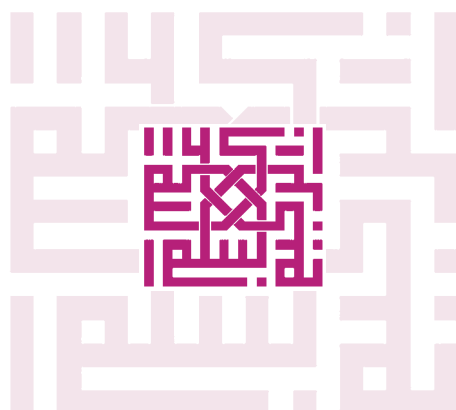


لینوکس مقدماتی



Linux

صوفیاء علی خانی



لینوکس



مقدّمات

نوشته: صوفیا علیخانی

سرشناسه	: علیخانی، صوفیا، ۱۳۴۷ -
عنوان و نام پدیدآور	: لینوکس مقدماتی/نوشته صوفیا علیخانی.
مشخصات نشر	: تهران: نشر تعالی، ۱۳۹۵.
مشخصات ظاهری	: ۱۴۰ ص.
شابک	: ۷-۱-۸۶۸۰-۰۰۰-۹۷۸-۶۰۰: ۱۸۰۰۰۰ ریال
وضعیت فهرست‌نویسی	: فیپا
موضوع	: سیستم عامل لینوکس
موضوع	: Linux
موضوع	: سیستم‌های عامل (کامپیوتر)
موضوع	: Operating systems (Computers)
رده‌بندی کنگره	: ۱۳۹۵ ۹۴ع۸ / ۷۶/۷۶ QA
رده‌بندی دیوئی	: ۰۰۵/۴۳۲
شماره کتاب‌شناسی ملی	: ۴۵۶۳۵۲۷



● لینوکس مقدماتی ●

- صوفیا علیخانی ●
- صفحه‌آرایی: جعفر خدادی ●
- ناشر: تعالی ●
- نوبت چاپ: اول-۱۳۹۵ ●
- شمارگان: ۱۰۰۰ نسخه ●
- قیمت: ۱۸۰۰۰ تومان ●
- شابک: ۷-۱-۸۶۸۰-۰۰۰-۹۷۸-۶۰۰ ●

- نشانی مرکز پخش: تهران، میدان انقلاب، خیابان جمالزاده جنوبی ●
- کوچه رشتچی، پلاک ۱۵، واحد ۵ ●
- تلفن: ۰۹۱۲۲۰۲۷۴۲۵-۰۶۶۹۱۲۰۴۵-۰۶۶۹۱۲۰۴۴ ●
- نشانی پست الکترونیک: taalibook@yahoo.com ●
- نشانی اینترنتی: www.taalibook.com ●

Course Title	Linux Essentials
Date	10/24/2016
Location	Tehran/Iran
Instructor Name	Sophia Alikhani
Instructor Email	Sophia.alikhani47@gmail.com admin@linuxmahz.com https://www.linuxmahz.com

بازخورد شما نسبت به مطالب ارائه شده
باعث ارائه بهتر و بیشتر مطالب در وبسایت خواهد شد.
پس منتظر ارائه نظرات خوب شما عزیزان هستم.

با سپاس
صوفیا علیخانی

مقدمه / ۹

انقلاب لینوکس / ۱۱

آشنایی با برنامه‌های کاربردی متن آزاد / ۱۹

مهارت ICT و کار با لینوکس / ۲۵

دستورات اولیه لینوکس / ۳۵

Help linux / ۴۹

مدیریت فایل‌ها و دایرکتوری‌ها در لینوکس / ۵۹

ایجاد حذف و جابه‌جایی فایل‌ها در لینوکس / ۶۵

نحوه پشتیبان‌گیری از داده‌ها در لینوکس / ۷۵

نحوه جستجوی متن در فایل‌ها / ۸۷

داده‌های ما کجا ذخیره می‌شوند؟ / ۹۹

انواع کاربران سیستم عامل لینوکس و مفاهیمی از

امنیت اولیه در ارتباط با کاربران / ۱۰۷

ایجاد کاربران و گروه‌های کاربری / ۱۱۵

نحوه اجازه دسترسی به کاربران / ۱۲۳



مقدمه

تمام افرادی که می‌خواهند با سیستم عامل لینوکس آشنا شوند دوره‌هایی را به صورت بین‌المللی طی می‌کنند که یکی از این دوره‌ها با نام دوره LPIC-1 می‌باشد. در طی تجارب تدریس متوجه شده‌ام که بسیاری از کاربرانی که خیلی با سیستم عامل لینوکس آشنایی ندارند هنگام گذراندن دوره LPIC-1 دچار سختی می‌شوند. لذا به نظر رسید که دوره لینوکس مقدماتی را به چاپ برسانم تا برای افرادی که چیزی از این سیستم عامل نمی‌دانند، دنبال کردن و یادگیری مطالب آن راحت‌تر بوده و درواقع این دوره مثل پلی باشد که راه رسیدن به دوره LPIC-1 را هموارتر سازد.

بنابراین اگر هیچ چیزی از لینوکس نمی‌دانید نگران نباشید و برای یادگیری آن با من همراه شوید.

کلیدهای موفقیت:

تنها راه موفقیت و یادگیری لینوکس، تمرین مداوم می‌باشد.

انقلاب لینوکس و شناسایی سیستم های عامل مطرح



LinuxTM



آنچه در این بخش می‌خواهم عنوان کنم این است که اصلاً لینوکس چیست و مفهوم open source که به فارسی به آن متن‌باز می‌گوییم یعنی چی؟ ابتدا به تعریف فلسفه open source می‌پردازم.

اما مفهوم آزادی در این سیستم‌ها چیست؟

مفهوم از آزادی را می‌توان به اشکال زیر تعریف نمود:

۱- آزادی در داشتن کل متن برنامه نوشته‌شده.

۲- آزادی در تعداد کپی‌هایی که می‌توان از برنامه داشت.

۳- آزادی در داشتن متن برنامه و تغییر در آن و اجرای آن به شکل دلخواه و در اختیار قرار دادن این نسخه جدید به دیگران.

۴- آزادی در قرار دادن متن برنامه برای دیگران، اما گرفتن پول برای پشتیبانی و یا اعمال تغییرات در آن.

در حالت‌های بالا مشاهده می‌کنیم که تعاریف مختلف و بنابراین مفاهیم مختلفی از دامنه‌های تعریف‌شده برای مفهوم آزادی در برنامه‌ها داریم.

گاهی اوقات آزادی در متن برنامه به معنی پول گرفتن و در اختیار قرار دادن کل آن به خریدار است و گاهی بدون گرفتن پول می‌باشد، مانند نسخه RedHat و CentOS برای روشن شدن مطلب یک مثال می‌زنم.

مثلاً شما به یک شیرینی فروشی می‌روید و شیرینی فروشی می‌گوید شما آزادید هر مقدار باقلوا می‌خواهید همین‌جا میل کنید.



این یک مفهوم از آزادی است و شما پولی بابت این مقدار پرداخت نمی کنید. حالا فرض کنید به شیرینی فروشی می روید و شیرینی فروشی می گوید من دستور پخت این باقلوا را بدون گرفتن هزینه به شما می دهم.

در این حالت شما می توانید این دستور پخت را گرفته و بر اساس ذائقه خود آن را تغییر داده کم شیرین یا شیرین تر کرده یا گردو اضافه کنید و یا هر سلیقه ای که دارید روی آن پیاده کرده و محصول جدید خود را به دیگران، یا بفروشید یا آزادانه در اختیارشان قرار دهید.

این ها مثال هایی از آزادی است.

در سیستم عامل و برنامه هایی از این دست دقیقاً این مفهوم به کار می رود. سیستم های متن بازی داریم که در ازای گرفتن پول، کل دستور پخت و ساختار سیستم عامل یا برنامه خاص در اختیارتان قرار می گیرد و یا بدون دریافت پول این نسخه به شما داده می شود. در حالت اول شما پولی پرداخت می کنید و دستور پخت را می گیرید بعد به سلیقه خود آن را تغییر می دهید.

در حالت دوم شما بدون پرداخت پول این دستورالعمل را تهیه کرده و تغییر می دهید. در سیستم عامل لینوکس و برنامه های تحت آن دقیقاً این مفاهیم آزادی مصداق پیدا می کند یعنی مواردی از برنامه ها هستند که در ازای پول نسخه اصلی آن و یا نحوه ساخت برنامه ها، با تمام سورس برنامه در اختیارتان قرار می گیرد و گاهی بدون دریافت پول و کاملاً آزاد تمام مراحل ساخت آن در اختیار شما قرار می گیرد.

هردوی این موارد نشانه آزادی هستند و هر کدام مزایا و معایب خود را دارا می باشند. مثلاً کمپانی RedHat کلیه سورس برنامه ها و ساخت سیستم عامل را در ازای دریافت پول به شما می فروشد و شما قادرید آن سورس را بر اساس نیاز خود تغییر داده و نسخه جدیدی ارائه دهید و بر اساس میل خود این نسخه را بفروشید یا آزادانه و بدون دریافت پول در اختیار دیگران قرار دهید.



یا مثلاً نسخه CentOS همان سورس نسخه RedHat می‌باشد که از آنجا خریداری شده ولی بعداً از طریق این کمپانی به شکل کاملاً آزاد در اختیار دیگران قرار می‌گیرد؛ یعنی این کمپانی برای در اختیار قرار دادن نسخه خود هیچ پولی از شما دریافت نمی‌کند و کاملاً آزادانه آن را در اختیار دیگران قرار می‌دهد.
به عنوان مثال به برنامه مقابل توجه کنید که کد آن را نوشته و اجرا می‌کنم:

```
#!/bin/bash
echo "what is your favorite color?"
echo
read COLOR
echo "what is your favorite number?"
echo
read NUMBER

echo "Here is my gift to you "
#for i in $(eval echo "{1..$COLOR}")
for i in $(eval echo "{1..$NUMBER}")
do
    echo "$i $COLOR"
done
```

زمانی که این برنامه را اجرا می‌کنیم با خروجی زیر مواجه می‌شویم:

```
[oracle@master linuxessential]$ ./1.sh
```

```
what is your favorite color?
```

```
RED
```

```
what is your favorite number?
```

```
3
```

```
Here is my gift to you
```




1 RED

2 RED

3 RED

کد نوشته شده برنامه بالا را می توان در اختیار دیگران قرار داده تا دیگران بر اساس سلیقه خود آن را تغییر دهند. این همان مفهوم آزادی در متن برنامه می باشد؛ یعنی ما متن و یا سورس برنامه هایی را که نوشتیم، در اختیار همگان قرار داده و دیگران می توانند بر اساس نیاز خود آن را دست کاری کرده و به شکل دلخواه خود نوشته و اجرا نمایند. دقیقاً مثل اینکه من دستور پخت باقلوا را در اختیار شما قرار دهم، اینجا دستور ساخت این فیل و خروجی آن را در اختیار شما قرار داده ام و این همان مفهوم سیستم متن باز یا آزادی در متن برنامه می باشد.

حالا اجازه دهید که در مورد انواع مختلف نسخه های لینوکس و آزادی آن ها صحبت کنیم

مثلاً در نسخه Debian هسته اصلی سیستم عامل یا همان دستور پخت اولیه سیستم عامل گرفته شده و تغییراتی بر اساس دلخواه در آن اعمال شده و مثلاً نحوه اجرای برنامه ها یا به اصطلاح پکیج ها و یا نحوه به روز رسانی ها متفاوت از نسخه RedHat می باشد.

یا مثلاً نسخه دیگر سیستم عامل لینوکس مثل SUSE از محیط گرافیک متفاوت و یا ابزارهای متفاوتی از نسخه Debian استفاده می کند و یا مثلاً Ubuntu نسخه دیگری از لینوکس که دارای محیط گرافیکی متفاوت و دستورات اجرایی متفاوت بوده و شبیه Debian می باشد ولی ممکن است در بعضی موارد فرق کنند، مثل اینکه شما به یک باقلوا گردو اضافه کنید و به دیگری بادام.

در CentOS , RedHat دقیقاً دستورالعمل ها مانند هم بوده و مثل این است که هر دو از دستورالعمل یکسان پخت باقلوا پیروی می کنند.



همچنین الحاق شدن لینوکس را نیز می‌توانید در موارد زیر نیز مشاهده نمایید:

مثلاً گوشی‌های موبایل که سیستم‌عامل آن‌ها اندروید می‌باشد یا Roku که وسیله‌ای است برای دانلود و تماشای فیلم از روی اینترنت یا Kindle که ابزاری است برای دانلود کتاب از آمازون و یا chrome book.

در همگی این موارد از سیستم‌عامل لینوکس استفاده شده است و ابزارهای متفاوتی با استفاده از آن ساخته شده و در اختیار دیگران قرار گرفته است.

پس می‌بینیم که لینوکس همه‌جا مشاهده می‌شود و زمانی که ما از لینوکس استفاده می‌کنیم می‌دانیم که سورس آن در اختیار ما قرار دارد تا بر اساس نیاز آن را تغییر دهیم. یعنی به شکل باورنکردنی می‌توانیم به هر شکلی که می‌خواهیم آن را تغییر داده و برنامه‌ای جدید از آن تولید نماییم.

آشنایی با برنامه‌های متن آزاد



LinuxTM



دسته‌بندی‌هایی که می‌توان برای این نوع برنامه‌ها در نظر گرفت می‌تواند به‌صورت

زیر باشد:

1-3- Desktop

2-3- Server

3-3- Admin

4-3- programming

5-3- mobile

در محیط دستکتاپ می‌توان برنامه‌های زیر را به‌عنوان برنامه‌های متن آزاد نام برد:

openoffice

liberooffice

firefox

Thunderbird

chromium

- در محیط سرور می‌توان برنامه‌های زیر را به‌عنوان برنامه‌های متن آزاد نام برد:

- Apache

- Samba

- mysql

- oracle



- NFS
- LDAP
- postfix

- در محیط ادمین ابزارهای زیر از نوع متن باز هستند:

- apt-get
- yum
- webmin
- GUI Tools

در محیط برنامه‌نویسی:

- C
- ++C
- java
- Python
- PHP
- perl
- Ruby
- shell

- در محیط موبایل:

- ssh client
- mobile browser
- Dolphin (desktop mode)



تعریف سیستم متن آزاد و یا open source و مفهوم license برای سیستم‌های متن باز

در دنیای سیستم‌های متن آزاد دو نوع سازمان هستند که لیسانس لازم برای دریافت مدرک متن آزاد بودن را ارائه می‌کنند:

1- FSF Free Software Foundation

2- OSI Open Source Initiative

هیچ کدام از این دو کمپانی بر دیگری ارجحیت ندارند، بلکه هر کدام سیاست مخصوص خود در ارائه مدرک را بر عهده دارند و هر کدام هدفی خاص را در این راستا دنبال می‌کنند.

FSF از استانداردهای GPL تبعیت می‌کند که در این استاندارد آزادی مطلق بر هر نرم‌افزاری وجود دارد. آزادی برای هر چیزی. هیچ حق تقدیمی وجود ندارد هیچ حق تقدیمی در برنامه‌ها یا درایورها و یا هر چیز دیگری در سیستم وجود ندارد و همه چیز آزاد است.

مہارت ICT وکار با لینوکس



LinuxTM



ICT به معنی Information and Communication Technology skills یا

مهارت استفاده از اطلاعات و ارتباطات در محیط لینوکس می‌باشد.

به عبارت ساده‌تر می‌خواهیم بدانیم چگونه در لینوکس می‌توانیم کارهای روزمره و یا

پروژه‌های اجرایی خود را به کار ببریم.

خب اول به این سؤال جواب بدهیم که لینوکس در کجاها یافت می‌شود؟

امروز می‌توان لینوکس را در همه‌جا مشاهده کرد، مانند:

– محیط server room

– virtualization hosts یا محیط‌های مجازی‌سازی مثل vmware

– cloud computing یا همان مبحث رایانش ابری مانند سرویس آمازون.

– سیستم‌های الحاقی یا embedded مانند دستگاه‌های موبایل.

– factory automation که از محیط DOS به لینوکس تغییر یافته است.

– mars rover





اما آنچه در ادامه در مورد آن می‌خواهیم صحبت کنیم درباره استفاده لینوکس در محیط دسکتاپ یا صفحه شخصی کامپیوتر شما می‌باشد.

از جمله مطالبی که در این قسمت پوشش خواهیم داد، عبارت‌اند از:

- چگونگی استفاده از مدیر دسکتاپ یا Desktop Manager

- برنامه‌های کاربردی آفیس

- مرورگرها Browser

- نحوه ذخیره فایل‌ها

- مفهوم ترمینال

- چگونگی به‌کارگیری کلمه رمز یا پسورد

در شکل زیر یک محیط شخصی‌سازی به‌نام KDE را نمایش داده‌ام. یک محیط

کاملاً گرافیکی شبیه آنچه در محیط ویندوز دارید و با همه ابزارهای گرافیکی مربوط به آن.





همچنین در شکل زیر محیط Gnome



همچنین شکل مقابل که محیط دسکتاپ اوبونتو به نام unity می باشد.
و بسیاری محیط های گرافیکی دیگر در مدیریت کارهای روزانه در سیستم عامل های
مختلف لینوکس داریم.

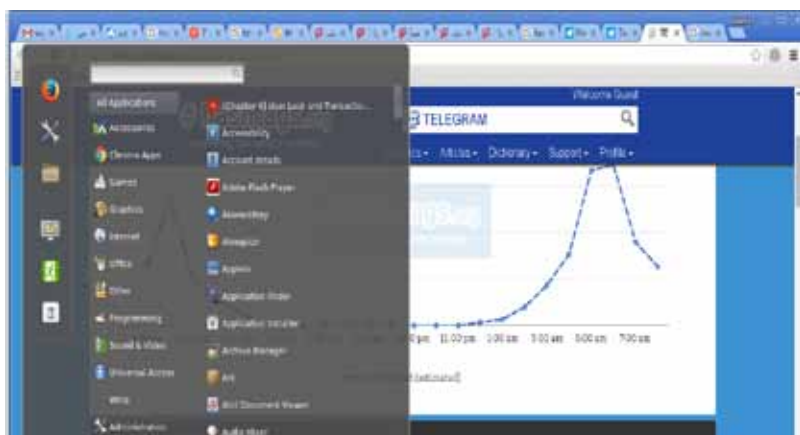


هر یک از نسخه‌های لینوکس به‌طور پیش‌فرض از یکی از این محیط‌های گرافیکی برای محیط دسکتاپ خود استفاده می‌کنند. مثلاً در نسخه SUSE به‌طور پیش‌فرض از KDE استفاده می‌شود و در fedora, centos, به‌صورت پیش‌فرض GNOME استفاده می‌شود یا در نسخه ubuntu از محیط unity استفاده می‌شود که هر کدام ویژگی‌های مخصوص خود را برای مدیریت دسکتاپ به کار می‌برند.

اما در هر کدام از نسخه‌های لینوکس می‌توان هر کدام از محیط‌های گرافیک دیگر را نیز نصب و اجرا کرد. مثلاً در SUSE می‌توان به جای محیط پیش‌فرض گرافیکی آن KDE از GNOME یا unity یا محیط‌های دیگر نیز استفاده نمود و به راحتی می‌توان هر کدام از محیط‌های مورد علاقه کاربر را نصب و به کار برد.

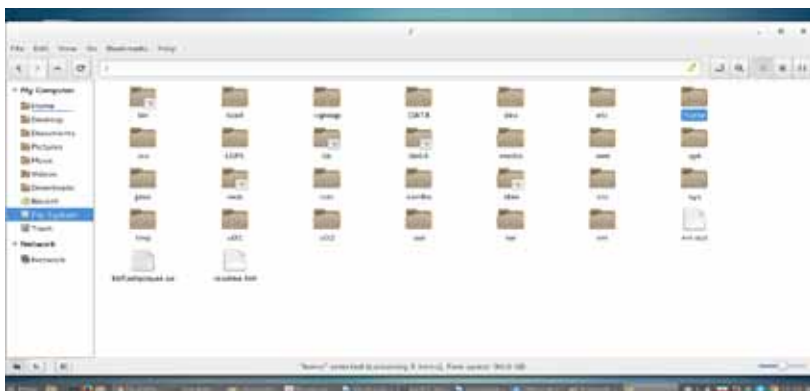
اینکه از چه نوع محیط گرافیکی برای کارهای روزانه‌تان استفاده می‌کنید کاملاً سلیقه‌ای بوده و بر اساس اینکه در کدام محیط راحت‌تر کار می‌کنید می‌توانید هر کدام را که مایلید روی سیستم نصب و به کار گیرید.

