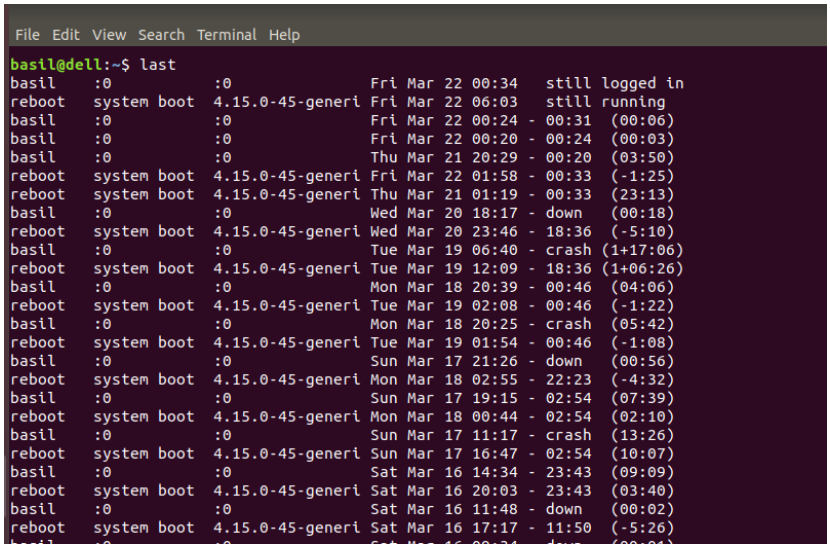
```
$ last
```

دستور **last** در لینوکس استفاده می شود برای نمایش لیستی از تمام کاربران در داخل و خارج به سیستم وارد از آنجا که فایل **/ var / log / wtmp /**ساخته شده است .می توان یک یا چند نام کاربری را به عنوان آرگومان برای نمایش زمان ورود (و یا خارج شدن) و نام میزبان آنها ارائه داد.

ساختار دستور:

```
last [نام کاربری] [گزینه ها ...] [tty ...]
```

مثال های دستور:



گزینه های دستور:

[number]: - از این گزینه برای تعیین تعداد خطوط برای نمایش استفاده می شود.

مثال های دستور:

```
last -5
```

خروجی دستور:

```
File  Edit  View  Search  Terminal  Help
basil@dell:~$ last -5
basil    :0           :0               Fri Mar 22 00:34   still logged in
reboot   system boot  4.15.0-45-generi Fri Mar 22 06:03   still running
basil    :0           :0               Fri Mar 22 00:24 - 00:31  (00:06)
basil    :0           :0               Fri Mar 22 00:20 - 00:24  (00:03)
basil    :0           :0               Thu Mar 21 20:29 - 00:20  (03:50)

wtmp begins Mon Mar  4 00:33:36 2019
basil@dell:~$
```

R:-این گزینه برای پنهان کردن قسمت نام میزبان استفاده می شود.

مثال :

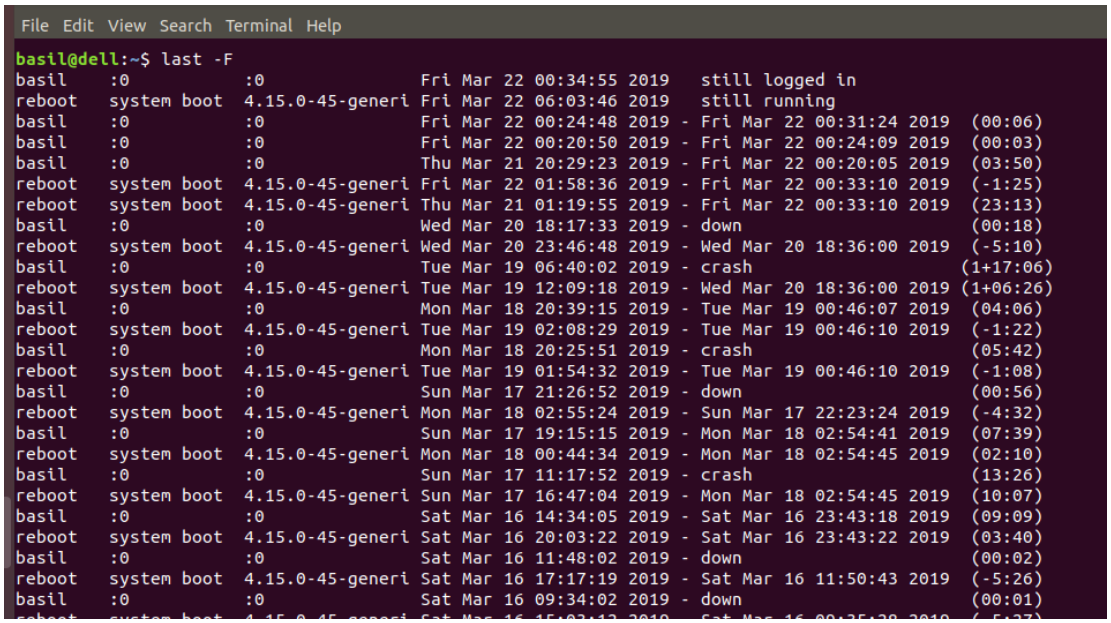
```
last -R ali
```

```
File  Edit  View  Search  Terminal  Help
basil@dell:~$ last -R basil
basil    :0           Fri Mar 22 00:34   still logged in
basil    :0           Fri Mar 22 00:24 - 00:31  (00:06)
basil    :0           Fri Mar 22 00:20 - 00:24  (00:03)
basil    :0           Thu Mar 21 20:29 - 00:20  (03:50)
basil    :0           Wed Mar 20 18:17 - down   (00:18)
basil    :0           Tue Mar 19 06:40 - crash (1+17:06)
basil    :0           Mon Mar 18 20:39 - 00:46  (04:06)
basil    :0           Mon Mar 18 20:25 - crash  (05:42)
basil    :0           Sun Mar 17 21:26 - down   (00:56)
basil    :0           Sun Mar 17 19:15 - 02:54  (07:39)
basil    :0           Sun Mar 17 11:17 - crash  (13:26)
basil    :0           Sat Mar 16 14:34 - 23:43  (09:09)
basil    :0           Sat Mar 16 11:48 - down   (00:02)
basil    :0           Sat Mar 16 09:34 - down   (00:01)
basil    :0           Fri Mar 15 18:29 - crash  (20:33)
```

F:-این گزینه برای نمایش زمان ورود به سیستم و از جمله تاریخ استفاده می شود.

مثال :

```
last -F
```



-a :این گزینه برای نمایش نام میزبان در ستون آخر استفاده می شود.

مثال :

```
last -a
```

-s، **-t** :این گزینه برای نمایش در یک بازه زمانی خاص استفاده می شود. (– ها) از (و) تا ، این دستورات برای قاب بندی دوره زمانی استفاده می شوند.

-d :این گزینه برای ترجمه مجدد آدرس IP به نام میزبان آن استفاده می شود.

مثال :

```
last -d
```

-x :از این گزینه برای نمایش اطلاعاتی مانند ورودی های سیستم و اجرای تغییرات سطح استفاده می شود.

مثال :

```
last -x
```

-w :از این گزینه برای نمایش کامل نام های کاربری و دامنه استفاده می شود.

مثال :

```
last -w ali
```

–help :از این گزینه برای نمایش راهنمایی در مورد همه گزینه های متعلق به **last** دستور استفاده می شود.

دستور hostid

hostid – شناسه عددی میزبان فعلی را چاپ کنید. منظور id است. دستور hostid در سیستم عامل‌های UNIX برای نمایش شناسه یکتای هاست استفاده می‌شود. دستور hostid در لینوکس برای نمایش شناسه عددی منحصر به فردی که به سیستم شما اختصاص داده شده است، استفاده می‌شود. این شناسه برای تمایز بین سیستم‌های مختلف در یک شبکه یا خوشه مورد استفاده قرار می‌گیرد.

ساختار دستور:

```
hostid [options]
```

گزینه‌های دستور:

- h – یا help – :نمایش راهنمای دستور

- l – یا long – :نمایش شناسه عددی به همراه نام سیستم عامل
- s – یا short – :نمایش شناسه عددی به صورت کوتاه

نمایش شناسه هاست به صورت عددی گزینه t–

```
$hostid -t
```

نمایش شناسه هاست به صورت عشری گزینه d–

```
$hostid -d
```

نمایش شناسه هاست به صورت هگزادسیمال گزینه h–

```
$hostid -h
```

دستورid

دستورid نشان می دهد که اطلاعات کاربر و گروه برای کاربر فعلی یا نام کاربری مشخص شده است که در زیر نشان داده شده است.

```
$ id hosein
```

از دستور id در Linux برای یافتن نام کاربر و گروه و id عددی (UID) یا id گروه) کاربر فعلی یا هر کاربر دیگر در سرور استفاده می شود .این دستور برای یافتن اطلاعات زیر به شرح زیر مفید است:

- نام کاربری و id کاربری واقعی
- UIDخاص کاربران.
- UIDو همه گروه های مرتبط با کاربر
- لیست تمام گروه های عضو کاربر.
- زمینه امنیتی کاربر فعلی

ساختار دستور:

```
id [OPTION]... [کاربر]
```

گزینه های دستور:

-g فقط id موثر گروه را چاپ کنید.

-G همه id های گروه را چاپ کنید.

-n نام را به جای شماره چاپ می کند.

-r id واقعی را به جای اعداد چاپ می کند.

-u فقط id کاربری موثر را چاپ می کند.

–help نمایش پیام های راهنما و خروج از آن.

–version اطلاعات نسخه را نمایش داده و از آن خارج شوید.

توجه :بدون هیچ OPTION ، آن مجموعه ای از اطلاعات شناسایی شده یعنی id های عددی را چاپ می کند.

مثال های دستور:

برای چاپ id شخصی خود بدون هیچ گزینه:

```
id
```

```
root@master-VirtualBox:~# id
uid=0(root) gid=0(root) groups=0(root)
root@master-VirtualBox:~#
```

خروجی id کاربر فعلی UID و GID را نشان می دهد.

برای یافتن id خاص کاربران :اکنون فرض کنید که ما یک کاربر به نام master داریم ، برای یافتن UID وی از دستور زیر استفاده خواهیم کرد:

```
id -u master
```

```
root@master-VirtualBox:~# id -u master
1000
root@master-VirtualBox:~#
```

برای یافتن GID کاربران خاص :باز هم با فرض یافتن GID master ، از دستور زیر استفاده خواهیم کرد:

```
id -g master
```

```
root@master-VirtualBox:~# id -g master
1000
root@master-VirtualBox:~#
```

برای پیدا کردن UID و همه گروه های مرتبط با نام کاربری :در این حالت ما از کاربر "master" برای پیدا کردن UID و همه گروه های مرتبط با آن استفاده خواهیم کرد ، از دستور زیر استفاده کنید:

```
root@master-VirtualBox:~# id master
uid=1000(master) gid=1000(master) groups=1000(master),4(adm),24(cdrom),27(sudo),
30(dip),46(plugdev),118(lpadmin),128(sambashare)
root@master-VirtualBox:~#
```

برای فهمیدن همه گروه هایی که یک کاربر به آن تعلق دارد :نمایش UID و همه گروه های "master" کاربر متعلق به:

```
id -G master
```

برای نمایش نام به جای اعداد :به طور پیش فرض دستور id به ما UDI و GID را در اعدادی نشان می دهد که ممکن است کاربر متوجه نشود ، با استفاده از گزینه n- با u- ، g-و G- ، از دستور (ها) استفاده کنید

```
id -ng master
```

یا

```
master  -nu master
```

یا

```
id -nG master
```

```
root@master-VirtualBox:~# id -nG master
master adm cdrom sudo dip plugdev lpadmin sambashare
root@master-VirtualBox:~#
```

برای نمایش id واقعی به جای id موثر :برای نشان دادن id واقعی با استفاده از گزینه r- با g- ، u-و G- ، از دستور (های) استفاده کنید:

```
id -r -g master
id -r -u master
```

```
id -r -G master
```

```
root@master-VirtualBox:~# id -r -G master
1000 4 24 27 30 46 118 128
root@master-VirtualBox:~#
```

دستور ac

این دستور چاپ آماری وضعیت مربوط به زمان اتصال کاربران است.

ساختار دستور

```
ac
[ -d | --daily-totals ] [ -y | --print-year ]
[ -p | --individual-totals ] [ people ]
[ -f | --file filename ] [ -a | --all-days ]
[ --complain ] [ --reboots ] [ --supplants ]
[ --timewarps ] [ --compatibility ]
[ --tw-leniency num ] [ --tw-suspicious num ]
[ -z | --print-zeros ] [ --debug ]
[ -V | --version ] [ -h | --help ]
```

شرح دستور

ac گزارشی از زمان اتصال (به ساعت) بر اساس ورود/خروج در فایل wtmp فعلی چاپ می‌کند. ابتدا ac، فایلی با طول صفر ایجاد کنید. توجه: فایل wtmp می تواند بسیار بزرگ و بسیار سریع ایجاد شود. در سیستم عامل های مشابه یونیکس، دستور ac آماری را در مورد مدت زمانی که کاربران به سیستم متصل هستند نمایش می‌دهد. در برخی از بسته ها، دستور ac به عنوان بخشی از بسته acct گنجانده شده است. اگر می خواهید دیدی کلی از میزان فعال بودن کاربران در یک سیستم داشته باشید، گزینه individual-totals– – آن را فراهم می‌کند. گزارش آن را بر اساس لاگین ها و خروج های ثبت شده در فایل /var/log/wtmp میتوانید ببینید.

```
$ ac --individual-totals
seth     20.16
larry    43.60
curly    10.32
moe      35.11
```

```
$ ac --daily-totals
Jan 20  total      22.61
Jan 21  total      73.60
Jan 22  total      84.00
Jan 23  total      100.69
Jan 24  total      18.24
Jan 25  total       2.43
Jan 27  total      35.36
Today   total      62.13
```

گزینه‌های دستور

- h– یا help– نمایش راهنمای دستور
- V– یا version– نمایش نسخه ac
- m– یا modify– تغییر ACL یک فایل یا دایرکتوری
- e– یا entry– اضافه کردن یک ورودی جدید به ACL

- -d یا --delete حذف یک ورودی از ACL
- -l یا --list نمایش ACL یک فایل یا دایرکتوری
- -R یا --recursive اعمال تغییرات به صورت بازگشتی در تمام زیرشاخه‌ها
- -c یا --check بررسی مجوزهای دسترسی یک کاربر یا گروه خاص

مثال های دستور:

```
-d | --daily-totals:
$ac -d
```

این دستور آمار استفاده روزانه از سیستم را نمایش می‌دهد.

```
-y | --print-year:
$ac -y
```

این دستور آمار استفاده را برای سال جاری نمایش می‌دهد.

```
-p | --individual-totals:
$ac -p
```

این دستور آمار استفاده را برای تمامی کاربران نمایش می‌دهد.

```
[people]:
$ac user1 user2
```

این دستور آمار استفاده را برای کاربران مشخص شده نمایش می‌دهد.

```
-f | --file [filename]:
$ac -f logfile.txt
```

این دستور آمار استفاده را برای فایل لاگ مشخص شده نمایش می‌دهد.

```
-a | --all-days:
ac -a
```

این دستور آمار استفاده را برای تمامی روزها نمایش می‌دهد.

```
--complain:
$ac --complain
```

این دستور هشدار می‌دهد اگر زمان استفاده بیش از حد طولانی باشد.

```
--reboots:
$ac --reboots
```

این دستور تعداد بارهایی که سیستم راهاندازی مجدد شده را نمایش می‌دهد.

```
--supplants:
$ac --supplants
```

این دستور تعداد بارهایی که یک کاربر دیگر را بر روی یک کاربر دیگر جایگزین کرده است را نمایش می‌دهد.

```
--timewarps:
$ac --timewarps
```

این دستور تعداد دفعاتی که زمان سیستم تغییر کرده است را نمایش می‌دهد.

```
--compatibility:
$ac --compatibility
```

این دستور نحوه تطبیق نسخه‌های قدیمی ac را فعال می‌کند.

```
--tw-leniency [num]:
$ac --tw-leniency 5
```

این دستور تنظیم میزان نرمش در تطبیق زمان را مشخص می‌کند.

```
--tw-suspicious [num]:
```

```
$ac --tw-suspicious 10
```

این دستور تنظیم میزان مشکوکیت در تطبیق زمان را مشخص می‌کند.

```
-z | --print-zeros:
$ac -z
```

این دستور آمار استفاده را برای کاربرانی که استفاده‌ای نداشته اند نمایش می‌دهد.

```
--debug:
$ac --debug
```

این دستور حالت اشکال‌زدایی را فعال می‌کند و اطلاعات بیشتری درباره عملکرد دستور نمایش می‌دهد.

```
-V | --version:
$ac -V
```

این دستور نسخه ac را نمایش می‌دهد.

```
-h | --help:
$ac -h
```

این دستور راهنمای استفاده از دستور ac را نمایش می‌دهد.

دستور accton

دستور accton در سیستم‌های لینوکس برای فعال‌سازی یا غیرفعال‌سازی مکانیزم گزارش‌دهی و ثبت فعالیت‌های سیستم (accounting) استفاده می‌شود. این دستور به کاربر اجازه می‌دهد تا فعالیت‌های مختلف سیستم را مانند زمان شروع و پایان هر فرآیند، مصرف منابع سیستم و ... ثبت و مدیریت کند. همچنین دستور accton در لینوکس برای فعال کردن و غیرفعال کردن ضبط اطلاعات مربوط به استفاده از حساب کاربری در سیستم استفاده می‌شود. این اطلاعات شامل مواردی مانند زمان ورود و خروج کاربران، مدت زمان استفاده از سیستم، دستورات اجرا شده و منابع مصرفی (مانند CPU و حافظه) می‌شود.

ساختار دستور:

```
accton [options] [file]
```

گزینه‌های دستور:

- c: فعال کردن ضبط اطلاعات مربوط به خط فرمان (مانند دستورات اجرا شده)
- d: غیرفعال کردن ضبط اطلاعات
- f: مشخص کردن نام فایلی برای ذخیره اطلاعات ضبط شده (پیش‌فرض: /var/account/pacct)
- i: فعال کردن ضبط اطلاعات مربوط به ورود و خروج کاربران
- l: فعال کردن ضبط اطلاعات مربوط به منابع مصرفی (مانند CPU و حافظه)
- m: فعال کردن ضبط اطلاعات مربوط به ارسال و دریافت پیام (مانند ایمیل)
- p: فعال کردن ضبط اطلاعات مربوط به چاپ
- s: فعال کردن ضبط اطلاعات مربوط به سیگنال‌ها (مانند SIGKILL)
- t: فعال کردن ضبط اطلاعات مربوط به زمان

بدون گزینه: نمایش وضعیت فعلی مکانیزم گزارش‌دهی فعالیت‌های سیستم.

```
accton
```

خروجی دستور: /var/log/account/pacct

E: فعال‌سازی مکانیزم گزارش‌دهی فعالیت‌های سیستم.

```
accton -E
```

خروجی دستور: (بدون خروجی)

E: غیرفعال‌سازی مکانیزم گزارش‌دهی فعالیت‌های سیستم.

```
accton -e
```

خروجی دستور: (بدون خروجی)

f: فعال‌سازی مکانیزم گزارش‌دهی فعالیت‌های سیستم با ذخیره‌سازی فایل گزارش.

```
accton -f /var/log/account/report
```

خروجی دستور: (بدون خروجی)

r: تنظیم مسیر فایل گزارش برای ذخیره فعالیت‌های سیستم.

```
accton -r /var/log/accounting
```

خروجی دستور: (بدون خروجی)

v: نمایش اطلاعات بیشتر درباره تنظیمات مکانیزم گزارش‌دهی فعالیت‌های سیستم.

```
accton -v
```

خروجی دستور: Accounting enabled at /var/log/account/pacct

h: نمایش راهنمای استفاده از دستور accton.

```
accton –h
```

مثال‌ها:

فعال کردن ضبط اطلاعات مربوط به ورود و خروج کاربران:

```
accton -i
```

فعال کردن ضبط اطلاعات مربوط به خط فرمان و ذخیره اطلاعات در فایل: my_acct.log

```
accton -c -f my_acct.log
```

غیرفعال کردن ضبط اطلاعات:

```
accton -d
```

نمایش اطلاعات مربوط به نحوه استفاده از دستور: accton

```
man accton
```

دستور LDAP

LDAP چیست؟

LDAP (Lightweight Directory Access Protocol) یک پروتکل شبکه برای خواندن و نوشتن اطلاعات در یک دایرکتوری است. دایرکتوری‌های LDAP اغلب برای مدیریت کاربران، گروه‌ها، دستگاه‌ها و سایر اطلاعات در شبکه‌های بزرگ استفاده می‌شوند.

دستورات LDAP

```
ldapsearch
```

شرح: برای جستجو در یک دایرکتوری LDAP استفاده می‌شود. **ساختار:**

```
ldapsearch [-h host] [-p port] [-D binddn] [-w bindpw] [-x] baseDN searchFilter
```

- `-h host`: آدرس IP یا نام میزبان سرور LDAP
- `-p port`: پورت LDAP (معمولاً ۳۸۹)

- `-D binddn`: نام مشخصه (DN) برای اتصال
- `-w bindpw`: کلمه عبور برای DN اتصال
- `-x`: استفاده از احراز هویت ساده
- `baseDN`: پایه جستجو در دایرکتوری
- `searchFilter`: فیلتر جستجو

مثال:

```
ldapsearch -h ldap.example.com -D "cn=admin,dc=example,dc=com" -w secret -x "(objectClass=user)"
```

دستور ldappasswd

این دستور تمام اشیاء با کلاس شیء user را در پایه جستجوی dc=example,dc=com جستجو می‌کند.

```
ldappasswd
```

شرح: برای تغییر کلمه عبور یک ورودی در دایرکتوری LDAP استفاده می‌شود. **ساختار:**

```
ldappasswd -h host -D binddn -w bindpw newpasswd dn
```

- `newpasswd`: کلمه عبور جدید
- `dn`: نام مشخصه (DN) ورودی که می‌خواهید کلمه عبور آن را تغییر دهید

مثال:

```
ldappasswd -h ldap.example.com -D "cn=admin,dc=example,dc=com" -w secret newpassword "uid=john,cn=users,dc=example,dc=com"
```

دستور ldapadd

این دستور کلمه عبور ورودی با DN uid=john,cn=users,dc=example,dc=com را به newpassword تغییر می‌دهد.

```
ldapadd
```

شرح: برای افزودن یک ورودی جدید به دایرکتوری LDAP استفاده می‌شود. **ساختار:**

```
ldapadd -h host -D binddn -w bindpw -x -f ldif-file
```

- `-f ldif-file`: فایل LDIF حاوی اطلاعات ورودی جدید

مثال:

```
ldapadd -h ldap.example.com -D "cn=admin,dc=example,dc=com" -w secret -x -f user.ldif
```

دستور ldapdelete

این دستور ورودی جدید را از فایل user.ldif به دایرکتوری اضافه می‌کند.

```
ldapdelete
```

شرح: برای حذف یک ورودی از دایرکتوری LDAP استفاده می‌شود. **ساختار:**

```
ldapdelete -h host -D binddn -w bindpw dn
```

- `dn`: نام مشخصه (DN) ورودی که می‌خواهید حذف کنید

مثال:

```
ldapdelete -h ldap.example.com -D "cn=admin,dc=example,dc=com" -w secret "uid=john,cn=users,dc=example,dc=com"
```

این دستور ورودی با DN uid=john,cn=users,dc=example,dc=com را حذف می‌کند.

گزینه‌های بیشتر و کاربردها

- **گزینه‌های دیگر :**هر دستور LDAP گزینه‌های دیگری نیز دارد که به شما امکان می‌دهند جزئیات بیشتری را کنترل کنید، مانند `-s base`, `-b`, `-L`, و غیره.
- **کاربردها:**
  - مدیریت کاربران و گروه‌ها
  - مدیریت دستگاه‌ها
  - پیکربندی شبکه
  - ایجاد و مدیریت سرویس‌های دایرکتوری
  - و بسیاری موارد دیگر

نکته‌ها

- **LDAPفیلترها :**فیلترهای LDAP برای جستجوی خاص در دایرکتوری استفاده می‌شوند.
- **LDIF:**فرمت Lightweight Directory Interchange Format برای نمایش داده‌های دایرکتوری به صورت متنی است.
- **احراز هویت :**برای اتصال به دایرکتوریLDAP ، معمولاً به یک DN و کلمه عبور معتبر نیاز دارید.
- **امنیت :**هنگام استفاده ازLDAP ، به امنیت ارتباط و احراز هویت توجه کنید.

دستور useradd

دستور useradd در سیستم عامل‌های Unix و مشتقات آن مانند Linux برای ایجاد یک کاربر جدید در سیستم استفاده می‌شود. با استفاده از این دستور، می‌توانید یک حساب کاربری جدید ایجاد کرده و تنظیمات مربوط به آن کاربر را مشخص کنید.

گزینه های دستور:

- c COMMENT توضیحات مربوط به کاربر را مشخص می‌کند.
- d HOME_DIR مسیر پوشه خانگی کاربر را مشخص می‌کند.
- e EXPIRATION_DATE تاریخ انقضای حساب کاربری را مشخص می‌کند.
- f INACTIVE_TIME تعداد روزهایی که حساب کاربری قبل از غیرفعال شدن می‌تواند بدون استفاده بماند را مشخص می‌کند.
- g GROUP_NAME گروه پیش‌فرض کاربر را مشخص می‌کند.
- G GROUP_NAMES لیستی از گروه‌های ثانویه که کاربر عضو آنها خواهد بود را مشخص می‌کند.
- m به جای ایجاد پوشه خانگی، فقط اطلاعات مربوط به کاربر را در فایل‌های سیستم ایجاد می‌کند.
- M از ایجاد پوشه خانگی و کپی کردن فایل‌های پیش‌فرض به آن جلوگیری می‌کند.
- n LOGIN_NAME نام ورود به سیستم کاربر را مشخص می‌کند.
- r یک حساب کاربری سیستمی ایجاد می‌کند.
- s SHELL پوسته پیش‌فرض کاربر را مشخص می‌کند.
- u UID شناسه کاربری ( )UIDرا مشخص می‌کند.

مثال های دستور:

ایجاد یک حساب کاربری ساده:

```
useradd ali
```

ایجاد یک حساب کاربری با توضیحات:

```
useradd -c “ ”کاربر جدیدali
```

ایجاد یک حساب کاربری با پوشه خانگی در مسیر دلخواه:

```
useradd -d /home/custom/ali ali
```

ایجاد یک حساب کاربری با تاریخ انقضا:

```
useradd -e “2024-12-31” ali
```

ایجاد یک حساب کاربری با گروه پیش‌فرض:

```
useradd -g developers ali
```

ایجاد یک حساب کاربری با عضویت در چند گروه:

```
useradd -G developers,admins ali
```

ایجاد یک حساب کاربری سیستمی:

```
useradd -r apache
```

ایجاد یک حساب کاربری با پوسته پیش‌فرض bash:

```
useradd -s /bin/bash ali
```

ایجاد یک حساب کاربری با شناسه کاربری دلخواه:

```
useradd -u 1000 ali
```

برای توضیحات بیشتر در مورد هر گزینه، می‌توانید به man page دستور useradd مراجعه کنید:

```
man 8 useradd
```

مثال‌های بیشتر با گزینه‌های دیگر دستور useradd

ایجاد یک حساب کاربری با UID و GID دلخواه:

```
useradd -u 1001 -g 1002 ali
```

ایجاد یک حساب کاربری که پس از ۳۰ روز بدون استفاده غیرفعال شود:

```
useradd -f 30 ali
```

ایجاد یک حساب کاربری با نام ورود به سیستم متفاوت از نام کاربری:

```
useradd -n alireza ali
```

ایجاد یک حساب کاربری بدون ایجاد پوشه خانگی:

```
useradd -M ali
```

ایجاد یک حساب کاربری با پوسته zsh:

```
useradd -s /bin/zsh ali
```

ایجاد یک حساب کاربری با حقوق sudo:

```
useradd ali
usermod -aG sudo ali
```

ایجاد یک حساب کاربری با محدودیت در تعداد جلسات همزمان:

```
useradd -l 2 ali
```

ایجاد یک حساب کاربری با تاریخ انقضای رمز عبور:

```
useradd -e “2024-12-31” -P ali
```

ایجاد یک حساب کاربری با رمز عبور رمزگذاری شده:

```
useradd -p $(openssl passwd -1 “mypassword”) ali
```

ایجاد یک حساب کاربری با استفاده از فایل پیکربندی:

```
useradd -f /etc/useradd.conf ali
```

دستور adduser

همه ما در مورد محبوب ترین دستور به نام ‘ useradd ‘ یا ‘ adduser ‘ در linux آگاهیم . در linux یک دستور useraddیک ابزار سطح پایین است که برای اضافه کردن / ایجاد حساب های کاربری در linux و سایر system عامل های مشابه یونیکس استفاده می شودadduser .بسیار شبیه به دستور useraddاست ، در بعضی توزیع های linux ، دستور useraddممکن است با نسخه های مختلف دیگرتفاوت داشته باشد. هنگام اجرای دستور “ useradd “ در ترمینال linux، موارد زیر را انجام می دهد :

۱. این ویرایش/etc/passwd ، /etc/ shadow ،/etc/group و /etc/gshadow که برای حساب کاربری تازه ایجاد شده است .
۲. ایجاد یک فهرست home برای کاربر جدید ایجاد می کند .
۳. مجوز ها و مالکیت ها را به فهرست خانگی تنظیم می کند .

دستور adduserو addgroupبرای اضافه کردن یک کاربر و گروه به system به ترتیب با توجه به پیکربندی پیش فرض در فایل /etc/adduser.confاستفاده می شود.

مثال های دستور:

ایجاد یک حساب کاربری ساده:

```
adduser ali
```

ایجاد یک حساب کاربری با توضیحات:

```
adduser -c “ ”کاربر جدیدali
```

ایجاد یک حساب کاربری با پوشه خانگی در مسیر دلخواه:

```
adduser -d /home/custom/ali ali
```

ایجاد یک حساب کاربری با تاریخ انقضا:

```
adduser -e “2024-12-31” ali
```

ایجاد یک حساب کاربری با گروه پیش‌فرض:

```
adduser -g developers ali
```

ایجاد یک حساب کاربری با عضویت در چند گروه:

```
adduser -G developers,admins ali
```

ایجاد یک حساب کاربری سیستمی:

```
adduser -r apache
```

ایجاد یک حساب کاربری با پوسته پیش‌فرض bash:

```
adduser -s /bin/bash ali
```

ایجاد یک حساب کاربری با شناسه کاربری دلخواه:

```
adduser -u 1000 ali
```

برای توضیحات بیشتر در مورد هر گزینه، می‌توانید به man page دستور adduser مراجعه کنید:

```
man 8 adduser
```

نکته:

به طور پیش‌فرض، رمز عبور برای کاربر جدید ایجاد نمی‌شود. برای تعیین رمز عبور، باید از دستور passwd استفاده کنید.

اگر از گزینه –m استفاده کنید، باید مجوزهای پوشه خانگی را به گونه‌ای تنظیم کنید که کاربر بتواند به آن دسترسی داشته باشد.

گزینه‌های اضافی:

- D: به جای ایجاد حساب کاربری جدید، اطلاعات مربوط به کاربر موجود را ویرایش می‌کند.
- H: پوشه خانگی کاربر را به /home/username تغییر نام می‌دهد.
- k SKELETON_DIR: از فایل‌های پیکربندی موجود در SKELETON_DIR برای ایجاد پوشه خانگی کاربر استفاده می‌کند.
- S: یک حساب کاربری سیستمی بدون پوسته ایجاد می‌کند.
- b BACKUP_DIR: اطلاعات مربوط به کاربر را در BACKUP_DIR ذخیره می‌کند.

مثال‌های دستور:

ویرایش اطلاعات مربوط به کاربر موجود:

```
adduser -D ali
```

تغییر نام پوشه خانگی کاربر:

```
adduser -H /home/new_home ali
```

ایجاد یک حساب کاربری سیستمی بدون پوسته:

```
adduser -S apache
```

ذخیره اطلاعات مربوط به کاربر در /etc/adduser.bak:

```
adduser -b /etc/adduser.bak ali
```

دستور passwd

دستور **passwd** در سیستم عامل‌های **Unix** و مشتقات آن مانند **Linux** برای تغییر رمز عبور یک کاربر استفاده می‌شود.

با استفاده از این دستور، کاربر می‌تواند رمز عبور خود را تغییر دهد.

گزینه‌های دستور:

- a رمز عبور را قبل از تغییر آن با استفاده از الگوریتم‌های رمزگذاری مختلف بررسی می‌کند.
- d حساب کاربری را غیرفعال می‌کند.
- e EXPIRATION_DATE تاریخ انقضای رمز عبور را مشخص می‌کند.
- f INACTIVE_TIME تعداد روزهایی که حساب کاربری قبل از غیرفعال شدن می‌تواند بدون استفاده بماند را مشخص می‌کند.
- l حساب کاربری را قفل می‌کند.
- n حداقل طول رمز عبور را مشخص می‌کند.
- r حساب کاربری را فعال می‌کند.
- s وضعیت رمز عبور را نمایش می‌دهد.
- u رمز عبور را به طور تصادفی تولید می‌کند.
- x حداکثر تعداد روزهایی که می‌توان از رمز عبور استفاده کرد را مشخص می‌کند.

مثال های دستور:

تغییر رمز عبور خود:

```
passwd
```

تغییر رمز عبور کاربر دیگر:

```
sudo passwd ali
```

غیرفعال کردن حساب کاربری:

```
sudo passwd -d ali
```

فعال کردن حساب کاربری:

```
sudo passwd -r ali
```

قفل کردن حساب کاربری:

```
sudo passwd -l ali
```

نمایش وضعیت رمز عبور:

```
passwd -s ali
```

تعیین تاریخ انقضای رمز عبور:

```
sudo passwd -e “2024-12-31” ali
```

تعیین حداقل طول رمز عبور:

```
sudo passwd -n 8 ali
```

تولید رمز عبور تصادفی:

```
sudo passwd -u ali
```

دستور htpasswd

شرح دستور

دستور `htpasswd` برای ایجاد و مدیریت فایل‌های پسورد استفاده می‌شود .این فایل‌ها معمولاً برای احراز هویت کاربران در وب‌سرورها و سایر سیستم‌های مبتنی بر وب استفاده می‌شوند .پسوردهایی که در این فایل‌ها ذخیره می‌شوند، به صورت رمزنگاری شده هستند تا امنیت آن‌ها حفظ شود.

ساختار دستور

```
htpasswd [options] passwdfile username
```

- `passwdfile`: نام فایلی که اطلاعات کاربران در آن ذخیره می‌شود.
- `username`: نام کاربری که می‌خواهید اضافه، حذف یا ویرایش کنید.

گزینه‌ها

- `-c`: ایجاد یک فایل پسورد جدید .اگر فایل وجود داشته باشد، خطا می‌دهد.
- `-m`: استفاده از الگوریتم رمزنگاری MD5.
- `-D`: حذف کاربر از فایل پسورد.
- `-b`: پسورد را به صورت مستقیم در خط فرمان وارد کنید.
- `-n`: نمایش رمزنگاری شده پسورد بدون افزودن آن به فایل.

- d: – استفاده از الگوریتم رمزنگاری DES.
- s: – استفاده از الگوریتم رمزنگاری SHA.
- p: – استفاده از الگوریتم رمزنگاری CRYPT.

