ارائه شده توسط وبسایت ایرانیان دانلود

مرجع دانلود کتاب الکترونیکی

www.ir-dl.com

لینوکس مقدماتی

Course Title	**Linux Essentials**
Date	**10/24/2016**
Location	**Tehran/Iran**
Instructor Name	**Sophia Alikhani**
Instructor Email	**Sophia.alikhani47@gmail.com**

بازخورد شما نسبت به مطالب ارائه شده باعث رشد ما در ارائه بهتر و بیشتر مطالب در وب سایت ما خواهد شد . پس منتظر ارائه نظرات خوب شما دوستان هستیم .

با سپاس

صوفیا علیخانی

Sophia.alikhani47@gmail.com
http://www.linuxmahz.com

سرفصل مطالب

1. **مقدمه**

2. انقلاب لینوکس
3. آشنایی با برنامه های کاربردی متن آزاد..
4. مهارت ICT و کار با لینوکس
5. دستورات اولیه لینوکس
6. Help linux
7. مدیریت فایلها و دایرکتوری ها در لینوکس
8. ایجاد حذف و جابجای فایلها در لینوکس
9. نحوه پشتیابن گیری از داده ها در لینوکس
10. نحوه جستجوی متن در فایلها
11. داده های ما کجا ذخیره میشوند
12. انواع کاربران سیستم عامل لینوکس و مفاهیمی از امنیت اولیه در سطح سیستم عامل لینوکس در ارتباط با کاربران
13. ایجاد کاربران و گروههای کاربری
14. نحوه اجازه دسترسی به کاربران

1- مقدمه

تمام افرادیکه میخواهند با سیستم عامل لینوکس آشنا شوند دوره هایی را بصورت بین المللی طی میکنند که یکی از این دوره ها با نام دوره LPIC-1 میباشد . در طی تجارب تدریسم متوجه شده ام که بسیاری از کاربرانی که خیلی با سیستم عامل لینوکس آشنایی ندارند هنگام گذراندن دوره LPIC-1 دچار سختی میشوند لذا بنظرم رسید که دوره لینوکس مقدماتی را به چاپ برسانم تا برای افرادیکه هیچ چیز زیادی از این سیستم عامل نمیدانند هم دنبال کردن و یادگیری مطالب آن راحتتر بوده و درواقع این دوره مثل پلی است که راه رسیدن به دوره LPIC-1 را هموار تر خواهد ساخت .

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 می باشد .

2- انقلاب لینوکس و شناسایی سیستم عاملهای مطرح

آنچه در این بخش میخواهم عنوان کنم این است که اصلا لینوکس چیست و مفهوم **open source** که به فارسی به آن متن باز میگوییم یعنی چی ؟

ابتدا به تعریف فلسفه **open source** می پردازم

اما مفهوم آزادی در این سیستم ها چیست ؟

مفهوم از آزادی را میتوان به اشکال زیر تعریف نمود :

۱- آزادی در داشتن کل متن برنامه نوشته شده

۲- آزادی در تعداد کپی هایی 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 به 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 دامنه های تعریف شده برای مفهوم آزادی در برنامه ها 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

مثلا 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 نمیکنید

حالا فرض کنید به شیرینی فروشی میروید و شیرینی فروشی میگوید من دستور پخت این باقلوا را بدون گرفتن هزینه به شما میدهم

در اینحالت شما میتوانید این دستور پخت را گرفته و بر اساس ذائقه خود آنرا تغییر داده کم شیرین یا شیرین تر کرده یا گردوه اضافه کنید و یا هر سلیقه ای که دارید روی‌آن پیاده کرده و محصول جدید خود را به دیگران یا بفروشید یا آزادانه در اختیارشان قرار دهید .

اینها مثالهایی از آزادی است.

در سیستم عامل و برنامه هایی از این دست دقیقا این مفهوم بکار میرود . س‌یستم های متن بازی داریم که در ازا گرفتن پول کل دستور پخت و ساختار سیستم عامل یا برنامه خاص در اختیارتان قرار میگیرد و یا بدون دریافت پول این نسخه به شما داده میشود .

در حالت اول شما پولی پرداخت میکنید و دستور پخت را میگیرد بعد به سلیقه خود آن را تغییر میدهید .

در حالت دوم شما بدون پرداخت پول این دستورالعمل را تهیه و تغییر می دهید

در سیستم عامل لینوکس و برنامه های تحت آن دقیقا این مفاهیم آزادی مصداق پیدا میکند یعنی مواردی از برنامه ها هستند که در ازای پول نسخه اصلی آن و یا نحوه ساخت برنامه ها و با تمام سورس برنامه در اختیارتان قرار میگیرد و گاهی بدون دریافت پول و کاملا آزاد تمام مراحل ساخت آن در اختیار شما قرار میگیرد .

هر دوی این موارد نشانه آزادی هستند و هر کدام مزایا و معایب خود را دارا میباشند

مثلا کمپانی RedHat کلیه سورس برنامه های و ساخت سیستم عامل را در ازا دریافت پول به شما میفروشد و حالا شما قادرید آن سورس را بر اساس نیاز خود تغییر داده و نسخه جدیدی ارایه دهید و حالا این میل شماست که این نسخه را بفروشید یا آزادانه و بدون دریافت پول در اختیار دیگران قرار دهید .

```
#!/bin/bash
echo "what is your favorite color?"
echo
```

یا مثلا نسخه **CentOS** همان سورس نسخه **RedHat** میباشد که از آنجا خریداری شده ولی بعدا از طریق این کمپانی به شکل کاملا آزاد در اختیار دیگران قرار میگیرد. یعنی این کمپانی برای در اختیار قراردادن نسخه خود هیچ پولی از شما دریافت نمیکند و کاملا آزادانه آنرا در اختیار دیگران قرار میدهد .

```
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
```

بعنوان مثال به برنامه مقابل توجه کنید که کد آنرا نوشته

و اجرا میکنم

زمانیکه 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 میتوان در اختیار دیگران قرار داده تا دیگران بر اساس سلیقه خود آن را تغییر دهند . این همان مفهوم آزادی در متن برنام میباشد . یعنی ما متن و یا سورس برنامه هایی که نوشتیم را در اختیار همگان قرار داده و دیگران میتوانند بر اساس نیاز خود آن را دست کاری کرده و به شکل درخواه خود نوشته و اجرا نمایند .

دقیقا مثل اینکه من دستور پخت باقلوا را در اختیار شما قرار دهم اینجا دستور ساخت این فایل 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 آنها صحبت کنیم

مثلا 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 مثلا نحوه اجرای برنامه ها یا به اصطلاح پکیج ها و یا نحوه بروز رسانی ها متفاوت از نسخه **RedHat** میباشد .

یا مثلا نسخه دیگر سیستم عامل لینوکس مثل **SUSE** از محیط گرافیک متفاوت و یا ابزارهای متفاوتی از نسخه **Debian** استفاده میکند و یا مثلا **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** دقیقا دستور العملها مانند هم بوده و مثل این است که هر دو از دستور العمل یکسان پخت باقلوا پیروی میکنند

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

مثلا گوشی های موبایل که سیستم عامل آنها اندروید میباشد یا **Roku** که وسیله ای 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 سیستم عامل لینوکس استفاده شده است و ابزارهای متفاوتی با استفاده از آن ساخته شده و در اختیار دیگران قرار گرفته است .

پس می بینیم که لینوکس همه جا 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 آنرا تغییر داده و برنامه ای جدید از آن تولید نماییم .

3- آشنایی با برنامه های متن آزاد

دسته بندی هایی که میتوان برای این نوع برنامه ها در نظر گرفت میتواند بصورت زیر باشد :

1-3- Desktop

2-3- Server

3-3- Admin

4-3- programming

5-3- mobile

در محیط دستکتاپ میتوان برنامه های زیر را بعنوان برنامه های متن آزاد نام برد :

openoffice

liberoffice

firefox

Thunderbird

chromium

- در محیط سرور میتوان برنامه های زیر را بعنوان برنامه های 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 برنامه نویسی

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 سیستم های متن باز

در دنیای سیستم های متن آزاد دو نوع سازمان هستند که لیسانس لازم برای دریافت مدرک متن آزاد بودن را ارایه میکنند

۱- **FSF Free Software Fundation**

2- **OSI Open Source Initiative**

هیچکدام از این دو کمپانی بر دیگری ارجحیت ندارند بلکه هر کدام سیاست مخصوص خود در ارایه مدرک را برعهده دارند و هر کدام هدفی خاص را در این راستا دنبال میکنند

FSF از استانداردهای **GPL** تبعیت میکند که در این استاندار آزادی مطلق بر هر نرم افزاری وجود دارد . آزادی برای هر چیزی . هیچ حق تقدمی وجود ندارد هیچ حق تقدمی در برنامه ها یا درایورها و یا هر چیز دیگری در سیستم وجود ندارد و همه چیز آزاد است

4- مهارت ICT و کار با لینوکس

ICT به معنی **Information and Communication Technology skills** می باشد یا مهارت استفاده از اطلاعات و ارتباطات در محیط لینوکس .

بعبارت ساده تر میخواهیم بدانیم چگونه در لینوکس میتوانیم کارهای روزمره و یا پروژه های اجرایی خود را بکار ببریم .

خب اول به این سوال جوا بدهیم که لینوکس در کجاها یافت میشود ؟

امروز میتوان لینوکس را در همه جا مشاهده کرد مانند

- محیط **server room**

- **virtualization hosts** یا محیط های مجازی سازی مثل **vmware**

- **cloud computing** یا همان مبحث رایانش ابری مانند سرویس آمازون

- سی‌ستم های الحاقی یا **embedded** مانند دستگاههای موبایل

- **factory automation** که از محیط **DOS** به لینوکس تغییر یافته است

- **mars rover**

اما آنچه که در ادامه در مورد آن میخواهیم صحبت کنیم در باره استفاده لینوکس در محیط دستکتاپ یا صفحه شخصی کامپیوتر شما می باشد .

از جمله مطالبی که در این قسمت پوشش خواهم داد عبارتند از

- چگونگی استفاده از مدیر دسکتاپ یا **Desktop Manager**

- برنامه های کاربردی آفیس

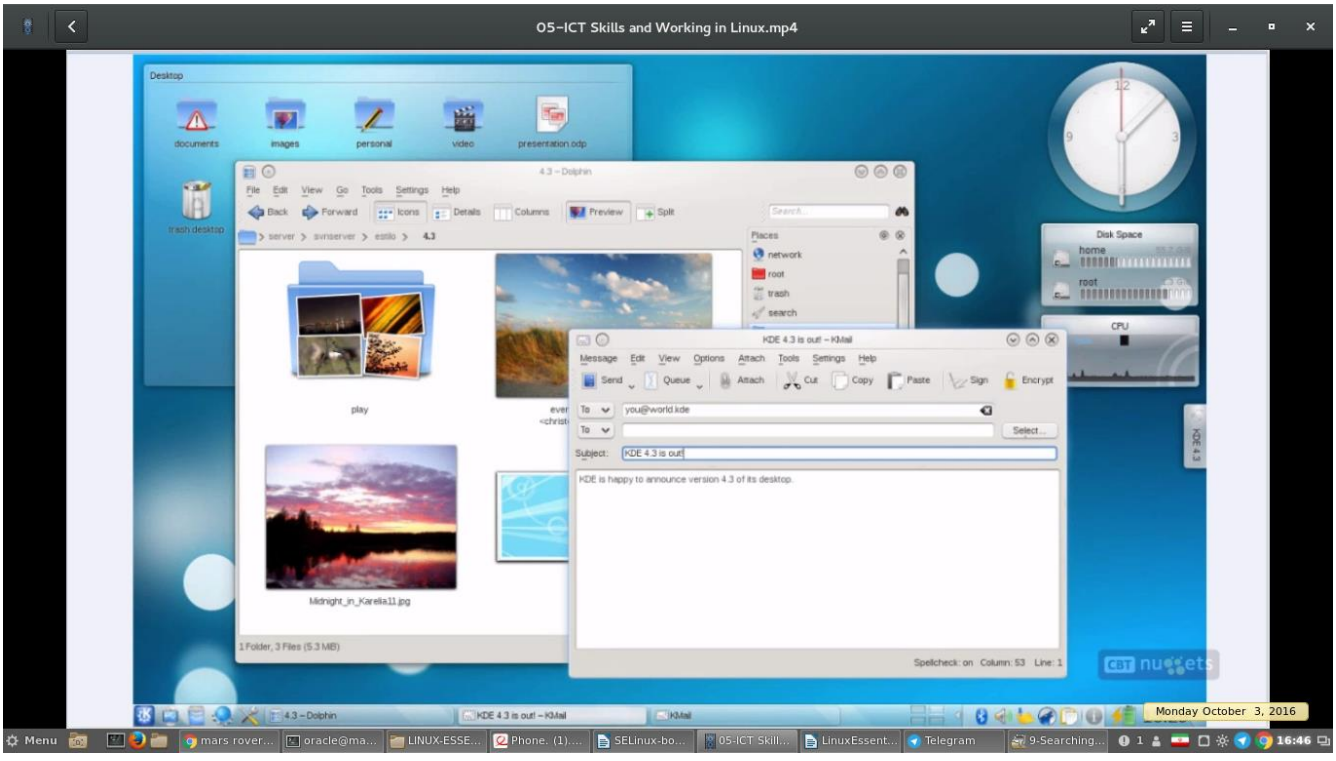
-مرورگرها **Browser**

- نحوه ذخیره فایلها

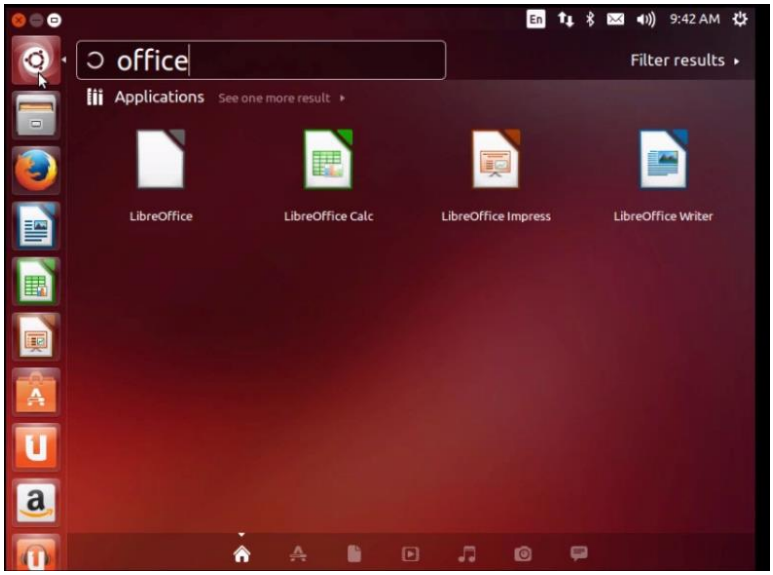
- مفهوم ترمینال

- چگونگی بکارگیری کلمه رمز یا پسورد

در شکل زیر یک محیط شخصی سازی بنام **KDE** را نمایش داده ام .یک محیط کاملا 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 بنام **unity** میباشد

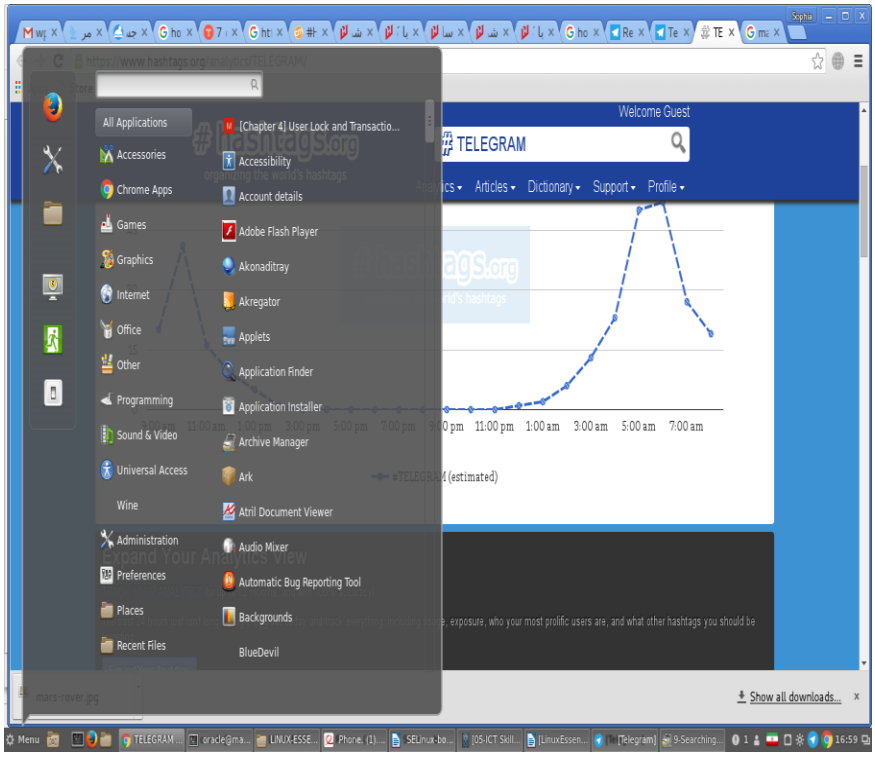
و بسیاری محیطهای گرافیکی دیگر در مدیریت کارهای روزانه در سیستم عاملهای مختلف لینوکس داریم .

