

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



صدا و سیماي جمهوری اسلامی ایران

معاونت فضای مجازی

مرکز تلویزیون تعاملی

اداره گل راهبری سرویس های تعاملی اپراتورها

واحد ارزیابی و نظارت

## لینوکس مقدماتی

کاوه رضائی شیراز

## فهرست

۴	مقدمه
۵	گنو / لینوکس، لینوکس و نرم افزار متن باز
۸	پیکره تنظیمات سخت افزاری
۱۵	معرفی مفهوم کتابخانه های اشتراکی یا <b>shared libraries</b>
۱۸	سامانه مدیریت بسته ( <b>package manager</b> )
۲۲	شروع کار با خط فرمان
۳۱	انجام مدیریت ساده فایل ها
۳۴	هدایت ورودی و خروجی استاندارد
۳۶	ایجاد نظارت و کشتن فرآیند ها
۴۰	متغیرهای محیطی
۴۲	مدیریت حساب های کاربری
۴۵	یافتن فایل های سیستم
۴۸	سرویس های سیستم
۵۱	مقدمات شبکه

## مقدمه

علاقه مندان به کار با سیستم عامل گنو / لینوکس برای کسب مهارت در این حوزه می توانند در دوره های مختلفی که در عرصه مدارک بین المللی لینوکس وجود دارد شرکت کنند، یکی از مطرح ترین آنها مربوط به موسسه lpi است که با نام lpic شناخته می شود، دوره ای که بر خلاف اکثر رقبا بر مبنای توزیع خاصی از لینوکس استوار نبوده و براساس مسایل کاربردی در تمامی توزیع های مطرح این سیستم عامل طراحی گردیده است.

بنا بر نیازهای کاربردی دایره ارزیابی و نظارت اداره کل راهبری سرویس های تعاملی اپراتورها، دوره ای در مهر و آبان ماه ۱۳۹۷ تدارک دیده شد که برگزیده ای از کلیات مطرح در دوره های Linux Essentials و ۱ Lpic می باشد، این مکتوب برگرفته از مفاهیمی است که در دوره فوق مرور شده است

بازخورد شما نسبت به مطالب مطرح شده در این مکتوب موجب ویرایش و ارتقا آن خواهد شد، لذا خوانندگان این نوشته را از نظرات خود محروم نکنید.

آخرین نسخه این نوشته را میتوانید از تارنمای (<http://files.kavehrs.com/linux/>) تهیه نمایید

کاوه رضائی  
me@kavehrs.com

## گنو / لینوکس، لینوکس و نرم افزار متن باز

در این بخش میخواهیم به این موضوع بپردازیم که لینوکس چیست و مفهوم متن باز یا اوپن سورس چه معنایی دارد؟ مفهوم فلسفه آزادی در نرم افزار چگونه مفهومی است  
نرم افزاری که آزادی های زیر را برای کاربر قائل شود، نرم افزار آزاد خوانده می شود (توجه کنید که کلمه Free به معنای آزاد استفاده می شود و نه رایگان<sup>ii</sup>)

- آزادی اجرای برنامه برای هر کاری (آزادی صفرم)
  - آزادی مطالعه چگونگی کار برنامه و تغییر آن (پیش نیاز: متن برنامه) (آزادی یکم)
  - آزادی تکثیر و کپی برنامه (آزادی دوم)
  - آزادی تقویت و بهتر کردن برنامه و توزیع آن برای همگان (پیش نیاز: متن برنامه) (آزادی سوم)
- گاهی آزادی در متن برنامه به معنای دریافت دستمزد و قراردادن کل آن در اختیار خریدار است و گاهی به بدون دریافت دستمزد مانند دو توزیع گنو/لینوکس **centos** و **redhat** لینوکس چیست؟

لینوکس (انگلیسی: **Linux**) یک سیستم عامل شبه یونیکس است که بخش عمده آن سازگار با استاندارد پازیکس است. از دید فنی لینوکس تنها نامی است برای هسته سیستم عامل و نه کل آن. هسته ای که نخستین بار در سال ۱۹۹۱ میلادی توسط لینوس توروالدز منتشر شد. اما به طور معمول این نام به سیستم عامل هایی که از ترکیب بندی لینوکس (به عنوان هسته سیستم عامل) با نرم افزارهای آزاد و متن باز بدست می آیند اطلاق می شود. در حالی که بنیاد نرم افزار آزاد سیستم عامل حاصل از این ترکیب بندی را «گنو/لینوکس» می نامد. <sup>viv iii</sup>

### توزیع های گنو / لینوکسی کدامند؟

یک توزیع لینوکس یا توزیع گنو/لینوکس که به اختصار «توزیع» نیز گفته می شود، سیستم عاملی است که بر پایه هسته لینوکس یا دیگر هسته های بر پایه لینوکس مانند لینوکس-لیبره ساخته شده باشد. معمولاً توزیع های لینوکس دسکتاپ نه تنها شامل اجزای اصلی سیستم عامل بلکه مجموعه بزرگی از <sup>vi</sup> نرم افزارهای را نیز شامل می گردند  
سیستم عامل، شامل هسته لینوکس و یک سری کتابخانه ها و برنامه های سودمند دیگر از پروژه گنو، همراه با قابلیت پشتیبانی از گرافیک می باشد.

یک توزیع لینوکس ممکن است در یکی یا چند دسته زیر قرار بگیرد:

- تجاری یا غیر تجاری
- طراحی شده برای استفاده حرفه ای یا خانگی
- طراحی شده برای سرویس دهندها، میزکار (خانگی) یا سیستم های موبایل
- طراحی شده برای کاربران تازه کار یا حرفه ای
- استفاده عمومی یا استفاده به عنوان یک دستگاه خاص، به طور مثال دیوار آتش، مسیریاب شبکه
- طراحی شده برای یک سخت افزار خاص
- هدفمند برای گروهی خاص از کاربران به طور مثال برای زبانی خاص یا همراه با بسیاری از بسته های کاربری برای کارهای موسیقی یا محاسبات علمی
- پیکربندی شده برای امنیت، قابلیت استفاده بیشتر، قابل حمل بودن یا جامعیت داشتن
- پشتیبانی شده بر روی سخت افزارهای متفاوت

تنوع موجود در توزیع‌های لینوکس به نیازهای فنی، سازمانی و اخلاقی تولیدکنندگان و کاربران آن برمی‌گردد، در مجوز نرم‌افزار آزاد هر کاربر علاقه‌مندی که به مقدار کافی اطلاعات داشته باشد قادر است با انجام تغییرات دلخواه در توزیع‌های موجود توزیع مورد علاقه خود که همسو با نیازهایش است تولید کند.

## توزیع‌های معروف لینوکس vii

- آرچ لینوکس، یک توزیع که بر پایه قوانین KISS بنا شده همراه با سیستم انتشار rolling
- اسلکور اولین توزیع هنوز زنده لینوکس که شاخه بزرگی از توزیعها مانند زوزه از آن اشتقاق یافته‌اند یا مانند آرچ بعضی دیگر آنرا در مواردی مانند اصل KISS الگو قرار دادند. اسلکور با کمترین تغییر در بسته‌های همراه خودش، پایدارترین لینوکس به حساب می‌آید.
- سنت‌اواس، یک توزیع که از همان سورسی که ردهت استفاده می‌کند، مشتق شده است، توسط یک گروه داوطلبی که به آن اختصاص داده شدند، نگهداری می‌شود همراه با سازگاری ۱۰۰٪ با ورژن ردهت و یک ورژن به روز رسیده شده که همیشه به‌طور ۱۰۰٪ سازگار و همراه نیست.
- دیبان، یک توزیع غیر تجاری که توسط جامعه‌ای از توسعه دهنده‌های داوطلب نگهداری می‌شود که تعهد قوی‌ای به قواعد نرم‌افزارهای آزاد دارند.
- فدورا یک توزیع بر پایه جامعه است که توسط ردهت سرپرستی می‌شود.
- جنتو، یک توزیع که هدفش کاربران قویتر هست، شناخته شده‌است بخاطر سیستم خودکار شبه بی‌آس‌دی‌اش برای کامپایل برنامه‌ها از سورس.
- نایبکس، اولین توزیع سی‌دی زنده، که کاملاً از یک رسانه قابل حمل اجرا می‌شود بدون نصب در دیسک سخت. از دیبان مشتق شده‌است.
- کوبونتو، نسخه کی‌دی‌ای از اوبونتو
- لینوکس مینت، یک توزیع محبوب بر پایه و سازگار با اوبونتو
- مندریوا، یک توزیع مشتق شده از ردهت که در فرانسه و برزیل محبوب است، امروزه با همان نام و به‌وسیله یک شرکت فرانسوی اداره می‌شود.
- این سوزه، بهترین توزیع ارائه شده با میزکار KDE، حامی پروژه آزاد KDE و پنج‌مین توزیع محبوب لینوکس )

## آشنایی با نرم‌افزارهای متن باز

اگر از پلتفرم ویندوز یا مکینتاش به لینوکس مهاجرت میکنید، شاید متعجب شوید اگر برنامه‌هایی که قبلاً استفاده میکردید برای لینوکس هم موجود باشند. بعضی از نرم‌افزارهایی که هم‌اکنون استفاده میکنید نسخه‌های لینوکسی دارند. برای آن دسته که به این شکل نیستند برنامه‌های آزاد و متن‌بازی موجود هست که نیازهای شما را پوشش می‌دهند. در این بخش نرم‌افزارهایی را معرفی می‌کنیم که به خوبی روی لینوکس کار می‌کنند: viii

### Office Suites (مجموعه‌ی آفیس)

- Windows: Microsoft Office, LibreOffice
- Apple os x: iWork, Microsoft Office, LibreOffice
- Linux: LibreOffice, KOffice, gnome Office, Kexi (database application)

#### Email Applications (برنامه‌های پست الکترونیکی)

- Windows: Microsoft Outlook, Mozilla Thunderbird
- Apple os x: Mail.app, Microsoft Outlook, Mozilla Thunderbird
- Linux: Mozilla Thunderbird, Evolution, KMail

#### Web Browsers (مرورگرهای وب)

- Windows: Microsoft Internet Explorer, Mozilla Firefox, Opera, Chromium, Google Chrome
- Apple os x: Safari, Mozilla Firefox, Opera, Chromium, Google Chrome
- Linux: Mozilla Firefox, Opera, Chromium, Google Chrome, Epiphany

#### PDF Readers (خواننده‌های پی‌دی‌اف)

- Windows: Adobe Acrobat Reader, Foxit
- Apple os x: Adobe Acrobat Reader
- Linux: Evince, Adobe Acrobat Reader, Okular

#### Multimedia Players (پخش‌کننده‌های چندرسانه‌ای)

- Windows: Windows Media Player, vlc
- Apple os x: Quicktime, vlc
- Linux: Totem, vlc, MPlayer, Kaffeine

#### Music Players and Podcatchers (پخش‌کننده‌های موسیقی و پادکچرها)

- Windows: Windows Media Player, iTunes, Winamp
- Apple os x: iTunes
- Linux: Rhythmbox, Banshee, Amarok, Audacity, Miro

#### CD/DVD Burning (رایت سی‌دی/دی‌وی‌دی)

- Windows: Nero Burning rom, InfraRecorder
- Apple os x: Burn, Toast Titanium
- Linux: Brasero, K3b, Gnome-baker

#### Photo Management (مدیریت عکس)

- Windows: Microsoft Office Picture Manager, Picasa
- Apple os x: Aperture, Picasa
- Linux: Shotwell, gThumb, Gwenview, F-Spot

#### Graphics Editors (ویرایشگرهای گرافیکی)

- Windows: Adobe Photoshop, gimp
- Apple os x: Adobe Photoshop, gimp
- Linux: gimp, Inkscape

#### Instant Messaging (مسنجر)

- Windows: Windows Live Messenger, aim, Yahoo! Messenger, Google Talk
- Apple os x: Windows Live Messenger, aim, Yahoo! Messenger, Adium, iChat
- Linux: Empathy, Pidgin, Kopete

#### VoIP Applications (برنامه‌های وی‌پی)

- Windows: Skype, Google Video Chat
- Apple os x: Skype, Google Video Chat
- Linux: Ekiga, Skype, Google Video Chat

#### BitTorrent Clients (کلاینت‌های بیت‌تورنت)

- Windows: µTorrent, Vuze
- Apple os x: Transmission, Vuze
- Linux: Transmission, Deluge, KTorrent, Flush, Vuze, BitStorm Lite