کاربردها

- ایجاد یک فایل پسورد جدید.
- افزودن کاربر جدید به یک فایل پسورد موجود.
- تغییر پسورد یک کاربر.
- حذف کاربر از یک فایل پسورد.
- نمایش رمزنگاری شده یک پسورد.

مثال های کاربردی

ایجاد فایل پسورد جدید و افزودن کاربر

- **ایجاد فایل پسورد جدید با نام htpasswd و افزودن کاربر ali با پسورد ۱۲۳ با استفاده از الگوریتم MD5:**

```
htpasswd -c -m /etc/passwd/htpasswd ali
```

هنگامی که این دستور را اجرا می‌کنید، از شما خواسته می‌شود که پسورد ۱۲۳ را وارد کنید .پس از تایید، فایل htpasswd ایجاد شده و کاربر ali با پسورد رمزنگاری شده به آن اضافه می‌شود.

ویرایش فایل پسورد موجود

- **تغییر پسورد کاربر ali به password123 با استفاده از الگوریتم SHA و وارد کردن مستقیم پسورد در خط فرمان:**

```
htpasswd -b -s /etc/passwd/htpasswd ali password123
```

- **حذف کاربر heidi از فایل:htpasswd**

```
htpasswd -D /etc/passwd/htpasswd heidi
```

نمایش پسورد رمزنگاری شده

- **نمایش رمزنگاری شده پسورد برای کاربر ali با استفاده از الگوریتم DES بدون افزودن آن به فایل:**

```
htpasswd -n -d ali
```

مثال های پیشرفته‌تر

- **ایجاد فایل پسورد با استفاده از الگوریتم CRYPT و سپس افزودن چندین کاربر:**

```
htpasswd -c -p /etc/passwd/htpasswd user1
htpasswd /etc/passwd/htpasswd user2
htpasswd /etc/passwd/htpasswd user3
```

- **نمایش تمام کاربران و پسوردهای رمزنگاری شده در یک فایل:**

```
for user in $(cut -d: -f1 /etc/passwd); do
```

```
htpasswd -n /etc/passwd/htpasswd $user
done > users_and_passwords.txt
```

توجه :این دستور ممکن است برای سیستم‌های بزرگ بسیار زمان‌بر باشد و همچنین از نظر امنیتی توصیه نمی‌شود، زیرا لیستی از پسوردهای رمزنگاری شده را در یک فایل متنی ذخیره می‌کند.

نکات مهم

- **فایل :htpasswd** معمولاً در یک مکان امن مانند `/etc/passwd` یا یک دایرکتوری خصوصی ذخیره می‌شود.
- **رمزنگاری :**پسوردها به صورت یک‌طرفه رمزنگاری می‌شوند، به این معنی که نمی‌توان پسورد اصلی را از روی رمزنگاری شده آن بازیابی کرد.
- **الگوریتم‌های رمزنگاری :**بهتر است از الگوریتم‌های رمزنگاری قوی مانند SHA استفاده کنید.
- **امنیت :**فایل htpasswd باید به خوبی محافظت شود تا از دسترسی‌های غیرمجاز جلوگیری شود.

کاربردهای رایج

- **حفاظت از دسترسی به صفحات وب :**فایل‌های htpasswd معمولاً در ترکیب با فایل‌های `.htaccess` برای حفاظت از دسترسی به دایرکتوری‌ها یا صفحات خاص در وب‌سرورها استفاده می‌شوند.
- **احراز هویت کاربران در سیستم‌های دیگر :**فایل‌های htpasswd می‌توانند برای احراز هویت کاربران در سیستم‌های دیگر مانند سیستم‌های مدیریت محتوا یا برنامه‌های کاربردی وب استفاده شوند.

دستور usermod

پس از ایجاد حساب های کاربری، در برخی از شرایط که در آن ما نیاز به تغییر ویژگی های یک کاربر موجود مانند تغییر فهرست خانگی کاربر، نام کاربری، پوسته ورود، تاریخ انقضا رمز عبور، و غیره، در جایی که در این صورت دستور 'usermod'استفاده می شود. هنگامی که ما دستور 'usermod' را در ترمینال اجرا می کنیم، فایل های زیر مورد استفاده و تحت تاثیر قرار می گیرند.

۱. - etc / passwd /اطلاعات حساب کاربری
۲. - etc / shadow /اطلاعات حساب امن
۳. - etc / group /اطلاعات حساب گروهی.
۴. - etc / gshadow /اطلاعات حساب امن گروه.
۵. - etc/login.defs/پیکربندی مجموعه ای از رمز عبور سایه

ساختار دستور:

```
usermod [ نام کاربری ] [ گزینه ها
```

گزینه های دستور:

- c- = میتوانیم فیلد comment را برای useraccount اضافه کنیم.
- d- = تغییر فهرست برای هر حساب کاربر موجود.
- e- = با استفاده از این گزینه می توانیم مدت زمان مشخص حساب کاربری را لغو کنیم.
- g- = گروه اصلی را برای یک کاربر تغییر دهید.

- G- = اضافه کردن یک گروه مکمل
- a- = برای اضافه کردن هر گروه از گروه به یک گروه ثانویه.
- l- = برای تغییر نام ورود از hosein به hosein _admin.
- L- = برای قفل کردن حساب کاربری ، بنابراین نمی توانیم از حساب استفاده کنیم.
- m- = انتقال محتویات فهرست خانگی از home موجود به فهرست جدید.
- p- = برای استفاده از رمز عبور رمزگذاری نشده برای رمز عبور جدید
- s- = یک پوسته مشخص برای حساب های جدید ایجاد کنید.
- u- = به UID اختصاص داده شده برای حساب کاربری بین ۰ تا ۹۹۹.
- U- = برای باز کردن حساب های کاربری.این قفل رمز عبور را حذف می کند و به ما اجازه می دهد از حساب کاربری استفاده کنیم.

مثال های دستور:

تغییر توضیحات مربوط به کاربر:

```
usermod -c “ ”کاربر با سابقهali
```

تغییر مسیر پوشه خانگی کاربر:

```
usermod -d /home/custom/ali ali
```

تغییر تاریخ انقضای حساب کاربری:

```
usermod -e “2024-12-31” ali
```

تغییر گروه پیش‌فرض کاربر:

```
usermod -g developers ali
```

تغییر عضویت کاربر در گروه‌ها:

```
usermod -G developers,admins ali
```

تغییر نام ورود به سیستم کاربر:

```
usermod -l alireza ali
```

قفل کردن حساب کاربر:

```
usermod -L ali
```

باز کردن حساب کاربر:

```
usermod -U ali
```

تغییر پوسته پیش‌فرض کاربر:

```
usermod -s /bin/zsh ali
```

تغییر شناسه کاربری:

```
usermod -u 1001 ali
```

تغییر اطلاعات مربوط به کاربر بدون تغییر در پوشه خانگی:

```
usermod -o ali
```

۱. اضافه کردن توضیحات به حساب کاربری:

```
usermod -c “ ”کاربر با سابقهali
```

۲. تغییر فهرست خانه کاربر:

```
usermod -d /home/custom/ali ali
```

۳. لغو انقضای حساب کاربری:

```
usermod -e “” ali
```

۴. تغییر گروه اصلی کاربر:

```
usermod -g developers ali
```

۵. اضافه کردن کاربر به یک گروه ثانویه:

```
usermod -G admins ali
```

۶. اضافه کردن گروه “ ”wheelبه گروه‌های ثانویه کاربر:

```
usermod -aG wheel ali
```

۷. تغییر نام ورود کاربر:

```
usermod -l alireza ali
```

۸. قفل کردن حساب کاربری:

```
usermod -L ali
```

۹. انتقال محتویات پوشه خانگی به مسیر جدید:

```
usermod -m -d /home/new/ali ali
```

۱۰. تعیین رمز عبور رمزگذاری نشده:

```
usermod -p “mypassword” ali
```

۱۱. تعیین پوسته پیش‌فرض zsh برای کاربر:

```
usermod -s /bin/zsh ali
```

دستور chfn

دستور ***chfn*** در لینوکس به شما امکان می دهد نام کاربر و سایر جزئیات کاربر را به راحتی تغییر دهید. *chfn* مخفف Change finger است. اصولاً ، برای تغییر اطلاعات انگشت شما در سیستم لینوکس استفاده می شود. این اطلاعات به طور کلی در پرونده *passwd / etc / ذخیره می شود* که شامل نام اصلی کاربر ، شماره تلفن کار و غیره است.

ساختار دستور:

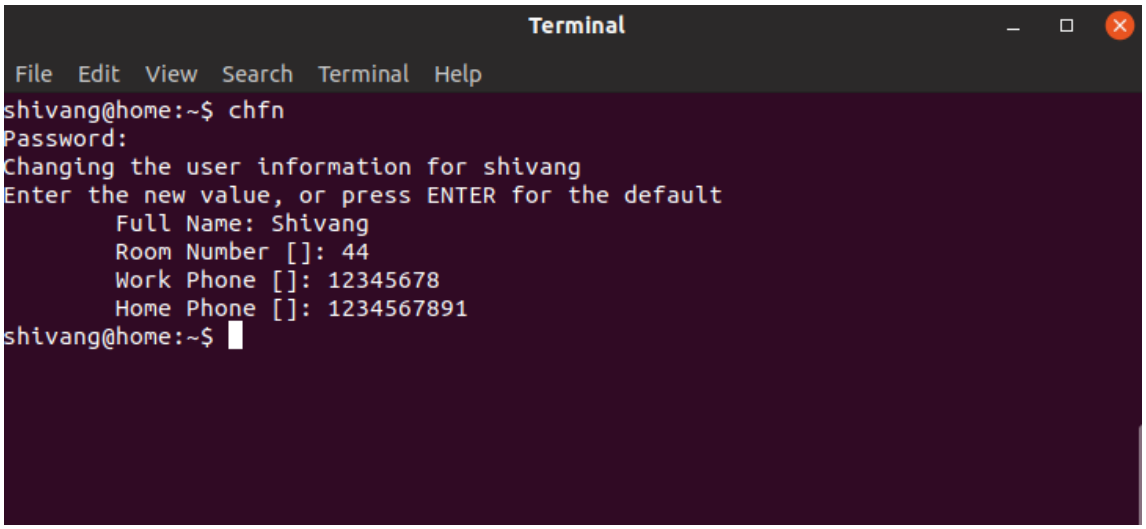
```
chfn [option] [login]
```

گزینه های دستور:

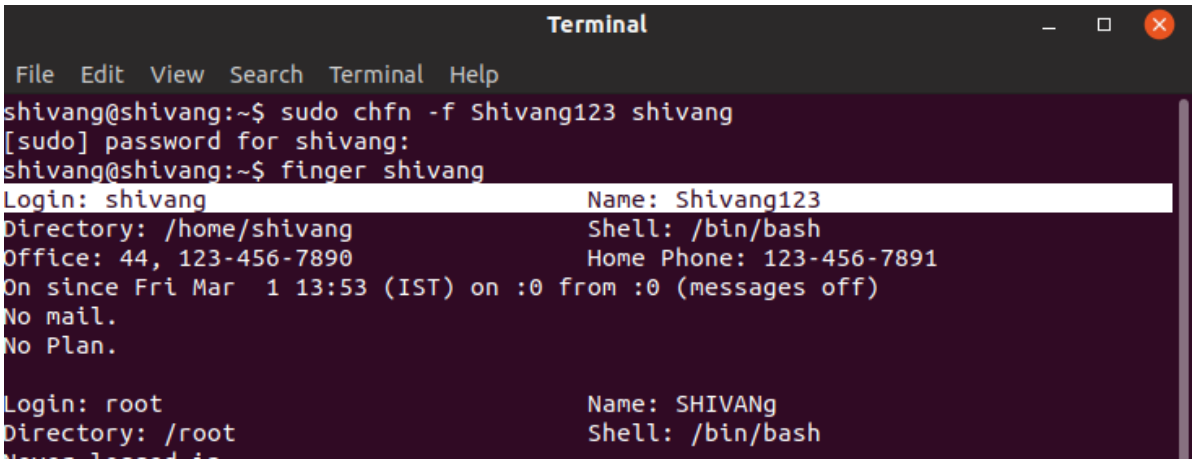
- f FULL_NAME– نام کامل کاربر را تغییر می‌دهد.
- h HOME_PHONE– شماره تلفن منزل کاربر را تغییر می‌دهد.
- o OFFICE_PHONE– شماره تلفن محل کار کاربر را تغییر می‌دهد.
- r ROOM_NUMBER– شماره اتاق کاربر را تغییر می‌دهد.
- u WORK_PHONE– شماره تلفن محل کار کاربر را تغییر می‌دهد.
- l LOGIN_NAME– نام ورود به سیستم کاربر را تغییر می‌دهد.

مثال های دستور:

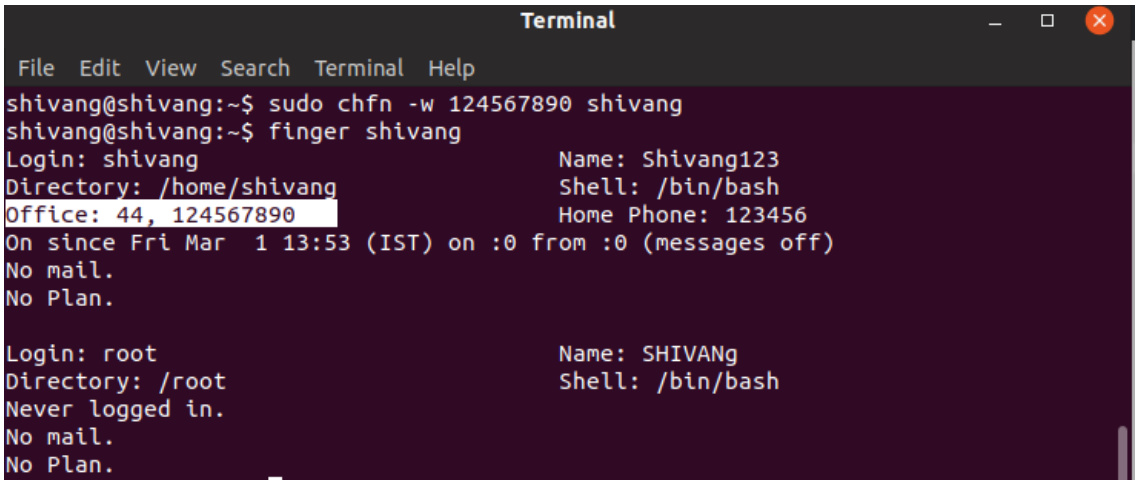
در این مثال ما از دستور پیش فرض *”chfn”* بدون هیچ گزینه ای استفاده کردیم. در این سیستم از خود کاربر می خواهد مقادیر برخی از ویژگیهای اساسی را تغییر دهد.



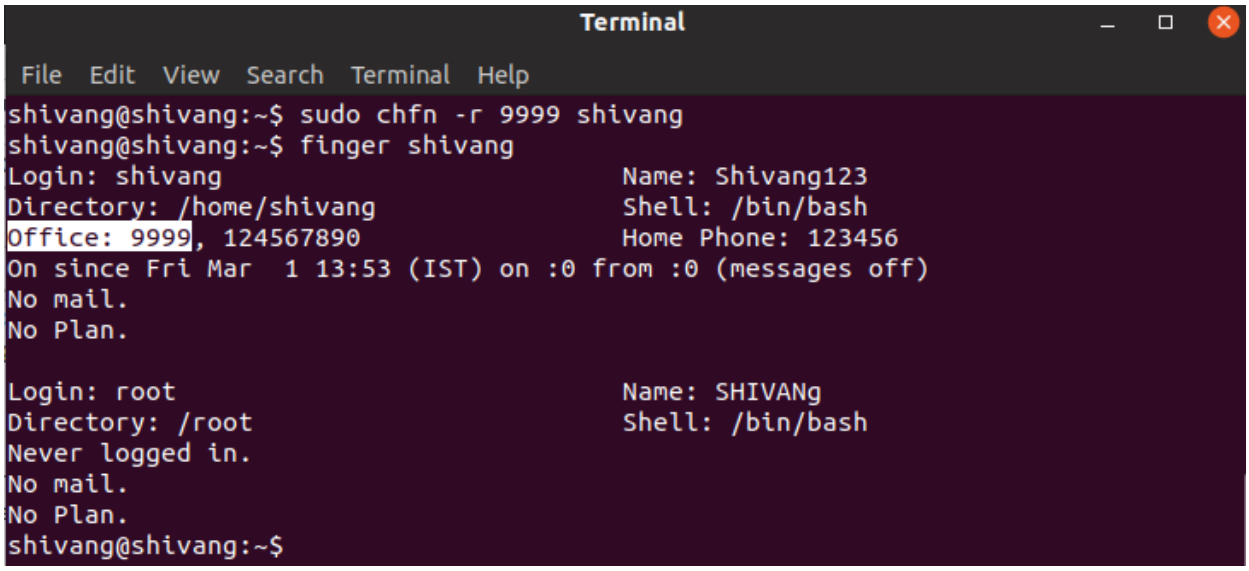
- **full_name -f**: به شما امکان می دهد نام کامل را در حساب تغییر دهید.



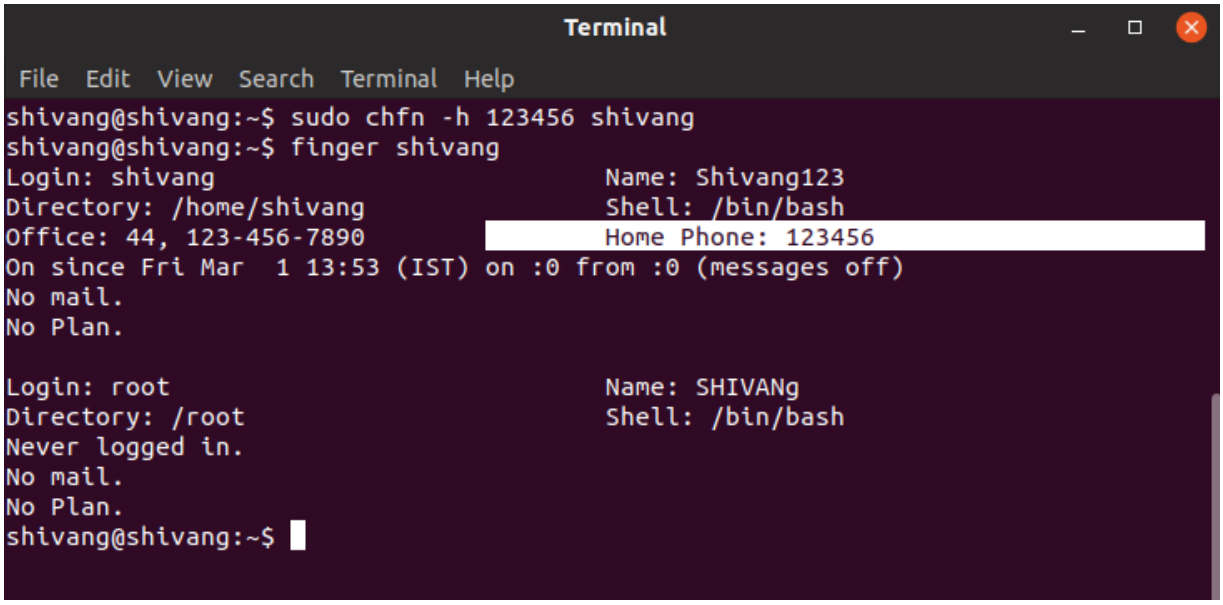
- **work_ph -w**: به شما اجازه می دهد شماره تلفن کار را در حساب تغییر دهید.



- **room_no -r**: به شما اجازه می دهد شماره اتاق موجود در حساب را تغییر دهید.



- **home_ph -h** :به شما اجازه می دهد شماره تلفن خانه را در حساب تغییر دهید.



other -o :به شما اجازه می دهد جزئیات دیگری در حساب تغییر دهید.

مثال های بیشتر:

تغییر نام کامل کاربر:

```
chfn -f “mohammad ali “
```

تغییر شماره تلفن منزل کاربر:

```
chfn -h “021-1234567” ali
```

تغییر شماره تلفن محل کار کاربر:

```
chfn -o “09123456789” ali
```

تغییر شماره اتاق کاربر:

```
chfn -r 101 ali
```

تغییر نام ورود به سیستم کاربر:

```
chfn -l alireza ali
```

نکته:

قبل از استفاده از هر گزینه، به man page دستور chfn مراجعه کنید تا از نحوه عملکرد آن مطمئن شوید.

برخی از گزینه‌ها ممکن است در همه توزیع‌های لینوکس موجود نباشند.

برای تغییر اطلاعات مربوط به گروه کاربری، می‌توانید از دستور chgrp استفاده کنید.

مواردی که در مثال‌ها ذکر نشده:

گزینه – s: برای تغییر پوسته پیش‌فرض کاربر.

گزینه – d: برای تغییر مسیر پوشه خانگی کاربر.

مثال برای گزینه –s:

```
chfn -s /bin/zsh ali
```

مثال برای گزینه –d:

```
chfn -d /home/custom/ali ali
```

نکته:

در مثال‌های بالا، فرض شده است که نام کاربر " "ali است. شما باید نام کاربر را با نام کاربری مورد نظر خودتان جایگزین کنید.

تفاوت chfn و usermod:

دستور chfn فقط برای تغییر اطلاعات مربوط به نام و مشخصات کاربر به کار می‌رود، در حالی که دستور usermod برای تغییر اطلاعات مختلف مربوط به حساب کاربری، مانند گروه‌های کاربری، UID و GID به کار می‌رود.

دستور chfn اطلاعات را در فایل /etc/passwd تغییر می‌دهد، در حالی که دستور usermod اطلاعات را در فایل‌های مختلفی مانند /etc/passwd، /etc/shadow، /etc/group و /etc/gshadow تغییر می‌دهد.

در چه مواقعی از chfn استفاده کنیم؟

اگر فقط می‌خواهید اطلاعات مربوط به نام و مشخصات کاربر را تغییر دهید، از دستور chfn استفاده کنید.

اگر می‌خواهید اطلاعات دیگری مانند گروه‌های کاربری، UID و GID را تغییر دهید، از دستور usermod استفاده کنید.

دستور Finger

دستور **Finger** یک دستور جستجوی اطلاعات کاربر است که جزئیات تمام کاربران وارد شده را ارائه می دهد. این ابزار به طور کلی توسط مدیران سیستم استفاده می شود .این جزئیات مانند نام ورود به سیستم ، نام کاربری ، زمان بیکاری ، زمان ورود و حتی در برخی موارد آدرس ایمیل آنها را در اختیار شما قرار می دهد.

گزینه های دستور:

- l: نمایش اطلاعات کامل
- p: نمایش اطلاعات مربوط به همه کاربران
- q: فقط نمایش نام کاربری و نام کامل
- s: نمایش status

w :نمایش who

این ابزار مشابه ابزار Pinky است اما ابزار Pinky فقط نسخه سبک این ابزار است .برای نصب **finger** از دستورات زیر به عنوان توزیع لینوکس استفاده کنید.

در مورد دبیان / اوبونتو

```
$ sudo apt install finger
```

در موردCentOS / RedHat

```
$ sudo yum install finger
```

در مورد سیستم عاملFedora

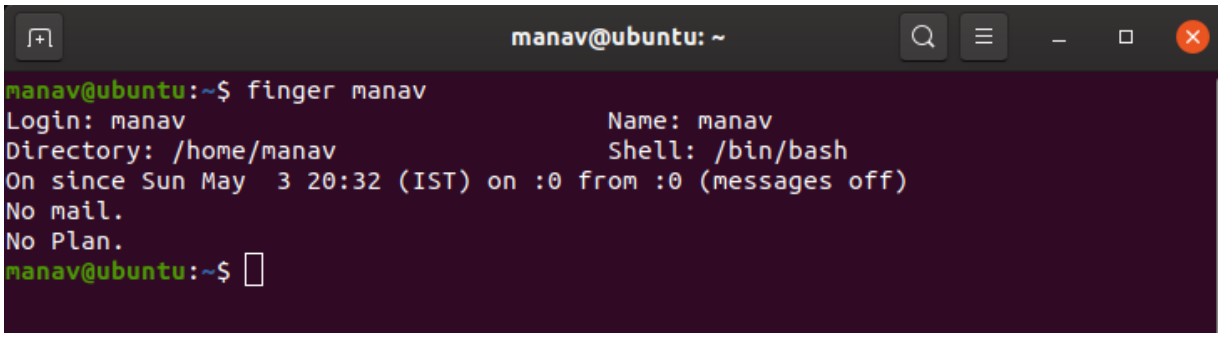
```
$ sudo dnf install finger
```

کار با finger ابزار جستجوی اطلاعات کاربر

۱. finger گذاری یا بدست آوردن اطلاعات کاربر.

```
$ finger  manav
```

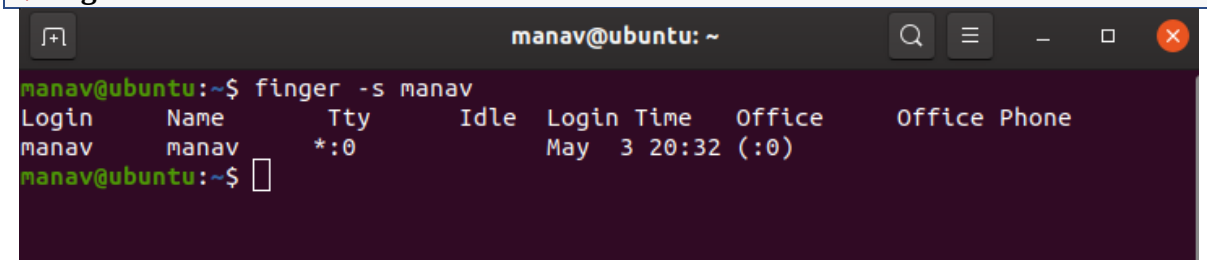
توجه :در اینجا "manav" نام کاربری است.



همانطور که مشاهده می شود ، نام ورود ، نام ، فهرست ، پوسته ، زمان ورود ، ایمیل و برنامه کاربر را نمایش می دهد.

۲. برای به دست آوردن وضعیت بیکار و ورود اطلاعات کاربر.

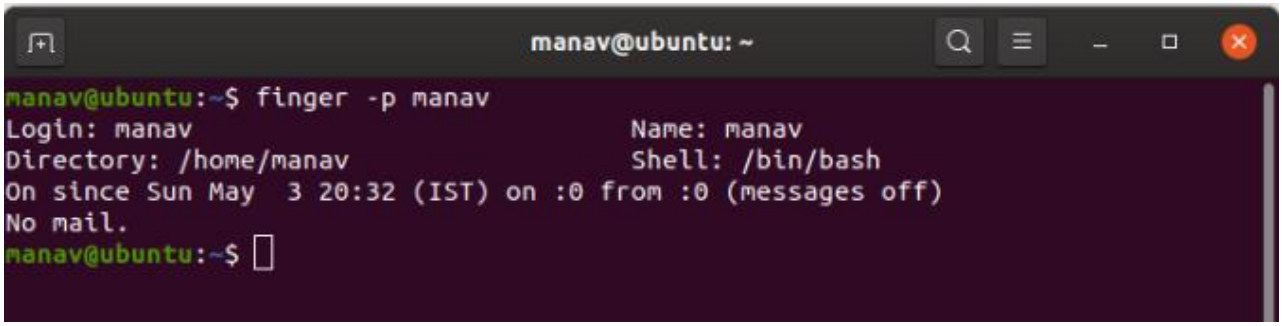
```
$ finger -s manav
```



همانطور که مشاهده می شود ، وضعیت بیکار را به همراه جزئیات کاربر نمایش می دهد.

۳. برای جلوگیری از چاپ کلید PGP ، جزئیات برنامه و پروژه را تهیه کنید

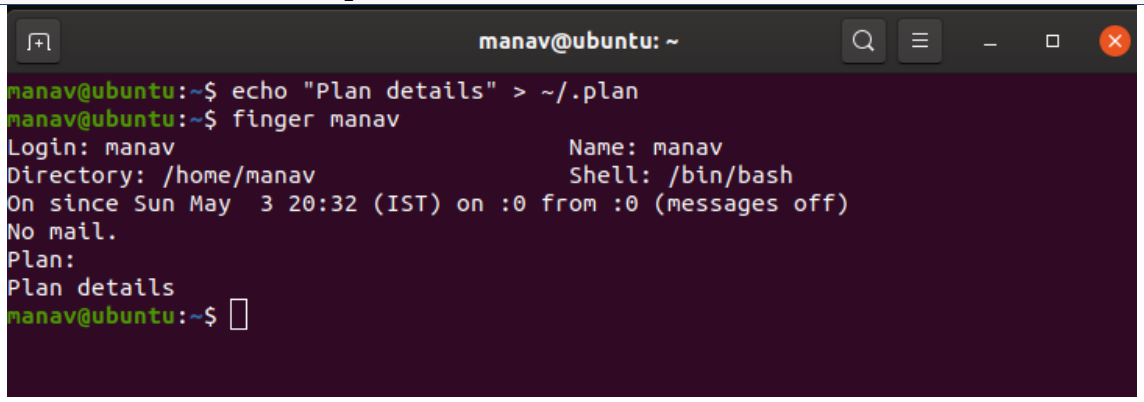
```
$ finger -p manav
```



همانطور که مشاهده می شود ، نام ورود ، نام ، فهرست ، پوسته ، زمان ورود ، ایمیل ، اما طرح ، کلید PGP و پروژه کاربر را نشان نمی دهد.

۴. ایجاد یک برنامه برای کاربر.

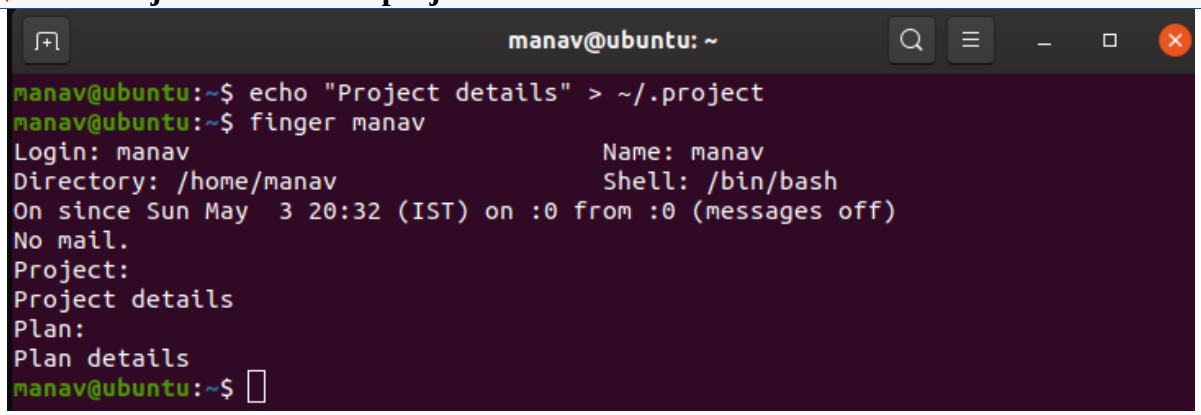
```
$echo “Plan details” > ~/.plan
```



اکنون ، پس از استفاده مجدد از دستور finger ، طرحی را برای کاربر نمایش می دهد.

۵. برای ایجاد یک پروژه برای کاربر.

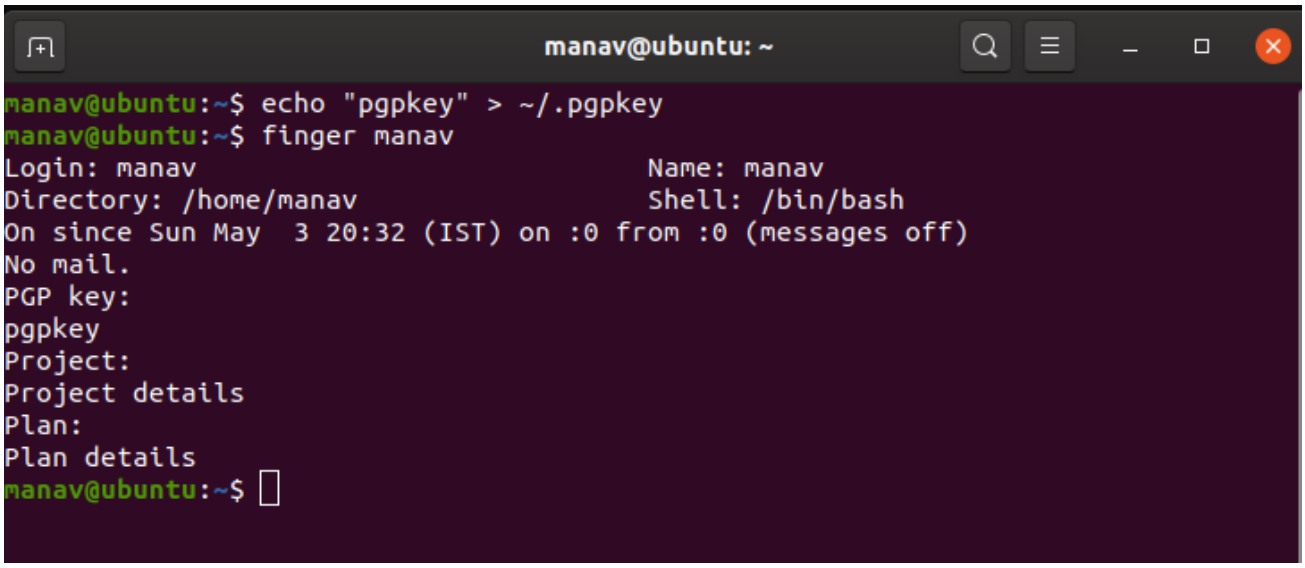
```
$echo “Project details” > ~/.project
```



اکنون ، پس از استفاده مجدد از دستور finger ، آن یک پروژه را برای کاربر نمایش می دهد.

۶. برای ایجاد یک کلید PGP برای کاربر.

```
$ echo “pgpkey”> ~ / .pgpkey
```



اکنون ، پس از استفاده مجدد از دستور finger ، یک کلید PGP برای کاربر نمایش داده می شود.

دستور lchfn

دستور lchfn در لینوکس برای تغییر اطلاعات مربوط به نام کاربری و اطلاعات مربوط به حساب کاربری در سیستم‌های یونیکس استفاده می‌شود. این دستور اطلاعات مربوط به کاربر را در فایل‌های /etc/passwd، /etc/shadow و /etc/login.defs ویرایش می‌کند.

ساختار دستور:

```
lchfn [OPTIONS] username
```

گزینه های دستور:

- GECOS: f نام کامل کاربر را تغییر می‌دهد.
- ROOM_NUMBER: h شماره اتاق کاربر را تغییر می‌دهد.
- OFFICE_NUMBER: شماره تلفن دفتر کاربر را تغییر می‌دهد.
- PHONE_NUMBER: s شماره تلفن منزل کاربر را تغییر می‌دهد.
- USERNAME: u نام کاربری را تغییر می‌دهد.
- WORK_LOCATION: w محل کار کاربر را تغییر می‌دهد.

مثال های دستور:

تغییر نام کامل کاربر:

```
lchfn -f “mohammad ali”
```

تغییر شماره اتاق کاربر:

```
lchfn -h 102 ali
```

تغییر شماره تلفن دفتر کاربر:

```
lchfn -o 123456789 ali
```

تغییر شماره تلفن منزل کاربر:

```
lchfn -s 987654321 ali
```

تغییر نام کاربری:

```
lchfn -u alireza ali
```

تغییر محل کار کاربر:

```
lchfn -w " "شرکت مثالali
```

نکته:

قبل از استفاده از هر گزینه، به man page دستور lchfn مراجعه کنید تا از نحوه عملکرد آن مطمئن شوید.