هر یک از نسخه های لینوکس بطور پیش فرض از یکی از این محیط های گرافیکی برای محیط دسکتاپ خود استفاده میکنند مثلا در نسخه **SUSE** بطور پیش فرض از **KDE** استفاده میشود و در **fedora , centos** بصورت پیش فرض **GNOME** استفاده میشود یا در نسخه **ubuntu** از محیط **unitry** استفاده میشود که هر کدام ویژگیهای مخصوص خود را برای مدیریت دسکتاپ بکار میبرند .

اما در هر کدام از نسخه های لینوکس میتوان هر کدام از محیط های گرافیک دیگر را نیز نصب و اجرا کرد مثلا در **SUSE** میتوان بجای محیط پیش فرض گرافیکی آن **KDE** از **GNOME** یا **unity** یا محیط های دیگر نیز استفاده نمود و براحتی میتوان هر کدام از محیط های مورد علاقه کاربر را نصب و بکار برد .

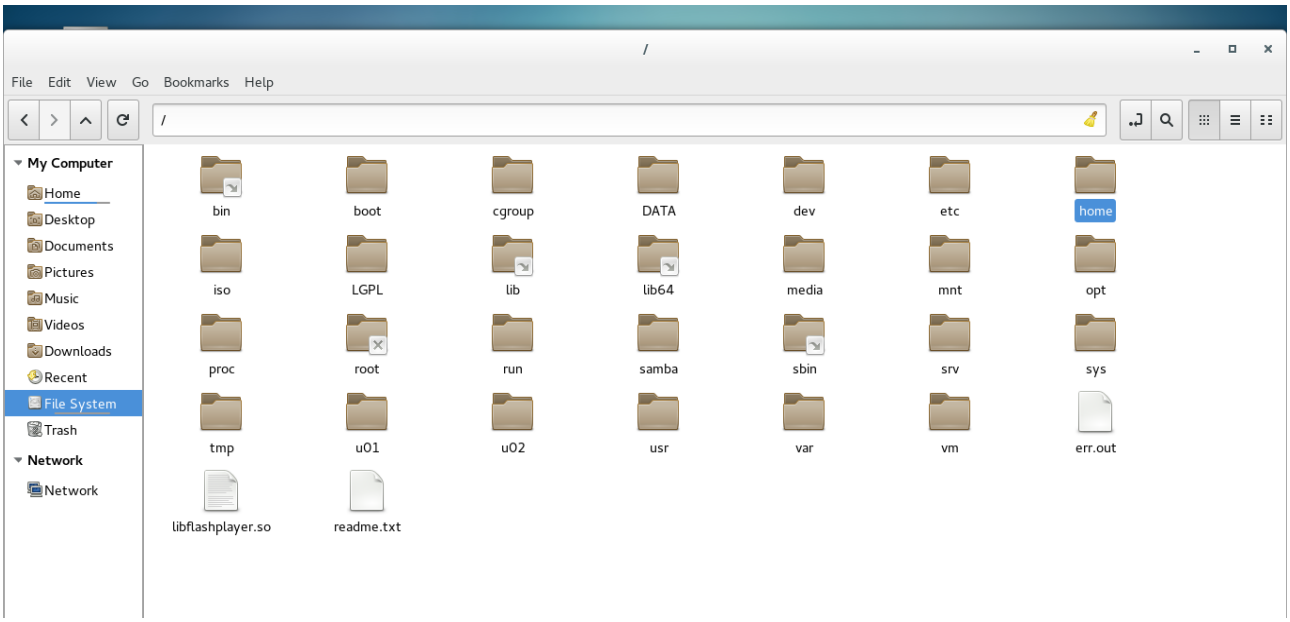
اینکه از چه نوع محیط گرافیکی برای کارهای روزانه تان استفاده میکنید کاملا سلیقه ای بوده و براساس اینکه در کدام محیط راحتتر کار میکنید میتوانید هر کدام را که مایلید روی سیستم نصب و بکارگیرید.

بعنوان نمونه من از محیط گرافیکی سیستم خودم که با **centos 7** است نمونه ای 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 دایرکتوری های خود دسترسی داشته با حرکت موس روی آن ها میتوانید آنها راحذف یا کپی و پیست نمایید دقیقا مثل زمانیکه در محیط ویندوز کار میکنید .

به شکل زیر دقت نمایید



در شکل بالا مشاهده می نمایید که کلیه فولدرها و فایلها بصورت گرافیک نمایش داده شده است و میتوانیم با حرکت موس داخل هر پوشه ای که میخواهیم پیمایش کنیم یا فایلهای داخل آنها را حذف کرده یا جای دیگری منتقل کنیم .

برای یادگیری بیشتر نیاز دارید که در کلاس های آموزشی ای که در این زمینه ترتیب داده ام شرکت کنید تا از نزدیک با محیط گرافیکی و نحوه بکارگیری ابزارهای آن آشنایی پیدا نمایید .

در این قسمت قصد دارم بطور خلاصه در مورد ساختار فایلهای داخل سیستم عامل لینوکس صحبت کنم .

فرض کنید یک هارد دیسک داریم که در محیط لینوکس قابل استفاده است

در زمانیکه من بعنوان کاربر سیستم عامل لینوکس وارد این سیستم عامل میشوم یعنی جاییکه نام کاربری و پسورد خودم را میدهم و در صورت شناخته شدن وارد محیط گرافیکی خود میشوم در واقع در مسیری خاص بنام /**home**/ قرار دارم

حالا اگر نام کاربری من مثلا **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 .

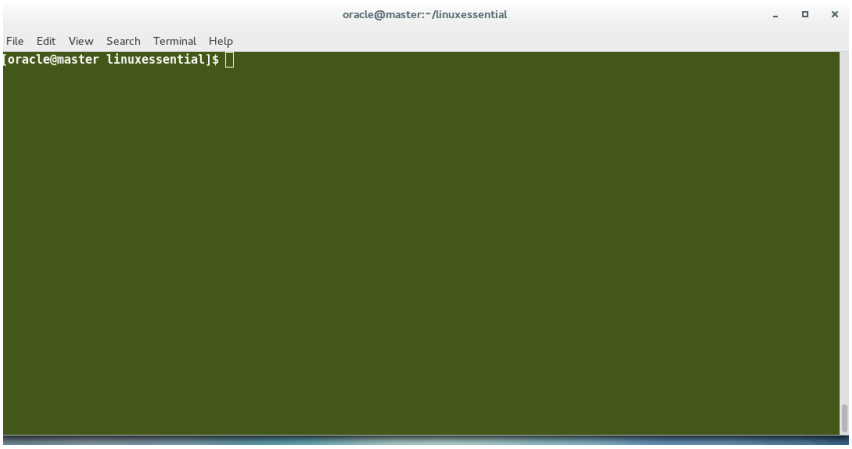
این موضوع دقیقا شبیه به آنچه که در محیط ویندوز مشاهده میکنید می باشد .

تمام مسیر ها در لینوکس از یک مسیر اصلی ریشه یا **root** که با نماد "/" نشان داده میشود نشات میگیرد .

/home/ به معنی مسیر خانه میباشد .

یعنی هر کاربری در سیستم عامل دارای خانه اختصاصی خود میباشد و کلید ورود به خانه به هر کاربر مالک آن خانه داده میشود .

در تمام این محیطهای گرافیکی ابزاری بنام ترمینال داریم که از آن میتوانیم وارد محیط متنی شده و دستورات لینوکس را تحت آن اجرا نماییم .



شکل مقابل نمایش یک ترمینال باز شده در محیط گرافیکی سیستم عاملی لینوکس نسخه **CentOS7** می باشد .

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

5- دستورات اولیه در لینوکس

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 وارد شده را به زبانی که لینوکس متوجه آن شود ترجمه میکند که به آن shell یا پوسته میگویند در واقع این پوسته دور هسته اصلی لینوکس کشیده شده است .

بطور پیش فرض کسی بطور 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 درآن جا 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 آنرا ترجمه نماید محیط متنی پوسته لینوکس یا Linux Command Shell میگویند .

در واقع این پوسته نقش مفسر را بازی میکند یعنی هر کلمه ای که درپوسته وارد کنیم سریعا برای هسته ترجمه میکند حالا اگر کلمه ما بتواند با توجه به دفترچه لغات سسیتم عامل لینوکس قابل ترجمه باشد و معنی داشته باشد نتیجه آن به ما باز میگردد و چنانچه کلمه یا جمله ما معنی خاصی برای سیستم عامل نداشته باشد با یک پیغام خطا ما را متوجه این موضوع میکند .

خب قصد دارم در این قسمت به شرح موضوعات زیر بپردازم :

- دستورات و گزینه های مربوط به آنها

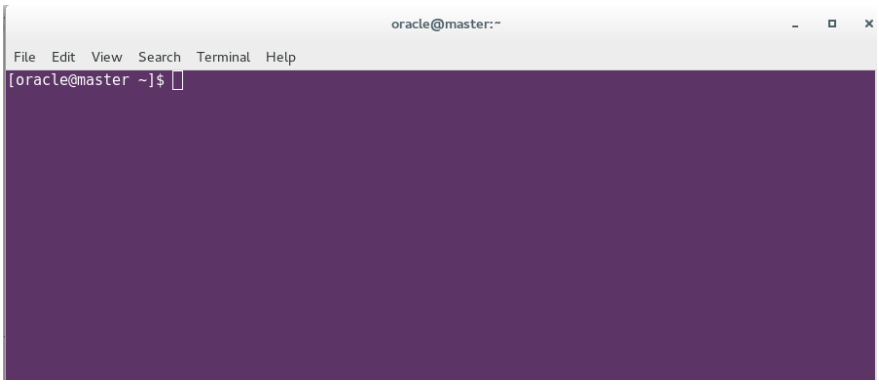
- استفاده از کارکترهای خاص برای یافتن نام فایلها و دایرکتوری ها

- استفاده از علامت نقل قول یا quote

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

خب 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 مثلا اگر در محیط اوبونتو هستیم کافیست کلمه **terminal** را تایپ کنیم تا شکل گرافیکی آن نمایش داده شود و سپس با انتخاب آن و زدن کلید موس آنرا انتخاب میکنیم در این حالت یک ترمینال روی صفحه برای ما ظاهر میشود به شکل زیر :



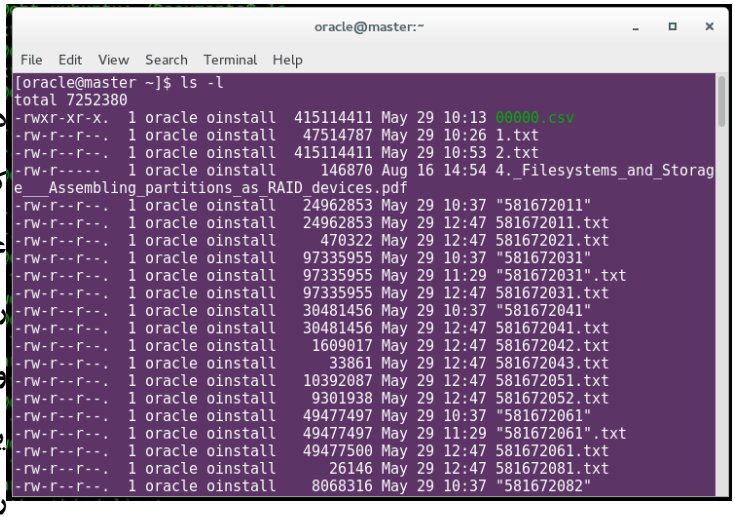
در شکل مقابل روی صفحه بنفش ظاهر شده که همان ترمینال ما هست این عبارت را مشاهده میکنید:

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** می باشد .

و علامت $ معمولا برای مشخص کردن نشانه پوسته افرادی که **root** نیستند بکار میرود.

در شکل من دستور **ls -l** را جلوی علامت $ اجرا کرده ام 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 آنرا ترجمه کند آنرا به هسته لینوکس یا کرنل میدهد و کرنل آنرا اجرا کرده و جواب خروجی را به ما برمیگرداند .



عبارت وارد شده 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** می باشد از کلیه فایلها و دایرکتوری های مسیری که در حال حاضر درآن قرار داریم یعنی مسیر خانه شخصی ما لیستی تهیه کرده و از خصوصیات آنها بصورت جزء به جزء اطلاعاتی روی ترمینال ظاهر می کند .

و گزینه **l**- به معنی این است که این لیست را با جزییات کامل به من نمایش بده

در لیست ظاهر شده 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 .

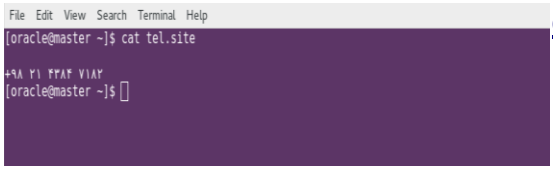
فرمان **ls** را اگر بدون هیچ گزینه ای اجرا نماییم لیست نمایش داده شده فقط شامل نام فایلها و دایرکتوری های مسیرمان میشود و اطلاعات کاملتر از مالک و سایز و غیره نمیدهد .

دستور بعدی دستور **cat** میباشد

این دستور برای نمایش محتوای یک فایل بکار میرود که عموما این فایلها بصورت متنی هستند حالا ممکن است این متن یک متن برنامه نویسی به زبانی خاص باشد یا متن خاصی از یک کتاب باشد و یا هر متنی

ولی عموما دستور **cat** برای نمایش محتوای فایلها بکار 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

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

یا مثلا دستور **cat tel.site** با فرض اینکه فایلی بنام **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 مثلا اگر **file1 , file2** داشته باشیم با اجرای فرمان **cat file1 file2** خروجی ظاهر شده روی ترمینال از محتوای دوفایل بنامهای **file1 , file2** می باشد .

در خروجی لیست فایلهای من فایلهایی هستند که همگی با عدد شروع شده اند و انتهای آنها رشته یا عبارت **txt** به چشم میخورد حالا چنانچه من بخواهم لیست کلیه فایلهایی را ببینم 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 کنم

میخواهم لیستی از فایلها و یا پوشه هایی بگیرم که نام آنها هر چه میخواهد باشد اما انتهای آنها عبارت **txt** وجود داشته باشد را نمایش دهد 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 هنگامیکه بخواهیم بصورت دسته جمعی دستوری را اجرا نمایی از علامت * به معنی رخداد صفر تا هر تعداد از یک واقعه را جایگزین می کنیم .

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

حالا اگر بخواهیم فقط بجای یک حرف خاص از علامتی استفاده کنیم مثلا فقط میخواهیم بگوییم کلیه فایلهایی را نشان بده که یک حرفی هستند آنوقت از علامت ? استفاده میکنیم به صورت زیر