## بیکره تنظیمات سخت افزاری

### proc directory

مکانی برای ذخیره اطلاعات کرنل و پردازنده ها است، در دایرکتوری **PROC** می توانید اطلاعات مرتبط با پردازنده ها را مشاهده کنید، باید در نظر گرفت این مفهوم بسیار قدیمی و به روزهای ابتدایی سیستم عامل لینوکس در سالهای ۱۹۹۰ باز میگردد که یکایک پروسس ها باید مدیریت می شد. این دایرکتوری روی حافظه **RAM** ساخته شده و ممکن است فایل ها قابل دسترسی باشد.

```
$ ls /proc/
۱ ۱۲۴۹ ۱۴۵۱ ۱۵۶۵ ۱۸۰۶۹ ۲۰۳۴۶ ۲۴۲۶ ۲۷۶۵ ۲۹۲۶ ۳۱۷۵ ۳۳۱۷ ۳۵۳۷ ۳۹ ۴۶۸ ۴۹۲۱
۵۳ ۶۸۹ ۹۶۹ filesystems misc sysvipc
۱۰ ۱۳ ۱۴۶ ۱۵۷ ۱۸۰۹۳ ۲۰۶۸۱ ۲۴۵۲ ۲۷۶۶ ۲۹۲۹ ۳۱۸۳ ۳۳۱۸ ۳۵۴ ۳۹۷ ۴۶۹۴ ۴۹۳۴
۵۳۸ ۷ ۹۷ fs modules timer_list
۱۰۳۹ ۱۳۲۱ ۱۴۷ ۱۵۷۲ ۱۸۲۴۳ ۲۱ ۲۴۵۶ ۲۸ ۲۹۳۴ ۳۱۸۷ ۳۴ ۳۵۴۱ ۴۰۴ ۴۶۹۵ ۴۹۵۵
۵۴ ۷۳۷ acpi interrupts mounts timer_stats
۱۰۸۹۹ ۱۳۳۴۶ ۱۴۸ ۱۵۷۶ ۱۸۲۷۴ ۲۱۰۲۱ ۲۴۶۲ ۲۸۴۱ ۲۹۳۶ ۳۱۹۱ ۳۴۵۰ ۳۵۵۰ ۴۱ ۴۷
۴۹۷۰ ۵۴۶ ۷۴ asound iomem mtrr tty
۱۰۹۶۰ ۱۳۴۳۸ ۱۴۸۱۷ ۱۵۸ ۱۸۵۹ ۲۱۱۳۹ ۲۵ ۲۸۵۱ ۲۹۴۵ ۳۲ ۳۴۵۹ ۳۵۷ ۴۲ ۴۷۲۰ ۴۹۸۲
۵۵ ۷۴۲ buddyinfo ioports net uptime
۱۱ ۱۳۶۱۹ ۱۴۹ ۱۶ ۱۸۶۱۷ ۲۱۲۹ ۲۵۹۲ ۲۸۵۲ ۲۹۴۷ ۳۲۰۲ ۳۴۶۶ ۳۶ ۴۳ ۴۷۳۱ ۴۹۹۵
۵۵۱ ۷۵ bus irq pagetypeinfo version
۱۱۱۲۰ ۱۳۶۶۱ ۱۵ ۱۶۱۳ ۱۸۷۸۱ ۲۱۴ ۲۶ ۲۸۶۲ ۲۹۴۸ ۳۲۰۶ ۳۴۶۷ ۳۶۸۳ ۴۴ ۴۷۵۶ ۵
۵۶ ۷۷ cgroups kallsyms partitions version_signature
۱۱۱۴۵ ۱۳۶۷۱ ۱۵۰ ۱۶۳۰ ۱۸۸۰ ۲۱۵ ۲۷ ۲۸۶۵ ۲۹۵۲ ۳۲۰۸ ۳۴۶۹ ۳۶۹۹ ۴۴۸۴ ۴۷۷۴ ۵۰
۵۷۷ ۸ cmdline kcore sched_debug vmallocinfo
۱۱۵۹ ۱۳۹۲۷ ۱۵۱ ۱۶۳۳ ۱۸۸۲ ۲۱۹۹ ۲۷۰۷ ۲۸۶۶ ۲۹۵۵ ۳۲۱۲ ۳۴۷۰ ۳۷ ۴۴۹۵ ۴۷۹۵
۵۰۰۸ ۵۸۰۶ ۸۹۲ consoles keys schedstat vmstat
۱۱۶۳ ۱۴ ۱۵۱۲ ۱۶۳۴ ۱۹ ۲۲ ۲۷۰۸ ۲۸۸۴ ۲۹۵۷ ۳۲۲۵ ۳۴۷۴ ۳۷۱۰ ۴۵ ۴۸ ۵۰۱۳ ۶۰
۹ cpuinfo key-users scsi zoneinfo
۱۱۶۴ ۱۴۰۴۵ ۱۵۱۵ ۱۶۹۳ ۱۹۰۶۱ ۲۲۱۹ ۲۷۰۹ ۲۸۸۷ ۲۹۶۱ ۳۲۳۶ ۳۴۷۵ ۳۷۵۲ ۴۵۰۶ ۴۸۱۱
۵۰۷۷ ۶۱ ۹۰۴ crypto kmsg self
۱۱۷۰ ۱۴۰۴۷ ۱۵۲ ۱۷ ۱۹۰۶۸ ۲۳ ۲۷۱۰ ۲۸۹۱ ۳ ۳۲۴ ۳۴۷۷ ۳۷۶۱ ۴۵۲۹ ۴۸۲۱ ۵۰۸۲
۶۲ ۹۰۶۱ devices kpagecount slabinfo
۱۱۷۴ ۱۴۰۵۲ ۱۵۳ ۱۷۱۷۳ ۱۹۰۶۹ ۲۳۰۵۵ ۲۷۱۱ ۲۸۹۵ ۳۰۴۷ ۳۲۶۱ ۳۵۱۷ ۳۷۷۸ ۴۵۵۸ ۴۸۴
۵۰۹۱ ۶۷۷ ۹۱۵ diskstats kpageflags softirqs
۱۲ ۱۴۰۹ ۱۵۴ ۱۷۳۲ ۱۹۰۷۵ ۲۳۵۴ ۲۷۱۸ ۲۹ ۳۰۹۳ ۳۲۸۴ ۳۵۲۲ ۳۸ ۴۵۶۲ ۴۸۶۱ ۵۱
۶۷۸ ۹۲۳ dma loadavg stat
۱۲۳۱ ۱۴۴۴ ۱۵۵ ۱۷۴۱۳ ۲ ۲۳۹۰ ۲۷۱۹ ۲۹۰۴ ۳۱ ۳۲۸۷ ۳۵۲۵ ۳۸۰۳ ۴۶ ۴۸۹۱ ۵۲
۶۷۹ ۹۳۹ driver locks swaps
۱۲۳۴ ۱۴۴۶ ۱۵۶ ۱۷۷۵۱ ۲۰ ۲۴ ۲۷۲۳ ۲۹۰۸ ۳۱۳۲ ۳۲۹۸ ۳۵۲۸ ۳۸۲۳ ۴۶۲۲ ۴۹ ۵۲۰۲
۶۸۰ ۹۴۰ execdomains mdstat sys
۱۲۳۶ ۱۴۵ ۱۵۶۳ ۱۸ ۲۰۲۸ ۲۴۱۸ ۲۷۶۳ ۲۹۱۱ ۳۱۷۱ ۳۳ ۳۵۳۳ ۳۸۴۵ ۴۶۶۱ ۴۹۰۷ ۵۲۵
۶۸۷ ۹۶ fb meminfo sysrq-trigger
```



## Sysfs

در توسعه کرنل لینوکس، فایل‌های مربوط به نگهداری اطلاعات سیستم و تجهیزات در یک دایرکتوری مجازی مستقل به عنوان **sysfs** نگهداری می‌شود، به عبارت دیگر این دایرکتوری اطلاعات **HAL** رو ذخیره می‌کند و **HAL** وظیفه کنترل و جداسازی لایه سخت افزار از لایه نرم افزار را برعهده دارد. به عنوان نمونه اگر بخواهید بدانید چه سخت افزارهایی به سیستم شما متصل هستند می‌توانید از این دایرکتوری با آدرس **sys/** اطلاعات فوق را بیابید. **sysfs** مخفف **sys file system** است.

```
$ ls /sys
```

```
block bus class dev devices firmware fs hypervisor kernel  
module power
```

dev

در این دایرکتوری دستگاه‌های انتزاعی مانند هارد دیسک است، مثل **dev/sda/** یا **dev/hd0/** بدون در نظر گرفتن نام تجاری، مدل یا تکنولوژی آن:

```
root@life:/dev# ls /dev/sda*
```

```
/dev/sda /dev/sda1 /dev/sda2 /dev/sda3 /dev/sda5 /dev/sda6
```

lsmod, lsusb, lspci

این دستورات لیستی از ماژول‌ها و سخت افزارها را در سیستم نشان می‌دهد.

**lsmod** ماژول‌های هسته را نشان می‌دهد:

```
root@life:/dev# lsmod
```

Module	Size	Used by
pci_stub	12622	1
vboxpci	23256	0
vboxnetadp	25670	0
vboxnetflt	27605	0
vboxdrv	418013	3 vboxnetadp,vboxnetflt,vboxpci
ctr	13049	3
ccm	17731	3
dm_crypt	23172	1
bnep	19543	2
rfcomm	69509	8
uvcvideo	81065	0
arc4	12608	2
videobuf2_vmalloc	13216	1 uvcvideo
intel_rapl	18783	0
iwldvm	236430	0
x86_pkg_temp_thermal	14205	0
intel_powerclamp	18786	0
btusb	32448	0
videobuf2_memops	13362	1 videobuf2_vmalloc
videobuf2_core	59104	1 uvcvideo
v4l2_common	15682	1 videobuf2_core
mac80211	660592	1 iwldvm
coretemp	13441	0
videodev	149725	3 uvcvideo,v4l2_common,videobuf2_core

```

media                21963  2  uvcvideo,videodev
bluetooth            446190 22  bnep,btusb,rftcomm
kvm_intel            143592  0
kvm                  459835  1  kvm_intel
snd_hda_codec_hdmi  47547  1
crct10dif_pclmul    14307  0
6lowpan_iphc        18702  1  bluetooth
crc32_pclmul        13133  0
snd_hda_codec_conexant 23064  1
ghash_clmulni_intel 13230  0
snd_hda_codec_generic 68914  1  snd_hda_codec_conexant
aesni_intel         152552 10
snd_seq_midi         13564  0
snd_seq_midi_event  14899  1  snd_seq_midi
aes_x86_64          17131  1  aesni_intel
mei_me              19742  0
lrw                 13287  1  aesni_intel
iwlwifi             183038 1  iwldvm

```

این ها ماژول های هسته ای هستند که بارگذاری می شوند.

اگر شما نیاز به اضافه کردن یک ماژول به هسته خود دارید (می گویند **driver** جدید برای یک سخت افزار) یا حذف آن (حذف یک **driver**) شما می توانید **rmmod** و **modprobe** استفاده کنید.

## Lspci

دستگاه های **PCI** که به کامپیوتر متصل هستند را نشان می دهد.

```

# lspci
00:00.0 Host bridge: Intel Corporation 2nd Generation Core Processor Family DRAM
Controller (rev 09)
00:02.0 VGA compatible controller: Intel Corporation 2nd Generation Core Processor
Family Integrated Graphics Controller (rev 09)
00:16.0 Communication controller: Intel Corporation 6 Series/C200 Series Chipset
Family MEI Controller #1 (rev 04)
00:19.0 Ethernet controller: Intel Corporation 82579LM Gigabit Network Connection
(rev 04)
00:1a.0 USB controller: Intel Corporation 6 Series/C200 Series Chipset Family USB
Enhanced Host Controller #2 (rev 04)
00:1b.0 Audio device: Intel Corporation 6 Series/C200 Series Chipset Family High
Definition Audio Controller (rev 04)
00:1c.0 PCI bridge: Intel Corporation 6 Series/C200 Series Chipset Family PCI Ex-
press Root Port 1 (rev b4)
00:1c.1 PCI bridge: Intel Corporation 6 Series/C200 Series Chipset Family PCI Ex-
press Root Port 2 (rev b4)
00:1c.4 PCI bridge: Intel Corporation 6 Series/C200 Series Chipset Family PCI Ex-
press Root Port 5 (rev b4)
00:1d.0 USB controller: Intel Corporation 6 Series/C200 Series Chipset Family USB
Enhanced Host Controller #1 (rev 04)
00:1f.0 ISA bridge: Intel Corporation QM67 Express Chipset Family LPC Controller
(rev 04)
00:1f.2 SATA controller: Intel Corporation 6 Series/C200 Series Chipset Family 6
port SATA AHCI Controller (rev 04)
00:1f.3 SMBus: Intel Corporation 6 Series/C200 Series Chipset Family SMBus Con-
troller (rev 04)
03:00.0 Network controller: Intel Corporation Centrino Wireless-N 1000 [Condor
Peak]

```

## Lsusb

همه دستگاه های USB متصل به سیستم را نشان می دهد.

```
# lsusb
```

```
Bus 002 Device 003: ID 1c4f:0026 Sigma Micro Keyboard
Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 005: ID 04f2:b217 Chicony Electronics Co., Ltd Lenovo Integrated
Camera (0.3MP)
Bus 001 Device 004: ID 0a5c:217f Broadcom Corp. BCM2045B (BDC-2.1)
Bus 001 Device 003: ID 192f:0916 Avago Technologies, Pte.
Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

## Lspcmcia

کارت های PCMCIA موجود در این رایانه را نشان می دهد.

## Lshal

داده های HAL را نشان می دهد.

## Lshw

سخت افزار را نشان می دهد

## Device UUIDs

هر دستگاه یک شناسه دارد اگر شما در مورد `dev/sda` صحبت می کنید، شما در مورد "هارد دیسک نخست" سیستم خود صحبت می کنید، اما اگر می خواهید یک درایو خاص برای `home/` باشد، شما باید از `UUID` استفاده کنید.

`Uuid` دستگاههای متصل به سیستم را میتواند در `proc/mounts/` دید.

```
root@life:/dev# cat /proc/mounts
rootfs / rootfs rw 0 0
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
udev /dev devtmpfs rw,relatime,size=4014804k,nr_inodes=1003701,mode=755 0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0
tmpfs /run tmpfs rw,nosuid,noexec,relatime,size=806028k,mode=755 0 0
/dev/disk/by-uuid/1651a94e-0b4e-47fb-aca0-f77e05714617 / ext4
rw,relatime,errors=remount-ro,data=ordered 0 0
```

## Runlevels

سطح `Runlevels` تعریف می کند که چه وظایف می تواند در حالت فعلی (یا سطح اجرا) یک سیستم لینوکس انجام شود

- 0- Halt
- 1- Single user mode (recovery)
- 2- Debian/Ubuntu default
- 3- RHEL/Fedora/SUSE text mode

- 4- free
- 5- RHEL/Fedora/SUSE graphical mode
- 6- reboot

توجه: سطح ۱ در حالت تک کاربر است

## استفاده از دستورات **runlevel** و **telinit**:

```
# runlevel
N ۳
# telinit ۵
# runlevel
۳ ۵
/etc/inittab
```

در حال جایگزینی با **upstart** و **systemd** است، اما هنوز در برخی سیستم های قدیمی م ورد استفاده است.

```
#
# inittab          This file describes how the INIT process should set up
#                  the system in a certain run-level.
#
# Author:          Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#                  Modified for RHS Linux by Marc Ewing and Donnie Barnes
#
# Default runlevel. The runlevels used by RHS are:
#  0 - halt (Do NOT set initdefault to this)
#  1 - Single user mode
#  2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#  3 - Full multiuser mode
#  4 - unused
#  5 - X11
#  6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# When our UPS tells us power has failed, assume we have a few minutes
# of power left.  Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
```

```
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"
```

```
# If power was restored before the shutdown kicked in, cancel it.
```

```
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"
```

```
# Run gettys in standard runlevels
```

```
1:2345:respawn:/sbin/mingetty tty1
```

```
2:2345:respawn:/sbin/mingetty tty2
```

```
3:2345:respawn:/sbin/mingetty tty3
```

```
4:2345:respawn:/sbin/mingetty tty4
```

```
5:2345:respawn:/sbin/mingetty tty5
```

```
6:2345:respawn:/sbin/mingetty tty6
```

```
# Run xdm in runlevel 5
```

```
x:5:respawn:/etc/X11/prefdm -nodaemon
```

این فرمت است:

```
id:runlevels:action:process
```

تمام اسکریپت ها در اینجا هستند:

```
ls -ltrh /etc/init.d
```

و شروع / توقف در سطوح اجرا از این فهرست ها کنترل می شود:

```
root@life:~# ls /etc/rc2.d/
```

## Shutdown

روش پیشنهادی برای خاموش کردن سیستم استفاده از دستور **shutdown** است، این دستور بعد از ۵ ثانیه مکس، سیستم را به **runlevel ۱** می برد

**h-** سیستم را متوقف (خاموش) خواهد کرد

**r-** سیستم را دوباره راه اندازی می کند

زمان **hh:mm** یا **n** (دقیقه) یا **now** است

اگر دستور در حال اجرا باشد، **ctrl + c** یا **shutdown -c** آن را لغو می کند

