



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 سایر فصل ها بر حسب کاربرد دستورات نسبت به هم نوشته شده اند

شرح دستورات

سافتار دستورات

گزینه های دستورات

مثال های کامل دستورات

مرجع

کامل

دستورات

لینوکس

هزار و یک دستور لینوکس

1001 Linux commands

حسین سیلانی
ویرایش سوم

۱۴۰۳

سرشناسه	لطفی، فهیمه، ۱۳۶۷-
عنوان و نام پدیدآور	هزار و یک دستور لینوکس / ۱۰۰۱ Linux commands/ فهیمه لطفی، حسین سیلانی؛ ویراستاری نسیم ناصری مقدم.
مشخصات نشر	تهران: کیان دانش، ۱۴۰۳.
مشخصات ظاهری	۷۳۳ ص.
شابک	۹۷۸-۶۲۲-۴۰۰۰-۱۵-۶
وضعیت فهرست نویسی	فیا:
موضوع	سیستم عامل لینوکس Linux
موضوع	سیستم‌های عامل (کامپیوتر) Operating systems (Computers)
شناسه افزوده	سیلانی، حسین، ۱۳۶۵-
رده بندی کنگره	QAV۶/۷۶:
رده بندی دیویی	۰۰۵/۲۳۳:
شماره کتابشناسی ملی	۹۶۷۳۵۹۳:
اطلاعات رکورد کتابشناسی	فیا:

نشر کیان دانش: ناشر کتاب‌های دانشگاهی



عنوان: هزار و یک دستور لینوکس
 تالیف: فهیمه لطفی، حسین سیلانی
 صفحه آرای و ویراستاری: نسیم ناصری مقدم
 چاپ و صحافی: موسسه مهراد
 طراحی جلد: امین حیدری
 چاپ اول: ۱۴۰۳
 شمارگان: ۱۰۰ جلد
 شابک: ۹۷۸-۶۲۲-۴۰۰۰-۱۵-۶
 حق چاپ محفوظ است.

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

فهرست مطالب

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

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
user binaries : /bin	۵۴
:system binaries /sbin	۵۴
/etc:configuration files	۵۴
فهرست‌های موجود در /etc	۵۴
/etc/apt/sources.list	۵۵
device files /dev:	۵۵
/proc: process information	۵۵
variable files /var :	۵۵
temporary files /tmp:	۵۵
/usr: user programs	۵۵
/home : home directories	۵۵
/boot: boot loader files	۵۵
: system libraries /lib	۵۵
/opt: optional add-on applications	۵۵
/mnt: mount directory	۵۶
removable media devices : /media	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

۵۶	_____ /etc/hosts
۵۶	_____ /etc/passwd
۵۶	_____ /etc/printcap
۵۶	_____ /etc/x11
۵۶	_____ /etc/resolv.conf
۵۶	_____ /usr/share
۵۷	_____ /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
۵۸	_____ توضیح مفاهیم و فایل‌های بوت
۵۸	_____ فایل‌های اصلی سیستم
۵۹	_____ تنظیمات شبکه
۵۹	_____ تنظیمات بوت و سرویس‌ها
۵۹	_____ فایل میزبان
۵۹	_____ مسیرهای مربوط به SSH
۶۰	_____ مسیرهای مربوط به مدیریت بسته
۶۰	_____ مسیرهای دیگر
۶۱	_____ 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
۶۲	انواع سیستم فایل پارتیشن
۶۲	سیستم فایل Ext2
۶۲	سیستم فایل Ext3
۶۲	سیستم فایل Ext4
۶۲	سیستم فایل XFS
۶۳	سیستم فایل Btrfs
۶۳	سیستم فایل ZFS
۶۳	مشاهده نوع سیستم فایل
۶۳	دستورات XFS
۶۳	دستور xfs_info
۶۴	دستور xfs_check
۶۴	دستور xfs_repair
۶۴	دستور xfsdump و xfsrestore
۶۵	دستور mke2fs
۶۵	ایجاد سیستم فایل Ext2
۶۵	ایجاد سیستم فایل Ext3
۶۶	ایجاد سیستم فایل Ext4
۶۶	تبدیل سیستم فایلها
۶۷	دستور fsck: بررسی و تعمیر سیستم فایل
۶۸	مثال های عملی برای دستور fsck با استفاده از گزینه های مختلف
۶۹	دستورات مدیریت آرایه های RAID
۶۹	دستور mdadm
۶۹	توضیحات و مثال های بیشتر برای گزینه های دستور mdadm
۷۲	مفهوم ژورنالینگ
۷۲	ویژگی های سیستم فایل ژورنالی
۷۲	سیستم فایل ReiserFS
۷۲	ویژگی های سیستم فایل ReiserFS
۷۳	سیستم فایل XFS
۷۳	ویژگی های سیستم فایل XFS
۷۳	سیستم فایل Btrfs
۷۳	ویژگی های سیستم فایل Btrfs

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

۱۰۵	-----
۱۰۶	----- فصل دوم -----
۱۰۶	----- آناتومی دستور -----
۱۰۷	----- دستورات داخلی شل (پوسته) -----
۱۰۷	----- توابع درون پوسته -----
۱۰۷	----- برنامه های اجرایی -----
۱۰۸	----- روش تایپ دستورات در ترمینال -----
۱۰۹	----- انواع پوسته -----
۱۱۰	----- پوسته بورن -----
۱۱۰	----- پوسته C Shell -----
۱۱۱	----- پوسته Korn -----
۱۱۲	----- پوسته GNU Bourne-Again -----
۱۱۲	----- پوسته Z -----
۱۱۲	----- پوسته ماهی-fish -----
۱۱۳	----- روند رشد و تکامل شل های لینوکس -----
۱۱۳	----- نمایش شل های نصب شده -----
۱۱۵	----- مشاهده شل فعال جاری -----
۱۱۸	----- تغییر پوسته پیش فرض -----
۱۱۸	----- دستور type -----
۱۱۹	----- دستور which -----
۱۲۲	----- فصل سوم -----
۱۲۲	----- ترمینال های مجازی -----
۱۲۲	----- تاریخچه و مفهوم ترمینال مجازی -----
۱۲۳	----- راه اندازی ترمینال مجازی -----
۱۲۴	----- پیمایش ترمینالهای مجازی -----
۱۲۵	----- دستور chvt -----
۱۲۶	----- دستور screen -----
۱۲۸	----- فصل چهارم -----
۱۲۸	----- تنظیمات bash -----
۱۲۸	----- دستور export -----
۱۲۹	----- متغیر های محیطی -----
۱۲۹	----- تنظیمات محلی و گلوبال -----
۱۳۰	----- ایجاد و حذف یک متغیر محیطی لوکال -----

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

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

۲۰۰	دستور sha512sum
۲۰۰	مشاهده محل اجرا دستورات
۲۰۳	فصل ششم
۲۰۳	مدیریت بسته ها
۲۰۳	مدیر بسته apt
۲۰۴	پیدا کردن فایل های نصب شده با apt
۲۰۴	جستجو برای بسته با apt
۲۰۵	مشاهده اطلاعات مربوط به بسته با apt
۲۰۵	تأیید بسته برای هر وابستگی ناقص با apt
۲۰۶	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
۲۰۷	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
۲۰۸	نصب بسته ها با پسوند .deb با apt
۲۰۸	راهنمای دستور apt
۲۰۹	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
۲۱۰	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	مدیر بسته dnf
۲۱۱	نحوه عملکرد dnf
۲۱۱	نصب مدیر بسته dnf
۲۱۱	نصب بسته و وابستگی های با dnf
۲۱۱	حذف یک بسته و وابستگی ها با dnf

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

- ۲۲۰ _____ pacman با سیستم عامل با
- ۲۲۰ _____ pacman با بسته به همراه یک بسته با
- ۲۲۰ _____ pacman با بسته های نصب شده با
- ۲۲۰ _____ pacman با بسته های نصب شده به همراه برنامه های وابسته با
- ۲۲۰ _____ pacman با نصب شدن با
- ۲۲۰ _____ pacman با حذف بسته های دانلود شده با
- ۲۲۰ _____ dpkg مدیر بسته
- ۲۲۱ _____ dpkg توضیحات گزینه های اضافی دستور
- ۲۲۳ _____ dpkg-reconfigure دستور
- ۲۲۴ _____ نصب میزهای کارلینوکسی
- ۲۲۵ _____ محیط های مختلف دسکتاپ
- ۲۲۵ _____ MATE میز
- ۲۲۶ _____ Cinnamon میز
- ۲۲۷ _____ GNOME میز
- ۲۲۸ _____ میز پلاسما
- ۲۲۹ _____ UKUI میز
- ۲۳۰ _____ XFCE میز
- ۲۳۱ _____ LXDE میز
- ۲۳۲ _____ LXQt میز

فصل هفتم ۲۳۴

- ۲۳۴ _____ شخصی سازی میزهای لینوکس
- ۲۳۴ _____ سفارشی سازی میز گنوم
- ۲۳۴ _____ gsettings دستور
- ۲۳۴ _____ تنظیم تصویر پس زمینه میز گنوم
- ۲۳۴ _____ تنظیم نمایش آیکن های میز کار میز گنوم
- ۲۳۴ _____ سفارشی کردن گفتگوی اعلان
- ۲۳۵ _____ GNOME Shell راه اندازی مجدد
- ۲۳۵ _____ نصب ابزارهای ضروری
- ۲۳۵ _____ غیرفعال کردن لیست کاربران
- ۲۳۵ _____ قفل شدن سیستم و رفتن به حالت لاگین
- ۲۳۶ _____ hibernate, sleep, hybrid sleep غیرفعال سازی و متوقف کردن انواع
- ۲۳۶ _____ screen saver فعال کردن

۲۳۷	دستور pactl
۲۳۷	غیرفعال کردن بلوتوث
۲۳۷	غیرفعال کردن وب کم
۲۳۷	غیرفعال کردن میکروفن
۲۳۷	نصب درایور تاج پد
۲۳۷	نصب درایور پرینتر
۲۳۷	نصب درایور اسکنر
۲۳۹	فصل هشتم
۲۳۹	پردازش متون
۲۳۹	دستور nl
۲۴۰	دستور head
۲۴۲	دستور tail
۲۴۳	دستور cat
۲۴۳	مشاهده محتوای یک فایل
۲۴۴	مشاهده محتوای چندین فایل
۲۴۴	مشاهده محتویات یک پرونده قبل با شماره خط
۲۴۴	ارسال محتوای یک پرونده را در پرونده دیگر
۲۴۶	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
۲۴۷	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
۲۴۸	دستور aspell
۲۵۱	دستور column
۲۵۳	دستور column
۲۵۴	دستور rev
۲۵۵	دستور Sort