برخی از گزینه‌ها ممکن است در همه توزیع‌های لینوکس موجود نباشند.

مواردی که در مثال ها ذکر نشده:

- گزینه d– برای غیرفعال کردن حساب کاربری.
- گزینه e– برای فعال کردن حساب کاربری.
- گزینه l– رای قفل کردن حساب کاربری.
- گزینه u– برای باز کردن حساب کاربری.

مثال برای گزینه d:–

```
lchfn -d ali
```

مثال برای گزینه e:–

```
lchfn -e ali
```

مثال برای گزینه l:–

```
lchfn -l ali
```

مثال برای گزینه u:–

```
lchfn -u ali
```

نکته:

در مثال‌های بالا، فرض شده است که نام کاربر " "aliاست. شما باید نام کاربر را با نام کاربری مورد نظر خودتان جایگزین کنید.

دستورPinky

دستور **Pinky** یک دستور جستجوی اطلاعات کاربر است که جزئیات تمام کاربران وارد شده را ارائه می دهد. این ابزار به طور کلی توسط مدیران سیستم استفاده می شود .این همانند ابزار انگشت است اما در بسیاری از لینوکس ها ، سیستم pinky از قبل نصب شده است در حالی که انگشت این کار را نمی کند ، بنابراین در صورتی که حق نصب یک برنامه را ندارید ، این دستور مفید است .برخلاف انگشت ، در رنگ صورتی ، ممکن است اطلاعات مورد علاقه خود را مرتب کنید.

گزینه های دستور:

- **a– نمایش اطلاعات همه کاربران**
- **f– نمایش اطلاعات مختصر**
- **i– نمایش شناسه فرآیند (PID)**
- **l– نمایش اطلاعات طولانی**
- **q– فقط نمایش نام کاربری**
- **t– نمایش زمان ورود**
- **u– نمایش نام واقعی**

نصب pinky ابزار جستجوی اطلاعات کاربر

برای نصب ابزار pinky از دستورات زیر به عنوان توزیع لینوکس استفاده کنید.

در مورد دبیان / اوبونتو

```
$ sudo apt intall pinky
```

در موردCentOS / RedHat

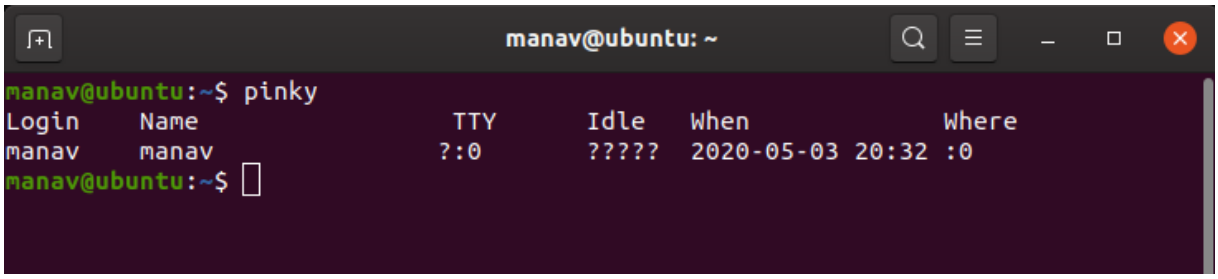
```
$ sudo yum install pinky
```

در مورد سیستم عاملFedora

```
$sudo dnf install pinky
```

کار با pinky ابزار جستجوی اطلاعات کاربر

برای دریافت اطلاعات ورود به سیستم کاربران.

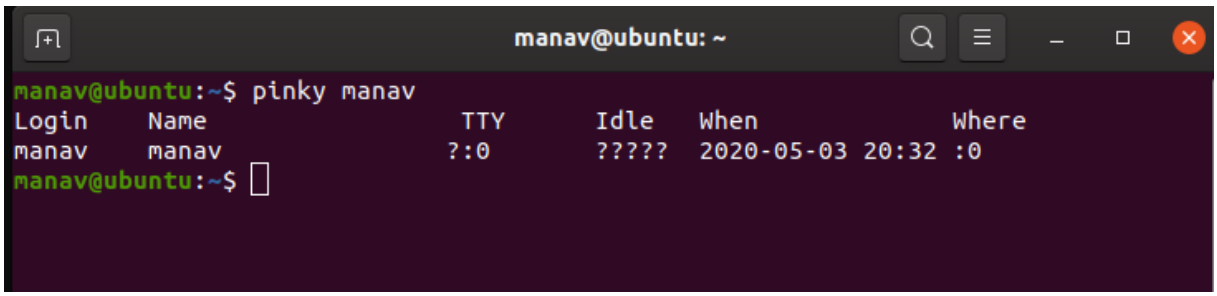


همانطور که مشاهده می شود ، نام ورود ، نام ، زمان بیکاری کاربرانی که به سیستم وارد شده اند را نمایش می دهد.

برای دریافت گزارش یک کاربر

```
$ pinky ali
```

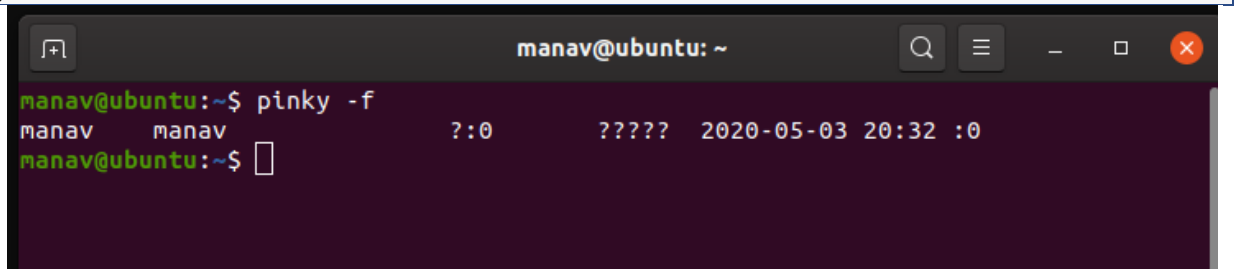
توجه :در اینجا "ali" نام کاربری است.



همانطور که مشاهده می شود ، وضعیت بیکار را به همراه جزئیات کاربر مشخص شده نمایش می دهد.

برای جلوگیری از چاپ عنوان ستون

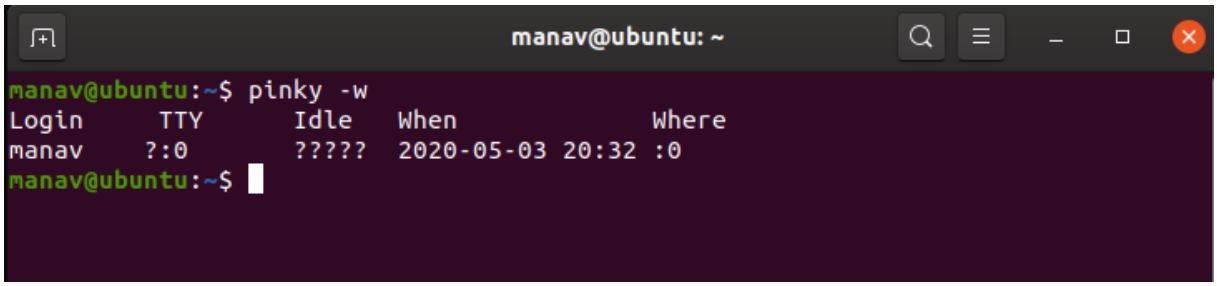
```
pinky -f
```



همانطور که مشاهده می شود ، این فقط اطلاعات کاربران وارد شده بدون عنوان آنها را نمایش می دهد.

برای حذف ستون نام

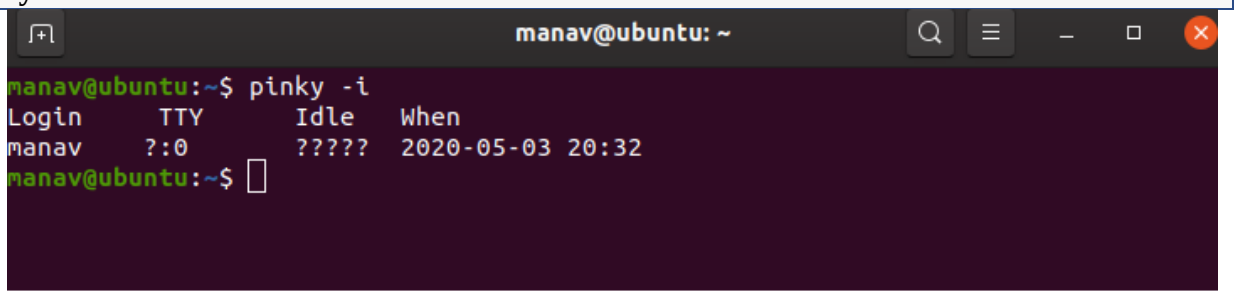
```
$ pinky -w
```



با این کار ستون نام از خروجی حذف می شود.

برای حذف نام و ستون محل

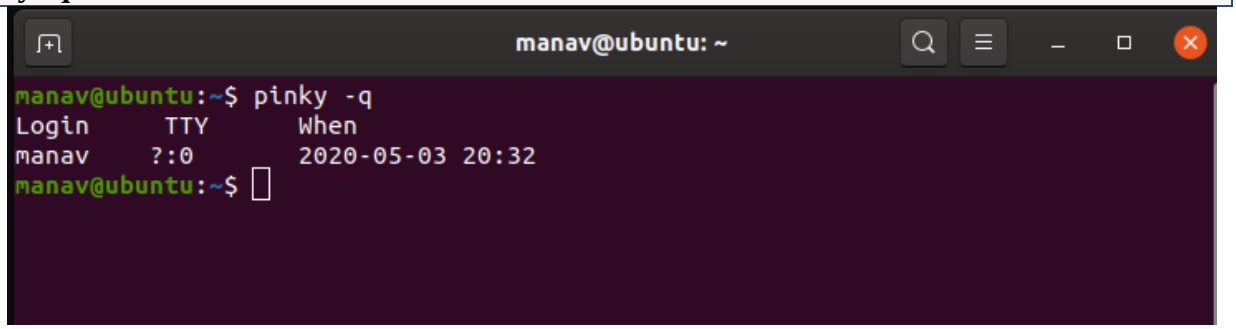
```
$ pinky -i
```



با این کار نام و ستون مکان از خروجی حذف می شود.

برای حذف ستون های نام ، بیکار و مکان

```
$ pinky -q
```

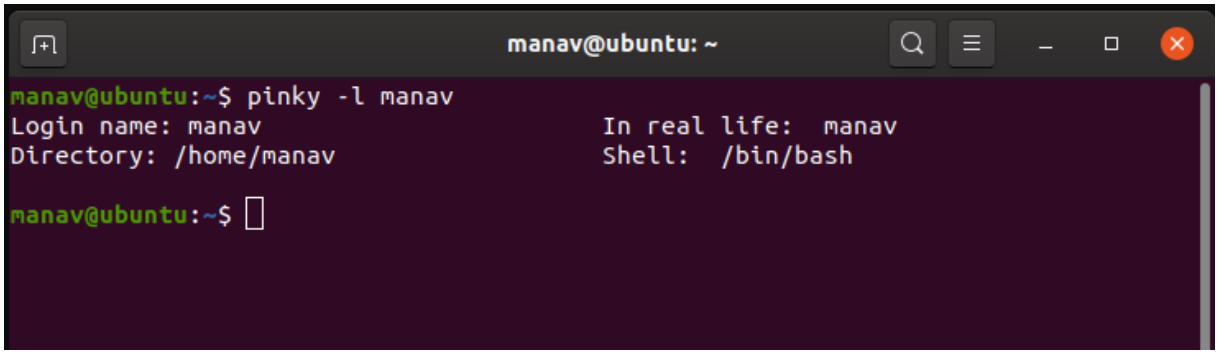


با این کار نام ، بیکار و ستون مکان از خروجی حذف می شود.

برای به دست آوردن جزئیات کامل کاربر (نمای طولانی)

```
$ pinky -l ali
```

توجه :در اینجا "ali" نام کاربری است.

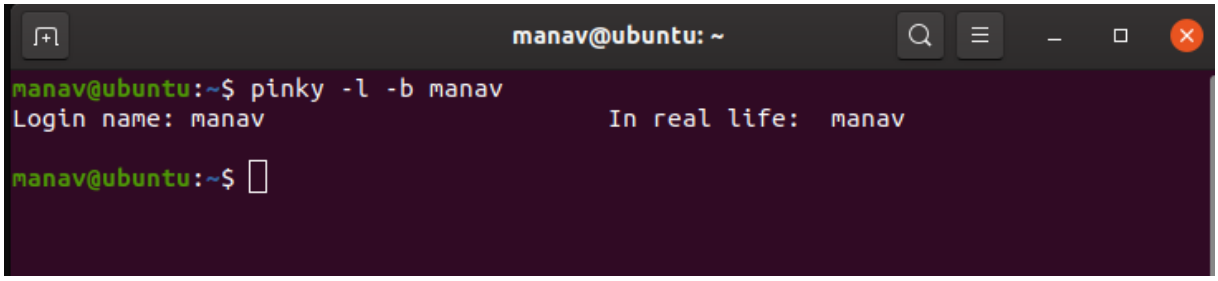


با این کار تمام جزئیات کاربر ارائه می شود.

برای حذف فهرست فهرست و پوسته از نمای طولانی کاربر

```
$ pinky -l -b ali
```

توجه: در اینجا "ali" نام کاربری است.

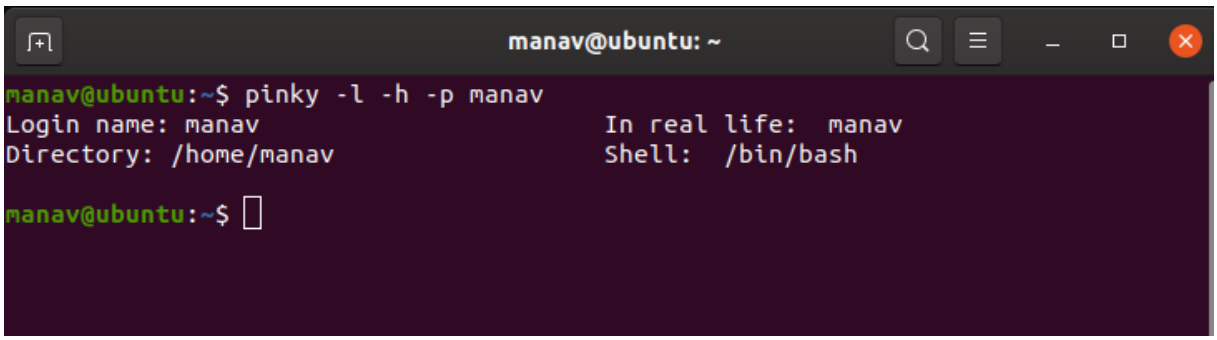


با این کار فهرست و پوسته از نمای جزئیات طولانی کاربر حذف می شوند.

برای حذف طرح و خط فایل پروژه از نمای طولانی یک کاربر.

```
$ pinky -l -h -p ali
```

توجه: در اینجا "ali" نام کاربری است.



این برنامه و خط پروژه را از جزئیات طولانی کاربر حذف می کند

دستور groups

در لینوکس، می‌توان چندین کاربر (کسانی که از سیستم استفاده می‌کنند/استفاده می‌کنند) و گروه‌ها (مجموعه‌های از کاربران) وجود داشته باشد. گروه ها مدیریت کاربران را با همان امنیت و امتیازات دسترسی آسان می کنند. یک کاربر می تواند بخشی از گروه های مختلف باشد. دستور group نام گروه های اولیه و هر گروه مکمل را برای هر نام کاربری داده شده یا فرآیند فعلی را در صورت عدم وجود نام چاپ می‌کند. اگر بیش از یک نام داده شود، نام هر کاربر قبل از لیست گروه های آن کاربر چاپ می شود و نام کاربری با علامت دونقطه از لیست گروه جدا می شود.

ساختار دستور:

```
$groups [username]
```

گزینه های دستور:

- a- لیست تمام گروه‌ها، اعم از گروه‌های اصلی و ثانویه را نمایش می‌دهد.
- l- لیست گروه‌ها را به صورت طولانی نمایش می‌دهد، به طوری که شناسه گروه ( )GIDهر گروه نیز نمایش داده می‌شود.
- o- فقط نام گروه‌ها را نمایش می‌دهد.
- p- فقط شناسه گروه‌ها ( )GIDرا نمایش می‌دهد.

مثال ۱

با نام کاربری ارائه شده است

```
$ groups math
```

در این **مثال**، نام کاربری ریاضی با دستور group ارسال می‌شود و خروجی گروه‌هایی را که کاربر ریاضی در آن‌ها حضور دارد را نشان می‌دهد که با یک دو نقطه از هم جدا شده‌اند.

مثال ۲

هنگامی که هیچ نام کاربری ارسال نمی شود، گروه برای کاربر فعلی نمایش داده می شود:

```
$ groups
```

در اینجا کاربر فعلی ریاضی است. بنابراین وقتی دستور group را بدون آرگومان اجرا می کنیم، گروه هایی را دریافت می کنیم که کاربرجاری در آنها کاربر فعال است .

تغییر گروه اولیه کاربر

برای تنظیم یا تغییر یک گروه اولیه کاربر، از گزینه ' g- ' با دستور usermod استفاده می کنیم.قبل از تغییر گروه اولیه کاربر، ابتدا اطمینان حاصل کنید که گروه فعلی hossein است.

```
# id  hosein
```

uid = 501 (hosein) gid = 502 (hosein) groups = 502 (hosein)

حالا، گروه babinرا به عنوان یک گروه اصلی به user hosein _test تنظیم کنید و تغییرات را تأیید کنید.

```
# usermod -g babin  hosein
# id  hosein
```

uid = 501 (hosein) gid = 502 (babin) ۵۰۲ = گروه (hosein _test)

افزودن گروه به یک کاربر موجود

اگر میخواهید یک گروه جدید به نام ' computer ' به کاربر ' hosein ' اضافه کنید، میتوانید از سویچ Gبا دستور usermodهمانطور که در زیر نشان داده شده است استفاده کنید.

```
# usermod  -g   computer   hosein
# id  hosein
```

قفل حساب کاربری

برای قفل کردن هر حساب کاربر system، ما می توانیم از سویچ ' L- ' قفل استفاده کنیم، بعد از اینکه حساب قفل شد نمی توانیم با استفاده از رمز عبور وارد شوید و قبل از رمز عبور رمز شده در فایل / etc / shadow اضافه شده است، یعنی رمز عبور غیرفعال شده است.

```
# usermod -L hossein
```

حساب کاربری قفل شده را بررسی کنید.

فعال کردن حساب کاربری

گزینه ' U- ' برای باز کردن هر کاربر قفل شده استفاده می شود، این را حذف می کند.

```
# usermod -U hossein
```

تغییر UID و GID یک کاربر

ما می توانیم UID و GID یک کاربر فعلی را تغییر دهیم.برای تغییر به GID جدید ما نیازداریم در یک گروه قرارداشته باشیم.در اینجا یک حساب کاربری به نام ali با GID از 777وجود دارد. در حال حاضر حساب کاربری با UID از 666و GID 777اختصاص داده شود.

قبل از اصلاح، UIDو GID فعلی را بررسی کنید.

```
# id ali
```

تغییر UID وGID

```
# usermod -u 666 -g 777 ali
```

برای تغییرات را چک کنید.

```
# id ali
```

دستورgroupadd

شرح دستور

دستور groupaddابزاری خط فرمان در سیستم‌عامل‌های لینوکس است که برای ایجاد گروه‌های کاربری جدید استفاده می‌شود . گروه‌ها در لینوکس برای مدیریت دسترسی کاربران به منابع مختلف سیستم بکار می‌روند .با استفاده از این دستور، شما می‌توانید یک گروه جدید با نام دلخواه ایجاد کرده و به آن کاربران مختلف را اضافه کنید.

ساختار دستور:

```
groupadd [options] groupname
```

- **groupadd:**نام خود دستور است.
- **options:**گزینه‌های مختلفی که رفتار دستور را تغییر می‌دهند.
- **groupname:**نام گروهی که می‌خواهید ایجاد کنید.

کاربردهای دستور:

- **ایجاد گروه جدید :**اصلی‌ترین کاربرد این دستور، ایجاد یک گروه جدید در سیستم است.
- **تنظیم شناسه گروه :(GID)** با استفاده از گزینه‌های خاص، می‌توانید شناسه گروه را به صورت دستی تنظیم کنید.
- **اضافه کردن توضیحات به گروه :**می‌توانید یک توضیحات کوتاه برای گروه ایجاد شده تعریف کنید.
- **گزینه‌های کامل دستور:**
- **g GID:-**شناسه گروه (GID) را به صورت دستی تنظیم می‌کند.
- **o:-**اجازه می‌دهد که شناسه گروه تکراری ایجاد شود (با احتیاط استفاده شود).
- **r:-**گروه را به عنوان یک گروه سیستم ایجاد می‌کند.
- **s system:-**پوسته پیش‌فرض برای اعضای گروه را تعیین می‌کند.
- **c comment:-**یک توضیحات برای گروه اضافه می‌کند.

مثال های دستور:

- **ایجاد یک گروه جدید با نام developers:**

```
sudo groupadd developers
```

- **ایجاد یک گروه جدید با نام adminو شناسه گروه 1000:**

```
sudo groupadd -g 1000 admin
```

- **ایجاد یک گروه سیستم با نام backupو توضیحات "گروه پشتیبان‌گیری":"**

```
sudo groupadd -r -c "گروه پشتیبان‌گیری" backup
```

نکات مهم:

- برای اجرای دستور `groupadd` معمولاً نیاز به دسترسی روت (sudo) دارید.
- شناسه گروه (GID) یک عدد منحصر به فرد است که به هر گروه اختصاص داده می‌شود و برای شناسایی گروه استفاده می‌شود.
- ایجاد گروه‌های تکراری با شناسه یکسان توصیه نمی‌شود و ممکن است باعث مشکلات امنیتی شود.
- گروه‌های سیستم معمولاً برای سرویس‌های سیستم استفاده می‌شوند و به کاربران عادی توصیه نمی‌شود که گروه‌های سیستم ایجاد کنند.

دستور groupdel

شرح دستور

دستور `groupdel` در سیستم‌عامل‌های مبتنی بر یونیکس، به ویژه لینوکس، برای حذف گروه‌های کاربری استفاده می‌شود .هنگامی که یک گروه حذف می‌شود، تمام مجوزهایی که به آن گروه اختصاص داده شده بود، از بین می‌روند و کاربران عضو آن گروه دیگر به آن مجوزها دسترسی نخواهند داشت.

ساختار دستور:

```
groupdel [options] groupname
```

- **groupdel:** نام خود دستور است.
- **options:** گزینه‌های اختیاری که رفتار دستور را تغییر می‌دهند.
- **groupname:** نام گروهی که می‌خواهید حذف کنید.

کاربردهای دستور:

- **حذف گروه‌های کاربری :** اصلی‌ترین کاربرد این دستور، حذف گروه‌های کاربری است که دیگر نیازی به آن‌ها ندارید.
- **مدیریت مجوزها :** با حذف یک گروه، می‌توانید مجوزهای مرتبط با آن گروه را از سیستم حذف کنید.
- **تمیز کردن سیستم :** حذف گروه‌های بلااستفاده می‌تواند به تمیز و مرتب نگه داشتن سیستم کمک کند.

گزینه‌های رایج:

- **-f, --force:** اگر گروهی که می‌خواهید حذف کنید، هنوز دارای اعضای فعال باشد، با استفاده از این گزینه می‌توانید به زور آن را حذف کنید .با این حال، بهتر است قبل از استفاده از این گزینه، مطمئن شوید که هیچ کاربری به این گروه وابسته نیست.
- **-r, --remove:** این گزینه مشابه گزینه `f-` است و برای حذف اجباری گروه استفاده می‌شود.

مثال‌ها:

- **حذف گروهی به نام: "developers"**

```
sudo groupdel developers
```

- **حذف اجباری گروهی به نام "old_group" حتی اگر دارای اعضای فعال باشد:**

```
sudo groupdel -f old_group
```

نکات مهم:

- **دسترسی روت :** برای اجرای دستور `groupdel` معمولاً نیاز به دسترسی روت (sudo) دارید.
- **مراقبت از وابستگی‌ها :** قبل از حذف یک گروه، مطمئن شوید که هیچ سرویس یا برنامه‌ای به آن گروه وابسته نیست .حذف یک گروه که برنامه‌های دیگر به آن وابسته هستند، می‌تواند باعث بروز مشکلات در سیستم شود.
- **گروه‌های سیستم :** از حذف گروه‌های سیستمی که برای عملکرد صحیح سیستم ضروری هستند، خودداری کنید.

- **کاربران عضو گروه :**قبل از حذف یک گروه، مطمئن شوید که کاربران عضو آن گروه به گروه‌های دیگری منتقل شده‌اند یا مجوزهای لازم را به صورت انفرادی دریافت کرده‌اند.

دستور groupmod

شرح دستور

دستور groupmodدر سیستم‌های عامل مبتنی بر یونیکس و لینوکس برای تغییر مشخصات یک گروه کاربری استفاده می‌شود .این دستور به شما اجازه می‌دهد تا نام گروه، شناسه گروه(GID) ، اعضای گروه و سایر ویژگی‌های مرتبط با گروه را تغییر دهید.

ساختار دستور:

```
groupmod [options] GROUP
```

- **groupmod:**نام خود دستور است.
- **options:**گزینه‌های مختلفی که رفتار دستور را تغییر می‌دهند.
- **GROUP:**نام گروهی که می‌خواهید تغییر دهید.

کاربردهای دستور:

- **تغییر نام گروه :**با استفاده از گزینه n-می‌توان نام یک گروه را تغییر داد.
- **تغییر شناسه گروه :(GID)** با استفاده از گزینه g-می‌توان شناسه گروه را تغییر داد.
- **اضافه کردن یا حذف اعضای گروه :**با استفاده از گزینه a-یا d-می‌توان اعضای جدیدی به گروه اضافه یا اعضای موجود را از گروه حذف کرد.
- **تغییر رمز عبور گروه :**اگر گروهی دارای رمز عبور باشد، می‌توان با استفاده از گزینه p-رمز عبور آن را تغییر داد.

گزینه‌های کامل دستور:

- **-a, --append GID:**اگر اعضای گروه با گزینه U-مشخص شده باشند، آن‌ها را به لیست اعضای موجود اضافه می‌کند.
- **-g, --gid GID:**شناسه گروه را به GID تغییر می‌دهد.
- **-h, --help:**نمایش پیام کمک و خروج.
- **-n, --new-name NEW_GROUP:**نام گروه را از GROUP به NEW_GROUP تغییر می‌دهد.
- **-o, --non-unique:**هنگام استفاده با گزینه g-، اجازه می‌دهد تا شناسه گروه به یک مقدار غیر منحصر به فرد تغییر کند.
- **-p, --password PASSWORD:**رمز عبور رمزنگاری شده را تنظیم می‌کند.
- **-R, --root CHROOT_DIR:**تغییرات را در دایرکتوری CHROOT_DIR اعمال می‌کند.
- **-P, --prefix PREFIX_DIR:**تغییرات را در دایرکتوری PREFIX_DIR اعمال می‌کند.

مثال های دستور:

- **تغییر نام گروه users به:developers**

```
sudo groupmod -n developers users
```

- **تغییر شناسه گروه admins به ۱۰۰۰:**

```
sudo groupmod -g 1000 admins
```

- **اضافه کردن کاربر ali به گروه:developers**

```
sudo usermod -a -G developers ali
```

- **حذف کاربر ali از گروه:developers**

```
sudo usermod -G --remove developers ali
```

نکات مهم:

- برای اجرای دستور groupmodبه دسترسی روت (sudo) نیاز است.

- هنگام تغییر شناسه گروه، مطمئن شوید که شناسه جدید منحصر به فرد باشد و با شناسه گروه‌های دیگر تداخل نداشته باشد.
- تغییر در گروه‌ها می‌تواند بر روی مجوزهای فایل‌ها و دایرکتوری‌ها تأثیر بگذارد.
- برای اطلاعات بیشتر در مورد گزینه‌های دستور `groupmod` می‌توانید از دستور `man groupmod` استفاده کنید.

دستور chpasswd

از دستور **chpasswd** برای تغییر رمز عبور استفاده می شود اگرچه دستور **passwd** نیز می تواند همین کار را انجام دهد .اما این یک بار رمز عبور یک کاربر را تغییر می دهد بنابراین برای چندین کاربر از **chpasswd** استفاده می شود .در شکل زیر استفاده ازدستور *passwd نشان* داده شده است .با استفاده از *passwd* در حال تغییر رمز ورود کاربر مهمان هستیم .در اینجا ابتدا باید رمز ورود کاربر در حال حاضر امضا شده را وارد کنید و سپس رمز ورود هر کاربر دیگری را تغییر دهید .فرد باید دارای امتیازات مدیر باشد.

```
himanshu@him:~$ sudo passwd guest
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
himanshu@him:~$
```

دستور **chpasswd** تعدادی از نام کاربری و رمزعبور را می خواند که با استفاده از ورودی استاندارد یا پرونده توسط colonجدا می شوند و سپس طبق گزینه ها رمزگذاری می شوند.

ساختار دستور:

```
$ chpasswd
user1: user1_password
user2: user2_password
user3: user3_password
```

توجه :پس از اتمام لطفا *ctrl + d را* فشار دهید تا از دستور خارج شوید .به محض فشار دادن *ctrl + d* رمز عبور تغییر می کند.

```
himanshu@him:~$ sudo chpasswd
guest:123
himanshu:76534259
himanshu@him:~$
```

از یک فایل متنی ساده می توان برای ذخیره نام کاربری و رمز عبور و سپس استفاده از آنها برای تغییر رمز عبور استفاده کرد.

```
$ cat> pass.txt
user1: user1_password
user2: user2_password
user3: user3_password
himanshu@him:~$ cat > password.txt
guest:123
himanshu:76534259
himanshu@him:~$
```

سپس این دستور را به ***chpasswd*** ارائه ***دهید***.

```
$ chpasswd <file_name.txt
```

```
himanshu@him:~$ sudo chpasswd < password.txt
himanshu@him:~$
```

گزینه های دستور:

- **-c، –crypt-Method Method_Name** :این گزینه دستور برای روش crypt استفاده می شود .این روش می تواند DES ، MD5 ، SHA256 ، SHA512یا NONE باشد.
- **-e، –encrypted** : برای تهیه رمزهای عبور رمزگذاری شده استفاده می شود.
- **-h، –help** : برای نمایش گزینه های دستور و پیام ها استفاده می شود.
- **-m، –md5** : برای رمزگذاری رمز ورود متن پاک شده با استفاده از الگوریتم MD5 استفاده می شود.
- **-s، –sha-no_of_rounds** :در اینجا می توانید تعداد دورهای الگوریتم دخمه SHA را بدست آورید.

مثال :الگوریتم رمزگذاری همچنین می تواند بر روی رمز عبور اعمال شود.

```
$ chpasswd -c SHA512
user1: user1_password
user2: user2_password
user3: user3_password
```

یا

```
$ chpasswd - md5
user1: user1_password
user2: user2_password
user3: user3_password
```

توجه :هر دو نوع متفاوتی از الگوریتم رمزگذاری هستند

مثال های دستور:

نمایش وضعیت برای همه کاربران:

```
$passwd -a
```

حذف رمز عبور کاربر:

```
$passwd -d [نام کاربری]
```

منقضی کردن گذرواژه یک حساب کاربری بلافاصله:

```
$passwd -e [نام کاربری]
```

نمایش پیام راهنما و خروج:

```
$passwd -h
```

غیرفعال کردن حساب کاربری پس از منقضی شدن رمز عبور برای چند روز:

```
$passwd -i [نام کاربری] [تعداد روزها]
```

تغییر رمز عبور برای توکن های اعتبارسنجی منقضی شده:

```
$passwd -k
```

قفل کردن رمز عبور حساب کاربری:

```
$passwd -l [نام کاربری]
```

حالت سکوت:

```
$passwd -q
```

تغییر رمز عبور در مخزن:

```
$passwd -r [نام مخزن]
```

نمایش اطلاعات وضعیت حساب:

```
$passwd status [نام کاربری]
```

فصل بیست و سوم

فصل بیست و سوم

دستورات ممنوعه

دستورات ترمینال لینوکس قدرتمند هستند. گاهی دستورانی هستند که استفاده ناخواسته یا عمدی آنها توانایی تخریب سیستم را دارند.برای کاربران معمولی لینوکس این دستورات را توصیه نمی کنم:

دستور stress

یک ابزار ساده است که برای اعمال بار بر روی سیستم عامل لینوکس استفاده می شود. این دستور می تواند برای تست کردن منابع سیستم و یا برای تولید بار برای تست سیستم های مانیتورینگ استفاده شود.

گزینه های دستور:

- c CPU_LOAD– میزان بارگذاری CPU را به صورت درصد مشخص می‌کند.
- m MEMORY_SIZE– میزان حافظه‌ای که باید توسط stress اشغال شود را به مگابایت مشخص می‌کند.
- i I/O_LOAD– میزان بارگذاری I/O را به صورت درصد مشخص می‌کند.
- t TIME– مدت زمان تست را به ثانیه مشخص می‌کند.
- n WORKERS– تعداد فرآیندهای stress که باید اجرا شوند را مشخص می‌کند.
- v– اطلاعات بیشتری در مورد تست ارائه می‌دهد.
- V– نسخه stress را نمایش می‌دهد.