```
shutdown -r 60 Reloading updated kernel
```

اگر شما این فرمان را لغو کنید، کاربران خبری دریافت نمیکنند، شما می توانید از دستور **"wall"** استفاده کنید تا به آنها بگویید که خاموش شدن لغو شده است.

## Upstart

**upstart** رویداد جانشینی برای بوت بر اساس **sbin/init/** هستش که اجرای سرویس ها و وظایفی

در حال بوت را مدیریت می کند و هنگام خاموش شدن آنها را متوقف می کند و در طول اجرا بر

سیستم نظارت میکند

ویژگی های کلیدی:

اجرا و متوقف کردن وظایف و سرویس ها توسط رویداد ها (**events**)

رویداد های ساخته شده همچون وظایف و سرویس ها هستند و شروع و متوقف می شوند

رویدادها ممکن است پردازشها را از جاهای دیگر سیستم دریافت کنند  
سرویسها ممکن بعد از حرکات غیر منتظره برگردند و کار کنند  
سرپرستی و قابلیت برگشت برنامه کمکی که جدا از هم هستند از پردازشهای اصلی از هم مجزا شدند  
ارتباط با برنامه کمکی **init** بر بستر **D-bus**  
کاربر سرویسها خود می‌توند سرویس را شروع و متوقف کند  
ویژگیهای برنامه ریزی شده:  
رویدادها تولید شده در زمان خاص به پایان می‌رسند یا برنامه ریزی شده در زمان خاص (تحت بررسی)  
رویدادها تولید شده به عنوان فایل و دایرکتوریها در زمان پردازش تغییر می‌کنند  
نحوه استفاده:

در هنگام بوت گراب کلید **shift** را زده و به قسمت **advanced** رفته و گزینه **upstart** انتخاب می‌کنیم

فعالیتها در **/etc/init/** و زیر شاخهها آن تعریف می‌شود. (در **ubuntu** استفاده می‌شود)  
**initctl list**

## **systemd**

که در سیستمهای **Fedora** و **SUSE** استفاده می‌شود:

**Systemctl**

## معرفی مفهوم کتابخانه های اشتراکی یا **shared libraries**

فایل‌های کتابخانه یا **Library** ها در هر سیستم عاملی و برای تقریباً هر نرم افزار وجود دارند ، در واقع اول بهتر است تعریفی از یک فایل کتابخانه یا **Library** داشته باشیم ، یک **Library** در واقع شامل یک سری کد کامپایل شده است که تمامی اشیاء یا **Object** های مورد نیاز برای اجرای یک نرم افزار را بصورت مجتمع داخل یک فایل نگهداری می کند. اگر شما یک نرم افزار را در نظر بگیرید ، این نرم افزار دارای یک سری **Function** ها یا توابع می باشد که هر کدام کاری انجام می دهند ، مثلاً یکی ورودی و یکی خروجی چاپ می کند ، یکی وظیفه ذخیره داده را بر عهده دارد و ... همه اینها قرار نیست از ابتدا برنامه نویسی شوند ، یک برنامه نویس می تواند با استفاده از کتابخانه ها این کارها را انجام بدهد بدون اینکه نیاز به نوشتن همه آنها از ابتدا باشد ، برای مثال شما دیگر نیازی نیست که برنامه دریافت ورودی اطلاعات را دوباره از نو بنویسید بلکه کافیست از کتابخانه ای که وظیفه دریافت داده ها از ورودی را دارد آنها را دریافت کنید . اگر بخواهیم خیلی ساده درک مبحث کتابخانه ها را داشته باشیم شما می توانید ماژول ها را در نظر بگیرید ، یعنی هر ماژول یک کار را انجام می دهد و کتابخانه ها هم برای خودشان وظایفی بر عهده دارند. یک کتابخانه ممکن است شامل یک یا چند تابع یا **Function** باشد که حاوی یک سری کد برای انجام یک سری کارها است ، اگر بخواهیم در سیستم عامل لینوکس مثال بزنیم شما می توانید کتابخانه ای به نام **pthread** را در لینوکس تصور کنید ، این کتابخانه زمانی استفاده و فراخوانی می شود که شما در برنامه خود بخواهید از توابع وابسته یا **Related Function** ها استفاده کنید ، بصورت کلی ما کتابخانه ها یا بهتر بگوییم کتابخانه های برنامه ها ( **Program Library** ) را به دو دسته زیر تقسیم بندی می کنیم :

1. کتابخانه های اشتراکی یا **Shared Library** ها

2. کتابخانه های ایستا یا **Static Library** ها

### **Shared Library** یا کتابخانه های اشتراکی چه هستند ؟

**Share Library** ها یا کتابخانه های اشتراکی در واقع **Library** هایی هستند که در هنگام اجرا شدن هر برنامه ای می توانند به آنها لینک شوند. در واقع آنها در اصطلاح فنی در زمان **Run-Time** یک برنامه می توانند به آن متصل شوند. در واقع آنها کدهایی هستند که هر زمان می توانند در حافظه سیستم **Load** شوند و شروع به کار کنند. زمانیکه اینگونه **Library** ها در حافظه **Load** شدند از آن به بعد می توانند بصورت همزمان در **N** برنامه مختلف مورد استفاده قرار بگیرند. خوب این چه مزیتی برای سیستم و منابع آن دارد ؟ دقت کنید که ممکن است برنامه شما و چند برنامه دیگر بخواهند بصورت همزمان از یک کتابخانه استفاده کنند ، اگر قرار باشد هر بار با فراخوانی یک نرم افزار یک کتابخانه در حافظه سیستم **Load** شوند حافظه زیادی اشغال می شود اما زمانیکه از **Shared Library** ها استفاده می کنیم فقط یکبار **Library** مورد نظر ما در حافظه **Load** می شود و از آن به بعد هر نرم افزار که به **Library** مورد نظر نیاز داشته باشد به آن می تواند دسترسی پیدا کند. یکی از نکات بسیار خوب در خصوص **Shared Library** ها این است که برنامه نویس ها می توانند قابلیت ماژولار بودن یا ماژوله شدن را به **Library** ها بدهند ، آنها می توانند کد **Library**

مورد نظر را تغییر بدهند بدون دغدغه اینکه بخواهند کل نرم افزارشان را **Recompile** کنند ، فرض کنید اگر نرم افزاری همانطور که گفته شد از **Library** به نام **pthread** استفاده کند ، ما آن را باز می کنیم ، تغییر می دهیم و بروز رسانی می کنیم و دست به نرم افزار نمی زنیم ، به محض اجرا شدن نرم افزار از **Library** بروز رسانی شده خودکار استفاده می شود ، نامگذاری یک **Shared Library** یا بهتر بگوییم نحوه دسترسی به **Shared Library** ها به شکلهای زیر انجام می شود :

نامگذاری که دارای **lib** در اسم و پسوند **so** با استفاده از **linker** ها انجام می شود ، برای مثال **libpthread.so**

نامگذاری **Fully Qualified Name** یا **soname** که دارای **lib** در اسم و پسوند **so** با استفاده از یک شماره نسخه یا **version number** است ، برای مثال **libpthread.so.1**

نامگذاری **Real Name** یا نام واقعی که دارای **lib** در اسم و پسوند **so** می باشد علاوه بر شماره نسخه یا **version number** یک عدد دیگر هم در ادامه دارد که به عنوان **Release Number** شناخته می شود ، برای مثال **libpthread.so.1.1**

اگر شما تغییری در کدهای یک **shared library** انجام بدهید که نحوی که دیگر این **shared library** جدید با **shared library** قدیمی هماهنگی نداشته باشد شما شماره نسخه یا **version number** آن را تغییر داده اید و یک نسخه جدید از **shared library** ایجاد کرده اید. برای مثال اگر یک **function** یا تابع بصورت کامل از سیستم حذف شود قطعا یک نسخه جدید از آن **Library** برای اجرای تابع مورد نیاز خواهد بود. آن عدد کوچک در انتهای نام و بعد از نسخه در واقع به این معناست که **Shared Library** شما مشکل یا **Bug** داشته است و یک سری تغییرات بر روی آن ایجاد شده است اما این تغییرات باعث نمی شود که بصورت کلی این **Library** با نسخه های قبلی آن هماهنگ نباشد و **Compatibility** با نسخه های قبلی همچنان پابرجاست. برای مثال یک **Bug fix** ساده نمی تواند کارایی کل **Shared Library** را با محصولات قبلی دچار اختلال و مشکل کند بنابراین یک عدد صرفاً به انتهای آن به نشانه برطرف شدن باگهای مربوطه اضافه خواه شد اما نسخه تغییری نخواهد کرد.

## Ldd

این دستور وابستگی های پکیج را به نمایش میگذارد<sup>ix</sup>

در مواقعی نیاز دارید تا کتابخانه های استفاده شده یک برنامه را دقیقاً متوجه شوید تا از آن ها در برنامه نویسی خود استفاده کنید و یا حتی آن کتابخانه به خصوص را آپدیت کنید توسط دستور **ldd** **/usr/bin/ssh** به طور مثال می توانید تمامی کتابخانه های مورد استفاده دستور **ssh** را مشاهده کنیم.



## Ldconfig (پیکربندی اتصال دهنده پویا اتصالات زمان اجرا)

زمانی که برنامه ای به اجرا در آورده میشود (مثلا برنامه ی **ls**)، این برنامه برای اجرا به یک سری کتابخانه نیازمند است و باید این کتابخانه ها در داخل **ram** قرار بگیرد. ابزارهایی هستند که این کار را انجام می دهند. به عنوان مثال در هنگام اجرای دستور **ls** ابتدا چک می شود که دستور به چه کتابخانه هایی احتیاج دارد و از بین این کتابخانه ها کدامیک در حال حاضر در داخل **ram** نیست، پس آن را به داخل **ram** برده و در نتیجه دستور **ls** به درستی کار خواهد کرد. این ابزار **linker** نام دارد. در یک سیستم عامل تعداد زیادی کتابخانه وجود دارد و **linker** نمی تواند هر بار برای اجرای هر برنامه، تمامی مسیر های کتابخانه ها را به صورت آنلاین بگردد لذا از دیتابیس استفاده می کند که در آن، آدرس دقیق هر کتابخانه نوشته شده است. پس زمانی که **linker** به کتابخانه ای احتیاج داشته باشد ابتدا در دیتابیس خود به دنبال نام آن گشته سپس مسیر آن را از داخل **db** می خواند و در آخر آن را در داخل **ram** قرار می دهد. این دیتابیس را می توان در دو فایل زیر دید که اولی یک فایل متنی و دومی یک فایل باینری است

/etc/ld.so.conf

/etc/ld.so.cache

بصورت پیش فرض در داخل این دیتابیس، بسیاری مسیر و کتابخانه وجود دارد و حتی ممکن است کتابخانه ای جدید بصورت دستی به این دیتابیس اضافه بشود. برای این منظور ابتدا مسیر کتابخانه های خود را در فایل متنی

### ld.so.conf

اضافه می کنیم سپس با استفاده از دستوری دیگر به نام

### ldconfig

ایندکس کتابخانه های جدید (موجود در مسیری که اضافه کردیم) را به دیتابیس اضافه می کنیم

etc/ld.so.cache/

### ldconfig

توجه داشته باشید که برای اضافه کردن کتابخانه به فایل باینری **ld.so.cache** نیاز به یوزر **root** است

### Static library

تمام توابع مورد نیاز همواره باید همراه برنامه باشد، حجم برنامه خیلی بزرگ خواهد شد

## سامانه مدیریت بسته (package manager)

در علم نرم افزار، سامانه مدیریت بسته یا مدیر بسته، به مجموعه ابزارهایی گفته می شود که برای نصب، بروزرسانی، پیکربندی یا حذف یک بسته نرم افزاری در سیستم عامل استفاده می شود. مدیر بسته معمولاً شامل پایگاه داده ای حاوی نسخه بسته نرم افزاری و وابستگی های آن می باشد؛ این اطلاعات از تداخل بسته ها با یکدیگر جلوگیری بعمل آورده و همه پیش نیازهای نصب یک بسته نرم افزاری را فراهم می آورد. بسته ها شامل نرم افزارها، برنامه های کاربردی و داده ها می باشند. بسته ها همچنین حاوی فایل های فراداده می باشند که اطلاعاتی از قبیل نام نرم افزار، توضیحی در باب وظیفه نرم افزار، شماره نسخه، سازنده، چک سام و لیست وابستگی های نرم افزار را در خود جای می دهند. همزمان با نصب و راه اندازی، فایل های فراداده در پایگاه داده محلی سیستم ذخیره می شوند.<sup>x</sup>

### مدیر بسته های دبیان

دبیان یک سازمان داوطلب است که توزیع های سازگار بسته های باینری قبل از کامپایل نرم افزار اوپن سورس را ایجاد می کند و آنها را از آرشیو خود توزیع می کند. سیستم مدیریت بسته های **Debian** زمانی که به درستی مورد استفاده قرار می گیرد، کاربر را قادر می سازد مجموعه ای از بسته های باینری را به سیستم از بایگانی نصب کند. در حال حاضر ۵۹۵۰۹ بسته برای معماری **amd64** وجود دارد.<sup>xi</sup>

## فهرست ابزارهای مدیریت بسته های Debian

package	description
<a href="#">apt</a>	Advanced Packaging Tool (APT), front-end for dpkg providing "http", "ftp", and "file" archive access methods (apt/apt-get/apt-cache commands included)
<a href="#">aptitude</a>	interactive terminal-based package manager with aptitude(8)
<a href="#">tasksel</a>	tool for selecting tasks for installation on the Debian system (front-end for APT)
<a href="#">unattended-upgrades</a>	enhancement package for APT to enable automatic installation of security upgrades
<a href="#">dselect</a>	terminal-based package manager (previous standard, front-end for APT and other old access methods)
<a href="#">dpkg</a>	package management system for Debian
<a href="#">synaptic</a>	graphical package manager (GNOME front-end for APT)
<a href="#">apt-utils</a>	APT utility programs: apt-extracttemplates(1), apt-ftparchive(1), and apt-sortpkgs(1)
<a href="#">apt-listchanges</a>	package change history notification tool
<a href="#">apt-listbugs</a>	lists critical bugs before each APT installation
<a href="#">apt-file</a>	APT package searching utility — command-line interface
<a href="#">apt-rdepends</a>	recursively lists package dependencies

مدیر سیستم جدید دبیان باید با انتشار پایدار دبیان با استفاده از به روزرسانی های امنیتی تنها باقی بماند. منظورم این است که بعضی از اقدامات معتبر زیر بهتر به عنوان یک احتیاط اجتناب می شود، تا زمانی که سیستم دبیان را کاملاً درک کنید. در اینجا چند یادآوری وجود دارد.

- سورس های تست و یا ناپایدار در **"etc/apt/sources.list/"** را وارد نکنید.
- سورس های استاندارد توزیع خود را با دیگر آرشیو ها در **"etc/apt/sources.list/"** مخلوط نکنید.

- " /etc/apt/preferences " را ایجاد نکنید.
- رفتار پیش فرض ابزار مدیریت بسته را از طریق فایل های پیکربندی بدون اطلاع از اثرات کامل آن تغییر ندهید.
- بسته های تصادفی را با " <dpkg -i <random\_package " نصب نکنید.
- بسته های تصادفی را با " <dpkg --force-all -i <random\_package " نصب نکنید.
- فایل ها را در " /var / lib / dpkg / " پاک نکنید و یا تغییر نکنید.
- فایل های سیستم را با نصب برنامه های نرم افزاری که مستقیماً از منبع تهیه شده اند، بازنویسی نکنید.
- در صورت لزوم آنها را به "usr/local/" یا "opt/" نصب کنید.

## عملیات مدیریت پایه بسته با فرمان با استفاده از **apt** ، **aptitude** و **apt-get** / **apt-cache**

apt syntax	apt-get/apt-cachesyntax	description
apt update	apt-get update	update package archive metadata
apt install foo	apt-get install foo	install candidate version of "foo" package with its dependencies
apt upgrade	apt-get upgrade	install candidate version of installed packages without removing any other packages
apt full-upgrade	apt-get dist-upgrade	install candidate version of installed packages while removing other packages if needed
apt remove foo	apt-get remove foo	remove "foo" package while leaving its configuration files
apt autoremove	apt-get autoremove	remove auto-installed packages which are no longer required
apt purge foo	apt-get purge foo	purge "foo" package with its configuration files
apt clean	apt-get clean	clear out the local repository of retrieved package files completely
apt autoclean	apt-get autoclean	clear out the local repository of retrieved package files for outdated packages
apt show foo	apt-cache show foo	display detailed information about "foo" package
apt search <regex>	apt-cache search <regex>	search packages which match <regex>
N/A	apt-mark showmanual	list manually installed packages

### مدیر بسته های ردهت

ابزار **Red Hat Package Management** تنها در **Red Hat** بلکه در سیستم های **Fedora**، **CentOS** و **SUSE** و همچنین بسیاری از توزیع های دیگر لینوکس استفاده می شود که از این نوع مدیریت نرم افزار استفاده می کنند. با استفاده از دستورات **RPM** خام نسبت به سیستم مخزن نرم افزار **YUM**، مزایای آن را دارد. ما می توانیم آخرین نسخه نرم افزار مستقیم را از وب سایت های فروشندگان دانلود کنیم، منتظر نماییم تا آنها در یک مخزن آنلاین ظاهر شوند. همچنین، با کار با ابزار **RAW**، ما بینش بهتر نسبت به وابستگی های نرم افزاری و آنچه که در سیستم های ما نصب

شده است، می گیریم. البته معایب همان طور که در آن ما باید وابستگی ها را مدیریت کنیم، باید مطمئن شویم که تمام نرم افزارهای مورد نیاز را نیز نصب و نصب کنیم. من فکر می کنم شما متوجه خواهید شد که چرا باید مکانیسم های RPM و YUM را برای امتحان بدانیم.

**Install** پنج عمل اصلی بر روی بسته های نصبی توسط RPM برای نصب بسته های rpm به کار می رود.

**Remove** برای پاک کردن، حذف بسته های rpm به کار می رود.

**Upgrade** برای به روز رسانی بسته های rpm به کار می رود.

**Verify** جهت بررسی بسته های مختلف rpm به کار می رود.

**Query** جهت بازبینی بسته های rpm به کار می رود.

چگونه یک بسته RPM را نصب کنیم؟

برای نصب بسته های rpm می بایست از گزینه **i** - استفاده می کنیم، برای مثال اقدام به نصب بسته **pidgin** می کنیم.

rpm -ivh pidgin-2.7.9-5.el6.2.i686.rpm

**i** - نصب بسته rpm

**V** - نمایش اتفاقات هنگام نصب

**h** - نمایش # هنگام نصب بسته

بررسی پیش نیازهای یک بسته RPM پیش از نصب آن

توسط دستور ذیل اقدام به بررسی پیش نیازهای بسته BitTorrent پیش از نصب می کنیم و تمایز پیش نیازها به درستی نمایش داده می شود.

چگونه یک بسته RPM را به روز رسانی کنیم؟

در صورتی که قصد به روز رسانی یک بسته rpm را داشته باشیم می توانیم از گزینه **U** - برای این منظور استفاده کنیم و یکی از مزایایی که در هنگام بسته rpm استفاده می کنیم این است که پیش از به روز رسانی بسته مورد نظر از آن یک پشتیبان تهیه کرده که در صورت عدم بروز رسانی صحیح بسته فعلی بازگردانی گردد.

rpm -Uvh nx-3.5.0-2.el6.centos.i686.rpm

آیا نرم افزارهای کاربردی در لینوکس، تنها با واسطه مدیر بسته های سیستم عامل نصب یا حذف می شوند؟

پاسخ این سوال، خیر است، بسیاری از بسته توسط سایر مدیر بسته ها از جمله مدیر بسته های اختصاصی زبان های برنامه نویسی یا شیوه های دیگر نصب می شوند، برای نصب بسته های نرم افزاری که میخواهیم به صورت آفلاین نصب کنیم، باید ابتدا فایل راهنمای آن را مطالعه کنیم تا از شیوه نصب آن اطلاع یابیم

### bash

بَش (به انگلیسی: Bash) یک پوسته یونیکس<sup>xii</sup> است که به صورت آزاد توسط برایان فاکس (به انگلیسی: Brian Fox) برای پروژه گنو نوشته شده است. نام بَش سرنام Bourne-again shell در انگلیسی است.

بش یک پوسته پازیکس با تعداد قابل توجهی افزونه است. در ابتدا این پوسته برای سیستم عامل گنو طراحی شده بود. اما بش در اکثر سیستم عامل های خانواده شبه-یونیکس قابل اجراست. پوسته پیش فرض در اکثر سیستم عامل هایی که از هسته لینوکس استفاده می کنند، بش است. همچنین پوسته پیش فرض سیستم عامل مک او اس ده که مبتنی بر داروین است نیز می باشد. این پوسته برای سیستم عامل مایکروسافت ویندوز به وسیله شبیه ساز پوزیکس به نام سیگورین انتقال داده شده است.

اکثر قریب به اتفاق اسکریپت های نوشته شده برای پوسته بون بدون تغییر در Bash نیز قابل اجرا هستند. همچنین در شکل دستورهای (Command Syntax) بش، از Korn shell و سی شل نیز الهام گرفته شده است، مانند ویرایش خط فرمان، تاریخچه فرمان، متغیرهای \$RANDOM و \$PPID و جایگزینی فرمان مطابق پازیکس (\$...). تکمیل خط فرمان نیز از ویژگی های پرکاربرد Bash است که می توان با زدن کلید تب نام ناکامل برنامه ها، فایل ها و متغیرها را به صورت خودکار کامل کرد.<sup>xiii</sup>

### Shell

شل یک محیط واسطه را برای ارتباط شما با سیستم فراهم میکند. شل از شما مقداری را گرفته محاسبات را بر اساس آن انجام میدهد. وقتی اجرای برنامه تمام شد خروجی را نمایش می دهد. شل محیطی است که میتواند دستورات برنامه ها و اسکریپت ها را انجام دهد.<sup>xiv</sup>

### echo

یکی از دستورهای خط فرمان یونیکس است. این دستور تمام آرگومانهایی را که به عنوان ورودی می گیرد را در خروجی استاندارد چاپ می کند. بعضی از نسخه های echo، گزینه های -e و -n را می پذیرند. گزینه -n باعث می شود تا کاراکتر خط جدید (به انگلیسی: newline) در پایان چاپ نشود. استفاده از گزینه -e باعث می شود بتوانیم از کاراکترهای ویژه استفاده کنیم.<sup>xv</sup>

کاراکترهای ویژه عبارتند از:

\n: یک کاراکتر خط جدید

<sup>1</sup> پازیکس عبارت است از مجموعه استانداردهایی که برای نامگذاری و تعریف شمایل رابط برنامه نویسی کاربردی در نام بین المللی IEEE 1003 محیط های شبه-یونیکس در آی ترپیل ایی تعریف شده اند. این استانداردها تحت نام کلی شناخته می شوند، امکان همسان سازی و ارتباط و پورت کردن آسان تر بین محیط های یادشده را فراهم ISO/IEC 9945 می آورد. واژه پازیکس پیشنهاد بنیانگذار بنیاد نرم افزار آزاد، ریچارد استالمن بود.

\t: یک کاراکتر **tab** افقی

\\: یک کاراکتر \

\a: پخش صدای بوق از ترمینال

\b: یک کاراکتر **backspace**

\e: یک کاراکتر **escape**

\v: یک کاراکتر **tab** عمودی

## env

یک فرمان پوسته یونیکس سیستم‌عامل‌های یونیکس شبه یونیکس است که برای چاپ فهرستی از متغیر محیطی یا اجرای ابزار دیگری در یک محیط تغییر داده شده بدون نیاز به ویرایش محیط فعلی است.<sup>xvi</sup>

## pwd

**pwd** (برگرفته از: **print working directory**) دستوری است در سیستم‌عامل‌های شبه یونیکس برای چاپ کردن مسیر دایرکتوری جاری.<sup>xvii</sup>