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

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

۳۳۵	VeraCrypt محافظتی قدرتمند برای حریم خصوصی شما
۳۳۵	نحوه نصب VeraCrypt در لینوکس
۳۳۵	دستور quota
۳۳۶	نحوه پاک کردن Swap Space
۳۳۶	دستور mkswap
۳۳۸	دستور parted
۳۳۹	دستور Baobab
۳۳۹	دستور ddrescue
۳۴۴	فصل دهم
۳۴۴	دریافت اطلاعات سخت افزاری در linux
۳۴۴	دستور Dmidecode
۳۴۵	دریافت انواع DMI
۳۴۵	دریافت اطلاعات memory
۳۴۶	دریافت اطلاعات BIOS
۳۴۶	بدست آوردن اطلاعات سیستم
۳۴۷	بدست آوردن اطلاعات سخت افزاری
۳۴۷	بدست آوردن کلاس دستگاه H / W
۳۴۸	بدست آوردن اطلاعات CPU
۳۴۸	بدست آوردن اطلاعات مربوط به بلوک ها
۳۴۹	بدست آوردن اطلاعات پردازنده
۳۵۰	دستور cpufreq-info
۳۵۲	دستور lscpu
۳۵۳	دستور mpstat
۳۵۴	دستور lnx
۳۵۵	دستور lshw
۳۵۵	دستور hardinfo
۳۵۷	دستور hwinfo
۳۵۷	دستور lshal
۳۵۷	دستور l-Nex
۳۵۸	دستور hdparm
۳۶۱	دستور sdparm
۳۶۲	دستور ftrim

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

۴۰۲	دستور iwconfig
۴۰۴	دستور iwevent
۴۰۵	دستور chrt
۴۰۸	دستور top
۴۰۸	مرتب سازی فرآیندهای لینوکس را بر اساس PID
۴۱۹	دستور htop
۴۱۹	نصب htop
۴۲۳	دستور monit
۴۲۳	دستور iotop
۴۲۵	دستور nice/renice
۴۲۵	دستور ionice
۴۲۶	دستور strace
۴۲۷	دستور Wait
۴۲۷	دستور kill
۴۲۸	دستور nohup
۴۲۹	دستور pstree
۴۳۱	دستور ps
۴۳۳	دستور tree
۴۳۵	دستور killall
۴۳۶	دستور login
۴۳۷	دستور ulimit
۴۳۸	مثال های کاربردی
۴۳۹	دستور lsof
۴۴۴	دستور fuser
۴۴۵	دستور tuptime
۴۴۵	نصب ابزار tuptime
۴۴۷	دستور Watch
۴۵۱	فصل دوازدهم
۴۵۱	مدیریت سرویس ها و اجرای سطوح در لینوکس
۴۵۱	دستور chkconfig
۴۵۳	مدیر سیستم systemd
۴۵۳	مدیریت سرویس ها

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

دستور cron ۴۸۲

فرمت Crontab ۴۸۳

دستور anacron ۴۸۵

دستور bg ۴۸۶

دستور atq ۴۸۷

دستور جایگزین برای atq ۴۸۸

دستور atrm ۴۸۸

دستور جایگزین برای atrm ۴۸۹

فصل چهاردهم ۴۹۱

مجوزها ۴۹۱

دستور chmod ۴۹۱

دستور chown ۴۹۳

دستور chattr ۴۹۵

دستور chgrp ۴۹۸

دستور getfacl ۵۰۰

دستور setfacl ۵۰۲

دسترسی به حساب root ۵۰۳

دستور chage ۵۰۵

دستور file ۵۰۸

ماژول های تأیید هویت ۵۱۰

دستور setenforce ۵۱۱

دستور chcon ۵۱۲

فصل پانزدهم ۵۱۴

۵۱۵

مدیریت چندرسانه ای ۵۱۵

دستور amixer ۵۱۵

دستور aplay ۵۲۰

دستور aplaymidi ۵۲۲

دستور ffmpeg ۵۲۳

دستور spd-say ۵۲۴

دستور cdrecord ۵۲۸

دستور رایت CD/DVD ۵۲۹

دستور brasero ۵۲۹

۵۳۱ ----- دستور mkisofs

۵۳۳ ----- دستور genisoimage

۵۳۴ ----- دستور eject

۵۳۹ ----- فصل شانزدهم

۵۳۹ ----- دستور find

۵۴۴ ----- دستور locate

۵۴۵ ----- دستور updatedb

۵۴۸ ----- فصل هفدهم

۵۴۸ ----- مدیریت شبکه

۵۴۸ ----- دستور hostname

۵۴۹ ----- ایجاد نام هاست دائمی

۵۴۹ ----- دستور host

۵۵۱ ----- دستور domainname

۵۵۲ ----- تنظیمات شبکه

۵۵۲ ----- دستور ifconfig

۵۵۳ ----- مشاهده تمام تنظیمات شبکه

۵۵۵ ----- دستور ethtool

۵۵۶ ----- دستور addresses

۵۵۷ ----- دستور ip

۵۶۲ ----- دستور fail2ban

۵۶۳ ----- دستور ufw

۵۶۴ ----- دستور iptables

۵۶۷ ----- دیگر ابزارهای شبکه

۵۶۷ ----- تعریف ip ثابت

۵۷۰ ----- دستور arp

۵۷۳ ----- دستور arp-scan

۵۷۴ ----- دستور traceroute

۵۷۶ ----- دستور tracepath

۵۷۸ ----- دستور mtr

۵۸۱ ----- دستور nmcli

۵۸۳ ----- دستور tcpdump

۵۸۷ ----- دستور Wget

۵۹۱	دستور ping
۵۹۶	دستور aria2c
۵۹۸	دستور ss
۶۰۳	دستور sar
۶۰۴	دستور dstat
۶۰۶	دستور nmap
۶۰۹	دستور snort
۶۱۱	دستور nc
۶۱۳	دستور iwconfig
۶۱۴	دستور iwlist
۶۱۶	دستور RFKILL
۶۱۷	دستور dig
۶۱۸	دستور drill
۶۲۰	دستور xdg-open
۶۲۱	دستور whois
۶۲۳	دستور ssh
۶۲۵	دستور sftp
۶۲۶	دستور rcp
۶۲۷	دستور rlogin
۶۲۸	دستور agetty
۶۳۲	فصل هجدهم
۶۳۲	مدیریت پرینتر
۶۳۲	دستور lpinfo
۶۳۳	دستور lpstat
۶۳۴	دستور lpc
۶۳۵	دستور lpadmin
۶۳۶	دستور lp
۶۳۷	دستور LPR
۶۳۸	دستور lpstat
۶۳۹	دستور lprm
۶۳۹	دستور lpq
۶۴۰	دستور cupsctl

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

دستور smem ۶۴۷

دستور exiftool ۶۴۸

فصل بیستم ۶۵۱

مدیریت فایل های آرشیو و پشتیبانی ۶۵۱

مدیریت فایل های آرشیو دستور tar ۶۵۱

ایجاد آرشیو فایل tar ۶۵۱

ایجاد فایل آرشیو tar.bz2 ۶۵۲

ایجاد فایل آرشیو tar و Untar ۶۵۲

فهرست محتوای فایل tar ۶۵۳

فهرست محتوای فایل tar.gz ۶۵۳

فهرست محتوای فایل tar.bz2 ۶۵۴

بررسی اندازه فایل های آرشیو ۶۵۵

دستور gzip ۶۵۵

دستور zcat ۶۵۶

دستور ar ۶۵۷

دستور bzip2 ۶۶۲

دستور arkr ۶۶۳

دستور unrar ۶۶۴

استخراج فایل war ۶۶۶

دستور unzip ۶۶۶

دستور backintime ۶۶۷

دستور djpeg ۶۶۸

دستور xz ۶۶۹

فصل بیست و یکم ۶۷۳

برنامه نویسی ۶۷۳

دستور make ۶۷۳

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

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

۷۴۹ _____ رمزگذاری / رمزگشایی در لینوکس

۷۴۹ _____ دستور Ccrypt

۷۵۰ _____ دستور GnuPG

۷۵۰ _____ دستور cryptdir

۷۵۱ _____ دستور flock

۷۵۴ _____ فصل بیست و پنجم

۷۵۴ _____ ماژول‌های کرنل

۷۵۴ _____ دستور lsmod

۷۵۵ _____ دستور modinfo

۷۵۷ _____ دستور insmod

۷۵۹ _____ دستور rmmod

۷۶۰ _____ دستور modprobe

۷۶۱ _____ دستور depmod

۷۶۲ _____ اضافه کردن پارامترهای کرنل

۷۶۳ _____ دستور grub-install

۷۶۴ _____ دستور grub-mkconfig

۷۶۶ _____ initramfs چیست؟

۷۶۶ _____ چرا initramfs مهم است؟

۷۶۶ _____ اجزای اصلی: initramfs

۷۶۶ _____ فرآیند بوت شدن با استفاده از: initramfs

۷۶۶ _____ دستور update-initramfs

۷۶۹ _____ فصل بیست و ششم

۷۶۹ _____ مدیریت لاگ‌ها

۷۶۹ _____ لاگ در لینوکس چیست؟

۷۷۳ _____ دستورات مدیریت لاگ‌ها

۷۷۵ _____ دستور journalctl

۷۷۸ _____ نمایش پیام‌های بوت سیستم

۷۸۰ _____ دستور logrotate

۷۸۲ _____ فصل بیست و هفتم

۷۸۲ _____ دستورات سرگرم‌کننده در Linux

۷۸۲ _____ دستور orca

۷۸۳ _____ دستور xeyes

۷۸۴ _____ دستور feh

۷۸۶	دستور Cowsay
۷۸۷	دستور xcowsay
۷۸۸	دستور cmatrix
۷۸۹	دستور fire
۷۹۱	دستور sl (لوکوموتیو بخار)
۷۹۲	دستور banner
۷۹۳	دستور bc
۷۹۵	دستور yes
۷۹۶	دستور gifdiff
۷۹۷	دستور gimp
۷۹۸	دستور lynx

فصل بیست و هشتم ۸۰۱

۸۰۱	عیب یابی
۸۰۱	عیب یابی صدا
۸۰۳	عیب یابی ALSA
۸۰۵	دستور lame
۸۰۶	دستور sox
۸۰۷	دستورات عیب یابی

پیوست ۸۰۹

۸۰۹	مقایسه دستورات shell و ms-dos
۸۱۱	Devices
۸۱۱	ارزیابی برنامه های رایج لینوکس و ویندوز

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 از آنها در مرحله 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>

تقديم به تو

يگانه 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 توروالدز 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 واسطه 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 توروالدز :** 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 توروالدز 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.