به عنوان نمونه من از محیط گرافیکی سیستم خودم که با 7 centos است نمونه‌ای نمایش دادم تا ببینید به راحتی می‌توان در این محیط گرافیکی کلیه کارهای روزمره را انجام داد





در داخل این محیط گرافیکی به راحتی به لیست فایل‌ها و دایرکتوری‌های خود دسترسی داشته با حرکت موس روی آن‌ها می‌توانید آن‌ها را حذف یا کپی و پیست نمایید، دقیقاً مثل زمانی که در محیط ویندوز کار می‌کنید.
به شکل زیر دقت نمایید:



در شکل بالا مشاهده می‌کنید که کلیه فولدرها و فایل‌ها به صورت گرافیک نمایش داده شده است و می‌توانیم با حرکت موس داخل هر پوشه‌ای که می‌خواهیم پیمایش کنیم یا فایل‌های داخل آن‌ها را حذف کرده یا جای دیگری منتقل کنیم.

برای یادگیری بیشتر نیاز دارید که در کلاس‌های آموزشی‌ای که در این زمینه ترتیب داده‌ام شرکت کنید تا از نزدیک با محیط گرافیکی و نحوه به کارگیری ابزارهای آن آشنایی پیدا نمایید.



در این قسمت قصد داریم به طور خلاصه در مورد ساختار فایل های داخل سیستم عامل لینوکس صحبت کنم.

فرض کنید یک هارد دیسک داریم که در محیط لینوکس قابل استفاده است. در زمانی که من به عنوان کاربر سیستم عامل لینوکس وارد این سیستم عامل می شوم، یعنی جایی که نام کاربری و پسورد خودم را می دهم و در صورت شناخته شدن وارد محیط گرافیکی خود می شوم در واقع در مسیری خاص بنام `/home/` قرار دارم.

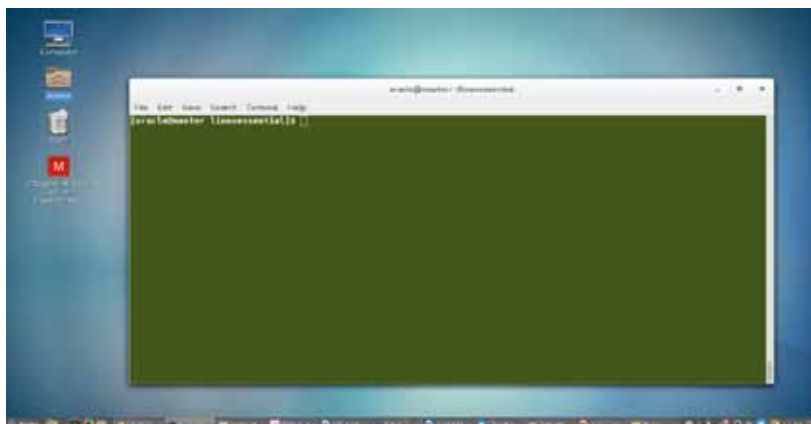
حالا اگر نام کاربری من مثلاً `Deeba` باشد در مسیر `/home/Deeba/` قرار دارم و در زیر این مسیر پوشه هایی با نام `Document`, `Download`, و یا پوشه های دیگر نیز وجود دارند.

این موضوع دقیقاً شبیه به آنچه در محیط ویندوز مشاهده می کنید می باشد. تمام مسیرها در لینوکس از یک مسیر اصلی ریشه یا `root` که با نماد `/` نشان داده می شود نشأت می گیرد.

`/home/` به معنی مسیر خانه می باشد.

یعنی هر کاربری در سیستم عامل دارای خانه اختصاصی خود می باشد و کلید ورود به خانه به هر کاربر مالک آن خانه داده می شود.

در تمام این محیط های گرافیکی ابزاری به نام ترمینال داریم که از آن می توانیم وارد محیط متنی شده و دستورات لینوکس را تحت آن اجرا نماییم.



شکل بالا نمایش یک ترمینال باز شده در محیط گرافیکی سیستم عامل لینوکس نسخه CentOS7 می باشد.

پس در هر سیستم عامل لینوکس به ازای کاربران تعریف شده در سیستم هر کس می تواند در محیط گرافیکی یا غیر گرافیکی وارد شده و کارهای خود را در آن دنبال نماید.

دستورات اولیه در لینوکس



LinuxTM



چگونه از دستورات ساده لینوکس استفاده کنیم.

برای اینکه در محیط لینوکس بتوانیم دستوراتمان را اجرا کنیم محیطی در اختیار کاربران قرار می‌گیرد که دستورات واردشده را به زبانی که لینوکس متوجه آن شود ترجمه می‌کند که به آن shell یا پوسته می‌گویند درواقع این پوسته دور هسته اصلی لینوکس کشیده شده است.

به‌طور پیش‌فرض کسی به‌طور مستقیم با هسته لینوکس صحبت نمی‌کند چون زبان صحبت با هسته متفاوت از زبانی است که من و شما برای صحبت معمولی استفاده می‌کنیم.

درست مثل اینکه شما به شهری با زبان دیگری می‌روید و در آنجا برای ترجمه حرف‌هایتان نیاز به مترجم دارید. در اینجا هم این پوسته کار ترجمه حرف‌ها و دستورات ما را برای هسته لینوکس انجام می‌دهد.

به محیطی که در آن دستورات لینوکس را وارد می‌کنیم تا برای هسته لینوکس آن را ترجمه نماید، محیط متنی پوسته لینوکس یا Linux Command Shell می‌گویند. درواقع این پوسته نقش مفسر را بازی می‌کند یعنی هر کلمه‌ای که در پوسته وارد کنیم سریعاً برای هسته ترجمه می‌کند. حالا اگر کلمه ما بتواند با توجه به دفترچه لغات سیستم‌عامل لینوکس قابل ترجمه بوده و معنی داشته باشد، نتیجه آن به ما باز می‌گردد و چنانچه کلمه یا جمله ما معنی خاصی برای سیستم‌عامل نداشته باشد با یک پیام خطا ما را متوجه این موضوع می‌کند.



خب قصد داریم در این قسمت به شرح موضوعات زیر بپردازیم:

- دستورات و گزینه‌های مربوط به آن‌ها

- استفاده از کاراکترهای خاص برای یافتن نام فایل‌ها و دایرکتوری‌ها

- استفاده از علامت نقل قول یا quote

- تعریف متغیرهای محیطی و چند مثال از آن‌ها مانند متغیر PATH

در ابتدا نیاز به محیطی داریم که دستورات لینوکس را در آن وارد کنیم.

نام این محیط ترمینال است که محیط پوسته را در اختیار ما قرار می‌دهد. برای استفاده از محیط ترمینال چنانچه در محیط گرافیکی باشیم، مثلاً اگر در محیط اوبونتو هستیم کفایت کلمه terminal را تایپ کنیم تا شکل گرافیکی آن نمایش داده شود و سپس با انتخاب آن و زدن کلید موس آن را انتخاب می‌کنیم. در این حالت یک ترمینال به شکل زیر روی صفحه برای ما ظاهر می‌شود:



در شکل فوق روی صفحه بنفش ظاهر شده که همان ترمینال ماست این عبارت را مشاهده می‌کنید:

`$ oracle@master ~]`

این عبارت می‌گوید نام کاربری که با آن وارد سیستم شدم oracle است و کلمه master نام ماشینی که روی آن کار می‌کنیم. اینجا نام ماشین من master است و علامت ~ نشان‌دهنده مسیری که در آن هستم، که به معنی خانه شخصی من یا home directory می‌باشد.

و علامت \$ معمولاً برای مشخص کردن نشانه پوسته افرادی که root نیستند به کار می‌رود.



```
oracle@localhost:~$ ls -l
total 725280
-rw-r--r-- 1 oracle oinstall 415114411 May 29 18:13 "581672011"
-rw-r--r-- 1 oracle oinstall 475147407 May 29 18:26 1.txt
-rw-r--r-- 1 oracle oinstall 415114411 May 29 18:53 2.txt
-rw-r--r-- 1 oracle oinstall 146879 Aug 18 14:54 4. Filesystems and Storage
e Assembling partitions as RAID devices.pdf
-rw-r--r-- 1 oracle oinstall 24962853 May 29 18:37 "581672011"
-rw-r--r-- 1 oracle oinstall 24962853 May 29 12:47 581672011.txt
-rw-r--r-- 1 oracle oinstall 470322 May 29 12:47 581672021.txt
-rw-r--r-- 1 oracle oinstall 97335955 May 29 18:37 "581672031"
-rw-r--r-- 1 oracle oinstall 97335955 May 29 11:39 "581672031"
-rw-r--r-- 1 oracle oinstall 97335955 May 29 12:47 581672031.txt
-rw-r--r-- 1 oracle oinstall 10481456 May 29 18:37 "581672041"
-rw-r--r-- 1 oracle oinstall 10481456 May 29 12:47 581672041.txt
-rw-r--r-- 1 oracle oinstall 1669817 May 29 12:47 581672042.txt
-rw-r--r-- 1 oracle oinstall 23861 May 29 12:47 581672043.txt
-rw-r--r-- 1 oracle oinstall 16352007 May 29 12:47 581672051.txt
-rw-r--r-- 1 oracle oinstall 9361938 May 29 12:47 581672052.txt
-rw-r--r-- 1 oracle oinstall 49477497 May 29 18:37 "581672061"
-rw-r--r-- 1 oracle oinstall 49477497 May 29 11:29 "581672061"
-rw-r--r-- 1 oracle oinstall 49477500 May 29 12:47 581672061.txt
-rw-r--r-- 1 oracle oinstall 28144 May 29 12:47 581672061.txt
-rw-r--r-- 1 oracle oinstall 8968316 May 29 18:37 "581672082"
```

در شکل، من دستور `ls -l` را جلوی علامت `$` اجرا کرده‌ام و با زدن کلید `Enter` این فرمان یا این عبارت به پوسته داده می‌شود و حالا پوسته این عبارت را با کتاب مترجم خود یا دیکشنری خود مقایسه کرده و چنانچه بتواند ترجمه کند، آن را به هسته لینوکس یا کرنل می‌دهد و کرنل آن را اجرا کرده و جواب خروجی را به ما برمی‌گرداند. عبارت واردشده یعنی `ls -l` از دو قسمت تشکیل شده است:

۱- فرمان `ls`

۲- گزینه `-l` به همراه این دستور

دستور `ls` که مخفف کلمه `list` می‌باشد از کلیه فایل‌ها و دایرکتوری‌های مسیری که در حال حاضر در آن قرار داریم یعنی مسیر خانه شخصی ما لیستی تهیه کرده و از خصوصیات آن‌ها به صورت جزء به جزء اطلاعاتی روی ترمینال ظاهر می‌کند.

و گزینه `-l` به معنی این است که این لیست را با جزییات کامل به من نمایش بده. در لیست ظاهرشده، ستون اول از چپ مجوزهای دسترسی این فایل و ستون سوم مالک فایل و ستون چهارم گروه این فایل و ستون بعدی سایز فایل و ستون بعدی تاریخ ایجاد یا آخرین تغییر فایل و ستون آخر نام فایل را نمایش می‌دهد.

فرمان `ls` را اگر بدون هیچ گزینه‌ای اجرا نماییم لیست نمایش داده‌شده فقط شامل



نام فایل‌ها و دایرکتوری‌های مسیرمان می‌شود و اطلاعاتی کامل‌تر از مالک و سایز و غیره نمی‌دهد.

دستور بعدی دستور cat می‌باشد

این دستور برای نمایش محتوای یک فایل به کار می‌رود که عموماً این فایل‌ها به صورت متنی هستند حالا ممکن است این متن یک متن برنامه‌نویسی به زبانی خاص باشد یا متن خاصی از یک کتاب و یا هر متنی دیگر.

ولی عموماً دستور cat برای نمایش محتوای فایل‌ها به کار می‌رود.

حالا در ترمینال بعد از اجرای فرمان ls -l لیست فایل‌ها را مشاهده می‌کنیم و یکی از فایل‌ها را انتخاب کرده و با دستور cat محتوای آن را نمایش می‌دهیم.

```
oracle@master:~$ cat 581672061.txt
581672061.txt
cat: 581672061.txt: No such file or directory
```

`oracle@master ~]$ cat "581672061".txt`

در مثال بالا "581672061".txt نام یکی از فایل‌های من می‌باشد که با اجرای دستور بالا محتوای داخل فایل نمایش داده می‌شود.

یا مثلاً دستور `cat tel.site` با فرض اینکه فایلی به نام `tel.site` وجود داشته باشد محتوای داخل آن را نمایش می‌دهد که در شکل زیر خروجی را مشاهده می‌کنید.

می‌توانیم در اجرای فرمان cat از نام چند فایل استفاده کنیم. مثلاً اگر `file1`, `file2` داشته باشیم با اجرای فرمان `cat file1 file2` خروجی ظاهر شده روی ترمینال از محتوای دو فایل به نام‌های `file1`, `file2` می‌باشد.

در خروجی لیست فایل‌های من فایل‌هایی هستند که همگی با عدد شروع شده‌اند و انتهای آن‌ها رشته یا عبارت `txt` به چشم می‌خورد. حالا چنانچه من بخواهم لیست کلیه



فایل‌هایی را ببینیم که می‌دانم انتهای آن‌ها عبارت txt وجود دارد، کافی است در دستور ls یا cat از علامت * به معنی هر چیزی به صورت زیر استفاده کنیم:

می‌خواهم لیستی از فایل‌ها و یا پوشه‌هایی بگیرم که نام آن‌ها هر چه می‌خواهد باشد اما انتهای آن‌ها عبارت txt وجود داشته باشد. به این منظور دستور زیر را وارد می‌کنیم:

```
oracle@master ~]$ ls *.txt
```

```
oracle@master ~]$ cat *.txt
```

در دو دستور بالا حرف * به معنی هر کاراکتری که باشد، یعنی می‌خواهد نام فایل عدد باشد یا حروف الفبا باشد یا هر چیز دیگری برایم مهم نیست، بلکه مهم این است که انتهای فایل txt باشد.

پس هنگامی که بخواهیم به صورت دسته‌جمعی دستوری را اجرا نماییم از علامت * به معنی رخداد صفر تا هر تعداد از یک واقعه را جایگزین می‌کنیم.

در مثال‌های بالا هر فایلی، چه یک حرفی، دو حرفی یا هر تعداد حرفی باشد، به شرط آن که انتهای آن رشته یا عبارت txt وجود داشته باشد برای ما نمایش داده می‌شود.

حالا اگر بخواهیم فقط به جای یک حرف خاص از علامتی استفاده کنیم، مثلاً فقط می‌خواهیم بگوییم کلیه فایل‌هایی را نشان بده که یک حرفی هستند، آن وقت از علامت * به صورت زیر استفاده می‌کنیم:

```
oracle@master ~]$ ls -l ?.txt
```

در مثال بالا لیست کلیه فایل‌ها و پوشه‌هایی را لازم داریم که نامشان یک حرفی است حالا این حرف الفبا باشد یا عدد مهم نیست ولی انتهای آن با txt. خاتمه می‌یابد. حالا اگر چنین فایل‌هایی در مسیر موردنظرمان وجود داشته باشد که نام فایل‌ها را مشاهده می‌کنیم و اگر وجود نداشته باشد سیستم پیغام خطا داده و می‌گوید چنین فایلی وجود ندارد.

یا وقتی بگوییم:



oracle@master ~]\$ cat ?.txt

یعنی محتویات تمام فایل‌های یک حرفی که انتهای آن‌ها عبارت txt است را نمایش بده.

یکی دیگر از دستورات ساده لینوکس دستور echo یا همان انعکاس می‌باشد. دستور echo هر عبارتی را که جلوی آن قرار گیرد، نمایش می‌دهد. یا به عبارتی روی صفحه منعکس می‌کند. مثلاً:

دستور Hello echo کلمه Hello را روی صفحه نمایش می‌دهد.
حالا اگر در دستور بالا یعنی ehco Hello بین کلمه echo و Hello هر تعداد فاصله خالی هم بگذارم باز هنگام نمایش روی صفحه تنها یک فاصله بین آن‌ها قرار می‌دهد. همچنین اگر دستور زیر را وارد کنیم:

```
echo Hello World
```

و یا

```
echo Hello World
```

خروجی در دو حالت بالا چه زمانی که فقط یک فاصله بین دو کلمه Hello world باشد یا زمانی که تعداد بیشتری فاصله بین این دو حرف باشد یعنی Hello world خروجی فرمان به صورت فقط یک فاصل ظاهر می‌شود، یعنی به صورت زیر:

```
Hello World
```

چنانچه قصد ما از نمایش متن این است که دقیقاً فاصله بین دو حرف عین آن چیزی که می‌نویسیم در نظر گرفته شود، بنابراین دو کلمه را در داخل علامت نقل قول یا “” قرار می‌دهیم بنابراین برای ظاهر شدن دو کلمه Hello world با در نظر گرفتن بیشتر از یک فاصله بین آن‌ها، این عبارت را داخل علامت نقل قول قرار می‌دهیم، یعنی به صورت زیر:

```
echo “Hello world”
```



در این حالت گزینه ورودی به دستور ehco به صورت یک رشته با در نظر گرفتن فاصله موجود بین آن‌ها به دستور داده شده و خروجی آن نیز عیناً مانند آنچه در داخل علامت نقل قول نشان دادیم روی صفحه نمایش داده می شود.

```
oracle@master:~$ echo ehco
ehco
```

حالا فرض کنیم من دستور زیر را روی ترمینال اجرا می کنم:

```
oracle@master ~]$ a=hello there
```

خروجی دستور با خطا مواجه می شود و به صورت زیر پیامی روی صفحه مشاهده می کنیم:

در شکل، بعد از خط اجرا، پیام bash: there command not found را مشاهده می کنید. این پیام، یعنی من دستوری به نام there پیدا نمی کنم که اجرا نمایم.

درواقع با زدن دستور a=hello there پوسته مجموعه این عبارات را به هسته یا کرنل لینوکس می فرستد اما وقتی هسته در کتابچه لیست دستورات خود نگاه می کند می بیند که در ابتدا عبارت a=hello دارد که برای سیستم عامل این معنی را می دهد که متغیری به نام a وجود دارد که با دستور بالا می گوئیم مقدار این متغیر را برابر با کلمه hello قرار بده و بعد کلمه there را می بیند و این کلمه برای سیستم عامل، چون به دنبال عبارت a=hello آمده، به معنی نام یک دستور می باشد، یعنی سیستم عامل در لیست دستوراتش به دنبال دستوری با نام there می گردد و می بیند چنین دستوری ندارد. پس در خروجی می نویسد که من چنین دستوری ندارم که بفهمم منظورش چیست.



there را به متغیر a تخصیص دهیم این اتفاق نمی افتد و فقط مقدار hello به متغیر a تخصیص داده می شود و چون فاصله ای بین کلمه hello و world وجود دارد این فاصله برای سیستم عامل به معنی اجرای فرمان بعدی می باشد.

```
oracle@master:~$  
File Edit View Search Terminal Help  
oracle@master ~$ a=hello there  
bash: there: command not found...  
oracle@master ~$ a="hello there"  
oracle@master ~$ echo $a  
hello there  
oracle@master ~$
```

حالا اگر منظور ما دقیقاً این باشد که عبارت hello there را به متغیر a نسبت بده در این صورت باید این عبارت را داخل علامت نقل قول قرار دهیم، یعنی به این صورت

a="hello there"

در این صورت مقدار متغیر a برابر با رشته hello there خواهد شد که شکل بالا گویای خروجی آن است. حالا برای نمایش مقدار متغیر a با دستور \$a echo محتوای این متغیر را نمایش می دهیم.