مثال های دستور:

نمایش راهنمایی و خروج

```
stress --help
```

نمایش اطلاعات نسخه و خروج

```
stress --version
```

باشید مفصل

```
stress -v
```

باشید ساکت

```
stress -q
```

نمایش چه کاری انجام خواهد شد

```
stress -n
```

خروج بعد از N ثانیه

```
stress -t 10
```

عامل انتظار N میکروثانیه قبل از شروع کار

```
stress --backoff 1000
```

ایجاد N کارگر در حال چرخش بر روی sqrt()

```
stress -c 4
```

ایجاد N کارگر در حال چرخش بر روی sync()

```
stress -i 2
```

ایجاد N کارگر در حال چرخش بر روی malloc()/free()

```
stress -m 1
```

malloc B بایت در هر کارگر) vmپیش فرض ۲۵۶(MB

```
stress --vm-bytes 512M
```

لمس یک بایت هر B بایت (پیش فرض ۴۰۹۶)

```
stress --vm-stride 1024
```

خواب N ثانیه قبل از آزاد کردن (پیش فرض هیچ، ۰ نامحدود است)

```
stress --vm-hang 5
```

redirty حافظه به جای آزاد کردن و reallocating

```
stress --vm-keep
```

ایجاد N کارگر در حال چرخش بر روی write()/unlink()

```
stress -d 3
```

نوشتن B بایت در هر کارگر) hddپیش فرض ۱(GB

```
stress --hdd-bytes 2G
```

ایجاد بارگذاری ۵۰٪ CPU به مدت ۱۰ ثانیه:

```
stress -c 50 -t 10
```

اشغال ۱۰۰ مگابایت حافظه به مدت ۳۰ ثانیه:

```
stress -m 100 -t 30
```

ایجاد بارگذاری ۲۰٪ I/O به مدت ۶۰ ثانیه:

```
stress -i 20 -t 60
```

اجرای ۴ فرآیند stress با بارگذاری ۲۵٪ :CPU

```
stress -c 25 -n 4
```

دستور حذف کامل

```
rm -rf /
```

به منظور حذف همه موارد در مسیر دلخواه، دستور زیر را وارد نمایید. دستور rm که مخفف دستور remove است، با سویچ f در مسیر داده شده همه موارد را حذف خواهد کرد. دستور در به سیستم فایل / همه موارد راحذف خواهد کرد.

```
sudo rm  -rf  /opt/mydata
```

دستور فرمت هارد دیسک

دستور زیر هارد دیسک نوع sata ,pata را فرمت خواهد کرد.

```
 sudo mkfs.ext4    /dev/sda          – Formats a Hard Drive
sudo mkfs.ext4    /dev/hda          – Formats a Hard Drive
```

حذف داده های دیسک با dd

```
sudo dd if=/dev/random  of=/dev/sda      – Writes Junk Onto a Hard Drive
```

حذف فهرست home

```
sudo mv  ~  /dev/null                    – Moves Your Home Directory to a Black Hole
```

فصل بیست و چهارم

فصل بیست و چهارم

رمزگذاری / رمزگشایی درلینوکس

رمزگذاری فرآیند رمزگذاری فایل ها به گونه ای است که فقط افراد مجاز می توانند به آن دسترسی داشته باشند. بشر از اعصار گذشته حتی زمانی که کامپیوترها وجود نداشتند از رمزگذاری استفاده می کرد. در طول جنگ آنها نوعی پیام را منتقل می کردند که فقط قبیله آنها یا کسانی که درگیر آن بودند قادر به درک آن بودند.

دستور Ccrypt

Ccrypt ابزاری خط دستور برای رمزگذاری و رمزگشایی داده ها است .رمزنگاری بر اساس رمز Rijndael است ، همان رمزگذاری مورد استفاده در استاندارد .AES از طرف دیگر ، در استاندارد AES ، از اندازه بلوک ۱۲۸ بیتی استفاده می شود ، در حالی که ccrypt از اندازه بلوک ۲۵۶ بیتی استفاده می کند Ccrypt .معمولاً از پسوند پرونده cpt. برای پرونده های رمزگذاری شده استفاده می کند.

- این یک ابزار سبک است ، نصب و استفاده از این ابزار بسیار آسان است.
- برای غلبه بر کاستی های ابزار مخفی استاندارد یونیکس طراحی شده است.

نصب و راه اندازی

```
$ sudo apt install ccrypt
```

گزینه های دستور :

- **-e ،encrypt–** :رمزگذاری .این حالت پیش فرض است .اگر آرگومان های نام پرونده داده شد ، پرونده ها را رمزگذاری کرده و پسوند cpt. را به نام آنها ضمیمه کنید .
- **-d ،decrypt–** :رمزگشایی .در صورت وجود آرگومان های نام پرونده ، پرونده ها را رمزگشایی کرده و پسوند cpt.را از نام پرونده ها حذف میکند .
- **-c ،cat–** :رمزگشایی یک یا چند پرونده برای خروجی استاندارد .اگر هیچ آرگومان نام پرونده ای داده نشده است ، به عنوان فیلتر رمزگشایی کنید l- .را معنی می کند.
- **-x ،keychange–** :کلید داده های رمزگذاری شده را تغییر دهید .در این حالت ، ccryptبرای دو رمز عبور درخواست می کند: رمز اول و رمز جدید .

برای رمزگذاری پرونده با استفاده از این ابزار ، از نحو زیر استفاده کنید:

```
ccrypt filename
```

برای رمزگذاری یک فایل یا فهرست

```
ccrypt   file_name
```

این رمز عبور را چند بار ارسال می کند و پس از اتمام ، فایل منبع شما را حذف می کند و فایل را با پسوند cpt. ذخیره می کند.

```
inuxpitstpo: /home/linuxpitstop
root@Linuxpitstpo:/home/linuxpitstop# ccrypt impfile
Enter encryption key:
Enter encryption key: (repeat)
root@Linuxpitstpo:/home/linuxpitstop#
```

پرونده منبع حذف شده است و فقط فایل رمزگذاری شده وجود دارد:

```
nuxpitstpo: /home/linuxpitstop
root@Linuxpitstpo:/home/linuxpitstop# ls -la | grep impfile
-rw-r--r--  1 root         root              9012 Jun 26 15:08 impfile.cpt
root@Linuxpitstpo:/home/linuxpitstop#
```

رمزگشایی پرونده باCcrypt

اکنون برای رمزگشایی یک فایل رمزگذاری شده از قبل ، دستور دستور:

```
ccrypt –d encryptedfilename
```

بنابراین ، فایل رمزگذاری شده را رمزگشایی کنید:

```
ccrypt –d impfile.cpt
```

این رمز عبور را می پرسد و پرونده را رمزگشایی می کند.

```
root@Linuxpitstpo:/home/linuxpitstop# ccrypt -d impfile.cpt
Enter decryption key:
root@Linuxpitstpo:/home/linuxpitstop#
```

این یک برنامه کوچک ، بسیار سبک وزن است زیرا تقریباً هیچ منابعی با تقریباً بدون اثر حافظه مصرف نمیکند .نصب و استفاده از آن بسیار آسان است ، این یک برنامه ضروری برای کاربران لینوکس است.

دستور GnuPG

GnuPG مخفف GNU Privacy Guard است و اغلب به عنوان GPG نامیده می شود که مجموعه ای از نرم افزارهای رمزنگاری است. نوشته شده توسط پروژه گنو به زبان برنامه نویسی C. آخرین نسخه پایدار ۲,۰,۲۷ است. در اکثر توزیع‌های لینوکس امروزی، بسته gnupg به‌صورت پیش‌فرض ارائه می‌شود، اگر در صورت نصب نشدن، می‌توانید آن را از مخزن استفاده کنید یا آن را استفاده کنید.

```
$ sudo apt-get install gnupg
# yum install gnupg
```

اکنون فایل tecmint.txt را با استفاده از gpg رمزگذاری کنید. به محض اجرای دستور gpc با گزینه – c رمزگذاری فقط با رمز متقارن) یک فایل texmint.txt.gpg ایجاد می کند. می توانید محتوای فهرست را برای تأیید فهرست کنید.

```
gpg -c ~/Desktop/Tecmint/tecmint.txt
$ ls -l ~/Desktop/Tecmint
```

اکنون، اگر می خواهید فایل رمزگذاری شده فوق را رمزگشایی کنید، می توانید از دستور زیر استفاده کنید، اما قبل از شروع رمزگشایی، ابتدا فایل اصلی یعنی tecmint.txt را حذف می کنیم و فایل رمزگذاری شده tecmint.txt.gpg را دست نخورده می گذاریم.

```
$ rm ~/Desktop/Tecmint/tecmint.txt
$ gpg ~/Desktop/Tecmint/tecmint.txt.gpg
```

دستور cryptdir

دستور cryptdir در لینوکس برای رمزنگاری فهرست‌ها و فایل‌ها به کمک الگوریتم رمزنگاری مشخص استفاده می‌شود. این دستور به کاربر امکان می‌دهد تا اطلاعات حساس خود را در فهرست‌های رمزنگاری شده ذخیره کرده و از دسترسی غیرمجاز جلوگیری کند.

گزینه‌های دستور:

- e, --encrypt: رمزنگاری فهرست یا فایل.
- d, --decrypt: رمزگشایی فهرست یا فایل.
- p, --passphrase <passphrase>: تعیین عبارت عبور برای رمزنگاری یا رمزگشایی.
- r, --recursive: اعمال عملیات رمزنگاری یا رمزگشایی به صورت بازگشتی بر روی زیرفهرست‌ها.
- v, --verbose: نمایش جزئیات بیشتر در حین عملیات.
- c رمزگذاری یک پوشه.
- d رمزگشایی یک پوشه.
- f رمز عبور را از فایل متنی می‌خواند.
- k KEY رمز عبور را به صورت مستقیم وارد می‌کند.
- m MODE حالت رمزگذاری را مشخص می‌کند.
- n بدون رمزگذاری، فقط نام پوشه را تغییر می‌دهد.
- p پیشرفت رمزگذاری/رمزگشایی را به صورت درصدی نمایش می‌دهد.
- v جزئیات بیشتری از عملیات را نمایش می‌دهد.

مثال‌های دستور:

رمزنگاری یک فهرست:

```
cryptdir -e /path/to/directory -p mypassphrase
```

رمزگشایی یک فایل:

```
cryptdir -d /path/to/file.enc -p mypassphrase
```

رمزنگاری یک فهرست و زیرفهرست‌های آن:

```
cryptdir -e /path/to/directory -p mypassphrase -r
```

رمزنگاری با نمایش جزئیات بیشتر:

```
cryptdir -e /path/to/file -p mypassphrase -v
```

دستور flock

دستور flock در لینوکس برای قفل گذاری بر روی یک فایل یا فهرست به منظور جلوگیری از همزمانی دسترسی به آن استفاده می‌شود. این دستور به کاربر امکان می‌دهد تا یک قفل را بر روی یک فایل تعیین کرده و عملیات خواندن یا نوشتن را برای دیگر فرآیندها مسدود کند تا از ایجاد تداخل در دسترسی به فایل جلوگیری شود.

ساختار دستور:

```
flock [-sxon] [-w timeout] lockfile [-c] command...
flock [-sxon] [-w timeout] lockdir [-c] command...
flock [-sxun] [-w timeout] fd
```

گزینه‌های دستور:

- x, --exclusive: قفل انحصاری بر روی فایل یا فهرست.
- s, --shared: قفل به اشتراک گذاری شده بر روی فایل یا فهرست.

- -n, --nonblock: اختیاری، اگر قفل انحصاری قابل اختیار نبود، دستور فشرده می‌شود.
- -u, --unlock: برای آزاد کردن قفل گذاری شده توسط فرآیند فعلی.
- -n قفل‌گذاری غیرمسدود کننده. اگر فایل قفل شده باشد، flock بلافاصله با کد خطا باز می‌گردد.
- -s قفل اشتراکی. چندین فرآیند می‌توانند به طور همزمان قفل اشتراکی را روی یک فایل داشته باشند.
- -x قفل انحصاری. فقط یک فرآیند می‌تواند در یک زمان قفل انحصاری را روی یک فایل داشته باشد.
- -w منتظر بمانید تا قفل آزاد شود قبل از ادامه.
- -o فقط قفل را بررسی کنید، بدون قفل کردن فایل.
- -u قفل را از روی فایل حذف کنید.

مثال های دستور:

قفل انحصاری بر روی یک فایل:

```
flock -x /path/to/file command_to_execute
```

قفل به اشتراک گذاری شده بر روی یک فایل:

```
flock -s /path/to/file command_to_execute
```

استفاده از قفل به صورت غیرقابل مسدود کردن:

```
flock -xn /path/to/file command_to_execute
```

آزاد کردن قفل گذاری شده توسط فرآیند:

```
flock -u /path/to/file
```

فصل بیست و پنجم

فصل بیست و پنجم

ماژول‌های کرنل

ماژول‌های کرنل قطعه‌هایی از کد هستند که می‌توانند در صورت تقاضا در هسته بارگیری و تخلیه شوند. آنها عملکرد هسته را بدون نیاز به راه اندازی مجدد سیستم گسترش می دهند. برای ایجاد یک ماژول هسته، می توانید راهنمای برنامه نویسی ماژول کرنل لینوکس را بخوانید. یک ماژول را می توان به صورت داخلی یا قابل بارگذاری پیکربندی کرد. برای بارگذاری یا حذف پویا یک ماژول، باید به عنوان یک ماژول قابل بارگذاری در پیکربندی هسته پیکربندی شود (بنابراین خط مربوط به ماژول حرف M را نشان می دهد). ماژول های لینوکس توده هایی از کد هستند که می توانند در هر نقطه پس از بوت شدن سیستم به صورت پویا به هسته پیوند داده شوند. آنها را می توان از هسته جدا کرد و زمانی که دیگر مورد نیاز نیست حذف شد. ماژول‌های هسته لینوکس عمدتاً درایورهای دستگاه، درایورهای شبه دستگاه مانند درایورهای شبکه یا سیستم‌های فایل هستند. ماژول های لینوکس (هسته) ماژول های نرم افزاری کامپایل شده ای هستند که می توانند به صورت پویا توسط هسته بارگذاری شوند (با استفاده از یک برنامه کاربردی توسط مدیر سیستم) که ویژگی های هسته لینوکس را گسترش می دهد. این شامل (اما محدود به) ویژگی هایی مانند:

۱. **سیستم های فایل**
۲. **درایورهای دستگاه**
۳. **پروتکل های شبکه**
٤. **ویژگی های فایروال** (iptables و iptables6).
۵. **درایورهای** GPU
٦. **به دست آوردن اطلاعات**

ماژول ها در /usr/lib/modules/kernel_release/ ذخیره می شوند. می توانید از دستور uname -r برای دریافت نسخه فعلی هسته خود استفاده کنید.

توجه: نام ماژول ها اغلب از زیرخط (_) یا خط تیره (-) استفاده می کنند. با این حال، این نمادها هنگام استفاده از دستور modprobe و در فایل های پیکربندی در /etc/modprobe.d/ / قابل تعویض هستند.

دستور lsmod

/z دستور lsmod برای نمایش وضعیت ماژول ها در هسته لینوکس استفاده می شود .این منجر به لیستی از ماژول های بارگذاری شده می شود lsmod .یک برنامه پیش پا افتاده است که به خوبی محتویات قالب بندی شده را/proc/modules/ نشان می دهد که ماژول های هسته در حال حاضر بارگیری می شوند.

ساختار دستور:

lsmod

گزینه های دستور:

- a– نمایش تمام ماژول‌ها، حتی ماژول‌های هسته داخلی.
- b– نمایش نام کامل درایور برای هر ماژول.
- c– نمایش تعداد دفعات استفاده از هر ماژول.
- d– نمایش وابستگی‌های هر ماژول.
- e– نمایش اطلاعات مربوط به ماژول‌های هسته خارجی.
- f– نمایش اطلاعات مربوط به ماژول‌های هسته بارگذاری شده در یک فایل.

- m– فقط ماژول‌هایی را که با نام یا شناسه مطابقت دارند نمایش می‌دهد.
- n– نمایش نام نمادین برای هر ماژول.
- p– نمایش شناسه فرآیندی که هر ماژول توسط آن استفاده می‌شود.
- q– فقط نام ماژول‌ها را نمایش می‌دهد.
- s– نمایش اندازه هر ماژول.
- t– نمایش نوع هر ماژول.
- v– نمایش اطلاعات نسخه برای هر ماژول.
- x– نمایش اطلاعات مربوط به ماژول‌های هسته بارگذاری شده در یک سیستم دیگر.

به عنوان مثال :اجرای **lsmod** در خط دستور به لیست تمام ماژول های کرنل فعال است.

```
lsmod
```

خروجی دستور:

```
narru@narru-HP-Pavilion-g4-Notebook-PC:~$ lsmod
Module                  Size  Used by
ccm                    20480  6
rfcomm                 77824  0
bnep                   20480  2
kvm_amd                86016  0
kvm                   598016  1 kvm_amd
irqbypass              16384  1 kvm
```

فرمت خروجی دستور: سه ستون در خروجی وجود دارد.

- ابتدا ستون برای " **نام** " پرونده ماژول استفاده می شود .یا می توانید بگویید
- نام = نام پرونده ماژول – پسوند پرونده ماژول (به عنوان مثال o. یا ko).
- دوم ، ستون برای " **اندازه** " ماژول در بایت استفاده می شود.
- ستون سوم در خروجی برای " **استفاده شده** "است .در اصل برای نمایش تعداد نمونه های ماژول استفاده می شود.

دستور modinfo

از دستور **modinfo** در سیستم لینوکس برای نمایش اطلاعات مربوط به ماژول هسته لینوکس استفاده می شود .این دستور اطلاعات را از ماژول های هسته لینوکس استخراج شده در خط فرمان استخراج می کند .اگر نام ماژول نام پرونده نباشد ، به طور پیش فرض فهرست **/ lib / modules / kernel-version** جستجو می شود **modinfo**. می تواند از هر یک از معماری هسته لینوکس ماژول ها را درک کند.

ساختار دستور:

```
modinfo [-0] [-F field] [-k kernel] [modulename|filename...]
```

مثال های دستور:

```
modinfo bluetooth
```

```
algoscale@algoscale-Lenovo-ideapad-330-15IKB:~$ modinfo bluetooth
filename:       /lib/modules/4.15.0-45-generic/kernel/net/bluetooth/bluetooth.ko
alias:          net-pf-31
license:        GPL
version:        2.22
description:    Bluetooth Core ver 2.22
author:         Marcel Holtmann <marcel@holtmann.org>
srcversion:     CF600F16CF3D8912A42EC69
depends:        ecdh_generic
retpoline:      Y
intree:         Y
name:           bluetooth
vermagic:       4.15.0-45-generic SMP mod_unload
parm:           disable_esco:Disable eSCO connection creation (bool)
parm:           disable_ertm:Disable enhanced retransmission mode (bool)
```

گزینه های دستور:

دستور modinfo با گزینه help: این دستورالعمل عمومی **modinfo** را به همراه گزینه های مختلف چاپ می کند و در مورد هر گزینه توضیح مختصری می دهد.

```
algoscale@algoscale-Lenovo-ideapad-330-15IKB:~$ modinfo --help
Usage:
        modinfo [options] filename [args]
Options:
        -a, --author                Print only 'author'
        -d, --description           Print only 'description'
        -l, --license               Print only 'license'
        -p, --parameters            Print only 'parm'
        -n, --filename              Print only 'filename'
        -0, --null                  Use \0 instead of \n
        -F, --field=FIELD           Print only provided FIELD
        -k, --set-version=VERSION   Use VERSION instead of `uname -r`
        -b, --basedir=DIR           Use DIR as filesystem root for /lib/modules
        -V, --version               Show version
        -h, --help                  Show this help
```

modinfo -V: این گزینه اطلاعات نسخه دستور **modinfo** را می دهد.

modinfo -V

```
algoscale@algoscale-Lenovo-ideapad-330-15IKB:~$ modinfo -V
kmod version 22
-XZ -ZLIB -EXPERIMENTAL
```

modinfo –F: این گزینه فقط این مقدار فیلد را چاپ می کند ، یکی در هر خط .نام زمینه ها به حروف کوچک حساس نیستند .زمینه های متداول که ممکن است شامل نویسنده ، شرح ، مجوز ، پارم ، وابسته و نام مستعار باشد .احتمالاً چندین پارک ، نام مستعار وجود دارد و به زمینه ها بستگی دارد .لیست پرونده های فیلد خاص نام پرونده ماژول است.

modinfo -F modulename

modinfo –b: این گزینه فهرست ریشه ماژول ها است.

modinfo -b modulename

modinfo –k: این گزینه اطلاعات مربوط به هسته دیگری غیر از هسته در حال اجرا را فراهم می کند .این گزینه به ویژه برای توزیع هایی که نیاز به استخراج اطلاعات از مجموعه جدیدی از ماژول های هسته دارند ، بسیار مفید است .به عنوان مثال ، برای پیدا کردن کدام پرونده های میان افزار مورد نیاز ماژول های مختلف درون هسته جدید است که قبل از راه اندازی نیاز به ایجاد تصویر **initrd** /**initramfs** دارید.

modinfo -k modulename

modinfo -0: این گزینه به جای خط جدید ، از نویسه صفر ASCII برای جدا کردن مقادیر استفاده می کند .ثابت شده است که این گزینه برای اسکریپت ها مفید است زیرا یک خط جدید می تواند از لحاظ تئوری در داخل یک قسمت ظاهر شود.

مثال :

modinfo bluetooth -0

```
algoscale@algoscale-Lenovo-ideapad-330-15IKB:~$ modinfo bluetooth -0
filename:       /lib/modules/4.15.0-45-generic/kernel/net/bluetooth/bluetooth
ion=CF600F16CF3D8912A42EC69depends=ecdh_genericretpoline=Yintree=Yname=blueto
```

modinfo -a –autor ،-d –description ،-l –license ،-p –parameters ،-n

این میانبرهایی است که برای **آرگومان نویسنده ، توضیحات ، مجوز ، پارم** و **نام پرونده** برای ایجاد انتقال استفاده می شود از modutils قدیمی modinfo آسان است.

مثال :

modinfo bluetooth -a

```
algoscale@algoscale-Lenovo-ideapad-330-15IKB:~$ modinfo bluetooth -a
Marcel Holtmann <marcel@holtmann.org>
```

modinfo bluetooth -n

```
algoscale@algoscale-Lenovo-ideapad-330-15IKB:~$ modinfo bluetooth -n
/lib/modules/4.15.0-45-generic/kernel/net/bluetooth/bluetooth.ko
```

modinfo bluetooth -d

```
algoscale@algoscale-Lenovo-ideapad-330-15IKB:~$ modinfo bluetooth -d
Bluetooth Core ver 2.22
```

modinfo bluetooth -l

```
algoscale@algoscale-Lenovo-ideapad-330-15IKB:~$ modinfo bluetooth -l
GPL
```

modinfo bluetooth -p

```
algoscale@algoscale-Lenovo-ideapad-330-15IKB:~$ modinfo bluetooth -p
disable_esco:Disable eSCO connection creation (bool)
disable_ertm:Disable enhanced retransmission mode (bool)
```

دستور insmod

از دستور insmod در سیستم های لینوکس برای قرار دادن ماژول ها در هسته استفاده می شود .لینوکس یک سیستم عامل است که به کاربر اجازه می دهد ماژول های هسته را در زمان اجرا برای گسترش ویژگی های هسته بارگیری کند LKM .ها (ماژول های هسته قابل بارگیری) معمولاً برای افزودن پشتیبانی از سخت افزار جدید (به عنوان درایورهای دستگاه) و / یا سیستم فایل یا برای افزودن تماس های سیستمی استفاده می شوند .این دستور در اینجا فایل شی **k هسته (.ko)** را به همراه چند گزینه اضافی با / بدون استدلال وارد هسته می کند.

ساختار دستور:

insmod [گزینه های ماژول] [نام پرونده ...]

گزینه های دستور:

- f– به طور پیش‌فرض، اگر ماژول مورد نظر قبلاً بارگذاری شده باشد، insmod پیغامی مبنی بر تکراری بودن بارگذاری ماژول نمایش می‌دهد. با استفاده از این گزینه، این پیغام نمایش داده نمی‌شود.
- k– فقط ماژول‌های هسته با نسخه مشخص شده در این گزینه را بارگذاری می‌کند.
- m– ماژول‌های مورد نیاز ماژول اصلی را به صورت خودکار بارگذاری می‌کند.

- p- ماژول را به صورت دائمی بارگذاری می‌کند. این کار باعث می‌شود که ماژول حتی بعد از ریستارت سیستم نیز بارگذاری باقی بماند.
- r- ماژول را از حافظه حذف می‌کند.
- s- لیست ماژول‌های بارگذاری شده در حال حاضر را نمایش می‌دهد.
- v- اطلاعات مربوط به ماژول مورد نظر را نمایش می‌دهد.

در صورت عدم عبور از گزینه ها ، نام پرونده یا استدلال ها ، دستور insmod خطایی ایجاد می کند .بنابراین ، وقتی از گزینه h- استفاده می کنیم ، آن را به عنوان نحوی کلی همراه با گزینه های مختلفی می توان با دستور insmod استفاده کرد.

insmod -h

```
mukkesh@mex:~$ insmod -h
Usage:
        insmod [options] filename [args]
Options:
        -V, --version     show version
        -h, --help        show this help
mukkesh@mex:~$
```

insmod command with version option :از این دستور برای مشاهده نسخه دستور insmod استفاده می شود.

insmod -V

```
mukkesh@mex:~$ insmod -V
kmod version 24
-XZ -ZLIB -EXPERIMENTAL
mukkesh@mex:~$
```

مثال های دستور:

insmod + name file:

از این دستور برای درج فایل (.ko) LKM در هسته لینوکس استفاده می شود .فهرست کاری به فهرست دارای پرونده LKM تغییر می کند و سپس دستور اجرا می شود .برای اجرای این دستورالعمل به ریشه نیاز است.

توجه :فایل LKM استفاده شده در اینجا در دسک تاپ (که فهرست کاری است) وجود دارد و هیچ استدلالی را نمی گیرد و پیامی را به سیستم سیستم چاپ می کند .با استفاده از دستور dmesg می توان این پیام را مشاهده کرد.

sudo insmod ali.ko

مثال :

insmod + file directory + name file:

مانند دستور قبلی ، insmodهنگامی که فهرست پرونده را به همراه نام فایل LKM تعیین می کنید ، کار می کند .این در زیر نشان داده شده است.

مثال :

sudo insmod /home/hossein/Desktop/ali.ko

عبور پارامترها به ماژول با استفاده از insmod: پرونده های LKM می توانند پارامترها / آرگومان هایی را از طریق دستور insmodبه آنها منتقل کنند .این استدلال ها اساساً به عنوان ورودی برای LKM عمل می کنند .بسته به نحوه نوشتن LKM ، از استدلال ها استفاده می شود.

توجه LKM :استفاده شده در اینجا پارامتر رشته "کاربر" را می گیرد و پیامی را چاپ می کند که شامل پارامتر عبور شده باشد.

مثال :

```
sudo insmod ali.ko user = "ali"
```

دستور rmmod

از دستور **rmmod** در سیستم لینوکس برای حذف ماژول از هسته استفاده می شود .بیشتر کاربران هنوز به جای استفاده از **rmmod** از **modprobe** با گزینه **-r** استفاده می کنند.

ساختار دستور:

```
rmmod [-f] [-s] [-v] [modulename]
```

مثال های دستور:

```
rmmod  bluetooth
```

```
algoscale@algoscale-Lenovo-ideapad-330-15IKB:~$ rmmod bluetooth
rmmod: ERROR: Module bluetooth is in use by: btrtl btintel btbcm bnep btusb rfcomm
```

گزینه های دستور:

- f– این گزینه به rmmod اجازه می‌دهد تا ماژول‌هایی را که در حال استفاده هستند حذف کند. با احتیاط از این گزینه استفاده کنید، زیرا می‌تواند باعث ناپایداری سیستم شود.
- m– این گزینه به rmmod اجازه می‌دهد تا ماژول‌هایی را که به طور خودکار توسط udev بارگیری می‌شوند حذف کند.
- n– این گزینه به rmmod اجازه می‌دهد تا بدون حذف ماژول، فقط اطلاعات مربوط به آن را نمایش دهد.
- q– این گزینه rmmod را از چاپ پیام‌های غیرضروری باز می‌دارد.
- r– این گزینه معادل دستور modprobe -r است.
- t– این گزینه به rmmod اجازه می‌دهد تا فقط ماژول‌هایی را که به طور مستقیم توسط کاربر بارگیری شده‌اند حذف کند.
- v– این گزینه rmmod را به چاپ اطلاعات اضافی در مورد عملیات حذف ماژول وادار می‌کند.
- V– این گزینه نسخه rmmod را نمایش می‌دهد.

- **گزینه rmmod command with help:** دستور کلی rmmod را به همراه گزینه های مختلفی که می توان با دستور rmmod استفاده کرد چاپ می کند و همچنین توضیح مختصری در مورد هر گزینه ارائه می دهد.

```
algoscale@algoscale-Lenovo-ideapad-330-15IKB:~$ rmmod --help
Usage:
        rmmod [options] modulename ...
Options:
        -f, --force       forces a module unload and may crash your
                          machine. This requires Forced Module Removal
                          option in your kernel. DANGEROUS
        -s, --syslog      print to syslog, not stderr
        -v, --verbose     enables more messages
        -V, --version     show version
        -h, --help        show this help
```

- **rmmod –v :** این گزینه پیام هایی را در مورد آنچه برنامه انجام می دهد چاپ می کند .معمولاً rmmod فقط در صورت بروز اشتباه پیام ها را چاپ می کند.

مثال :

rmmod -v bluetooth

```
algoscale@algoscale-Lenovo-ideapad-330-15IKB:~$ rmmod -v bluetooth
rmmod: ERROR: Module bluetooth is in use by: btrtl btintel btbcm bnep btusb rfcomm
```

rmmod –f :این گزینه می تواند بسیار خطرناک باشد .

.با استفاده از این گزینه می توانید ماژول های مشخص شده را که در حال استفاده هستند یا برای حذف شدن یا علامت گذاری نشده ایمن نیستند ، حذف کنید.

مثال :

sudo rmmod -f bluetooth

```
algoscale@algoscale-Lenovo-ideapad-330-15IKB:~$ sudo rmmod -f bluetooth
rmmod: ERROR: ../libkmod/libkmod-module.c:793 kmod_module_remove_module() could not remove 'bluetooth': Resource temporarily unavailable
rmmod: ERROR: could not remove module bluetooth: Resource temporarily unavailable
```

- **rmmod –s** :این گزینه قرار است به جای خطای استاندارد ، خطاهایی را به syslog ارسال کند.

مثال :

rmmod - bluetooth

```
algoscale@algoscale-Lenovo-ideapad-330-15IKB:~$ rmmod -s bluetooth
btrtl btintel btbcm bnep btusb rfcomm
```

- **rmmod –V** :این گزینه نسخه برنامه را نشان می دهد و سپس خارج می شود.

rmmod -V

```
algoscale@algoscale-Lenovo-ideapad-330-15IKB:~$ rmmod -V
kmod version 22
-XZ -ZLIB -EXPERIMENTAL
```

دستور modprobe

مقدمه:

در سیستم‌عامل لینوکس، هسته (kernel) نقش اصلی را در مدیریت سخت‌افزار و نرم‌افزار ایفا می‌کند .هسته لینوکس به صورت ماژولار طراحی شده است و این بدان معناست که قابلیت‌های آن را می‌توان با بارگذاری ماژول‌های مختلف گسترش داد .دستور modprobe ابزاری قدرتمند برای مدیریت این ماژول‌ها است.

ساختار دستور:

modprobe [options] module_name

- **modprobe:** نام خود دستور است.
- **options:** گزینه‌های مختلفی که می‌توان برای کنترل نحوه بارگذاری ماژول استفاده کرد.
- **module_name:** نام ماژولی که می‌خواهیم بارگذاری یا حذف کنیم.

کاربردهای دستور modprobe:

- **بارگذاری ماژول‌ها :** برای افزودن قابلیت‌های جدید به هسته، از این دستور برای بارگذاری ماژول‌های مورد نیاز استفاده می‌شود.
- **حذف ماژول‌ها :** برای حذف ماژول‌هایی که دیگر مورد نیاز نیستند، از این دستور استفاده می‌شود.
- **نمایش اطلاعات در مورد ماژول‌ها :** با استفاده از گزینه‌های مختلف، می‌توان اطلاعاتی در مورد ماژول‌های بارگذاری شده به دست آورد.

گزینه‌های مهم دستور modprobe:

- **-v, --verbose:** اطلاعات بیشتری در مورد فرآیند بارگذاری یا حذف ماژول نمایش می‌دهد.
- **-r, --remove:** برای حذف یک ماژول استفاده می‌شود.

- **-l, --list:** لیستی از همه ماژول‌های بارگذاری شده را نمایش می‌دهد.
- **-c, --config:** برای نمایش پیکربندی یک ماژول استفاده می‌شود.
- **-a, --alias:** برای ایجاد یک نام مستعار برای یک ماژول استفاده می‌شود.

مثال‌ها:

- **بارگذاری ماژول `snd_hda_intel`:**

```
sudo modprobe snd_hda_intel
```

این دستور معمولاً برای فعال کردن کارت صدای داخلی در سیستم‌های مبتنی بر اینتل استفاده می‌شود.

- **حذف ماژول `snd_hda_intel`:**

```
sudo modprobe -r snd_hda_intel
```

- **نمایش لیست همه ماژول‌های بارگذاری شده:**

```
sudo lsmod
```

- **نمایش اطلاعات در مورد ماژول `nvidia`:**

```
sudo modprobe -c nvidia
```

دستور depmod

دستور (Dep endency Mod ules) depmod برای تولید لیستی از توصیف وابستگی ماژول های هسته و پرونده های نقشه مربوط به آن استفاده می شود. این ماژول های هسته را در فهرست **/ lib / modules / kernel-release** تجزیه و تحلیل می کند و یک پرونده وابستگی مانند **Makefile** به نام **modules.dep** ایجاد می کندبر اساس نمادهای موجود در مجموعه ماژول ها. این ماژولها به طور کلی از فهرستهای مشخص شده در پرونده پیکربندی یا ذکر شده در خط فرمان گرفته می شوند. سپس وقتی پشته ماژول ها به طور خودکار با modprobe اضافه و حذف می شوند ، هیچ ماژولی بدون سایر ماژول های مرتبط مورد نیاز نیست. به طور همزمان ، یک نقشه مرتبط ایجاد می کند که شناسه های سخت افزاری و ماژول های مربوطه را با هم مرتبط می کند و آنها را برای استفاده در زیرساخت hotplug کنترل می کند. هسته لینوکس برای ارسال اولیه داده ها برای ماژول های خود ، به ترتیب مناسب برای بارگیری ، به depmod و modprobe متکی است .

ساختار دستور:

```
Syntax:
depmod [ -a ] [ -b basedir ] [ -e ] [ -F System.map ] [ -n ] [ -v ] [ version ] [ -A ]
    [-n] [-v] [-A] [-P prefix] [-w] [version]

depmod [-e] [-E Module.symvers] [-F System.map] [-m] [-n] [-v] [-P prefix]
    [-w] [version] [filename...][Tex]
```

دستور depmod همچنین لیستی از نمادها را که توسط ماژول ها در پرونده با نام **modules.symbols** ارائه می شوند ، ایجاد می کند. دستور depmod با استفاده از نسخه هش شده باینری خود ، **modules.symbols.bin** لیستی از نمادها را در ماژول ها ایجاد می کند. اگر ماژول ها نام دستگاه های منحصر به فردی (devname) را تهیه کنند که باید توسط یک ابزار خاص مانند udevجمع شود ، در خروجی پرونده ای به نام **modules.devname** ذخیره می کند.

کاربردهای دستور:

بهبود زمان بوت با ایجاد وابستگی های ماژول، هسته می تواند فقط ماژول های مورد نیاز را در زمان بوت بارگیری کند، که به نوبه خود زمان بوت را بهبود می بخشد.

حل مشکل وابستگی اگر ماژولی به ماژول دیگری وابسته باشد که هنوز بارگیری نشده است، هسته می تواند به طور خودکار ماژول وابسته را قبل از بارگیری ماژول اصلی بارگیری کند.

گزینه‌های دستور:

- a- وابستگی های تمام ماژول های موجود در دایرکتوری /lib/modules/<نسخه_هسته> را دوباره محاسبه می کند.
- n- وابستگی های ماژول های جدید را بدون آپدیت وابستگی های ماژول های موجود محاسبه می کند.
- F- وابستگی های ماژول های مشخص شده در فایل را به جای دایرکتوری /lib/modules/<نسخه_هسته> محاسبه می کند.
- r- وابستگی های ماژول های مشخص شده را حذف می کند.
- v- اطلاعات مربوط به وابستگی های ماژول را به صورت verbose چاپ می کند.
- h- لیست کاملی از گزینه های موجود را چاپ می کند.