نرم افزار متن باز (Open Source)

- 🐧 **کد منبع عمومی**: مشابه نرم افزار آزاد، کد منبع نرم افزار متن باز نیز به صورت عمومی در دسترس است.
- 🐧 **همکاری جمعی**: نرم افزارهای متن باز معمولاً توسط جامعه‌ای از توسعه‌دهندگان به صورت همکاری ایجاد و بهبود می‌یابند.

"آزاد و متن باز" 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 نرم افزار، زمینه را برای نوآوری و ایجاد ویژگی‌های جدید فراهم کرده است.
- 🐧 **پایداری**: جامعه بزرگ توسعه‌دهندگان به رفع مشکلات و بهبود امنیت لینوکس کمک می‌کند.
- 🐧 **انعطاف پذیری**: لینوکس می‌تواند برای طیف گسترده‌ای از کاربردها، از سیستم‌های کوچک تا ابر کامپیوترها، استفاده شود.
- 🐧 **کاهش هزینه**: از آنجایی که لینوکس یک نرم افزار آزاد است، هزینه‌های مربوط به خرید مجوز و پشتیبانی آن بسیار پایین است.

مفاهیم پایه

- 🐧 **Free Software Foundation (FSF)**: بنیاد نرم افزار آزاد، سازمانی است که برای ترویج و دفاع از نرم افزار آزاد تلاش می‌کند.
- 🐧 **Open Source Initiative (OSI)**: ابتکار نرم افزار متن باز، سازمانی است که تعریف نرم افزار متن باز را ارائه می‌دهد و مجوزهای متن باز را تایید می‌کند.
- 🐧 **Licenses**: مجوزها قوانینی هستند که شرایط استفاده، تغییر، توزیع و دیگر حقوق مرتبط با نرم افزار 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**: نرم افزار حوزه عمومی نرم افزاری است که هیچ حق مالکیتی بر آن وجود ندارد و هر کسی می‌تواند آن را بدون محدودیت استفاده، تغییر و توزیع کند.
- 🐧 **Freeware**: نرم افزار رایگان نرم افزاری است که بدون هزینه قابل استفاده است اما ممکن است محدودیت‌هایی در استفاده، تغییر یا توزیع آن وجود داشته باشد.
- 🐧 **Shareware**: نرم افزار اشتراکی نرم افزاری 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 نرم افزاری و حل اختلافات مرتبط با آن را بر عهده دارند.

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 توروالدز در سال ۱۹۹۱ ایجاد شدند. چیز جالب این است که هر کسی 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 (پرسروپرنس روور، کپسول خانواده داراگون و موشک فالکن ۹)، خودروها (تویوتا، هیوندای، مرسدس بنز، آودی و تسلا)، تلویزیون‌ها (تلویزیون‌های هوشمند ال‌جی و سامسونگ)، کنسول‌های بازی و 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.

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

اس. تانباوم بود. این کتاب حاوی یک نسخه از سیستم عامل مینیکس تانباوم بود. این با این دوره بود که توروالدز ابتدا با یونیکس آشنا شد. او در سال ۱۹۹۱ به سیستم عامل‌ها علاقه مند شد. او از تنگنایی مجوز مینیکس که آن را در آن زمان تنها برای استفاده آموزشی محدود می کرد، ناراحت بود و شروع به کار روی هسته سیستم عامل خود کرد که در نهایت هسته لینوکس شد.

توروالدز شروع به توسعه هسته لینوکس بر روی مینیکس کرد و نرم افزار نوشته شده برای مینیکس هم بر روی لینوکس استفاده شد. بعداً، لینوکس توسعه یافت و سپس توسعه هسته لینوکس بر روی سیستم های لینوکس ظاهر شد. همچنین، برنامه های GNU تمامی مؤلفه های مینیکس را به خاطر اینکه از کد آزاد استفاده کردند، جایگزین کردند؛ از طریق پروژه GNU با مجوز GNU GPL، کد می تواند 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 می کرد، به GNU GPL توسعه دهندگان سعی کردند که با هسته لینوکس، عناصر GNU را توسعه دهند، که سیستم عاملی آزاد و کاملاً کارآمد ایجاد می کند.

پذیرش محبوب و تجاری

در محیط های تولید، پذیرش لینوکس ابتدا در اوایل دهه ۹۰ میلادی در جامعه محاسباتی فوق العاده آغاز شد به جای استفاده تنها توسط علاقمندان، جایی که سازمان هایی مانند NASA آغاز به جایگزینی ماشین های گرانقیمت خود با خوشه های کامپیوتری ارزان قیمت که لینوکس را اجرا می کردند، 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 سیستم عامل های رومیزی فراهم کردند. سیستم های لینوکس امروزه کاملاً در محاسبات استفاده می شوند، از سیستم های تعبیه شده تا تقریباً هر سوپر کامپیوتر، و موقعیتی در نصب های سروری مانند پشته برنامه های معروف LAMP ایجاد کرده اند. استفاده از توزیع های لینوکس در رومیزی های شرکتی و خانگی در حال توسعه است. همچنین، توزیع های لینوکس در بازار نت بوک معروف شده اند، با چندین دستگاه که با توزیع های لینوکس سفارشی نصب شده اند و گوگل که ChromeOS خود را برای نت بوک ها منتشر کرده است.

تکامل کامپیوتر

در روزهای اولیه، کامپیوترها به اندازه خانه ها یا پارک ها بودند. بنابراین می توانید 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 می شد. هر نرم افزار برای یک هدف خاص طراحی شده بود و قادر به اجرا در کامپیوتر دیگر نبود. این بسیار گران قیمت بود و افراد عادی نه تنها نمی توانستند هزینه آن را بپردازند بلکه نمی توانستند آن را هم درک کنند.

تکامل یونیکس

در سال ۱۹۶۹، یک تیم توسعه دهندگان از آزمایشگاه های بل اقدام به شروع یک پروژه جهت ساخت یک نرم افزار مشترک برای تمامی کامپیوترها کردند و آن را 'یونیکس' نامیدند. این سیستم ساده و زیبا بود، از زبان C به جای زبان اسمبلی استفاده می کرد و کد آن قابل بازیافت بود. به عنوان یک کد قابل بازیافت، بخشی از کد آن که اکنون به طور رایج 'هسته' نامیده می شود برای توسعه سیستم عامل و سایر عملکردها استفاده شد و قابل استفاده در سیستم های مختلف بود. همچنین کد منبع آن منبع باز بود.

اولاً، یونیکس تنها در سازمان های بزرگی مانند دولت، دانشگاه یا شرکت های مالی بزرگ با مینی کامپیوترها و میان کامپیوترها (کامپیوترهای شخصی کامپیوترهای کوچک میکرو) یافت می شد.

گسترش یونیکس

در دهه هشتاد، بسیاری از سازمان ها مانند آی بی ام، اچ پی و دوازده شرکت دیگر شروع به ایجاد یونیکس خود کردند. این منجر به یک آشفتگی از گویش های 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 توزیع‌ها

توزیع‌های^۱ لینوکس: تنها چیزی که در مورد هر توزیع منحصر به فرد است، نرم‌افزار مدیریت بسته‌بندی^۲ یا نصب برنامه‌ها است. مهم است که نسخه‌های کامپایلر، کتابخانه‌ها و هسته با هم بدون اشکال کار کنند. این برای بسته‌های نرم‌افزاری همراه نیز مهم است. پس از انتخاب نسخه‌های کامپایلر، کتابخانه و هسته، سایر بسته‌های نرم‌افزاری انتخاب، کامپایل و با هم جمع می‌شوند.

ویژگی‌های توزیع

تفاوت‌ها هنگام مقایسه توزیع‌های 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. توزیع‌ها اغلب یک طرح فهرست منحصر به فرد و پیکربندی اسکریپت را انتخاب می‌کنند. 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 می‌کند. 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 نرم‌افزاری 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 نرم‌افزارهای ارائه شده آزاد و نرم‌افزار منبع باز هستند که هم به صورت باینری کامپایل شده و هم به صورت کد منبع^۴ در دسترس هستند و امکان تغییرات در نرم‌افزار اصلی را فراهم می‌کنند. 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

/mnt: mount directory

محل نگهداری نقطه‌های متصل شده^۱ به طور موقت به سیستم مثل فلش دیسک یا یک هارد خارجی^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 معمولاً در پس زمینه^۴ در حال اجرا هستند.

سایر سیستم فایل ها**/lost+found**

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

/sys

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

/boot/vmlinuz

محل برای نگهداری فایل‌های کرنل 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 اسکریپت‌های **start/stop** سرویس ها

/etc/hosts

محل نگهداری آدرس‌های های **ip**هاست‌های مجازی و دیگر آدرس‌های **ip**

/etc/passwd

محل نگهداری پسوندهای کاربران سیستم

/etc/printcap

محل ذخیره اطلاعات مربوط به پرینترهای موجود بر روی سیستم عامل

/etc/x11

محل ذخیره تنظیمات پنجره‌های **X**. تنظیمات رابط‌های گرافیکی سیستم

/etc/resolv.conf

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

/usr/share

محل ذخیره فهرست‌های اشتراک گذاشته شده در میان سیستم ها

^۱ Mount

^۲ External

^۳ Removable Devices

^۴ Background

```

Activities  Tilix  Nov 11 00:43
Tilix: ubuntu@tecmint: ~
1: ubuntu@tecmint: ~
ubuntu@tecmint:~$ pwd
/home/ubuntu
ubuntu@tecmint:~$

2: ubuntu@tecmint: ~
ubuntu@tecmint:~$ ls -l
total 36
drwxr-xr-x 2 ubuntu ubuntu 4096 Feb  6  2021 Desktop
drwxr-xr-x 2 ubuntu ubuntu 4096 Feb  6  2021 Documents
drwxr-xr-x 4 ubuntu ubuntu 4096 Oct 26 04:35 Downloads
drwxr-xr-x 2 ubuntu ubuntu 4096 Feb  6  2021 Music
drwxrwxr-x 3 ubuntu ubuntu 4096 Oct 26 02:44 NetBeansProjects
drwxr-xr-x 2 ubuntu ubuntu 4096 Feb  6  2021 Pictures
drwxr-xr-x 2 ubuntu ubuntu 4096 Feb  6  2021 Public
drwxr-xr-x 2 ubuntu ubuntu 4096 Feb  6  2021 Templates
drwxr-xr-x 2 ubuntu ubuntu 4096 Feb  6  2021 Videos
ubuntu@tecmint:~$