پس زمانی که علامت \$ را به همراه نام متغیر می آوریم منظورمان محتوای این متغیر می باشد که همه جا از آن استفاده می کنیم.

```
File Edit View Search Terminal Help  
oracle@master ~$ a="hello there"  
oracle@master ~$ b="good day"  
oracle@master ~$ echo $a $b  
hello there good day  
oracle@master ~$
```

در شکل مثالی دیگر از به کارگیری مقدار متغیرها می باشد. در مثال بعدی که در شکل مشاهده می کنید مقدار متغیر a را برابر با مقدار "hello there" و مقدار متغیر b را برابر با "good day" قرار دادم. حالا با دستور \$a \$b echo مقدار این دو متغیر به صورتی که در شکل با یک فاصله می بینید ظاهر می شود.

حالا چنانچه دستور زیر را وارد کنیم:



echo "\$a\$b"

خروجی به صورتی ظاهر می شود که هیچ فاصله ای بین دو رشته حرفی helo there و good day وجود ندارد.

```

oracle@ma
File Edit View Search Terminal Help
[oracle@master ~]$ a="hello there"
[oracle@master ~]$ b="good day"
[oracle@master ~]$ echo $a $b
hello there good day
[oracle@master ~]$ echo "$a$b"
hello theregood day
[oracle@master ~]$

```

در شکل، در داخل علامت نقل قول محتوای دو متغیر a, b را بدون هیچ فاصله ای قراردادیم و خروجی آن، مقدار محتوای این دو متغیر را بدون فاصله کنار هم نمایش می دهد، یعنی به صورت hello theregoodday.

با گفتن این مقدمه در مورد متغیرها و چگونگی مقداردهی به آن ها، می خواهیم راجع به بعضی از متغیرهای داخل سیستم عامل لینوکس صحبت کنم.

در سیستم عامل لینوکس یکسری متغیر داریم که برای سیستم عامل معنی خاصی دارد و در آن ها مقادیر خاصی را ذخیره و نگهداری می کند و در جای مناسب از آن ها استفاده می کند. به عنوان مثال یکی از این متغیرها PATH نام دارد. این متغیر همان طور که از اسم آن پیداست مسیری را در خود نگهداری می کند. PATH یعنی مسیر.

در ضمن، کلیه متغیرهای داخل سیستم عامل که خود سیستم عامل از آن ها استفاده می کند با حروف بزرگ نگهداری شده و معنی خاص خود را دارند.

یادمان باشد که در لینوکس حروف بزرگ و کوچک هر کدام به صورت مجزا هستند، یعنی اگر فایلی با نام txt.File و فایلی با نام txt.file داشته باشیم این دو نام، دو نام مجزا برای سیستم عامل محسوب می شوند. در واقع یعنی دو فایل داریم که یکی با حرف F بزرگ و دیگری با حرف f کوچک نوشته شده است.

[illegible]

فعلاً تا اینجا بدانید که متغیری به این نام در سطح سیستم عامل وجود دارد و کارش این است که یکسری مسیر و آدرس‌هایی را در خود ذخیره می‌کند که سیستم عامل از این مسیرها استفاده می‌کند.

مثلاً



```
oracle@racnode1:~$
```

```
File Edit View Search Terminal Help
```

```
oracle@racnode1:~$ echo $PATH; echo $B
```

```
/usr/local/oracle/product/11.2.0/rownum: /bin:/usr/sbin:/usr/bin:/usr/local/app/oracle/product/11.9.0/er: /bin:/usr/sbin:/usr/local/oracle/product/11.2.0/dhmm: /bin:/usr/sbin:/usr/bin:/usr/local/er: /bin:/usr/local/bin:/usr/local/sbin:/usr/sbin:/usr/local/erache2/bin:/opt/hadoop/bin:/usr/local/php7/usr/bin:/usr/local/erache2/bin:/opt/hadoop/hadoop/bin:/usr/local/php7/usr/bin
```

```
hello there
```

```
good day
```

```
oracle@racnode1:~$
```

درواقع به جای اینکه هر دستور را در خطوط مجزا بدهیم همگی را در یک خط داده و با علامت آن‌ها را جداسازی می‌نماییم.

* برای کلیه حروف یک عبارت مثل: ls *.txt

”برای قرار دادن یک رشته که بین آن‌ها فاصله است مثلاً `a="hello Ali "`

ls;echo \$PATH : برای جداسازی فرامین در یک خط فرمان مثل:



تا اینجا با تعدادی از دستورات ساده لینوکس آشنا شدید.

از آنجاکه نمی‌خواهم در ابتدای راه گیج شوید به همین مقدار بسنده می‌کنم. بسیاری از فرامین مفید دیگری هست که با شرکت در دوره‌ها و کلاس آموزشی لینوکس می‌توانید با آن‌ها آشنا شده و موارد استفاده آن‌ها را بیاموزید.

چگونه در لینوکس از Help استفاده کنیم



LinuxTM



خب فرض کنید که رفتید به فضا و زبان فضایی‌ها را بلد نیستید و دنبال این هستید که حرفه‌ای‌تان را برایشان ترجمه کنید در این حالت چه کار می‌کنید؟

خب نیاز به یه مترجم یا یه راهنما دارید که در جاهایی که کلمات آن‌ها را متوجه نمی‌شوید به آن مراجعه و ترجمه حرف‌های فضایی‌ها را در آن بیابید.

دقیقاً همین کار را در سیستم عامل لینوکس می‌کنیم. یعنی می‌توانیم در ترمینال و در قسمتی که علامت shell ظاهر شده است دستور مورد نظر را با کلمه help- وارد کنیم

یعنی زمانی که ما اصلاً معنی بعضی از دستورات لینوکس را متوجه نمی‌شویم نیاز به یک مترجم داریم که آن‌ها را برایمان ترجمه کند و بگوید نحوه استفاده از آن چگونه است.

اما روش‌های مختلفی در سیستم عامل لینوکس وجود دارد که می‌تواند ما را در مسیر یادگیری صحیح دستورات به خوبی راهنمایی کند.

که عبارت‌اند از:

- Man
- Info
- Main -k
- Apropos
- Which



- Whereis

یکی از روش‌هایی که این مشکل را برای ما برطرف می‌کند استفاده از دستور `man` می‌باشد. دستور `Man` مخفف `manual` می‌باشد به معنی دستورالعمل که به ما توضیحات کاملی از یک دستور به همراه تمام گزینه‌های مرتبط با این دستور را نمایش می‌دهد. به منظور به کارگیری `man` باید این دستور را به همراه دستوری که می‌خواهیم راجع به آن اطلاعاتی کسب کنیم به کار ببریم. مثلاً:

```
$ man ls
```

یعنی می‌خواهم بدانم دستور `ls` چیست چکار می‌کند و با چه گزینه‌هایی قابل استفاده است.

برای اینکه بدانیم با خود دستور `man` چگونه کارکنیم، از دستور `man man` استفاده می‌کنیم که صفحه‌ای به شکل زیر ظاهر می‌شود:

```
MAN(1) Manual pager utils MAN(1)
NAME
man - an interface to the on-line reference manuals

SYNOPSIS
man [-C file] [-d] [-D] [--warnings=warnings] [-R encoding] [-l
locale] [-m system[...]] [-M path] [-S list] [-e extension] [-i|-I
|--regex] [--wildcard] [--names-only] [-a] [-u] [--no-subpages] [-P
pager] [-r prompt] [-7] [-E encoding] [--no-hyphenation] [--no-justi
fication] [-p string] [-t] [-T device] [-H browser] [-X dpi] [-Z
[section] page ...] ...
man -k [apropos options] regexp ...
man -K [-w|-W] [-S list] [-i|-I] [--regex] [section] term ...
man -f [whatis options] page ...
man -l [-C file] [-d] [-D] [--warnings=warnings] [-R encoding] [-l
locale] [-P pager] [-r prompt] [-7] [-E encoding] [-p string] [-t
-T device] [-H browser] [-X dpi] [-Z] file ...
man -w|-W [-C file] [-d] [-D] page ...
man -c [-C file] [-d] [-D] page ...
man [-?V]

DESCRIPTION
Manual page man(1) line 1 (press h for help or q to quit)
```

با زدن دستور `man ls` صفحه‌ای به شکل زیر ظاهر می‌شود:



```
File Edit View Terminal Tabs Help
LS(1) User Commands LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default).
    Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
    fied.

    Mandatory arguments to long options are mandatory for short options
    too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author
    Manual page ls(1) line 1 (press h for help or q to quit)
```

یک روش دیگر برای فهمیدن اینکه یک دستور در لینوکس چگونه کار می‌کند، استفاده از `info` می‌باشد.

برای استفاده از دستور `info` این فرمان را با یکی از دستورات دلخواهتان اجرا می‌کنیم مانند شکل زیر:

`$ info ls`

```
Terminal - sophia@devuan: ~
File Edit View Terminal Tabs Help
LS(1) User Commands LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default).
    Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
    fied.

    Mandatory arguments to long options are mandatory for short options
    too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author
    Manual page ls(1) line 1 (press h for help or q to quit)
```



این دستور هم مانند Man ls توضیحاتی راجع به دستور می‌دهد و در بعضی از موارد مثال‌هایی از گزینه‌های مختلف به همراه دستور را نمایش می‌دهد. شکل زیر نمایشی از این دستور می‌باشد.

```
File Edit View Terminal Tabs Help
1 1m 0 Daily Backup at 6pm

4 tasks
sophia@devuan:~$ man man
sophia@devuan:~$ man ls
sophia@devuan:~$ info ls
sophia@devuan:~$ man -k ping
Compose (5) - X client mappings for multi-key input sequences
blkmapped (8) - pNFS block layout mapping daemon
getkeycodes (8) - print kernel scancode-to-keycode mapping table
loadunimap (8) - load the kernel unicode-to-font mapping table
mapscrm (8) - load screen output mapping table
ntfs-3g.usermap (8) - NTFS Building a User Mapping File
ping (8) - send ICMP ECHO_REQUEST to network hosts
ping6 (8) - send ICMP ECHO_REQUEST to network hosts
setkeycodes (8) - load kernel scancode-to-keycode mapping table entries
swapoff (8) - enable/disable devices and files for paging and swap...
swapon (8) - enable/disable devices and files for paging and swap...
Text::WrapI18N (3pm) - Line wrapping module with support for multibyte, full...
ts (1ssl) - Time Stamping Authority tool (client/server)
tsget (1ssl) - Time Stamping HTTP/HTTPS client
XCompose (5) - X client mappings for multi-key input sequences
xmodmap (1) - utility for modifying keymaps and pointer button map...
sophia@devuan:~$
```

در هنگام استفاده از دستور man ls ما باید نام دستوری را که می‌خواهیم همراه man بیایم، بدانیم، یعنی دستور ls. حالا فرض کنید که اصلاً ما نمی‌دانیم که نام دستورمان چیست ولی مثلاً می‌دانیم که این دستور قرار است چکاری انجام دهد یا مرتبط با چه کاری می‌باشد.

مثلاً می‌خواهیم دستوری مرتبط با پیکربندی شبکه را ببایم. خوب پس می‌دانیم که این دستور مرتبط با شبکه است اما نمی‌دانیم که نامش چیست. در این حالت کافی است که دستور man را با گزینه -k به کار ببریم که به صورت زیر نمایشی از آن را می‌دهم:

`$man -k ping`

در دستور بالا من می‌دانم که مثلاً می‌خواهم یک ارتباط شبکه‌ای را تست نمایم. پس حداقل می‌دانیم که ping بین دو ماشین را می‌خواهم حالا با اجرای دستور man -k ping لیستی از دستورات مرتبط با ping برای ما ظاهر می‌شود که وقتی این لیست را



ببینیم دستوری را که به دنبال آن هستیم خواهیم یافت. ممکن است این لیست طولانی و شامل مثلاً ۲۰۰ یا ۳۰۰ دستور مرتبط باشد اما برای هر دستور در یک خط یک توضیح کوتاه از کار آن دستور داده شده است که بدین وسیله کاربر متوجه می شود که این دستور آیا همان دستوری هست که دنبالش هست یا خیر؟ این دستور در بسیاری موارد کمک کننده می باشد و خیلی از موارد حتی مدیران سیستم تمامی دستورات را حفظ نیستند و با کمک این روش می توانند دستوری را که دنبالش هستند، بیابند. شکل زیر نمایشی از اجرای این دستور می باشد.

روشی دیگر استفاده از دستور apropos می باشد دستور apropos به همراه یک دستور درست مانند man-k عمل می کند. یعنی لیستی از دستورات مرتبط با دستوری را که همراه apropos آمده است، می دهد. مثلاً apropos ping یا apropos echo. تمامی این موارد لیست دستورات مرتبط با دستور ping یا دستور echo را نمایش می دهند.

یکی از روش هایی که می توانیم ببینیم در سیستم عامل لینوکس چه دستوراتی داریم و توضیحات هر دستور را به صورت یک دفترچه راهنمای کامل مطالعه کنیم، رفتن به مسیر `usr/share/doc/` می باشد. وقتی به این مسیر می رویم لیستی از تمامی دستورات لینوکس را که در سطح سیستم عامل نصب شده است، مشاهده می کنیم که به صورت پوشه هایی از نام آنها قابل مشاهده است. حالا اگر داخل یک پوشه برویم لیستی از فایل های داخل آن پوشه می بینیم که به صورت فشرده یا gz موجود می باشد. مثلاً فرض کنید که دستور python را روی سیستم نصب کرده ایم و می خواهیم در مورد آن مطالعه کاملی داشته باشیم. بنابراین به مسیر `usr/share/doc/python/` می رویم و لیستی از فایل های این مسیر را مشاهده می کنیم حالا برای آنکه فایل های فشرده شده را بدون اینکه بخواهیم از حالت فشرده گی در بیاوریم مشاهده کنیم، از دستور less به همراه نام فایل استفاده می کنیم.



مثلاً `less python-policy.txt.gz` با اجرای این دستور توضیحات موجود در این فایل فشرده به‌طور کامل و صفحه به صفحه نمایش داده می‌شود که می‌توانیم آن را به‌طور کامل مطالعه نماییم.

روش دیگر برای اینکه بفهمیم یک دستور چکار می‌کند، استفاده از دستور `whatis` می‌باشد. بدین‌صورت که این دستور را به همراه دستوری که می‌خواهیم از آن اطلاعات کسب نماییم به کار می‌بریم.

مثلاً `whatis ls` یا `whatis cat` که در این صورت یک خط خروجی و توضیح کوتاهی از اینکه این فرمان چیست به ما نشان داده می‌شود. مانند شکل زیر: دستور دیگر `whereis` می‌باشد این دستور به همراه دستور `whereis ls` می‌گوید که دستور `ls` از چه مسیرهایی اجرا می‌شوند و در چه مسیری مستندات مربوط به این دستور قرار دارد.

مثلاً `whereis ls`

```
sophia@devuan:~$  
sophia@devuan:~$  
sophia@devuan:~$ whatis ls  
ls (1)                - list directory contents  
sophia@devuan:~$
```

روش دیگر استفاده از راهنمای دستورات استفاده از دستور `which` می‌باشد. این دستور زمانی به کار می‌رود که می‌خواهیم بدانیم که دستوری که در حال اجرای آن هستیم یا می‌خواهیم اجرا کنیم از چه مسیری در حال اجرا شدن است. به عبارتی مسیر اجرایی دستورات را به ما نشان می‌دهد. به عبارت بهتر می‌گوید کرنل از کجا متوجه می‌شود که مثلاً دستور `ls` از کجا اجرا شود.

برای مثال می‌دانیم در سطح سیستم‌عامل دستوری به نام `ls` داریم و حالا فرض کنید که کاربر نیز برنامه‌ای نوشته و نام آن را `ls` گذاشته است. حالا وقتی من فرمان `ls` را اجرا می‌کنم سیستم‌عامل کدام `ls` را اجرا می‌کند؟ آیا `ls`ی که خود سیستم‌عامل جزو



دستورات اولیه‌اش دارد یا دستور ls که خودم نوشته‌ام؟

در این حالت با اجرای فرمان `which ls` می‌توانیم بفهمیم که مسیر اجرایی که سیستم‌عامل برای اجرای دستور ls استفاده می‌کند چیست.

پس به‌طور خلاصه در این فصل یاد گرفتیم:

۱- با استفاده از گزینه `help` با یک دستور دلخواه، نحوه استفاده از دستور به همراه گزینه‌های آن دستور را به‌طور خلاصه روی صفحه می‌بینیم. مثل `ls --help` یا `date --help` که به‌طور خلاصه توضیحاتی از دستور ls یا دستور date به همراه گزینه‌هایی که با این دستورات همراه است برای ما نمایش داده می‌شود.

۲- دستور `man ls` یا `man echo` یا `man ping` یعنی همراه کردن دستور با دستور `man` به ما توضیحی کامل از دستور و گزینه‌های آن می‌دهد.

۳- دستور `ls info` یا `ping info` یا `man Info` یعنی دستور `info` به همراه دستور دلخواه برای ما توضیحاتی از دستور را می‌دهد.

۴- دستور `ls -k man` یعنی همراه کردن `man -k` به همراه دستور موردنظر یا قسمتی از دستوری که به یاد می‌آوریم باعث می‌شود لیستی از کلیه دستوراتی که به‌نوعی با عبارتی که مابعد از گزینه `k` وارد کرده‌ایم به همراه یک توضیح کوتاه از دستور روی صفحه ظاهر شود و در این صورت می‌توانیم دستورات را مشاهده و دستوری را که فراموش کرده‌ایم به یاد بیاوریم.

۵- دستور `ls apropos` که مانند `ls -k man` عمل می‌کند.

۶- دستور `ls whatis` که توضیح کوتاهی از دستور ls را در این مثلاً نشان می‌دهد.

۷- دستور `ls whereis` می‌گوید که دستور ls از چه مسیری اجرا می‌شود و همچنین در چه مسیری توضیحات این دستور آمده است.

۸- دستور `ls which` می‌گوید که می‌گوید مثلاً دستور ls از چه مسیری در حال

اجراست. این‌طوری ما می‌فهمیم که آیا دستور ls را از مسیر درست اجرا کرده‌ایم یا خیر؟

مدیریت فایل‌ها و دایرکتوری‌ها در لینوکس



LinuxTM



در این بخش قصد داریم به توضیح در مورد دستوراتی که به وسیله آن می توانیم لیست فایل ها و پوشه های خود را در یک مسیر خاص نمایش دهیم و دستورات مرتبط به لیست فایل ها و دایرکتوری ها را توضیح دهیم.

دستوراتی که در این بخش دنبال می کنیم عبارت اند از:

- نحوه استفاده از دستور `cd` یا `change directory`

- استفاده از مسیر مطلق و مسیر نسبی

- معنی `.` و `..` در مسیر

- دستور `pwd` یا `Present working directory`

لازم به توضیح است که در مورد ساختار فایل سیستم های در لینوکس در فصل دیگری توضیح خواهیم داد.

طریقه استفاده از دستور `cd`

گفتیم دستور `cd` یعنی مسیرت را عوض کن یا برو به فلان مسیر.

اولاً با فرض اینکه می دانیم در لینوکس همه چیز از `/` یا ریشه شروع می شود و هر مسیر از مسیر دیگر با علامت `" / "` جدا می گردد. حالا وقتی مثلاً بخواهیم به مسیر `home//oracle` برویم با استفاده از دستور `cd /home/oracle` این کار را انجام می دهیم.

یا مثلاً اگر بخواهیم به مسیر `etc/` بروم با اجرای فرمان `cd /etc` این کار را انجام

می دهیم.



اما در سیستم عامل لینوکس در هر مسیری دو فایل می بینیم، یکی با نام "." و دیگری با نام ".."

فایلی که نام "." را دارد به معنی مسیری است که در آن قرار داریم یا به اصطلاح home directory

و "." به معنی یک مسیر بالاتر از مسیری که در آن قرار داریم یا به اصطلاح parent مسیر جاری می باشد.

پس با اجرای فرمان "cd ." اتفاق خاصی نمی افتد. یعنی می گویم برو به مسیر جاری و این در حالی است که در مسیر جاری قرار داریم.

اما دستور "cd .." یعنی برو به یک مسیر بالاتر یا برو به مسیر والد من یا یک مسیر به عقب برگرد، یعنی اگر در مسیر home/oracle/ باشیم با اجرای فرمان "cd .." یعنی یک مسیر برو عقب، یعنی به مسیر home/ وارد می شویم.