مثال های دستور:

ایجاد وابستگی های تمام ماژول ها:

```
depmod -a
```

ایجاد وابستگی های ماژول های جدید:

```
depmod -n
```

محاسبه وابستگی های ماژول های موجود در فایل modules.conf:

```
depmod -F modules.conf
```

حذف وابستگی های ماژول foo:

```
depmod -r foo
```

چاپ اطلاعات مربوط به وابستگی های ماژول bar:

```
depmod -v bar
```

چاپ لیست کاملی از گزینه های موجود:

```
depmod -h
```

اضافه کردن پارامترهای کرنل

اگر می خواهید پارامترهای هسته را در طول زمان بوت در یک سیستم اضافه کنید، قالب پیکربندی GRUB را در آدرس زیر ویرایش کنید:

```
/etc/default/grub
```

نحوه اضافه کردن پارامتر هسته به شکل name=value در :

```
GRUB_CMDLINE_LINUX_DEFAULT
```

فایل زیر را در یک ادیتور دلخواه با دسترسی ریشه باز کنید.

```
sudo geany /etc/default/grub
```

سپس در فایل باز شده در خطی که مقدار

GRUB_CMDLINE_LINUX_DEFAULT=“quiet splash name=value”

وجود دارد. نام و مقدار مورد نظر خود را که پارامتر کرنل محسوب میشود وارد نمایید.

acpi=off OR noacpi

این پارامتر کل سیستم(ACPI پیکربندی پیشرفته و رابط برق) را غیرفعال می کند. این ممکن است بسیار مفید باشد، برای مثال ، اگر رایانه شما از ACPI پشتیبانی نمی کند یا اگر فکر می کنید اجرای ACPI ممکن است مشکلاتی ایجاد کند (به عنوان مثال راه اندازی مجدد تصادفی یا قفل شدن سیستم)

apm=off OR noapm
Disable the Advanced Power Management.
ro - mount the root disk read only.
quiet - don’t display console messages
splash - show a graphical “splash” screen while booting.
GRUB_CMDLINE_LINUX_DEFAULT=“quiet splash”

Nomodeset

تنظیم حالت هسته را غیرفعال کنید.

Disable Kernel mode setting.

سپس فایل را ذخیره نمایید و دستورزیر را برای اعمال تغییر وارد نمایید:

sudo update-grub

دستور grub-install

مقدمه

GRUB مخفف **GRand Unified Bootloader** است و یکی از پرکاربردترین بوت لودرهای سیستم‌های لینوکسی است .این ابزار وظیفه دارد که پس از روشن شدن سیستم، سیستم عامل مورد نظر را شناسایی و بارگذاری کند .دستور `grub-install` برای نصب و پیکربندی GRUB روی دیسک سخت یا دستگاه بوت قابل استفاده است.

ساختار دستور

sudo grub-install <device>

- **sudo:** برای اجرای دستور با دسترسی ادمین استفاده می‌شود.
- **grub-install:** نام خود دستور است.
- **<device>:** دستگاهی که می‌خواهیم GRUB روی آن نصب شود .معمولاً یک پارتیشن از هارد دیسک است.

کاربردهای دستور

- **نصب GRUB:** برای نصب GRUB روی دیسک سخت و پیکربندی آن برای بوت شدن از سیستم عامل مورد نظر.
- **به‌روزرسانی GRUB:** برای بروزرسانی نسخه GRUB به نسخه جدیدتر.
- **پیکربندی GRUB:** برای تغییر تنظیمات بوت مانند ترتیب بوت شدن سیستم عامل‌ها یا تنظیمات پیشرفته‌تر.

گزینه‌های دستور

- **--root-directory=/path/to/root:** مشخص کردن مسیر ریشه سیستم عامل.
- **--boot-directory=/path/to/boot:** مشخص کردن مسیر دایرکتوری بوت.

- **--target=<target>:** مشخص کردن هدف نصب (مثلاً bios ، efi).
- **--recheck:** مجدداً بررسی کردن دستگاه‌ها و پارتیشن‌ها قبل از نصب.
- **--force:** نصب GRUB را حتی اگر سیستم از قبل بوت لودر دیگری داشته باشد، اجبار می‌کند.

مثال‌های دستور

- **نصب GRUB روی دیسک /dev/sda:**

```
sudo grub-install /dev/sda
```

- **نصب GRUB روی دیسک /dev/sda و مشخص کردن مسیر ریشه:**

```
sudo grub-install --root-directory=/boot /dev/sda
```

- **نصب GRUB روی سیستم UEFI:**

```
sudo grub-install --target=x86_64-efi --efi-directory=/boot/efi /dev/sda
```

نکات مهم

- **دستگاه هدف :** اطمینان حاصل کنید که دستگاهی که مشخص می‌کنید، دستگاهی است که سیستم عامل شما روی آن نصب شده است.
- **حالت UEFI یا BIOS:** اگر سیستم شما از UEFI استفاده می‌کند، باید از گزینه‌های مناسب برای نصب GRUB در محیط UEFI استفاده کنید.
- **بک‌آپ :** قبل از انجام هرگونه تغییری در بوت لودر، از سیستم خود بک‌آپ بگیرید.
- **مشکلات بوت :** اگر پس از نصب GRUB با مشکل بوت شدن مواجه شدید، می‌توانید از یک لایو سی‌دی یا لایو USB برای تعمیر GRUB استفاده کنید.

چه زمانی از grub-install استفاده می‌شود؟

- **نصب سیستم عامل جدید :** هنگام نصب یک سیستم عامل جدید، معمولاً این دستور برای نصب GRUB و پیکربندی آن برای بوت شدن سیستم عامل جدید استفاده می‌شود.
- **تغییر ترتیب بوت :** اگر می‌خواهید ترتیب بوت شدن سیستم عامل‌ها را تغییر دهید، می‌توانید از این دستور برای پیکربندی مجدد GRUB استفاده کنید.
- **رفع مشکلات بوت :** در صورت بروز مشکل در بوت شدن سیستم، ممکن است نیاز به استفاده از این دستور برای تعمیر یا بازگرداندن GRUB باشد.

دستور grub-mkconfig

معرفی

دستور `grub-mkconfig` یک ابزار خط فرمان قدرتمند است که برای تولید فایل پیکربندی اصلی بوت لودر GRUB مورد استفاده قرار می‌گیرد (GRand Unified Bootloader). یک بوت لودر محبوب در سیستم‌های لینوکسی است که وظیفه بارگذاری سیستم‌عامل را بر عهده دارد. فایل پیکربندی GRUB شامل اطلاعاتی در مورد سیستم‌عامل‌های قابل بوت، پارتیشن‌ها و تنظیمات بوت است.

ساختار دستور

```
grub-mkconfig -o <فایل_پیکربندی>
```

- **grub-mkconfig:** خود دستور است.
- **-o:** گزینه تعیین فایل خروجی است.

- **فایل_پیکربندی :**مسیر کاملی به فایل پیکربندی GRUB که قرار است تولید شود .معمولاً /boot/grub/grub.cfg است.

کاربردهای دستور

- **ایجاد فایل پیکربندی جدید :**هنگامی که تغییراتی در سیستم ایجاد می‌شود (مانند نصب یک سیستم‌عامل جدید یا تغییر پارتیشن‌ها)، این دستور برای ایجاد یک فایل پیکربندی جدید و به روز شده استفاده می‌شود.
- **به‌روزرسانی فایل پیکربندی موجود :**اگر تغییراتی در سیستم ایجاد شده باشد، می‌توان از این دستور برای به‌روزرسانی فایل پیکربندی موجود استفاده کرد.
- **سفارشی‌سازی تنظیمات بوت :**با استفاده از گزینه‌های مختلف، می‌توان تنظیمات بوت را به صورت دلخواه تغییر داد.

گزینه‌های مهم دستور

- **-o فایل_پیکربندی :**<همانطور که گفته شد، این گزینه برای تعیین فایل خروجی استفاده می‌شود.
- **-v:**حالت مفصل را فعال می‌کند و اطلاعات بیشتری در حین اجرای دستور نمایش می‌دهد.
- **--batch:**دستور را به صورت غیر تعاملی اجرا می‌کند.
- **--force:**فایل پیکربندی موجود را بدون پرسیدن سوال بازنویسی می‌کند.

مثال‌ها

- **ایجاد فایل پیکربندی پیش‌فرض:**

```
sudo grub-mkconfig -o /boot/grub/grub.cfg
```

- **ایجاد فایل پیکربندی با حالت مفصل:**

```
sudo grub-mkconfig -o /boot/grub/grub.cfg -v
```

- **بازنویسی اجباری فایل پیکربندی:**

```
sudo grub-mkconfig -o /boot/grub/grub.cfg --force
```

نکات مهم

- **اجرای دستور با دسترسی روت :**برای اجرای این دستور، نیاز به دسترسی روت (sudo) است.
- **تغییر در فایل پیکربندی :**بعد از اجرای این دستور، تغییرات در فایل پیکربندی اعمال می‌شود و در بوت بعدی سیستم اعمال خواهد شد.
- **خطاهای رایج :**اگر در حین اجرای این دستور خطایی رخ دهد، معمولاً به دلیل مشکلات در پارتیشن‌ها، سیستم فایل‌ها یا تنظیمات GRUB است.
- **توجه به تغییرات سیستم :**هر زمان که تغییراتی در سیستم ایجاد می‌شود (مانند نصب یک سیستم‌عامل جدید، تغییر پارتیشن‌ها یا تغییر در تنظیمات بوت)، توصیه می‌شود که فایل پیکربندی GRUB را با استفاده از این دستور به‌روزرسانی کنید.

initramfs چیست؟

initramfsمخفف عبارت **initial RAM filesystem**است و به یک سیستم فایل رم اولیه اشاره دارد که در هنگام بوت شدن سیستم‌عامل لینوکس در حافظه بارگذاری می‌شود .این سیستم فایل کوچک و فشرده، شامل مجموعه‌ای از ابزارها و اسکریپت‌ها است که برای پیکربندی اولیه سیستم و بارگذاری سیستم فایل اصلی مورد نیاز است.

چرا initramfs مهم است؟

- **بوت سریع‌تر :**با بارگذاری سریعinitramfs ، فرآیند بوت شدن سیستم تسریع می‌شود.
- **انعطاف‌پذیری بیشتر** initramfs **:**به سیستم اجازه می‌دهد تا در شرایط مختلف، مانند خرابی دیسک یا عدم شناسایی صحیح سخت‌افزار، به صورت خودکار پیکربندی شود.
- **پشتیبانی از سیستم‌های فایل‌های مختلف** initramfs **:**می‌تواند برای پشتیبانی از انواع مختلف سیستم‌های فایل پیکربندی شود.
- **مدیریت دستگاه‌های سخت‌افزاری** initramfs **:**می‌تواند برای شناسایی و پیکربندی دستگاه‌های سخت‌افزاری جدید استفاده شود.

اجزای اصلی:initramfs

- **Busybox:**یک مجموعه کوچک از ابزارهای لینوکسی است که به صورت استاتیکی کامپایل شده و فضای کمی را اشغال می‌کند .بسیاری از ابزارهای مورد نیاز برای پیکربندی اولیه سیستم در Busybox وجود دارد.
- **اسکریپت‌های :init** این اسکریپت‌ها وظیفه اجرای فرآیند بوت را بر عهده دارند .آن‌ها دستگاه‌ها را شناسایی می‌کنند، سیستم فایل اصلی را مانت می‌کنند و فرآیند init اصلی را شروع می‌کنند.
- **ماژول‌های هسته :**برخی از ماژول‌های هسته که برای بوت شدن سیستم ضروری هستند، ممکن است در initramfs گنجانده شوند.

فرآیند بوت شدن با استفاده از:initramfs

۱. **بارگذاری هسته :**هنگامی که سیستم روشن می‌شود، هسته لینوکس بارگذاری می‌شود.
۲. **بارگذاری :initramfs** هسته، initramfsرا از روی دیسک خوانده و آن را در حافظه بارگذاری می‌کند.
۳. **اجرای اسکریپت‌های :init** اسکریپت‌های init موجود در initramfs اجرا می‌شوند و فرآیند پیکربندی اولیه را انجام می‌دهند.
۴. **مانت سیستم فایل اصلی :**پس از پیکربندی اولیه، سیستم فایل اصلی مانت می‌شود و فرآیند init اصلی شروع به کار می‌کند.

دستور update-initramfs

مقدمه:

همانطور که در توضیح قبلی در مورد initramfs گفتیم، این سیستم فایل نقش مهمی در فرآیند بوت شدن سیستم‌عامل لینوکس ایفا می‌کند .با تغییر در سخت‌افزار، نصب ماژول‌های جدید یا تغییر در پیکربندی هسته، ممکن است نیاز به به‌روزرسانی initramfs باشد تا سیستم بتواند به درستی بوت شود .دستور `update-initramfs`ابزاری است که برای انجام این کار استفاده می‌شود.

ساختار دستور:

```
sudo update-initramfs -u
```

- **Sudo** برای اجرای این دستور با دسترسی روت (superuser) نیاز است.

- **update-initramfs** نام خود دستور است.
- **-u** این گزینه اصلی برای به‌روزرسانی initramfs است.

کاربردهای دستور update-initramfs:

- **به‌روزرسانی خودکار initramfs پس از نصب یا حذف ماژول‌های هسته :**بسیاری از بسته‌های نرم‌افزاری که ماژول‌های هسته را نصب یا حذف می‌کنند، به صورت خودکار دستور `update-initramfs`را اجرا می‌کنند تا اطمینان حاصل شود که initramfs با تغییرات ایجاد شده سازگار است.
- **به‌روزرسانی دستی :initramfs** اگر تغییراتی در پیکربندی هسته یا سیستم فایل ایجاد کرده‌اید، می‌توانید به صورت دستی این دستور را اجرا کنید تا initramfs را به‌روزرسانی کنید.
- **ایجاد یک initramfs سفارشی :**با استفاده از گزینه‌های مختلف این دستور، می‌توانید یک initramfs سفارشی با ویژگی‌های خاص ایجاد کنید.

گزینه‌های مهم دستور update-initramfs:

- **-c:**برای ایجاد یک initramfs جدید بدون جایگزینی initramfs موجود استفاده می‌شود.
- **-k:**برای مشخص کردن نسخه هسته‌ای که می‌خواهیم initramfs برای آن ایجاد شود استفاده می‌شود.
- **-d:**برای فعال کردن حالت دیباگ استفاده می‌شود که اطلاعات بیشتری در مورد فرآیند به‌روزرسانی نمایش می‌دهد.

مثال‌ها:

- **به‌روزرسانی استاندارد:initramfs**

```
sudo update-initramfs -u
```

- **ایجاد یک initramfs جدید با نام:myinitramfs**

```
sudo update-initramfs -c -o /boot/myinitramfs
```

- **به‌روزرسانی initramfs برای هسته نسخه ۵,۴:**

```
sudo update-initramfs -k 5.4
```

نکات مهم:

- پس از اجرای این دستور، بهتر است سیستم را ریبوت کنید تا تغییرات اعمال شده مؤثر شوند.
- برای مشاهده اطلاعات بیشتر در مورد این دستور، می‌توانید از دستور `man update-initramfs`استفاده کنید.

فصل بیست و ششم

فصل بیست و ششم

مدیریت لاگ ها

لاگ در لینوکس چیست؟

فایل‌های گزارش مجموعه‌ای از سوابق هستند که لینوکس برای پیگیری رویدادهای مهم توسط مدیران نگهداری می‌کند. آنها حاوی پیام هایی در مورد سرور، از جمله هسته، سرویس ها و برنامه های در حال اجرا بر روی آن هستند. لینوکس یک مخزن متمرکز از فایل‌های log فراهم می‌کند که می‌تواند در زیر فهرست /var/log قرار گیرد. فایل های گزارش تولید شده در یک محیط لینوکس معمولاً به چهار دسته مختلف طبقه بندی می شوند:

Application Logs
Event Logs
Service Logs
System Logs

- گزارش های برنامه
- گزارش رویداد
- گزارش های خدمات
- گزارش های سیستم

چرا فایل های لاگ لینوکس را پایش کنیم؟

مدیریت گزارش بخشی جدایی ناپذیر از مسئولیت هر مدیر سرور است. با نظارت بر فایل‌های لاگ لینوکس، می‌توانید بینش دقیقی در مورد عملکرد سرور، امنیت، پیام‌های خطا و مشکلات اساسی به دست آورید. اگر می خواهید رویکردی فعال در مقابل یک رویکرد واکنشی برای مدیریت سرور داشته باشید، تجزیه و تحلیل منظم فایل گزارش ۱۰۰٪ مورد نیاز است. به طورخلاصه، فایل‌های گزارش به شما این امکان را می‌دهند که مشکلات آتی را قبل از وقوع واقعی پیش‌بینی کنید .

کدام فایل های لاگ لینوکس را باید نظارت کنیم؟

نظارت و تجزیه و تحلیل همه آنها می تواند یک کار چالش برانگیز باشد. حجم انبوه لاگ‌ها گاهی اوقات می‌تواند باعث ناامید شدن شما شود و فایل مناسبی که حاوی اطلاعات مورد نظر است را پیدا کنید. برای اینکه کار را کمی برای شما آسان کنیم، برخی از حیاتی ترین فایل های لاگ لینوکس را که باید نظارت داشته باشید به شما معرفی می کنیم.

توجه: لطفاً توجه داشته باشید که این یک لیست جامع نیست، بلکه فقط زیرمجموعه ای از فایل های گزارش مهم است که بیشترین اهمیت را دارند. هر چه بیشتر بتوانید مدیریت کنید، برای سلامت سرور شما بهتر است. لیست زیر حداقل حداقل هایی است که باید بدون شکست آنها را نظارت کنید

چه چیزی در /var/log/ ثبت شده است؟:

این فایل گزارش شامل گزارش‌های عمومی فعالیت سیستم است. عمدتاً برای ذخیره پیام های اطلاعاتی و غیر بحرانی سیستم استفاده می شود. در سیستم های مبتنی بر دبیان، فهرست /var/log/syslog به همین منظور عمل می کند.

چگونه می توانم از این لاگ ها استفاده کنم؟:

در اینجا می توانید خطاهای بوت غیر هسته، خطاهای سرویس مربوط به برنامه و پیام هایی که در هنگام راه اندازی سیستم ثبت می شوند را ردیابی کنید.این اولین فایل گزارشی است که مدیران لینوکس باید بررسی کنند که آیا مشکلی پیش آمده است. به عنوان مثال ، شما با مشکلاتی در مورد کارت صدا مواجه هستید. برای بررسی اینکه آیا مشکلی در فرآیند راه‌اندازی سیستم رخ داده است، می‌توانید به پیام‌های ذخیره شده در این فایل گزارش نگاهی بیندازید. تمام رویدادهای مربوط به احراز هویت در سرور دبیان و اوبونتو در / var/log/auth.logثبت می شوند. اگر به دنبال چیزی هستید که مربوط به مکانیسم مجوز کاربر باشد، می‌توانید آن را در این فایل لاگ پیدا کنید.سیستم های مبتنی بر RedHat و CentOS از این فایل log به جای /var/log/auth.log استفاده می کنند. این عمدتا برای ردیابی استفاده از سیستم های مجوز استفاده می شود. تمام پیام های مربوط به امنیت از جمله خرابی های احراز هویت را ذخیره می کند. همچنین لاگین‌های sudo، ورود SSH و سایر خطاهای ثبت‌شده توسط دیمون سرویس‌های امنیتی سیستم را ردیابی می‌کند. همه رویدادهای احراز هویت کاربر در اینجا ثبت می شوند. این فایل گزارش می‌تواند بینش دقیقی درباره تلاش‌های غیرمجاز یا ناموفق برای ورود به سیستم ارائه دهد می تواند برای شناسایی تلاش های احتمالی هک بسیار مفید باشد. همچنین اطلاعات مربوط به ورودهای موفق را ذخیره می کند و فعالیت های کاربران معتبر را ردیابی می کند.

چه چیزی در /var/log/boot.log ثبت شده است؟

اسکریپت اولیه سازی سیستم، /etc/init.d/bootmisc.sh، تمام پیام های بوت آپ را به این فایل گزارش ارسال می کند. این مخزن اطلاعات مربوط به بوت شدن و پیام های ثبت شده در طول فرآیند راه اندازی سیستم است.

چگونه می توانم از این لاگ ها استفاده کنم؟:

شما باید این فایل گزارش را تجزیه و تحلیل کنید تا مسائل مربوط به خاموش شدن نامناسب، راه اندازی مجدد برنامه ریزی نشده یا خرابی های بوت را بررسی کنید. همچنین می تواند برای تعیین مدت زمان توقف سیستم ناشی از خاموش شدن غیرمنتظره مفید باشد.

چه چیزی در /var/log/dmesg ثبت شده است؟

این فایل گزارش شامل پیام های بافر حلقه هسته است. اطلاعات مربوط به دستگاه های سخت افزاری و درایورهای آنها در اینجا ثبت می شود. همانطور که هسته دستگاه های سخت افزار فیزیکی مرتبط با سرور را در طول فرآیند بوت شناسایی می کند، وضعیت دستگاه، خطاهای سخت افزاری و سایر پیام های عمومی را ضبط می کند.

چگونه می توانم از این لاگ ها استفاده کنم؟:

این فایل لاگ بیشتر برای مشتریان سرور اختصاصی مفید است. اگر سخت افزار خاصی به درستی کار نمی کند یا شناسایی نمی شود، می توانید برای عیب یابی به این فایل گزارش اعتماد کنید. یا، می توانید یک سرور مدیریت شده از ما خریداری کنید و ما آن را برای شما نظارت خواهیم کرد.

چه چیزی در /var/log/kern.log ثبت شده است؟

این یک فایل log بسیار مهم است زیرا حاوی اطلاعات ثبت شده توسط هسته است.

چگونه می توانم از این لاگ ها استفاده کنم؟:

ایده آل برای عیب یابی خطاها و هشدارهای مربوط به هسته. گزارش‌های هسته می‌توانند برای عیب‌یابی یک هسته سفارشی‌سازی شده مفید باشند. همچنین می تواند در رفع اشکال سخت افزاری و مشکلات اتصال مفید باشد.

چه چیزی در /var/log/ faillog ثبت شده است؟

این فایل حاوی اطلاعاتی درباره تلاش های ناموفق برای ورود به سیستم است.

چگونه می توانم از این لاگ ها استفاده کنم؟:

این می تواند یک فایل لاگ مفید برای کشف هرگونه تلاش برای نقض امنیتی شامل هک نام کاربری/رمز عبور و حملات brute-force باشد.

چه چیزی در / var/log/cronثبت شده است؟

این فایل لاگ اطلاعات مربوط به cron job ها را ثبت می کند.

چگونه می توانم از این لاگ ها استفاده کنم

هر زمان که یک کار cron اجرا می شود، این فایل لاگ تمام اطلاعات مربوطه از جمله اجرای موفقیت آمیز و پیام های خطا در صورت خرابی را ثبت می کند. اگر با cron برنامه ریزی شده خود مشکل دارید، باید این فایل گزارش را بررسی کنید.

چه چیزی در /var/log/yum.log ثبت شده است؟

این شامل اطلاعاتی است که هنگام نصب بسته جدید با استفاده از دستور yum ثبت می شود.

چگونه می توانم از این لاگ ها استفاده کنم؟:

نصب اجزای سیستم و بسته های نرم افزاری را پیگیری کنید. پیام های ثبت شده در اینجا را بررسی کنید تا ببینید بسته به درستی نصب شده است یا خیر. به شما کمک می کند تا مشکلات مربوط به نصب نرم افزار را عیب یابی کنید. فرض کنید سرور شما غیرعادی عمل می‌کند و گمان می‌کنید که بسته نرم‌افزاری اخیراً نصب شده دلیل اصلی این مشکل است. در چنین مواردی، می‌توانید این فایل لاگ را بررسی کنید تا بسته‌هایی را که اخیراً نصب شده‌اند را بررسی کنید و برنامه خراب را شناسایی کنید.

چه چیزی در /var/log/maillog یا /var/log/mail.log ثبت شده است؟

همه گزارش‌های مربوط به سرور ایمیل در اینجا ذخیره می‌شوند.

چگونه می توانم از این لاگ ها استفاده کنم؟

اطلاعاتی در مورد postfix، smtpd، MailScanner، SpamAssassain یا هر سرویس مرتبط دیگر ایمیلی که در سرور ایمیل اجرا می شود ، بیابید. تمام ایمیل هایی که در یک دوره خاص ارسال یا دریافت شده اند را ردیابی کنید. مسائل مربوط به تحویل نامه ناموفق را بررسی کنید. اطلاعاتی در مورد تلاش های احتمالی ارسال هرزنامه مسدود شده توسط سرور پست دریافت کنید. منشاء ایمیل دریافتی را با بررسی دقیق این فایل گزارش ردیابی کنید.

چه چیزی در /var/log/httpd/ ثبت شده است؟

این فهرست شامل لاگ های ثبت شده توسط سرور آپاچی است. اطلاعات ورود به سیستم سرور آپاچی در دو فایل گزارش مختلف – error_log و access_log ذخیره می شود.

چگونه می توانم از این لاگ ها استفاده کنم؟:

error_log حاوی پیام های مربوط به خطاهای httpd مانند مشکلات حافظه و سایر خطاهای مربوط به سیستم است. این مکانی است که سرور آپاچی رویدادها و سوابق خطاهایی را که هنگام پردازش درخواست‌های httpd با آن مواجه می‌شوند، می‌نویسد. اگر مشکلی با وب سرور آپاچی پیش آمد، این گزارش را برای اطلاعات تشخیصی بررسی کنید. علاوه بر فایل error-log، Apache یک لیست جداگانه از access_log نیز دارد. تمام درخواست های دسترسی دریافت شده از طریق HTTP در فایل access_log ذخیره می شوند.به شما کمک می کند تا هر صفحه ارائه شده و هر فایل بارگذاری شده توسط آپاچی را پیگیری کنید. نشانی IP و شناسه کاربری همه کلاینت هایی را که درخواست اتصال به سرور می کنند، ثبت می کند. اطلاعات مربوط به وضعیت درخواست‌های دسترسی را ذخیره می‌کند، – آیا پاسخی با موفقیت ارسال شده است یا درخواست منجر به شکست شده است.

چه چیزی در /var/log/mysqld.log یا /var/log/mysql.log ثبت شده است؟

همانطور که از نام آن پیداست، این فایل لاگ MySQL است.
همه پیام‌های اشکال‌زدایی، شکست و موفقیت مربوط به دیمون [ ]mysqld و [ ]mysqld_safe در این فایل ثبت شده‌اند. RedHat، CentOS و Fedora گزارش‌های MySQL را در /var/log/mysqld.log ذخیره می‌کنند، در حالی که دبیان و اوبونتو ورود به فهرست /var/log/mysql.log را حفظ می‌کنند.

چگونه می توانم از این گزارش استفاده کنم؟

از این گزارش برای شناسایی مشکلات هنگام شروع، اجرا یا توقف mysqld استفاده کنید. اطلاعاتی در مورد اتصالات کلاینت به فهرست داده MySQL دریافت کنید.همچنین می‌توانید پارامتر « «long_query_timeرا برای ثبت اطلاعات مربوط به قفل‌های پرس و جو و پرس‌وجوهای آهسته تنظیم کنید. در حالی که نظارت و تجزیه و تحلیل همه فایل‌های گزارش تولید شده توسط سیستم می‌تواند کار دشواری باشد، می‌توانید از یک ابزار نظارت بر گزارش متمرکز برای ساده‌سازی فرآیند استفاده کنید. برخی از مشتریان ما از استفاده از Nagios Log Server برای مدیریت گزارش های سرور خود استفاده می کنند. گزینه های منبع باز بسیاری در دسترس هستند اگر خارج از بودجه باشد. نیازی به گفتن نیست که نظارت دستی لاگ های لینوکس سخت است.

بنابراین اگر می‌خواهید یک رویکرد واقعاً فعالانه برای مدیریت سرور داشته باشید، سرمایه‌گذاری در یک پلتفرم جمع‌آوری و تجزیه و تحلیل گزارش متمرکز کنید که به شما امکان می‌دهد داده‌های گزارش را در زمان واقعی مشاهده کنید و هشدارهایی را تنظیم کنید تا در صورت بروز تهدیدات احتمالی به شما اطلاع دهند.

دستورات مدیریت لاگ ها

مشاهده لاگ ها

```
ls -l /var/log
```

```
otal 1472
-rw-------. 1 root root   4524 Nov 15 16:04 anaconda.ifcfg.log
-rw-------. 1 root root  59041 Nov 15 16:04 anaconda.log
-rw-------. 1 root root  42763 Nov 15 16:04 anaconda.program.log
-rw-------. 1 root root 299910 Nov 15 16:04 anaconda.storage.log
-rw-------. 1 root root  40669 Nov 15 16:04 anaconda.syslog
-rw-------. 1 root root  57061 Nov 15 16:04 anaconda.xlog
-rw-------. 1 root root   1829 Nov 15 16:04 anaconda.yum.log
drwxr-x---. 2 root root   4096 Nov 15 16:11 audit
-rw-r--r--  1 root root   2252 Dec  9 10:27 boot.log
```

برای اینکه ببینید چه کسی در حال حاضر وارد سرور لینوکس شده است، به سادگی از دستور who استفاده کنید .این دستور مقادیر خود را از فایل (/var/run/utmp برای CentOS و (Debian یا /run/utmp برای اوبونتو) دریافت می کند. در اینجا یک مثال از CentOSآورده شده است:

```
[root@TestLinux ~]# who
root     tty1         2013-12-09 10:44
root      pts/0        2013-12-09 10:29 (10.0.2.2)
sysadmin pts/1        2013-12-09 10:31 (10.0.2.2)
joeblog  pts/2        2013-12-09 10:39 (10.0.2.2)
```

در این مورد خاص، من تنها کاربر سیستم هستم .من سرور را از Oracle VirtualBox اجرا می کردم و به عنوان root از کنسول و یک جلسه SSH به آن دسترسی داشتم .دو حساب کاربری دیگر (sysadmin) و joebolg) نیز به سیستم دسترسی داشتند.

آخرین دستور تاریخچه ورود کاربران را به ما می گوید:

```
[root@TestLinux ~]# last | grep sysadmin
sysadmin pts/1        10.0.2.2         Mon Dec  9 10:31   still logged in
sysadmin pts/0        10.0.2.2         Fri Nov 29 15:42 - crash  (00:01)
sysadmin pts/0        10.0.2.2         Thu Nov 28 17:06 - 17:13  (00:06)
sysadmin pts/0        10.0.2.2         Thu Nov 28 16:17 - 17:05  (00:48)
sysadmin pts/0        10.0.2.2         Thu Nov 28 09:29 - crash  (06:04)
sysadmin pts/0        10.0.2.2         Wed Nov 27 16:37 - down   (00:29)
sysadmin tty1                          Wed Nov 27 14:05 - down   (00:36)
sysadmin tty1                          Wed Nov 27 13:49 - 14:04  (00:15)
```

در این مثال ، من سعی می کنم تاریخچه ورود کاربر sysadmin را پیدا کنم .همانطور که می بینید، چندین مورد وجود داشت که او موفق شد سیستم را خراب کند .برای اطلاع از آخرین باری که سیستم راه اندازی مجدد شده است، می توانیم دستور زیر را اجرا کنیم:

```
[root@TestLinux ~]# last reboot
```

نتیجه ممکن است به این شکل باشد

```
reboot   system boot  2.6.32-358.el6.x Mon Dec  9 10:27 - 10:47  (00:19)
reboot   system boot  2.6.32-358.el6.x Fri Dec  6 16:37 - 10:47 (2+18:10)
```

```
reboot   system boot  2.6.32-358.el6.x Fri Dec  6 16:28 - 16:36  (00:08)    reboot
system boot  2.6.32-358.el6.x Fri Dec  6 11:06 - 16:36  (05:29)
reboot   system boot  2.6.32-358.el6.x Mon Dec  2 17:00 - 16:36 (3+23:36)
reboot   system boot  2.6.32-358.el6.x Fri Nov 29 16:01 - 16:36 (7+00:34)
reboot   system boot  2.6.32-358.el6.x Fri Nov 29 15:43 - 16:36 (7+00:53)
...
...
wtmp begins Fri Nov 15 16:11:54 2013
```

برای اینکه ببینید آخرین بار کی وارد سیستم شده است، از lastlog استفاده کنید:

```
[root@TestLinux ~]# lastlog
```

در سیستم من، خروجی به این صورت بود:

```
Username         Port     From             Latest
root             tty1                      Mon Dec  9 10:44:30 +1100 2013
bin                                        Never logged in
daemon                                     Never logged in
adm                                        Never logged in
lp                                         Never logged in
sync                                       Never logged in
shutdown                                   Never logged in
halt                                       Never logged in
mail                                       Never logged in
uucp                                       Never logged in
operator                                   Never logged in
games                                      Never logged in
gopher                                     Never logged in
ftp                                        Never logged in
nobody                                     Never logged in
vcsa                                       Never logged in
saslauth                                   Never logged in
postfix                                    Never logged in
sshd                                       Never logged in
sysadmin         pts/1    10.0.2.2         Mon Dec  9 10:31:50 +1100 2013
dbus                                       Never logged in
joeblog          pts/2    10.0.2.2         Mon Dec  9 10:39:24 +1100 2013
```

برای سایر فایل‌های گزارش متنی، می‌توانید از دستوراتcat ، headیا tail برای خواندن مطالب استفاده کنید. در مثال زیر، سعی می کنم ده خط آخر فایل /var/log/messages را در جعبه دیبان بررسی کنم:

```
debian@debian:~$ sudo tail /var/log/messages
```

خروجی دستور:

```
Dec 16 01:21:08 debian kernel: [    9.584074] Bluetooth: BNEP (Ethernet Emulation)
ver 1.3
Dec 16 01:21:08 debian kernel: [    9.584074] Bluetooth: BNEP filters: protocol
multicast
Dec 16 01:21:08 debian kernel: [    9.648220] Bridge firewalling registered
Dec 16 01:21:08 debian kernel: [    9.696728] Bluetooth: SCO (Voice Link) ver 0.6
Dec 16 01:21:08 debian kernel: [    9.696728] Bluetooth: SCO socket layer
initialized
Dec 16 01:21:08 debian kernel: [    9.832215] lp: driver loaded but no devices
found
Dec 16 01:21:08 debian kernel: [    9.868897] ppdev: user-space parallel port
driver
Dec 16 01:21:11 debian kernel: [   12.748833] [drm] Initialized drm 1.1.0 20060810
Dec 16 01:21:11 debian kernel: [   12.754412] pci 0000:00:02.0: PCI INT A ->
Link[LNKB] -> GSI 11 (level, low) -> IRQ 11
```

دستور journalctl

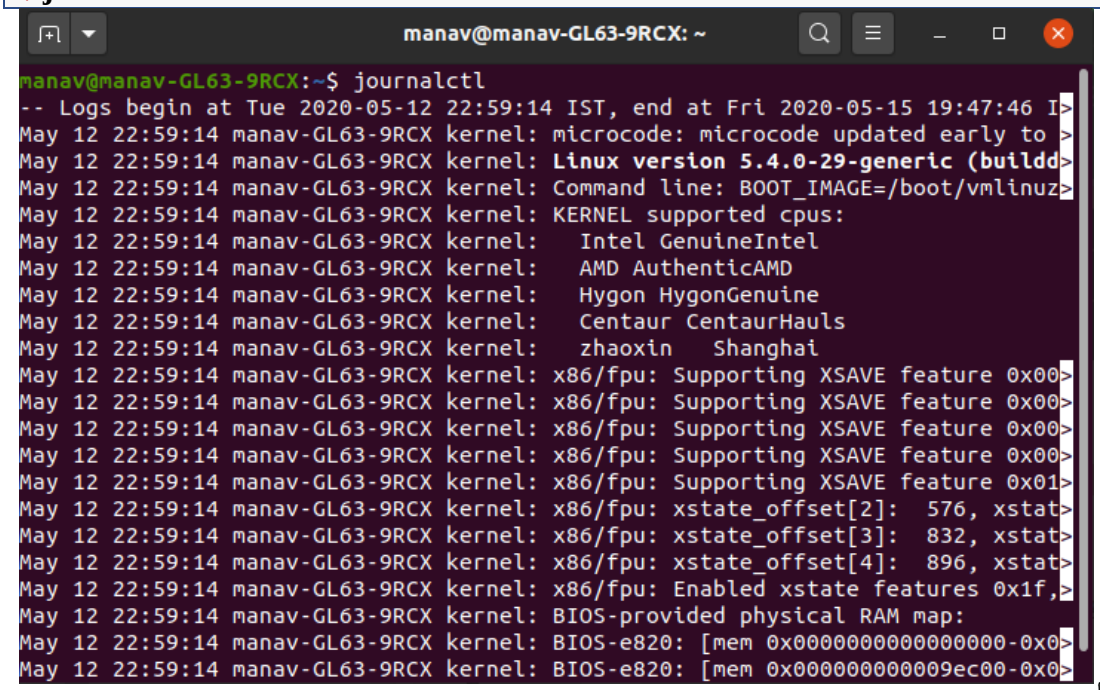
از دستور **journalctl** در لینوکس برای مشاهده log های systemd ، kernalو ژورنال استفاده می شود .گزارش ها به روش زیر ارائه می شوند:

```
-- Logs begin at [date_time_stamp], end at [dat_time_stamp]
[date_time_stamp] [host_name] kernel:
```

این خروجی صفحه بندی شده را نمایش می دهد ، از این رو حرکت در بسیاری از سیاههها کمی آسان است .این اولویت ورود به سیستم را با ترتیب قدیمی تر چاپ می کند.