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

در مثال بالا لیست کلیه فایلها و پوشه هایی را لازم داریم که نامشان یک حرفی است حالا این حرف الفبا باشد یا عدد مهم نیست ولی انتهای آن با .**txt** خاتمه می یابد را نمایش میدهد حالا اگر چنین فایلهایی در مسیر مورد نظرمان 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 بگویم

```
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 می باشد .

دستور **echo** هر عبارتی جلوی آن قرار گیرد را نمایش میدهد یا بعبارتی روی صفحه منعکس میکند مثلا

دستور **echo** کلمه **Hello** را روی صفحه نمایش میدهد . `Hello`

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 زمانیکه تعداد بیشتری فاصله بین این دو حرف باشد یعنی **Hello world** خروجی فرمان بصورت فقط یک فاصل ظاهر میشود یعنی بصورت زیر :

Hello World

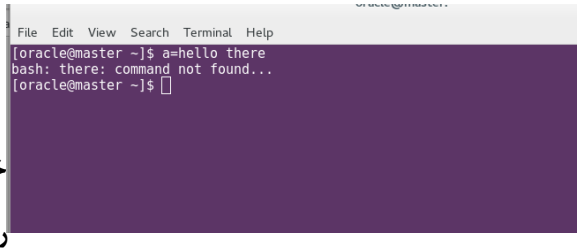
چنانچه قصد ما از نمایش متن این است که دقیقا فاصله بین دو حرف عین آن چیزی که می نویسیم در نظر گرفته شود بنابراین دو کلمه را در داخل علامت نقل قول یا "" قرا رمیدهیم بنابراین برای ظاهر شدن دو کلمه **Hello world** با در نظر گرفتن بیشتر از یک فاصله بین آنها این عبارت را داخل علامت نقل قول قرار میدهیم یعنی بصورت زیر :

echo "Hello world"

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

حالا فرض کنیم من دستور زیر را روی ترمینال اجرا میکنم

oracle@master ~]$ a=hello there



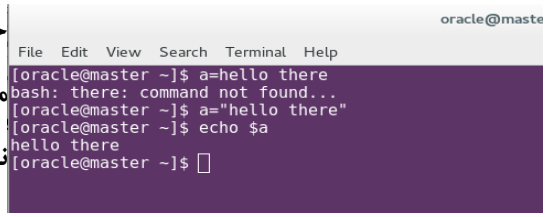
خروجی دستور با خطا مواجه میشود و بصورت زیر پیامی روی صفحه مشاهده میکنیم :

در شکل بعد از خط اجرا پیغام

bash: there command not found را مشاهده میکنید این پیام یعنی من دستوری بنام **there** پیدا نمیکنم که اجرا نمایم.

در واقع با زدن دستور **a=hello there** پوسته مجموعه این عبارات را به هسته یا کرنل لینوکس میفرستد اما وقتی هسته در کتابچه لیست دستورات خود نگاه میکند می بیند که در ابتدا عبارت **a=hello** دارد که برای سیستم عامل این معنی را دارد که متغیری بنام **a** وجود دارد که با دستور بالا میگوییم مقدار این متغیر را برابر با کلمه **hello** قرار بده و بعد از کلمه **there** را می بیند و این کلمه برای سیستم عامل چون بدنبال عبارت **a=hello** آمده به معنی نام یک دستور می باشد یعنی سیستم عامل در لیست دستوراتش بدنبال دستوری با نام **there** میگردد و می بیند همچین دستوری ندارد پس در خروجی مینویسد که من همچین دستوری ندارم که بفهمم منظورش چیست

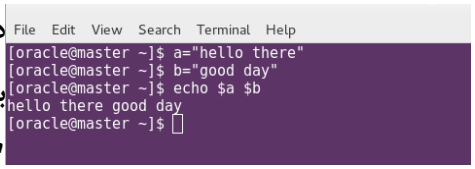
بعبارتی اگر منظور ما از اجرای **a=hello there** این باشد که کل عبارت **hello there** را به متغیر **a** تخصیص دهیم این اتفاق نمی افتد و فقط مقدار **hello** به متغیر **a** تخصیص داده میشود و چون فاصله ای بین کلمه **hello** و **world** وجوددارد این فاصله برای سیستم عامل به معنی اجرای فرمان بعدی می باشد .



حالا اگر منظور ما دقیقا 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 میدهیم

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

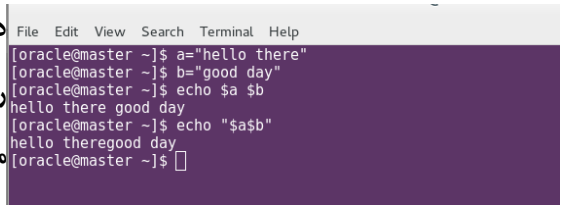


در شکل مثالی دیگر از بکارگیری مقدار متغیر ها میباشد در مثال بعدی که در شکل مشاهده میکنید مقدار متغیر **a** را برابر با مقدار "**hello there** " و مقدار متغیر **b** را برابر با "**good day**" قرار دادم حالا با دستور **echo $a $b** مقدار این دو متغیر بصورتی که در شکل با یک فاصله می بنید ظاهر میشود

حالا چنانچه دستور زیر را وارد کنیم

echo "ab"

خروجی بصورتی ظاهر میشود که هیچ فاصله بین دو رشته حرفی **helo there** و **good day** وجود ندارد



در شکل در داخل علامت نقل قول محتوای دو متغیر **a, b** را بدون هیچ فاصله ای قرار دادیم و خروجی آن مقدار محتوای این دو متغیر را بدون فاصله کنار هم نمایش میدهد یعنی بصورت **hello theregoodday**

با گفتن این مقدمه در مورد متغیرهای و چگونگی مقدار دهی به آنها میخواهم 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 آنها مقادیر خاصی را ذخیره و نگهداری میکند و در جای مناسب از آنها استفاده میکند بعنوان مثلا یکی از این متغیرها **PATH** نام دارد .

این متغیر همانطور که از اسم آن پیداست مسیری را در خود نگهداری میکند . **PATH** یعنی مسیر

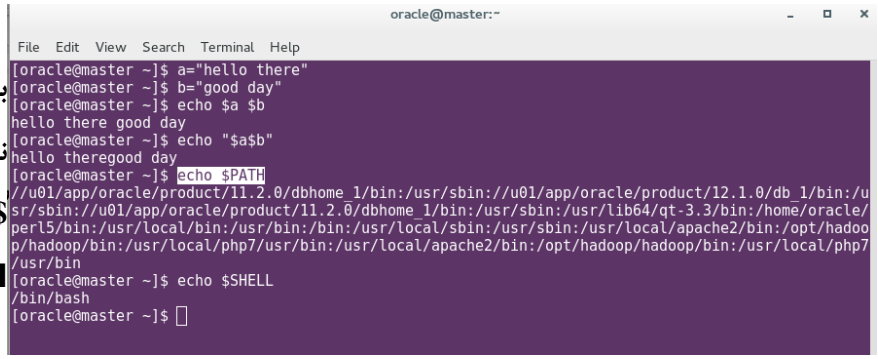
در ضمن کلیه متغیرهای داخل سیستم عامل که خود سیستم عامل از آنها استفاده می کند با حروف بزرگ نگهداری شده و معنی خاص خود را دارند .

یادمان باشد که در لینوکس حروف بزرگ و کوچک هر کدام بصورت مجزا هستند یعنی اگر فایلی با نام **File1.txt** و فایلی با نام **file1.txt** 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 .

از اینرو در سیستم عامل متغیرهای خاصی که برای لینوکس مقدار خاصی را ذخیره میکند نیز با حروف بزرگ ثبت شده است متغیر **PATH** که همگی با حرف بزرگ هستند به معنی این است که مسیر برنامه های اجرایی در آن ذخیره میگردد .

و از مقدمه این بخش آموختیم که چنانچه بخواهیم مقدار یک متغیر را نمایش دهیم از علامت $ در ابتدای آن استفاده میکنیم .

بنابراین اگر بخواهیم مقدار متغیر **PATH** را نمایش دهیم قبل از نام متغیر علامت $ میگذاریم و با دستو ر **echo** **$PATH** مقدار متغیر را مشاهده میکنیم



فعلا 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 هایی میکند

متغیر بعدی متغیر **SHELL** با حروف بزرگ است که نوع پوسته ای که درسیستم عامل استفاده میکنیم را در خود نگهداری می کند و برای اینکه بدانیم با چه پوسته ای کار میکنیم با اجرای دستور **echo $SHELL** مقدار آنرا مشاهده میکنیم .

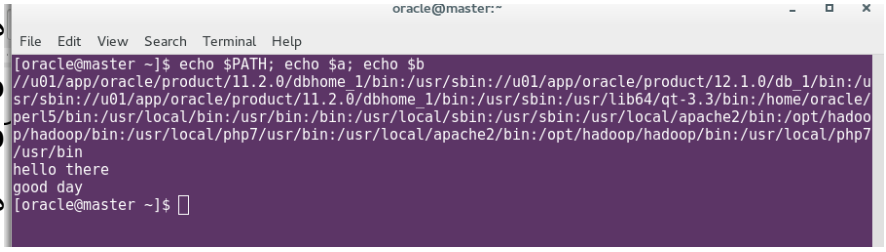
همچنین اگر بخواهیم چند دستور از دستورات سیستم عامل را روی یک خط و بصورت ترتیبی و پشت سرهم بیاوریم بطوریکه به ترتیب اجرا گردند از علامت ; در بین دستورات استفاده میکنیم

مثلا

oracle@master ~]$ ls ; echo $PATH

دستور بالا میگوید اول دستور **ls** را اجرا کن و بعد از آن دستور **echo $PATH** را اجرا کن .

در شکل ، سه دستور **echo $PATH;echo $a; echo $b** را مشاهده میکنید که هر دستور در خطوطی مجزا اجرا شده است .



درواقع بجای اینکه هر دستور را در خطوط مجزا بدهیم همگی را در یک خط داده و با علامت ; آنها را جدا سازی مینماییم .

تا اینجا متوجه شدیم که پاره ای از علائم برای سیستم عامل معنی دارد مثل

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

? بجای یک حرف مثل **ls ?.txt**

"" برای قرار دادن یک رشته که بین آنها فاصله است مثلا **a="hello Ali "**

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

یکی دیگر از دستورات ساده لینوکس دستور **history** به معنی تاریخچه میباشد . این دستور تاریخچه دستوراتی را که از قبل اجرا کرده اید را نمایش میدهد .