همچنین علامت "~" برای سیستم عامل معنی دارد و معنی آن یعنی مسیر خانه ما یا به اصطلاح home directory یعنی "~ cd" یعنی برو به مسیر خانه خودت یا به home directory خودت برو. برای مثلاً فرض کنید در مسیر etc/ قرار داریم و می خواهیم به مسیر خانه خودمان برگردیم در این حالت می توانیم به دو صورت عمل کنیم

1- Cd /home/oracle

2- Cd ~

هر دو دستور بالا ما را به مسیر خانه خودمان هدایت می کند.

فرض کنید شخصی به تهران سفر می کند و به دنبال آدرس برج میلاد است و زمانی که به ما مراجعه می کند ما می توانیم بگوییم تهران - بزرگراه همت - خروجی برج میلاد یا اینکه نسبت به مکانی که در آن قرار داریم آدرس را به شخص مورد نظر بدهیم. یعنی دیگر نمی گوییم تهران - بزرگراه همت، بلکه بسته به اینکه الان کجا هستیم مثلاً آیا



در جنوب تهرانیم یا شمال یا شرق یا غرب، آدرس رسیدن به برج میلاد را می‌دهیم.

در حالتی که آدرس‌دهی را از نقطه آغازین یعنی تهران می‌دهیم از مسیردهی مطلق استفاده کرده‌ایم و زمانی که آدرس را نسبت به مسیری که الآن در آن قرار داریم می‌دهیم از مسیردهی نسبی استفاده نموده‌ایم.

خب پس وقتی من می‌گویم برو به مسیر خانه یا `home directory` وقتی می‌گویم `cd /home/oracle` یعنی مسیردهی مطلق استفاده کرده‌ام و وقتی مثلاً در `home/` قرار دارم و دستور دهم `cd oracle` یعنی نسبت به جایی که الآن قرار دارم برو به مسیر `oracle`. دقیقاً مثل مثال بالا که نسبت به آدرسی که در آن قرار داریم آدرس برج میلاد را به طرف مقابل می‌دهیم.

همچنین استفاده از دستور `cd ..`، یعنی دو تا مسیر برو عقب‌تر. یعنی اگر مثلاً در مسیر `home/oracle/app/product/` قرار داریم و دستور `cd ..` را اجرا کنیم به مسیر `home/oracle/` وارد می‌شویم.

یا مثلاً اگر در هر مسیری باشیم و دستور `cd ~/app` را اجرا کنیم یعنی برو به مسیر خانه من و در آنجا برو به مسیر `app`.

در اینجا خانه من منظور خانه کاربری است که با آن وارد سیستم شده‌ایم.

حالا از کجا بفهمیم که اصلاً در چه مسیری قرار داریم؟

با اجرای دستور `pwd` می‌فهمیم که الآن در چه مسیری قرار داریم. مثلاً فرض کنید دستور `pwd` را اجرا کردیم و دیدیم که مثلاً در مسیر `home/oracle/app/product/` قرار داریم حالا می‌توانیم مثلاً دستور `cd ..` را اجرا کنیم و یک مسیر به عقب یعنی به مسیر `home/oracle/app/` وارد شویم.

حالا که فهمیدیم چگونه به مسیرهای موردنظرمان در سیستم عامل لینوکس برویم می‌خواهیم بدانیم در این مسیرها چگونه لیست فایل‌ها و دایرکتوری‌های آن مسیر را ببینیم که این کار با اجرای دستور `ls` انجام می‌شود.



اما در هر مسیر فایل‌هایی هستند که با اجرای دستور ls به صورت ساده نمایش داده نمی‌شوند. به این نوع فایل‌ها پنهان می‌گویند. یعنی نامشان قابل رؤیت نیست. برای مشاهده نام این نوع فایل‌ها باید از دستور ls به همراه گزینه a- استفاده نماییم.

بنابراین اجرای دستور "ls -a" لیستی از تمامی فایل‌ها و دایرکتوری‌های مسیر موردنظر که چه پنهان باشند یا نباشند را نشان می‌دهد. توجه داشته باشید که در هر مسیری دو تا فایل به صورت پنهان وجود دارند، یکی به نام "." و دیگری به نام ".." که "به معنی مسیر جاری و "به معنی مسیر عقب‌تر از این مسیر یا والد این مسیر می‌باشد. حالا فرض کنید در مسیر موردنظر فایلی با نام "httpasswd" وجود داشته باشد.

دیدن نام این فایل بدون گزینه "ls -a" امکان‌پذیر نمی‌باشد.

برای مشاهده لیست فایل‌های داخل یک مسیر می‌توانیم از گزینه "ls -la" که نام فایل‌ها و دایرکتوری‌ها را با جزئیات بیشتر و همچنین نام فایل‌های پنهان را نیز مشاهده نماییم.

ایجاد، حذف و یا انتقال فایل‌ها در لینوکس



LinuxTM



در این بخش قصد داریم به چگونگی استفاده از دستوراتی که برای ایجاد یک فایل جدید یا حذف فایل یا انتقال فایل به مسیر دیگر صحبت کنیم. در این راستا لیستی از دستورات به شرح زیر به کار می‌رود:

- دستور touch که برای ایجاد یک فایل خالی به کار می‌رود.
 - دستور rm یا remove که برای حذف یک فایل و دایرکتوری به کار می‌رود.
 - دستور mv یا move directory or file که برای انتقال فایل یا دایرکتوری و یا تغییر نام فایل یا دایرکتوری استفاده می‌شود.
 - دستور cp برای کپی کردن فایل و دایرکتوری
 - دستور mkdir برای ایجاد یک دایرکتوری خالی
 - دستور rmdir برای پاک کردن یک دایرکتوری خالی
- در هر کدام از این دستورات با استفاده از گزینه `-r` و یا `-R` می‌توانیم دستورات را به صورت تودرتو یا recursive انجام دهیم.
- حالا به توضیح هر یک از دستورات بالا می‌پردازم.

دستور touch

این دستور برای ایجاد یک فایل با سائز صفر یا یک فایل خالی استفاده می‌شود. یعنی اگر فایلی به نام myfile وجود نداشته باشد با استفاده از دستور touch myfile در مسیر جاری یک فایل با نام myfile که خالی است ایجاد می‌گردد.



حالا اگر مثلاً فایل‌ای به نام test داشته باشیم و دستور touch test را بزنیم اتفاقی که می‌افتد این است که زمان فایل تغییر می‌کند. یعنی با فرمان "ls -l" می‌توانیم مشاهده کنیم که زمان فایل روی سیستم تغییر کرده است.

نکات مورد توجه

در لینوکس حروف بزرگ از کوچک متمایز است. یعنی فایل‌ی با نام Myfile با فایل‌ی با نام myfile دو فایل متفاوت هستند که در یکی با حرف بزرگ M و در دیگری با حرف کوچک m نام فایل آمده است.

فاصله بین نام فایل یعنی اگر می‌خواهیم فایل‌ی با نام مثلاً My File داشته باشیم در هنگام استفاده از دستور touch My File به جای یک فایل، دو فایل با نام‌های My File خواهیم داشت حالا اگر منظورمان تنها یک فایل با نام My File است در این صورت نام فایل را در داخل " " قرار می‌دهیم. یعنی دستور را به این شکل اجرا می‌کنیم "touch "My File" در این حالت رشته "My File" به همراه فاصله بین آن‌ها به صورت نام فایل لحاظ خواهد شد.

برای ساختن یک پوشه یا دایرکتوری از دستور mkdir به همراه نام پوشه استفاده می‌کنیم.

مثلاً `mkdir Mydir1` یعنی پوشه‌ای با نام `Mydir1` ایجاد کن.

حالا اگر این پوشه از قبل وجود نداشته باشد با اجرای این دستور ایجاد می‌شود. قوانین فاصله در نام فایل یا دایرکتوری و حروف بزرگ و کوچک در مورد نام دایرکتوری‌ها هم صدق می‌کند.

دستور mv که برای انتقال یا تغییر نام یک فایل یا دایرکتوری به کار می‌رود. مثلاً اگر دستور `mv myfile mydir1` را بدهیم، یعنی اگر فایل‌ی به نام myfile در مسیر جاری وجود دارد به مسیر myfile یا پوشه myfile که در مسیر جاری قرار

دارد انتقال بده.



/home/oracle

→ myfile

→ mydir1

\$ mv myfile mydir1

\$ ls mydir1

در این حالت فایل myfile از مسیر قبلی که قرار داشت پاک شده و به مسیر جدید منتقل می‌شود.

نکته: هنگام انتقال فایل‌های بزرگ از یک مسیر به مسیر دیگر دقت کنید، خصوصاً زمانی که این انتقال از یک ماشین به ماشین دیگر و تحت شبکه انجام می‌شود. از آنجاکه انتقال به معنی حذف فایل از مسیر جاری و انتقال به مسیر جدید است. اگر فایل بزرگ باشد و در حین انتقال برق برود یا به هر شکل ارتباط دو ماشین قطع شود فایل اولیه خراب خواهد شد و ممکن است به راحتی قابل دسترس نباشد، لذا در این گونه موارد ابتدا فایل را کپی کرده و سپس در مسیر اولیه آن را حذف می‌کنیم.

دستور mv همچنین برای تغییر نام فایل و دایرکتوری نیز به کار می‌رود. یعنی اگر دستور mv myfile newfile را اجرا کنیم اتفاقی که می‌افتد این است که فایل my-file به newfile تغییر نام می‌دهد.

دستور cp یا copy که برای کپی کردن فایل و دایرکتوری از یک مسیر به مسیر دیگر به کار می‌رود.

مثلاً cp myfile mydir1 یعنی فایلی با نام myfile را به شاخه mydir1 در مسیر جاری کپی کن.



با فرض اینکه فایلی با نام myfile در مسیر جاری وجود داشته باشد این فایل علاوه بر اینکه در مسیر جاری وجود دارد، بلکه یک کپی از آن در مسیر mydir1 نیز کپی می‌شود.

دستور rm یا remove که برای پاک کردن فایل و دایرکتوری‌ها به کار می‌رود. مثلاً اجرای دستور rm myfile یعنی اگر فایلی با نام myfile وجود دارد آن را پاک کن.

حالا هنگام پاک کردن ممکن است سیستم‌عامل از ما سؤال کند که آیا مطمئن هستی که فایل پاک شود و شما با دادن پاسخ بله یا خیر که با "y" یا "n" آن را مشخص می‌کنید.

اگر حرف 'y' را وارد کنید یعنی بله پاک کن.
اگر حرف 'n' را انتخاب کنید یعنی پاک نکن.

دستور rmdir یعنی یک دایرکتوری را پاک کن

مثلاً rmdir mydir1

در اجرای دستور بالا چنانچه mydir1 خالی نباشد سیستم با اعلان خطایی ما را متوجه می‌سازد که این پوشه خالی نیست. بنابراین اول باید فایل‌های داخل این مسیر را با دستور rm پاک کنیم. برای این کار می‌توانیم با دستور rm و نام هر فایل به ترتیب هر فایل را پاک نماییم، اما فرض کنید صد تا فایل در این مسیر دارید و باید صدبار این فرمان را وارد کنید که وقت‌گیر است. بنابراین از علائم مخصوص برای خلاصه کردن استفاده می‌کنیم چگونه؟

مثلاً اگر می‌خواهیم هر چه فایل در مسیر /home/oracle/mydir1 وجود دارد را پاک کنیم از دستور

\$ rm -f /home/oracle/myfile1



استفاده می‌کنیم و سپس می‌توانیم دستور `rm -rf /home/oracle/mydir1` را اجرا کنیم.

حالا زمان‌هایی پیش می‌آید که مطمئن به پاک کردن یک دایرکتوری و فایل‌های داخل آن هستیم، در این حالت از دستور `rm -rf /home/oracle/mydir1` استفاده می‌کنیم. در این حالت کل پوشه `mydir1` به همراه تمام فایل‌ها و پوشه‌های داخل آن به یک‌باره پاک خواهند شد. خب این دستور کار ما را راحت می‌کند اما خطری نیز ما را تهدید خواهد کرد و اینکه در بسیاری از موارد مشاهده کرده‌ام که کاربران به اشتباه مسیری را حذف می‌کنند که از اطلاعات آن مسیر نیز نسخه پشتیبانی ندارند و در این هنگام کلیه داده‌های ما در این پوشه از بین خواهد رفت؛ بنابراین بهتر است زمانی که می‌خواهیم اطلاعات یک پوشه را به‌طور کامل پاک کنیم حتماً از قبل برای آن نسخه پشتیبانی تهیه کنیم و سپس اقدام به حذف پوشه نماییم. زیرا در این حالت اگر اشتباهاً چیزی را حذف کرده باشیم قابل بازیابی می‌باشد. بدین منظور از دستور `cp -R mydir1 mydir1.backup` می‌توانیم استفاده کنیم. در این دستور گفته‌ام که از پوشه `mydir1` عیناً یک نسخه با نام `mydir1.backup` بساز. حالا که خیالم راحت شد که یک نسخه مثل نسخه قبل را دارم اگر با دستور `rm -rf /home/oracle/mydir1` تمام اطلاعات پوشه `mydir1` را پاک کنم نیز اشکالی پیش نخواهد آمد. دوستان به این نکته بسیار توجه کنید زیرا در بسیاری از موارد مشاهده کرده‌ام که افرادی که خصوصاً روی برنامه‌نویسی کار می‌کنند و کدهای یک ماه یا یک هفته‌ای خود را به‌صورت نسخه پشتیبان تهیه نکرده‌اند، گاهی با یک اشتباه ساده کل پوشه تغییرات را حذف می‌کنند و اگر نسخه پشتیبانی توسط مدیر سیستم وجود نداشته باشد یعنی کل تغییرات آن‌ها از بین رفته است.



نکته:

یک روش استفاده از دستورات لینوکس به صورت استفاده از علائم خاص می باشد. یعنی اگر من می خواهم تمام فایل هایی که انتهای نامشان مثلاً حروف txt دارد را حذف کنم از علامت "*" به جای کلیه نام فایل به همراه حروف txt استفاده می کنم. یعنی چی؟

فرض کنید فایل های با نام "file4.", "file3.txt", "file2.txt", "file1.txt" داریم و می خواهیم همه این فایل ها را حذف کنیم. در این صورت با دستور "rm *.txt" کلیه فایل هایی که اسم آن ها هر چه باشد ولی پسوند انتهای آن "txt." باشد را حذف خواهد کرد.

مثال بعدی:

فرض کنید فایل هایی با نام "4.cc.1", "3.cc", "2.cc", "cc" داریم در این صورت برای پاک کردن این فایل ها از دستور "rm *.cc" یعنی فایل هایی که نامشان یک حرفی است و با پسوند "cc" خاتمه پیدا می کند را پاک کن. علامت؟ به جای یک کاراکتر یا یک حرف به کار می رود.

مثال:

فرض کنید فایل هایی با نام "Ahamah", "ali" داریم و می خواهیم آن ها را به مسیری دیگر کپی کنیم برای این حالت می توانیم از دستور "cp [aA]* /home/oracle/mydir1" استفاده کنیم. این دستور می گوید از مسیر جاری هر فایلی که نام آن با حرف a کوچک یا A بزرگ شروع می شد و انتهای آن هر چه می خواهد باشد را به مسیر "home/oracle/mydir1" کپی کن.

توجه داشته باشید که کلیه موارد و حروف خاص مثل "*" یا "?" در کلیه دستورات ذکر شده بالا قابل استفاده می باشد و باعث می شود که عملیات حذف و انتقال و یا کپی کردن را به صورت دسته جمعی روی فایل ها و دایرکتوری هایمان انجام دهیم و در زمان



انجام صرفه‌جویی می‌کند؛ زیرا زمانی که شما با تعداد مثلاً هزار فایل در یک مسیر مواجه هستید و می‌خواهید این هزار فایل که مشخصه مشترکی دارند را کپی یا حذف یا تغییر مسیر دهید بهتر است از این علائم برای خلاصه کردن عمل خود و صرفه‌جویی در زمان استفاده نمایید.

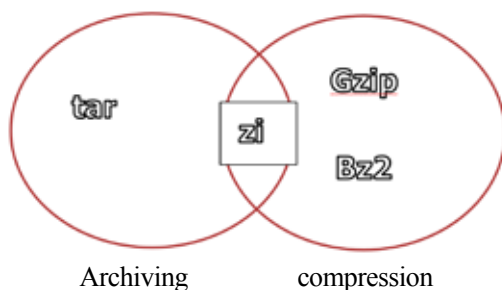
نکته: استفاده از گزینه “r-” به همراه دستورات mv , cp , rm به این معنی است که عملیاتی که قرار است انجام دهید را به صورت تودرتو در مسیر موردنظر و کلیه فایل‌ها و پوشه‌های داخل آن مسیر انجام بده.

یعنی اگر در مسیر `home/oracle/mydir1/mydir2/mydir3/` چندین پوشه و زیرپوشه وجود داشته باشد با استفاده از گزینه “r-” می‌توانیم به یک‌باره و تودرتو کلیه فایل‌ها و دایرکتوری مسیر `home/oracle/mydir1/mydir2/mydir3/` را حذف کرده یا انتقال دهیم یا کپی نماییم.

نحوه گرفتن نسخه پشتیبان از فایل ها و دایرکتوری های موجود در سیستم عامل لینوکس



LinuxTM



در شکل بالا دو دایره می‌بینیم، یکی برای گرفتن نسخه پشتیبان که در آن از دستور tar که مخفف کلمه Tape Archive می‌باشد و در دایره بعدی دستوراتی که برای فشرده‌سازی به کار می‌روند، مانند gzip, bz2 نشان داده شده است. در شکل بالا نقطه اشتراک میان این دو قسمت، دستور zip می‌باشد. در سیستم عامل لینوکس عموماً برای فشرده‌سازی از مکانیزم gzip, bz2 استفاده می‌شود و استفاده از دستور zip خیلی در سیستم عامل‌های لینوکسی باب نیست و یا کمتر استفاده می‌گردد. دستور tar که از دیرباز از سیستم‌های یونیکسی آمده است می‌تواند به همراه دستوراتی برای فشرده‌سازی نسخه پشتیبان مانند gzip, bz2 به کار رود.

آنچه در این فصل می‌خواهم به آن اشاره کنم عبارت است از استفاده از دستور tar برای تهیه نسخه پشتیبان یا backup و چگونگی استفاده از دستورات فشرده‌سازی gzip, bz2 برای فشرده کردن نسخه‌های پشتیبان.



به ازای هر دستور فشرده‌سازی، دستوری برای بازگردانی آن نسخه از حالت فشرده به حالت غیر فشرده نیز وجود دارد که در این فصل به تفصیل درباره آن صحبت خواهیم کرد.

دستور فشرده‌سازی	gzip	bz2	zip
دستور غیرفشرده‌سازی	gunzip	Bunzip2	unzip

در جدول بالا به اختصار نام هر دستور فشرده‌سازی به همراه دستوری که از حالت فشرده خارج می‌سازد آورده شده است.

اما برای تهیه نسخه پشتیبان از فایل‌ها و یا پوشه‌های داخل سیستم‌عامل می‌توانیم از دستور tar استفاده نماییم.

برای استفاده از دستور tar حالت‌های زیر را می‌توانیم به کار ببریم:

- برای ایجاد نسخه پشتیبان با استفاده از دستور `tar -c`
 - برای دیدن محتوای نسخه پشتیبان با استفاده از دستور `tar -t`
 - برای استخراج نسخه پشتیبان تهیه‌شده از دستور `tar -x` استفاده می‌کنیم.
- دستور tar را می‌توانیم با استفاده از گزینه‌های فشرده‌سازی نیز به کار ببریم. بدین منظور در هر یک از حالات سه‌گانه بالا می‌توانیم گزینه `-Z` یا `-j` را به کار ببریم. برای روشن شدن مطلب به مثال‌های زیر توجه کنید.