برای نمایش همه

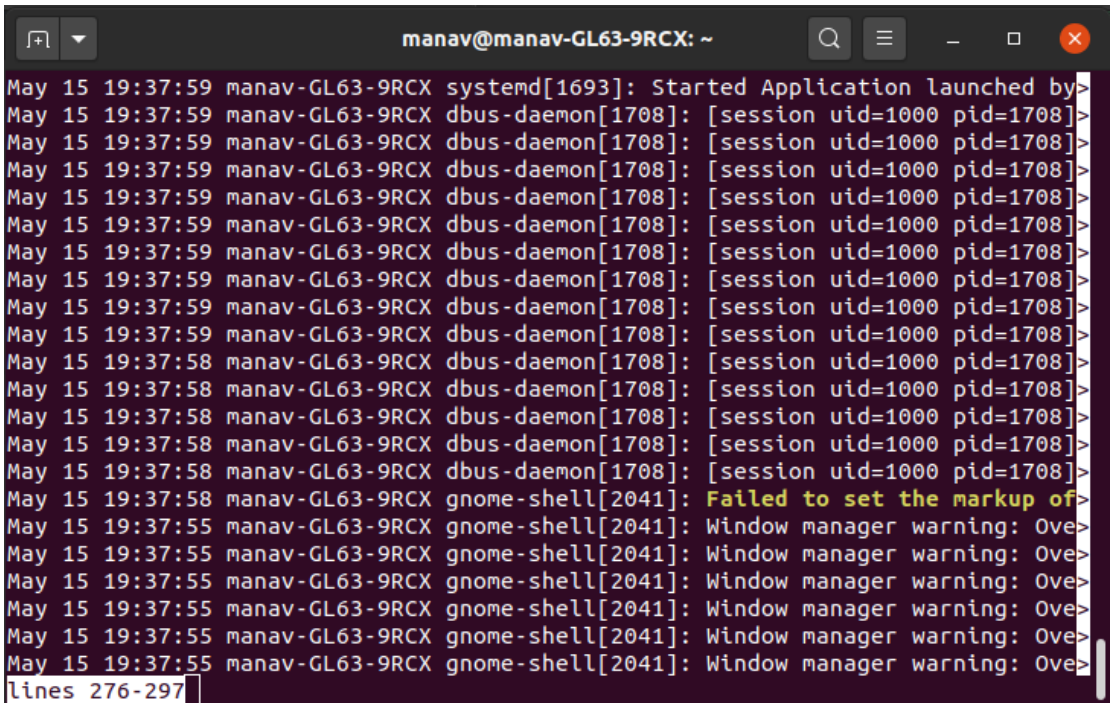
```
$ journalctl
```



این همه logs ی مربوط به نمایش صفحه بندی شده را نشان می دهد.

برای معکوس کردن سفارش یا نمایش ورودی های اول.

```
$ journalctl -r
```



این logs به ترتیب زمانی با جدیدترین ها نمایش داده می شود.

برای نمایش فقط چند ورودی ورود به سیستم

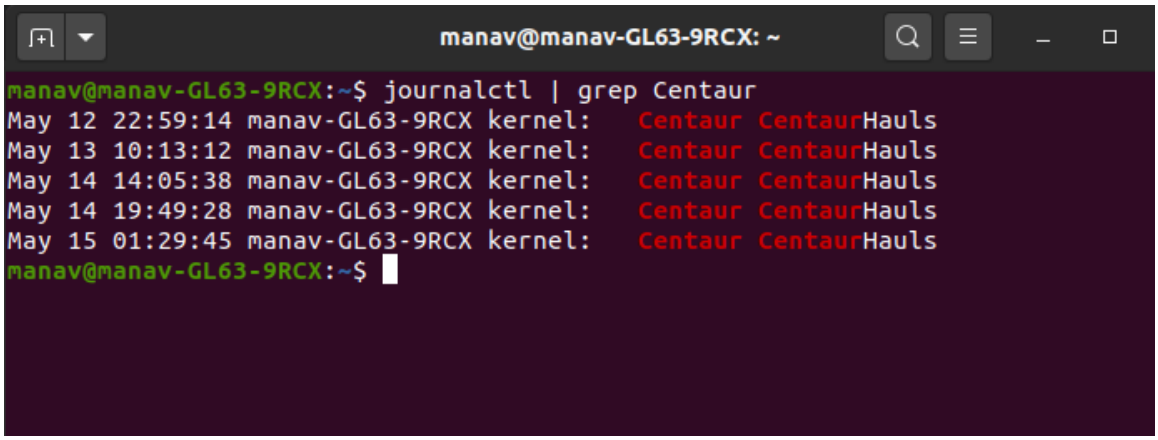
```
$ journalctl -n 2
```



این فقط ۲ ورودی ورود به سیستم را نشان می دهد.

برای دریافت ورودی های ورودی حاوی یک کلمه کلیدی خاص.

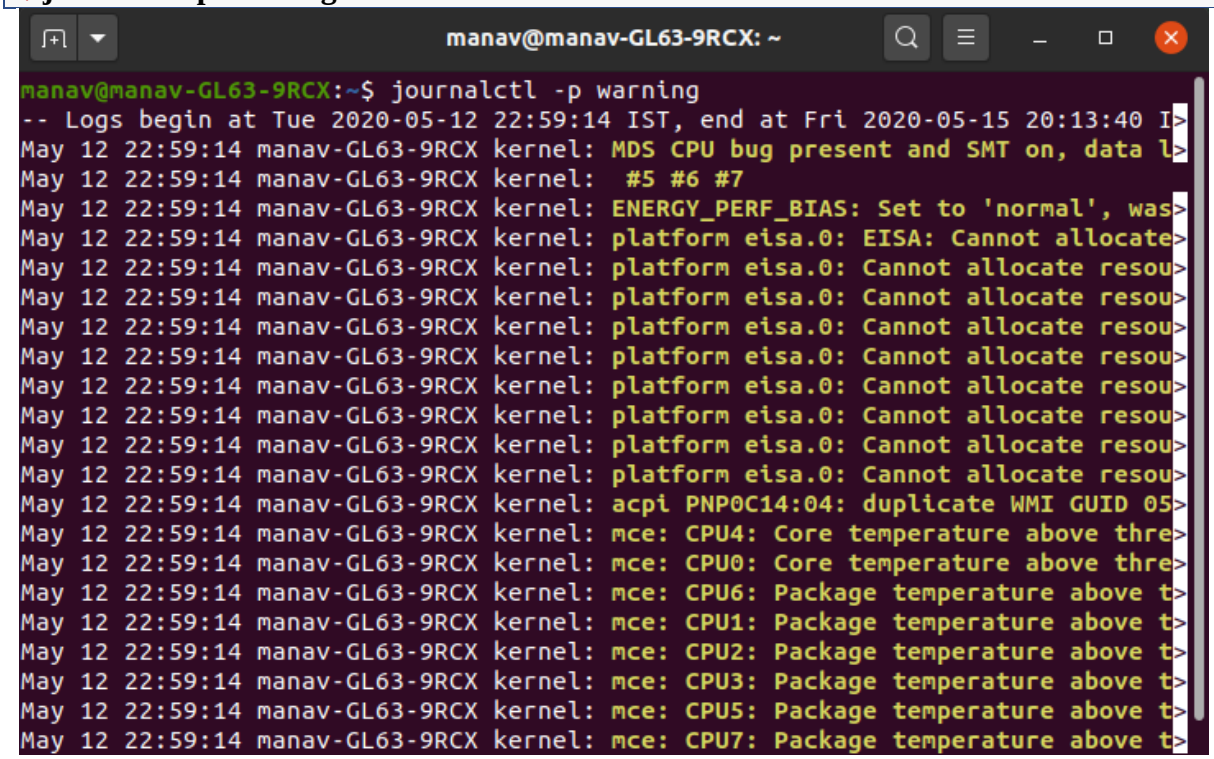
```
$ journalctl | grep Centaur
```



با این کار تمام ورودی های حاوی کلمه Centaur در آنها نمایش داده می شود.

برای نمایش اولویت ورودی های خاص.

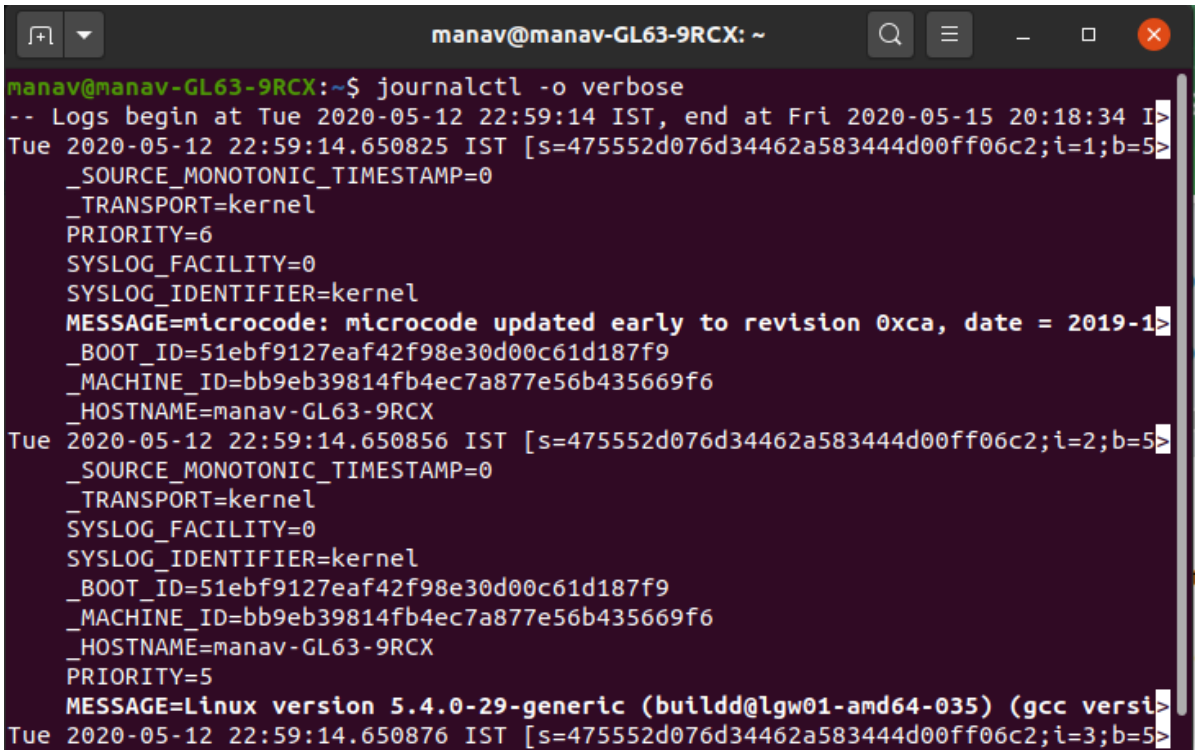
```
$ journalctl -p warning
```



تمام ورودی های گزارش را با اولویت به عنوان یک warning نمایش می دهد.

برای چاپ خروجی متناسب سفارشی.

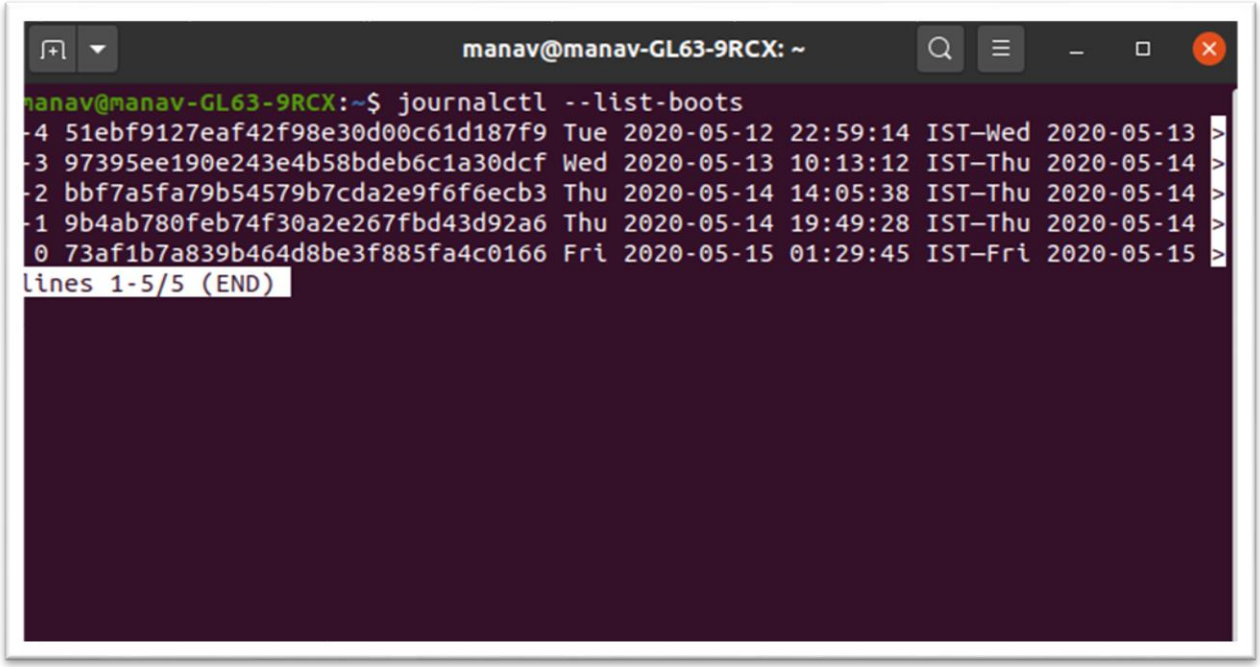
```
$ journalctl -o verbose
```



با این کار خروجی قالب بندی شده در حالت پر حرف نمایش داده می شود.

نمایش پیام های بوت سیستم

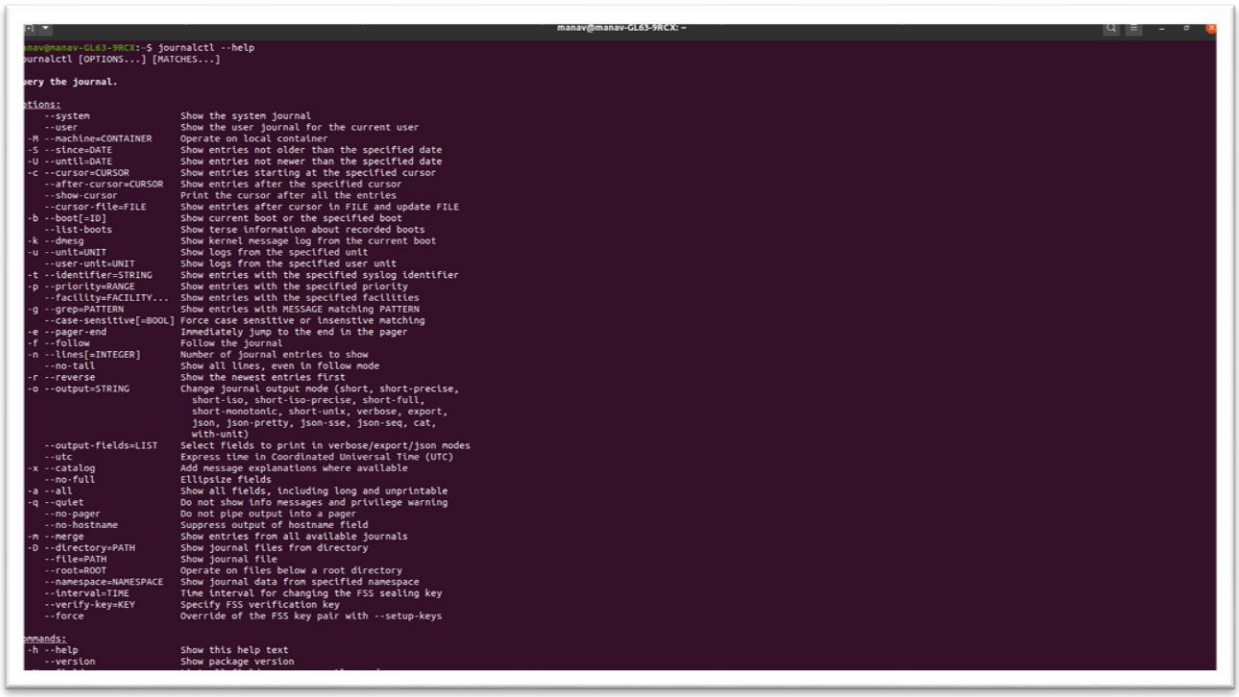
```
$ journalctl --list-boots
```

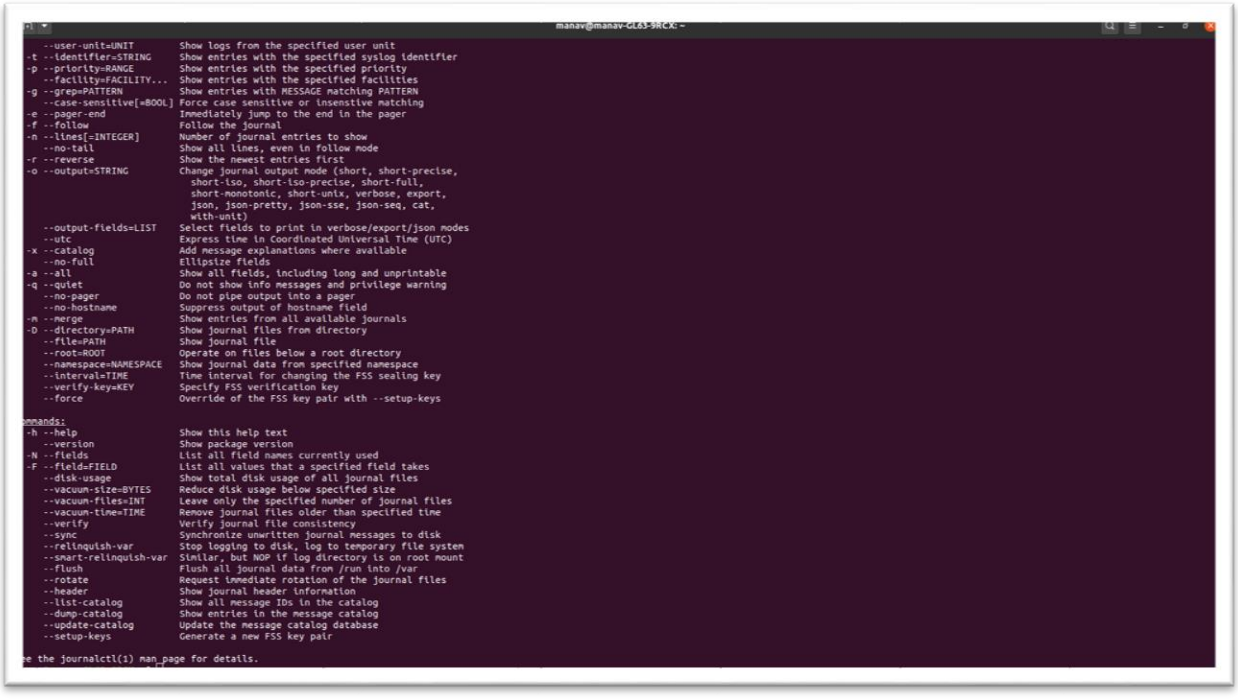


برای نمایش تمام بوت های سیستم.

برای نمایش راهنمای journalctl

```
$journalctl --help
```





دستور بالا بخش راهنمای دستور journalctl را نمایش می دهد.

دستور logrotate

دستور **logrotate** در سیستم‌عامل‌های **Unix/Linux** برای مدیریت و فشرده‌سازی فایل‌های **log** (لاگ) استفاده می‌شود. این دستور به کاربر امکان می‌دهد فایل‌های **log** را بر اساس تنظیمات مشخصی فشرده کند، پاک کند و مدیریت کند تا از فضای دیسک بهینه‌تری استفاده شود و جلوگیری از پر شدن فضای دیسک شود.

گزینه های دستوری:

- d- فعال کردن حالت debug
- f- اجبار به پردازش فایل های log حتی اگر نیازی به چرخش (rotate) باشد.
- l- نمایش لیست فایل های log که باید پردازش شوند.
- m- ارسال ایمیل به کاربر مشخص شده پس از پردازش هر فایل log.
- s- تنظیم نام فایل های log چرخیده شده.
- t- تنظیم تعداد فایل های log چرخیده شده که باید نگه داشته شوند.
- v- نمایش اطلاعات بیشتر در مورد عملیات logrotate.

مثال های دستور:

۱. **f-** یا **--force:** اجبار **logrotate** برای اعمال تنظیمات بدون در انتظار برنامه‌های **log** نویس.

```
logrotate -f
```

۲. **v-** یا **--verbose:** نمایش پیام‌های **verbose** برای هر مرحله از فشرده‌سازی **log**.

```
logrotate -v
```

۳. **d-** یا **--debug:** اجرای **logrotate** به‌صورت **debug** برای دیباگ کردن مشکلات.

```
logrotate -d
```

۴. **s-** یا **--state:** استفاده از یک فایل **state** خاص برای ذخیره اطلاعات وضعیت **logrotate**.

```
logrotate -s /var/log/logrotate-state
```

۵. **c-** یا **--force:** اجبار **logrotate** برای اجرای تنظیمات بدون فشرده‌سازی فایل‌های **log**.

```
logrotate -c
```

۶. **l-** یا **--log:** تعیین یک فایل **log** برای ذخیره پیام‌های **logrotate**.

```
logrotate -l /var/log/logrotate.log
```

۷. **t-** یا **--test:** اجرای **logrotate** به‌صورت آزمایشی برای بررسی تنظیمات بدون اعمال تغییرات.

```
logrotate -t
```

فصل بیست و هفتم

فصل بیست و هفتم

دستورات سرگرم کننده در Linux

دستور orca

orca یک صفحه خوان برای افراد دارای اختلالات بینایی است و با استفاده از ترکیب گفتار، خط بریل و پشتیبانی از بزرگنمایی در پلت فرم، دسترسی جایگزین به دسکتاپ را فراهم می‌کند.علاوه بر این، orca فقط به برنامه‌ها/کیت‌های ابزاری دسترسی دارد که از رابط ارائه خدمات فناوری کمکی ((AT-SPIپشتیبانی می‌کنند، که شامل GTK، Mozilla، Firefox، Evolution، OpenOffice، StarOffice، Java/Swing و غیره می‌شود.

ساختار دستور

```
$ orca [option...]
```

گزینه های دستور :

- s یا guisetup یا setup: هنگام شروع اجرای برنامه Orca، پنجره تنظیمات گرافیکی را نمایش دهید.
- t یا textsetup: هنگام شروع اجرای برنامه Orca، تنظیمات مبتنی بر متن را آغاز کنید.
- n یا nosetup: هنگام شروع اجرای برنامه Orca، برنامه را بدون تنظیمات اجرا کنید، حتی اگر نیاز به تنظیمات داشته باشد. این وقتی مفید است که Orca را از طریق مواردی مانند gdm شروع می‌کنید.
- u dirname یا userprefsdir=dirname: هنگام شروع اجرای برنامه Orca، dirname را به عنوان یک دایرکتوری جایگزین برای تنظیمات کاربر استفاده کنید.
- ? یا h یا help: نمایش پیام راهنمای این نسخه از برنامه Orca.
- l یا listapps: نام‌های تمام برنامه‌های در حال اجرا را چاپ کنید. این عمدتاً برای اهداف اشکال‌زدایی است تا ببینید آیا Orca می‌تواند با زیرساخت دسترسی ارتباط برقرار کند. توجه کنید که اگر Orca در حال اجرا باشد، این فرآیند دیگر Orca را نمی‌کشد. فقط لیست برنامه‌های در حال اجرا را نمایش می‌دهد و شما Orca را دو بار لیست می‌کنید: یک بار برای Orca موجود و یک بار برای این نمونه.
- Debug: فعال‌سازی خروجی اشکال‌زدایی برای Orca و ارسال تمام خروجی‌های اشکال‌زدایی به یک فایل با نامی به شکل ‘debugYYYYMMDDHH:MM:SS.out’ در دایرکتوری فعلی. بخش YYYYMMDDHH:MM:SS با تاریخ و زمان فعلی جایگزین می‌شود.
- debugfile=filename: فعال‌سازی خروجی اشکال‌زدایی برای Orca و ارسال تمام

مثال های دستور :

s یا guisetup یا setup این دستور برای اجرای راه‌اندازی گرافیکی برنامه استفاده می‌شود. مثال

```
orca guisetup
```

t یا textsetup این دستور برای اجرای راه‌اندازی متنی برنامه استفاده می‌شود. مثال

```
orca textsetup
```

n یا nosetup این دستور برای عدم اجرای هیچ نوع راه‌اندازی خاصی استفاده می‌شود. مثال

```
orca nosetup
```

u dirname یا userprefsdir=dirname این دستور برای تعیین مسیر دایرکتوری ترجیحات کاربر استفاده می‌شود. مثال

```
orca userprefsdir=dirname
```

این دستور برای فعال‌سازی یکی از ویژگی‌های موجود اعم از گفتار، برلیل، نظارت بر برلیل، بزرگنمایی یا پنجره اصلی استفاده می‌شود.
مثال

```
orca enable=speech
```

این دستور برای غیرفعال‌سازی یکی از ویژگی‌های موجود اعم از گفتار، برلیل، نظارت بر برلیل، بزرگنمایی یا پنجره اصلی استفاده می‌شود. مثال

```
orca disable=braille
```

? یا h یا help این دستور برای دریافت راهنمای استفاده از برنامه استفاده می‌شود. مثال

```
orca help
```

l یا listapps این دستور برای فهرست کردن برنامه‌های موجود استفاده می‌شود. مثال

```
orca listapps
```

debug این دستور برای فعال‌سازی حالت اشکال‌زدایی در برنامه استفاده می‌شود. مثال

```
orca debug
```

debugfile=filename این دستور برای تعیین فایلی برای ذخیره اطلاعات اشکال‌زدایی استفاده می‌شود. مثال

```
orca debugfile=filename
```

دستور xeyes

Xeyes یک برنامه رابط کاربری گرافیکی است که مجموعه ای از چشم ها را روی دسکتاپ ایجاد می‌کند که حرکت مکان نما ماوس را دنبال می‌کند. به نظر می رسد این دستور بسیار خنده دار است تا هر کاربرد مفیدی. بامزه بودن به همان اندازه که مفید است، جنبه دیگری است.

ساختار دستور:

```
$ xeyes
```

گزینه های دستور:

- geometry– اندازه پنجره را به صورت عرضxارتفاع پیکسل مشخص می کند.
- display– نام نمایشگر X Window را مشخص می کند.
- iconic– پنجره را به صورت icon نمایش می دهد.
- name– نام پنجره را مشخص می کند.
- noborder– پنجره را بدون لبه نمایش می دهد.
- root– پنجره را به عنوان فرزند ریشه پنجره نمایش می دهد.
- synchro– حرکات چشم را با حرکات ماوس همگام سازی می کند.
- visual– نوع visual را برای پنجره مشخص می کند.

هدف xeyes چیست؟

xeyes برای سرگرمی نیست. هدف از این برنامه این است که به شما اجازه دهد نشانگر ماوس را دنبال کنید که گاهی اوقات به سختی قابل مشاهده است. در رایانه‌های چند مانیتور بسیار مفید است، جایی که مانیتورها با فاصله کمی از هم جدا شده‌اند و اگر کسی (مثلاً معلم مدرسه) بخواهد چیزی را روی صفحه نمایش ارائه دهد، سایرین روی مانیتورشان می‌توانند به راحتی ماوس را دنبال کنند .

برای نمایش پنجره xeye با اندازه x200۲۰۰ پیکسل:

```
xeye -geometry 200x200
```

برای نمایش پنجره xeye با نام "My Eye":

```
xeye -name "My Eye"
```

برای نمایش پنجره xeye بدون لبه:

```
xeye -noborder
```

برای همگام سازی حرکات چشم با حرکات ماوس:

```
xeye -synchro
```

نکات:

می توانید از کلیدهای Esc یا q برای خروج از برنامه xeye استفاده کنید.

برای جابجایی چشم، می توانید از کلیدهای جهت دار استفاده کنید.

برای بزرگنمایی و کوچکنمایی چشم، می توانید از کلیدهای + و – استفاده کنید.

برای تغییر رنگ چشم، می توانید از کلیدهای ۱ تا ۹ استفاده کنید.

دستور feh

feh یک نمایشگر تصویر مبتنی بر حالت است. این به ویژه برای کاربران خط فرمان که نیاز به یک نمایشگر سریع تصویر بدون وابستگی های بزرگ رابط کاربری گرافیکی دارند، مورد توجه قرار می گیرد، اگرچه می تواند توسط مدیران فایل (گرافیکی) برای مشاهده یک تصویر نیز راه اندازی شود.

feh از لیست فایل‌ها، حالت‌های مختلف مرتب‌سازی تصویر، شرح تصاویر، HTTP و موارد دیگر پشتیبانی می‌کند. برای کنترل آن از میانبرهای صفحه کلید قابل تنظیم استفاده می شود. ماوس نیز پشتیبانی می شود، اما فقط برای اقدامات بسیار کمی مورد نیاز است.از feh به عنوان کاغذ دیواری نیز می توان استفاده کرد. نکته ای کوچک در مورد پشتیبانی EXIF: روش توصیه شده برای نمایش داده های EXIF استفاده از exifgrep / exiv2 از طریق info -- است . این حال، اگر feh را با exif=1 کامپایل کنید، می توانید آن را مستقیماً نیز نمایش دهید.

فرمت های پشتیبانی شده

feh می تواند هر فرمتی را که توسط Imlib2 پشتیبانی می شود، از جمله jpeg و png باز کند. اگر تبدیل باینری (ارائه شده توسط ImageMagick) در دسترس باشد، از بسیاری از انواع فایل های دیگر مانند svg، xcf و otf نیز پشتیبانی محدودی دارد. از magick-timeout num -- با یک مقدار غیر منفی برای فعال کردن آن استفاده کنید. برای تصاویر متحرک، فقط اولین فریم نشان داده می شود.

feh مخفف Fancy Exhange Host است. این برنامه یک نمایشگر تصویر قدرتمند و انعطاف‌پذیر برای لینوکس و سایر سیستم‌های مبتنی بر یونیکس است. feh می‌تواند تصاویر را در فرمت‌های مختلف نمایش دهد، از جمله :

- JPEG
- PNG
- GIF
- BMP
- TGA
- TIFF
- PBM
- PGM
- PPM

گزینه‌های دستور:

feh دارای طیف گسترده‌ای از گزینه‌ها برای کنترل نحوه نمایش تصاویر است. برخی از رایج‌ترین گزینه‌ها عبارتند از

- f– تصویر را در حالت تمام صفحه نمایش می‌دهد.
- g– اندازه تصویر را به گونه‌ای تنظیم می‌کند که تمام صفحه را بپوشاند.
- s– تصویر را به اندازه مشخصی (مثلاً – s 640x480) تغییر اندازه می‌دهد.
- z– تصویر را بزرگ‌نمایی می‌کند.
- m– تصویر را به صورت موزاییک نمایش می‌دهد.
- d– تأخیر بین نمایش تصاویر را در اسلایدشو مشخص می‌کند.
- r– تصاویر را به صورت تصادفی نمایش می‌دهد.
- t– عنوان تصویر را نمایش می‌دهد.
- i– اطلاعات EXIF تصویر را نمایش می‌دهد.
- h– لیست کاملی از گزینه‌ها را نمایش می‌دهد.

برای مرور سریع تصاویر در یک فهرست خاص، می توانید feh را با آرگومان های زیر راه اندازی کنید:

```
$ feh -g 640x480 -d -S نام فایل /path/to/directory
```

پرچم g – تصاویر را مجبور می‌کند که بزرگتر از ۶۴۰*۴۸۰ ظاهر نشوند

پرچم d – نام فایل را ترسیم می‌کند

پرچم فایل S – تصاویر را بر اساس نام فایل مرتب می‌کند

این فقط یک نمونه است؛ در صورت تمایل به انعطاف پذیری بیشتر، گزینه های دستور:ی بیشتری وجود دارد.

ایجاد تصویر برای پس زمینه صفحه نمایش

```
$ feh --bg-scale /path/to/image.file
```

باز کردن تصاویر svg

```
$ feh --conversion-timeout 1 file.svg
```

نمایش تصاویر پس زمینه به صورت راندوم

```
$ feh --bg-fill --randomize ~/.wallpaper/*
```

برای نمایش تصویر image.jpg در حالت تمام صفحه:

```
feh -f image.jpg
```

برای نمایش تصویر image.png با اندازه ۶۴۰:x480

```
feh -s 640x480 image.png
```

برای نمایش تصاویر در پوشه images به صورت اسلایدشو با تأخیر ۵ ثانیه:

```
feh -d 5 -r images/*
```

برای نمایش تصویر image.jpg به صورت موزاییک ۳x۳:

```
feh -m 3x3 image.jpg
```

برای بزرگ‌نمایی تصویر image.png با ضریب ۲:

```
feh -z 2 image.png
```

برای نمایش عنوان و اطلاعات EXIF تصویر image.jpg:

```
feh -t -i image.jpg
```

دستور Cowsay

نصب دستور

```
sudo apt install cowsay
```

cowsay یک دستور در لینوکس است که برای نمایش پیام‌های مورد نظر کاربر به‌صورت گرافیکی با استفاده از یک "گاو" یا شکل دیگری از حیوانات (که به آن‌ها " "cowsگفته می‌شود) استفاده می‌شود.