```
pwd$
/usr/local/bin
$ cd ..
$ pwd
/usr/local
```

## set, unset

مجموعه ای کمی پیچیده‌تر از **env** است. می‌تواند نحوه رفتار شما را پیکربندی کند.

```
echo $
himBH
$ echo $VAR1

$ set -u;echo $-
himuBH
$ echo $VAR1
-bash: VAR1: unbound variable
$ VAR1=v1;echo $VAR1
v1
$ unset VAR1;echo $VAR1
-bash: VAR1: unbound variable
```

## uname

uname یکی از دستوره‌های خط فرمان [یونیکس](#) است و برخی اطلاعات سیستم را چاپ می‌کند. مختصر شده *unix name* است

```
uname -a $
```

```
Linux life 4.4.0-138-generic #164+8.0trisquel2 SMP Tue Oct 23 23:09:29 UTC 2018 x86_64 x86_64  
x86_64 GNU/Linux
```

سایر گزینه ها :

نام گزینه	عملکرد
-a	نام اطلاعات را چاپ می‌کند.
-s	نام هسته سیستم عامل را چاپ می‌کند.
-n	نام میزبان در شبکه را چاپ می‌کند.
-r	شماره انتشار هسته سیستم عامل را چاپ می‌کند.
-v	نسخه هسته را چاپ می‌کند.
-p	نوع پردازنده را چاپ می‌کند.
	نام سیستم عامل را چاپ
-o	می‌کند.

## Man

مستنداتی است که برای موجودیتی که عموماً در سیستم عامل یونیکس یا شبه یونیکس می‌باشد وجود دارد. چنین موجودیت‌هایی شامل برنامه‌های کامپیوتری (کتابخانه‌ها فراخوان‌های سیستمی)، استانداردهای رسمی، اصطلاحات و قراردادهای و حتی مفاهیم انتزاعی می‌باشند. کاربر می‌تواند با فراخوان دستور راهنما (به انگلیسی: *man*) را احضار کند. کاربر برای مطالعه صفحه راهنمای یک دستور در سیستم عامل یونیکس خواهد داشت:

```
man <command_name>
```

## history

"تاریخ" یک ویژگی در بسیاری از پوسته‌های سیستم عامل، برنامه‌های جبر کامپیوتری و سایر نرم افزارهایی است که به کاربر اجازه می‌دهد تا دستورات قبلی را بازخوانی، ویرایش و تجدید نظر کند. *bash*، دستورات شما را در یک پرونده تعریف شده در *HISTFILE ev* ذخیره می‌کند. تاریخچه فرمانها را کامل نشان می‌دهد (۵۰۰ دستورات عمل به طور معمول در آن ذخیره می‌شود اما *HISTSIZ* را میتوان تغییر داد).



## Paths

محل پیدا کردن فایل دستورات را در خود دارد و **bash** از طریق آن دستورات را می یابد.

```
i@life:~$ echo $PATH
```

```
/home/i/miniconda3/bin:/home/i/bin:/home/i/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/java-8-oracle/jre/bin
```

## cat

برنامه **cat** یک ابزار استاندارد یونیکس و یکی از دستوره‌های پایه‌ای و اصلی در سیستم‌عامل‌های شبه یونیکس می‌باشد که از آن می‌توان برای الحاق و چاپ کردن محتوای یک پرونده در خروجی استاندارد استفاده کرد. نام این دستور از واژه **catenate** به معنای متصل کردن به هم گرفته شده است. این دستور فایلی را به صورت ترتیبی می‌خواند و سپس آن را به همان صورت در خروجی استاندارد چاپ می‌کند. همچنین می‌توان چند فایل را در خط فرمان مشخص کرد که در این صورت همه آنها به ترتیب از چپ به راست خوانده شده و در خروجی استاندارد نوشته می‌شوند. اگر در خط فرمان فایلی مشخص نشده باشد یا یک (-) نوشته شده باشد، **cat** فایل را از ورودی استاندارد می‌خواند. اگر فایل مورد یک سوکت باشد، **cat** به آن متصل شده و آن را تا رسیدن به کاراکتر **EOF** می‌خواند<sup>xviii</sup> گزینه‌های این دستور عبارتند از:

نام گزینه	کاربرد
-b	خطوط غیر تهی را شماره گذاری می‌کند.
-e	کاراکترهای غیر قابل چاپ را هم نمایش می‌دهد. همینطور در پایان هر خط هم یک کاراکتر \$ چاپ می‌شود.
-n	تمام خطوط را شماره گذاری می‌کند و در خروجی چاپ می‌کند.
-u	بافر خروجی را غیر فعال می‌کند.
-t	نمایش می‌یابد ^I کاراکترهای غیر قابل چاپ را هم نمایش می‌دهد. کاراکتر تب به صورت
-s	اگر چند خط خالی پشت سر هم در ورودی وجود داشته باشد، آنها را به یک خط خالی فشرده می‌کند و در خروجی چاپ می‌کند.
-v	نمایش -X به صورت ctrl-x کاراکترهای غیر قابل چاپ را هم نمایش می‌دهد. کاراکترهای کنترلی نمایش می‌یابد ^? به صورت delete می‌یابد. کاراکتر

## CUT

این دستور بخش خاصی از خط‌های فایلی که به آن داده شود را در قالب یک خروجی استاندارد نمایش می‌دهد.

```
cut -c1-6 namef
```

بخشی از قابلیت‌ها:

STDIN: اولین شانزده حرف هر خط  
cut -c {{1-16}}

شانزده حرف اول هر خط از فایل های داده شده  
cut -c {{1-16}} {{file}}  
سه کاراکتر انتهای هر خط را بردارید  
cut -c{{3-}}

## head

یک برنامه یونیکس و سیستم های مشابه یونیکس است که برای نمایش ابتدای یک فایل متنی یا داده های لوله گذاری استفاده می شود.

head [options] <file\_name>

## tail

یک برنامه یونیکس و سیستم های مشابه یونیکس است که برای نمایش انتهای یک فایل متنی یا داده های لوله گذاری استفاده می شود.

tail [options] <file\_name>

## ( | ) Piping

در سیستم عامل های شبه یونیکس یک خط لوله معادل همان مفهوم کلی خط لوله نرم افزار است. این مفهوم به تعدادی پروسه اشاره دارد که به صورت زنجیر وار به یکدیگر متصل هستند. به این صورت که خروجی استاندارد یک فرآیند به ورودی استاندارد فرآیندی بعدی متصل است و به همین ترتیب. هر اتصال به وسیله یک لوله ناشناس پیاده سازی شده است. این ایده در ابتدا توسط داگلاس مکیلروی ابداع شد و در حقیقت تمثیلی از یک خط لوله انتقال واقعیت است. این ایده به سرعت پذیرفته شد و جزء یکی از ارکان اصلی فلسفه یونیکس درآمد. مفهوم لوله کشی فرآیندها در زیر نشان داده شده است. خط اول حالت کلی و انتزاعی را نشان می دهد و خط دوم هم یک مثال واقعی از اتصال سه برنامه ls, grep و more را نشان می دهد.

```
program1 | program2 | program3  
ls -l | grep key | more
```

و یا در مثالی دیگر

```
i@life:~/w$ ls -l | sort  
12  
62  
amir  
kaveh  
neda  
you  
i@life:~/w$ ls -l | sort -r  
you  
neda  
kaveh  
amir  
62  
12
```

## ( < ) Redirection

یک روش مفید دیگر برای کنترل جریانها < است. این به شما کمک می کند که خروجی خود را هدایت کنید (بیشتر به یک فایل).

```
i@life:~/Downloads$ ls -ltrh
total 3.4G
-rwxrwxrwx 1 i i 436K Aug 27 11:43 tsm100.docx
-rwxrwxrwx 1 i i 4.7M Sep  5 12:59 VIDEO WALL.docx
-rwxrwxrwx 1 i i 1.2M Sep  5 12:59 tsm100-0.docx
-rwxrwxrwx 1 i i 692K Sep  5 12:59 Stream Recorder.docx
-rwxrwxrwx 1 i i 545K Sep 20 09:42 GNU-FDL-OO-LPI-201-0.1.pdf
-rwxrwxrwx 1 i i 10M Oct  1 10:45 pandas.pdf
-rw-rw-r-- 1 i i 1.0G Oct 11 14:33 Oracle Database 12c Release 2 (12.2.0.1.0) Linux.part1.rar
drwxrwxr-x 3 i i 4.0K Oct 11 15:11 OracleDatabase18c Documents
-rw-rw-r-- 1 i i 1.0G Oct 11 15:14 Oracle Database 12c Release 2 (12.2.0.1.0) Linux.part2.rar
-rw-rw-r-- 1 i i 1.0G Oct 11 15:48 Oracle Database 12c Release 2 (12.2.0.1.0) Linux.part3.rar
-rw-rw-r-- 1 i i 294M Oct 11 16:24 Oracle Database 12c Release 2 (12.2.0.1.0) Linux.part4.rar
-rwxrwxrwx 1 i i 1.7M Oct 13 18:26 lecture6.pdf
-rwxrwxrwx 1 i i 201K Oct 13 18:28 imm6614.pdf
-rwxrwxrwx 1 i i 1.6M Oct 14 08:21 10.1.1.682.5996.pdf
drwxrwxrwx 2 i i 4.0K Oct 14 21:15 Tsm Doc
-rwxrwxrwx 1 i i 2.9M Oct 16 08:58 MSRT-Bank @GeneralEnglish.pdf
drwxrwxrwx 4 i i 4.0K Oct 16 21:02 datacamp python
-rw-rw-r-- 1 i i 3.1M Oct 18 09:26 linuxsys.pdf
drwxrwxrwx 11 i i 4.0K Nov  5 15:03 mividi
drwxrwxr-x 2 i i 4.0K Nov  7 15:37 congre advar
drwxrwxr-x 2 i i 4.0K Nov  7 15:41 dig
drwxrwx--- 2 i i 4.0K Nov  7 15:42 python-3.7.docs-pdf
drwxrwxr-x 2 i i 4.0K Nov  7 23:26 Telegram Desktop
drwx----- 2 i i 4.0K Nov  8 14:33 persian-fonts-linux-master
drwxrwxrwx 3 i i 4.0K Nov  8 16:37 class
```