مثلاً فرض کنید که می‌خواهیم از مسیر خانه شخصی خود در لینوکس یا به اصطلاح `home directory` خود، یک نسخه پشتیبان تهیه کنیم. برای این کار از دستور زیر استفاده می‌نماییم:

```
$ tar -cf /tmp/home_oracle.tar /home/Oracle
```

در دستور بالا می‌گوییم با استفاده از گزینه‌های `-c` یعنی ایجاد نسخه پشتیبان و قرار دادن این نسخه پشتیبان در فایل در مسیر `tmp/` و با نام `home_oracle.tar` از



مسیر `home/Oracle` نسخه پشتیبان تهیه کن.

دقت کنید که در دستور `tar` ابتدا مقصد یا مکانی را که می‌خواهیم فایل پشتیبان در آنجا ذخیره گردد، می‌نویسیم و بدین منظور از گزینه `-f` در دستور `tar` استفاده کردیم. این گزینه می‌گوید که نسخه پشتیبان مرا در فایلی با مشخصه‌ای که در دستور آورده‌ام قرار بده.

دستور:

```
tar -cf /tmp/home_oracle.tar /home/Oracle
```

درواقع از کلیه فایل‌ها و دایرکتوری‌های مسیر `home/Oracle` یک نسخه پشتیبان در مسیر `/tmp` و با نام `home_oracle.tar` ایجاد می‌کند. دقت کنید که این نام یک نام دلخواه است و هر نام دیگری می‌تواند باشد، اما برای درک کاربر که بعدها زمانی که به سیستم مراجعه می‌کند و اینکه بداند این فایل چه بوده، بهتر است نام‌های انتخابی را با موضوع فایل نزدیک انتخاب نماییم؛ یعنی در لینوکس فرقی نمی‌کند که شما نام فایل را چه بگذارید اما چنانچه چند روز یا چند ماه بعد به سیستم مراجعه کردید برای اینکه با یک نگاه متوجه شوید که محتوای این فایل چه می‌تواند باشد، نام فایل‌ها در دستور `tar` را با پسوند `tar` انتخاب می‌کنیم.

اتفاقی که در دستور `tar` می‌افتد این است که کلیه فایل‌ها و دایرکتوری‌ها را به‌صورت یک فایل و با فرمت `tar` در مکانی ذخیره می‌کند. شکل زیر، نمایش دستور:

```
*~ tar -cf /tmp/oracle.tar
```

در این دستور به‌جای استفاده از مسیر `home/oracle` با فرض اینکه در مسیر `/home/oracle` قرار دارم و یا اینکه با نام کاربری `oracle` وارد سیستم شده‌ام از علامت `~` به معنی `home directory` استفاده می‌کنم.



```

oracle@localhost:~$
File Edit View Search Terminal Help
tar: removing leading '/' from member names
oracle@localhost ~$ tar -cf /tmp/home_oracle.tar.gz /home/oracle/
tar: removing leading '/' from member names
oracle@localhost ~$ tar -cf /tmp/home_oracle.tar.gz /
oracle@localhost ~$ ls -ltr /tmp
total 72
drwxr-xr-x. 9 root root 4096 Nov 11 2015 oracle-tools-distri
-rw-r--r--. 1 root root 6 Oct 8 23:55 yum.log
drwx-----. 2 root root 4096 Oct 9 08:19 pulse-gstreamer08fl
drwx-----. 2 root root 4096 Oct 9 08:19 virtual-root.308314
drwx-----. 2 oracle oracle 4096 Oct 14 09:12 virtual-oracle.M2+KZ
drwxr-xr-x. 2 root root 4096 Oct 14 09:16 virtual-oracle.M2+KZ
drwxrwxr-x. 2 root root 4096 Oct 14 09:17 oracle-config
-rw-r--r--. 1 root root 3439 Oct 14 09:18 oracle-log.txt.0
drwx-----. 2 root root 4096 Oct 14 09:18 oracle-root
drwx-----. 2 gdm gdm 4096 Oct 14 09:18 oracle-gdm
drwx-----. 2 oracle oracle 4096 Oct 14 09:18 oracle-origins
drwx-----. 2 oracle oracle 4096 Oct 14 09:18 oracle-oracle
drwx-----. 2 oracle oracle 4096 Oct 14 09:18 oracle-oracle
drwx-----. 2 gdm gdm 4096 Oct 14 09:18 oracle-oracle
drwx-----. 2 gdm gdm 4096 Oct 14 09:18 oracle-oracle
-rw-r--r--. 1 oracle oracle 10240 Oct 14 09:18 home_oracle.tar
oracle@localhost ~$

```

حالا فرض کنید که تعداد فایل‌ها و دایرکتوری‌های شما زیاد باشد و فایل پشتیبانی که با دستور tar می‌گیرید بزرگ بوده و فضای کافی برای ذخیره‌سازی نداشته باشید، در چنین حالتی با استفاده از گزینه‌های فشرده‌سازی، این نسخه tar را فشرده می‌کنیم تا حجم فایل تولیدشده کم گردد. بدین منظور از دستور tar با گزینه -z یا -j به صورت زیر استفاده می‌کنیم:

```
$ tar -czf /tmp/home_oracle.tar.gz /home/Oracle
```

در دستور بالا می‌گوییم که از مسیر home/oracle/ یک نسخه پشتیبان بگیر و آن را فشرده کن و در فایل به نام home_oracle.tar.gz در مسیر tmp/ قرار بده. دقت کنید که نام فایل را بر اساس گزینه -z به صورت home_oracle.tar.gz انتخاب کردیم که به من یادآوری می‌کند که این فایل از نوع tar فایل بوده و به صورت فشرده و از نوع gzip می‌باشد.

شکل زیر نحوه استفاده از گزینه فشرده‌سازی را در دستور tar نشان می‌دهد:



```
oracle@localhost:~$ tar -czf /tmp/oracle.tar.gz -/*
tar: Removing leading '/' from member names
[oracle@localhost ~]$ tar czvf oracle.tar.gz -/*
tar: Removing leading '/' from member names
/home/oracle/Desktop/
/home/oracle/Downloads/
/home/oracle/Music/
/home/oracle/Pictures/
/home/oracle/Public/
/home/oracle/Templates/
/home/oracle/Videos/
[oracle@localhost ~]$ ls -lft
total 35
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:28 Videos
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:28 Templates
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:28 Public
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:28 Pictures
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:28 Music
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:28 Downloads
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:28 Documents
drwxr-xr-x. 1 oracle oracle 244 Oct 14 09:58 oracle.tar.gz
[oracle@localhost ~]$
```

حال فرض کنید که می‌خواهیم از مکانیزم bz2 برای فشرده‌سازی استفاده کنیم. بدین منظور با استفاده از دستور زیر عمل می‌کنیم:

```
$ tar -cjf /tmp/home_oracle.tar.gz /home/oracle
```

حالا چنانچه هنگام گرفتن نسخه پشتیبان بخواهیم محتوای مسیری که در حال پشتیبان‌گیری است روی صفحه مانیتور نمایش داده شود از گزینه -v استفاده می‌کنیم. با استفاده از این گزینه کلیه فایل‌ها و دایرکتوری‌های مسیر موردنظر به‌صورت هم‌زمان که در فایل مقصد پشتیبان‌گیری می‌شوند روی صفحه نمایش داده می‌شوند.

بدین منظور از دستور زیر استفاده می‌کنیم:

```
$ tar -czvf /tmp/home_oracle.tar.gz /home/oracle
```

و یا

```
$ tar -cjvf /tmp/home_oracle.tar.bz2 /home/oracle
```



```
oracle@localhost:~$ tar -czvf /tmp/oracle.tar.gz /home/oracle
tar: Removing leading '/' from member names
/home/oracle/
/home/oracle/.cache/
/home/oracle/.cache/event-sound-cache.tds,46e200123d120608e6de360300000013.a86.0
4-unknown-linux-gnu
/home/oracle/.bash_logout
/home/oracle/.zshrc
/home/oracle/.spice-vdagent/
/home/oracle/.spice-vdagent/log
/home/oracle/.recently-used.xbel
/home/oracle/.ssh/
/home/oracle/.bash_history
/home/oracle/.ICAuthority
/home/oracle/.ssh/authorized_keys
/home/oracle/Pictures/
/home/oracle/Videos/
/home/oracle/Templates/
/home/oracle/.mozilla/
/home/oracle/.mozilla/extensions/
/home/oracle/.mozilla/plugins/
/home/oracle/Downloads/
```

جهت نمایش محتوای یک tar file از گزینه t- استفاده می کنیم. فرض کنید که در دستور بالا نسخه پشتیبانی تهیه کرده ایم حالا می خواهیم ابتدا محتوای آن را ببینیم و بعد به استخراج آن پردازیم. بدین منظور از دستور زیر استفاده می کنیم:

```
$ tar -tzvf /tmp/home_oracle.tar.gz
```

```
$ tar -tjvf /tmp/home_oracle.tar.bz2
```

با استفاده از دستورات بالا می توانیم محتوای فایل پشتیبانی که با فرمت های gzip, bz2 را که قبلاً ایجاد کرده بودیم، تماشا کنیم.

حالا فرض کنید که می خواهیم این نسخه های پشتیبان را استخراج کنیم.

نکته: در هنگام استخراج یک نسخه پشتیبان، حتماً در ابتدا به مکانی غیر از مکان اصلی استخراج بروید. مثلاً فرض کنید که من می خواهم فایل home_oracle.tar.gz را استخراج کنم بنابراین در ابتدا به مسیر home/oracle نمی روم که در آنجا این استخراج را انجام دهم، زیرا اگر این کار را انجام دهم کلیه فایل های نسخه پشتیبان روی فایل های قبلی نوشته می شود و کلیه فایل های قبلی از بین خواهند رفت. بنابراین در ابتدا یک پوشه مثلاً با نام دلخواه mybackup در مسیری دلخواه حالا یا در مسیر /home/oracle یا در مسیری مثل tmp/ می سازم و سپس نسخه پشتیبان را در آنجا استخراج می کنم و پس از اطمینان از صحت فایل های استخراج شده می توانم این فایل ها



را در مسیر دلخواهم کپی نمایم و یا پس از اطمینان از صحت آنچه این فایل پشتیبان در اختیارم گذاشته است به مسیر اصلی رفته و در آنجا این عمل استخراج را انجام می‌دهم.

برای استخراج فایل‌های پشتیبان از گزینه `x-` استفاده می‌کنیم:

```
$ tar -xzf /tmp/home_oracle.tar.gz
```

```
$ tar -xjf /tmp/home_oracle.tar.bz2
```

در دو دستور بالا فرض بر این است که فایل پشتیبان از نوع فشرده بوده و در مسیر `/tmp` و با نام `home_oracle.tar.gz` یا `home_oracle.tar.bz2` قرار دارد. حالا می‌توانیم یک پوشه با نام `mybackup` ساخته و به داخل آن رفته و یکی از دستورات بالا را اجرا نماییم. بنابراین به‌صورت زیر عمل می‌کنیم:

```
$ mkdir /tmp/mybackup
```

```
$ cd /tmp/mybacku
```

```
$ tar -xzf /tmp/home_oracle.tar.gz
```

ویژگی دستور `tar` در این است که `owner` و `group` و اجازه‌های دسترسی فایل‌ها را در خود ذخیره می‌کند. این مسئله به‌شدت در زمان استخراج نسخه پشتیبان مهم می‌باشد؛ زیرا در زمان استخراج فایل‌ها و دایرکتوری‌ها اگر مثلاً از دستور کپی استفاده کنیم `group`, `owner` فایل‌ها بر اساس نام و گروه شخصی که در حال انجام استخراج است تغییر می‌کند و همچنین اجازه‌های دسترسی فایل‌ها نیز بر اساس اجازه‌های دسترسی مسیری که در آن عمل استخراج انجام می‌شود تغییر خواهد کرد و گاهی تغییر یک اجازه دسترسی روی یک فایل و یا تغییر نام کاربری و گروه فایل منجر به اجرا نشدن آن فایل می‌گردد. لذا با استفاده از دستور `tar` هم می‌توانیم `owner`, `group`, فایل و هم اجازه‌های دسترسی فایل‌ها را ذخیره و در هنگام استخراج عیناً به‌صورت اول بازیابی نماییم.



مکانیزم خاصی دارد که می‌تواند خصوصیتی از فایل را که در آن owner, group و اجازه‌های دسترسی فایل را نیز در خود ذخیره کند.

یکی دیگر از روش‌های فشرده‌سازی استفاده از دستور Zip می‌باشد. مثلاً فرض کنید می‌خواهیم یک فایل را به‌صورت فشرده و با مکانیزم Zip فشرده نماییم. بدین منظور از دستور `zip file.zip file` استفاده می‌کنیم. حالا فرض کنید که می‌خواهیم یک پوشه یا دایرکتوری را فشرده نماییم. در این حالت باید بدانیم که دستور Zip به‌تنهایی قادر به انجام عمل فشرده‌سازی به‌صورت تودرتو نمی‌باشد. به عبارتی وقتی که از دستور `zip dir.zip dir_name` می‌زنیم در این حالت دستور Zip متوجه نمی‌شود که `dir_name` یک دایرکتوری است که باید محتوای داخل آن را فشرده کند و بنابراین عملاً کاری انجام نمی‌دهد. بدین منظور از گزینه `-r` به همراه دستور Zip استفاده می‌نماییم که به‌صورت زیر نمایش داده‌ام:

`$ zip -r dir.zip dir_name`

```
oracle@localhost:~  
File Edit View Search Terminal Help  
[oracle@localhost ~]$ mkdir mydir  
[oracle@localhost ~]$ cd mydir  
[oracle@localhost mydir]$ touch file1 file2 file3  
[oracle@localhost mydir]$ cd ..  
[oracle@localhost ~]$ ls -lrt  
total 36  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Videos  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Templates  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Public  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Pictures  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Music  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Downloads  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Documents  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Desktop  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 10:23 mydir  
[oracle@localhost ~]$ zip -r mydir.zip mydir  
adding: mydir/ (stored 0%)  
adding: mydir/file3 (stored 0%)  
adding: mydir/file1 (stored 0%)  
adding: mydir/file2 (stored 0%)  
[oracle@localhost ~]$ ls -l  
total 40  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Desktop  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Documents
```

در شکل بالا یک دایرکتوری با نام mydir ساختم و در داخل آن سه فایل با



نام‌های file1, file2, file3 ایجاد کردم و سپس با دستور `zip -r mydir` این دایرکتوری و محتوای داخل آن را فشرده کرده‌ام، اما چون فایل‌های من خالی می‌باشد لذا مقدار ۰٪ را در شکل بالا مشاهده می‌کنیم. اما اگر فایل‌ها و دایرکتوری‌های مسیر انتخابی خالی نباشند این میزان درصد نیز متفاوت خواهد بود و صرفاً مثال بالا برای چگونگی استفاده از دستور zip می‌باشد.

برای گرفتن نسخه پشتیبان با گزینه `f-` می‌توانیم بگوییم که این نسخه پشتیبان را در روی `tape` نگهداری کن. بدین منظور باید نام `tape device` متناظر را روی سیستم‌عامل لینوکس بدانیم. مثلاً `tape device`‌های از نوع اسکاژی معمولاً با نام `st0` یا `scsi tape` نامیده می‌شوند که عدد ۰ به اولین `device` اشاره می‌کند. بنابراین اگر بخواهیم که محتوای فایل پشتیبان را روی `tape` منتقل کنیم دستور زیر را وارد می‌کنیم:

```
# tar -czf /dev/st0 /home/oracle
```

در دستور بالا می‌گوییم که یک فایل پشتیبان بساز روی `device` به نام `dev/st0` و از کلیه محتوای مسیر `home/oracle/` و در ضمن فشرده هم کردیم. `dev/st0/` به مسیر `tape device` روی سیستم‌عامل اشاره می‌کند. از آنجاکه در سیستم‌عامل لینوکس کلیه `device`‌های جانبی در مسیر `dev/` قرار دارند و نیز به خاطر اینکه `tape` نیز یک وسیله خارجی متصل به سیستم‌عامل است لذا نام آن در این مسیر نگهداری می‌شود. به عبارتی سیستم‌عامل با دیدن نام `dev/st0/` به دنبال `tape device`ی از نوع اسکاژی می‌گردد و اگر این دستگاه به ماشین لینوکس ما متصل باشد فایل پشتیبان را روی آن منتقل می‌کند.

در مورد تهیه نسخه پشتیبان روی `tape` با گزینه‌های متفاوتی روبرو هستیم. مثلاً می‌دانیم در این روزها سایز و فضای `tape device`‌ها زیاد است و به ترابایت می‌رسد بنابراین اگر می‌خواهیم نسخه پشتیبانی روی `tape` تهیه کنیم که حجم آن کمتر از



حجم tape می‌باشد بهتر است از فضای tape بهینه استفاده نماییم. در این حالت از گزینه `tar -czf /dev/st0n /hom/oracle` استفاده می‌کنیم. دقت کنید /dev/st0n به سیستم‌عامل می‌گوید که نسخه پشتیبانی تهیه کن که بعد از تهیه این نسخه tape به عقب و به اول نوار برنگردد. حرف n مخفف non rewined می‌باشد. در حالتی که از /dev/st0n استفاده کنیم بعد از عمل پشتیبان‌گیری tape به عقب و به ابتدا بازمی‌گردد و سرخط می‌ایستد اما در حالتی که از گزینه /dev/st0n استفاده کنیم مثل این است که نوار را بخش‌بندی کرده‌ایم و هر قسمت از نسخه پشتیبان را روی یک بخش از نوار مغناطیسی ذخیره نموده‌ایم. در حالت استفاده از st0n بعد از تهیه نسخه پشتیبان؛ هد نوار مغناطیسی درست بعد از آخرین داده پشتیبان قرار گرفته و چنانچه بخواهیم نسخه پشتیبان دیگری روی نوار قرار دهیم از این نقطه به بعد اطلاعات ما در نوار ذخیره می‌گردد.

از آنجاکه این کتاب مربوط به دانستنی‌های لینوکس مقدماتی می‌باشد لذا در این خصوص بیشتر صحبت نمی‌کنم و صرفاً خواستم تا نسبت به گزینه‌های مختلف پشتیبان‌گیری آشنایی پیدا نمایید. استفاده از دستورات نوار مغناطیسی، خود بخش مجزایی را طلب می‌کند.

چگونه داخل فایل ها
و دایرکتوری های
لینوکس جستجو کرده
و متن مورد نظر را
بیابیم؟

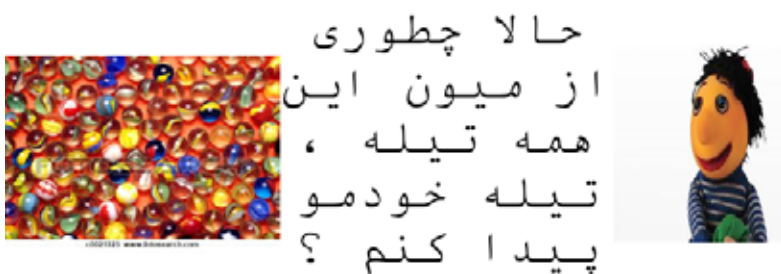


LinuxTM



با داستانی موضوع را شروع می‌کنم

فرض کنید که کلاه قرمزی یک ظرف پر از تیلۀ دارد و تیلۀ خود را در میان این تیلۀها گم کرده است و حالا فکر می‌کند که چگونه تیلۀ خود را از میان این همه تیلۀ پیدا کند.



به‌راستی چگونه می‌توانیم به کلاه قرمزی کمک کنیم تا تیلۀ خودش را از میان این همه تیلۀ پیدا کند؟

خب ممکن است بگوییم می‌توانیم تیلۀها را بر اساس اندازه مرتب کنیم و سپس بر اساس رنگ دسته‌بندی کنیم و این‌طوری دسته‌های کوچک‌تری پیدا کرده و بنابراین پیدا کردن تیلۀ کلاه قرمزی در این دسته‌های کوچک‌تر راحت‌تر خواهد بود. بدین منظور در لینوکس دستور بسیار قدرتمندی به‌نام `grep` داریم که به‌صورت زیر از آن استفاده می‌کنیم:

”تیلۀ من ” ”همه تیلۀها“ `$ grep`



جستجو و نام کلیه فایل‌هایی که می‌خواهیم این عبارت در آن جستجو گردد را وارد می‌کنیم. بنابراین مثلاً اگر بخواهیم به دنبال عبارت “abc” در داخل فایل file1.txt بگردیم از دستور `grep “abc” file1.txt` استفاده می‌کنیم.