```
File  Edit  View  Search  Terminal  Help
  987  ssh -l hendooneh sophaalikhani
  988  ssh -l hendooneh sophiaalikhani
  989  clear
  990  ls -l
  991  vi .bashrc
  992  source .bashrc
  993  clear
  994  ls -l
  995  exit
  996  ls -l
  997  cat tel.site
  998  clear
  999  cat tel.site
 1000  clear
 1001  a=hello there
 1002  a="hello there"
 1003  echo $a
 1004  b="اه مچ مالس"
 1005  echo $b
 1006  clear
 1007  a="hello there"
 1008  b="good day"
 1009  echo $a $b
 1010  echo "$a$b"
 1011  echo $PATH
 1012  echo $SHELL
 1013  ls; echo $a; echo $b
 1014  clear
 1015  echo $PATH; echo $a; echo $b
 1016  history
[oracle@master ~]$
```

فرض کنید امروز صبح وارد سیستم شدید و دستوری را اجرا کردید و حالا بعد از ظهر است و یادتان نمی آید کدام دستور بود که صبح اجرا کردید با اجرای فرمان **history** میتوانید لیست کلیه دستوراتی را که اجرا کرده بودید را مشاهده کنید و دستور مورد نظر را از لیست نمایش داده شده بیابید .

تا اینجا با تعدادی از دستورات ساده لینوکس آشنا شدید .

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

6- چگونه در لینوکس از Help استفاده کنیم

خب فرض کنید که رفتید به فضا و زبان فضایی ها رو بلد نیستید و دنبال این هستید که حرفهاتون را براشون ترجمه کنید در این حالت چیکار 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 فضایی ها را در آن بیابید

دقیقا همین کار را در سیستم عامل لینوکس میکنیم . یعنی میتوانیم در ترمینال و در قمستی که علامت **shell** ظاهر شده است دستور مورد نظر را با کلمه **help–** وارد کنیم.

یعنی زمانیه ما اصلا معنی بعضی از دستورات لینوکس را متوجه نمیشویم پس 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 بخوبی راهنمایی کند

که عبارتند از

- **Man**
- **Info**
- **Main -k**
- **Apropos**
- **Which**
- **Whereis**

یکی از روشهایی که این مشکل را برای ما برطرف میکند استفاده از دستور **man** می باشد . دستور **Man** مخفف **manual** میباشد به معنی دستور العمل که به ما توضیحات کاملی از یک دستور به همراه تمام گزینه های مرتبط با این دستور را نمایش میدهد . بمنظور بکارگیری **man** باید این دستور را به همراه دستوری که میخواهیم راجع به آن اطلاعاتی کسب کنیم بکار ببریم . مثلا

$ man ls

یعنی میخواهم بدانم دستور **ls** چی هست جکار میکند و با چه گزینه هایی قابل استفاده است .

برای اینکه بدانیم با خود دستور **man** چگونه کار کنیم از دستور **man man** استفاده میکنیم که صفحه ای به شکل زیر ظاهر میشود

با زدن دستور **man ls** صفحه ای به شکل زیر ظاهر میشود

یک روش دیگر برای فهمیدن اینکه یک دستور در لینوکس چگونه کار میکند استفاده از **info** می باشد .

برای استفاده از دستور **info** این فرمان را با یکی از دستورات دلخواهمان اجرا میکینم مانند شکل زیر

$ info ls

این دستور هم مانند **Man ls** توضیحاتی راجع به دستور میدهد و در بعضی از موارد مثالهایی از گزینه های مختلف به همراه دستور را نمایش میدهد .

شکل زیر نمایشی از این دستور میباشد

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

مثلا میخواهیم دستوری مرتبط با پیکربندی شبکه را بیابیم خب 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** بکار ببریم که بصورت زیر نمیاشی از آن را میدهم

$man -k ping

در دستور بالا من میدانم که مثلا میخواهم یک ارتباط شبکه ای 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 بدنبال آن هستیم خواهیم یافت . لازم بذکر است ممکن است این لیست طولانی و شامل مثلا **200** یا **300** 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 میشود که این دستور آیا همان دستوری هست که دنبالش هست

یا خیر ؟ این دستور در بسیاری موارد کمک کننده میباشد و خیلی از موارد حتی مدیران سیستم تمامی دستورات را حفظ نیستند و با کمک این روش میتوانند دستوری را که دنبالش هستند را بیابند

شکل زیر نمایشی از اجرای این دستور میباشد

روشی دیگر استفاده از دستور **apropos** میباشد دستور **apropos** به همراه یک دستور درست مانند **man –k** عمل میکند یعنی لیستی از دستورات مرتبط با دستوری که همراه **apropos** آمده است را میدهد مثلا **apropos ping** یا **apropos echo**

تمامی این موراد لیست دستورات مرتبط با دستور **ping** یا دستور **echo** را نمایش میدهند .

یکی از روشهایی که میتوانیم ببینیم در سیستم عامل لینوکس چه دستوراتی داریم و توضیحات هر دستور را بصورت یک دفترچه راهنمای کامل مطالعه کنیم رفتن به مسیر **/usr/share/doc** میباشد وقتی به این مسیر میرویم لیستی از تمامی دستورات لینوکس را که در سطح سیستم عامل نصب شده است را مشاهده میکنیم که بصورت پوشه هایی از نام آنها قابل مشاهده هست حالا اگر داخل یک پوشه بریم لیستی از فایلهای داخل آن پوشه می بینیم که بصورت فشرده یا **.gz** موجود می باشد . مثلا فرض کیند که دستور **python** را روی سیستم نصب کرده ایم و میخواهیم در مورد آم مطالعه کاملی داشته باشیم بنابراین به مسیر **/usr/share/doc/python** میرویم و لیستی از فایلهای این مسیر را مشاهده میکنیم حالا برای آنکه فایلهای فشرده شده را بدون اینکه بخواهیم از حالت فشردگی دربیاوریم مشاهده کنیم از دستور **less** به همراه نام فایل استفاده میکنیم .

مثلا **less python-policy.txt.gz** با اجرا این دستور توضیحات موجود در این فایل فشرده بطور کامل و صفحه به صفحه نمایش داده میشود که میتوانیم آنرا بطور کامل مطالعه نماییم.

روش دیگر برای اینکه بفهمیم یک دستور چکار میکند استفاده از دستور **whatis** میباشد بدین صورت که این دستور را به همراه دستوری که میخواهیم از آن اطلاعات کسب نماییم بکار میبریم .

مثلا **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 دلخواه مثلا **whereis ls** میگوید که دستور **ls** از چه مسیرهایی اجرا میشوند و در چه مسیری مستندات مربوط به این دستور قرار دارد

مثلا **whereis ls**

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

برای مثال میدانیم در سطح سیستم عامل دستوری بنام **ls** داریم و حالا فرض کنید که کاربر نیز برنامه ای نوشته و نام آنرا **ls** گذاشته است . حالا وقتی من کاربر فرمان **ls** را اجرا میکنم سیستم عامل کدام **ls** را اجرا میکند آیا **ls** ی که خود سیستم عامل جزو دستورات اولیه اش دارد یا دستور **ls** ی که خودم نوشته ام ؟

در این حالت با اجرای فرمان **which ls** میتوانیم بفهمیم که مسیر اجرایی که سیستم عامل برای اجرای دستور **ls** استفاده میکند چیست

پس بطور خلاصه در این فصل یاد گرفتیم

1- با استفاده از گزینه **help–** با یک دستور دلخواه نحوه استفاده از دستور به همراه گزینه های آن دستور را بطور خلاصه روی صفحه می بینم مثل **ls --help** یا **date --help** که بطور خلاصه توضیحاتی از دستور **ls** یا دستور **date** به همراه گزینه هایی که با این دستورات همراه است برای ما نمایش داده میشود.
2- دستور **ls man** یا **echo man** یا **ping man** یعنی همراه کردن دستور با دستور **man** به ما توضیحی کامل از دستور و گزینه های آن میدهد .
3- دستور **ls info** یا **ping info** یا **Info man** یعنی دستور **info** به همراه دستور دلخواه برای ما توضیحاتی از دستور را میدهد.
4- دستور **ls -k man** یعنی همراه کردن **man –k** به همراه دستور مورد نظر یا قسمتی از دستوری که بیاد می آوریم باعث میشود لیستی از کلیه دستوراتی که به نوعی با عبارتی که ما بعد از گزینه **k** وارد کرده ایم به همراه یک توضیح کوتاه از دستور روی صفحه ظاهر شود و در این صورت میتوانیم دستورات را مشاهده و دستوری را که فراموش کرده ایم بیاد بیاوریم.
5- دستور **ls apropos** که مانند **ls -k man** عمل میکند
6- دستور **ls whatis** که توضیح کوتاهی از دستور **ls** را در این مثلا نشان میدهد
7- دستوور **ls whereis** میگوید که دستور **ls** از چه مسیری اجرا میشود و همچنین در چه مسیری توضیحات این دستور آمده است .

8- دستور **which ls** میگوید که میگوید مثلا دستور **ls** از چه مسیری در حال اجراست اینطوری ما میفهمیم که آیا دستور **ls** را از مسیر درست اجرا کرده ایم یا خیر

7- مدیریت فایل ها و دایرکتوری ها در لینوکس

در این بخش قصد دارم به توضیح در مورد دستوراتی که بوسیله آن میتوانیم لیست فایلها و پوشه های خود را در یک مسیر خاص نمایش دهیم و دستورات مرتبط به لیست فایلها و دایرکتوری ها را توضیح دهم

دستوراتی که در این بخش دنبال میکنم عبارتند 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 خواهم داد

طریقه استفاده از دستور **cd**

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

اولا با فرض اینکه میدانیم در لینوکس همه چیز از **/** یا ریشه شروع میشود و هر مسیر از مسیر دیگر با علامت "/" جدا میشود حالا وقتی مثلا بخواهیم به مسیر **/home/oracle** برویم با استفاده از دستور **cd /home/oracle** این کار را انجام میدهیم

یا مثلا اگر بخواهم به مسیر **/etc** بروم با اجرای فرمان **cd /etc** این کار را انجام میدهم

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 میباشد .

پس با اجرای فرمان **"cd ."** اتفاق خاصی نمی افتد یعنی میگویم برو به مسیر جاری و این در حالیست که در مسیر جاری قرار داریم

اما دستور "cd .." بعنی 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

مچنین 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 مثلا فرض کنید در مسیر /etc قرار داریم و میخواهیم به مسیر خانه خودمان برگردیم در این حالت میتوانیم به دو صورت عمل کنیم

1- Cd /home/oracle
2- Cd ~

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

فرض کنید شخصی به تهران سفر میکند و بدنبال آدرس برج میلاد است و زمانیکه به ما مراجعه میکند ما میتوانیم بگوییم تهران- بزرگ راه همت – خروجی برج میلاد

یا اینکه نسبت به مکانی که در آن قرار داریم آدرس را به شخص مورد نظر بدهیم یعنی دیگر نمیگوییم تهران – بزرگ راه همت بلکه بسته به اینکه الان کجا هستیم مثلا آیا در جنوب تهرانیم یا شمال یا شرق یا غرب آدرس رسیدن به برج میلاد را میدهیم .

در حالتیکه آدرس دهی را از نقطه آغازین یعنی تهران و بقیه میدهیم از مسیر دهی مطلق استفاده کرده ایم و زمانیه آدرس را نسبت به مسیری که الان در آن قرار داریم میدهیم از مسیر دهی نسبی استفاده نموده ایم .

خب پس وقتی من میگویم برو به مسیر خانه یا home directory وقتی میگویم cd /home/oracle یعنی مسیر دهی مطلق استفاده کرده ام و وقتی مثلا در /home قرار دارم و بزنم cd oracle یعنی نسبت به جایی که الان قرار دارم برو به مسیر oracle دقیقا مثل مثال بالا که نسبت به آدرسی که در آن قرار داریم آدرس برج میلاد به طرف مقابل میدهیم

همچنین استفاده از دستور **cd ../..** یعنی دو تا مسیر برو عقب تر یعنی اگر مثلا در مسیر **/home/oracle/app/product** قرار داریم و دستور **cd ../..** را اجرا کنیم به مسیر **/home/oracle/** وارد میشویم .

یا مثلا 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 شده ایم .

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

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

حالا که فهمیدیم چگونه به مسیرهای مورد نظرمان در سیستم عامل لینوکس برویم میخواهیم بدانیم در این مسیرها چگونه لیست فایلها و دایرکتوری های آن مسیر را ببینم که این کار با اجرای دستور **ls** انجام میشود.

اما در هر مسیری فایلهایی هستند که با اجرای دستور **ls** بصورت ساده نمایش داده نمیشوند و به این نوع فایلها پنهان میگویند یعنی نامشان قابل رویت 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 دایرکتوری های مسیر مورد نظر که چه پنهان باشند یا نباشند را نشان میدهد . توجه داشته باشید که در هر مسیری دو تا فایل بصورت پنهان وجوددارند یکی بنام **"."** و دیگری بنام ".." که "." به معنی مسیر جاری و ".." به معنی مسیر عقب تر از این مسیر یا والد این مسیر میباشد .

حالا فرض کنید در مسیر مورد نظر فایلی با نام **".htpasswd"** وجود داشته باشد . دیدن نام این فایل بدون گزینه **"ls –a"** امکان پذیر نمیباشد .

برای مشاهده لیست فایلهای داخل یک مسیر میتوانیم از گزینه **"ls -la"** که نام فایل ها و دایرکتوری ها را با جزییات بیشتر و همچنین نام فایلهای پنهان را نیز مشاهده نماییم.

8- ایجاد حذف و یا انتقال فایل ها در لینوکس

در این بخش قصد دارم به چگونگی استفاده از دستوراتی که برای ایجاد یک فایل جدید یا حذف فایل یا انتقال فایل به مسیر دیگر صحبت کنم در این راستا لیستی از دستورات به شرح زیر بکار میرود

- دستور **touch** که برای ایجاد یک فایل خالی بکار میرود
- دستور **rm** یا **remove** که برای حذف یک فایل و دایرکتوری بکار میرود
- دستور **mv** یا **move directory or file** که برای انتقال فایل یا دایرکتوری ویا 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 بصورت تو درتو یا **recursive** انجام دهیم