گزینه‌های دستور:

- b: تبدیل خطوط خالی به خطوط فاصله.
- d: چشمان گاو به سمت پایین نگاه می‌کنند.
- g: گاو خندان.
- p: چشمان گاو خندان به سمت چپ.
- s: گاو خفه شده.
- t: تاج بر سر گاو.
- w: چشمان گاو وحشی.
- y: چشمان گاو به سمت راست.

مثال‌های دستور:

نمایش پیام ساده با گاو:

```
cowsay Hello, world!
```

نمایش پیام با استفاده از گاو خندان:

```
cowsay -g HaHaHa!
```

نمایش پیام با چشمان گاو به سمت چپ:

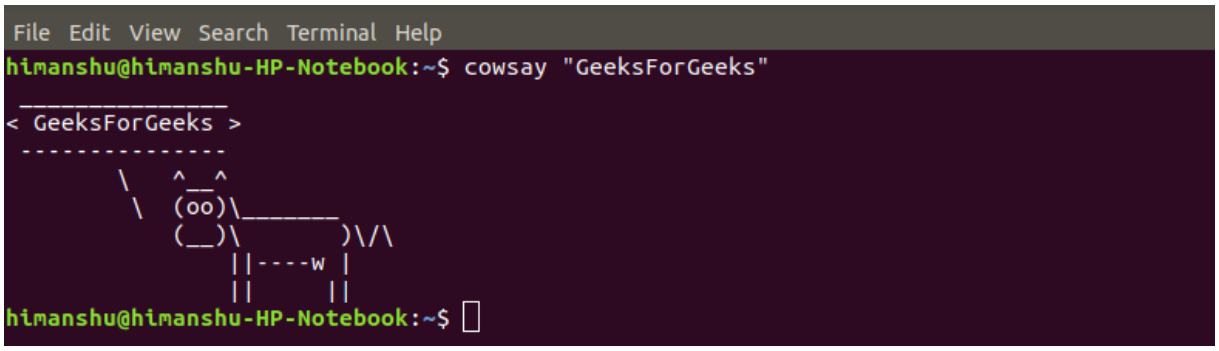
```
cowsay -p Hey there!
```

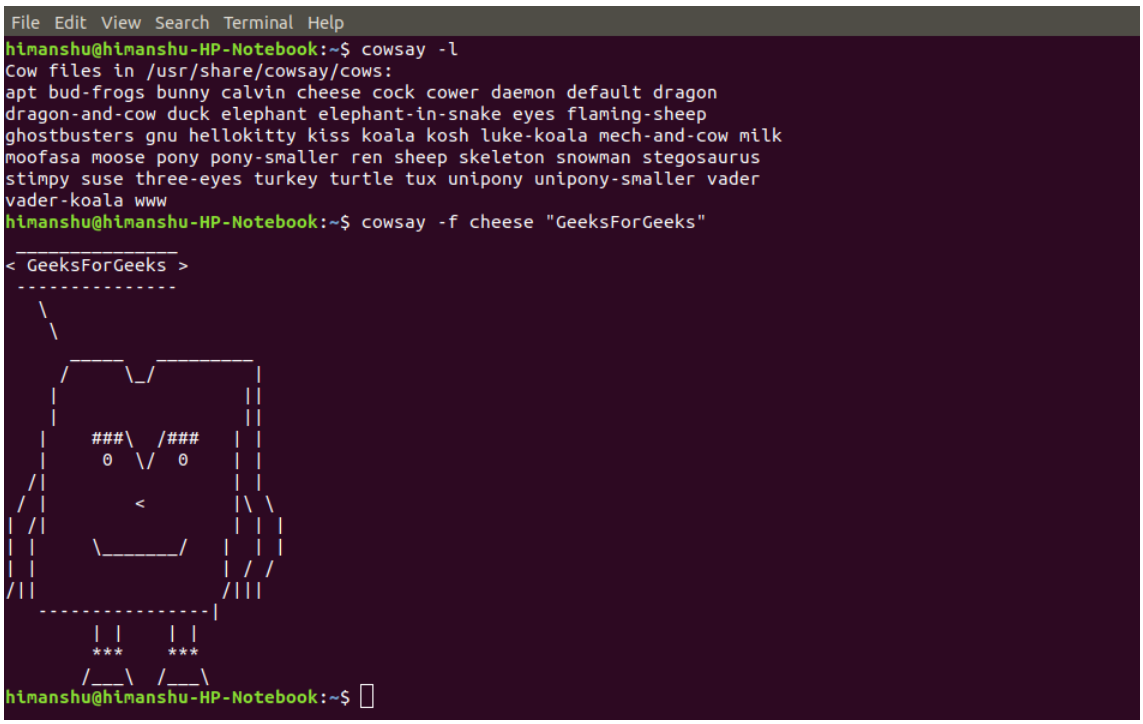
نمایش پیام با تاج بر سر گاو:

```
cowsay -t All hail the cow!
```

نمایش پیام با چشمان گاو وحشی:

```
cowsay -w Beware of the cow!
```





دستور xcowsay

ابتدا ، ما آن را با استفاده از **sudo apt install xcowsay** نصب می کنیم .در اینجا روش استفاده از آن وجود دارد.

xcowsay یک برنامه سطح بالا برای نمایش پیام‌های کاربر به‌صورت گرافیکی با استفاده از یک شخصیت گرافیکی از جنس گاو است.
این برنامه از کتابخانه GTK برای رابط کاربری گرافیکی خود استفاده می‌کند.

نشانه‌های دستور:

- d: تعیین مدت زمان نمایش پنجره .xcowsay
- f: انتخاب شخصیت گرافیکی برای استفاده.
- h: نمایش راهنمای استفاده از .xcowsay
- t: تعیین مدت زمان نمایش پیام .xcowsay
- v: نمایش نسخه فعلی .xcowsay

مثال های دستور:

نمایش پیام با شخصیت پیش‌فرض گاو:

```
xcowsay Hello, world!
```

نمایش پیام با استفاده از شخصیت دیگر (مثلاً شخصیت خرگوش):

```
xcowsay -f bunny How are you doing?
```

نمایش پیام با تعیین مدت زمان نمایش پنجره:

```
xcowsay -d 5 Hello, it's me!
```

نمایش راهنمای استفاده از xcowsay:

```
xcowsay -h
```

نمایش نسخه فعلی xcowsay:

```
xcowsay –v
```

دستور cmatrix

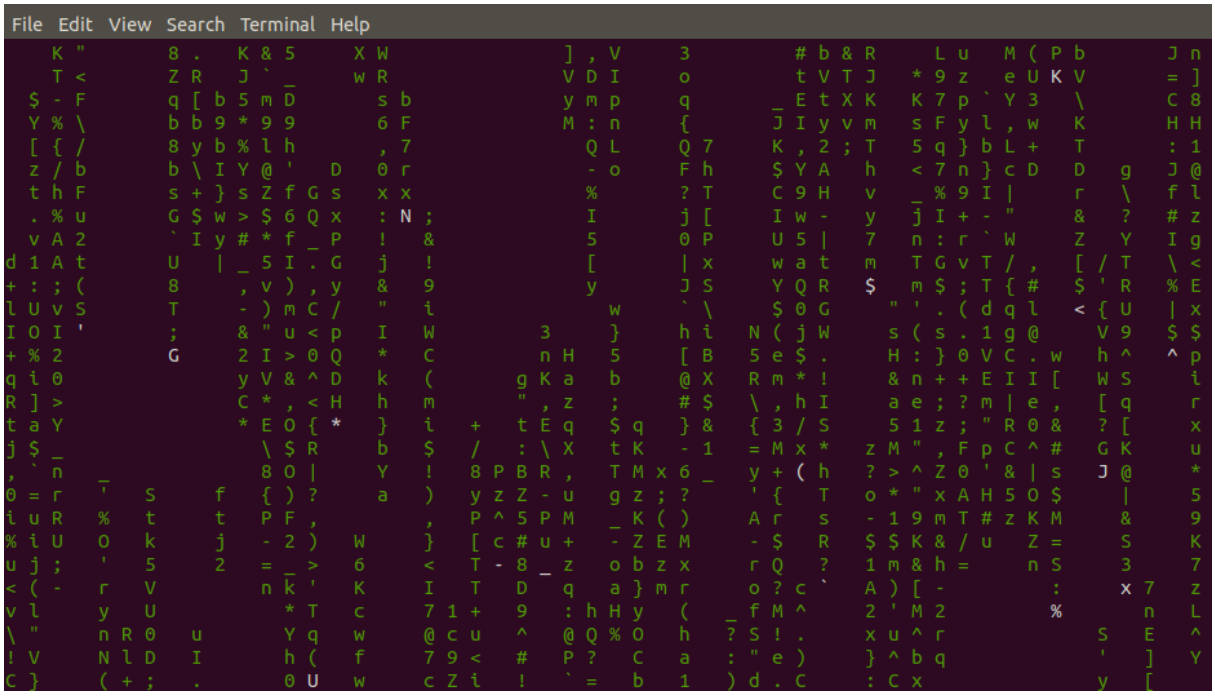
از دستور زیر برای نصب آن استفاده کنید

```
sudo apt install cmatrix
```

پس از نصب ، از دستور زیر برای چاپ استفاده کنید.

```
cmatrix
```

برای نمایش ماتریس سبک نئو در ترمینال.



cmatrix یک برنامه سطح بالا است که یک افکت ماتریس استایل (مانند فیلم "ماتریکس") را در ترمینال نمایش می‌دهد. این برنامه شامل حرکت نوشته‌ها و عددها در سبک ماتریس است.

گزینه‌های دستور:

- a: نمایش همه کاراکترها به‌جای تنها اعداد.
- b: استفاده از کاراکترهای ASCII به‌جای اعداد.
- l: تعیین سرعت حرکت متن.
- n: تعیین تعداد ستون‌ها.
- s: تعیین سرعت حرکت ستون‌ها.
- C: نمایش رنگ‌ها.

- V: نمایش نسخه cmatrix.
- h: نمایش راهنمای استفاده از cmatrix.

مثال های دستور:

نمایش افکت ماتریس استایل با اعداد:

```
cmatrix
```

نمایش افکت ماتریس استایل با کاراکترهای ASCII:

```
cmatrix -b
```

تنظیم سرعت حرکت متن:

```
cmatrix -l 2
```

تنظیم تعداد ستون‌ها:

```
cmatrix -n 80
```

نمایش افکت ماتریس با رنگ‌ها:

```
cmatrix -C
```

نمایش نسخه فعلی cmatrix:

```
cmatrix -V
```

نمایش راهنمای استفاده از cmatrix:

```
cmatrix -h
```

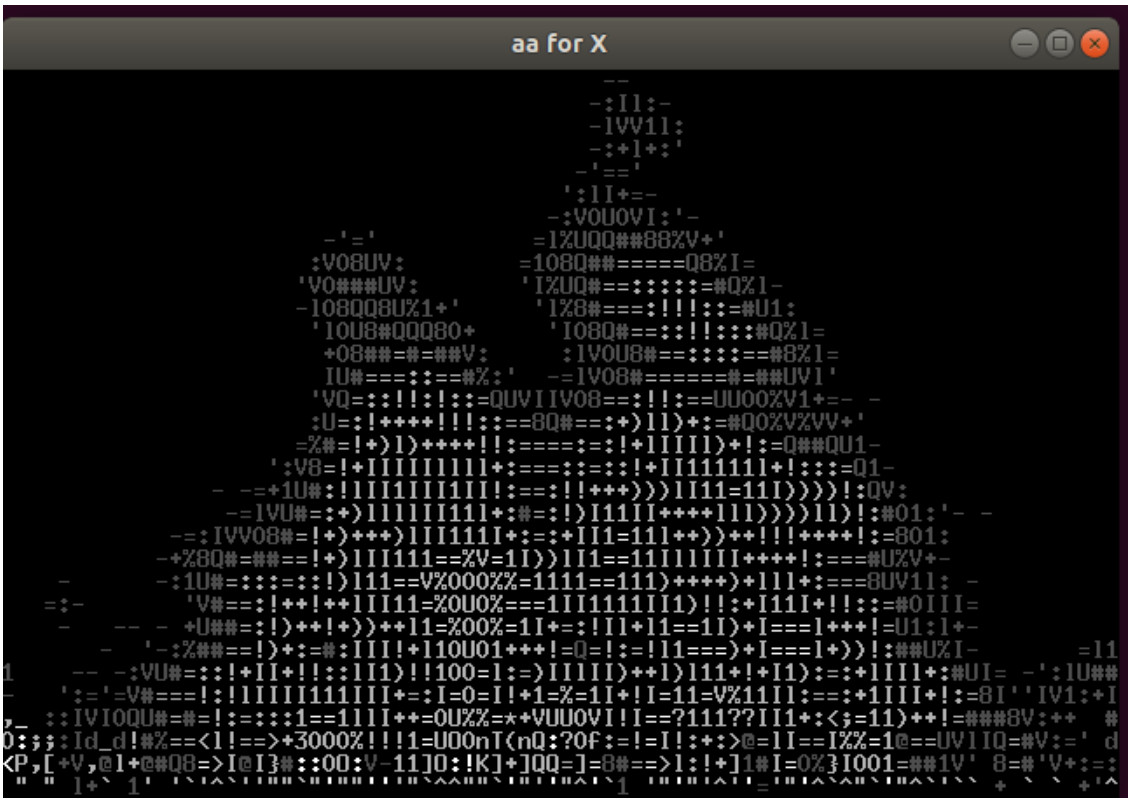
دستور fire

برای نصب آن از دستور زیر استفاده کنید

```
sudo apt install libaa-bin
```

پس از نصب ، از دستور زیر برای چاپ استفاده کنید.

```
aafire
```



aafire یک برنامه سطح بالا است که یک افکت آتش‌سوزی سه‌بعدی را به‌صورت ASCII در ترمینال نمایش می‌دهد. این برنامه بر اساس کاراکترهای ASCII اقدام به نمایش آتش و حرکات آن می‌کند.

گزینه‌های دستور:

- f: - تنظیم سرعت آتش‌سوزی.
- r: - انعکاس آتش به صورت عمودی.
- R: - انعکاس آتش به صورت افقی.
- w: - تعیین عرض و ارتفاع صفحه نمایش.
- h: - نمایش راهنمای استفاده از aafire.

مثال‌های دستور:

نمایش افکت آتش‌سوزی با تنظیمات پیش‌فرض:

```
aafire
```

تنظیم سرعت آتش‌سوزی:

```
aafire -f 2
```

انعکاس آتش به صورت عمودی:

```
aafire -r
```

انعکاس آتش به صورت افقی:

```
aafire -R
```

تعیین ابعاد صفحه نمایش:

```
aafire -w 80x40
```

نمایش راهنمای استفاده از aafire:

```
aafire -h
```

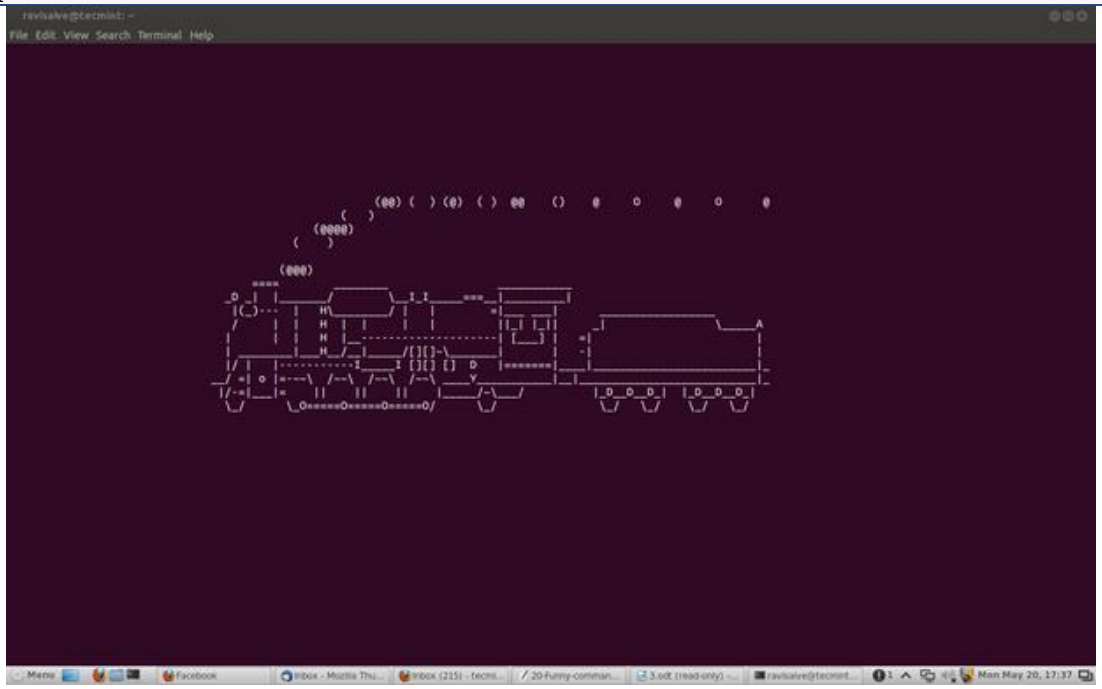
دستور sl (لوکوموتیو بخار)

ممکن است از دستور " ls " فرمان لیست آگاه باشید ، که اغلب برای مشاهده محتویات یک پوشه استفاده می شود، اما به دلیل تایپ اشتباه گاهی اوقات به " sl " منجر می شود، چگونه در ترمینال کمی سرگرم کننده باشید و نه " دستور یافت نشد ."
slرا نصب کنید

```
$ sudo apt install sl [در Debian/Ubuntu & Mint]
$ sudo yum install sl [در CentOS/RHEL 7]
$ sudo dnf install sl [در CentOS/RHEL 8 و Fedora]
$ sudo pacman -S sl [در آرچ لینوکس]
$ sudo pkg_add -v sl [در FreeBSD]
```

خروجی دستور:

```
root@predator-os :~# sl
```



sl یک برنامه سطح بالا است که یک قطار به‌صورت افتراقی در ترمینال نمایش می‌دهد. این برنامه اغلب به‌صورت شوخی برای کاربران نمایش داده می‌شود و ممکن است به‌صورت پیش‌فرض بر روی سیستم‌های خاصی نصب نباشد.

گزینه‌های دستور

- a: نمایش انواع مختلف قطارها.
- l: تنظیم سرعت حرکت قطار.
- F: فعال کردن صدا.
- e: نمایش خروجی برای انواع مختلف قطارها.
- L: تعیین طول قطار.
- h: نمایش راهنمای استفاده از sl.

مثال های دستور

نمایش قطار به‌صورت پیش‌فرض:

```
sl
```

نمایش قطار با تنظیم سرعت حرکت:

```
sl -l fast
```

فعال کردن صدا برای قطار:

```
sl -F
```

نمایش انواع مختلف قطارها:

```
sl -a
```

تعیین طول قطار:

```
sl -L 10
```

نمایش خروجی برای انواع مختلف قطارها:

```
sl -e
```

نمایش راهنمای استفاده از sl:

```
sl -h
```

دستور banner

از دستور **banner** در linux برای چاپ رشته کاراکتر ASCII با حروف بزرگ برای خروجی ایستاده استفاده می شود.

ساختار دستور:

```
banner text
```

مثال ۱ : چاپ "۱۲۳۴۵۶۷۸۹۰" با حروف بزرگ.



banner یک برنامه ساده است که متن ورودی را به صورت بزرگ و با استفاده از کاراکترهای ASCII در ترمینال نمایش می‌دهد. این برنامه برای ایجاد عنوان‌های بزرگ و جذاب در اسکریپت‌ها یا پیام‌های خوشامدگویی استفاده می‌شود.

گزینه‌های دستور:

- w: – تنظیم عرض بنر.
- f: – انتخاب فونت برای بنر.
- t: – تنظیم خروجی بنر به صورت پایدار.
- v: – نمایش اطلاعات نسخه.
- h: – نمایش راهنمای استفاده از banner.

مثال های دستور:

ایجاد بنر با متن "Hello":

```
banner Hello
```

تنظیم عرض بنر:

```
banner -w 40 Hello
```

انتخاب فونت برای بنر:

```
banner -f big Hello
```

تنظیم خروجی بنر به صورت پایدار:

```
banner -t Hello
```

نمایش اطلاعات نسخه:

```
banner -v
```

نمایش راهنمای استفاده از banner:

```
banner -h
```

توجه: برای نصب ابزار ازدستور زیر استفاده نمایید.

```
$ sudo apt install sysvbanner
$ sudo apt install banner
```

برای بارگیری آن استفاده کنید.

دستور bc

دستور bc قابلیت انجام محاسبات ریاضی را از طریق خط دستور فراهم می‌کند.

گزینه های دستور:

- l– کتابخانه استاندارد bc را بارگیری می کند.
- q– خروجی را فقط به اعداد محدود می کند.
- s– حالت تعاملی را فعال می کند.
- w– عرض پیش فرض اعداد را تنظیم می کند.
- i– فایل ورودی را مشخص می کند.
- o– فایل خروجی را مشخص می کند.

مثال های دستور:

۱. ریاضی:

```
Input : $ echo "11+5" | bc
output : 16
```

۲. افزایشی:

var ++ عملگر افزایش پست، ابتدا از نتیجه متغیر استفاده می شود و سپس متغیر افزایش می یابد.

var++ : عملگر قبل از افزایش، متغیر ابتدا افزایش می یابد و سپس نتیجه متغیر ذخیره می شود.

```
Input: $ echo "var=3;++var" | bc
Output: 4
```

۳. کاهش:

var – – : عملگر Post decrement ابتدا از نتیجه متغیر استفاده می شود و سپس متغیر کاهش می یابد.

– – var : عملگر قبل از کاهش، ابتدا متغیر کاهش یافته و سپس نتیجه متغیر ذخیره می شود.

```
Input: $ echo "var=3;--var" | bc
```

خروجی دستور: ۲

۴. Assignment:

```
var = value : Assign the value to the variable
var += value : similar to var = var + value
var -= value : similar to var = var – value
var *= value : similar to var = var * value
var /= value : similar to var = var / value
var ^= value : similar to var = var ^ value
var %= value : similar to var = var % value
```

```
Input: $ echo "var=4;var" | bc
```

4

خروجی دستور: ٤

۵. مقایسه یا رابطه ای:

expr1<=expr2: نتیجه ۱ است، اگر expr1 کمتر یا مساوی expr2 باشد.

expr1>expr2: نتیجه ۱ است، اگر expr1 به شدت بزرگتر از expr2 باشد.

expr1>=expr2: نتیجه ۱ است، اگر expr1 بزرگتر یا مساوی با expr2 باشد.

expr1==expr2: نتیجه ۱ است، اگر expr1 برابر با expr2 باشد.

expr1!=expr2: نتیجه ۱ است، اگر expr1 برابر با expr2 نباشد.

```
Input: $ echo "6
```

خروجی دستور: ۰

```
Input: $ echo "2==2" | bc
```

خروجی دستور: ۱

۶. منطقی یا بولی:

expr1 && expr2: اگر هر دو عبارت غیر صفر باشند، نتیجه ۱ است.

expr1 || expr2 : اگر هر کدام از عبارت ها غیر صفر باشد، نتیجه ۱ است.

! expr: نتیجه ۱ است، اگر expr 0 باشد.

```
Input: $ echo "! 1" | bc
```

خروجی دستور: ۰

```
Input: $ echo "10 && 5" | bc
```

خروجی دستور: ۱

مثال های دستور !

اجرای یک محاسبه ساده با استفاده از bc:

```
$echo "2 + 3" | bc
```

استفاده از مقیاس دهدهی برای نمایش اعداد اعشاری با دقت تعیین شده (۲ رقم اعشار):

```
$echo “scale=2; 10 / 3” | bc
```

اجرای یک فایل متنی حاوی دستورات bc:

```
$bc script.bc
```

استفاده از توابع ریاضی در bc، مانند پی:

```
$echo “scale=10; 4*a(1)” | bc -l
```

استفاده از متغیرها در bc:

```
$echo “a = 5; b = 2; a + b” | bc
```

استفاده از توابع ریاضی پیش‌فرض bc، مانند جذر مربعی:

```
$echo “sqrt(16)” | bc -l
```

استفاده از توابع پیش‌فرض bcبرای محاسبه لگاریتم طبیعی:

```
$echo “l(10)” | bc -l
```

دستور yes

دستور yes در لینوکس برای چاپ یک جریان خروجی پیوسته از STRING استفاده می شود . اگر STRING ذکر نشده باشد، “y” را چاپ می‌کند. (با استفاده از ctrl + c) خروجی را متوقف کنید.

مثال های دستور :

hello world را بی نهایت در ترمینال چاپ می‌کند تا زمانی که kill شود:

```
$  yes hello world
```

ساختاردستور :

```
$  yes [STRING]
```

گزینه‌های دستور:

- c- تعداد دفعات تکرار پاسخ مثبت را مشخص می‌کند.
- n- رشته‌ای را که به عنوان پاسخ مثبت استفاده می‌شود، مشخص می‌کند.
- s- پاسخ مثبت را به صورت رشته‌ای خالی در نظر می‌گیرد.
- u- پاسخ مثبت را به صورت حرف “ ”yدر نظر می‌گیرد.

برای تکرار رشته دلخواه به جای:“yes”

```
$yes “Hello, world!”
```

برای تکرار رشته خالی:

```
$yes ““
```

برای تکرار رشته و انتهای هر خط:

```
$yes | sed ‘s/$/ End of line/’
```

برای تکرار رشته تا تعداد دفعات مشخص شده:

```
$yes | head -n 10
```

دستور gifdiff

تصاویر GIF را مقایسه می‌کند

ساختار دستور:

```
$ gifdiff [options] GIF-file-1 GIF-file-2
```

شرح دستور:

gifdiff دو فایل GIF را با هم مقایسه می‌کند و تعیین می‌کند که آیا آنها یکسان به نظر می رسند یا خیر. تفاوت‌هایی که بر ظاهر تأثیر نمی‌گذارند (مانند ترتیب نقشه رنگی یا میزان بهینه‌سازی یک انیمیشن) گزارش نمی‌شوند. gifdiff جزئیات هر تفاوتی را که پیدا می‌کند چاپ می‌کند. اگر گیف ها یکسان باشند، چیزی چاپ نمی‌کند. در صورت عدم وجود تفاوت با وضعیت ۰، در صورت عدم وجود اختلاف با ۱ و در صورت وجود مشکل با ۲ خارج می شود.

گزینه های دستور:

- h- یا help-- نمایش راهنمای دستور
- V- یا version-- نمایش نسخه برنامه
- d- یا delay-- تعیین تاخیر بین فریم ها (بر حسب میلی ثانیه)
- l- یا loop-- تعیین تعداد دفعات تکرار انیمیشن
- o- یا output-- تعیین نام فایل خروجی
- s- یا scale-- تعیین مقیاس تصویر
- t- یا tolerance-- تعیین میزان تحمل برای تفاوت رنگ ها
- v- یا verbose-- نمایش اطلاعات بیشتر

مثال های دستور:

مقایسه دو فایل GIF با نام file1.gif و file2.gif

```
gifdiff file1.gif file2.gif
```

مقایسه دو فایل GIF با نام file1.gif و file2.gif با تاخیر ۱۰۰ میلی ثانیه بین فریم ها

```
gifdiff -d 100 file1.gif file2.gif
```

مقایسه دو فایل GIF با نام file1.gif و file2.gif با مقیاس ۵۰٪

```
gifdiff -s 50 file1.gif file2.gif
```

مقایسه دو فایل GIF با نام file1.gif و file2.gif با tolerance 10

```
gifdiff -t 10 file1.gif file2.gif
```

ذخیره نتیجه مقایسه دو فایل GIF با نام file1.gif و file2.gif در فایلی با نام output.gif

```
gifdiff -o output.gif file1.gif file2.gif
```

نکات:

می توانید از ابزار gifsicle برای ویرایش و تبدیل فایل های GIF استفاده کنید.

برای اطلاعات بیشتر در مورد gifdiff می توانید به صفحه man مربوط به این دستور (man gifdiff) مراجعه کنید.

مثال های بیشتر:

مقایسه دو فایل GIF با نام file1.gif و file2.gif و نمایش اطلاعات بیشتر

```
gifdiff -v file1.gif file2.gif
```

مقایسه دو فایل GIF با نام file1.gif و file2.gif و تکرار انیمیشن ۵ بار

```
gifdiff -l 5 file1.gif file2.gif
```

دستور gimp

gimp – یک برنامه دستکاری و رنگ آمیزی تصویر.

ساختار دستور :

```
$ gimp [-h] [--help] [--help-all] [--help-gtk] [-v] [--version] [--license] [--verbose] [-n] [--new-instance] [-a] [--as-new] [-i] [--no-interface] [-d] [--no-data] [-f] [--no-fonts] [-s] [--no-splash] [--no-shm] [--no-cpu-accel] [--display display] [--session <name>] [-g] [--gimprc <gimprc>] [--system-gimprc <gimprc>] [--dump-gimprc] [--console-messages] [--debug-handlers] [--stack-trace-mode <mode>] [--pdb-compat-mode <mode>] [--batch-interpreter <procedure>] [-b] [--batch <command>] [filename] ...
```

شرح دستور:

GIMP برنامه دستکاری تصویر است. برای ویرایش و دستکاری تصاویر استفاده می شود. این می تواند انواع فرمت های تصویر را بارگیری و ذخیره کند و می توان از آن برای تبدیل بین فرمت ها استفاده کرد. GIMP همچنین می تواند به عنوان یک برنامه رنگ استفاده شود. دارای مجموعه ای از ابزارهای طراحی و نقاشی مانند ایربراش، کلون، مداد و قلم مو. ابزارهای نقاشی و طراحی را می توان روی یک تصویر با حالت های مختلف رنگ اعمال کرد. همچنین مجموعه گسترده ای از ابزارهای انتخاب مانند مستطیل، بیضی، انتخاب فازی، انتخاب بیزیر، قیچی هوشمند و انتخاب بر اساس رنگ را ارائه می‌دهد. GIMP انواع پلاگین هایی را ارائه می‌دهد که دستکاری های تصویری مختلفی را انجام می دهند. به عنوان مثال می توان به bumpmap، edge detect، gaussian blur و بسیاری موارد دیگر اشاره کرد. علاوه بر این، GIMP چندین پسوند برنامه نویسی دارد که امکان پردازش غیر تعاملی پیشرفته و ایجاد تصاویر را فراهم می‌کند. GIMP با یک باینری دوم به نام gimp-console عرضه می شود. این باینری یک نسخه فقط کنسولی است و طوری رفتار می‌کند که گویی gimp با گزینه --no-interface command-line فراخوانی شده است.

گزینه های دستور:

- -h یا --help نمایش اطلاعات راهنما
- -n یا --new ایجاد یک تصویر جدید
- -o یا --output ذخیره تصویر
- -i یا --input باز کردن یک تصویر
- -f یا --filter اعمال فیلتر بر روی تصویر
- -s یا --script اجرای یک اسکریپت
- -b یا --batch پردازش دسته ای تصاویر
- -d یا --debug فعال کردن حالت اشکال زدایی
- -v یا --version نمایش نسخه برنامه

مثال های دستور:

ایجاد یک تصویر جدید

```
gimp -n 500x500
```

باز کردن یک تصویر

```
gimp image.jpg
```

ذخیره تصویر

```
gimp -o image.png
```

اعمال فیلتر بر روی تصویر

```
gimp -f blur image.jpg
```

اجرای یک اسکریپت

```
gimp -s script.py
```

پردازش دسته ای تصاویر

```
gimp -b batch.txt
```

گزینه های اضافی

- -a یا --antialias فعال کردن antialiasing
- -c یا --color-space تنظیم فضای رنگی
- -l یا --layers نمایش لیست لایه ها
- -m یا --mode تنظیم حالت ویرایش
- -p یا --plugins فعال کردن پلاگین ها
- -r یا --resolution تنظیم رزولوشن تصویر
- -t یا --tools نمایش لیست ابزارها

دستور lynx

یک مرورگر اطلاعاتی با هدف عمومی برای شبکه جهانی وب

شرح دستور

Lynx یک سرویس گیرنده وب جهانی ((WWWبا امکانات کامل برای کاربرانی است که از دستگاه‌های نمایشگر سلول کاراکتری قابل آدرس‌دهی نشان‌دهنده مکان‌نما استفاده می‌کنند.اسناد زبان نشانه گذاری فرامتن ((HTMLحاوی پیوندهایی به فایل های موجود در سیستم محلی و همچنین فایل های موجود در سیستم های راه دور دارای سرورهای Gopher، HTTP، FTP، WAIS و NNTP را نمایش می‌دهد. نسخه‌های فعلی Lynx بر روی Unix، VMS، Windows 95/NT، DOS۳۸۶ و OS/2 EMX اجرا می‌شوند. از Lynx می توان برای دسترسی به اطلاعات در شبکه جهانی وب یا ساخت سیستم های اطلاعاتی که عمدتاً برای دسترسی محلی در نظر گرفته شده است استفاده کرد. به عنوان **مثال**، از Lynx برای ساخت چندین سیستم اطلاعاتی پردیس ((CWISاستفاده شده است. علاوه بر این، Lynx می تواند برای ساختن سیستم های ایزوله شده در یک LAN استفاده شود.

lynx یک مرورگر وب مبتنی بر متن است که برای استفاده در ترمینال لینوکس طراحی شده است. این مرورگر از رابط کاربری گرافیکی استفاده نمی کند و صفحات وب را به صورت متن نمایش می دهد. lynx به شما امکان می دهد تا صفحات وب را مشاهده کنید، پیوندها را دنبال کنید، فرم ها را پر کنید و فایل ها را دانلود کنید.

گزینه های دستور:

- -help نمایش لیست گزینه های lynx
- -version نمایش نسخه lynx
- -display=DISPLAY استفاده از X display خاص
- -dump فقط متن صفحه را نمایش می دهد
- -force-html صفحات را به صورت HTML نمایش می دهد
- -no-numbering شماره خطوط را نمایش نمی دهد

- no-pause- بین صفحات مکث نمی کند
- reload- صفحه فعلی را دوباره بارگیری می کند
- source- کد منبع صفحه را نمایش می دهد
- URL- URL=url را باز می کند
- useragent=USERAGENT- استفاده از user agent خاص

مثال های دستور:

```
lynx google.com
```

صفحه اصلی Google را باز می کند

```
lynx -dump google.com
```

فقط متن صفحه اصلی Google را نمایش می دهد

```
lynx -force-html wikipedia.org
```

صفحه ویکی پدیا را به صورت HTML نمایش می دهد

```
lynx -no-numbering www.seilany.ir
```

فصل بیست و هشتم

فصل بیست و هشتم

عیب یابی

عیب یابی صدا

تنظیمات صدا یا بسته های مربوط به سیستم صوتی ممکن است خراب یا خراب شوند. اغلب اوقات، حذف فایل های پیکربندی، نصب مجدد بسته های مربوط به صدا، و بارگیری مجدد ALSA و/یا PulseAudio می تواند کمک کننده باشد. این دستورات همچنین می توانند به رفع تنظیمات صدا کمک کنند که «خروجی ساختگی» را به عنوان خروجی صدا نشان می دهد.