فرض کنید چندین فایل داریم و نمی‌دانیم که عبارت موردنظر ما در کدام فایل است. بنابراین عبارت موردنظر را در همه فایل‌ها جستجو می‌کنیم و این عمل را با دستور `grep “abc” file*.txt` انجام می‌دهیم. در این مثال می‌خواهیم به دنبال عبارت “abc” در کلیه فایل‌هایی که نام آن‌ها با کلمه file شروع شده و ادامه آن هر حرفی می‌خواهد باشد و انتهای آن با کلمه.txt پایان یافته باشد، بگردیم.

اما روش‌های دیگر جستجو این است که مانند مثال بالا که برای کلاه قرمزی استفاده کردیم ابتدا محتوای یک فایل را مرتب کنیم و بر اساس مقادیر مرتب‌شده بر اساس مثلاً رنگ تیل یا اندازه تیل‌ها به دنبال تیل موردنظر بگردیم. این عمل در سیستم عامل لینوکس نیز قابل انجام است و با استفاده از ترکیب چند دستور با هم که به آن عمل `pipng` یا لوله‌کشی می‌گویند انجام می‌شود. درواقع در مثال کلاه قرمزی مثل اینکه ما از چند سبد سوراخ‌دار با اندازه‌های متفاوت برای جداسازی تیل‌های درشت از تیل‌های کوچک استفاده کنیم.



استفاده از pipe برای فیلتر کردن اطلاعات

خب حالا میام این
تیل‌ها رو از یک
صافی رد میکنم



مورا از آبکش دومی که
رد کردم همون تیل
هایی که میخواستم رو
پتیداشون کردم





عمل pipe یا لوله‌کشی در لینوکس باعث می‌شود که خروجی اجرای یک دستور به‌عنوان ورودی دستور بعدی به‌کار رود. دقیقاً مثل زمانی که کل تپله‌های کلاه قرمزی را از یک آبکش اولیه عبور دادیم و نتیجه تپله‌های به‌دست‌آمده را از آبکش بعدی عبور دادیم.

در استفاده از دستور pipe در لینوکس در هنگام جستجوی یک عبارت در میان محتوای یک فایل می‌توانیم به روش زیر عمل کنیم که ابتدا محتوای فایل را با دستور cat نمایش داده و این محتوا را از دستور grep عبور دهیم تا عبارت موردنظر را پیدا نماییم.

بدین منظور به روش زیر عمل می‌کنیم:

```
$ cat file\text | grep abc
```

Command line pipe



در شکل بالا خروجی دستور cat را به دستور more داده‌ایم که بتوانیم محتوای فایل را صفحه به صفحه مشاهده کنیم.

این‌ها روش‌هایی از pipe کردن دستورات در لینوکس می‌باشند. می‌خواهم در این بخش مجموعه‌ای از دستورات مفید لینوکس جهت جستجوی متنی در فایل یا فایل‌های موردنظر را به‌صورت اختصار توضیح دهم.



دستور tail

این دستور برای نمایش خطوط انتهایی یک فایل به کار می‌رود و جالب است بدانید که این دستور به همراه گزینه `-f` جهت مانیتورینگ لاگ فایل‌های سیستم بسیار مفید است و نحوه استفاده آن به صورت می‌باشد.

مثلاً اگر بخواهیم یکی از لاگ فایل‌های مهم سیستم عامل لینوکس به نام `mes-sages` را که در مسیر `/var/log/` قرار دارد، مانیتور کنیم از دستور:

```
tail -f /var/log/messages
```

استفاده می‌کنیم.

اگر در دستور `tail` تعداد خطوطی را که می‌خواهیم روی صفحه نمایش ببینیم، ندهیم سیستم عامل به طور پیش فرض ده خط آخر فایل را نمایش می‌دهد، اما می‌توانیم تعداد خطوط مورد نیاز را هم در این دستور ارائه کنیم. مثلاً بگوییم `tail -n 50 /var/log/messages` یعنی ۵۰ خط آخر فایل `/var/log/messages/` را برایم نمایش بده.

دستور head

این دستور برعکس دستور `tail` از ابتدای فایل تعدادی خطوط را نمایش می‌دهد. چنانچه هیچ عددی در این دستور ندهیم، به طور پیش فرض ۱۰ خط ابتدای فایل را نمایش می‌دهد و چنانچه عدد بدهیم به تعداد عدد داده شده از خطوط ابتدای فایل را به ما نمایش می‌دهد.

مثلاً `tail -n 20 /var/log/messages` یعنی ۲۰ خط اول این فایل را برایم نمایش بده.

دستور less

`Less` دستوری است که باعث می‌شود خروجی یک فرمان به صورت صفحه به



صفحه‌نمایش داده شود. علاوه بر آن، این امکان را به ما می‌دهد که بتوانیم روی صفحات مرور کنیم، یعنی به صفحه بالایی و پایینی حرکت کنیم.

مثلاً اگر می‌خواهیم محتوای فایل messages در مسیر `var/log/` را مشاهده کنیم با دستور `less /var/log/messages` این کار را به راحتی می‌توانیم انجام دهیم.

دستور find

این دستور مکان مستقر یک فایل را در سیستم عامل لینوکس نمایش می‌دهد. مثلاً اگر می‌خواهیم به دنبال فایلی بگردیم که اسم آن را کامل نمی‌دانیم اما می‌دانیم این فایل در انتها با پسوند `tar.gz` تمام شده ولی نمی‌دانیم کجای سیستم ذخیره شده است، با استفاده از دستور `find` آن را جستجو می‌کنیم.

مثلاً می‌نویسیم `find / -name "*.tar.gz"`

در این دستور، مسیری را که می‌خواهیم به دنبال فایل یا فایل‌های موردنظر بگردیم، وارد کرده‌ام که در مثال بالا علامت `/` یعنی از ریشه اصلی به دنبال فایل موردنظر بگرد و به دنبال تمام فایل‌هایی باش که نام آن‌ها شامل عبارت `"tar.gz"` می‌باشد.

در این حالت سیستم عامل از ریشه شروع به جستجوی فایل یا فایل‌هایی با ویژگی موردنظر ما می‌گردد و در صورت پیدا کردن نام، مسیر فایل را روی صفحه‌نمایش می‌دهد.

دستور sort

`Sort` برای مرتب‌سازی محتوای یک فایل به کار می‌رود و دقیقاً مثل حالتی که کلاه قرمزی می‌خواهد تپله‌های موردنظر خود را در میان انبار تپله‌ها پیدا کند می‌توانیم این تپله‌ها را بر اساس اندازه یا رنگ مرتب کنیم که این عمل یک نوع مرتب‌سازی می‌باشد. دستور `sort` دقیقاً عمل مرتب‌سازی را روی محتوای فایل‌ها انجام می‌دهد. مثلاً وقتی می‌گوییم `sort myfile` در این حالت فایل `myfile` بر اساس حروف الفبا از `a-z`



مرتب‌شده و روی صفحه نمایش داده می‌شود.

دستور cut

این دستور برای قیچی کردن تکه‌ای از محتوای یک فایل به کار می‌رود و قابلیت‌های زیادی را در اختیار ما قرار می‌دهد. مثلاً فرض کنید که از یک فایل داده‌ای، خصوصاً در فایل‌های مربوط به ذخیره‌سازی پایگاه‌های اطلاعاتی، ما با انبوهی از اطلاعات به صورت ستون به ستون روبرو هستیم و حالا فرض کنید که ما ستون یا دسته خاصی از این داده‌ها را نیاز داریم. در این زمان با استفاده از دستور cut می‌توانیم ستون یا ستون‌های موردنیاز را از فایل کنده و در جای دیگر ذخیره نماییم. در دستور cut دو گزینه مهم داریم، یکی جداکننده ستون‌ها که مشخصاً می‌گوییم از چه علامتی برای جداسازی ستون‌ها استفاده کرده‌ایم و یکی شماره ستون یا ستون‌هایی که می‌خواهیم قیچی کنیم. مثلاً فرض کنید فایل `etc/passwd` در سیستم‌عامل را که محل ذخیره‌سازی نام و مشخصات کاربران سیستم می‌باشد، می‌خواهیم به صورتی قیچی کنیم که فقط نام کاربران از آن پاره شده و این نام‌ها در فایل دیگری ذخیره گردد. بدین منظور به صورت زیر عمل می‌کنیم:

اولاً قسمتی از محتوای یک فایل `etc/passwd` را در زیر نمایش می‌دهم تا دقیقاً ببینیم که چه علامتی بین ستون‌ها جداکننده است:



```

crack@localhost:~$ cat /etc/passwd | grep nologin
nobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
naildaemon:x:68:68:daemon:/:/sbin/nologin
pulse:x:494:493:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
gdm:x:42:42:/var/lib/gdm:/sbin/nologin
tomcat8:x:91:91:Apache Tomcat:/usr/share/tomcat8:/sbin/nologin
stap-server:x:155:155:Systemtap Console Server:/lib/stap-server:/sbin/nologin
xshd:x:74:74:Privilege-separated SSH:/var/empty/xshd:/sbin/nologin
quagga:x:92:92:quagga routing suite:/var/run/quagga:/sbin/nologin
mailman:x:41:41:GNU Mailman List Manager:/usr/lib/mailman:/sbin/nologin
webalizer:x:67:67:webalizer:/var/www/usage:/sbin/nologin
arpwatch:x:77:77:/var/lib/arpwatch:/sbin/nologin
libert:x:88:98:/:/sbin/nologin
quid:x:493:487:UID generator: helger:daemon:/var/lib/libuid:/sbin/nologin
nslcd:x:45:55:NAP Client User:/:/sbin/nologin
ldap:x:55:55:LDAP User:/var/lib/ldap:/sbin/nologin
cyrus:x:78:12:Cyrus IMAP Server:/var/lib/imap:/sbin/nologin
radius:x:75:75:radius user:/:/sbin/nologin
squid:x:12:12:/var/spool/squid:/sbin/nologin
memcached:x:482:482:Memcached daemon:/var/run/memcached:/sbin/nologin
tcpdump:x:72:72:/:/sbin/nologin
radius:x:55:35:radiusd user:/home/radiusd:/sbin/nologin
crack:x:560:561:/home/crack:/bin/bash
(END)

```

در شکل بالا نمایی از محتویات فایل `/etc/passwd` را می بینیم که هر ستون با علامت ":" از ستون بعدی جدا شده است. ستون اول نام کاربر را می باشد. بنابراین اگر می خواهیم ستون اول را از این فایل قیچی کرده و محتوا را در فایلی مثلاً به نام `users` ذخیره کنیم، با دستور زیر این عمل انجام می شود:

```
$ cut -d":" -f1 /etc/passwd > /tmp/usres
```

در فرمان بالا می گوئیم که از فایل `/etc/passwd` فیلد ۱ را که با `-f1` نشان داده ایم جدا کند و خروجی آن را در فایلی به نام `tmp/usders` قرار دهد که این عمل را با علامت "<" انجام می دهیم. علامت ">" به معنی این است که خروجی یک دستور را به فایلی دیگر وارد یا ذخیره کن.

حالا در فایل `/etc/passwd` اگر من ستون های اول و سوم و چهارم را بخواهم چه دستوری را اجرا می کنم؟

```
$ cut -d":" -f۱,۳,۴ /etc/passwd
```

در این حالت خروجی اجرای فرمان بالا به صورت زیر خواهد بود:



```

orville@localhost:~$ cat /etc/passwd
profile:16:16:
rcs:89:89:
rtkt:468:497:
slrt:173:173:
apache:48:48:
ssslauth:457:76:
postfix:89:89:
dovecot:97:97:
dovecot1:496:496:
avahi:76:76:
nssd:495:495:
mysql:27:27:
nail:11:47:47:
nmap:51:51:
amandbackup:33:6:
ntp:38:38:
dhcpd:177:177:
tss:59:59:
rpsuser:129:129:
nfsnobody:65534:65534:
haldaemon:68:68:
pulse:494:493:
gdm:42:42:

```

دستور wc

این دستور مخفف کلمه word count می‌باشد، یعنی تعداد کلمات. با استفاده از دستور wc می‌توانیم تعداد کلمات یا تعداد حروف یا تعداد خطوط یک فایل را بشماریم. این دستور گزینه‌های مختلفی دارد. مثلاً اگر بگوییم wc myfile در این حالت دستور wc، هم تعداد حروف، هم تعداد کلمات و هم تعداد خطوط فایل myfile را به ما نشان می‌دهد، اما اگر فقط تعداد خطوط را نیاز داشته باشیم می‌توانیم از گزینه -l به معنی Line فقط تعداد خطوط فایل را در خروجی نمایش دهیم. بدین منظور از دستور wc -l myfile استفاده می‌کنیم تا تعداد خطوط یک فایل را ببابیم.

در جستجوی عبارات موردنظر در داخل فایل‌ها یکسری علائم داریم که برای سیستم‌عامل معنی‌دار هستند که این علائم عبارت‌اند از:

- \wedge این علامت به معنی ابتدای فایل می‌باشد. مثلاً وقتی می‌گوییم برو داخل فایل و به دنبال تمام عباراتی باش که در ابتدای آن مثلاً حرف a قرار دارد و آن‌ها را نشان بده در این صورت با دستور `grep ^a myfile` این عمل را انجام می‌دهیم.

- \$ این علامت برای نمایش انتهای فایل است و زمانی که به دنبال متنی هستیم که



مثلاً در انتهای فایل یک حرف به خصوصی دارد می توانیم از این علامت استفاده کنیم. مثلاً می توانیم بگوییم برو داخل فایل و به دنبال تمام عباراتی باش که انتهای آن ها با txt تمام می شود و آن ها را پیدا کن که با دستور `grep $txt myfile` می توانیم این جستجو را انجام دهیم.

چند مثال خوب

۱- فرض کنید که در سیستم به دنبال لاگ فایل های بزرگی که مثلاً سایز آن ها از ۱۰ مگابایت بیشتر بوده و بیشتر از یک ماه است که در سیستم دست نخورده باقی مانده و نام آن ها با پسوند log تمام می شود، می گردیم. در این حالت با استفاده از فرمان `find` و گزینه های آن، این کار را انجام می دهیم.

```
$ find / -atime ۳۰ -size ۱۰M -name "*.log"
```

با مطالعه `man find` یا `Info find` می توانید گزینه های مختلف دستور `find` را بیابید.

۲- فرض کنید به دنبال فایل های فقط خواندنی یا `read only` سیستم هستیم. با دستور زیر این فایل ها را جستجو می کنیم.

```
# find / -perm /u=r
```

۳- فرض کنید به دنبال پوشه های خالی در مسیر `tmp/` می گردیم.

```
# find /tmp -type d -empty
```

۴- فرض کنید که در مسیر `home/oracle/` به دنبال فایل های مخفی می گردیم.

```
# find /home/oracle -type f -name ".*"
```

۵- لیست فایل هایی را می خواهیم که بین ۵۰ تا ۱۰۰ روز پیش تغییر کرده اند:

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

۶- لیست فایل هایی را که یک ساعت قبل تغییر کرده اند، می خواهیم:

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



۷- به دنبال تمام فایل‌هایی هستیم که مالک آن‌ها کاربر oracle باشد:

```
#find / -user oracle
```

۸- لیست تمام فایل‌هایی را که سایز آن‌ها بین ۵۰ تا ۱۰۰ مگابایت است، می‌خواهیم:

```
#find / -size +50M -size -100M
```

۹- به دنبال عبارت “php” در فایل‌های مختلف می‌گردیم:

```
# grep php -r /
```

۱۰- به دنبال فایل‌هایی با پسوند php در سیستم می‌گردیم:

```
# find / -type f -name "*.php"
```

داده‌های ما
در سیستم عامل
در کجا ذخیره
می‌شوند؟



LinuxTM



در این بخش می‌خواهیم به این موضوع بپردازیم که داده‌های ما در سیستم‌عامل لینوکس در کجا نگهداری می‌شوند. بدین منظور بهتر است طرحی از ساختار پوشه‌های سیستم‌عامل لینوکس را توضیح دهیم.

مسیر / etc

در سیستم‌عامل لینوکس، مسیری با نام /etc داریم که محل ذخیره فایل‌های پیکربندی سیستم‌عامل می‌باشند. در سیستم‌عامل لینوکس فایل‌هایی هستند که اطلاعاتی در ارتباط با نام ماشین و یا اطلاعات شبکه را در خود نگهداری می‌کنند. این اطلاعات برای سیستم‌عامل بسیار مهم می‌باشند و هنگامی که سیستم روشن می‌شود به این اطلاعات نیازمند است. بیشتر این اطلاعات در مسیر /etc نگهداری می‌شوند.

مسیر /lib و /usr/lib و /usr/lib64 و /lib64

این مسیرها محل نگهداری فایل‌های کتابخانه‌ای یا library های سیستم‌عامل می‌باشند. فایل‌های کتابخانه‌ای درواقع کدهای نوشته‌شده و اجراشده‌ای هستند که فایل‌های اجرایی سیستم به آن‌ها نیاز دارند. در سیستم‌های کنونی که از معماری ۶۴



بیتی استفاده می کنند علاوه بر /usr/lib/ , lib مسیر /lib۶۴/ , /usr/lib۶۴/ نیز اضافه شده است. درواقع /usr/lib/ , lib محل نگهداری library های ۳۲ بیتی و /lib۶۴/ , /usr/lib۶۴/ محل نگهداری library های ۶۴ بیتی سیستم می باشد. فایل هایی که در این مسیر قرار می گیرند خیلی شبیه به فایل های dll در ویندوز می باشند.

مسیر /usr/bin/ bin

این مسیر محل نگهداری فایل های اجرایی سیستم، مانند دستورات مورد استفاده سیستم مثل دستور wc , find , ls و غیره می باشند.

مسیر /var/log/

در مسیر /var/ کلیه فایل هایی که سائز آن ها در طول حیات و روشن بودن سیستم تغییر می کند، نگهداری می شود. مانند لاگ فایل های سیستم. از جمله این لاگ فایل ها در مسیر /var/log/ می باشد که محل نگهداری لاگ فایل های سیستم عامل می باشد. مثلاً فایل مهم messages در مسیر /var/log/ در این مسیر نگهداری می شود.

مسیر /home/

محل نگهداری فایل های مربوط به کاربران موجود در سیستم عامل که قادر به وارد شدن به سیستم هستند، می باشد.

مثلاً کاربر oracle که اجازه ورود به سیستم را دارد، دارای خانه شخصی /home/ oracle می باشد که می تواند فایل های خود را در آنجا ذخیره و نگهداری کند.

بنابراین متوجه می شویم که داده های ما در سطح سیستم عامل بر طبق نظمی خاص



و بر اساس موضوعی که آن داده در خود ذخیره می کند در مکانی خاص در سیستم عامل نگهداری می شود.

مفهومی به نام process

زمانی که ما فرمان ls را اجرا می کنیم در واقع یک فایل اجرایی را اجرا می کنیم. این فایل اجرایی یا همان ls یک کپی از فایل ls می باشد که کاربر در لحظه مورد نیاز آن را به کار می گیرد. حالا اگر هم زمان ۱۰ نفر این فرمان را اجرا نمایند انگار ۱۰ نسخه کپی از فرمان ls در اختیار این ده نفر قرار گرفته است. این نسخه های کپی از برنامه اجرایی را که در لحظه اجرا می شوند process می گویند. بنابراین process چیزی بیش از یک برنامه اجرایی نیست. منتها نسخه ای از یک برنامه اجرایی ای می باشد که در لحظه در حال استفاده شدن است و به عبارتی در حافظه سیستم بارگذاری شده است.

بنابراین یک process برنامه اجرایی ای می باشد که در حافظه قرار گرفته است. کاربران سیستم های عامل لینوکسی می توانند هم زمان چندین دستور را یا یک دستور را به صورت چندین مرتبه اجرا نمایند. لذا سیستم عامل لینوکس حاوی پاره ای از دستورات جهت مدیریت و نمایش process ها می باشد. از جمله این دستورات، دستور ps یا process status یا همان نشان دادن وضعیت process ها می باشد.