```

ترمینال Tilix برای لینوکس

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

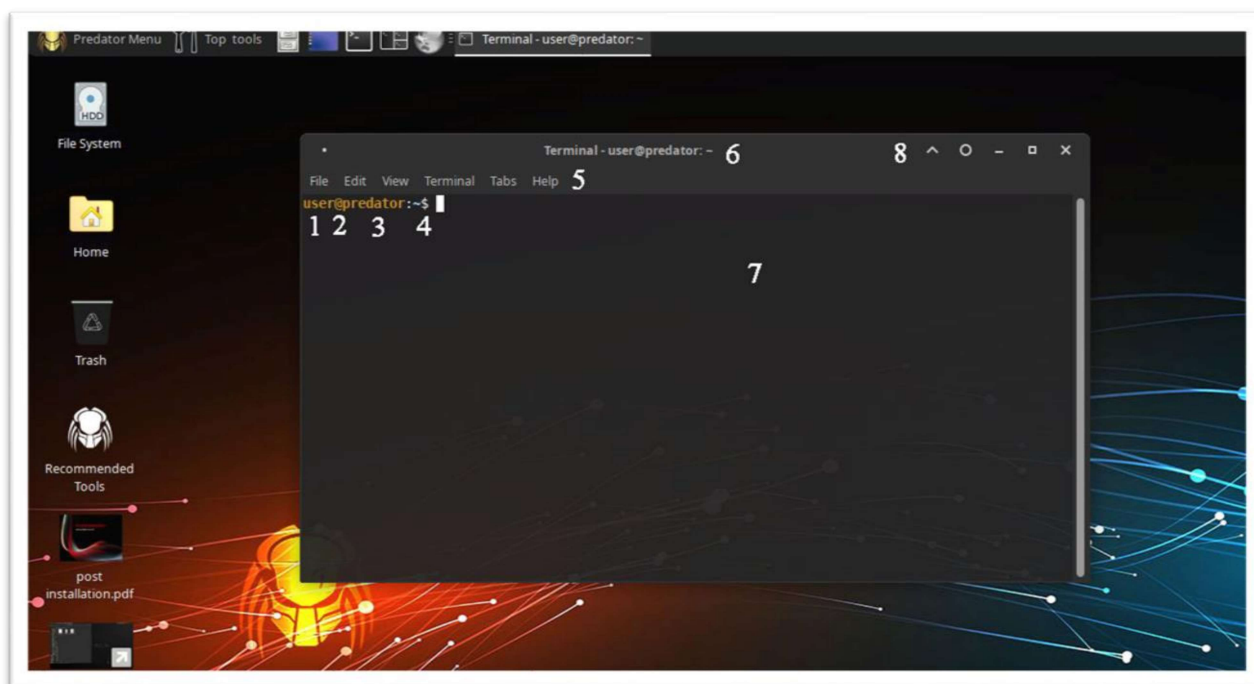
\$ sudo apt install tilix [در دیبیا، اوبونتو و مینت]
\$ sudo yum install tilix [در RHEL/CentOS/Fedora و Rocky Linux/AlmaLinux]
\$ sudo emerge -a x11-terms/tilix [در لینوکس جنٹو]
\$ sudo pacman -S tilix [در آرچ لینوکس]
\$ sudo zypper install tilix [در OpenSUSE]

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 شبیه ساز ترمینال پیش فرض است که بر اساس VTE برای محیط رومیزی سبک LXDE طراحی شده است. این ترمینال بدون هیچ گونه وابستگی غیر ضروری، امکانات لازم را برای کاربران فراهم می کند. بین ویژگی های خوب LXTerminal می توان به موارد زیر اشاره کرد:

سبک و سریع: LXTerminal به عنوان یک شبیه ساز ترمینال سبک طراحی شده است و به کاربر امکان استفاده آسان و سریع را می دهد.

بدون وابستگی غیر ضروری: LXTerminal بدون هیچ گونه وابستگی غیر ضروری ارائه می شود که باعث افزایش کارایی و کاربردی بودن آن می شود.



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. (از طریق تنظیمات ترمینال)

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

۸. دکمه های کنترل پنجره ترمینال

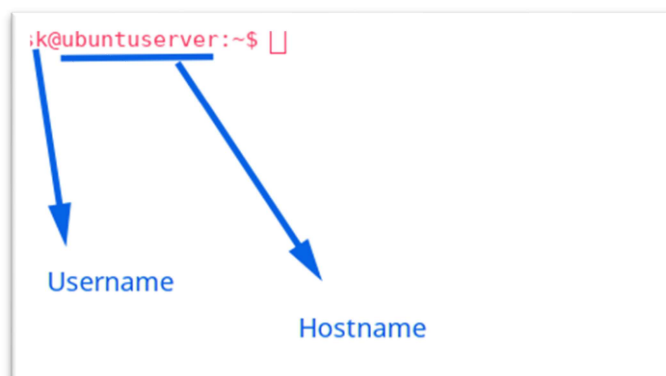
در ابتدا که ترمینال خود را راه اندازی می کنید، ممکنه با یک پنجره نسبتاً کسل کننده همراه باشید که کمی متن عجیب و غریب در بالا دارد، بسته به سیستم لینوکس شما ممکن است رنگها یکسان نباشد، و متن احتمالاً چیز دیگری را بیان می کند، اما طرح کلی یک پنجره با یک متن متن بزرگ (عمدتاً خالی) باید مشابه مواردی که ذکر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

^۱ useraccount

^۲ hostname

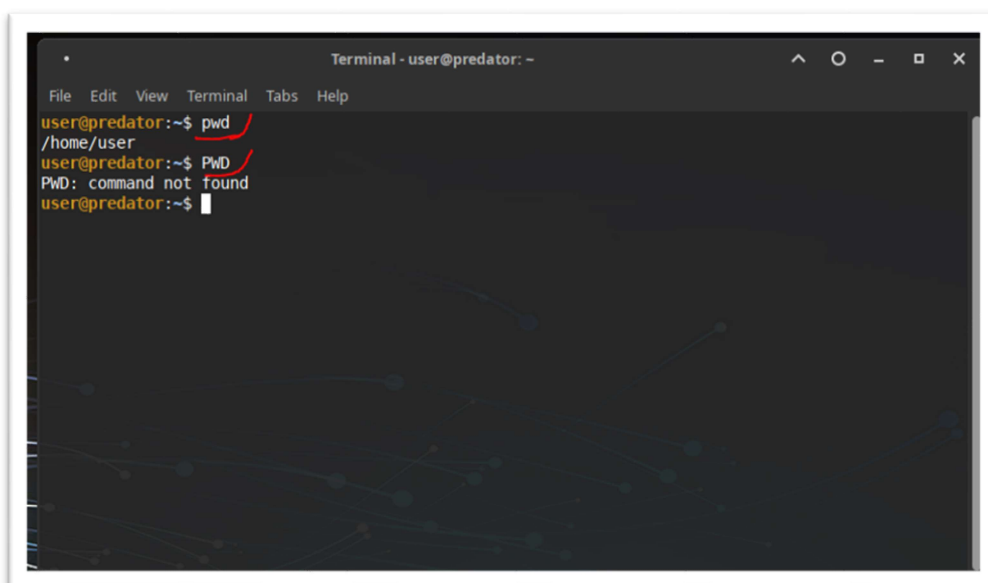


دومین چیزی که باید درک کنید این است که وقتی یک دستور را اجرا می کنید، هر خروجی که تولید می کند، مستقیماً در ترمینال چاپ و نمایش داده می شود. پس از اتمام آن یک پیام دیگر به شما نشان داده می شود. برخی از دستورات می توانند متن زیادی را چاپ کنند، برخی دیگر بی صدا کار می کنند و به هیچ وجه خروجی نخواهند داشت. 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 می باشد.



حساسیت دستورات در ترمینال

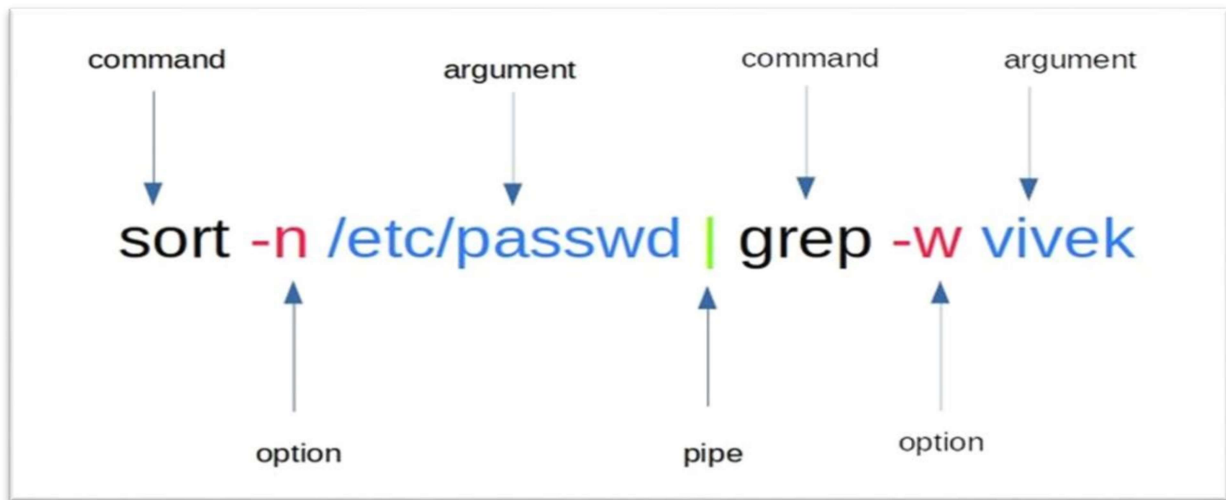
هنگام تایپ کردن در خط فرمان ترمینال، به موارد بزرگ یا کوچک تایپ کردن دستورات و مقادیر ورودی دقت کنید. تایپ کردن **PWD** به جای **pwd** خطایی ایجاد می کند، اما گاهی اوقات مورد اشتباه ممکن است منجر به اجرای دستور شود، اما آنچه را که انتظار داشتید انجام ندهد.