```
i@life:~/Downloads$ ls -ltrh > directory_data
```

```
i@life:~/Downloads$ cat directory_data
```

```
total 3.4G
```

```
-rwxrwxrwx 1 i i 436K Aug 27 11:43 tsm100.docx
-rwxrwxrwx 1 i i 4.7M Sep  5 12:59 VIDEO WALL.docx
-rwxrwxrwx 1 i i 1.2M Sep  5 12:59 tsm100-0.docx
-rwxrwxrwx 1 i i 692K Sep  5 12:59 Stream Recorder.docx
-rwxrwxrwx 1 i i 545K Sep 20 09:42 GNU-FDL-OO-LPI-201-0.1.pdf
-rwxrwxrwx 1 i i 10M Oct  1 10:45 pandas.pdf
-rw-rw-r-- 1 i i 1.0G Oct 11 14:33 Oracle Database 12c Release 2 (12.2.0.1.0) Linux.part1.rar
drwxrwxr-x 3 i i 4.0K Oct 11 15:11 OracleDatabase18c Documents
```

```

-rw-rw-r-- 1 i i 1.0G Oct 11 15:14 Oracle Database 12c Release 2 (12.2.0.1.0) Linux.part2.rar
-rw-rw-r-- 1 i i 1.0G Oct 11 15:48 Oracle Database 12c Release 2 (12.2.0.1.0) Linux.part3.rar
-rw-rw-r-- 1 i i 294M Oct 11 16:24 Oracle Database 12c Release 2 (12.2.0.1.0) Linux.part4.rar
-rwxrwxrwx 1 i i 1.7M Oct 13 18:26 lecture6.pdf
-rwxrwxrwx 1 i i 201K Oct 13 18:28 imm6614.pdf
-rwxrwxrwx 1 i i 1.6M Oct 14 08:21 10.1.1.682.5996.pdf
drwxrwxrwx 2 i i 4.0K Oct 14 21:15 Tsm Doc
-rwxrwxrwx 1 i i 2.9M Oct 16 08:58 MSRT-Bank @GeneralEnglish.pdf
drwxrwxrwx 4 i i 4.0K Oct 16 21:02 datacamp python
-rw-rw-r-- 1 i i 3.1M Oct 18 09:26 linuxsys.pdf
drwxrwxrwx 11 i i 4.0K Nov 5 15:03 mividi
drwxrwxr-x 2 i i 4.0K Nov 7 15:37 congre advar
drwxrwxr-x 2 i i 4.0K Nov 7 15:41 dig
drwxrwx--- 2 i i 4.0K Nov 7 15:42 python-3.7.docs-pdf
drwxrwxr-x 2 i i 4.0K Nov 7 23:26 Telegram Desktop
drwx----- 2 i i 4.0K Nov 8 14:33 persian-fonts-linux-master

```

## WC

WC به مفهوم تعداد کلمه است. این کاراکترها، خطوط و بایت را در جریان ورودی شمارش می کند.

```

i@life:~/Downloads$ wc lecture۶.pdf
lecture۶.pdf ۱۷۴۷۵۳۸ ۴۸۶۷۴ ۹۰۸۶

```

## expand & unexpand & tr

Expand زبانه ها را در یک stream با فضاها جایگزین (به طور معمول ۸، اما می تواند با -۱۲ برای ۱۲ تعریف شود):

```

i@life:~/w$ cat howcool
kaveh 5
sina 6
rubic 2
you 12
i@life:~/w$ od -tc howcool
0000000 j a d i \t 5 \n s i n a \t 6 \n r u
0000020 b i c \t 2 \n y o u \t 1 2 \n
0000036
i@life:~/w$ expand howcool | od -tc
0000000 j a d i 5 \n s i n a
0000020 6 \n r u b i c 2 \n y o
0000040 u 1 2 \n
0000051

```

Unexpand معکوس آن را انجام خواهد داد.

دستور tr به ترتیب A به ۱، B به ۲ و C به ۳ تبدیل می شود که شما باید ۱۲۳ Tr 'ABC' را انتخاب کنید. این یک فیلتر خالص است، بنابراین اگر شما نیاز به آن فایل را برای کار بر روی، شما باید از cat استفاده کنید:

```

i@life:~/w$ cat mydata
hello
this is second line
but as you can see we are
still writing
and this is getting longer
.
.
and longer
and longer!
i@life:~/w/lpic/101$ cat mydata | tr 'and' 'AND'
hello
this is second liNe
but As you cAN see we Are
still writiNg
AND this is gettiNg loNger
.
.
AND loNger
AND loNger!

```

## nl

به معنی تعداد خطوط

```

i@life:~/w$ nl mydata | head -3
 1 hello
 2 this is second line
 3 but as you can see we are

```

## fmt

فرمت حاشیه یک فایل متنیرا ویرایش کنید  
(اگر میخواهید از ستون عرضی ۸۰ تا ۶۰ استفاده کنید -W۶۰).

```

i@life:~/w$ fmt mydata
hello this is second line but as you can see we are still writing and
this is getting longer . . and longer and longer!

```

## sort & uniq

ورودی (ها) خود را انتخاب (مرتب) می کند.

```

i@life:~/w$ cat uses
you fedora
kaveh ubuntu
rubic windows
neda mac
i@life:~/w$ cat howcool
kaveh 5
sina 6
rubic 2
you 12
i@life:~/w$ sort howcool uses
kaveh 5
kaveh ubuntu
neda mac
rubic 2
rubic windows

```

```

sina    6
you     12
i@life:~/w$ uniq what_i_have.txt
laptop
socks
tshirt
ball
socks
glasses
i@life:~/w$ sort what_i_have.txt | uniq
ball
glasses
laptop
socks
tshirt

```

## join

که فایل ها را براساس فیلد تطبیق پیوند می دهد. فایل ها باید بر روی فیلد پیوست مرتب شوند.

```

i@life:~/w/lpic/101$ cat howcool
kaveh  5
sina   6
rubic  2
you    12
i@life:~/w/lpic/101$ cat uses
you fedora
kaveh ubuntu
rubic windows
neda mac
i@life:~/w/lpic/101$ sort howcool > howcool.sorted
i@life:~/w/lpic/101$ sort uses > uses.sorted
i@life:~/w/lpic/101$ join howcool.sorted uses.sorted
kaveh 5 ubuntu
rubic 2 windows
you 12 fedora

```

انجام مدیریت ساده فایل ها

**ls**

برای فهرست دایرکتوری ها و فایل ها استفاده می شود.

```
i@life:~$ ls -l
total 33644
-rw----- 1 i i 10751 Nov 2 09:47 basic%20linux%20iptv.odt_1.odt
-rw-rw-r-- 1 i i 1096043 Oct 13 17:43 bfonts.zip
drwxrwxrwx 2 i i 4096 Sep 23 12:27 bin
drwxrwxrwx 6 i i 4096 Sep 23 12:27 Calibre Library
drwxr-xr-x 2 i i 4096 Nov 7 15:36 Desktop
drwxr-xr-x 3 i i 4096 Oct 15 16:42 Documents
drwxr-xr-x 15 i i 4096 Nov 8 16:48 Downloads
drwxrwxrwx 10 i i 4096 Nov 8 13:03 DriveKaveh
-rw-rw-r-- 1 i i 176217 Oct 13 17:43 FPF.zip
-rw-rw-r-- 1 i i 1642522 Sep 27 14:48 get-pip.py
drwxrwxr-x 4 i i 4096 Sep 27 11:53 GNUstep
drwxr----- 22 i i 4096 Oct 5 20:08 hplip-3.18.7
drwxr----- 22 root root 4096 Oct 5 20:16 hplip-3.18.9
```

برخی قابلیت ها :

- -l is for long (more info for each file)
- -1 will print one file per line
- -t sorts based on modification date
- -r reverses the search (so -tr is reverse time (newer files at the bottom)).

**cp**

دیگر کپی می کند. اگر هدف یک دایرکتوری باشد، تمام /name به مکان /name این فایل ها را از یک محل منابع در آنجا کپی می شوند.

`cp source destination`

**mv**

فایل را تغییر مکان یا تغییر نام میدهد، نحوه انجام کار مانند کپی است.

**rm**

فایل را حذف میکند

**mkdir**

دایرکتوری را ایجاد میکند

**touch**

تاریخ اصلاح یک فایل را به زمان فعلی به روز رسانی خواهد کرد (اگر یک فایل ایجاد نشده باشد، آن را ایجاد می کند).

```
/touch$ ls -l
```

```
total 0
```

```
-rw-rw-r-- 1 i i 0 Jan 8 17:47 myfile
```

```
/touch$ touch myfile #after a minute
```

```
/touch$ ls -l
total 0
-rw-rw-r-- 1 i i 0 Jan  8 17:48 myfile
```

## Finding files

. به ما کمک می کند فایل های بر اساس معیارهایی پیدا کنیم .

```
$ find . -iname "[a-j]*"
```

```
./howcool.sort
./alldata
./mydir/howcool.sort
./mydir/newDir/insideNew
./howcool
```

## Identify a file

تعیین نوع فایل

```
$ file mydata.tab
```

```
mydata.tab: ASCII text
```

```
$ file /bin/bash
```

```
/bin/bash: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked
(uses shared libs), for GNU/Linux 2.6.32,
BuildID[sha1]=cb63ec0718f2022619814c04a5b6cd8a36752a83, stripped
```

```
$ file mydata.tab
```

```
mydata.tab: ASCII text
```

```
$ file /bin/bash
```

```
/bin/bash: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked
(uses shared libs), for GNU/Linux 2.6.32,
BuildID[sha1]=cb63ec0718f2022619814c04a5b6cd8a36752a83, stripped
```

```
$ file -i mydir
```

```
mydir: inode/directory; charset=binary
```

## Compressing files

### zip

در لینوکس استفاده می کنیم. بسیار آسان است و gzip و gunzip ما اغلب از

```
$ ls * -ltrh
```

```
-rw-r--r-- 1 i i 79K Dec 22 11:52
```

```
The.Equalizer.2014.1080p.BluRay.x264.anoXmous_eng.srt
```

```
$ gzip The.Equalizer.2014.1080p.BluRay.x264.anoXmous_eng.srt
```

```
$ ls * -ltrh
```

```
-rw-r--r-- 1 i i 30K Dec 22 11:52
```

```
The.Equalizer.2014.1080p.BluRay.x264.anoXmous_eng.srt.gz
```

```
$ gunzip The.Equalizer.2014.1080p.BluRay.x264.anoXmous_eng.srt.gz
```

```
$ ls * -ltrh
```

```
-rw-r--r-- 1 i i 79K Dec 22 11:52
```

```
The.Equalizer.2014.1080p.BluRay.x264.anoXmous_eng.srt
```

### bzip2

یکی دیگر از ابزار فشرده سازی است. تفاوتش این است که با یک الگوریتم فشرده سازی دیگر کار می کند.



## Archiving files

### *tar*

معمولترین ابزار آرشیو است. به صورت خودکار فایل آرشیو را از یک پوشه و تمام `tar` یا `TapeARchive` زیر شاخه های آن ایجاد کنید.

### *cpio*

یک لیست از فایل ها را دریافت می کند و بایگانی (یک فایل) از آن ایجاد می کند که بعدا می تواند باز شود.

```
$ ls | cpio -o > allfilesls.cpio
```

```
3090354 blocks
```

### **dd**

داده ها را از یک مکان به دیگری کپی می کند. این داده ها از فایل ها می آید:

```
$ cat howcool
```

```
kaveh 5
```

```
sina 6
```

```
rubic 2
```

```
you 12
```

```
$ dd if=howcool of=newcool
```

```
0+1 records in
```

```
0+1 records out
```

```
30 bytes (30 B) copied, 0.0227904 s, 1.3 kB/s
```

```
$ cat newcool
```

```
kaveh 5
```

```
sina 6
```

```
rubic 2
```

```
you 12
```

**if is In File      of is Out File**

: میتوانید بک آپ بگیرید

```
# dd if=/dev/sda of=backup.dd bs=4096    یا    # dd if=/dev/sda2 |gzip >backup.dd.gzip
```

## هدایت ورودی و خروجی استاندارد

در سیستم لینوکس اکثر پوسته ها از `stream` برای ورودی و خروجی (یک لیست از کاراکترها) استفاده می کنند. این `stream` ها می توانند از (و به سمت) چیزهای مختلف از جمله صفحه کلید، دستگاه بلوک (هارد، فلش `USB`، ...)، پنجره برنامه، ... باشد.

۱. `stdout` جریان خروجی استاندارد است که خروجی را از دستورات نمایش می دهد (توصیفگر فایل ۱)

۲. `stderr` جریان خطای استاندارد است که خروجی خطا از دستورات را نمایش می دهد (توصیفگر فایل ۲)

۳. `stdin` جریان ورودی استاندارد است که ورودی را برای دستورات فراهم می کند (توصیفگر فایل ۰)

اگر شما نیاز به کنترل خروجی خود دارید، می توانید `<n` یا `<<n` را اضافه کنید.

• `<n` فایل توضیحات `n` را به یک فایل یا دستگاه هدایت می کند. اگر فایل در حال حاضر وجود دارد آن را رونویسی می کند و اگر آن وجود نداشته باشد، آن ایجاد خواهد شد.

• `>>n` فایل توضیحات `n` را به یک فایل یا دستگاه هدایت می کند. اگر فایل در حال حاضر وجود دارد، جریان به انتهای آن افزوده خواهد شد و اگر آن وجود نداشته باشد، آن ایجاد خواهد شد.

## هدایت هر دو `stdout` و `stderr` به یک مکان

گاهی اوقات (در انجام وظایف خودکار می گویند) ما ترجیح می دهیم هر دو خروجی استاندارد و خطای استاندارد را به همان مکان بفرستیم، استفاده از `&&` و `&<` برای گفتن هر دو `stderr` و `stdout`

همچنین ممکن است از `&۱`، `&۲` و `&۰` برای اشاره به مکان فعلی `stdout`، `stderr` و `stdin` استفاده کنید. در این مورد `۱ & ۲ > file۱` به معنی تغییر خروجی به `file۱` و خروجی `stderr` به همان محل به عنوان `(file۱) stdout`

## ارسال به `null`

در لینوکس، `/dev/null` مانند یک سطل زباله است. شما می توانید هر چیزی را در آنجا بفرستید و ناپدید می شود. بنابراین طبیعی است که بگوییم:

```
$ ls j* m* > file1
ls: cannot access j*: No such file or directory
$ ls j* m* > file1 2>/dev/null
$ cat file1
mahmoodrm
minoo
mojtaba
```

## هدایت ورودی

عملگر `>` ورودی را هدایت می کند.  
اینجا اسناد

شما از `>>` و یک کلمه استفاده می کنید و پس از آن هر آنچه که شما وارد می کنید، `stdin` در نظر گرفته می شود تا شما تنها یک کلمه را در یک خط قرار دهید.

```
$ tr ' ' '.' << END_OF_DATA
> this is a line
> and then this
> we'll still type
> and,
> done!
> END_OF_DATA
this.is.a.line
and.then.this
we'll.still.type
and,
done!
```

### Pipes

ارسال می کند. شما از | برای (Piping stdout به stdin) لوله گذاری یک دستور را به یک ورودی دیگر دستور این کار استفاده میکنید.

## Xargs

این دستور ورودی را از `stdin` می خواند و از آن به عنوان استدلال استفاده می کند.