حالا به توضیح هر یک از دستورات بالا می پردازم

- دستور **touch**

این دستور برای ایجاد یک فایل با سایز صفر یا یک فایل خالی استفاده میشود یعنی اگر فایلی بنام **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 مثلا فایلی بنا **test** داشته باشیم و دستور **touch test** را بزنیم اتفاقی که می افتداین 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 میخواهم فایلی با نام مثلا **My File** داشته باشم در هنگام استفاده از دستور **touch My File** بجای یک فایل دو فایل با نامهای **My** و **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 بصورت نام فایل لحاظ خواهد شد .

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

مثلا **mkdir Mydir1** یعنی پوشه ای 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 دایرکتوری ها هم صدق میکند .

- دستور **mv** که برای انتقال یا تغییر نام یک فایل یا دایرکتوری بکار میرود

مثلا اگر دستور **mv myfile mydir1** را بدهیم یعنی اگر فایلی بنام **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 خصوصا زمانیکه این انتقال از یک ماشین به ماشین دیگر و تحت شبکه انجام میشود . از آنجا که 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 وممکن است براحتی 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 بکار میرود یعنی اگر دستور **mv myfile newfile** را اجرا کنیم اتفاقی که می افتد این است که فایل **myfile** به **newfile** تغییر نام میدهد .

- دستور **cp** یا **copy** که برای کپی کردن فایل و دایرکتوری از یک مسیر به مسیر دیگر بکار میرود
مثلا **cp myfile mydir1** یعنی فایلی با نام **myfile** را به شاخه **mydir1** در مسیر جاری کپی کن
یعنی با فرض اینکه در مسیر جاری پوشه ای با نام **mydir1** وجود داشته باشد و 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 دایرکتوری ها بکار میرود
مثلا اجرای دستور **rm myfile** یعنی اگر فایلی با نام **myfile** وجود دارد آن را پاک کن
حالا هنگام پاک کردن ممکن است سیستم عامل از ما سوال کند که آیا مطمئن هستی که فایل پاک شود و شما با دادن پاسخ بله یا خیر که با **"y"** یا **"n"** آنرا 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

مثلا **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** پا ک 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 وقت گیر است بنابراین از علائم مخصوص برای خلاصه کردن استفاده میکنیم چگونه

مثلا اگر میخواهیم هر چی فایل در مسیر **/home/oracle/mydir1** وجود دارد را پاک کنیم از دستور **rm -f /home/oracle/mydie1/*** استفاده میکنیم و سپس میتوانیم دستور **rmdir /home/oracle/mydir1** را اجرا کنیم

حالا زمانهایی پیش میآید که مطمئن به پاک کردن یک دایرکتوری و فایل های داخل آن هستیم در اینحالت از دستور "rm –rf /home/oracle/mydir1" استفاده میکیم دراین حالت کل پوشه mydir1 به همراه تمام فایلها و پوشه های داخل آن به یکباره پاک خواهند شد . خب این دستور کار ما را راحت میکند اما خطری نیز ما را تحدید خواهد کرد و اینکه در بسیاری از موارد مشاهده کرده ام که کاربران به اشتباه مسیری را حذف میکنند که از اطلاعات آن مسیر نیز نسخه پشتیبانی ندارند و در این هنگام کلیه داده های ما در این پوشه از بین خواهد رفت . بنابراین بهتر است زمانیکه میخواهیم اطلاعات یک پوشه را بطور کامل پاک کنیم حتما از قبل برای آن نسخه پشتیبانی تهیه کنیم و سپس اقدام به حذف پوشه نماییم زیرا در این حالت اگر اشتباها چیزی را حذف کرده باشیم قابل بازیابی میباشد بدین منظور از دستور "cp -R mydir1 mydir1.backup" میتوانیم استفاده کنیم . در این دستور گفته ام که از پوشه mydir1 عینا 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 کرده ام که افرایکه خصوصا روی برنامه نویسی کار میکنند و کدهای یک ماه یا یک هفته ای خود را بصورت نسخه پشتیبان تهیه نکرده اند گاها 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 بصورت استفاده از علایم خاص میباشد یعنی اگر من میخواهم تمام فایلهایی که انتهای نامشان مثلا حروف txt دارد را حذف کنم از علامت "*" بجای کلیه نام فایل به همراه حروف txt استفاده میکنیم یعنی چی ؟

فرض کنید فایلهای با نام "file1.txt" , "file2.txt" , "file3.txt" , "file4.txt" داریم و میخواهیم همه این فایلها را حذف کنیم دراین صورت با دستور "rm *.txt" کلیه فایلهایی که اسم آنها هرچی باشه ولی پسوند انتهای آن ".txt" باشد را حذف خواهد کرد .

مثال بعدی :

فرض کنید فایلهایی با نام "1.cc" , "2.cc" , "3.cc" , "4.cc" داریم در این صورت برای پاک کردن این فایلها از دستور

"rm ?.cc" یعنی فایلهایی که نامشان یک حرفی است و با پسوند ".cc" خاتمه پیدا میکند را پاک کن

علامت ؟ بجای یک کاراکتر یا یک حرف بکار میرود .

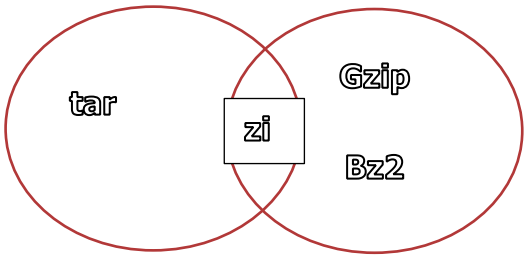
مثال :

فرض کنید فایلهایی با نام "Ahamah" , "ali" داریم و میخواهیم آنها را به مسیری دیگر کپی کنیم برای این حالت میتوانیم از دستور "cp [aA]* /home/oracle/mydir1" استفاده کنیم . این دستور میگوید از مسیر جاری هر فایلی که نام آن با حرف **a** کوچک یا **A** بزرگ شروع میشد و انتهای آن هر چی میخواهد باشد را به مسیر **/home/oraclr/mydir1** کپی کن

توجه داشته باشید که کلیه موارد و حروف خاص مثل "*" یا "؟" در کلیه دستورات ذکر شده بالا قابل استفاده میباشد و باعث میشود که عملیات حذف و انتقال و یا کپی کردن را بصورت دسته جمعی روی فایلها و دایرکتوریهایمان انجام دهیم و در زمان انجام صرفه جویی میکند . زیرا زمانیکه شما با تعداد مثلا 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 علایم برای خلاصه کردن عمل خود و صرفه جویی در زمان استفاده نماییم.

- نکته : استفاده از گزینه "-r" به همراه دستورات **cp , mv , rm** به این معنی است که عملیاتی که قرار است انجام دهید را بصورت تو در تو در مسیر مورد نظر و کلیه فایلها و پوشه های داخل آن مسیر انجام بده

یعنی اگر در مسیر **/home/oracle/mydir1/mydir2/mydir3** چندین پوشه و زیر پوشه وجود داشته باشد با استفاده از گزینه "-r" میتوانیم به یکباره و تو در تو کلیه فایلها و دایرکتوری مسیر **/home/oracle/mydir1/mydir2/mydir3**را به یکباره حذف یا انتقال یا کپی نماییم



Archiving **compression**

در شکل بالا دو دایره می بینیم یکی برای گرفتن نسخه پشتیبان که در آن از دستور **tar** که مخفف کلمه **Tape Archive** می باشد و در دایره بعدی دستوراتی که برای فشرده سازی بکار میروند مانند **gzip , bz2** نشان داده شده است . در شکل بالا نقطه اشتراک میان این دو قسمت دستور **zip** میباشد . در سیستم عامل لینوکس عموما برای فشرده سازی 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 فشرده سازی نسخه پشتیبان مانند **gzip , bz2** بکار رود.

آنچه در این فصل میخواهم به آن اشاره کنم عبارتست از استفاده از دستور **tar** برای تهیه نسخه پشتیبان یا **backup** و چگونگی استفاده از دستورات فشرده سازی **gzip , bz2** برای فشرده کردن نسخه های پشتیبان .

از آنجا که برای هر فشرده سازی می بایست نحوه بازآوری فایل فشرده را نیز بدانیم پس به ازاء هر دستور فشرده سازی ، دستوری برای بازگردانی آن نسخه از حالت فشرده به حالت غیر فشرده نیز وجود دارد که در این فصل به تفصیل در باره آن صحبت خواهم کرد.

دستور فشرده سازی	gzip	bz2	zip
دستور غیر فشرده سازی	gunzip	Bunzip2	unzip

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

اما برای تهیه نسخه پشتیبان از فایل های و یا پوشه های داخل سیستم عامل میتوانیم از دستور **tar** استفاده نماییم .

برای استفاده از دستور **tar** حالتهای زیر را میتوانیم بکار بریم :

- 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 تهیه شده از دستور **tar -x**

استفاده میکنیم .

دستور **tar** را میتوانیم با استفاده از گزینه های فشرده سازی نیز بکار بریم . بدین منظور در هر یک از حالات سه گانه بالا میتوانیم گزینه **z–** و یا **j-** را بکار ببریم .برای روشن شدن مطلب مثال های زیر را میزنم .

مثلا فرض کنید که میخواهم از مسیر خانه شخصی خودم در لینوکس یا به اصطلاح **home directory** خودم ، یک نسخه پشتیبان تهیه کنم برای این کار از دستور زیر استفاده مینمایم

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

در دستور بالا میگویم با استفاده از گزینه های **c-** یعنی ایجاد نسخه پشتیبان و قرار دادن این نسخه پشتیبان در فایلی در مسیر **/tmp** و با نام **home_oracle.tar** از مسیر **/home/Oracle** نسخه پشتیبان تهیه کن

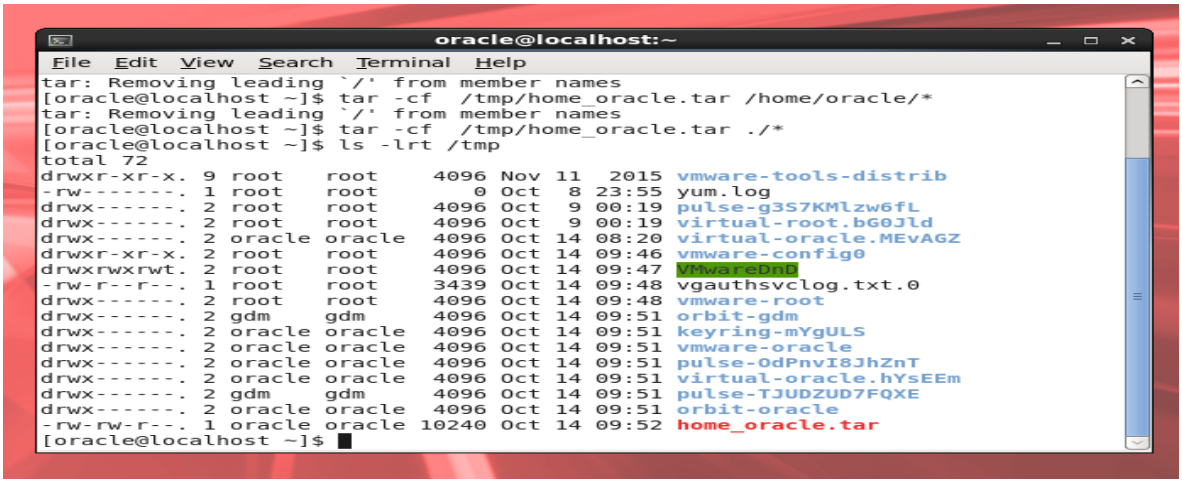
دقت کنید که در دستور **tar** ابتدا مقصد یا مکانی که میخواهیم فایل پشتیبان در آنجا ذخیره گردد را مینویسیم و بدین منظور از گزینه **f–** در دستور **tar** استفاده کردیم این گزینه میگوید که نسخه پشتیبان مرا در فایلی با مشخه ای که در دستور آورده ام قرار بده.

دستور **tar -cf /tmp/home_oracle.tar /home/Oracle** در واقع 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 بعد ها زمانیکه به سیستم مراجعه میکند و اینکه بداند این فایل چه بوده است بهتر است نام های 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 لذا نام فایل ها در دستور **tar** را با پسوند **tar** انتخاب میکنیم .

اتفاقی که در دستور **tar** می افتد این است که کلیه فایلها و دایرکتوری ها را بصورت یک فایل و با فرمت **tar** در مکانی ذخیره میکند .

شکل زیر نمایش دستور **tar -cf /tmp/oracle.tar ~/*** در این دستور بجای استفاده از مسیر **/home/oracle** با فرض اینکه در مسیر **/home/oracle** قرار دارم و یا اینکه با نام کاربری **oracle** وارد سیستم شده ام از علامت **~** به معنی **home directory** استفاده میکنم



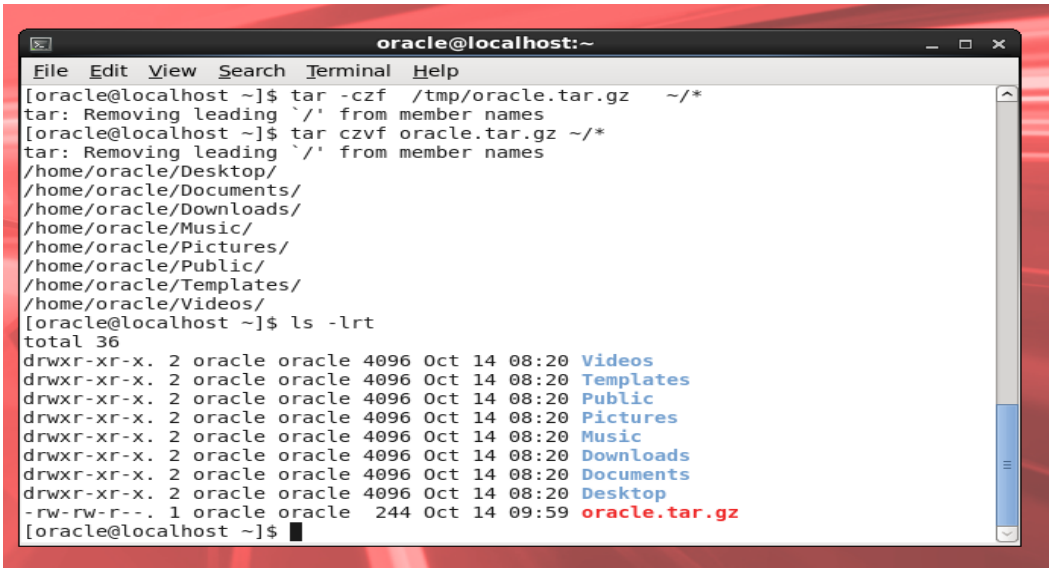
حالا فرض کنید که تعداد فایلها و دایرکتوریهای شما زیاد باشد و فایل پشتیبانی که با دستور **tar** میگیرید بزرگ بوده و فضای کافی برای ذخیره سازی ندارید ، در چنین حالتی با استفاده از گزینه های فشرده سازی ، این نسخه **tar** را فشرده میکنیم تا حجم فایل تولید شده کم گردد . بدین منظور از دستور **tar** با گزینه **-z** یا **-j** بصورت زیر استفاده میکنیم :

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

در دستور بالا میگویم که از مسیر **/home/oracle** یک نسخه پشتیبان بگیر و آن را فشرده کن و در فایلی بنام **home_oracle.tar.gz** در مسیر **/tmp** قرار بده .

دقت کنید که نام فایل را بر اساس گزینه **-z** بصورت **home_oracle.tar.gz** انتخاب کردم که به من یادآوری میکند که این فایل از نوع **tar** فایل بوده و بصورت فشرده و از نوع **gzip** می باشد .

شکل زیر نحوه استفاده از گزینه فشرده سازی را در دستور **tar** نشان میدهد:



حال فرض کنید که میخواهیم از مکانیزم **bz2** برای فشرده سازی استفاده کنیم بدین منظور با استفاده از دستور

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

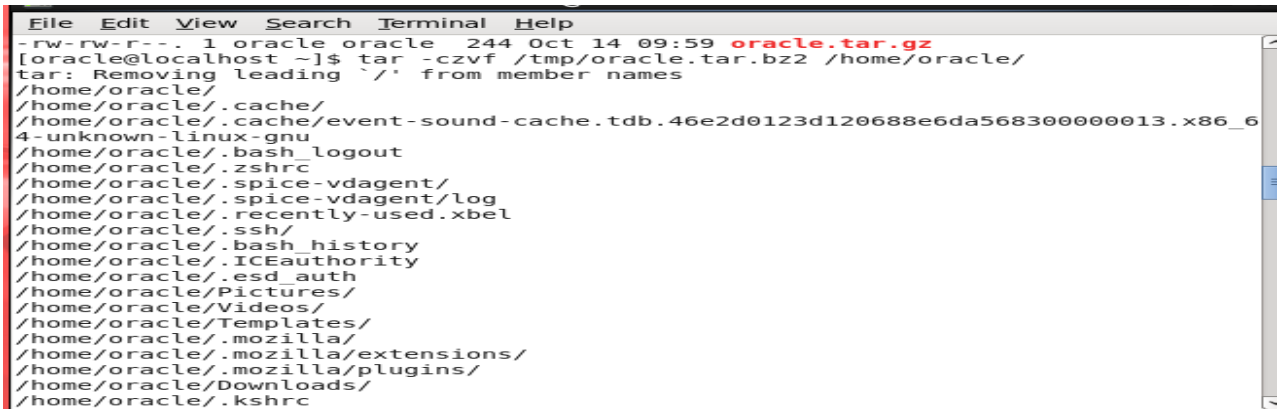
حالا چنانچه هنگام گرفتن نسخه پشتیبان بخواهیم محتوای مسری که در حال پشتیبان گیری است روی صفحه مانیتور نمایش داده شود از گزینه **-v** استفاده میکنیم . با استفاده از این گزینه کلیه فایلها و دایرکتوریهای مسیر مورد نظر بصورت همزمان که در فایل مقصد پشتیبان گیری میشوند روی صفحه نمایش داده میشوند .

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



جهت نمایش محتوای یک **tar file** از گزینه **t-** استفاده میکنیم . مثلا فرض کنید که در دستور بالا نسخه پشتیبانی تهیه کرده ایم حالا میخواهیم ابتدا محتوای آنرا ببینم و بعد به استخراج آن بپردازیم بدین منظور از دستور زیر استفاده میکنیم :

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

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

با استفاده از دستورات بالا میتوانیم محتوای فایل پشتیبانی که با فرمت های **gzip ,bz2** قبلا ایجاد کرده بودیم را تماشا کنیم .

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

نکته : در هنگام استخراج یک نسخه پشتیبان ، حتما در ابتدا به مکای غیر از مکان اصلی استخراج بروید مثلا فرض کنید که من میخواهم فایل **home_oracle.tar.gz** را استخراج کنم بنابراین در ابتدا به ساکن به مسیر **/home/oracle** نمیروم که درآنجا این استخراج را انجام دهم زیرا اگر این کار را انجام دهم کلیه فایلهای نسخه پشتیبان روی فایلهای قبلی نوشته می شود و کلیه فایلهای قبلی از بین خواهند رفت بنابراین در ابتدا به ساکن یک پوشه مثلا با نام دلخواه **mybackup** در مسیری دلخواه حالا یا در مسیر **/home/oracle** یا در مسیری مثل **/tmp** میسازم و سپس نسخه پشتیبان را در آنجا استخراج میکنم و پس از اطمینان از صحت فایلهای استخراج شده میتوانم این فایلها 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 -xzvf /tmp/home_oracle.tar.gz

$ tar -xjvf /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** و 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 بصورت زیر عمل میکنیم :