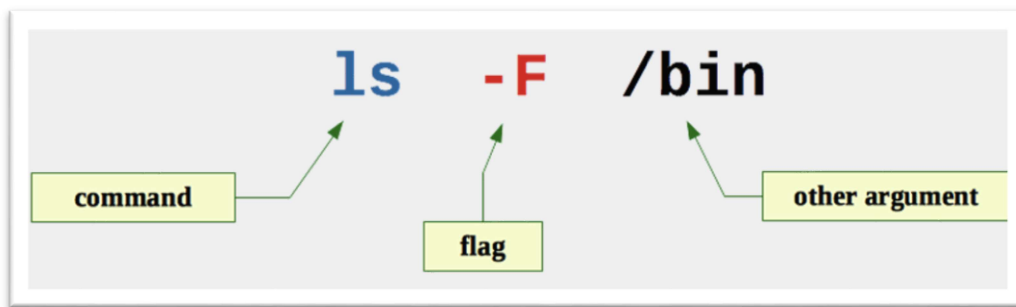
فصل دوم

آناتومی دستور

در تصویر زیر ساختار 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 (Command): این 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/Arguments): گاهی 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 (Parameters): برخی 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 Termination): هر 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. مثلاً در دستور `ls -a` وقتی `option` را `-a` قرار دهید، در خروجی فایل‌های مخفی نمایش داده می‌شود.

\$ls -a

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.

گزینه‌های 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.

مثال‌های 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 می‌توانیم نام مستعار را در پرونده‌های پیکربندی پسته ذخیره کنیم. به عنوان مثال برای `Bash Shell`، می‌توانیم 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 نشان می‌دهد.

```
# 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 فلش‌های 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
```


فصل پنجم

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 پیش فرض.

man --debug cat

چاپ جزئیات اشکال زدایی برای دستور cat.

دستور ln

دستور ln در سیستم عامل های Unix/Linux برای ایجاد لینک های سخت و نرم به فایل ها یا فهرست ها استفاده می شود. لینک ها به کاربر امکان می دهند تا به فایل یا فهرست های 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)

گزینه های دستور:

-  **b-** پیوند سخت (hard link) ایجاد می کند.
-  **s-** پیوند نمادین (symbolic link) ایجاد می کند.
-  **f-** اگر فایل مقصد وجود داشته باشد، آن را حذف و سپس پیوند جدید را ایجاد می کند.
-  **i-** قبل از ایجاد پیوند، از کاربر تأیید می گیرد.
-  **n-** پیوند را بدون ایجاد آن، فقط شبیه سازی می کند.
-  **v-** اطلاعات مربوط به پیوندهای ایجاد شده را نمایش می دهد.

مثال های دستور:

۱. ایجاد لینک سخت:

این دستور برای ایجاد یک لینک سخت به یک فایل یا فهرست استفاده می شود.

ln file1.txt link1

۲. ایجاد لینک نرم:

این دستور برای ایجاد یک لینک نرم به یک فایل یا فهرست استفاده می شود.

ln -s file2.txt link2

در این مثال، الگوی جستجوی `(*)` برای هر فایل با پسوند `txt` مطابقت می‌کند و با استفاده از الگوی جایگزینی `$1` `new.txt` نام فایل را تغییر می‌دهد. الگوی `(*)` هر چیزی را که قبل از `txt` قرار دارد، گرفته و آن را در الگوی جایگزینی `$1` `new.txt` 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

[نام_فایل] file

گزینه‌های 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.

مثال‌های 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
```

مثال‌های پیشتر:

تشخيص نوع فايل بدون نام:

file file.txt

خروجی:

file.txt: ASCII text

تشخيص 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
```

خروج

```
/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

```

. 2. 73 69 2. 73 69 68 54 :.....a 00 00 00 00 00 00 00

```

[illegible]

نمایش ۱۰ خط اول فایل:

```
file -n 10 file.txt
```

خود

This is a text file.

It has multiple lines.

نمایش اطلاعات مربوط به نوع فایل به صورت تفصیلی:

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

مدیر بسته apt

می توانید یک بسته را به صورت زیر تنظیم و نصب 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

```
tecmint@tecmint ~$ sudo apt install glances
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  putty-tools
Use 'apt-get autoremove' to remove it.
Recommended packages:
  python-jinja2
The following NEW packages will be installed:
  glances
0 upgraded, 1 newly installed, 0 to remove and 230 not upgraded.
Need to get 0 B/509 kB of archives.
After this operation, 1,040 kB of additional disk space will be used.
Selecting previously unselected package glances.
(Reading database ... 234340 files and directories currently installed.)
Preparing to unpack .../glances_1.7.3-2ubuntu1_all.deb ...
Unpacking glances (1.7.3-2ubuntu1) ...
Processing triggers for ureadahead (0.100.0-16) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Setting up glances (1.7.3-2ubuntu1) ...
* Starting Glances server glances
* Not starting glances: disabled by /etc/default/glances.
tecmint@tecmint ~$
```

دستور زیر کمک خواهد کرد که لیستی از تمام فایل هایی که در بسته ای به نام `glances` وجود دارد.

```
$ sudo apt install nethogs goaccess
```

نصب چند بسته 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 دانلود کد 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
```

مدیر بسته dnf

مقدمه ای بر دستور DNF

Rocky، CentOS، RHEL مانند RPM یک ابزار مدیریت بسته در توزیع های لینوکس مبتنی بر RPM استفاده می کنند. 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 به روزرسانی بسته ها در سیستم مبتنی بر RPM استفاده می شوند. 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

dnf repolist enabled

خروجے 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 غیر فعال فعلی با dnf

برای دریافت لیست تمامی مخازن غیر فعال شده می توانید از گزینه disabled استفاده کنید.

dnf repolist disabled

خروجے 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

خروجے دستور:

[root@rockylinux ~]# dnf repolist all

repo id	repo name
appstream	Rocky Linux 8 - AppStream
appstream-source	Rocky Linux 8 - AppStream - Source
baseos	Rocky Linux 8 - BaseOS
baseos-source	Rocky Linux 8 - BaseOS - Source
devel	Rocky Linux 8 - Devel WARNING! FOR BUILDROOT USE ONLY!
extras	Rocky Linux 8 - Extras
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
plus	Rocky Linux 8 - Plus
plus-source	Rocky Linux 8 - Plus - Source

status
enabled
disabled
enabled
disabled
disabled
disabled
enabled
disabled
disabled
disabled
disabled
disabled
disabled
enabled
disabled

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:

- 🔗 **--install, -i:** نصب بسته
- 🔗 **--remove, -r:** حذف بسته
- 🔗 **--build, -b:** ساخت بسته
- 🔗 **--list, -l:** لیست تمام بسته‌های نصب شده
- 🔗 **--status, -s:** نمایش وضعیت یک بسته
- 🔗 **--configure, -c:** پیکربندی مجدد بسته
- 🔗 **--purge, -P:** حذف کامل یک بسته به همراه فایل‌های پیکربندی
- 🔗 **--listfiles, -L:** نمایش لیست فایل‌های یک بسته
- 🔗 **--search, -S:** جستجوی بسته بر اساس نام یا توضیحات

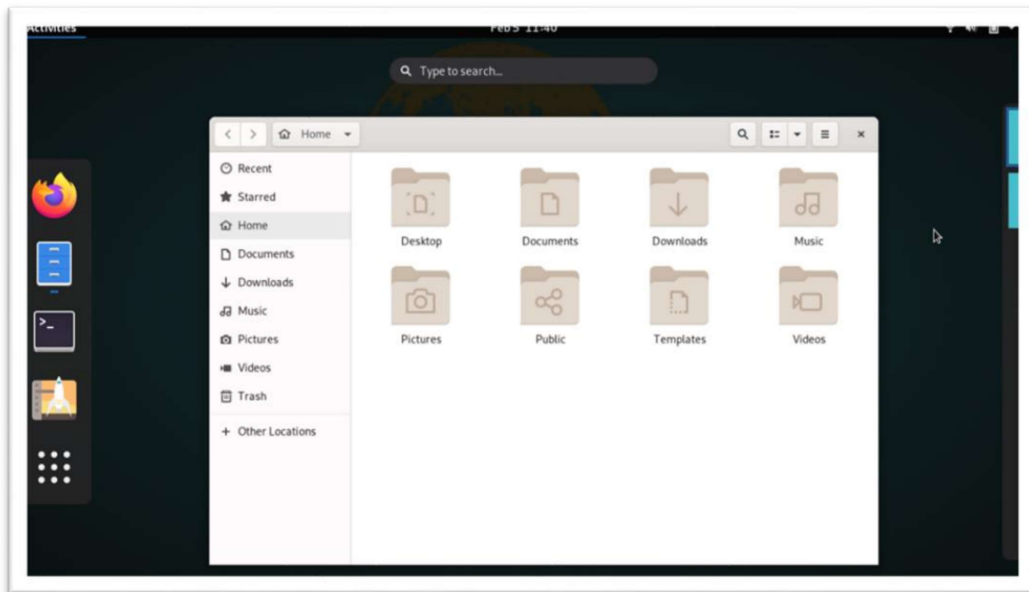
توضیحات گزینه‌های اضافی دستور dpkg

نصب و باز کردن بسته

- 🔗 **--install package-file...: -i** 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.
- 🔗 **--pending** و **-a** گزینه‌های `--pending` برای پیکربندی تمام بسته‌های منتظر استفاده می‌شوند.



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

فصل هفتم

شخصی سازی میزهای لینوکس

سفارشی سازی میز گنوم

نرم افزار GNOME Tweaks یک ابزار ساده و قدرتمند برای تنظیمات و سفارشی سازی بر روی محیط گرافیکی GNOME است. با استفاده از این ابزار، کاربران می توانند انواع تنظیمات مربوط به ظاهر، رفتار و عملکرد GNOME Shell را تغییر دهند و به سلیقه خود تنظیم کنند.

نصب GNOME Tweaks برای سفارشی سازی میز گنوم:

```
sudo apt install gnome-tweaks -y
```

نصب افزونه های GNOME Shell به کاربران امکان می دهد تا قابلیت ها و ویژگی های جدیدی را به محیط گرافیکی GNOME Shell اضافه کنند. این افزونه ها می توانند به شکل دهی به منوها، پوسته ها، نوارها و دیگر عناصر محیط گرافیکی کمک کرده و تجربه کار با GNOME Shell را بهبود بخشند.