```
$ ls | xargs echo these are files:
these are files: errors f file1 fiona habib mahmoordm minoo mojtaba output out-
put.txt sina uses what_i_have.txt
```

### tee

اگر شما نیاز دارید خروجی را روی صفحه ببینید و همچنین آن را به یک فایل ذخیره کنید چه؟ شیوه `tee` در یک پنجره دیگر است. راه خوب استفاده از `tail -f` نامناسب تغییر مسیر به فایل و استفاده از فایل دارید، ابتدا آن را به `stderr` (اگر شما نیاز به ذخیره) و دادن یک یا چند نام فایل برای خروجی استاندارد است: این خروجی را در آن فایل ها می نویسند و همچنین آنها را در صفحه می نویسند. (هدایت کنید `stdout`):

```
$ ls -l | tee allfiles myfiles
allfiles
f
fiona
habib
mahmoordm
minoo
mojtaba
myfiles
sina
```

## ایجاد، نظارت و کشتن فرآیندها پیش زمینه و کارهای پس زمینه

یکی از نکات برجسته لینوکس در روزهای آغازین، توانایی اجرای همزمان برنامه های بسیار بود. این کار با ارسال برنامه ها به پس زمینه انجام می شود. به طور معمول اگر شما برنامه ای را در ترمینال اجرا می کنید، ترمینال شما را متوقف می کند، اما ارسال یک فرمان به پس زمینه از این کار جلوگیری می کند:

```
xeyes &
```

اما برای توقفش چه باید بکنیم؟ ما می توانیم آن را با **Ctrl + C** بلاک / لغو کنیم یا با استفاده از **Ctrl + Z** آن را معلق کنیم.

```
$ xeyes
^Z
[1]+  Stopped                  xeyes
$ jobs
[1]+  Stopped                  xeyes
$ bg
[1]+ xeyes &
$ jobs
[1]+  Running                  xeyes &
$ sleep 1000 &
[2] 7395
$ jobs
[1]-  Running                  xeyes &
[2]+  Running                  sleep 1000 &
$ fg %2
sleep 1000
^Z
[2]+  Stopped                  sleep 1000
$ jobs
[1]-  Running                  xeyes &
[2]+  Stopped                  sleep 1000
$ bg sle
[2]+  sleep 1000 &
$ jobs
[1]-  Running                  xeyes &
[2]+  Running                  sleep 1000 &
```

## nohup

فرمان **nohup** به شما این امکان را می دهد که دستورات خود را اجرا کنید حتی پس از خروج از سیستم و خروجی آن را در **nohup.out** می نویسد:

```
nohup ping 4.2.2.4
nohup: ignoring input and appending output to 'nohup.out'
^C$ cat nohup.out
PING 4.2.2.4 (4.2.2.4) 56(84) bytes of data.
64 bytes from 4.2.2.4: icmp_seq=1 ttl=51 time=225 ms
64 bytes from 4.2.2.4: icmp_seq=3 ttl=51 time=223 ms

--- 4.2.2.4 ping statistics ---
4 packets transmitted, 2 received, 50% packet loss, time 3010ms
rtt min/avg/max/mdev = 223.584/224.767/225.950/1.183 ms
```

## kill

شما می توانید فرآیندهای سیگنال را کنترل کنید. در واقع فشار دادن **Ctrl + C** و **Ctrl + Z** همچنین سیگنال ها را ارسال می کند.

```
$ jobs
[3]  Running                  xeyes &
[4]  Running                  sleep 1000 &
[5]-  Running                  sleep 2000 &
```

```
[6]+  Running                  sleep 3000 &
$ kill %4
$ jobs
[3]   Running                  xeyes &
[4]   Terminated             sleep 1000
[5]-  Running                  sleep 2000 &
[6]+  Running                  sleep 3000 &
$ jobs
[3]   Running                  xeyes &
[5]-  Running                  sleep 2000 &
[6]+  Running                  sleep 3000 &
```

ها از طریق کشتن یا ارسال سیگنال های دیگر وجود دارد PID اگر همچنین امکان استفاده از

signal number	signal name	meaning
1	SIGHUP	Informing the process that its controlling terminal (like an ssh connection) is terminated
15	SIGTERM	normal termination request
9	SIGKILL	forcefully kills the process

## killall

سیگنال داده شده (یا ۱۵) را به تمام فرآیندهای با نام داده شده ارسال کنید:

```
$ jobs
[3]   Running                  xeyes &
[5]-  Running                  sleep 2000 &
[6]+  Running                  sleep 3000 &
$ ps -ef | grep sleep
i    7864  7651  0 21:07 pts/1    00:00:00 sleep 2000
i    7865  7651  0 21:07 pts/1    00:00:00 sleep 3000
i    7977  7651  0 21:14 pts/1    00:00:00 grep sleep
$ killall sleep
[5]-  Terminated             sleep 2000
[6]+  Terminated             sleep 3000
$ jobs
[3]+  Running                  xeyes &
$ ps -ef | grep sleep
i    7980  7651  0 21:14 pts/1    00:00:00 grep sleep
```

## ps

دستور ps نشان می دهد فرآیندهای در حال اجرا بر روی رایانه شما است.

```
$ sleep 1000 &
[1] 7678
$ sleep 1001 &
```

```
[2] 7679
$ xeyes &
[3] 7680
$ ps
```

```
  PID TTY          TIME CMD
 7651 pts/1    00:00:00 bash
 7678 pts/1    00:00:00 sleep
 7679 pts/1    00:00:00 sleep
 7680 pts/1    00:00:00 xeyes
 7681 pts/1    00:00:00 ps
```

اما استفاده از `ps -aux` (= `-aux`) یا `ps -ef` نیز رایج است و **ALL processes** در این سیستم را نشان می دهد:

```
$ ps -aux | wc -l
۲۵۴
```

## پیدا کردن فرآیندها

شما دیده ام که `ps -ef` فرآیندها را از همه کاربران نشان می دهد. ما می توانیم روی آن کار کنیم و ببینیم چه کسی در حال استفاده از `nano` است و شناسه فرآیند آن چیست:

```
$ ps -ef | grep nano
i        6213  4604   9 20:06 ?                00:04:43 nano
i        7725  7651   0 20:55 pts/1          00:00:00 grep nano
```

## top

فرآیندها تغییر می کنند و گاهی اوقات شما باید آنها را به صورت زنده بررسی کنید. دستور بالا به شما کمک خواهد کرد:

key during top	functionality
h	help
q	quit
M	sort based on memory usage"
c	show full commands
k	kill after asking pid and signal

## free

اطلاعاتی را در مورد حافظه سیستم نشان می دهد. پیش فرض کیلوبایت است، اما شما می توانید آن را با `m-` برای مگابایت، `g-` برای گیگابایت یا حتی `b-` برای بایت ها تغییر دهید:

```
$ free -m
              total        used        free      shared    buffers     cached
Mem:           7871         5231         2640          332         169         2195
-/+ buffers/cache:
Swap:          7627           0         7627
```

## uptime

فرمان `uptime` نشان می دهد که زمان، چه مدت سیستم است، چگونه کاربران ممکن است وارد شوید و میانگین بار ۱، ۵ و ۱۵ دقیقه:

```
$ uptime
21:18:52 up 1:24, 2 users, load average: 2.38, 2.64,
2.41
```

## متغیرهای محیطی

در بخش های پیش در مورد متغیرهای صحبت کردیم، میخواهیم در این مورد که چگونه در هنگام ورود به سیستم این متغیرها را تنظیم کنیم، صحبت کنیم

### login vs non-login shell

باز کنید؛ *shell* وارد می شوید، اما گاهی اوقات شما فقط یک *shell* به *ssh* گاهی اوقات شما از *user* یا رابط کاربری گرافیکی می گویند، این اتفاق زمانی می افتد که شما *GUI* که به آن وارد می کنید *shell* خود را برای ورود به یک *pass*.

مراحل ورودی به این ترتیب هستند:

1- /etc/profile is run

2- A line in /etc/profile runs whatever is in /etc/profile.d/\*

Now the global profile is loaded and system will go for user specific profiles:

3- /home/USERNAME/.bash\_profile

4- /home/USERNAME/.bash\_login

5- /home/USERNAME/.profile

اقدام *bash\_profile*. توجه داشته باشید که تنها یکی از 3، 4 و 5 اجرا خواهد شد. سیستم برای اجرای *bash\_login* را نیابد برای اجرای *bash\_login* خواهد کرد و اگر آن را در مورد 3 نیابد برای *profile* /، اگر هیچ یک از اینها وجود نداشته باشد، سیستم به هیچ وجه به مورد دیگری نگاه *profile* / نخواهد کرد. بنابراین اگر فقط 4 و 5 دارید، فقط 4 اجرا خواهد شد.

در پایان، سیستم بارگذاری می شود:

6. /home/USERNAME/.bashrc

which is users information (like aliases).

را بارگذاری میکند */etc/bash.bashrc* یا */etc/bashrc* فایل */etc/profile* : نکته

### Interactive (non-login) shell

اگر پوسته را در یک حالت تعاملی (غیر ورودی) اجرا کنید، (در حالت رابط کاربری *GUI*)، فقط دو مورد اجرا می شود:

1. /etc/bash.bashrc (یا /etc/bashrc)

2. /home/USERNAME/.bashrc

### adding global configs for login shell



در پایان `sh` با استفاده از `int /etc/profile.d/` ( شما می توانید فایل های پیکربندی جدید خود را به ( اضافه کنید).

است، زیرا در سایر موارد یک بروزرسانی می تواند تغییرات شما `profile / etc /` این تمیزتر و بهتر از ویرایش را تغییر دهد

## adding global configs for interactive/non-login shell

و `aliases` استفاده کنید. این برای `bashrc / etc /` ( برخی از سیستم) `bashrc /etc/bash` شما می توانید فایل دیگر پیکربندیهای عام خوب است

## User specific configs

بیشتر وقتها `PATH` و `env vars` در `bash_profile` / `~` وارد میشوند و نامهای مستعار به `~` `bashrc` / `bashrc` میروند.

## Aliases

تعریف شده اند و شبیه به این هستند `bashrc` / `~` اغلب آنها در

```
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'
```

.این یک نوع میانبر است

## /etc/skel

.این پوشه حاوی فایل هایی است که برای هر کاربر جدید به عنوان یک قالب شروع می شود

## mailing the root user

ایمیل ها را ارسال می کند. با ارسال این دستور می توانید ایمیل را `"mail"` برای ارسال ایمیل، فرمان: یا هر کاربر دیگری ارسال کنید `root` به کاربر

```
i@life:~$ mail root
```

```
Cc:
```

```
Subject: Hi there
```

```
hello. This is my mail
```

کاربر ریشه این ایمیل را دریافت خواهد کرد. او می تواند آن را با استفاده از دستور ایمیل بخواند. اگر شما نیاز به ارسال ایمیل در یک اسکریپت دارید، باید کار را اینگونه انجام دهید

```
$ echo "Body!" | mail -s "Subject" root
```

این به راحتی می تواند ایمیل شما را در یک اسکریپت جاسازی کند

## مدیریت حساب‌های کاربری

### تغییر گذرواژه

دستور `passwd` به منظور تغییر گذرواژه کارایی دارد

```
passwd $
Changing password for i.
(current) UNIX password:
New password:
Retype new password:
passwd: password updated successfully
```

برای تغییر گذرواژه سایر کاربران نام کاربر فوق را بعد از دستور قرار می‌دهیم، در این مورد باید حتما کاربر `root` باشد :

```
passwd i #
New password:
BAD PASSWORD: it does not contain enough DIFFERENT characters
BAD PASSWORD: is too simple
Retype new password:
passwd: password updated successfully
```

اگر پسورد شما بسیار کوتاه یا بسیار مشابه پسورد پیشین شما باشد یا از پسوردهای بسیار آسان که در دیکشنوری کلمات وجود دارد باشد، دستور تغییر پسورد پیغام خطای مناسب را برای شما نمایش داده و خواهان پسورد امن تری می‌شود

### کاربران و گروهها

لینوکس یک سیستم مولتی-یوزر است و شما می‌توانید کاربران را مدیریت کنید، شما می‌توانید کاربران را اضافه، حذف یا ویرایش کنید، در لینوکس مفهومی به نام `groups` وجود دارد که با آن می‌توانید قابلیت‌هایی را به گروهی از کاربران بدهید، به عنوان نمونه می‌توانید گروه `printer` را ایجاد کنید و کاربرانی که باید به دستگاه پرینتر دسترسی داشته باشند را با افزودن به این گروه مدیریت کنید. هر کاربر می‌تواند عضو تعداد زیادی گروه باشد و هر فایل باید متعلق به حداقل یک کاربر و یک گروه باشد. هر کاربر به صورت پیش‌فرض عضو گروه تک کاربره خودش می‌باشد.

### Adduser

برای افزودن کاربر جدید از این دستور استفاده می‌شود، این دستور چند سویچ اصلی دارد که به اختصار در زیر می‌آید، برای شرح کامل آن می‌توانید به صفحه راهنمای دستور مراجعه کنید

#### switch

#### meaning

-d	home directory (-d /home/user)
-m	create home directory
-s	specify shell
-G	add to additional groups
-c	comment. most of the time, users actual name. Use quotes if comments has spaces or special characters in them

در برخی سیستم‌ها، به صورت خودکار با ساخت یک یوزر برایش پوشه `home` ساخته می‌شود ولی در برخی باید حتماً از `m-` استفاده کرد، به همین دلیل بهتر است همیشه از این سویچ استفاده کنیم، وقتی پوشه کاربر جدید ساخته می‌شود، سیستم یک کپی از `etc/skel/` را ایجاد میکند، این پوشه به عنوان یک تمپلت برای ساخت پوشه کاربران جدید استفاده می‌شود.

## اصلاح کاربران

برای این کار باید از سویچ‌های `useradd` استفاده کنیم، به است سه سویچ توجه کنید :

switch	meaning
-L	lock this account
-U	Unlock the account
-aG	add to more groups (say <code>usermod -aG wheel i</code> )

## حذف کاربران

برای اینکار می‌توانید از دستور `userdel` استفاده کنید

```
userdel i
```

اگر از سویچ `-r` استفاده کنید فایل‌های مرتبط نیز پاک می‌شوند

## مدیریت گروهها

برای ساخت یک گروه جدید از دستور `groupadd` استفاده میکنیم

```
groupadd -g 1200 newgroup #
```

این فرمان گروه جدید را با شماره شناسه ۱۲۰۰ ایجاد میکند، اگر نیاز به تغییر بود، کاربر `root` می‌تواند با فرمان زیر این کار را انجام دهد (مثلاً تغییر شناسه گروه به ۲۰۰۰)

```
groupmod -g 2000 newgroup
```

برای حذف گروه نیز میتوان از دستور زیر استفاده کرد

```
groupdel newgroup
```

## فایل‌های مهم

### `etc/passwd/`

.. آن و غیره است `shell` این فایل است که حاوی تمام اطلاعات کاربران از جمله نام کاربر و

```
i@life:~$ tail /etc/passwd  
pulse:x:115:122:PulseAudio daemon,,,:/var/run/pulse:/bin/false
```

```

saned:x:116:124::/var/lib/saned:/bin/false
uidd:x:117:125::/run/uidd:/bin/false
i:x:1000:1000:i,,,:/home/i:/bin/bash
mysql:x:119:128:MySQL Server,,,:/nonexistent:/bin/false
geoclue:x:120:130::/var/lib/geoclue:/bin/false
gnunet:x:121:131::/var/lib/gnunet:/bin/false
sshd:x:122:65534::/var/run/sshd:/usr/sbin/nologin

```

فرمت اطلاعات به شکل زیر است :

```
username:password:userid:primary group id:Name and comments:home dir:shell
```

در گذشته پسورد این فایل بود، اما در حال حاضر به فایل etc/shadow/ ذخیره می‌شود

### etc/shadow/

در این فایل پسوردهای کاربران به صورت هش شده ذخیره می‌شود، این فایل با یوزر root قابل دسترسی است

```
i@life:~$ ls -ltrh /etc/passwd /etc/shadow
```

```
rw-r--r-- 1 root root 2.3K Nov 7 13:19 /etc/passwd-
```

```
rw-r----- 1 root shadow 1.3K Nov 7 13:19 /etc/shadow-
```

در داخل این فایل با یوزر root چنین چیزی خواهید دید