$ mkdir /tmp/mybackup

$ cd /tmp/mybacku

$ tar –xzvf /tmp/home_oracle.tar.gz

ویژگی دستور **tar** در این است که **owner** و **group** و اجازه های دسترسی فایلها را در خود ذخیره میکند . این مسئله بشدت در زمان استخراج نسخه پشتیبان مهم می باشد . زیرا در زمان استخراج فایلها و دایرکتوریها اگر مثلا از دستور کپی استفاده کنیم **owner , group** فایلها بر اساس نام و گروه شخصی که درحال انجام استخراج است تغییر میکند و همچنین اجازه های دسترسی فایلها نیز بر اساس اجازه های 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 اجازه های دسترسی فایلها را ذخیره و در هنگام استخراج عینا بصورت اول بازیابی نماییم.

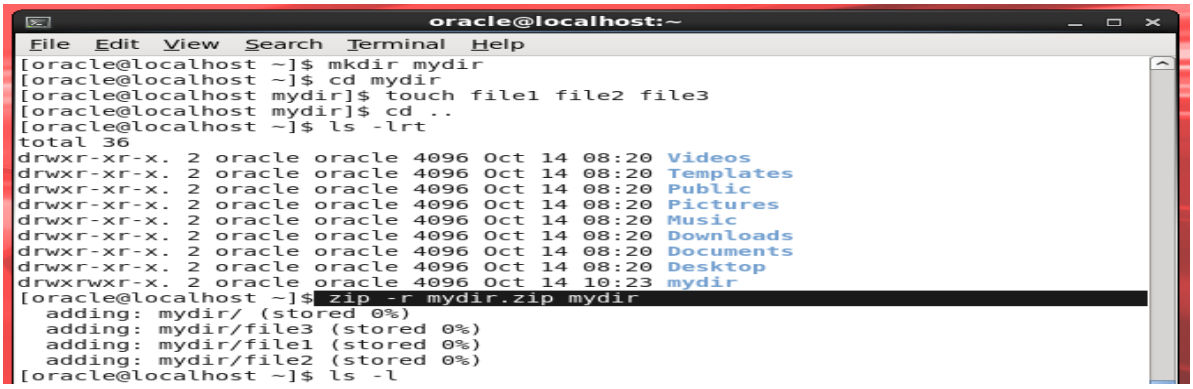
در واقع دستور **tar** بنوع مانند کپی کردن فایلها در جای دیگر میباشد ولی مکانیزم خاصی دارد که میتواند خصوصیاتی از فایل را که در آن **owner , group** و اجازه های دسترسی فایل را نیز در خود ذخیره کند .

یکی دیگر از روشهای فشرده سازی استفاده از دستور **zip** میباشد . مثلا فرض کنید میخواهیم یک فایل را بصورت 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** به تنهایی قادر به انجام عمل فشرده سازی بصورت تو در تو نمیباشد بعبارتی وقتی که از دستور **zip** جهت فشرده سازی یک پوشه استفاده میکنیم و مثلا دستور **zip dir.zip dir_name** را میزنیم در اینحالت دستور **zip** متوجه نمیشود

که **dir_name** یک دایرکتوری است که باید محتوای داخل آن را فشرده کند و بنابراین عملا کاری انجام نمیدهد

بدین منظور از گزینه **-r** به همراه دستور **zip** استفاده می نماییم که بصورت زیر نمایش داده ام

$ zip -r dir.zip dir_name



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.zip mydir** این دایرکتوری و محتوای داخل آن را فشرده کرده ام اما چون فایلهای من خالی میباشد لذا مقدار **0**٪ را در شکل بالا مشاهده میکنیم. اما اگر فایلها و دایرکتوریهای مسیر انتخابی خالی نباشند این میزان درصد نیز متفاوت خواه بود و صرفا 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 سیستم عامل لینوکس بدانیم مثلا **tape device** های از نوع اسکازی معمولا با نام **st0** یا **scsci tape** نامیده میشوند که عدد **0** 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**ی بنام **/dev/st0** از کلیه محتوای مسیر **/home/oracle**

و در ضمن فشرده هم کردیم .

/dev/st0 به مسیر **tape device** روی سیستم عامل اشاره میکند . از آنجا که در سیستم عامل لینوکس کیلیه **device** های جانبی در مسیر **/dev** قرار دارند و نیز بخاطر اینکه **tape** نیز یک وسیله خارجی متصل به سیستم عامل است لذا نام آن در این مسیر نگهداری میشود بعبارتی سیستم عامل با دیدن نام **/dev/st0** بدنبال **tape device** ی از نوع اسکازی میگردد و اگر این دستگاه به ماشین لینوکس ما متصل باشد فایل پشتیبان را روی آن منتقل میکند .

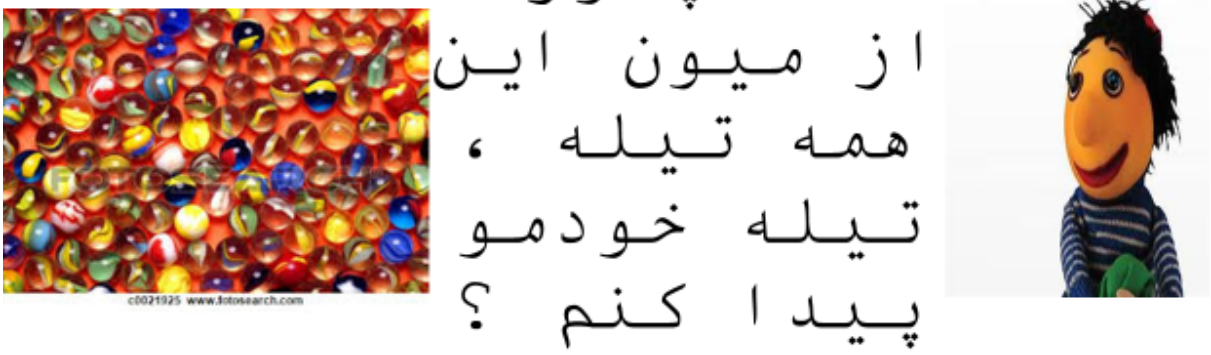
در مورد تهیه نسخه پشتیبان روی **tape** با گزینه های متفاوتی روبرو هستیم مثلا میدانیم در این روزها سایز و فضای **tape device** ها زیاد است و به ترابایت میرسد بنابراین اگر میخواهیم نسخه پشتیبانی روی **tape** تهیه کنیم که حجم آن کمتر از حجم **tape** میباشد بهتر است از فضای **tape** بهینه استفاده نماییم در این حالت از گزینه **/hom/oracle** **/dev/st0n** **-czf** **tar** استفاده می کنیم . دقت کیند **/dev/st0n** به سیستم عامل می گوید 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 حالتیکه از **/dev/st0** استفاده کنیم بعد از عمل پشتیبان گیری **tape** به عقب و به ابتدا باز میگردد و سرخط می ایستد اما در حالتیکه از گزینه **/dev/st0n** استفاده کنیم مثل این است که نوار را بخش بندی کرده ایم و هر قسمت از نسخه پشتیبان را روی یک بخش از نوار مغناطیسی ذخیره نموده ایم . 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 آنجا که این کتاب مربوط به دانستنیهای لینوکس مقدماتی میباشد لذا در این خصوص بیشتر صحبت نمیکنم و صرفا خواستم تا نسبت به گزینه های مختلف پشتیبان گیری آشنایی پیدا نمایید . استفاده از دستورات نوار مغناطیسی خود بخش مجزایی را طلب میکند .

10- چگونه داخل فایلها و دایرکتوری های لینوکس جستجو کرده و متن مورد نظر را بیابیم

با داستانی موضوع را شروع میکنم

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



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

خب ممکن است بگوییم می توانیم تیله ها را بر اساس اندازه مرتب کنیم و سپس بر اساس رنگ دسته بندی کنیم و اینطوری دسته های کوچکتری پیدا کرده و بنابراین پیدا کردن تیله کلاه قرمزی در این دسته های کوچک تر راحتتر خواهد بود . بدین منظور در لینوکس دستور بسیار قدرتمندی بنام grep داریم که بصورت زیر از آن استفاده میکنیم:

$ grep "تیله من" "همه تیله ها"

خط بالا یک مثال است که بجای عبارت "تیله من " و "همه تیله ها" در واقع عبارت مورد جستجو و نام کلیه فایلهایی که میخواهیم این عبارت در آن جستجو گردد را وارد میکنیم .

بنابراین مثلا اگر بخواهیم بدنبال 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 مورد نظر ما در کدام فایل است بنابراین عبارت مورد نظر را در همه فایلها جستجو میکنیم و این عمل را با دستور **grep "abc" file*.txt** انجام می دهیم . در این مثال میخواهم بدنبال 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 بگردم.

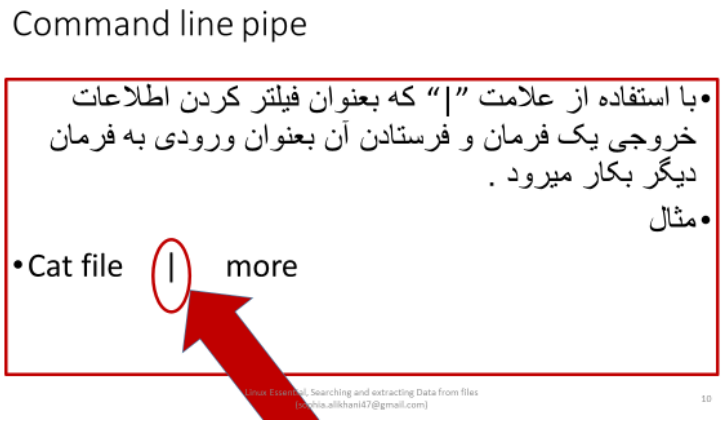
اما روشهای دیگر جستجو این است که مانند مثال بالا که برای کلاه قرمزی استفاده کردیم ابتدا محتوای یک فایل را مرتب کنیم و براساس مقادیر مرتب شده بر اساس مثلا رنگ تیله یا اندازه تیله ها بدنبال تیله مورد نظر بگردیم . این عمل در سیستم عامل لینوکس نیز قابل انجام است و با استفاده از ترکیب چند دستور با هم که به آن عمل **piping** یا لوله کشی میگویند انجا می شود در واقع در مثال کلاه قرمزی مثل اینکه ما از چند سبد سوراخ دار با اندازه های متفات برای جدا سازی تیله های درست از تیله های کوچک استفاده کنیم .

عمل **pipe** یا لوله کشی در لینوکس باعث میشود که خروجی اجرای یک دستور بعنوان ورودی دستور بعدی بکار رود دقیقا مثل زمانیکه کل تیله های کلاه قرمزی را از یک آبکش اولیه عبور دادیم و نتیجه تیله های بدست آمده 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 مورد نظر را پیدا نماییم.

بدین منظور به روش زیر عمل میکنیم :

$ cat file1.txt | grep abc



در شکل بالا خروجی دستور **cat** را به دستور **more** داده ایم که بتوانیم محتوای فایل را صفحه به صفحه مشاهده نماییم.

اینها روشهایی از **pipe** کردن دستورات در لینوکس میباشند .

میخواهم در این بخش مجموعه ای از دستورات مفید لینوکس جهت جستجو متنی در فایل یا فایلهای مورد نظر را بصورت اختصار توضیح دهم .

دستور **tail**

این دستور برای نمایش خطوط انتهایی یک فایل بکار میرود و حالب است بدانید که این دستور به همراه گزینه **–f** جهت مانیتورینگ لاگ فایلهای سیستم بسیار مفید است و نحوه استفاه آن بدین صورت است .

مثلا اگر بخواهیم یکی از لاگ فایلهای مهم سیستم عامل لینوکس بنام **messages** که در مسیر **/var/log** قرار دارد را مانیتور کنیم از دستور **tail -f /var/log/messages** استفاده میکنیم .

اگر در دستور **tail** تعداد خطوطی که میخواهیم روی صفحه نمایش ببینم را ندهیم سیستم عامل بطور پیش فرض ده خط آخر فایل را نمایش میدهد اما میتوانیم تعداد خطوط مورد نیاز را هم در این دستور بدهیم مثلا بگویمم **tail -n 50 /var/log/messages** یعنی **50** 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 بطور پیش فرض **10** خط ابتدای فایل را نمایش میدهد و چنانچه عدد بدهیم به تعداد عدد داده شده از خطوط ابتدای فایل را به ما نمایش میدهد.

مثلا **tail -n 20 /var/log/messages** یعنی **20** خط اول این فایل را برایم نمایش بده .

دستور **less**

Less دستوری است که باعث میشود خروجی یک فرمان بصورت صفحه به صفحه نمایش داده میشود علاوه بر آن این امکان را به ما میدهد که بتوانیم روی صفحات مرور کنیم یعنی به صفحه بالایی و پایینی حرکت کنیم .

مثلا اگر میخواهیم محتوای فایل **messages** در مسیر **/var/log** را مشاهده کنیم با دستور **less /var/log/messages** این کار را براحتی میتوانیم انجام دهیم

دستور **find**

این دستورمکان مستقر یک فایل را در سیستم عامل لینوکس نمایش میدهد .

مثلا اگر میخواهیم بدنبال فایلی بگردیم که اسم آن را کامل نمیدانیم اما میدانیم این فایل در انتها با پسوند **.tar.gz** تمام شده است ولی نمیدانیم کجای سیستم ذخیره شده است با استفاده ازدستور **find** آن را جستجو میکنیم

مثلا مینویسیم **find / -name "*.tar.gz"**

در این دستور مسیری که میخواهم بدنبال فایل یا فایلهای مورد نظرم بگردم را وارد کرده ام که در مثال بالا علامت **"/"** یعنی از ریشه اصلی بدنبال فایل مورد نظر بگرد و بدنبال 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 ای مورد نظر ما میگردد و در صورت پیدا کردن نام مسیر فایل را روی صفحه نمایش میدهد .

دستور **sort**

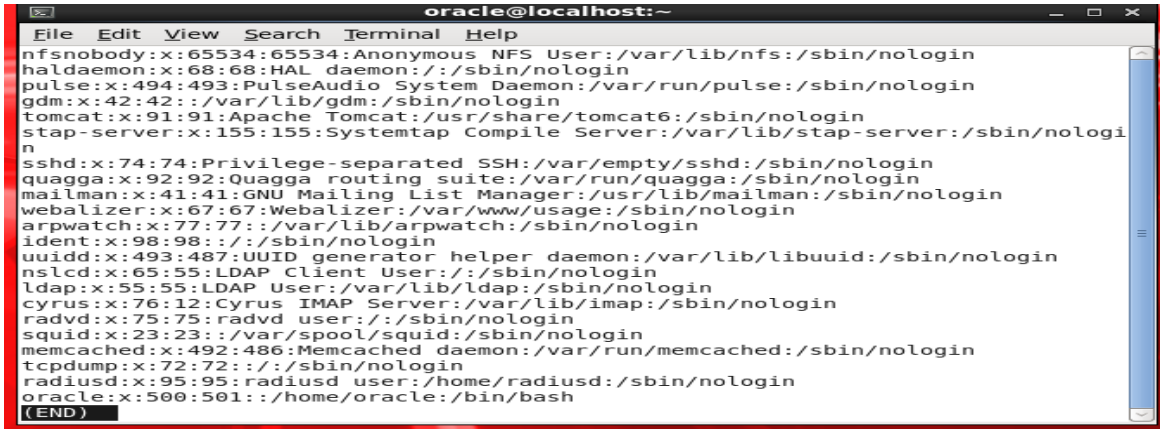
Sort برای مرتب سازی محتوای یک فایل بکار میرود و دقیقا مثل حالتیکه کلاه قرمزی میخواهد تیله های مورد نظر خود را در میان انبار تیله ها پیدا کند میتوانیم این تیله ها را بر اساس اندازه یا رنگ مرتب کنیم که این عمل یک نوع مرتب سازی میباشد . دستور **sort** دقیقا عمل مرتب سازی را روی محتوای فایلها انجام می دهد . مثلا وقتی میگویم **sort myfile** در این حالت فایل **myfile** بر اساس حروف الفبا از **a-z** مرتب شده و روی صفحه این محتوای مرتب شده نمایش داده میشود .

دستور **cut**

این دستور برای قیچی کردن تکه ای از محتوای یک فایل بکار میرود و قابلیتهای زیادی را در اختیار ما قرار میدهد . مثلا فرض کنید که از یک فایل داده ای خصوصا در فایلهای مربوط به ذخیره سازی پایگاه های اطلاعاتی ما با انبوهی از اطلاعات بصورت ستون به ستون روبرو هستیم و حالا فرض کنید که ما ستون خاصی یا دسته خاصی از این داده ها را نیاز داریم که در این زمان با استفاده از دستور **cut** میتوانیم ستون یا ستونهای مورد نیاز را از فایل کنده و در جای دیگر ذخیره نماییم . در دستور **cut** دو گزینه مهم داریم یکی جداکننده ستونها که مشخصا میگوییم از چه علامتی برای جدا سازی ستونها استفاده کرده ایم و یکی شماره ستون یا ستونهایی که میخواهیم قیچی کنیم.

مثلا فرض کنید فایل **/etc/passwd** در سیستم عامل که محل ذخیره سازی نام و مشخصات کابران سیستم میباشد را میخواهیم بصورتی قیچی کنیم که فقط نام کاربران از آن پاره شده و این نام ها در فایل دیگری ذخیره گردد . بدین منظور به صورت زیر عمل میکنیم .

اولا قسمتی از محتوای یک فایل **/etc/passwd** را در زیر نمایش میدهم تا دقیقا ببینم که چه علامتی بین ستون ها جداکننده است :



در شکل بالا نمایی از محتویات فایل **/etc/passwd** را می بنیم که هر ستون با علامت ":" از ستون بعدی جدا شده است . ستون اول نام کاربران میباشد بنابراین اگر میخواهیم ستون اول را از این فایل قیچی کرده و محتوا را در فایلی مثلا بنام **users** ذخیره کنیم با دستور زیر این عمل انجام میشود

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

در فرمان بالا میگوییم که از فایل **/etc/passwd** فیلد **1** که با **f1–** نشان داده ایم جدا کند و خروجی آنرا در فایلی بنام **/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 میکنم

$ cut -d":" -f1,3,4 /etc/passwd

در این حالت خروجی اجرای فرمان بالا بصورت زیر خواهد بود

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 گزینه های مختلفی دارد مثلا اگر بگویم **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 بیابیم .

در جستجوی عبارات مورد نظر در داخل فایلها یکسری علائم داریم که برای سیستم عامل معنی دار هستند که این علائم عبارتند از :

- ^ این علامت به معنی ابتدای فایل میباشد . مثلا وقتی میگوییم برو داخل فایل و بدنبال تمام عباراتی باش که در ابتدای آن مثلا حرف **a** قرار دارد را نشان بده در این صورت با دستور **grep ^a myfile** این عمل را انجام میدهیم .
- $ این علامت برای نمایش انتهای فایل است و زمانیکه بدنبال متنی هستیم که مثلا در انتهای فایل یک حرف بخصوصی دارد میتوانیم از این علامت استفاده کنیم مثلا میتوانیم بگوییم برو داخل فایل و بدنبال تمام عباراتی باش که انتهای آنها با **txt** تمام میشود را پیدا کن که با دستور **grep $txt myfile** میتوانیم این جستجو را انجام دهیم .

چند مثال خوب

1- فرض کنید که در سیستم بدنبال لاگ فایلهای بزرگی که مثلا سایز آنها از **10** مگابایت بیشتر بوده و بیشتراز یکماه است که در سیستم دست نخورده باقی مانده و نام آنها با پسوند **log** تمام میشود را بیابیم د راین حالت با استفاده از فرمان **find** و گزینه های آن این کار را انجام میدهیم