نصب افزونه های GNOME Shell:

```
sudo apt install gnome-shell-extensions -y
```

دستور gsettings

تنظیم تصویر پس زمینه میز گنوم

```
gsettings set org.gnome.desktop.background picture-uri 'wallpaper1.png'
```

تنظیم نمایش آیکن های میز کار 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
```

فصل هشتم

فصل هشتم

پردازش متون

دستور nl

ابزار nl یک 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.

گزینه‌های 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.

مثال‌های 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:

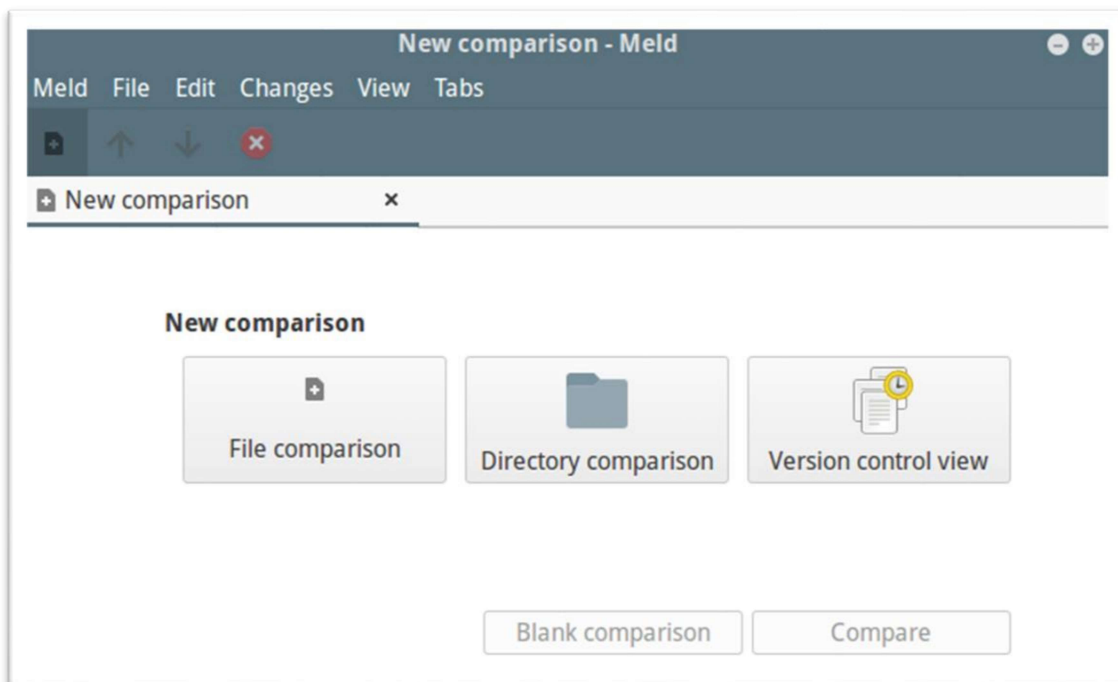
```
nl -l 2 file.txt
```

ابزار Meld

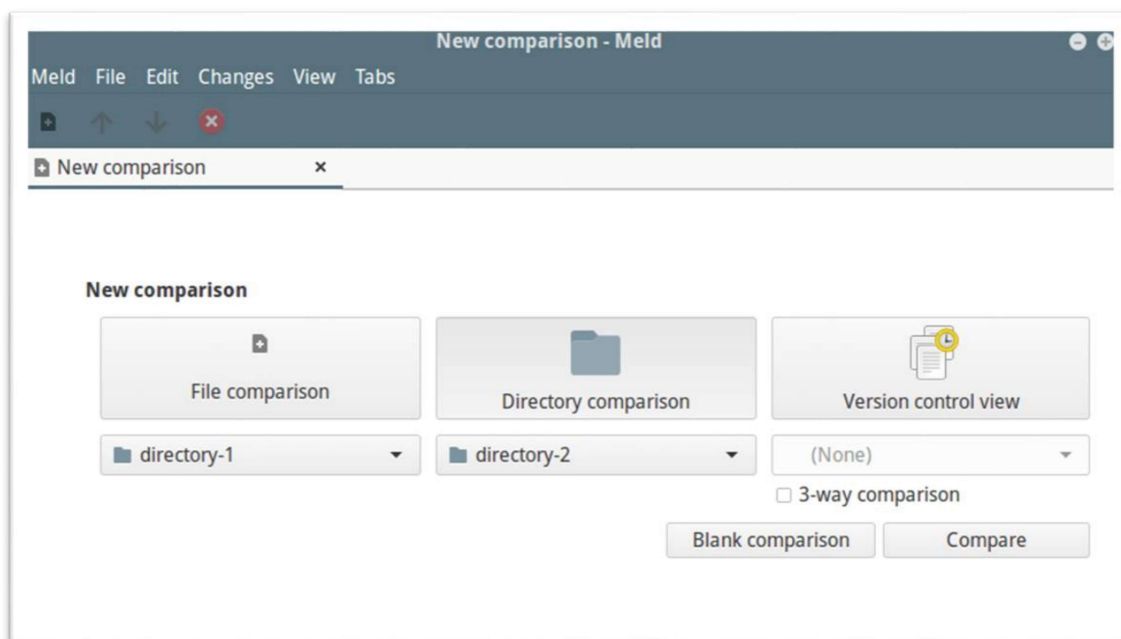
یک ابزار گرافیکی meld یک ابزار گرافیکی برای مشاهده تفاوت ها است که می توانید آن را به صورت زیر نصب کنید. داخلش کروشه^۱ توضیح است و جز دستور نیست. روش نصب:

\$ **sudo apt install meld** [system / دیبای / ubuntu]
 \$ **sudo yum install meld** [system های RHEL / CentOS]
 \$ **sudo dnf install meld** [+فدورا ۲۲]

در ادامه رابط گرافیکی برنامه Meld را میتوانید ببینید، جایی که می توانید فایل یا فهرست را مقایسه کنید.



ابزار مقایسه Meld



^۱ Brackets

فصل نهم

مدیریت سیستم فایل، دیسک و پارتیشن ها

دستور smartctl

دستور smartctl در سیستم عامل های Unix/Linux برای دسترسی و نمایش اطلاعات SMART (Self-Monitoring, Analysis, and Reporting Technology) دیسک های سخت (SSD و hard drive) استفاده می شود. این دستور به شما امکان می دهد تا وضعیت و عملکرد دیسک های خود را بررسی کنید و از پیش اخطارهای ممکن درباره خرابی یا نقص های احتمالی مطلع شوید.

مثال های دستور:

نمایش اطلاعات SMART دیسک: این دستور برای نمایش جزئیات و اطلاعات SMART دیسک مورد استفاده قرار می گیرد.

```
smartctl -a /dev/sda
```

تست کامل دیسک: این دستور برای انجام یک تست کامل بر روی دیسک و بررسی وضعیت آن استفاده می شود.

```
smartctl -t long /dev/sdb
```

نمایش وضعیت خطاهای دیسک: این دستور برای نمایش وضعیت خطاهای دیسک و تعداد بلوک های پیش اخطاری استفاده می شود.

```
smartctl -l error /dev/sdc
```

تنظیم و ذخیره سازی تنظیمات SMART: این دستور برای تنظیم و ذخیره سازی تنظیمات SMART بر روی دیسک استفاده می شود.

```
smartctl -s on -S /dev/sdd
```

تست سریع دیسک: این دستور برای انجام یک تست سریع بر روی دیسک و بررسی وضعیت آن استفاده می شود.

```
smartctl -t short /dev/sde
```

دستور badblocks

دستور badblocks در سیستم عامل های Unix/Linux برای تشخیص و شناسایی بلاک های خراب یا بد در یک دیسک سخت (هارد دیسک) استفاده می شود. این دستور به کاربر این امکان را می دهد تا بلاک های خراب یا بد را بررسی کرده و اطلاعات مربوط به آنها را دریافت کند.

مثال های دستور:

بررسی بلاک های خراب در یک دیسک سخت: این دستور برای بررسی بلاک های خراب در یک دیسک سخت با استفاده از حالت خواندن نوشتن تصادفی (read-write mode) استفاده می شود.

```
badblocks -w /dev/sda
```

بررسی بلاک های خراب با استفاده از حالت خواندن نوشتن تصادفی: این دستور برای بررسی بلاک های خراب در یک دیسک سخت با استفاده از حالت خواندن نوشتن تصادفی و نمایش اطلاعات جامع در خروجی استفاده می شود.

```
badblocks -vw /dev/sdb
```

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 pydf را 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 می تواند به روش های مختلفی انجام شود. در زیر دو روش نصب pydf را برای شما توضیح می دهیم:

۱. استفاده از مدیر بسته ها

بیشتر توزیع های لینوکس، از جمله Ubuntu، Debian و Fedora، دارای مدیر بسته هستند که امکان نصب و مدیریت بسته های نرم افزاری را فراهم می کنند. در این حالت، می توانید از مدیر بسته مورد نظر خود برای نصب pydf استفاده کنید. به طور مثال، اگر از Ubuntu استفاده می کنید، می توانید 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 بسته های پایتون از منابع خارجی را می دهد. اگر pip را نصب کرده اید، می توانید از دستور زیر برای نصب pydf استفاده کنید:

```
pip install pydf
```

با اجرای این دستور، pydf و تمامی وابستگی های آن نصب خواهند شد. بعد از نصب، می توانید pydf را به صورت مستقیم اجرا کنید تا از آن استفاده کنید. به طور معمول، دستور زیر را برای اجرای pydf در ترمینال وارد می کنید: با اجرای این دستور، اطلاعات دیسک ها و پارتیشن های شما به صورت یک جدول فرمت بندی شده نمایش داده می شود.

```
# 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
```

دستور: xfs_repair تعمیر فایل سیستم XFS

شرح دستور:

دستور `xfs_repair` یک 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_repair` می‌تواند 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_repair` می‌تواند 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_repair` می‌تواند تلاش کند تا آن را تعمیر کند.
 🐞 **بازیابی داده‌ها:** در برخی موارد، `xfs_repair` می‌تواند 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/sdb2` با خروجی مفصل:

```
sudo xfs_repair -v /dev/sdb2
```


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 می دهد که می توان انجام داد.

```
$ sudo parted
```

نرم 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
```

فصل دهم

دریافت اطلاعات سخت افزاری در linux

دستور Dmidecode

چگونه می‌توان از دستور **Dmidecode** برای بازیابی اطلاعات سخت افزاری هر **linux system** استفاده کرد؟ فرض کنید ما می‌خواهیم اطلاعات یک **system** را که نیاز داریم جمع‌آوری کنیم. مانند **memory**، بایوس^۱ و پردازنده و غیره. با استفاده از دستور **Dmidecode**، ما می‌خواهیم جزئیات را بدون باز کردن **system** تشخیص دهیم. دستور **Dmidecode** برای **RHEL / CentOS / Fedora / Ubuntu Linux** کار می‌کند.