```
# tail /etc/shadow
```

```
:::pulse*:17638:0:99999:7
```

```
:::saned*:17638:0:99999:7
```

```
:::uidd*:17638:0:99999:7
```

```
i:
```

```
$6$DJv.WyWj$64iS4NdvDlp.umXtz6x1C4.QzMOEzMwMdTe.XjSoDQLc67PXZEYW3j8DX7zKClk
```

```
:::YiIBymNaKY4M46LZhC7o8q.:17797:0:99999:7
```

```
:::mysql!:17799:0:99999:7
```

```
:::geoclue*:17799:0:99999:7
```

```
:::gnunet*:17799:0:99999:7
```

```
:::sshd*:17799:0:99999:7
```

! : یعنی پسورد ندارد

برای سایر مقادیر به جدول زیر نگاه کنید :

#### filed

#### meaning

16737 When was the last time this password changes

0 User wont be able to change the password 0 days after each change

99999 After this many days, the user HAVE to change his password

7 ...and the user will be informed 7 days before the expiration to change his password

## یافتن فایل های سیستم

استاندارد سلسله مراتب فایل سیستم (FHS) (FHS) یک سند توصیف سلسله مراتب فایل لینوکس / یونیکس است. این بسیار مفید است که این را بدانید زیرا به شما امکان می دهد به راحتی آنچه را که دنبال آن هستید پیدا کنید:

directory	usage
bin	Essential command binaries
boot	Static files of the boot loader
dev	Device files
etc	Host-specific system configuration
lib	Essential shared libraries and kernel modules
media	Mount point for removable media
mnt	Mount point for mounting a filesystem temporarily
opt	Add-on application software packages
sbin	Essential system binaries
srv	Data for services provided by this system
tmp	Temporary files
usr	Secondary hierarchy
var	Variable data
home	User home directories (optional)
lib	Alternate format essential shared libraries (optional)
root	Home directory for the root user (optional)

## Path

سیستم عامل لینوکس فایل های زیادی دارد، 741341 فایلدر سیستم من، پس در هنگام تایپ یک فرمان فایل آن را از کجا می یابد؟ این کار با یک متغیر PATH انجام می شود

```
$ echo $PATH
```

```
/
```

```
home/i/miniconda3/bin:/home/i/bin:/home/i/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/java-8-oracle/jre/bin
```

## which, type and whereis

فرمان "which" اولین نمایش دستور داده شده در path را نشان می دهد:

```
i@life:~$ which mkfd
```

```
i@life:~$ which mkfs
```

sbin/mkfs/

فرمان type نوع فرمان را نشان میدهد

```
i@life:~$ type mkfs
mkfs is /sbin/mkfs
i@life:~$ type for
for is a shell keyword
i@life:~$ type type
type is a shell builtin
```

بر خلاف which، فرمان whereis محل صفحه راهنما man page و کد منبع و فایل باینری را نشان می دهد.

```
i@life:~$ whereis mkfs
mkfs: /bin/mkfs.btrfs /sbin/mkfs.ext2 /sbin/mkfs.cramfs /sbin/mkfs.fat /sbin/mkfs.vfat
/sbin/mkfs.msdos /sbin/mkfs.ext4 /sbin/mkfs.reiserfs /sbin/mkfs.bfs /sbin/mkfs.reiser4 /sbin/mkfs.jfs
/sbin/mkfs.ext3 /sbin/mkfs.ext4dev /sbin/mkfs.xfs /sbin/mkfs /sbin/mkfs.minix /sbin/mkfs.ntfs
/usr/share/man/man8/mkfs.8.gz
```

## Users and Groups

یک سیستم لینوکس می تواند بسیاری از کاربران و بسیاری از گروه ها داشته باشد. شما می توانید با یک کاربر وارد شوید و از دستور su استفاده کنید تا به گروه دیگری تغییر کند. هر کاربر متعلق به یک گروه اصلی است و می تواند عضو گروه های دیگر نیز باشد. برای تعیین اینکه چه کسی هستید، دستوراتی مانند "id" ، "groups" ، "whoami" وجود دارد.

```
i@life:~$ groups
i adm cdrom sudo dip plugdev lpadmin sambashare
i@life:~$ id
uid=1000(i) gid=1000(i)
groups=1000(i),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),111(lpadmin),126(sambashare)
i@life:~$ whoami
i
```

## مالکیت فایل

فایل ها نیز متعلق به یک کاربر و یک گروه است. امکان استفاده از دستور chmod برای تغییر دسترسی به فایل ها و دایرکتوری ها وجود دارد. از راههای زیر می توانید به این دستور بگویید که چه میخواهید

1. با استفاده از کدهای مبنای هشت
2. با استفاده از کدهای کوتاه

هنگام استفاده از کدهای مبنای هشت، شما باید یک رشته از مبنای هشت ایجاد میکنید تا chmod مفهوم درخواست شما را دریابد. به این ترتیب، 0 به معنای عدم دسترسی (به معنی اجرا)، 2 یعنی نوشتن و 4 به

معنای خواندن است. بنابراین اگر میخواهید «read + exec» را بنزید، باید 4 + 1 را که 5 باشد، نشان دهید. این جدول هر ترکیبی را نشان میدهد:

### Symbolic Octal

rwx	7
rw-	6
r-x	5
r-	4
-wx	3
-w-	2
-x	1
—	0

## سرویس های سیستم

### syslog و rsyslog

اکثر سیستم های مدرن به جای "syslog" از "rsyslog" استفاده می کنند. قابلیت آنها عمدتاً یکسان است و در اینجا فقط "rsyslog" را پوشش خواهیم داد. فایل پیکربندی اصلی در "rsyslog"، همانطور که شما باید قادر به حدس زدن باشید، "etc/syslog.conf" است. /etc/rsyslog.d/ وجود دارد و برای نرم افزارهای مختلف و مدیران بهتر است به جای ویرایش فایل اصلی پیکربندی، تنظیمات خاص خود را به آن اضافه کنید. اگر شما نیاز به شروع یک شنونده syslog و گرفتن پیام های دیگر عملکردهای سیستم دارید، فقط کافی است گزینه -r را به گزینه syslog بی افزایید. فقط / default / rsyslog / etc را ویرایش کنید و گزینه ها را از "" به "-r" تغییر دهید.

```
cat /etc/default/rsyslog
# Options for rsyslogd
# -x disables DNS lookups for remote messages
# See rsyslogd(8) for more details
RSYSLOGD_OPTIONS="-r"
```

و راه اندازی مجدد:

```
systemctl restart rsyslog
```

### journalctl

توزیع های جدیدتر به systemd تغییر می کنند و از systemd journal برای logs سیستم استفاده می کنند. همانطور که قبلاً ذکر شد systemd log خود را به عنوان فایل های باینری نگه می دارد و کاربر باید از journalctl برای دسترسی به آنها استفاده کند.

```
i@life:~$ journalctl
+ Logs begin at Thu 2018-11-08 10:21:57 +0330, end at Fri 2018-11-09 19:08:46 --
Nov 08 10:21:57 life systemd-journal[358]: Runtime journal (/run/log/journal/)
Nov 08 10:21:57 life kernel: Initializing cgroup subsys cpuset
Nov 08 10:21:57 life kernel: Initializing cgroup subsys cpu
Nov 08 10:21:57 life kernel: Initializing cgroup subsys cpuacct
Nov 08 10:21:57 life kernel: Linux version 4.4.0-138-generic (jenkins@devel.tris
Nov 08 10:21:57 life kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-4.4.0-138-ge
:Nov 08 10:21:57 life kernel: KERNEL supported cpus
Nov 08 10:21:57 life kernel: Intel GenuineIntel
Nov 08 10:21:57 life kernel: AMD AuthenticAMD
Nov 08 10:21:57 life kernel: Centaur CentaurHauls
:Nov 08 10:21:57 life kernel: x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]
Nov 08 10:21:57 life kernel: x86/fpu: Supporting XSAVE feature 0x01: 'x87 floati
Nov 08 10:21:57 life kernel: x86/fpu: Supporting XSAVE feature 0x02: 'SSE regist
Nov 08 10:21:57 life kernel: x86/fpu: Supporting XSAVE feature 0x04: 'AVX regist
Nov 08 10:21:57 life kernel: x86/fpu: Enabled xstate features 0x7, context size
:Nov 08 10:21:57 life kernel: e820: BIOS-provided physical RAM map
Nov 08 10:21:57 life kernel: BIOS-e820: [mem 0x0000000000000000-0x0000000000009cf
Nov 08 10:21:57 life kernel: BIOS-e820: [mem 0x0000000000009d00-0x0000000000009ff
Nov 08 10:21:57 life kernel: BIOS-e820: [mem 0x000000000000e000-0x000000000000fff
Nov 08 10:21:57 life kernel: BIOS-e820: [mem 0x0000000000100000-0x00000000afcdcf
Nov 08 10:21:57 life kernel: BIOS-e820: [mem 0x00000000afcd000-0x00000000bce3ef
Nov 08 10:21:57 life kernel: BIOS-e820: [mem 0x00000000bce3f000-0x00000000bcf7ef
lines 1-23
```

فایل config of journalctl در /etc/systemd/journald.conf/ واقع شده است.



## logger

برای تولید برخی از سیاهه ها از دستور *logger* استفاده می شود:

```
logger local1.info i was here
```

## مبانی انتقال ایمیل (MTA)

## aliases

برخی از نامهای پست الکترونیکی در سیستم وجود دارد در / *etc/aliases* تعریف شده اند.

```
cat /etc/aliases
# /etc/aliases
mailer-daemon: postmaster
postmaster: root
nobody: root
hostmaster: root
usenet: root # <--- I'm using this sample
news: root
webmaster: root
www: root
ftp: root
abuse: root
noc: root
security: root
root: i
```

## ارسال نامه

امکان ارسال یک ایمیل از خط فرمان با استفاده از دستور *mail*:

```
mail news $[~ i@life]
Subject: Email to news user
hahah.. we know where this will go.
this will go to root and then to Kaveh
```

Hi Kaveh!

Cc:

```
[i@life ~]$ mail
```

```
Mail version 8.1.2 01/15/2001. Type ? for help.
```

```
"/var/mail/i": 12 messages 12 new
```

```
>N 1 root@life Sat Jan 02 08:50 39/1373 apt-listchanges: news for f
N 2 root@life Sat Jan 02 09:01 165/7438 apt-listchanges: news for f
N 3 i@life Sat Jan 02 19:58 18/640 *** SECURITY information fo
N 4 i@life Sat Jan 02 20:04 18/631 *** SECURITY information fo
N 5 i@life Sun Jan 03 10:15 18/664 *** SECURITY information fo
N 6 root@life Mon Jan 04 12:42 27/941 Cron <i@life> /home/j
N 7 root@life Mon Jan 04 17:11 26/845 apt-listchanges: news for f
N 8 root@life Tue Jan 05 18:42 27/945 Cron <i@life> /home/j
N 9 root@life Wed Jan 06 09:17 46/1788 apt-listchanges: news for f
N 10 root@life Thu Jan 07 12:42 27/945 Cron <i@life> /home/j
N 11 root@life Thu Jan 07 18:42 27/943 Cron <i@life> /home/j
N 12 i@life Thu Jan 7 19:53 17/478 Email to news user
```

```
& 12
```

Message 12:  
From i@life Thu Jan 7 19:53:08 2016  
X-Original-To: news  
To: news@life  
Subject: Email to news user  
Date: Thu, 7 Jan 2016 19:53:08 +0330 (IRST)  
From: i@life (i)

hahah.. we know where this will go.  
this will go to root and then to Kaveh!

Hi Kaveh!

& d

& q

Held 11 messages in /var/mail/i

## مقدمات شبکه

### down و up و ifconfig

ifconfig دستور اصلی برای پیکربندی آداپتورهای شبکه به صورت دستی است. در حال اجرا بدون استدلال، تمام آداپتورهای شبکه و تنظیمات آنها را نشان می دهد.

```
i@life:~$ ifconfig
  enp0s20u2 Link encap:Ethernet HWaddr xx:xx:xx:xx:xx:xx
  inet addr:192.168.42.137 Bcast:192.168.42.255 Mask:255.255.255.0
  inet6 addr: fe80::4485:54c8:f0ee:6115/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:578296 errors:17 dropped:0 overruns:0 frame:17
  TX packets:529081 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:606024454 (606.0 MB) TX bytes:68875310 (68.8 MB)
```

```
  enp4s0 Link encap:Ethernet HWaddr 28:d2:44:d8:9b:82
  UP BROADCAST MULTICAST MTU:1500 Metric:1
  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

```
  enp4s0:avahi Link encap:Ethernet HWaddr xx:xx:xx:xx:xx:xx
  inet addr:169.254.9.217 Bcast:169.254.255.255 Mask:255.255.0.0
  UP BROADCAST MULTICAST MTU:1500 Metric:1
```

```
  lo Link encap:Local Loopback
  inet addr:127.0.0.1 Mask:255.0.0.0
  inet6 addr: ::1/128 Scope:Host
  UP LOOPBACK RUNNING MTU:65536 Metric:1
  RX packets:134767 errors:0 dropped:0 overruns:0 frame:0
  TX packets:134767 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1
  RX bytes:59608995 (59.6 MB) TX bytes:59608995 (59.6 MB)
```

```
  wlp5s0 Link encap:Ethernet HWaddr xx:xx:xx:xx:xx:xx
  UP BROADCAST MULTICAST MTU:1500 Metric:1
  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

## DNS config file

همانطور که مشاهده کردید، ما توانستیم پیکربندی DNS را در فایل‌های پیکربندی رابط شبکه تنظیم کنیم. اما این تنها راه نیست. فایل دیگری است که حاوی این اطلاعات است: `/etc/resolv.conf`.

```
i@life:~$ cat /etc/resolv.conf
Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8) #
DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN #
nameserver 127.0.1.1
```

### hostname

یک فایل متنی دیگر وجود دارد که نام میزبان را نشان می‌دهد یا تنظیم میکند `/etc/hostname`.

```
i@life:~$ cat /etc/hostname
life
```

### hosts

فایل `/etc/hosts` شامل نام سرور و IP های آن است. این درست مثل آنچه که DNS انجام می‌دهد اما اولویت بالاتری نسبت به DNS دارد.

```
i@life:~$ cat /etc/hosts
localhost 127.0.0.1
life.life.lo 127.0.1.1
oracle 127.0.0.2
```

The following lines are desirable for IPv6 capable hosts #

```
ip6-localhost ip6-loopback 1 ::
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
.ff02::2 ip6-allrouters
```

### route

دستور `route` می‌تواند سیستم مسیریابی را نشان دهد یا تغییر دهد. همانطور که در بخش پیش فرض `gateway` دیده‌اید، مسیریابی مسئول ارسال بسته‌های خود به مقصد صحیح است.