```
$ find /  -atime  30  -size 10M  -name "*log"
```

با مطالعه **man find** یا **Info find** میتوانید گزینه های مختلف دستور **find** را بیابید .

2- فرض کنید بدنبال فایلهای فقط خواندنی یا **read only** سیستم هستیم . با دستور زیر این فایلها را جستجو میکنیم

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

3- فرض کنید بدنبال پوشه های خالی در مسیر **/tmp** میگردیم .

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

4- فرض کنید که در مسیر **/home/oracle** بدنبال فایلهای مخفی میگردیم .

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

5- لیست فایلهایی را میخواهیم که بین **50** تا **100** روز پیش تغییر کرده اند

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

6- لیست فایلهایی که یکساعت قبل تغییر کرده اند را میخواهیم

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

7- بدنبال تمام فایلهایی هستیم که مالک آنها کاربر **oracle** باشد

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

8- لیست تمام فایلهایی که سایز آنها بین **50** تا **100** مگابایت است را میخواهیم

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

9- بدنبال عبارت **"php"** درفایلهای مختلف میگردیم .

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

10-بدنبال فایلهایی با پسوند **php** در سیستم میگردیم.

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

11- داده های ما در سیستم عامل کجا ذخیره میشوند

در این بخش میخواهیم به این موضوع بپردازیم که داده های ما در سیستم عامل لینوکس کجا نگهداری میشوند.

بدین منظور بهتر است طرحی از ساختار پوشه های سیستم عامل لینوکس را توضیح دهم .

مسیر **/etc**

در سیستم عامل لینوکس مسیری با نام **/etc** داریم که محل ذخیره فایل های پیکربندی سیستم عامل می باشند

یعنی چی ؟

در سیستم عامل لینوکس فایلهایی هستند که اطلاعاتی در ارتباط با نام ماشین و یا اطلاعات شبکه را در خود نگهداری میکنند ای اطلاعات برای سیستم عامل بسیار مهم میباشند و هنگامیکه 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 کتابخانه ای یا **library** های سیستم عامل می باشند . فایل های کتابخانه ای در واقع کدهای نوشته شده و اجرا شده ای هستند که فایلهای اجرایی سیستم به آنها نیاز دارند . در سیستم های کنونی که از معماری **64** بیتی استفاده میکند علاوه بر **/usr/lib** , **/lib** مسیر **/usr/lib64** , **/lib64** نیز اضافه شده است .در واقع **/usr/lib** , **/lib** محل نگهداری **library** های **32** بیتی و **/usr/lib64**, **/lib64** محل نگهداری **library**های **64** بیتی سیستم باشد .

فایلهایی که در این مسیر قرار میگیرند خیلی شبیه به فایلهای **dll.** در ویندوز میباشند.

مسیر **/usr/bin** **/bin**

این مسیر محل نگهداری فایلهای اجرایی سیستم مانند دستورات مورد استفاده سیستم مانند دستور **ls , find,** **wc** و غیره میباشند

مسیر /var/log

در مسیر **/var** کلیه فایلهایی که سایز آنها در طول حیات و روشن بودن سیستم تغییر میکند نگهداری میشود مانند لاگ فایلهای سیستم

از جمله این لاگ فایلها در مسیر **/var/log/** می باشد که محل نگهداری لاگ فایلهای سیستم عامل میباشد . مثلا فایل مهم **messages** در مسیر **/var/log** در این مسیر نگهداری میشود

مسیر /home

محل نگهداری فایلهای مربوط به کابران موجود در سیستم عامل که قادر به وارد شدن به سیستم می باشند است .

مثلا کاربر **oracle** که اجازه ورود به سیستم را دارد دارای خانه شخصی **/home/oracle** می باشد که میتواند فایلهای خود را در آنجا ذخیره و نگهداری کند

بنابراین متوجه میشویم که داده های ما در سطح سیستم عامل بر طبق نظمی خاص و براساس موضوعی که آن داده در خود ذخیره میکند در مکانی خاص در سیستم عامل نگهداری میشود.

در این قسمت میخواهم مفهومی بنام **process** را توضیح دهم.

زمانیکه ما فرمان **ls** را اجرا میکنیم در واقع یک فایل اجرایی را اجرا میکنیم . این فایل اجرای یا همان **ls** در واقع یک کپی از فایل **ls** میباشد که کاربر در لحظه مورد نیاز آنرا بکار میگیرد . حالا اگر همزمان **10** نفر این فرمان را اجرا نمایند انگار **10** نسخه کپی از فرمان **ls** در اختیار این ده نفر قرار گرفته است . این نسخه های کپی از برنامه اجرایی که در لحظه اجرا میشوند را **process** میگویند . بنابراین **process** چیزی بیش از یک برنامه اجرایی نیست . منتها نسخه ای از یک برنامه اجرایی ای میباشد که در لحظه در حال استفاده شدن است و بعبارتی در حافظه سیستم بارگذاری شده است .

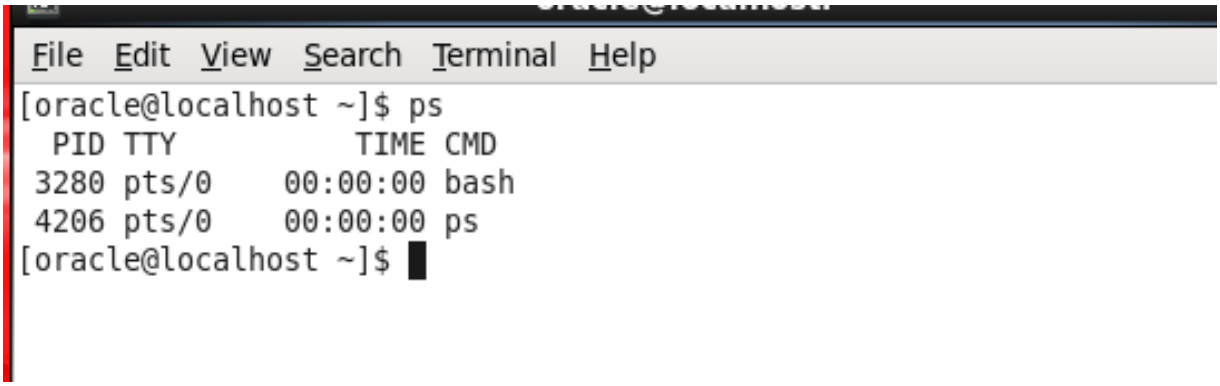
بنابراین یک **process** برنامه اجرایی ای میباش که در حافظه قرار گرفته است .

کاربران سیستم عاملهای لینوکسی متیوانند هم زمان چندین دستورو یا یک دستور را بصورت چندین مرتبه اجرا نمایند . لذا سیستم عامل لینوکس حاوی پاره ای از دستورات جهت مدیریت و نمایش **process** ها میباشد . از جمله این دستورات دستور **ps** یا **process status** یا همان نشان دادن وضعیت **process** ها می باشد .

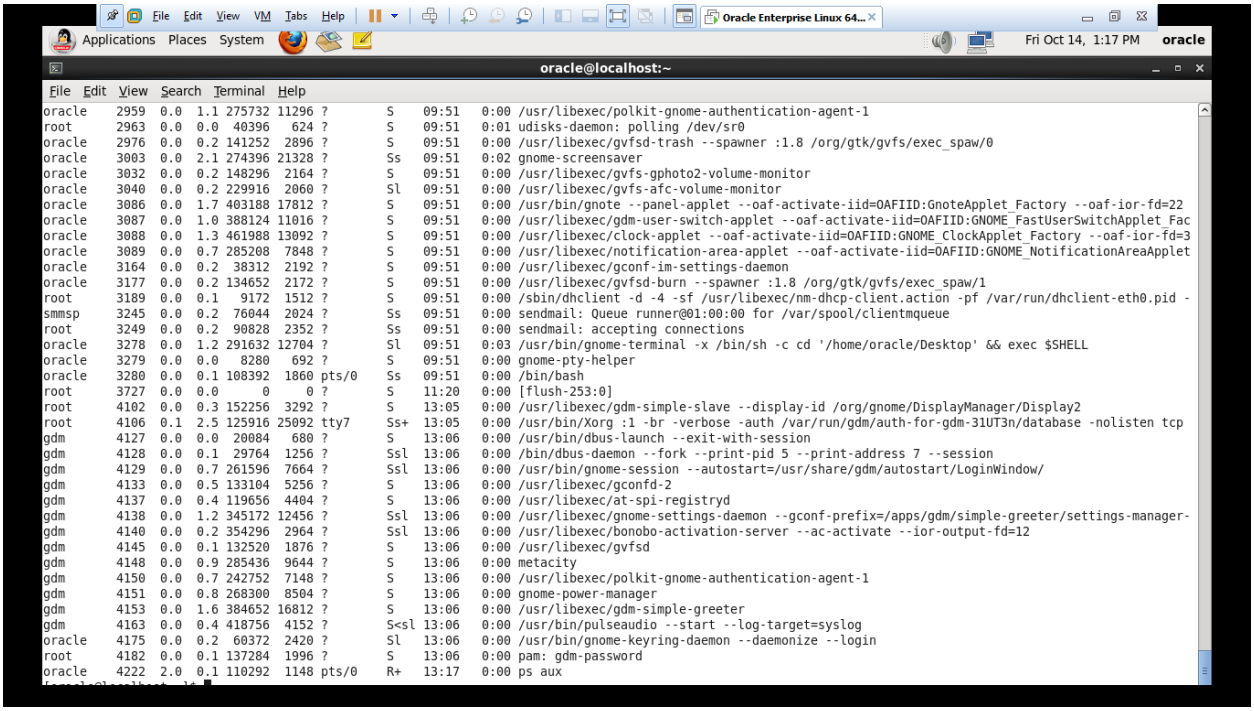
فرمان **ps**

نمایش **process** های در حال اجرا

با اجرای فرمان **ps** بدون هیچ گزینه ای لیستی از پردازشهای کاربر مورد نظر که لاگین کرده است نمایش داده میشود مانند شکل زیر:



حال اگر بخواهیم اطلاعات جامع تری از پردازشهای در سطح سیستم عامل مشاهده کنیم میتوانیم از دستور **ps** به همراه گزینه های مختلف استفاده کنیم مثلا **ps -aux** کلیه پردازشهای روی سیستم عامل را نمایش میدهد .



همانگونه که درشکل بالا مشاهده میکنید خروجی دستور **ps aux** بسیار طولانی میباشد ولی اگر از این لیست ما بخواهیم مثلا ببینیم آیا سرویس پست الکترونیکی ما مثل **postfix** در حال اجرا هست یا خیر میتوانیم با استفاده از عمل **Pipe** دو دستور **ps** و **grep** را بخوبی برای رسیدن به منظور خود استفاده نماییم . بدین منظور کافی است این فرمان را اجرا نماییم

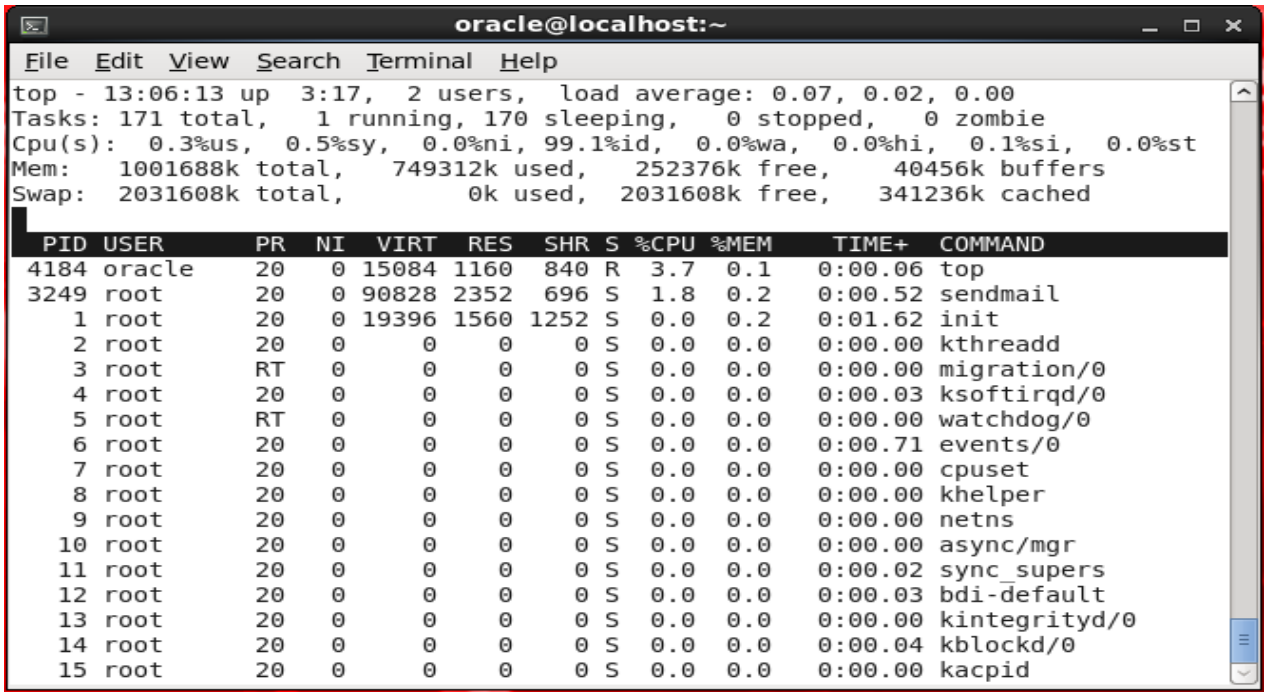
```
$ ps aux | grep postfix
```

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

فرمان top

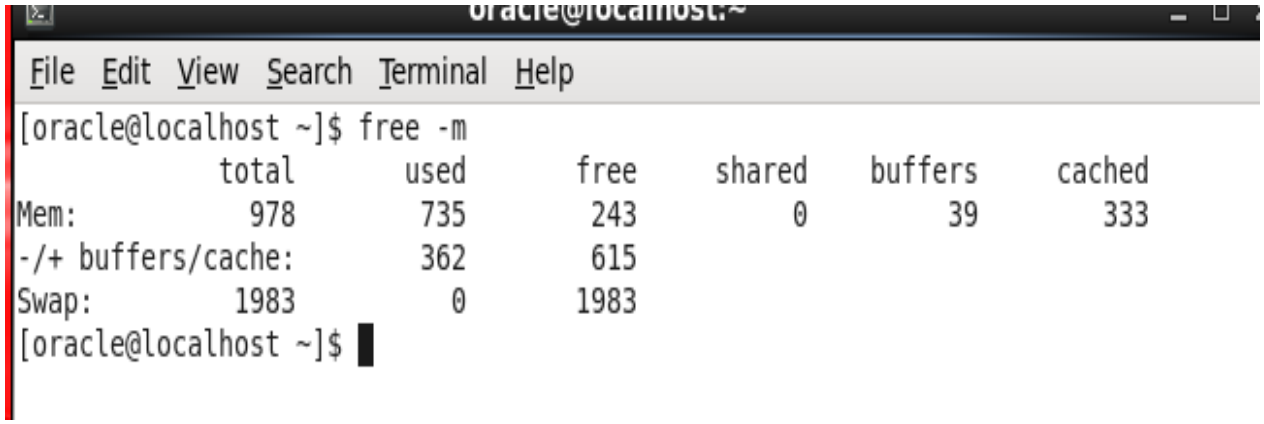
فرمانی که برای مانیتورینگ **process** ها بکار میرود اطلاعات جامعی از نام پردازشها و میزان فضای اشغالی و بر اساس اولیت میزان حافظه و یا میزان پردازنده ای که برای آن پردازش در سیستم مصرف میگردد این لیست نمایش داده میشود

دستور **top** خروجی ای شبیه به شکل زیر دارد :



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:



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

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

بطور کلی در سطح سیستم عامل سه نوع کاربر وجود دارد

1- کابران استاندارد **standard**
2- کاربر **root**
3- کاربر سیستمی **system**

1- کاربران استاندارد سیستم عامل لینوکس چه کسانی هستند؟

این کاربران شامل کاربرانی میشود که مدیر سیستم آنها بعد از روشن شدن سیستم ایجاد میکند که میتوانند شامل تمام افرادیکه قرار است با این سیستم عامل بنوعی کار کنند باشد . مثلا 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 میتوانند قدرت مدیر سیستم یا **root** را دارا و یا نداشته باشند.

2- کابر **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 .

3- کابر سیستمی

این کاربران برای اجرای یک برنامه خاص در سطح سیستم عامل تعریف میشوند . بعنوان مثال فرض کنید که قرار است در ماشین لینوکس شما یک سرویس وب اجرا گردد بنابرین سیستم عامل نیاز به کاربری برای اجرای این سرویس دارد که مثلا با نام **apache** این کاربر را جهت اجرای سرویس وب تخصیص میدهد . در واقع این کاربر نیاز به ورود به سیستم بصورتیکه بتواند از دستورات خط فرمان لینوکس استفاده کند ندارد بلکه فقط به این منظور ساخته شده است تا سرویس وب را اجرا نماید.

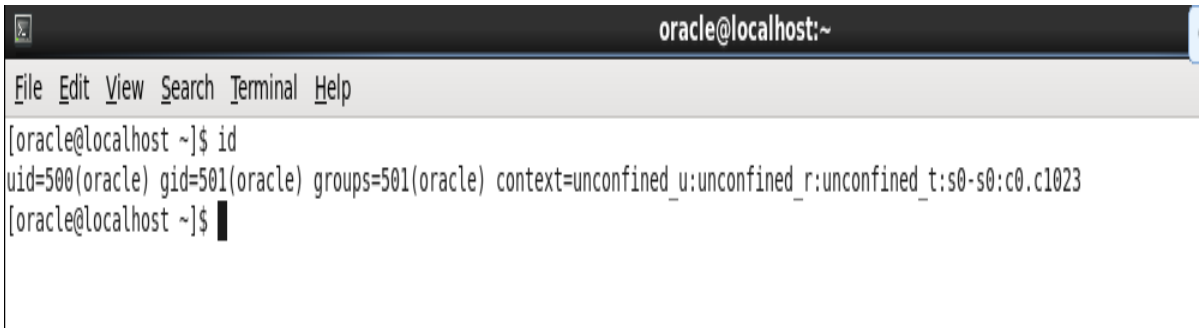
یا مثلا اگر قرار است پایگاه داده ای **mysql** را روی سیستم خود اجرا نمایید نیاز به کاربری با نام**mysql** دارید تا این سرویس را اجرا نماید .

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**

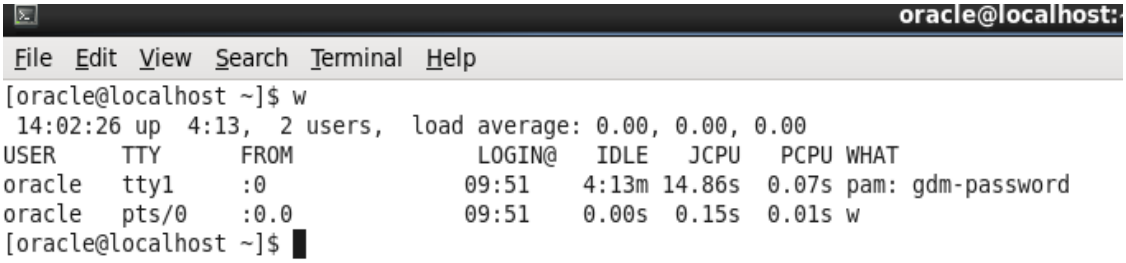
دستور **id** اطلاعاتی در مورد شخصی که به سیستم وارد یا به اصطلاح لاگین کرده است میدهد با اجرا این فرمان خروجی ای به شکل زیر نمایش داده میشود



مثلا 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 **500** و نام گروهی که این کاربر عضو آن است که گروه **oracle** با عدد **501** و اطلاعات دیگر در مورد این کاربر به ما میدهد

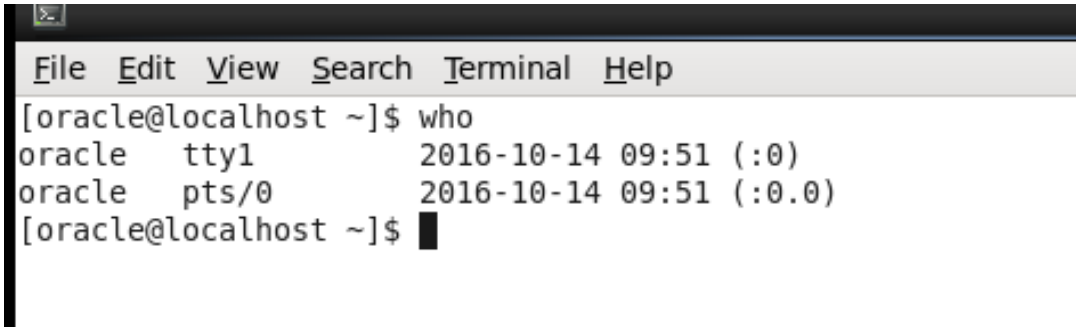
دستور **w**

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



دستور **who**

دستور **who** میگوید در حال حاضر چه کسانی وارد سیستم شده اند و از طریق چه ترمینالی به سیستم متصل هستند



دستور **sudo**

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

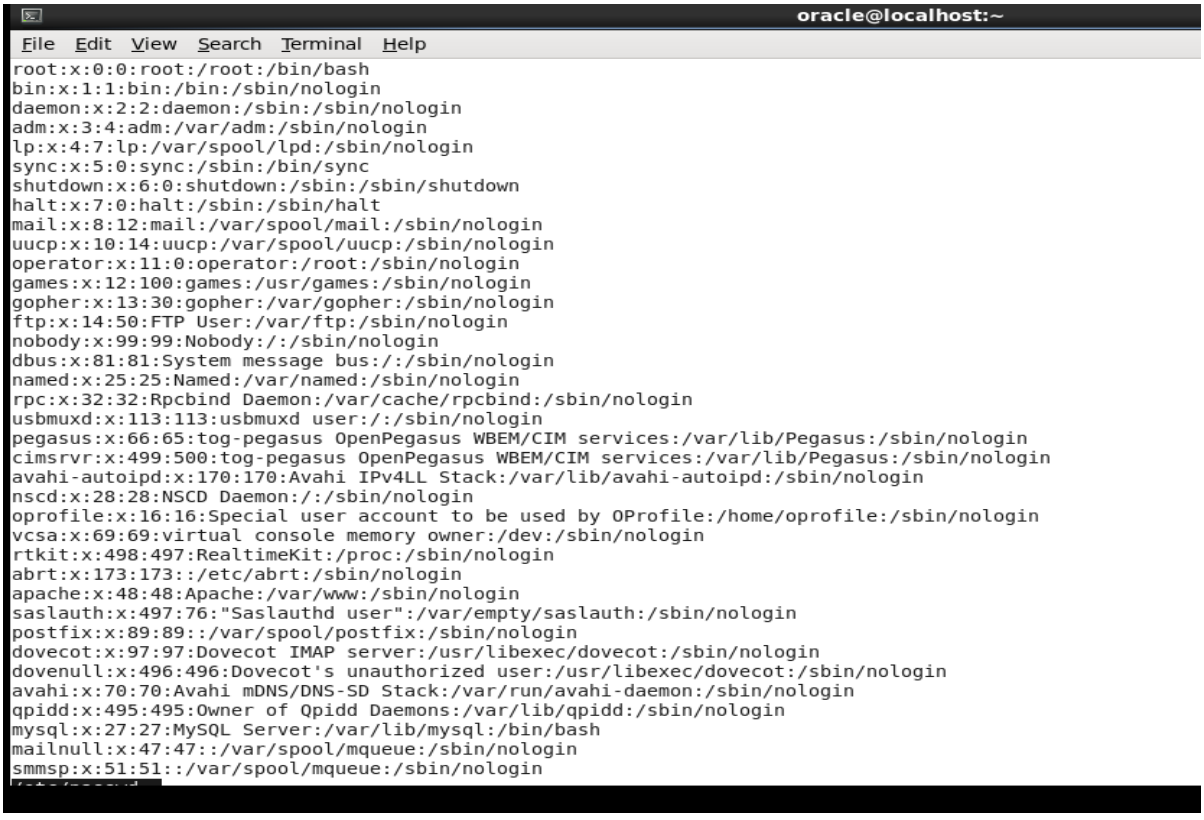
فایل های مرتبط با کاربران

1- فایل /etc/passwd

این فایل اطلاعاتی از کلیه کابران سیستم اعم از کابراین سیستمی یا استاندارد و یا **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 چکاری انجام دهد و مسیر خانه شخصی این کاربر یا **home directory** این کاربر می باشد .

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.

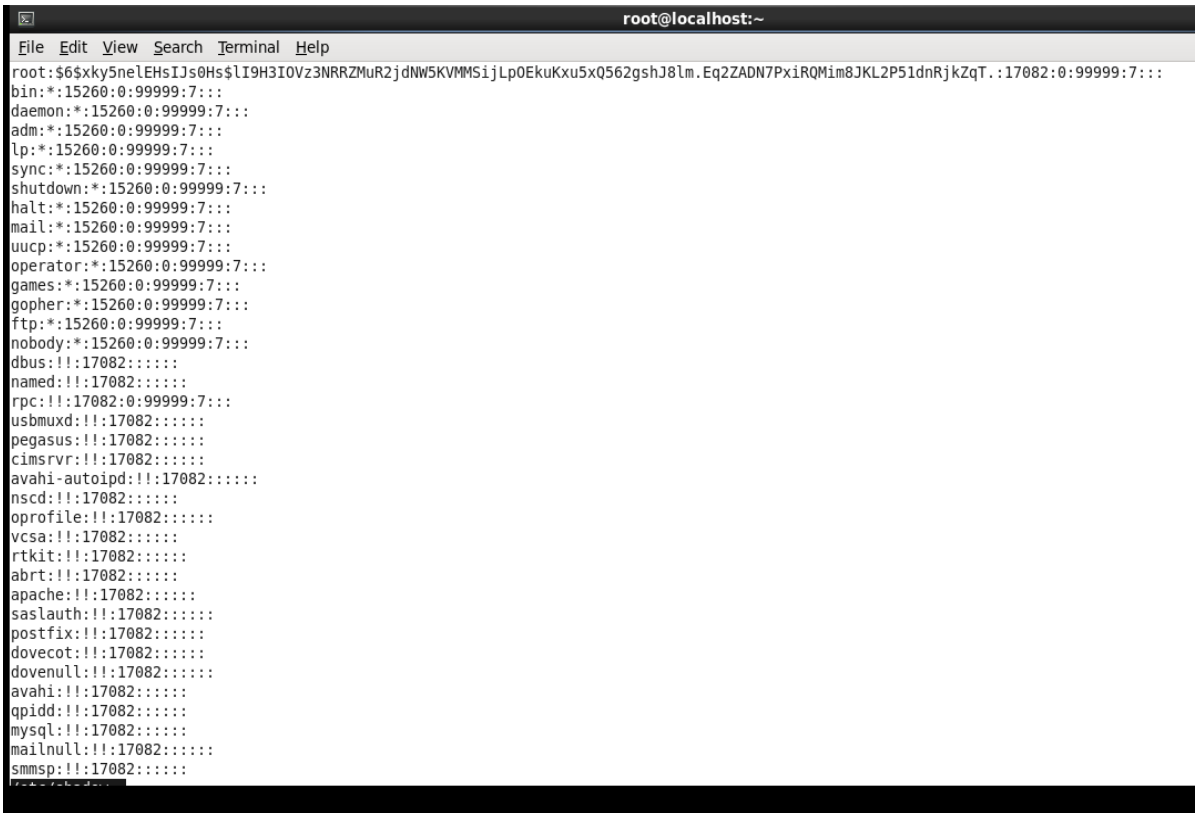
نمونه فایل /etc/passwd



2- فایل /etc/shadow

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

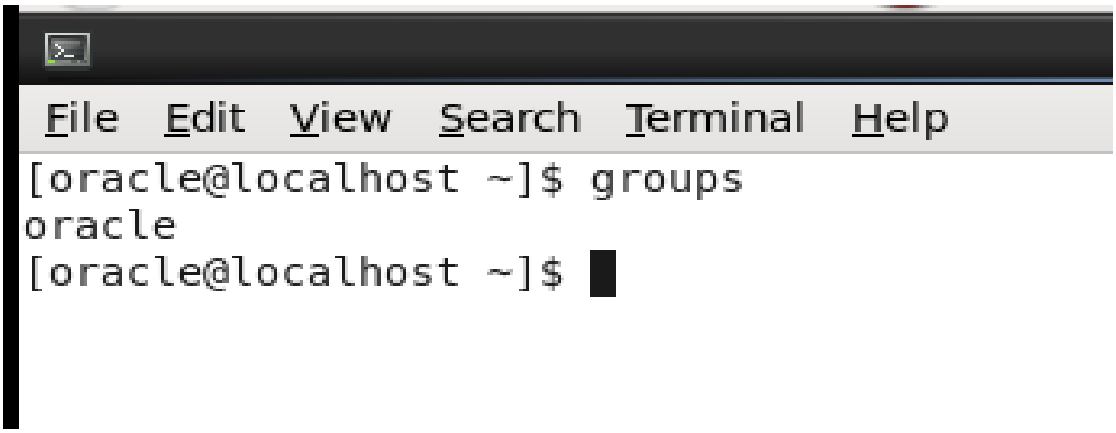
نمونه ای از فایل **/etc/shadow**



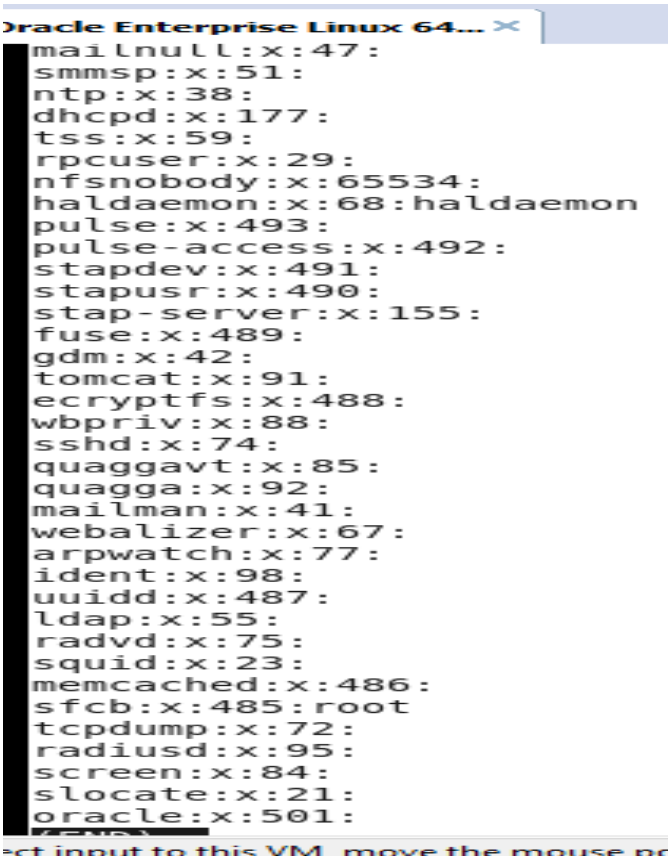
از جمله اطلاعات دیگری که در این فایل ذخیره شده است در مورد طول عمر کلمه عبور ، اینکه تا چه زمانی فرصت دارند کلمه عبور خود را تغییر دهند . اینکه چه کاربری قفل یا **Lock** شده است و اجازه ورود ندارد.

فایل **/etc/group**

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



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



نکته

برای استفاده از دستور **sudo** برای کاربری که **root** نیست باید حتما این کاربر جزء کاربرانی که حق استفاده از دستور **sudo** را دارند تعریف شده باشد وگرنه سیسم عامل اجازه استفاده از این قدرت به هر کاربری

نمیدهد . این عمل عموما در فایل **/etc/sudoers** انجام میشود که فایلی با پیکربندی مخصوص خود می باشد که جهت تعریف دسته ای از دستورات و کاربرانی می باشد 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 نمیخواهم سردرگم نمایم.

در دوره های **LPIC-1** در مورد این فایل و نحوه استفاده از آن شرح کامل داده میشود.

13- ایجاد کاربران و گروههای کاربری

دستوراتی که در این قسمت دنبال شرح خواهم 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 استفاد قرار میگیرد . برای ایجاد کاربر جدید حتما 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 می کنیم

#useradd oracle

مثلا در دستور بالا کاربری با نام **oracle** در سیستم ایجاد کرده ام . چنانچه این کاربر قبلا وجودداشته باشد که سیستم عامل این خطا را روی صفحه ظاهر میکند و اگروجود نداشته باش این کاربر جدید در سیستم با یک عدد منحصر بفرد ایجاد میشود .

در هر سیستم عاملی از مدل لینوکس عددی خاص برای شروع ایجاد کاربران استاندارد سیستم در نظر گرفته میشود مثلا در بعضی از نسخه های لینوکس عدد **500** به بعد برای ایجاد کاربران استاندارد در نظر گرفته میشود بنابراین چنانچه در هنگام ایجاد کاربر جدید عددی را به سیستم عامل معرفی نکنیم این سیستم عامل است که این عدد را بر اساس پیش فرضهای خود در نظر میگیرد و مثلا از از **500** 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 با فرمان **useradd** سیستم عامل یک گروه اختصاصی برای این کاربر با نام کاربری او در نظر میگیرد و یک عدد واحد را برای این گروه اختصاصی نیز به این کاربر تخصیص میدهد مثلا در دستور بالا با اجرای فرمان **useradd oracle** اگراین کاربر اولین کاربر استاندارد سیستم باشد **uid** این کاربر معادل **500** و **gid** این کاربر مثلا با همان **500** می باشد البته اگر عدد **500** قبلا برای هیچ گروه دیگری تخصیص نیافته باشد.

پس در ابتدای اجرای فرمان **useradd oracle** چه اتفاقاتی می افتد:

1- کاربری با نام **oracle** در سطح سیستم عامل ایجاد میشود
2- یک عدد واحد بنام **uid** بر اساس پیش فرض سسیستم عامل مثلا از عدد **500** به بالا به این کاربر تخصیص داده میشود
3- یک گروه مخصوص این کاربر با نام **oracle** در سطح سیستم بوجود می آید
4- یک **gid** یا عدد واحد بر اساس پیش فرضهای سیستم عامل برای این کاربر تخصیص داده میشود.
5- پوشه ای با نام این کاربر (در اینجا **oracle**) در مسیر **/home** ایجاد شده و مجموعه ای از حق دسترسی ها به این کاربر در این مسیر داده میشود تا بتواند فایلهای خود را در این مسیر ذخیره و مدیریت نماید .
6- یک **shell** یا مترجمی که بتواند دستورات کاربر را برای سیستم عامل ترجمه کند در اختیار کاربر قرار داده میشود . بطور پیش فرض این **shell** از نوع **bash** میباشد که نوعی مفسر یا ترجمه کننده دستورات برای سیسیتم عامل لینوکس است .

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

مثلا