به ابزار **DmidecodeDMI** برخی می‌گویند (**SMBIOS**) که جدولی، برای جمع‌آوری داده‌ها و نمایش اطلاعات مفید **system** مانند اطلاعات سخت افزاری، شماره سریال و نسخه **BIOS**، پردازنده و غیره در فرمت قابل خواندن برای انسان است. می‌توانید از دستور **root** برای اجرای دستور **dmidecode** استفاده کنید. در زیر نمونه خروجی دستور **Dmidecode** است.

```
# dmidecode
```

خروجی دستور:

اصلاح نسخه (2.31 -> 2.3) **SMBIOS**
SMBIOS 2.3 موجود است
 45 سازه اشغال ۱۶۴۲ بایت.
 جدول در 000E0010x
 دسته 0000x، نوع 0 DMI، ۲۰ بایت
 اطلاعات **BIOS**
 فروشنده: فونیکس فن آوری های محدود
 نسخه: ۶.۰۰
 تاریخ انتشار: ۲۰۰۶/۰۶/۱۲
 آدرس: xE78A0۰

^۱ bios

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 می کند.

ساختار دستور

```
$ 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 می کند. خروجی فرمان را می توان برای تجزیه و تحلیل و یا برای خوانایی آسان تر برای کاربر تنظیم کرد. برای مثال، این اطلاعات شامل تعداد پردازنده ها، رشته ها، هسته ها، سوکت ها و گره های دسترسی به حافظه غیریکنواخت (NUMA) می شود. 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 lscpu با 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 lscpu در 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:

این دستور، اولویت I/O فرآیند با شناسه پردازش 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
```

- **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 جدول وار 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 بایت 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 بایت 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
```

نام فرآیند **\$ls -c**

[c] +: نمایش فایل‌های باز شده توسط همه فرآیندهای با نام شامل c

نام فرآیند **\$ls +c**

d [d] -| +: نمایش فایل‌های باز شده با میزان تأخیر بیشتر یا کمتر از d

میزان تأخیر **\$ls +d**

میزان تأخیر **\$ls -d**

D [D] -| +: نمایش فایل‌های باز شده با میزان توسعه بیشتر یا کمتر از D

میزان توسعه **\$ls +D**

میزان توسعه **\$ls -D**

e [s] -| +: نمایش فایل‌های باز شده با پسوند s

پسوند **\$ls +e**

پسوند **\$ls -e**

E -| +: نمایش فایل‌های باز شده توسط همه فرآیندها بدون پرونده اضافی

\$ls +E

\$ls -E

f [cfgGn] -| +: نمایش فایل‌های باز شده با حالت‌های مشخص (**c**: character devices, **f**: regular files, **g**: NFS files, **G**: non-NFS files)

\$ls +f [cfgGn]

\$ls -f [cfgGn]

F [f] -: خروجی دستور **ls -F** را به فرمت مورد نظر تغییر دهید

فرمت **\$ls -F**

-g [s]: نمایش فایل‌های باز شده توسط گروه با شماره s

[شماره گروه] **\$ls -g**

-i [i]: نمایش فایل‌های باز شده توسط پورت i

[شماره پورت] **\$ls -i**

-k [k]: نمایش فایل‌های باز شده توسط فایل‌های اسکرپتی با نام k

[نام فایل اسکرپت] **\$ls -k**

-K [k]: بستن تمامی فایل‌های باز شده توسط فایل‌های اسکرپتی با نام k

[نام فایل اسکرپت] **\$ls -K**

+|-L [l]: نمایش فایل‌های باز شده با سطح لاگ‌ها بیشتر یا کمتر از l

[سطح لاگ] **\$ls +L**

[سطح لاگ] **\$ls -L**

+|-m [m]: نمایش فایل‌های باز شده با نوع موجودیت بیشتر یا کمتر از m

[نوع موجودیت] **\$ls +m**

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 ~]#
```

```
[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]'.

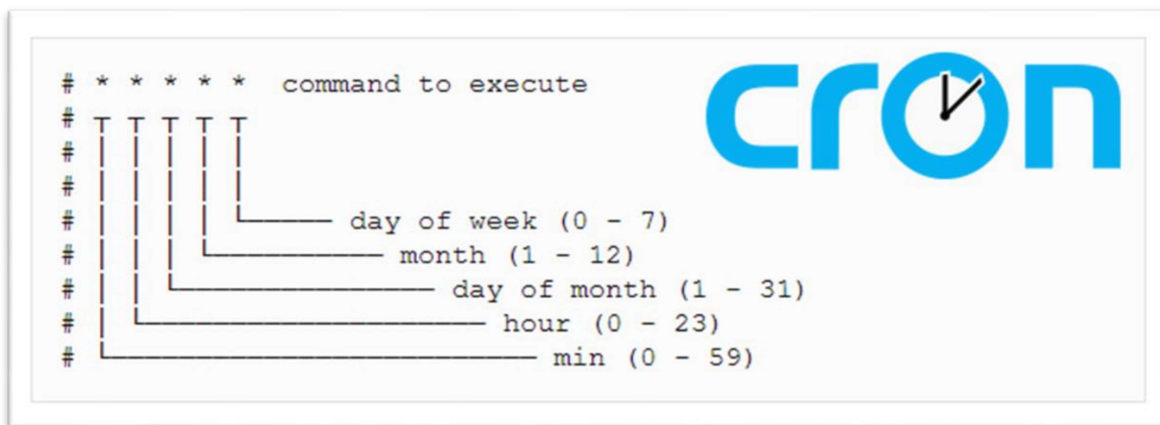
```
rhnsd         0:off  1:off  2:on   3:on   4:on   5:on   6:off
[root@localhost ~]#
```


- **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

cron 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:

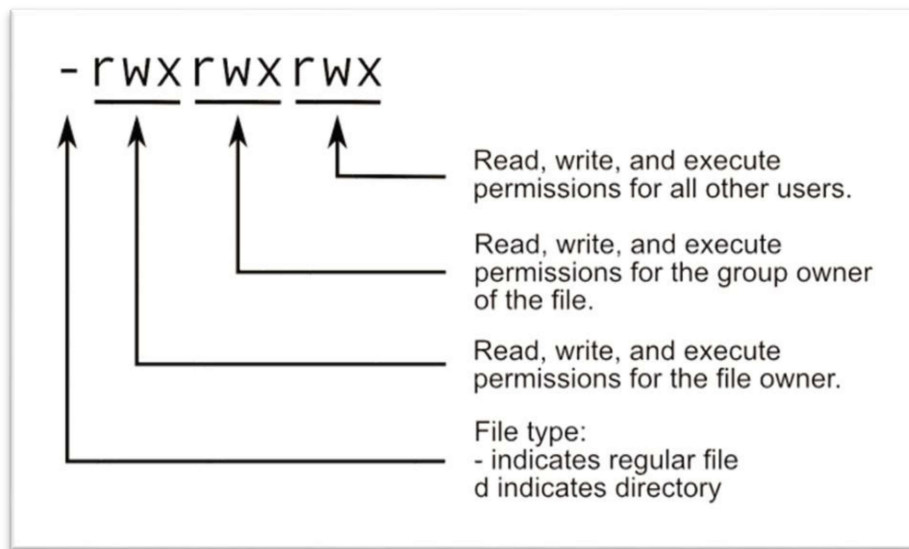
- f** : برای ماندن در حالت پیش زمینه استفاده می شود و تقلب نمی کنید.
- l** : این نام های سازگار با LSB را برای `/etc/cron.d` پرونده ها فعال می کند.
- n** : برای افزودن FQDN در موضوع هنگام ارسال نامه های الکترونیکی استفاده می شود.
- L loglevel** : این گزینه به cron می گوید که چه کارهایی را باید با مقادیر 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. **نحوه** 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.

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 (ssh پورته امن) 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

دستور **chmod** برای 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
```

مثال های دستور:

نمایش فقط دسترسی های ACL برای کاربران و گروه ها.

```
getfacl -a file.txt
```

خروجی دستور: نمایش فقط دسترسی های ACL برای کاربران و گروه ها 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 تب ها در خروجی.

```
getfacl -e 4 file.txt
```

خروجی دستور: نمایش لیست کنترل دسترسی برای فایل file.txt با اندازه تب ۴.

نادیده گرفتن ACL پیش فرض.

```
getfacl -E file.txt
```

خروجی دستور: نادیده گرفتن ACL پیش فرض 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 فایل ها و زیرفهرست ها به صورت بازگشتی.

```
getfacl -R directory
```

خروجی دستور: نمایش لیست کنترل دسترسی برای فهرست directory و تمام زیرفهرست ها به صورت بازگشتی.

نمایش ACL برای لینک های 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 لینک های 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 ACL های پیش فرض.

```
getfacl -p file.txt
```

مثال های دستور:

گزینه **-b** یا **brief** برای نمایش خلاصه نتیجه بدون توضیحات اضافی استفاده می شود. مثال:

```
$file -b myfile.txt
```

گزینه **-c** یا **checking-printout** برای نمایش چک کردن پرینت استفاده می شود. مثال:

```
$file -c myfile.txt
```

گزینه **-h** یا **no-dereference** برای عدم از بین بردن پیوندهای نمادین استفاده می شود. مثال:

```
$file -h myfile.txt
```

گزینه **-i** یا **mime** برای نمایش نوع MIME فایل استفاده می شود. مثال:

```
$file -i myfile.txt
```

گزینه **-k** یا **keep-going** برای ادامه تشخیص نوع فایل برای فایل های متعدد استفاده می شود. مثال:

```
$file -k myfile1.txt myfile2.txt
```

گزینه **-L** یا **dereference** برای دنبال کردن پیوندهای نمادین استفاده می شود. مثال:

```
$file -L myfile.txt
```

گزینه **-N** یا **no-pad** برای عدم پر کردن فاصله در خروجی استفاده می شود. مثال:

```
$file -N myfile.txt
```

گزینه **-n** یا **no-buffer** برای عدم استفاده از بافر در خواندن فایل استفاده می شود. مثال:

```
$file -n myfile.txt
```

گزینه **-p** یا **preserve-date** برای حفظ تاریخچه فایل استفاده می شود. مثال:

```
$file -p myfile.txt
```

گزینه **-r** یا **raw** برای نمایش خروجی به صورت خام بدون تغییرات استفاده می شود. مثال:

```
$file -r myfile.txt
```

گزینه **-s** یا **special-files** برای تشخیص نوع فایل های ویژه استفاده می شود. مثال:

```
$file -s myfile.txt
```

گزینه **-v** یا **version** برای نمایش نسخه file استفاده می شود. مثال:

```
$file --version
```

گزینه **-z** یا **uncompress** برای از حالت فشرده سازی خارج کردن فایل استفاده می شود. مثال:

```
$file -z myfile.txt.gz
```

گزینه **-0** برای جدا کردن خروجی با صفر استفاده می شود. مثال:

```
$file -0 myfile.txt
```

گزینه **apple** برای تشخیص نوع فایل برای سیستم عامل اپل استفاده می شود. مثال:

```
$file --apple myfile.txt
```

گزینه **mime-encoding** برای نمایش رمزگذاری MIME فایل استفاده می شود. مثال:

```
$file --mime-encoding myfile.txt
```

گزینه **mime-type** برای نمایش نوع MIME فایل استفاده می شود. مثال:

```
$file --mime-type myfile.txt
```

گزینه **e testname** برای انجام تست مشخص بر روی فایل استفاده می شود. مثال:

```
$file -e elf myfile
```

گزینه **F separator** برای استفاده از جداکننده مشخص بین خروجی ها استفاده می شود. مثال:

```
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 1
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 Brasero یک 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.