فرمان ps

نمایش process های در حال اجرا، با اجرای فرمان ps بدون هیچ گزینه ای لیستی از پردازش های کاربر مورد نظر که لاگین کرده است نمایش داده می شود. مانند شکل زیر:

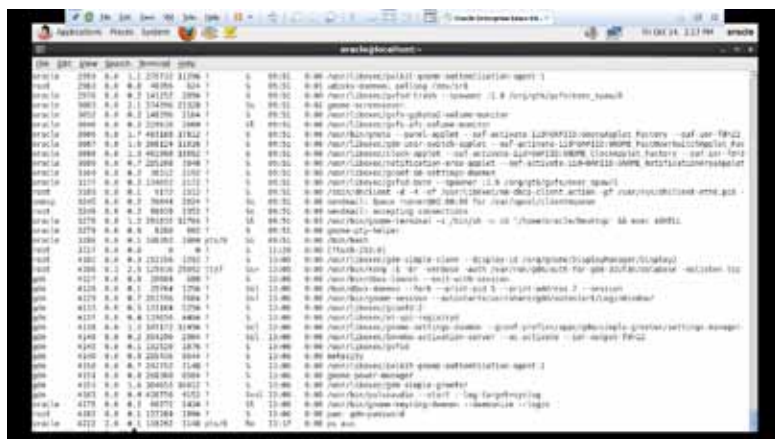
```
oracle@localhost:~$ ps
  PID TTY          TIME CMD
 3280 pts/0    00:00:00 bash
 4200 pts/0    00:00:00 ps
oracle@localhost:~$
```




همان گونه که در شکل بالا مشاهده می کنید خروجی دستور `ps aux` بسیار طولانی می باشد ولی اگر بخواهیم از این لیست مثلاً ببینیم آیا سرویس پست الکترونیکی ما مثل postfix در حال اجرا هست یا خیر، می توانیم با استفاده از عمل Pipe دو دستور `ps` و `grep` را به خوبی برای رسیدن به منظور خود استفاده نماییم. بدین منظور کافی است این فرمان را اجرا نماییم:

در این حالت در لیست پردازش‌های سیستم، فقط آن‌هایی نمایش داده می‌شوند که عبارت postfix در آن‌ها یافت شود.

فرمانی که برای مانیتورینگ process ها به کار می‌رود اطلاعات جامعی از نام





پردازش‌ها و میزان فضای اشغالی و بر اساس اولویت میزان حافظه و یا میزان پردازنده‌ای که برای آن پردازش در سیستم مصرف می‌گردد، این لیست نمایش داده می‌شود.

دستور top خروجی‌ای شبیه به شکل زیر دارد:

```

oracle@localhost:~$ top
top - 13:06:13 up 3:17, 2 users, load average: 0.07, 0.02, 0.00
Tasks: 171 total, 1 running, 170 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.3%us, 0.5%sy, 0.0%ni, 99.1%id, 0.0%wa, 0.0%hi, 0.1%st, 0.0%st
Mem: 1001688k total, 749312k used, 252376k free, 40456k buffers
Swap: 2031608k total, 0k used, 2031608k free, 341236k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU  MEM     TIME+ COMMAND
 4104 oracle    20   0 13004 1160 840  R 3.7   0.1   0:00.00 top
3249 root      20   0 90028 2352 630  S 1.0   0.2   0:00.52 sendmail
   1 root      20   0 19396 1560 1232  S 0.0   0.2   0:01.02 init
   2 root      20   0   0     0     0  S 0.0   0.0   0:00.00 kthreadd
   3 root      RT  0   0     0     0  S 0.0   0.0   0:00.00 migration/0
   4 root      20   0   0     0     0  S 0.0   0.0   0:00.03 ksoftirqd/0
   5 root      RT  0   0     0     0  S 0.0   0.0   0:00.00 watchdog/0
   6 root      20   0   0     0     0  S 0.0   0.0   0:00.71 events/0
   7 root      20   0   0     0     0  S 0.0   0.0   0:00.00 cpuset
   8 root      20   0   0     0     0  S 0.0   0.0   0:00.00 khelper
   9 root      20   0   0     0     0  S 0.0   0.0   0:00.00 netns
  10 root      20   0   0     0     0  S 0.0   0.0   0:00.00 async/mgr
  11 root      20   0   0     0     0  S 0.0   0.0   0:00.02 sync supers
  12 root      20   0   0     0     0  S 0.0   0.0   0:00.03 bdi-default
  13 root      20   0   0     0     0  S 0.0   0.0   0:00.00 kintegrityd/0
  14 root      20   0   0     0     0  S 0.0   0.0   0:00.04 kblockd/0
  15 root      20   0   0     0     0  S 0.0   0.0   0:00.00 kacpid
  
```

دستور free

این دستور میزان حافظه روی سیستم و میزان فضای آزاد را نشان می‌دهد. با استفاده از گزینه -m می‌توانیم میزان فضای آزاد را بر اساس مگابایت مشاهده کنیم. مانند شکل زیر:

```

oracle@localhost:~$ free -m
              total        used        free      shared    buffers     cached
Mem:           978         735          243           0          39         333
-/+ buffers/cache:         362          615          1983
Swap:          1983           0          1983
  
```


انواع کاربران و مفاهیمی از امنیت اولیه در سطح سیستم عامل لینوکس در ارتباط با کاربران



LinuxTM



در این فصل قصد داریم توضیحاتی در مورد انواع کاربرانی که در سطح سیستم عامل لینوکس وجود دارند صحبت کنم و بعد در مورد حق دسترسی هر گروه از این کاربران صحبت خواهیم کرد.

به طور کلی در سطح سیستم عامل سه نوع کاربر وجود دارد:

۱- کاربران استاندارد standard

۲- کاربر root

۳- کاربر سیستمی system

۱- کاربران استاندارد سیستم عامل لینوکس چه کسانی هستند؟

این کاربران شامل کاربرانی می شود که مدیر سیستم آن ها، بعد از روشن شدن سیستم، ایجاد می کند که می توانند شامل تمام افرادی که قرار است با این سیستم عامل کار کنند، باشد. مثلاً کاربرانی که قرار است با پایگاه داده ای اوراکل کار کنند از این نوع کاربران هستند. یا کاربرانی که قرار است در یک پروژه خاص کار کنند نیز از نوع کاربران استاندارد سیستم می باشند.

این کاربران می توانند قدرت مدیر سیستم یا root را دارا بوده و یا فاقد آن باشند.

۲- کاربر root

این کاربر، کاربر ارشد سیستم عامل یا مدیر ارشد سیستم عامل می باشد که اجازه هر



کاری را در سطح سیستم عامل دارا می باشد.

۳- کاربر سیستمی

این کاربران برای اجرای یک برنامه خاص در سطح سیستم عامل تعریف می شوند. به عنوان مثال فرض کنید که قرار است در ماشین لینوکس شما، یک سرویس وب اجرا گردد. بنابراین سیستم عامل نیاز به کاربری برای اجرای این سرویس دارد که مثلاً با نام apache این کاربر را جهت اجرای سرویس وب تخصیص می دهد. در واقع این کاربر نیاز به ورود به سیستم به صورتی که بتواند از دستورات خط فرمان لینوکس استفاده کند ندارد، بلکه فقط به این منظور ساخته شده است تا سرویس وب را اجرا نماید. یا مثلاً اگر قرار است پایگاه داده ای mysql را روی سیستم خود اجرا نمایید نیاز به کاربری با نام mysql دارید تا این سرویس را اجرا نماید. پس کاربران سیستمی، کاربرانی هستند که در سطح سیستم عامل لینوکس قرار است سرویسی را اجرا نمایند.

ابزارهای لازم برای مشاهده نوع کاربر در سیستم عامل لینوکس

دستور id

دستور id اطلاعاتی در مورد شخصی که به سیستم وارد یا به اصطلاح لاگین کرده است می دهد. با اجرای این فرمان خروجی ای به شکل زیر نمایش داده می شود:

```
oracle@localhost:~$ id
uid=500(oracle) gid=501(oracle) groups=501(oracle) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0c1:c0c2
oracle@localhost ~$
```

مثلاً در شکل بالا با اجرای فرمان id روی صفحه اطلاعاتی از کاربری با نام oracle و عدد کاربری او در سطح سیستم عامل یعنی عدد ۵۰۰ و نام گروهی که این کاربر عضو آن



است که گروه oracle با عدد ۵۰۱ و اطلاعات دیگر در مورد این کاربر را به ما می‌دهد.

دستور w

این دستور می‌گوید چه دستورات در حال اجرا توسط شخصی که لاگین کرده است در سیستم وجود دارد.

```
oracle@localhost:~$ w
14:02:26 up 4:13, 2 users, load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@   IDLE   XCPU  PCPU WHAT
oracle    tty1     :0             09:51    4:13m  14.86s  0.07s pam: gdm-password
oracle    pts/0    :0.0           09:51    0.00s  0.15s  0.01s w
[oracle@localhost ~]$
```

دستور who

دستور who می‌گوید در حال حاضر چه کسانی وارد سیستم شده‌اند و از طریق چه ترمینالی به سیستم متصل هستند.

```
[oracle@localhost ~]$ who
oracle    tty1                2016-10-14 09:51 (:0)
oracle    pts/0               2016-10-14 09:51 (:0.0)
[oracle@localhost ~]$
```

دستور sudo

با استفاده از دستور sudo می‌توانیم اجازه‌های کاربر root سیستم را داشته و دستوراتی را با اجازه کاربر root اجرا نماییم.

فایل‌های مرتبط با کاربران

۱- فایل etc/passwd/



root را در خود ذخیره می‌کند. اطلاعات داخل این فایل شامل نام کاربران و عدد واحدی که برای آن کاربر در سیستم در نظر گرفته شده و نام گروهی که کاربر در آن عضو است به همراه عدد گروه و نیز توضیح مختصری از این کاربر که قرار است چه کاری انجام دهد و مسیر خانه شخصی این کاربر یا home directory این کاربر چه هست، می‌باشد. همچنین ستون دوم این فایل شامل کلمه عبور کاربران می‌باشد که به لحاظ امنیت با حرف "x" نشان داده شده است تا کسی نتواند به کلمه عبور کاربران پی ببرد. نمونه فایل `/etc/passwd`

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/etc:/sbin/nologin
adbcd:x:3:3:adbcd:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
lpc:x:5:5:lpc:/sbin:/sbin/nologin
nfsnobody:x:6:6:nfsnobody:/var/lib/nfs:/sbin/nologin
mail:x:8:8:mail:/var/spool/mail:/sbin/nologin
nscd:x:9:9:nscd:/var/spool/nscd:/sbin/nologin
nmap:x:10:10:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:11:11:nmap:/var/spool/nmap:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:ftp:/var/ftp:/sbin/nologin
nobody:x:16:16:nobody:/var/lib/nobody:/sbin/nologin
dbus:x:17:81:system message bus:/var/lib/dbus:/sbin/nologin
nmap:x:20:20:nmap:/var/spool/nmap:/sbin/nologin
lpd:x:22:22:lpd:/var/spool/lpd:/sbin/nologin
nmap:x:23:23:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:24:24:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:25:25:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:26:26:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:27:27:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:28:28:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:29:29:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:30:30:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:31:31:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:32:32:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:33:33:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:34:34:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:35:35:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:36:36:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:37:37:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:38:38:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:39:39:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:40:40:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:41:41:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:42:42:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:43:43:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:44:44:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:45:45:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:46:46:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:47:47:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:48:48:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:49:49:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:50:50:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:51:51:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:52:52:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:53:53:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:54:54:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:55:55:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:56:56:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:57:57:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:58:58:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:59:59:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:60:60:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:61:61:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:62:62:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:63:63:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:64:64:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:65:65:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:66:66:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:67:67:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:68:68:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:69:69:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:70:70:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:71:71:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:72:72:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:73:73:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:74:74:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:75:75:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:76:76:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:77:77:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:78:78:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:79:79:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:80:80:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:81:81:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:82:82:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:83:83:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:84:84:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:85:85:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:86:86:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:87:87:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:88:88:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:89:89:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:90:90:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:91:91:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:92:92:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:93:93:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:94:94:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:95:95:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:96:96:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:97:97:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:98:98:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:99:99:nmap:/var/spool/nmap:/sbin/nologin
nmap:x:100:100:nmap:/var/spool/nmap:/sbin/nologin
```

۲- فایل `/etc/shadow`

این فایل شامل اطلاعاتی از کاربران می‌باشد که ستون اول آن دقیقاً مثل ستون اول فایل `/etc/passwd` شامل نام کاربران سیستم و ستون دوم آن رمز عبور کاربران است که به صورت رمزگذاری شده در این فایل قرار دارد، تا کسی پی به ماهیت آن نبرد.

نمونه‌ای از فایل `/etc/shadow`



```
File Edit View Search Terminal Help
[oracle@localhost ~]$ cat /etc/group
#groupname:password:gid:members
#
# This file is the default system group file.  It contains
# the information used by the system to map user names
# to group names.  It is the master file for the system.
#
# The format of the file is:
#
# groupname:password:gid:members
#
# groupname is the name of the group.
# password is the password for the group.
# gid is the group ID number.
# members is a list of the users who are members of the group.
```

از جمله اطلاعات دیگری که در این فایل ذخیره شده است در مورد طول عمر کلمه عبور، اینکه تا چه زمانی فرصت دارند کلمه عبور خود را تغییر دهند. اینکه چه کاربری قفل یا Lock شده است و اجازه ورود ندارد.

فایل `/etc/group`

این فایل اطلاعاتی در مورد گروه‌های موجود در سیستم را ذخیره می‌کند. مثلاً با اجرای فرمان `groups` متوجه می‌شویم که کاربر `oracle` به چه گروهی متعلق است.

```
File Edit View Search Terminal Help
[oracle@localhost ~]$ groups
oracle
[oracle@localhost ~]$
```

با اجرای فرمان `less /etc/group` می‌توانیم محتوای این فایل را صفحه به صفحه مشاهده کنیم

[illegible]

برای استفاده از دستور sudo برای کاربری که root نیست باید حتماً این کاربر جزء کاربرانی که حق استفاده از دستور sudo را دارند تعریف شده باشد وگرنه سیستم عامل اجازه استفاده از این قدرت به هر کاربری نمی‌دهد. این عمل عموماً در فایل `/etc/sudoers` انجام می‌شود که فایلی با پیکربندی مخصوص خود می‌باشد که جهت تعریف دسته‌ای از دستورات و کاربرانی می‌باشد که قرار است قدرت کاربری دیگر را به ارث ببرند.

از آنجا که این بخش توضیحات مقدماتی لینوکس است بنابراین کاربران را با مفاهیم خیلی تخصصی نمی‌خواهیم سردرگم نمایم.

در دوره‌های LPIC-1 در مورد این فایل و نحوه استفاده از آن شرح داده می‌شود.

ایجاد کاربران و گروه‌های کاربری



LinuxTM



دستوراتی که در این قسمت شرح خواهیم داد عبارتند از:

- Useradd
- Groupadd
- Id
- Last
- Passwd

دستور useradd

این دستور که همگی با حروف کوچک نوشته می‌شود برای ایجاد یک کاربر جدید در سیستم مورد استفاده قرار می‌گیرد. برای ایجاد کاربر جدید حتماً باید root بوده و یا با استفاده از sudo اجازه این کار به کاربری دیگر داده شده باشد.

برای ایجاد نام کاربر جدید دستور زیر را اجرا می‌کنیم:

```
#useradd oracle
```

مثلاً در دستور بالا کاربری با نام oracle در سیستم ایجاد کرده‌ام. چنانچه این کاربر قبلاً وجود داشته باشد که سیستم عامل این خطا را روی صفحه ظاهر می‌کند و اگر وجود نداشته باش این کاربر جدید در سیستم با یک عدد منحصر به فرد ایجاد می‌شود.

در هر سیستم عاملی از لینوکس، عددی خاص برای شروع ایجاد کاربران استاندارد سیستم در نظر گرفته می‌شود. مثلاً در بعضی از نسخه‌های لینوکس عدد ۵۰۰ به بعد



برای ایجاد کاربران استاندارد در نظر گرفته می‌شود. بنابراین چنانچه در هنگام ایجاد کاربر جدید عددی را به سیستم‌عامل معرفی نکنیم این سیستم‌عامل است که این عدد را بر اساس پیش‌فرض‌های خود در نظر می‌گیرد و مثلاً از ۵۰۰ به بعد اگر عددی وجود داشته باشد که هنوز به کاربری تخصیص داده نشده به این کاربر جدید تخصیص داده می‌شود.

همچنین در هنگام ایجاد کاربر جدید بافرمان `useradd` سیستم‌عامل یک گروه اختصاصی برای این کاربر با نام کاربری او در نظر می‌گیرد و یک عدد واحد را برای این گروه اختصاصی به این کاربر تخصیص می‌دهد. مثلاً در دستور بالا با اجرای فرمان `useradd oracle` اگر این کاربر، اولین کاربر استاندارد سیستم باشد `uid` این کاربر معادل ۵۰۰ و `gid` این کاربر مثلاً با همان ۵۰۰ می‌باشد. البته اگر عدد ۵۰۰ قبلاً برای هیچ گروه دیگری تخصیص نیافته باشد.

پس در ابتدای اجرای فرمان `useradd oracle` این اتفاقات رخ می‌دهد:

- ۱- کاربری با نام `oracle` در سطح سیستم‌عامل ایجاد می‌شود.
- ۲- یک عدد واحد به نام `uid` بر اساس پیش‌فرض سیستم‌عامل مثلاً از عدد ۵۰۰ به بالا به این کاربر تخصیص داده می‌شود.
- ۳- یک گروه مخصوص این کاربر با نام `oracle` در سطح سیستم به وجود می‌آید.
- ۴- یک `gid` یا عدد واحد بر اساس پیش‌فرض‌های سیستم‌عامل برای این کاربر تخصیص داده می‌شود.
- ۵- پوشه‌ای با نام این کاربر (در اینجا `oracle`) در مسیر `/home` ایجاد شده و مجموعه‌ای از حق دسترسی‌ها به این کاربر در این مسیر داده می‌شود تا بتواند فایل‌های خود را در این مسیر ذخیره و مدیریت نماید.
- ۶- یک `shell` یا مترجمی که بتواند دستورات کاربر را برای سیستم‌عامل ترجمه کند در اختیار کاربر قرار داده می‌شود. به‌طور پیش‌فرض این `shell` از نوع `bash` می‌باشد که



نوعی مفسر یا ترجمه کننده دستورات برای سیستم عامل لینوکس است.

در این میان چنانچه هنگام ایجاد کاربری جدید بخواهیم این کاربر حتماً از یک عدد مخصوص برای uid و gid خود استفاده نماید لازم است که این اعداد را در زمان اجرای فرمان به سیستم عامل معرفی کنیم.

مثلاً

```
#useradd -u 5000 -g 5000 -d /var/vmail -s /sbin/nologin  
vmail
```

در دستور بالا کاربری با نام vmail ایجاد کرده ایم که uid آن برابر ۵۰۰۰ و gid او برابر ۵۰۰۰ و مسیر خانه شخصی این کاربر در /var/vmail و مفسر یا shell این کاربر nologin تعریف شده است. زمانی که برای کاربری shell را از نوع nologin یا false انتخاب می نماییم یعنی این کاربر از نوع کاربرانی است که قرار است سرویسی را در سطح سیستم عامل ارائه دهند و نیاز به گرفتن shell برای اجرای دستورات ندارند. در دستور بالا تمامی پارامترها را از حالت پیش فرض خارج کرده و هنگام ایجاد این پارامترها را به صورت دستی تعیین کرده ایم.

یا مثلاً اگر می خواهیم کاربری با نام oracle ایجاد کنیم که قرار است در گروه oinstall عضو باشد ابتدا گروهی به نام oinstall ایجاد کرده و سپس کاربر oracle را به صورت زیر ایجاد می کنیم:

```
#useradd -g oinstall oracle
```

دستور groupadd

این دستور برای ایجاد یک گروه جدید به کار می رود که همانند دستور useradd می توانیم از پیش فرض سیستم عامل استفاده کرده و بگذاریم سیستم عامل عدد مورد نظر را به گروه بدهد یا اینکه در ابتدای ایجاد عدد را خودمان وارد نماییم.



مثلاً:

groupadd oinstall یا groupadd -g 1001 oinstall

۳- استفاده از دستور id

دستور id به ما می گوید که این کاربر متعلق به چه گروهی یا گروه‌هایی می باشد و uid, gid کاربر را نیز نمایش می دهد مثلاً:

```
# id oracle
```

به ما نشان می دهد که کاربر oracle در چه گروه یا گروه‌هایی عضو است و مقدار uid, gid این کاربر در سیستم چه می باشد.

دستور last

این دستور می گوید که آخرین بار در چه ساعت و تاریخی چه کسانی به سیستم وارد شده اند.

اجرای آن به صورت زیر می باشد:

```
oracle@localhost:~$ last
oracle pts/0 :0.0 Fri Oct 14 09:51 still logged in
oracle tty1 :0 Fri Oct 14 09:51 still logged in
reboot system boot 2.6.32-300.3.1.e Fri Oct 14 09:48 - 14:46 (04:57)
oracle pts/0 :0.0 Fri Oct 14 08:20 down (01:27)
oracle tty1 :0 Fri Oct 14 08:20 down (01:27)
reboot system boot 2.6.32-300.3.1.e Fri Oct 14 08:20 - 09:48 (01:27)
root pts/0 :0.0 Sun Oct 9 00:19 down (00:00)
root tty1 :0 Sun Oct 9 00:19 down (00:00)
reboot system boot 2.6.32-300.3.1.e Sun Oct 9 00:17 - 00:20 (00:02)

wtmp begins Sun Oct 9 00:17:35 2016
oracle@localhost ~$
```

- در راستای دستوراتی برای ایجاد کاربر جدید یا گروه کاربری جدید، دستوراتی برای حذف و تغییر کاربر و گروه کاربری نیز داریم که عبارتند از:

Usedel حذف کاربر



Grouped حذف گروه

Usermod تغییر کاربر

Goupmod تغییر گروه

نکته:

کلید این دستورات با حروف کوچک در سیستم عامل لینوکس استفاده می شود.

دستور userdel برای حذف یک کاربر به کار می رود زمانی که این دستور را با گزینه

“-f” به کار ببریم در سطح سیستم عامل حذف می کنیم چه اتفاقاتی می افتد؟

۱- نام کاربری او از فایل etc/passwd حذف می گردد.

۲- نام کاربر از گروهی که متعلق به آن است حذف می شود.

۳- مسیر home directory کاربر حذف می گردد بنابراین کلید داده های ذخیره شده

در این مسیر پاک خواهند شد.

۴- کاربر قادر به ورود به سیستم نخواهد بود.

۵- هیچ سرویسی با نام این کاربر قادر به اجرا نخواهد بود.

بنابراین زمان حذف کاربر بهتر است به موارد بالا توجه داشته باشیم و شاید لازم باشد

نسخه پشتیبانی از مسیر home directory این کاربر در جایی خاص داشته باشیم و

بعد اقدام به حذف او کنیم.

دستور usermod برای انجام پاره ای از تغییرات در نام یا shell یا uid یا gid یا

home directory کاربر به کار می رود.

مثلاً کاربری با نام oracle داریم و می خواهیم علاوه بر اینکه عضو گروه oinstall

است، عضو گروهی به نام grid نیز باشد با استفاده از دستور:

```
# groupmod -a -G grid oracle
```



می‌توانیم تغییری در کاربر oracle دهیم که عضو گروه grid نیز باشد.

دستور passwd

این دستور برای تغییر کلمه رمز کاربری به کار می‌رود و هر کاربری حق دارد کلمه رمز خود را تغییر دهد.

البته در سطح کل سیستم این کاربر root است که قادر است کلمه رمز کلیه کاربران را تغییر دهد.

به عنوان مثال، اگر من به عنوان کاربر oracle می‌خواهم رمز خود را تغییر دهم کافی است در سطح shell دستور passwd را وارد کنم. در این صورت سیستم عامل در ابتدا کلمه رمزی را که در حال حاضر در سیستم استفاده می‌کنم، می‌پرسد و اگر این کلمه را درست وارد کرده باشیم اجازه می‌دهد کلمه رمز جدید را برای خود انتخاب نماییم و دو بار این کلمه را سؤال می‌کند و سپس تغییرات در فایل /etc/shadow اعمال می‌گردد.

نحوه اجازه دسترسی به کاربران



LinuxTM



در این فصل قصد داریم مختصری در مورد اجازه‌های دسترسی فایل‌ها و پوشه‌هایی که در سطح سیستم عامل می‌سازیم و همچنین مدیریت فایل‌ها و دایرکتوری‌ها در لینوکس بحث کنیم.

در ابتدا فرض کنید با کاربر oracle وارد سیستم شده‌ایم و دستور `ls -l` را که لیستی کامل از کلیه فایل‌ها و پوشه‌هایمان، نمایش می‌دهد، نشان می‌دهیم:

```
oracle@localhost:~  
File Edit View Search Terminal Help  
[oracle@localhost ~]$ ls -  
ls: cannot access -: No such file or directory  
[oracle@localhost ~]$ ls -l  
total 40  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Desktop  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Documents  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Downloads  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Music  
drwxrwxr-x. 2 oracle oracle 4096 Oct 14 10:23 mydir  
-rw-rw-r--. 1 oracle oracle 612 Oct 14 10:24 mydir.zip  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Pictures  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Public  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Templates  
drwxr-xr-x. 2 oracle oracle 4096 Oct 14 08:20 Videos  
[oracle@localhost ~]$
```

همان‌گونه که در شکل بالا مشاهده می‌کنید لیست فایل‌ها و پوشه‌های مربوط به کاربر oracle لیست شده است. اطلاعاتی که در صفحه مشاهده می‌کنید به‌قرار زیر است:

- ستون اول که به‌صورت `drwxr-xr-x` و یا `-rw-rwr-` نمایش داده شده است چه چیزی را نشان می‌دهد؟

- ستونی که `drwxr-xr-x` را دارا می‌باشد می‌گوید آنچه مشاهده می‌کنید یک



پوشه یا دایرکتوری است و حرف "d" مخفف directory می‌باشد. همچنین این عبارت می‌گوید که این پوشه دارای مجوز read به‌طور اختصار "r" و write یا خواندن که به‌طور اختصار با "w" و همچنین مجوز اجرای یا execute که به‌طور اختصار با حرف "x" نمایش داده شده است برای خود کاربر oracle می‌باشد.

پس rwx در اولین ستون اجازه‌های دسترسی کاربر را روی این پوشه مشخص می‌کند که می‌گوید کاربر oracle اجازه دارد وارد این پوشه شود و یا این پوشه و محتویاتش را حذف کند یا تغییر دهد و یا لیستی از فایل‌ها و پوشه‌های داخل این مسیر بگیرد.

اما r-x ستون بعدی می‌گوید که گروهی که این کاربر به آن متعلق است قادر است محتویات این پوشه را بخواند یا فرمان ls را روی این پوشه اعمال کند و همچنین قادر است روی پوشه قدرت اجرا داشته باشد. قدرت اجرا یا execute در سطح پوشه، یعنی گروهی که کاربر به آن متعلق است قادر است با فرمان cd وارد این پوشه شود و لیست فایل‌ها را مشاهده نماید؛ اما از آنجاکه به جای حرف "w" حرف "-" در این قسمت گذاشته شده یعنی گروه این کاربر قادر به تغییر این فایل‌ها در این پوشه نمی‌باشد.

اما r-x در قسمت آخر این عبارت می‌گوید افرادی به جز کاربر و گروهی که کاربر به آن متعلق است یعنی باقی کاربران سیستم چه اجازه‌ها و حقوق دسترسی به این پوشه را دارند. در اینجا می‌گوید دیگران در سطح سیستم اجازه خواندن یا read و اجازه اجرا یا execute را روی این پوشه دارند؛ یعنی در سطح سیستم عامل باقی کاربران می‌توانند داخل این پوشه با دستور cd روند و یا دستور ls برای لیست کردن فایل‌های داخل پوشه را اجرا نمایند اما قادر به تغییر این فایل‌ها یا نام پوشه یا محتویات آن نیستند.

تا اینجا کار با مبانی امنیت در لینوکس در سطح فایل و پوشه کمی آشنایی پیدا کردیم.

اما ستون سوم از دستور "ls -l" می‌گوید که مالک این فایل و پوشه‌ها چه کسی



است و ستون چهارم می گوید گروهی که این کاربر به آن متعلق است چه گروهی است و ستون پنجم سائز فایل یا پوشه را ذکر می کند. ستون های ۶ و ۷ تاریخ و زمان ایجاد فایل یا پوشه را نمایش می دهند و ستون آخر نام فایل یا پوشه را نشان می دهد.

پس متوجه شدیم که با یک دستور ساده "ls-l" در یک مسیر خاص می توانیم اطلاعات امنیتی اولیه مربوط به فایل ها و پوشه های موجود در آن مسیر را پیدا کنیم. در این قسمت می خواهیم در مورد اجازه های دسترسی که در بالا صحبت کردیم بیشتر توضیح دهیم.

گفتیم در لینوکس چه برای کاربر چه برای گروهی که کاربر در آن عضو است و چه برای باقی کاربران یکسری اجازه های دسترسی در نظر گرفته می شود. شرح این موضوع به این صورت است که سه دسته اجازه در سطح سیستم عامل داریم

۱- کاربر که می تواند اجازه خواندن، نوشتن و یا اجرا را در سطح فایل یا پوشه داشته باشد. این اجازه ها را به صورت read, write, execute و با حروف اختصاری "rwx" نمایش می دهیم؛ اما از آنجاکه سیستم عامل دارای زبان ماشین است و ماشین، عدد می فهمد، لذا این حروف به اعداد تبدیل می شوند و این اعداد غالباً در مبنای ۲ می باشند. اما چگونه؟

در سیستم عامل اجازه های دسترسی از راست به چپ به این صورت لحاظ می شود که اولین ستون سمت راست با عدد ۲ به توان صفر در نظر گرفته می شود. $1 = 2^0$

ستون دوم می شود $2 = 2^1$

ستون سوم $4 = 2^2$

این محاسبه برای اجازه های دسترسی در سطح گروه و دیگران نیز به همین صورت می باشد، یعنی ۱ و ۲ و ۴.

بنابراین اگر فایل یا پوشه ای دارای همه مجوزهای خواندن، نوشتن و اجرا باشد یعنی مجموع اعداد ۴ و ۲ و ۱ را داراست که مجموع این ارقام یعنی $4 + 2 + 1 = 7$ می باشد؛



بنابراین اگر فایل یا پوشه‌ای مجوز عددی ۷ را دارا باشد یعنی هر کاری روی این فایل می‌شود انجام دهد یا به عبارتی کلیه مجوزها را دارا می‌باشد.

حالا اگر فایلی اجازه خواندن و اجرا داشته باشد اما اجازه نوشتن نداشته باشد مجوز عددی آن چگونه می‌شود؟

خب فایل اجازه خواندن دارد و گفتیم از سمت چپ آخرین مجوز مربوط به خواندن است که در جایگاه $2^4=4$ قرار دارد. پس فایلی که مجوز خواندن دارد از لحاظ عددی مقدار ۴ را خواهد داشت.

همچنین فایل مجوز نوشتن ندارد. مجوز نوشتن دومین مجوز و یا در مکان $2^1=2$ است؛ اما وقتی فایل مجوز نوشتن ندارد یعنی این مجوز صفر است. پس مقدار ۰ را برای نوشتن در نظر می‌گیریم.

و فایل اجازه اجرا دارد که این اجازه از سمت راست اولین مکان یا $2^0=1$ را شامل می‌شود.

بنابراین فایل ما در سطح کاربر دارای مجوز ۴ برای خواندن و ۰ برای نوشتن و ۱ برای اجرا می‌باشد که مجموع این ارقام یعنی $4+0+1=5$ مقدار عددی مجوز این فایل در سطح کاربر را نمایش می‌دهد.

اگر این فایل در سطح گروه هم همین مجوزها را داشته باشد بنابراین در سطح گروه هم عدد ۵ را دارا می‌باشد.

حالا اگر این فایل برای باقی کاربران فقط مجوز خواندن داشته باشد مقدار آنچه می‌شود؟

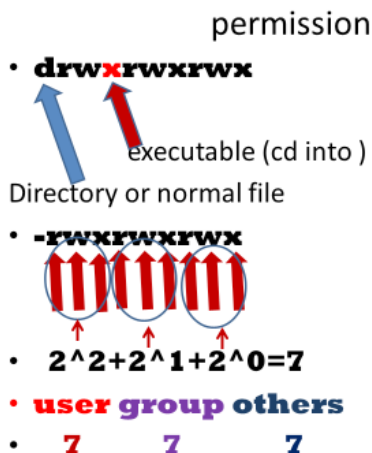
مجوز خواندن یا read یعنی $2^4=4$ و چون مجوزهای دیگر را دارا نیست یعنی مقدار ۰ را برای نوشتن و اجرا در نظر می‌گیریم. بنابراین مجموع مجوزهای آن می‌شود $4+0+0=4$.

در اینجا می‌توانیم بگوییم که این فایل دارای مجوز ۷ برای کاربر و ۵ برای گروه و ۴



برای دیگران می باشد. یعنی عدد ۷۵۴ که به صورت متنی عبارت "rwxr-xr-" را برای فایل مشاهده خواهیم کرد.

شکل زیر گویای این شرح می باشد:



اما شناخت این اعداد برای ما چه سودی دارد؟

زمانی که ما می خواهیم مجوزهای یک فایل را وارد کنیم دو راه در پیش رو داریم:

- ۱- به صورت مطلق این مجوزها را به فایل بدهیم.
- ۲- به صورت نسبی این مجوزها را به فایل بدهیم.

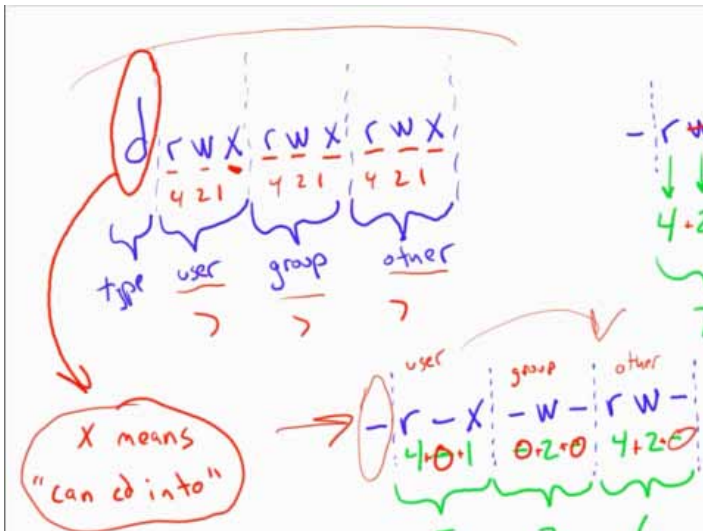
۱- حالت مطلق

در حالت مطلق ما کاری به اینکه در حال حاضر مجوزهای فایل یا پوشه چه هستند نداریم و به یک باره مجوز موردنظر را اعمال می کنیم. یعنی اگر قبلاً اجازه خواندن و نوشتن و اجرا را در سطح کاربر یا گروه یا دیگران داشته یا نداشته برای ما مهم نیست ما می خواهیم از الآن به بعد مثلاً مجوز فقط خواندن و نوشتن برای کاربر و فقط خواندن برای گروه و فقط اجرا برای دیگران داشته باشد.



۲- حالت نسبی

در این حالت ما مجوزهایی را به فایل یا پوشه نسبت به مجوزهایی فعلی آن می‌دهیم؛ یعنی می‌گوییم اگر فایل مجوز خواندن و اجرا دارد، داشته باشد، حالا مجوز نوشتن را به آن اضافه کن. یا به مجوز گروه و کاربران دیگر دست نزن و فقط مجوز کاربر را تغییر بده. برای تغییر مجوز روی فایل و پوشه‌ها از دستور `chmod` استفاده می‌کنیم. بدین صورت که می‌توانیم از حالت مطلق به صورت `chmod 755` روی یک فایل استفاده کنیم که در این حالت می‌گوییم کلیه مجوزهای خواندن و نوشتن و اجرا را به کاربر یا مالک فایل بده و فقط مجوز خواندن و اجرا را به گروه و دیگران اعطا کن. می‌توانیم با استفاده از دستور `chmod g-w` بگوییم که مجوز نوشتن را از گروه بگیر. در این حالت کاری به باقی مجوزها نداریم. مثلاً کاری نداریم که مجوز روی افراد غیر از گروه چه هست و فقط می‌گوییم مجوز نوشتن را از گروه بگیر یعنی به صورت نسبی تغییر اجازه دسترسی می‌دهیم. در شکل زیر مثال دیگری آورده شده است:





مثلاً وقتی می‌گوییم “`chmod go-w`” یعنی مجوز نوشتن یا تغییر دادن فایل را از گروه و دیگران سلب کرده‌ایم اما به باقی مجوزها کاری نداریم.
یا وقتی می‌گوییم “`chmod ugo-w`” یعنی از کاربر و گروه و دیگران مجوز نوشتن را بگیر.

علامت “-” یعنی گرفتن یک اجازه
و علامت “+” یعنی دادن یک اجازه
مثلاً دستور “`chmod go+x`” یعنی به گروه و افراد غیر از گروه اجازه اجرای فایل را بده.

دستور `chown`

این دستور برای تغییر مالک فایل به کار می‌رود. البته چون اجازه‌های دسترسی و امنیت فایل‌ها با این روش به خطر می‌افتد لذا تنها کاربر `root` و یا کاربرانی که قبلاً با قدرت `root` در سیستم تعریف شده باشند قادر به اجرای این دستور می‌باشند.

با این دستور می‌توانیم مالک یک فایل را تغییر دهیم. مثلاً `chown root` با این دستور یعنی می‌گوییم که مالک مسیر `home/oracle` کاربر `root` باشد. اگر بخواهیم عمل تغییر مالک را به صورت تودرتو در داخل یک مسیر و تمام زیر مسیرهای آن و روی تمام فایل‌های داخل مسیرهای زیرین آن انجام دهیم از گزینه “-R” استفاده می‌کنیم.

مثلاً اگر بخواهیم در مسیر `/u01` مالک کل مسیر و تمام زیر مسیر و فایل‌های آن به نام کاربر `oracle` شود از دستور

“`chown -R oracle /u01`”

استفاده می‌کنیم



دستور chgrp

این دستور برای تغییر گروه یک مسیر و فایل‌های یک مسیر می‌باشد و مانند chown باید اجازه root را دارا باشد.

تغییر دهیم با اجرای دستور oinstall /u01 را به u مثلاً اگر می‌خواهیم گروه مسیر
“chgrp -R oinstall /u01”

این عمل را انجام می‌دهیم.

نکته: در دستور chown می‌توانیم هم‌زمان مالک و گروه را تغییر بدهیم.

شکل استفاده از دستور به این صورت است “chown oracle:oinstall /u01” در اینجا فقط پوشه بیرونی یعنی /u01 دارای مالک oracle و گروه oinstall می‌شود و اگر بخواهیم این عمل را روی کل پوشه و زیرپوشه‌ها و فایل‌های آن انجام دهیم از گزینه “-R” به صورت “chown -R oracle:oinstall /u01” استفاده می‌کنیم. علامت “:” برای جداسازی مالک از گروه می‌باشد.



... در طی تجارب تدریسم متوجه شده‌ام که بسیاری از کاربران که خیلی با سیستم عامل لینوکس آشنایی ندارند هنگام گذراندن دوره LPIC-1 دچار سختی می‌شوند، لذا به‌نظرم رسید که دوره لینوکس مقدماتی را به چاپ برسانم تا برای افرادی که هیچ چیز زیادی از این سیستم عامل نمی‌دانند، دنبال کردن و یادگیری مطالب آن راحت‌تر بوده و درواقع این دوره مثل پلی است که راه رسیدن به دوره LPIC-1 را هموارتر خواهد ساخت. بنابراین اگر هیچ چیزی از لینوکس نمی‌دانید نگران نباشید و برای یادگیری آن با من همراه شوید.



Taali Publishing

شابک ۹۷۸-۶۰۰-۸۶۸۰-۰۱-۷