برنامه PulseAudio را ریستارت کنید:

اگر سیستم صدا را پخش نمی کند، ابتدا دایمون PulseAudio را مجدداً راه اندازی کنید:

```
systemctl --user restart pulseaudio
```

پس از راه اندازی مجدد دیمون، ممکن است نیاز به راه اندازی مجدد برنامه ها برای اتصال مجدد به PulseAudio باشد. اگر سیستم هنوز صدا پخش نمی کند، فایل های پیکربندی کاربر PulseAudio را حذف کنید:

```
rm -r ~/.config/pulse
```

سپس، تمام نمونه های PulseAudio را بکشید:

```
pulseaudio -k
```

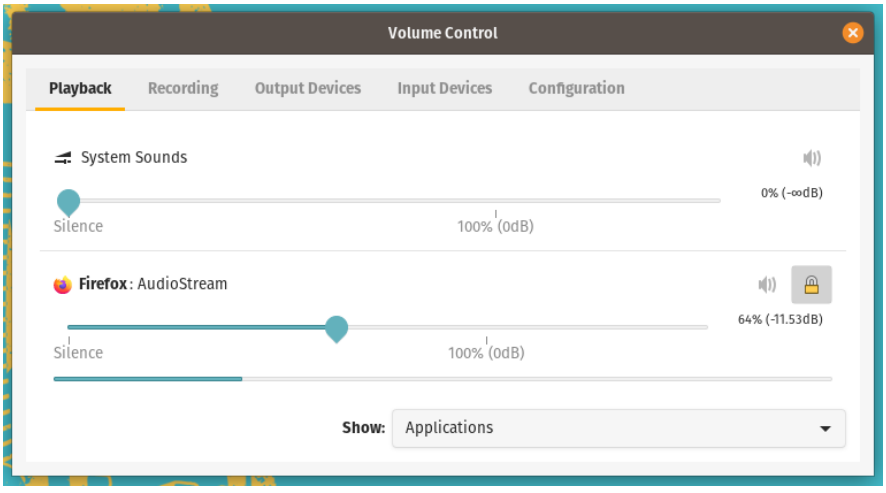
هنگامی که PulseAudio دوباره راه اندازی می شود (که باید به طور خودکار انجام شود)، فایل های پیکربندی جدیدی ایجاد می کند.

کنترل های PulseAudio را بررسی کنید:

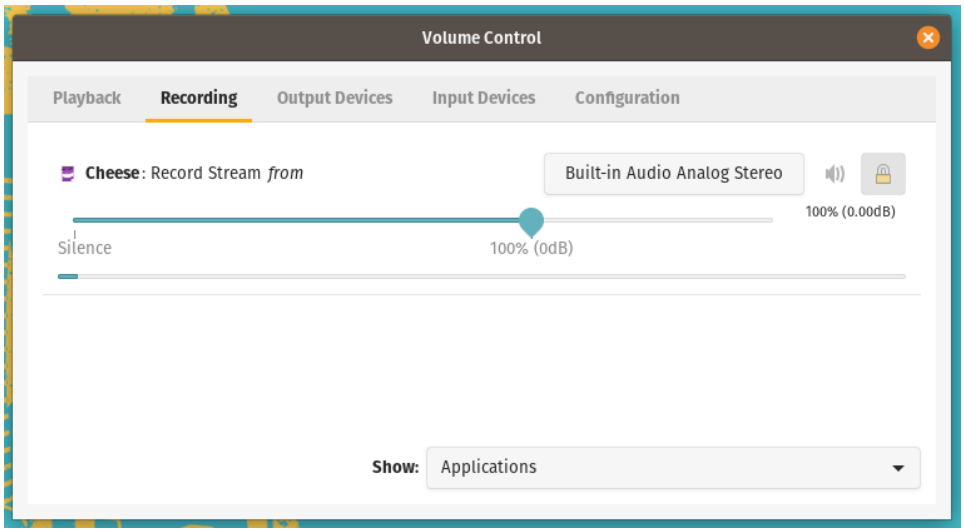
برنامه PulseAudio Volume Controlدر تشخیص اینکه کدام برنامه صدا تولید می کند، آن صدا در کجا ارسال می شود، دستگاه های ورودی/خروجی پیش فرض کدام هستند و سطح صدا روی چه میزان تنظیم شده است مفید است .

```
sudo apt install pavucontrol
$ sudo pacman -S PulseAudio
```

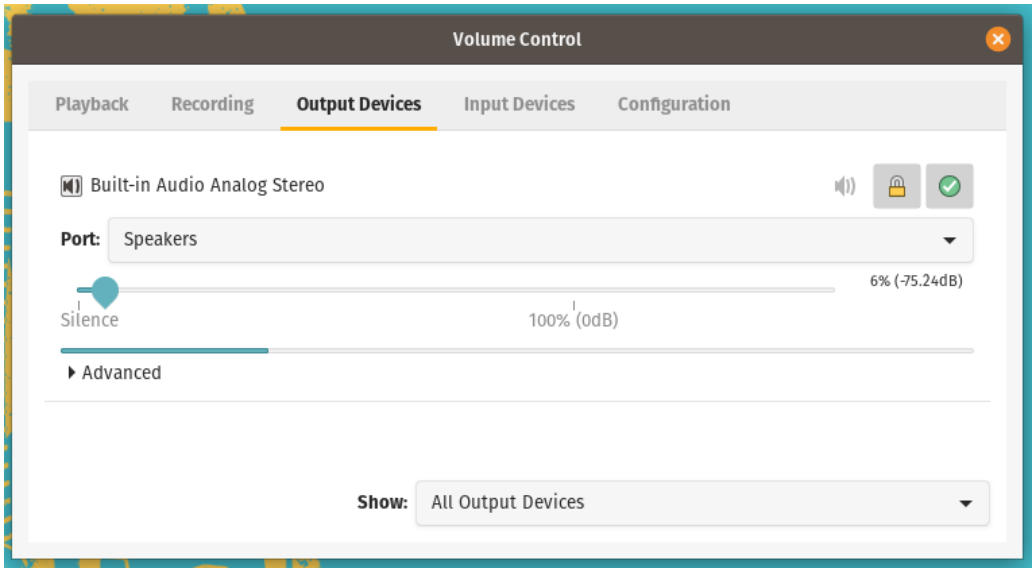
پس از نصب، برنامه PulseAudio Volume Controlرا در منوی Activities باز کنید یا pavucontrol در ترمینال اجرا کنید . برگه "پخش" لیستی از برنامه هایی را که در حال پخش صدا هستند نشان می دهد. میزان صدا برای هر برنامه را می توان به صورت جداگانه کنترل کرد.



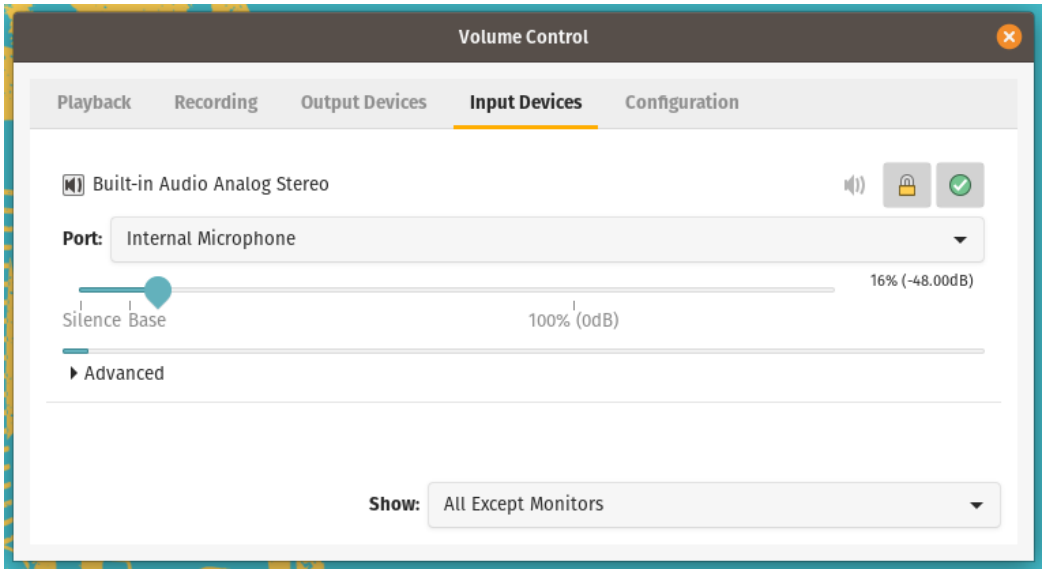
به طور مشابه، قسمت "ضبط" لیستی از برنامه هایی را نشان می دهد که در حال حاضر در حال ضبط صدا هستند.



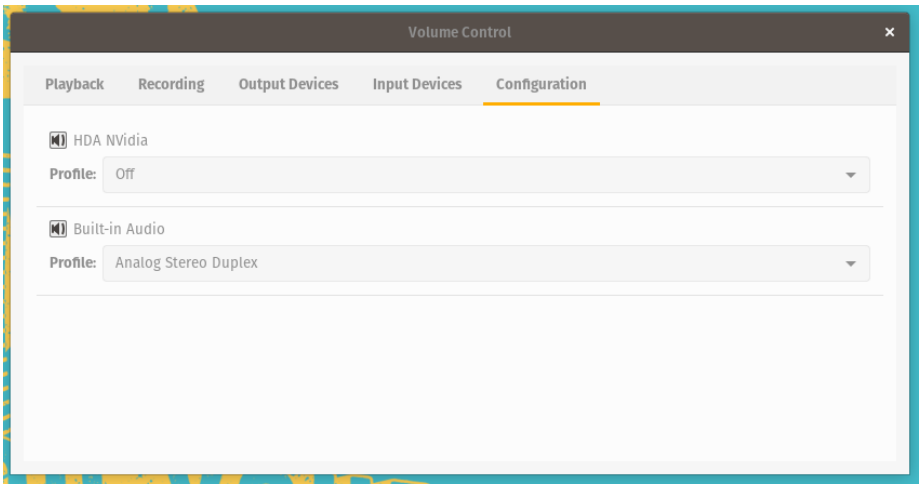
قسمت «دستگاه‌های خروجی» فهرستی از دستگاه‌های خروجی و شاخصی از آنچه در هر دستگاه پخش می‌شود را نشان می‌دهد .علامت سبز انتخاب شده نشان می دهد که دستگاهی دستگاه خروجی پیش فرض است.



قسمت «دستگاه‌های ورودی» فهرست مشابهی را برای دستگاه‌های ورودی نشان می‌دهد.



در بخش «پیکربندی»، هر کارت صدا باید فهرست شود.



عیب یابیALSA

PulseAudioدر بالای ALSA قرار دارد .اگر PulseAudio هیچ دستگاه ورودی/خروجی را نمی بیند، بررسی کنید که ALSA چه دستگاه های پخشی را شناسایی می کند:

```
aplay -l
```

یا، بررسی کنید که ALSA چه دستگاه‌های ضبط را شناسایی می‌کند:

```
arecord -l
```

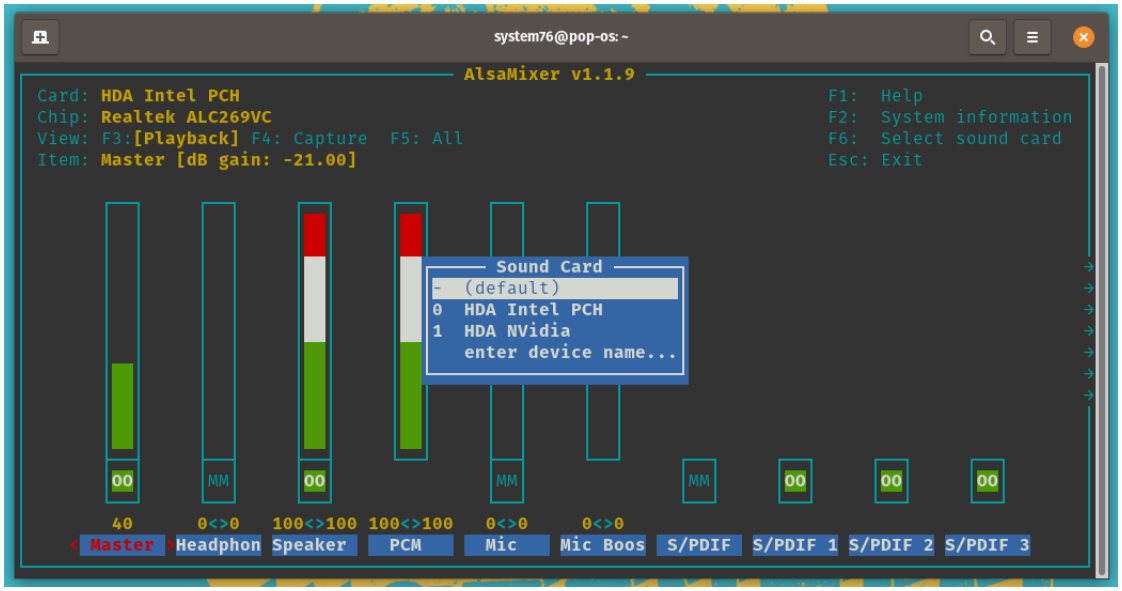
ممکن است دستگاهی در میکسر ALSA بی‌صدا شود، که هر گونه تنظیمات صدای PulseAudio را لغو می‌کند .میکسر ALSA را می توانید با این دستور باز کنید:

```
alsamixer
```

می‌توانید با استفاده از کلیدهای جهت‌نمای چپ و راست، بین ولوم‌سنج‌های مختلف حرکت کنید .هر متر را می توان با استفاده از کلیدهای جهت دار بالا و پایین تنظیم کرد .یک "MM" در پایین یک متر نشان می دهد که متر خاموش است .اگر کنترل صدای PulseAudio نشان می‌دهد که صدا در حال پخش است، اما صدایی نمی‌شنوید، سعی کنید با فشار دادن Mکلید در حالی که هر متر انتخاب شده است ، تمام ولوم سنج‌ها را در alsamixer خارج میکند .



اگر سیستم شما بیش از یک کارت صدا دارد) مثلاً یک کارت صدای اینتل و یک کارت صدای NVIDIA در سیستم‌های گرافیکی قابل تعویض(، می‌توانید با فشار دادن F6 بین آنها جابجا شوید.



بررسی میکند که آیا سخت افزار شناسایی شده است

اگر ALSA یک کارت صدا را لیست نکند، ممکن است به هیچ وجه توسط سیستم شناسایی نشود. اگر هسته لینوکس کارت صدا را ببیند، در lspciخروجی شما نمایش داده می شود. این دستور هر کارت صدایی را که سیستم شما تشخیص می دهد لیست می کند و درایور مورد استفاده برای هر یک را نشان می دهد:

```
lspci -v | grep -A6 Audio
```

اگر خروجی آن دستور خالی باشد، سیستم شما هیچ کارت صوتی را تشخیص نمی دهد (که می تواند نشان دهنده مشکل سخت افزاری باشد.)

سایر دستورات مفید

این دستور ALSA و برخی دیگر از بسته های صوتی اصلی را دوباره نصب می کند:

```
sudo apt install --reinstall alsa-base alsa-utils pulseaudio linux-sound-base libasound2
```

این دستور ماژول های درایور صدا را دوباره بارگذاری می کند:

```
sudo alsa force-reload
```

این دستور پس از توقف، PulseAudioرا شروع می کند) این معمولاً مورد نیاز نیست، زیرا PulseAudio معمولاً زمانی که متوقف می شود خود را مجدداً راه اندازی می کند:

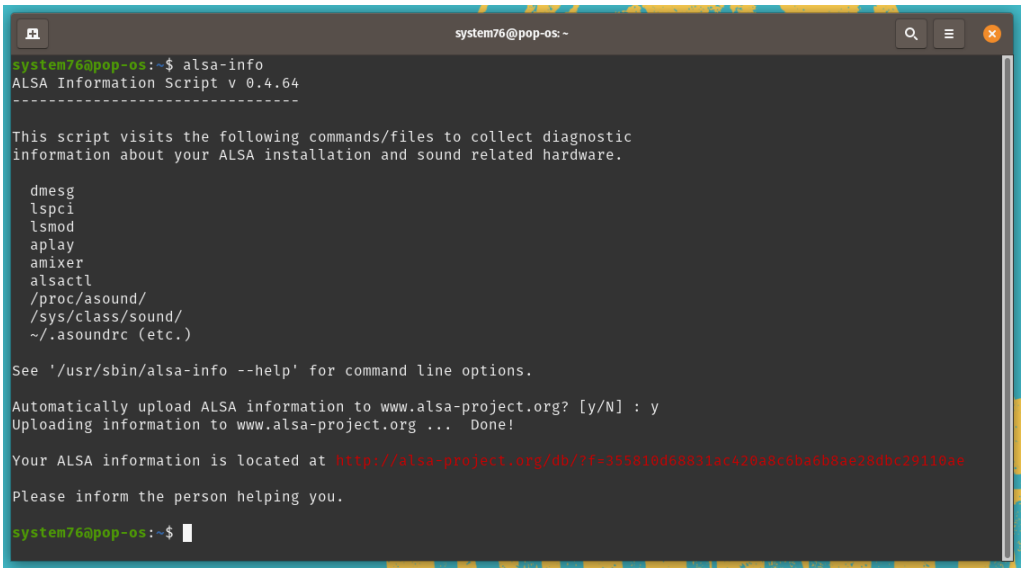
```
pulseaudio --start
```

جمع آوری اطلاعات برای پشتیبانی

این alsa-infoدستور تعدادی خروجی از جمله برخی از خروجی های ذکر شده در بالا را جمع آوری می کند و آنها را بسته بندی می کند تا به راحتی به اشتراک گذاشته شوند. در ترمینال دستور زیر را اجرا کنید:

```
alsa-info
```

هنگامی که اسکریپت جمع آوری اطلاعات به پایان رسید، تایپ کرده yو Enter را فشار دهید تا خروجی شما در وب سایت ALSAآپلود شود، سپس پیوند را به پشتیبانی ارسال کنید.



دستور lame

فایل های صوتی به mp3 تبدیل میکند.

ساختار دستور:

```
$ lame [options] <infile> <outfile>
```

شرح دستور:

LAME برنامه ای است که می توان از آن برای ایجاد فایل های صوتی فشرده استفاده کرد. (لنگ رمزگذار MP3 نیست). این فایل های صوتی را می توان توسط پخش کننده های MP3 محبوب مانند mpg123 یا madplay پخش کرد.

راهنمای دستور

```
$ man lame
```

گزینه های دستور:

- b bitrate– تنظیم bitrate فایل MP3 خروجی (بر حسب kbps). bitrate های رایج شامل ۳۲، ۶۴، ۱۲۸، ۱۹۲ و ۳۲۰ kbps هستند.
- h– نمایش راهنمای دستور lame.
- m mode– تنظیم مد رمزگذاری. مدهای رایج شامل cbr (constant bitrate)، vbr (variable bitrate) و abr (average bitrate) هستند.
- q quality– تنظیم کیفیت فایل MP3 خروجی. مقادیر بالاتر کیفیت بهتری را ارائه می دهند، اما bitrate را نیز افزایش می دهند. مقادیر رایج شامل ۰ (بهترین کیفیت) تا ۹ (بدترین کیفیت) هستند.
- V version– نمایش نسخه lame.
- f– فعال کردن فیلتر psychoacoustic. این فیلتر به طور پیش فرض فعال است و به lame اجازه می دهد تا با bitrate پایین تر به کیفیت صدای مشابه دست یابد.
- r resampling rate– تنظیم فرکانس نمونه برداری فایل MP3 خروجی. فرکانس های رایج شامل ۴۴٫۱ kHz و ۴۸ kHz هستند.
- s stereo mode– تنظیم حالت استریو. حالت های رایج شامل mono، stereo و joint stereo هستند.
- a album– تنظیم نام آلبوم برای فایل MP3 خروجی.

- t title– تنظیم عنوان آهنگ برای فایل MP3 خروجی.
- c artist– تنظیم نام هنرمند برای فایل MP3 خروجی.
- y year– تنظیم سال انتشار آهنگ برای فایل MP3 خروجی.
- g genre– تنظیم ژانر آهنگ برای فایل MP3 خروجی.

مثال های دستور:

رمزگذاری jstereo با نرخ بیت ثابت ۱۲۸ کیلوبیت بر ثانیه، بالاترین کیفیت (توصیه می شود):

```
$ lame -h sample.wav sample.mp3
```

رمزگذاری jstereo با نرخ بیت ثابت ۱۱۲ کیلوبیت بر ثانیه:

```
$ lame -b 112 sample.wav sample.mp3
```

برای غیرفعال کردن رمزگذاری مشترک استریو (کمی سریعتر، اما کیفیت کمتر در نرخ بیت => ۱۲۸ کیلوبیت بر ثانیه):

```
$ lame -m s sample.wav sample.mp3
```

رمزگذاری سریع، کیفیت پایین :

```
$ lame -f sample.wav sample.mp3
```

نرخ بیت متغیر (از –V n برای تنظیم کیفیت/اندازه فایل استفاده میکند):

```
$ lame -h -V 6 sample.wav sample.mp3
```

پخش جریانی مونو ۲۲,۰۵ کیلوهرتز خام pcm، خروجی ۲۴ کیلوبیت بر ثانیه:

```
$ cat inputfile | lame -r -m m -b 24 -s 22.05 - - > output
```

پخش جریانی مونو ۴۴,۱ کیلوهرتز خام pcm، با کاهش نمونه تا ۲۲,۰۵ کیلوهرتز:

```
$ cat inputfile | lame -r -m m -b 24 --resample 22.05 - - > output
```

با پیش تنظیم استاندارد رمزگذاری میکند:

```
$ lame --preset standard sample.wav sample.mp3
```

دستور sox

دستور sox در سیستم‌عامل‌های Unix و Linux برای پردازش و ویرایش فایل‌های صوتی به صورت خط فرمان استفاده می‌شود. sox یک ابزار قدرتمند برای ایجاد، تبدیل، و ویرایش فایل‌های صوتی است که انواع مختلفی از فرمت‌های صوتی را پشتیبانی می‌کند.

مثال های دستور:

تبدیل یک فایل صوتی به فرمت دیگر: در این مثال، یک فایل صوتی با فرمت WAV به فرمت MP3 تبدیل می‌شود.

```
sox input.wav output.mp3
```

اعمال افکت‌های صوتی به یک فایل: در این مثال، یک فایل صوتی با افکت گیتار افکت به آن اعمال می‌شود.

```
sox input.wav output.wav echo 0.8 0.88 60 0.4
```

ترکیب چندین فایل صوتی: در این مثال، چندین فایل صوتی ترکیب شده و یک فایل خروجی تولید می‌شود.

```
sox input1.wav input2.wav input3.wav output.wav
```

تغییر فرکانس یک فایل صوتی: در این مثال، فرکانس یک فایل صوتی از ۴۴۱۰۰ هرتز به ۲۲۰۵۰ هرتز تغییر می‌کند.

```
sox input.wav -r 22050 output.wav
```

دستور sox به کاربر این امکان را می‌دهد تا فایل‌های صوتی را پردازش و ویرایش کند، افکت‌های صوتی اعمال کند، فرمت‌های مختلف صوتی را تبدیل کند و دستکاری‌های دیگری روی فایل‌های صوتی انجام دهد.

دستورات عیب یابی

گرافیک:

توضیح: اطلاعات مربوط به کارت‌های گرافیک نصب شده را نمایش می‌دهد.

مثال:

```
lspci | grep -i vga
```

رفع عیب: اگر کارت گرافیک شما به درستی شناسایی نشده است، درایور مربوطه را نصب کنید.

GPU:

دستور: (nvidia-smiبرای کارت‌های گرافیک (Nvidia

توضیح: اطلاعات مربوط به GPU را نمایش می‌دهد.

مثال:

```
nvidia-smi
```

رفع عیب: اگر GPU شما به درستی کار نمی‌کند، درایور مربوطه را نصب کنید.

صدا:

توضیح: اطلاعات مربوط به کارت‌های صدا نصب شده را نمایش می‌دهد.

مثال:

```
aplay -l
```

رفع عیب: اگر کارت صدا شما به درستی کار نمی‌کند، درایور مربوطه را نصب کنید.

درایورهای سیستم:

```
دستور: lsmod
```

توضیح: لیست درایورهای نصب شده را نمایش می‌دهد.

مثال:

```
lsmod
```

رفع عیب: اگر درایور مورد نیاز شما نصب نشده است، آن را نصب کنید.

وای فای:

توضیح: اطلاعات مربوط به اتصالات وای فای را نمایش می‌دهد.

مثال:

```
iwconfig
```

رفع عیب: اگر نمی‌توانید به وای فای متصل شوید، تنظیمات وای فای خود را بررسی کنید.

صفحه کلید و موس:

توضیح: اطلاعات مربوط به دستگاه‌های ورودی (مانند صفحه کلید و موس) را نمایش می‌دهد.

مثال:

```
xinput list
```

رفع عیب: اگر صفحه کلید یا موس شما کار نمی‌کند، درایور مربوطه را نصب کنید.

صفحه تاچ:

توضیح: اطلاعات مربوط به دستگاه‌های ورودی (مانند صفحه تاچ) را نمایش می‌دهد.

مثال:

```
xinput list
```

رفع عیب: اگر صفحه تاچ شما کار نمی‌کند، درایور مربوطه را نصب کنید.

خطاهای حافظه کش:

توضیح: خطاهای مربوط به حافظه کش را نمایش می‌دهد.

مثال:

```
dmesg | grep -i cache
```

رفع عیب: اگر خطای حافظه کش دارید، حافظه کش را پاک کنید.

خطاهای بافر:

توضیح: خطاهای مربوط به بافر را نمایش می‌دهد.

مثال:

```
dmesg | grep -i buffer
```

رفع عیب: اگر خطای بافر دارید، برنامه‌ای که باعث خطا می‌شود را پیدا کنید و آن را ببندید.

خطاهای حافظه رم:

توضیح: حافظه رم را برای خطاها تست می‌کند.

مثال:

```
memtest86+
```

رفع عیب: اگر خطای حافظه رم دارید، ممکن است نیاز به تعویض رم داشته باشید.

خطاهای CPU:

توضیح: CPU را برای خطاها تست می‌کند.

مثال:

```
stress-ng
```

رفع عیب: اگر خطای CPU دارید، ممکن است نیاز به تعویض CPU داشته باشید.

عیب یابی کرنل:

توضیح: اطلاعات مربوط به کرنل را نمایش می‌دهد.

مثال:

```
dmesg
```

رفع عیب: اگر مشکلی با کرنل دارید، می‌توانید از اطلاعات dmesg برای پیدا کردن مشکل استفاده کنید.

هارد دیسک:

توضیح: اطلاعات مربوط به هارد دیسک را نمایش می‌دهد.

مثال:

```
smartctl -a /dev/sda
```

پیوست

مقایسه دستورات shell و ms-dos

DOS Command	UNIX or Bash Command	توضیحات
DIR	ls -l (or use ls -lF)(-a all files) (df -k Space remaining on filesystem)	فهرست محتویات فهرست
DIR *.* /o-d **DIR *.* /v /os** **DIR /s** **DIR /aa**	ls -tr ls -ls ls -R ls -a	فهرست محتویات فهرست بر اساس زمان معکوس تغییر/ایجاد. فهرست فایل ها و اندازه فهرست محتویات فهرست/زیر فهرست به صورت بازگشتی. لیست فایل های مخفی
TREE	ls -R	فهرست فهرست به صورت بازگشتی
CD	cd	فهرست را تغییر میدهد
MKDIR **MD**	mkdir	یک فهرست جدید میسازد
ASSIGN	ln	پیوند فایل یا فهرست ایجاد میکند
RMDIR **RD**	rmdir	یک فهرست را حذف میکند
CHDIR	pwd	نمایش محل فهرست
DEL **ERASE**	rm -iv	یک فایل را حذف میکند
RMDIR /S (NT) **DELTREE (Win 95...)**	rm -R	همه فهرست ها و فایل های زیر فهرست داده شده را حذف میکند
COPY	cp -piv	یک فایل را کپی میکند
XCOPY	cp -R	تمام فایل فهرست را به صورت بازگشتی کپی میکند
RENAME or MOVE	mv -iv	تغییر نام / انتقال یک فایل
TYPE	cat	محتویات یک فایل را روی صفحه نمایش کاربران نشان میدهد
MORE	more	خروجی pipe یک صفحه در یک زمان
HELP or COMMAND /?	man	کتابچه راهنمای آنلاین
CLS	clear ctrl-l	صفحه نمایش را پاک میکند
EXIT **EXIT0**	exit exit 0	از یک پوسته خارج شوید
FIND **FINDSTR**	grep	به دنبال کلمه ای در فایل های داده شده در خط دستور میگردد
COMP	diff	دو فایل را مقایسه میکند و تفاوت ها را نشان میدهد .همچنین comm، cmp، mgdiff و tkdiff را ببینید.
FC	diff	دو فایل را مقایسه میکند و تفاوت ها را نشان میدهد .همچنین comm، cmp، mgdiff و tkdiff را ببینید.
SET	set and env	همه متغیرهای محیط را فهرست میکند
SET *variable=value* **echo %*variable*%**	set export *variable=value* echo $*variable*	تنظیم متغیرهای محیطی نمایش متغیرهای محیط
ECHO *text*	echo *text*	اکو متن به صفحه نمایش
SET *variable*	setenv (for C shell) or export *VAR=val* (for Korn shell. Also *VAR=val*)	تنظیم متغیرهای محیطی
PATH **PATH %PATH%;C:\DIR**	echo $PATH PATH=$PATH:/dir	نمایش مسیر جستجو برای فایل های اجرایی متغیر محیطی PATH را تنظیم میکند.
PROMPT pg	export PS1='\h(\u)\W> '	خط فرمان کاربر را تنظیم میکند.
DATE or TIME	date	نمایش تاریخ)همچنین تاریخ تنظیم – فقط DOS)
DOSKEY /h	history	تاریخچه فرمان را فهرست میکند

DOSKEY *NAME=command*	alias *NAME=command*	نام مستعار دستور را تنظیم میکند
BREAK ON	trap	تله گذاری / ctrl-break تله گذاری سیگنال ها.
SORT	sort	مرتب سازی داده ها بر اساس حروف الفبا/عددی
EDLIN	ed	ویرایشگر حالت خط
EDIT *filename.txt*	pico, gnp, vi, xedit, xemacs,dtpad	یک فایل را ویرایش میکند .ویرایشگر لینوکس که بیشتر شبیه ویرایش DOS است احتمالاً Pico است) .توصیه نیست(!
BACKUP files A:	tar -cvf /dev/fd0 files mdir, mcopy doswrite -a file (AIX only)	فایل ها را در فلاپی ذخیره میکند .استفاده از فلاپی های DOS را ببینید
RESTORE A:\ files	tar -xvf /dev/fd0 files mdir, mcopy dosread -a file (AIX only)	خواندن فایل ها از فلاپی استفاده از فلاپی های DOS را ببینید
ATTRIB [+r|-r] [+a|-a] [+s|-s] [path\file] /s	chmod	تغییر مجوزهای فایل DOS: +:تنظیم به r:remove -: فقط خواندن s:System /s: a:Archive به صورت بازگشتی
ATTRIB +h or -h	mv *file .file*	تغییر فایل به یک فایل مخفی – تغییر نام فایل با پیشوند"."
PRINT	lpr	یک فایل چاپ میکند
CALL COMMAND /C (DOS), CMD (NT)	source *script* (cshrc) . *script* (bash) sh *script*	اسکریپت را از داخل پوسته دسته ای اجرا میکند.
MEM	free top	نمایش حافظه آزاد در سیستم
TASKLIST (WIN2K, XP)	ps -aux top	نام فایل اجرایی، شماره شناسه پردازش و استفاده از حافظه فرآیندهای فعال را فهرست میکند
MSD	lsdev	نمایش اطلاعات سیستم) دستور قرض گرفته شده از(AIX
SCANDISK DEFRAG C:	fsck debugfs	سیستم فایل هارد دیسک را بررسی و تعمیر میکند
CHDISK	du -s	استفاده از دیسک
FDISK	fdisk	ابزاری برای پارتیشن بندی هارد دیسک.
SUBST V: C:\directory\path	mount	یک نامه درایو را به پوشه/فهرست روی هارد دیسک خود نصب میکند.
FORMAT	mke2fs See: fdformat and mformat for floppies	سیستم فایل درایو را فرمت میکند. برای استفاده از فلاپی درایو، آموزش YoLinux با استفاده از فلاپی های DOS با لینوکس را ببینید
VER	uname -a echo $SHELL cat /etc/issue	سیستم عامل/نسخه پوسته
pkzip	tar and zip	فشرده سازی و از حالت فشرده خارج کردن فایل ها / فهرست ها .از tarبرای ایجاد کامپایل یک فهرست قبل از فشرده سازی استفاده کنید .لینوکس همچنین دارای فشرده سازی ، gzipاست
HOSTNAME	hostname	چاپ نام میزبان کامپیوتر
PING	ping	ارسال بسته ها به میزبان شبکه
TRACERT	traceroute	نمایش مسیرها و پرش روتر به مقصد شبکه داده شده.
IPCONFIG (NT) WINIPCFG (Win 95...)	ifconfig or ip addr show	نمایش/پیکربندی رابط شبکه
NBTSTAT (Netbios info: -n, -c) NBTSTAT -A *IP-address*	nslookup *host-name* host *host-name*	چاپ اطلاعات DNS برای میزبان.
NBTSTAT -a hostname	nmblookup -A hostname	جستجوی نام هایNetBIOS
ROUTE PRINT	route -n	چاپ جدول مسیریابی
NET HELP START	chkconfig --list |grep on	لیست خدمات
NET START*service-name* NET STOP*service-name*	service *service-name* start service *service-name* stop	شروع/توقف سرویس/دیمون.

NET SHARES	df	نمایش اشتراک‌ها/فایل سیستم‌های نصب شده
NET SEND <*node-name*> <*message*> (NT)	smbclient -M *MS-Windows-host-name* talk	ارسال پاپ آپ به کامپیوترMS/Windows ارسال پیام به کاربر یونیکس/لینوکس دیگر .آموزش یولینوکس را ببینید
WIN	startx	X-Windowsرا راه اندازی میکند.
REBOOT	shutdown -r now	دوباره بوت شدن سیستم.

Devices

DOS Device	Linux Device	Description
NUL	/dev/null	Send into nothingness
CON	stdin	stdin from console
PRN ***LPT1***	/dev/lp0	First printer device
COM1	/dev/ttyS0	Firsst serial port

ارزیابی برنامه های رایج لینوکس و ویندوز

MS/Windows Command	Linux Command	Description
C:\WINDOWS\cmd	gnome-terminal konsole	Command Text Terminal
C:\WINDOWS\explorer	nautilus --no-desktop	File browser
c:\Program Files\Internet Explorer\iexplore	firefox mozilla	Web browser
C:\WINDOWS\notepad ***C:\Program Files\Windows NT\Accessories\wordpad***	gedit	Text editor
C:\Program Files\Microsoft Office\Office10\winword ***excel*** ***powerpnt***	oowriter oocalc ooimpress	MS/Office and Open Office suites (ooffice)
C:\Program Files\Adobe\Acrobat 7.0 Reader\acrord32	acroread	Adobe PDF viewer

LEARNING HIVE

contact me:

seilany.ir
learninghive.ir
emperor-os.ir
Hubuntu.ir
predator-os.ir
telegram:@seilany

HosseinSeilani

Designer,Developer and Linux sysadmin



Founder and Developer of Emperor-OS, Hubuntu and Predator-OS. I bring significant experience as a Linux/Windows sysadmin and graphical web design,UX/UI to the Open Source Community.

LITTLE-PSYCHO
DESTRUCTIVE LINUX

Emperor-OS

Predator-OS