```
i@life:~$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default 192.168.42.129 0.0.0.0 UG 100 0 0 enp0s20
u2
default * 0.0.0.0 U 1002 0 0 enp4s0
link-local * 255.255.0.0 U 0 0 0 enp4s0
U 100 0 0 enp0s20 255.255.255.0 * 192.168.42.0
u2
```

و برای به طور موقت اضافه کردن یک مسیر پیش فرض، شما می‌توانید انجام دهید:

```
route add default gw 192.168.1.1
```

## ip

دستور ip ابزار جدیدی برای پیکربندی واسط شبکه است. شما می توانید چیزهای زیادی را با استفاده از آن انجام دهید. `addr show` به شما رابط های فعلی و تنظیمات آنها را نشان می دهد:

```
i@life:~$ ip addr show
lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen :1
1
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
enp4s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN :2
group default qlen 1000
link/ether xx:d2:44:xx:9b:82 brd ff:ff:ff:ff:ff:ff
inet 169.254.9.217/16 brd 169.254.255.255 scope link enp4s0:avahi
valid_lft forever preferred_lft forever
wlp5s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group :3
default qlen 1000
link/ether xx:0e:c4:xx:72:xx brd ff:ff:ff:ff:ff:ff
enp0s20u2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state :4
UNKNOWN group default qlen 1000
link/ether xx:xx:ea:xx:1e:55 brd ff:ff:ff:ff:ff:ff
inet 192.168.42.137/24 brd 192.168.42.255 scope global dynamic enp0s20u2
valid_lft 3415sec preferred_lft 3415sec
inet6 fe80::4485:54c8:f0ee:6115/64 scope link
valid_lft forever preferred_lft forever
```

## ping

پینگ بهترین دستورالعمل عیب یابی شبکه است. شما می توانید اتصال خود را با هر سرور با استفاده از آن بررسی کنید. می بینیم که آیا کامپیوتر من 1.1.1.1 را می بیند:

```
i@life:~$ ping 1.1.1.1
.PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data
bytes from 1.1.1.1: icmp_seq=1 ttl=51 time=226 ms 64
bytes from 1.1.1.1: icmp_seq=2 ttl=51 time=114 ms 64
bytes from 1.1.1.1: icmp_seq=3 ttl=51 time=120 ms 64
bytes from 1.1.1.1: icmp_seq=4 ttl=51 time=121 ms 64
bytes from 1.1.1.1: icmp_seq=5 ttl=51 time=119 ms 64
bytes from 1.1.1.1: icmp_seq=6 ttl=51 time=110 ms 64
C ^
--- ping statistics 1.1.1.1 ---
packets transmitted, 6 received, 0% packet loss, time 5006ms 6
```

rtt min/avg/max/mdev = 110.951/135.497/226.894/41.038 ms

## nsswitch

فایل `etc/nsswitch.conf` استفاده می شود تا پیکربندی کنید که کدام سرویس ها برای تعیین اطلاعات مانند نام های میزبان، فایل های رمز عبور و فایل های گروه مورد استفاده قرار گیرد.

```
i@life:~$ sudo cat /etc/nsswitch.conf
:password for i [sudo]
etc/nsswitch.conf/ #
#
.Example configuration of GNU Name Service Switch functionality #
.If you have the `glibc-doc-reference' and `info' packages installed, try #
.info libc "Name Service Switch" for information about this file ` #
```

```
passwd:    compat
group:     compat
shadow:    compat
gshadow:   files
```

```
hosts:     files mdns4_minimal [NOTFOUND=return] dns
networks:  files
```

```
protocols: db files
services:  db files
ethers:    db files
rpc:       db files
```

```
netgroup:  nis
```

## netstat

این فرمان می تواند اطلاعات مختلفی در مورد شبکه ما نشان دهد. دو ویژگی اصلی نشان دادن جدول مسیریابی و بررسی پورت های گوش دادن است. با استفاده از آن می توانید جدول مسیریابی خود را بررسی کنید.

```
i@life:~$ netstat -nr
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
UG 0 0 0 enp0s20 0.0.0.0 192.168.42.129 0.0.0.0
u2
U 0 0 0 enp4s0 0.0.0.0 0.0.0.0 0.0.0.0
U 0 0 0 enp4s0 255.255.0.0 0.0.0.0 169.254.0.0
U 0 0 0 enp0s20 255.255.255.0 0.0.0.0 192.168.42.0
u2
```

همچنین می توانید ببینید که پورت ها در حال گوش دادن به سرور در سرور ما هستند:

```
netstat -na | grep 80 $[~ i@life]
tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN
```

یافتن پورتهای باز

برای یافتن پورتهای باز میتوان از گزینه - tuna بهره برد

```
i@life:~$ netstat -tuna
```

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	127.0.0.1:3306	0.0.0.0:*	LISTEN
tcp	0	0	127.0.1.1:53	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:9050	0.0.0.0:*	LISTEN
tcp	0	0	192.168.42.207:49152	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:2086	0.0.0.0:*	LISTEN
tcp	0	0	192.168.42.207:45858	185.199.109.153:443	ESTABLISHED
tcp	0	0	192.168.42.207:58795	172.217.22.14:443	ESTABLISHED
tcp	0	0	192.168.42.207:30810	91.250.84.156:9001	ESTABLISHED
tcp6	0	0	:::22	:::*	LISTEN
tcp6	0	0	:::1:631	:::*	LISTEN
tcp6	0	0	:::49152	:::*	LISTEN
tcp6	0	0	:::2086	:::*	LISTEN
*:udp	0	0	0.0.0.0:55685	0.0.0.0	
udp	0	0	192.168.42.207:13034	192.168.42.129:5351	ESTABLISHED
*:udp	0	0	0.0.0.0:19543	0.0.0.0	
*:udp	0	0	0.0.0.0:5310	0.0.0.0	
*:udp	0	0	0.0.0.0:5353	0.0.0.0	
*:udp	0	0	0.0.0.0:23836	0.0.0.0	
*:udp	0	0	127.0.0.1:56634	0.0.0.0	
*:udp	0	0	0.0.0.0:1900	0.0.0.0	
*:udp	0	0	0.0.0.0:24491	0.0.0.0	
*:udp	0	0	0.0.0.0:2086	0.0.0.0	
*:udp	0	0	127.0.1.1:53	0.0.0.0	
*:udp	0	0	0.0.0.0:68	0.0.0.0	
*:udp	0	0	0.0.0.0:68	0.0.0.0	
*:*			udp6	0	0 :::5310
*:*			udp6	0	0 :::5353
*:*			udp6	0	0 :::42723
*:*			udp6	0	0 :::34711
*:*			udp6	0	0 :::2086

برای این کار میتوان از `lsof -i` و `fuser` همچنین نیز استفاده کرد، البته این دستور نیاز به دسترسی `root` دارد

## traceroute

این ابزار عیب یابی پیشرفته تر است. این مانند پیینگ کردن تمام سرورها بین شما و مقصد شما یک به یک است و ببینید کدام بسته های ما اشتباه هستند. اجازه می دهیم که به google.com بروم.

```
i@life:~$ traceroute google.com
```

```
traceroute to google.com (172.217.22.14), 30 hops max, 60 byte packets
```

```
1 192.168.42.129 (192.168.42.129) 0.381 ms 0.406 ms 0.446 ms
2 * * *
3 * * *
4 10.222.97.69 (10.222.97.69) 49.510 ms 10.222.97.85 (10.222.97.85) 50.519 ms 10.222.97.82
(10.222.97.82) 52.385 ms
5 * * *
6 * * *
7 * * *
8 10.221.57.142 (10.221.57.142) 43.026 ms 18.815 ms 40.956 ms
9 10.221.59.130 (10.221.59.130) 43.022 ms 40.767 ms 39.879 ms
10 10.21.252.150 (10.21.252.150) 41.919 ms 38.947 ms 10.21.251.26 (10.21.251.26) 41.732 ms
11 10.21.21.10 (10.21.21.10) 40.901 ms 35.862 ms 54.895 ms
12 10.21.211.10 (10.21.211.10) 29.848 ms 34.869 ms 39.807 ms
13 de-cix.fra.google.com (80.81.193.108) 126.986 ms 103.000 ms 103.780 ms
14 108.170.251.193 (108.170.251.193) 118.830 ms 118.808 ms 118.980 ms
15 108.170.235.253 (108.170.235.253) 129.895 ms 119.935 ms 108.170.235.255 (108.170.235.255)
112.854 ms
16 fra16s14-in-f14.1e100.net (172.217.22.14) 128.028 ms 117.870 ms 112.936 ms
```

## dig

دستور dig یک ابزار جستجوی DNS است. اگر با یک نام دامنه مشکل دارید، می توانید ببینید که چگونه به IP هامتصل می شود و توسط چه کسی

```
i@life:~$ dig google.com
```

```
DiG 9.10.3-P4-Ubuntu <<>> google.com <<>> ;
global options: +cmd ;;
:Got answer ;;
HEADER<<- opcode: QUERY, status: NOERROR, id: 21478 <<- ;;
flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0 ;;

:QUESTION SECTION ;;
A      IN                google.com. ;

:ANSWER SECTION ;;
172.217.22.14      A      IN      96      google.com.
```



```
Query time: 0 msec ;;
SERVER: 127.0.1.1#53(127.0.1.1) ;;
WHEN: Fri Nov 09 20:28:43 +0330 2018 ;;
MSG SIZE rcvd: 44 ;;
```

## netcat

ابزار nc (یا netcat) برای تقریبا هر چیزی شامل TCP، UDP یا سوکت های دامنه یونیکس استفاده می شود. این ابزار می تواند اتصالات TCP را باز کند، بسته های UDP را ارسال کند، در پورتهای دلخواه TCP و UDP گوش دهد، پورت اسکن کند، و با IPv4 و IPv6 مقابله کند. بر خلاف telnet، اسکریپت nc به سادگی، و پیام های خطا را بر روی خطای استاندارد جدا می کند به جای ارسال آنها به خروجی استاندارد، به عنوان telnet می کند با برخی از. این یک فرمان بسیار توانمند است

## nmap

nmap جعبه ابزار هکرها است! شما می توانید با این ابزار اطلاعات زیادی در مورد یک سرور کسب کنید و برای تست امنیت سرورتان بسیار برایتان کارایی خواهد داشت:

```
i@life:~$ sudo nmap localhost
```

```
Starting Nmap 7.01 ( https://nmap.org ) at 2018-11-09 20:40 +0330
```

```
Nmap scan report for localhost (127.0.0.1)
```

```
.Host is up (0.0000040s latency)
```

```
Not shown: 996 closed ports
```

```
PORT      STATE SERVICE
```

```
tcp open  ssh/22
```

```
tcp open  ipp/631
```

```
tcp open  mysql/3306
```

```
tcp open  tor-socks/9050
```

```
Nmap done: 1 IP address (1 host up) scanned in 1.63 seconds
```

## sudo vs su

ما از sudo و su در تمام فصل ها استفاده کرده ایم و این زمان برای نگاه دقیق تر به آنها است! SU تغییر حساب شما را به نام کاربری دیگر است

```
i@life ~$ whoami
```

```
i
```

```
i@life ~$ su -
```

```
Password:
```

```
root@life:~# whoami
```

```
root
```

```
root@life:~# su i -
```

```
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
```

```
bash: no job control in this shell
```

```
i@life /root$ whoami
```

```
i
```

```
i@life /root$ exit
```

```
exit
```

```
root@life:~# whoami
```

```
root
```

```

root@life:~# exit
logout
i@life ~$ whoami
i
i@life ~$

```

توجه داشته باشید که هنگام اجرای SU باید رمز عبور ریشه را برای تبدیل شدن به ریشه فراهم کنید؛ یا هر کاربر دیگری رمز عبور برای تبدیل شدن به آن کاربر!

## user limits

منابع موجود در یک دستگاه لینوکس می توانند توسط فرمان ulimit برای کاربران مدیریت شوند. این بخشی از سیستم PAM است. اگر می خواهید محدودیت های سیستم را کنترل کنید:

```

i@life:~$ ulimit -a
core file size          (blocks, -c) 0
data seg size          (kbytes, -d) unlimited
scheduling priority    (-e) 0
file size              (blocks, -f) unlimited
pending signals        (-i) 14546
max locked memory      (kbytes, -l) 64
max memory size        (kbytes, -m) unlimited
open files             (-n) 1024
pipe size              (512 bytes, -p) 8
POSIX message queues   (bytes, -q) 819200
real-time priority     (-r) 0
stack size             (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited
max user processes    (-u) 14546
virtual memory         (kbytes, -v) unlimited
file locks            (-x) unlimited

```

و شما می توانید آنها را مانند این تغییر دهید:

```
ulimit -t ۱
```

این کار CPU TIME هر فرایند را به ۱ ثانیه محدود می کند. اگر بیشتر از آن استفاده کنید، این روند به طور خودکار (توسط ماژول PAM) کشته خواهد شد. لطفاً توجه داشته باشید که زمان ساعت متفاوت است از زمان CPU. برای دیدن اینکه چقدر پردازنده CPU زمان استفاده از آن را اجرا می کند، مانند این است:

```
time firefox
```

## بررسی کاربران در سیستم

اگر شما نیاز به بررسی اینکه چه کسی در سیستم شما باشد (و به برخی از آنچه انجام می دهید) می توانید از این دستورات استفاده کنید:

```

i@life:~$ w
up 33 min, 3 users, load average: 0.08, 0.20, 0.29 20:47:54
USER  TTY  FROM          LOGIN@  IDLE  JCPU  PCPU  WHAT
-i   tty7  :0            20:15  32:42 39.69s 0.06s /sbin/upstart
i    pts/3  life         20:16  31:45 0.02s 0.02s /bin/bash

```

```
i pts/0 :0 20:17 0.00s 0.06s 0.00s w
```

```
i@life:~$ who
```

```
i tty7 2018-11-09 20:15 (:0)
```

```
i pts/3 2018-11-09 20:16 (life)
```

```
i pts/0 2018-11-09 20:17 (:0)
```

همانطور که می بینید هر دو این دستورات به شما می گویند زمانیکه کاربر وارد سیستم شد، اما افراد خروج را نشان نمی دهد (زیرا دیگر در سیستم نیستند). اگر شما نیاز به این داده ها از دستور last استفاده کنید:

```
i@life:~$ last|head
```

```
i pts/0 :0 Fri Nov 9 20:17 gone - no logout
```

```
i pts/3 life Fri Nov 9 20:16 gone - no logout
```

```
i tty7 :0 Fri Nov 9 20:15 gone - no logout
```

```
reboot system boot 4.4.0-138-generi Fri Nov 9 20:14 still running
```

```
i pts/2 :0 Fri Nov 9 15:09 - 20:13 (05:03)
```

```
i pts/6 :0.0 Thu Nov 8 18:35 - 20:33 (01:57)
```

```
i pts/5 :0.0 Thu Nov 8 18:35 - 18:35 (00:00)
```

```
i pts/5 :0.0 Thu Nov 8 16:49 - 16:49 (00:00)
```

```
i pts/2 :0.0 Thu Nov 8 14:33 - 18:36 (04:02)
```

```
i pts/2 :0.0 Thu Nov 8 12:32 - 12:47 (00:15)
```

- <sup>i</sup> <https://www.debian.org/intro/free.fa.html>
- <sup>ii</sup> <http://www.gnu.org/philosophy/free-sw.html>
- <sup>iii</sup> <http://archive.is/9aNu>
- <sup>iv</sup> <https://www.computerworld.com/article/2521639/computer-hardware/ibm-s-newest-mainframe-is-all-linux.html>
- <sup>v</sup> [http://www.trl.ibm.com/projects/ngm/wp10\\_e.htm](http://www.trl.ibm.com/projects/ngm/wp10_e.htm)
- <sup>vi</sup> [https://en.wikipedia.org/wiki/Linux\\_distribution](https://en.wikipedia.org/wiki/Linux_distribution)
- <sup>vii</sup> *distrowatch.com*
- <sup>viii</sup> صفحات ۳۵-۳۷ Getting Started with Ubuntu ترجمه و تلخیص از کتاب ۱۴۰۴
- <sup>ix</sup> LPIC1 exam guide in plain English | Amir 'Jadi' Mirmirani
- <sup>x</sup> [سامانه\\_مدیریت\\_بسته](https://fa.wikipedia.org/wiki/سامانه_مدیریت_بسته)
- <sup>xi</sup> <https://www.debian.org/doc/manuals/debian-reference/ch02.en.html>
- <sup>xii</sup> [http://www.opengroup.org/austin/papers/posix\\_faq.html](http://www.opengroup.org/austin/papers/posix_faq.html)
- <sup>xiii</sup> <https://www.gnu.org/software/bash/>
- <sup>xiv</sup> [https://en.wikipedia.org/wiki/Unix\\_shell](https://en.wikipedia.org/wiki/Unix_shell)
- <sup>xv</sup> [unixhelp.ed.ac.uk/CGI/man-cgi?echo](http://unixhelp.ed.ac.uk/CGI/man-cgi?echo)
- <sup>xvi</sup> <http://pubs.opengroup.org/onlinepubs/9699919799/utilities/env.html>
- <sup>xvii</sup> <http://cm.bell-labs.com/7thEdMan/v7vol1.pdf>
- <sup>xviii</sup> <http://www.freebsd.org/cgi/man.cgi?cat>