```
# 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^۱ 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^۲، 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.txt زیر 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 50 روز به آنها دسترسی پیدا شده است.

```
# find / -atime 50
```

جستجو تمام فایل هایی که بیش از 50 روز قبل و کمتر از 100 روز تغییر کرده اند.

```
# find / -mtime +50 -mtime -100
```

جستجو همه فایل هایی که در 1 ساعت گذشته تغییر کرده اند.

```
# find / -cmin -60
```

جستجو همه فایل هایی که در 1 ساعت گذشته اصلاح شده اند.

```
# find / -mmin -60
```

^۱ empty file

^۲ hidden file

در مثال زیر، می توانید در ابتدا نام دامنه "هیچکدام" برگشت داده شود اما پس از تنظیم دستور، نام جدید را بر می گردانند.

```
[root@os ~]# domainname
(none)
[root@os ~]# domainname -b allnone
[root@os ~]# domainname
allnone
[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 -yp یا -nis: 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 DNS را 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 NIS و سپس سرور DNS نگاه کنید.
Red Hat/Fedora/CentOS: /etc/sysconfig/network	پیکربندی شبکه را مشخص میکند. به عنوان مثال IP، استاتیک، DHCP، NIS و 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، DHCP و غیره

دستور ifconfig

ifconfig: دستور شبکه فراموش شده که توسط دستور **ip** جایگزین شده است. دستور **ifconfig** برای پیکربندی و نمایش اطلاعات مربوط به رابط های شبکه در سیستم عامل لینوکس استفاده می شود. این ابزار قدرتمندی است که می تواند برای تنظیم آدرس IP، ماسک شبکه، دروازه پیش فرض، و سایر پارامترهای شبکه استفاده شود.

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 سیستم عامل های Unix/Linux برای پیکربندی و نمایش اطلاعات مربوط به تنظیمات و ویژگی های کارت شبکه Ethernet استفاده می شود. این دستور به کاربر این امکان را می دهد تا وضعیت و تنظیمات کارت شبکه خود را بررسی و تنظیم کند. همچنین دستور ethtool ابزاری قدرتمند برای مدیریت و پیکربندی رابط های شبکه در سیستم عامل لینوکس است. این ابزار اطلاعات مختلفی راجع به کارت شبکه، مانند سرعت، نوع اتصال، و آمار مربوط به ترافیک را نمایش می دهد. همچنین می توان از ethtool برای تنظیم پارامترهای مختلف کارت شبکه، مانند سرعت، duplex، و offload، 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 گروه های بندی کانال (فقط برای رابط های ۸۰۲,۳ad).
- x نمایش اطلاعات مربوط به تنظیمات فلوی کنترل (فقط برای رابط های ۸۰۲,۳x).
- k نمایش اطلاعات مربوط به تنظیمات offload (فقط برای رابط های ۸۰۲,۱Q).
- r تنظیمات رابط شبکه را به مقادیر پیش فرض باز می گرداند.
- S تنظیمات رابط شبکه را از یک فایل مشخصه بارگذاری می کند.
- c تنظیمات رابط شبکه را به یک فایل مشخصه ذخیره می کند.
- p تنظیمات PHY رابط شبکه را نمایش می دهد.
- n تنظیمات PHY رابط شبکه را تنظیم می کند.
- t تست های مختلفی را برای رابط شبکه اجرا می کند.
- D اطلاعات مربوط به درایور رابط شبکه را نمایش می دهد.
- M اطلاعات مربوط به firmware رابط شبکه را نمایش می دهد.
- u اطلاعات مربوط به EEPROM رابط شبکه را نمایش می دهد.

15. ردیابی مسیر به آدرس 192.168.1.100 IP با فاصله زمانی ۳ ثانیه بین پروب ها:

```
tracpath -i 3 192.168.1.100
```

16. ردیابی مسیر به آدرس 192.168.1.100 IP با حداکثر ۵ ثانیه انتظار برای پاسخ:

```
tracpath -T 5 192.168.1.100
```

17. ردیابی مسیر به آدرس 192.168.1.100 IP با حداکثر ۵ ثانیه انتظار برای پاسخ:

```
tracpath -T 5 192.168.1.100
```

18. ردیابی مسیر به آدرس 192.168.1.100 IP با حداکثر ۲ تلاش برای هر هاپ:

```
tracpath -w 2 192.168.1.100
```

19. ردیابی مسیر به آدرس 192.168.1.100 IP با استفاده از پروتکل UDP:

```
tracpath -U 192.168.1.100
```

20. ردیابی مسیر به آدرس 192.168.1.100 IP با استفاده از پروتکل TCP:

```
tracpath -S 192.168.1.100
```

21. ردیابی مسیر به آدرس 192.168.1.100 IP با استفاده از پروتکل ICMP:

```
tracpath -I 192.168.1.100
```

22. ردیابی مسیر به آدرس 192.168.1.100 IP با استفاده از پروتکل IPv4:

```
tracpath -4 192.168.1.100
```

23. ردیابی مسیر به آدرس 192.168.1.100 IP با استفاده از پروتکل IPv6:

```
tracpath -6 192.168.1.100
```

24. ردیابی مسیر به آدرس 192.168.1.100 IP و نمایش اطلاعات TTL:

```
tracpath -x 192.168.1.100
```

25. ردیابی مسیر به آدرس 192.168.1.100 IP و عدم نمایش اطلاعات پیشرفت:

```
tracpath -q 192.168.1.100
```

26. ردیابی مسیر به آدرس 192.168.1.100 IP و نمایش اطلاعات مربوط به نسخه:

```
tracpath -V 192.168.1.100
```

27. ردیابی مسیر به آدرس 192.168.1.100 IP با استفاده از رابط شبکه eth1 و پروتکل UDP:

```
tracpath -o eth1 -U 192.168.1.100
```

28. ردیابی مسیر به آدرس 192.168.1.100 IP با شروع از هاپ ۲ و حداکثر ۸ هاپ:

```
tracpath -f 2 -M 8 192.168.1.100
```

29. ردیابی مسیر به آدرس 192.168.1.100 IP با فاصله زمانی ۱ ثانیه بین پروب ها و حداکثر ۴ ثانیه انتظار برای پاسخ:

```
tracpath -i 1 -T 4 192.168.1.100
```

30. ردیابی مسیر به آدرس 192.168.1.100 IP با حداکثر ۳ تلاش برای هر هاپ و عدم نمایش اطلاعات پیشرفت:

```
tracpath -w 3 -q 192.168.1.100
```

دستور mtr

دستور mtr در سیستم عامل های Unix/Linux برای انجام تست پینگ و traceroute به صورت ترکیبی استفاده می شود. این دستور به شما این امکان را می دهد که همزمان پینگ ping را بر روی یک مسیر خاص انجام دهید و همچنین مسیری که بسته ها برای رسیدن به مقصد طی می کنند را نیز نمایش دهید.

تزیینات 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 ((PID مربوط 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 

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 "mypassword" تنظیم 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 بی سیم در سیستم عامل های Unix/Linux استفاده می شود. با استفاده از این دستور، می توانید اطلاعات جزئی تری در مورد شبکه های بی سیم در دسترس را مشاهده کنید، اطلاعاتی مانند لیست شبکه ها، کیفیت سیگنال، فرکانس و غیره را به شما نمایش می دهد.

گزینه های دستور:**گزینه های عمومی**

 h نمایش راهنمای دستور

 v افزایش جزئیات خروجی

 i رابط وایرلس مورد نظر

 گزینه های مربوط به اسکن

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 عیب‌یابی

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-smi` برای 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:

`lsmod` دستور:

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 میگرد
COMP	diff	دو فایل را مقایسه میکند و تفاوت ها را نشان میدهد. همچنین cmp, comm, tdiff و tdiff را ببینید.
FC	diff	دو فایل را مقایسه میکند و تفاوت ها را نشان میدهد. همچنین cmp, comm, tdiff و tdiff 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 \$p\$g	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 <i>NAME=command</i>	<i>alias NAME=command</i>	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 <i>filename.txt</i>	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 <i>A:\</i>	tar -cvf /dev/fd0 files mkdir, mcopy doswrite -a file (AIX only)	فایل ها را در فلاپی ذخیره میکند. استفاده از فلاپی های DOS را ببینید
RESTORE <i>A files</i>	tar -xvf /dev/fd0 files mkdir, 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] [<i>path file</i>] /s	chmod	تغییر مجوزهای فایل +: DOS تنظیم به r: remove فقط خواندن s: System 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)	<i>source script (cshrc)</i> <i>. script (bash)</i> <i>sh script</i>	اسکرپت 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: <i>directory path</i>	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 tar برای ایجاد کامپایل یک فهرست قبل از فشرده سازی استفاده کنید. لینوکس همچنین دارای فشرده سازی، gzip است
HOSTNAME	hostname	چاپ نام میزبان کامپیوتر
PING	ping	ارسال بسته ها به میزبان شبکه
TRACERT	tracert	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 های NetBIOS
ROUTE PRINT	route -n	چاپ جدول مسیریابی
NET HELP START	chkconfig --list grep on	لیست خدمات
NET START <i>service name</i> NET STOP <i>service name</i>	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 کامپیوتر MS/Windows ارسال پیام به کاربر یونیکس/لینوکس دیگر. آموزش یولینوکس را ببینید
WIN	startx	X-Windows را 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

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.



Emperor-OS



Predator-OS