```
#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 **5000** و **gid** او برابر **5000** , و مسیر خانه شخصی این کاربر در **/var/vmail** و مفسر یا **shell** این کاربر **nologin** تعریف شده است . زمانیکه برای کاربری **shell** را از نوع **nologin** یا **false** انتخاب مینماییم 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 بصورت دستی تعیین کرده ایم.

یا مثلا اگر میخواهیم کاربری با نام **oracle** ایجاد کنیم که قرار است در گروه **oinstall** عضو باشد ابتدا گروهی بنام **oinstall** ایجاد کرده و سپس کاربر **oracle** را بصورت زیر ایجاد میکنیم

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

2- دستور groupadd

این دستور برای ایجاد یک گروه جدید بکار میرود که همانند دستور **useradd** میتوانیم 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.

مثلا **groupadd oinstall** یا **groupadd -g 1001 oinstall**

3- استفاده از دستور Id

دستور **id** به ما میگوید که این کاربر متعلق به چه گروهی یا گروههایی میباشد و **uid, gid** کا ربر را نیز نمایش میدهد مثلا

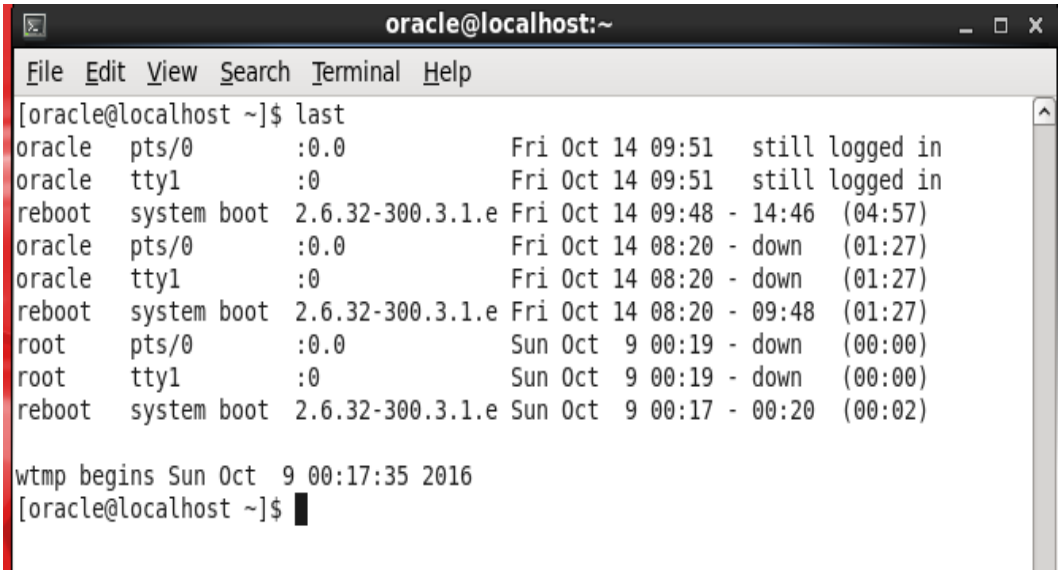
 # id oracle

به ما نشان میدهد که کاربر **oracle** در چه گروه یا گروههایی عضو است و مقدار **uid , gid** این کاربر در سیستم چه میباشد.

دستور last

این دستور میگوید که آخرین باردر چه ساعت وتاریخی چه کسانی به سیستم وارد شده اند

اجرای آن به صورت زیر میباشد :



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

کلیه این دستورات با حروف کوچک در سیستم عامل لینوکس استفاده میشود .

- دستور **userdel** برای حذف یک کاربر بکار میرود زمانیکه این دستور را با گزینه **"r-"** بکاربریم در سطح سیستم عامل حذف میکنیم چه اتفاقاتی می افتد؟
  1- نام کاربری او از فایل **/etc/passwd** حذف میگردد
  2- نام کاربر از گروهی که متعلق به آن است حذف میشود .
  3- مسیر **home directory** کاربر حذف میگردد بنابراین کلیه داده های ذخیره شده در این مسیر پاک خواهند شد
  4- کاربر قادر به ورود به سیستم نخواهد بود

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

بنابراین زمان حذف کاربر به موارد بالا بهتر است 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 بکنیم.

دستور **usermod** برای انجام پاره ای از تغییرات در نام یا **shell** یا **uid** یا **gid** یا **home directory** کاربر بکار میرود .

مثلا کاربری با نام **oracle** داریم و میخواهیم علاوه بر اینکه عضو گروه **oinstall** است عضو گروهی بنام **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 بکار 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 کابران را تغییر دهد.

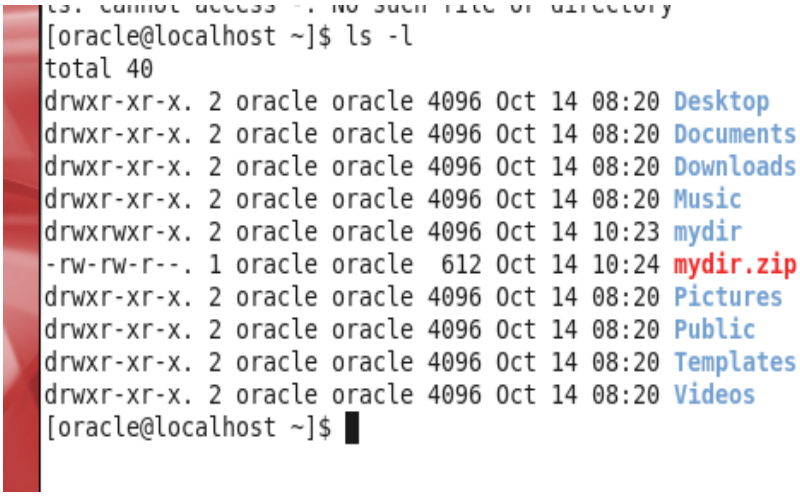
بعنوان مثال اگر من بعنوان کاربر **oracle** میخواهم رمز خود را تغییر دهم کافی است در سطح **shell** دستور

Passwd را وارد کنم در این صورت سیستم عامل در ابتدا کلمه رمزی که در حال حاضر در سیستم استفاده میکنم را می پرسد 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 سوال میکند و سپس تغییرات در فایل **/etc/shadow** اعمال میگردد.

14- نحوه اجازه دسترسی به کاربران

در این فصل قصد دارم مختصری در مورد اجازه های دسترسی فایلها و پوشه هایی که در سطح سیستم عامل میسازیم و همچنین مدیریت فایلها و دایرکتوری ها در لینوکس بپردازم.

در ابتدا فرض کنید با کاربر **oracle** وارد سیستم شده ایم و دستور **ls -l** که لیستی کامل از کلیه فایلها و پوشه هایمان را نمایش میدهد را نشان میدهیم :



همانگونه که در شکل بالا مشاهده میکنید لیست فایلها و پوشه های مربوط به کاربر **oracle** لیست شده است .
اطلاعاتی که در صفحه مشاهده میکنید به قرار زیر است :

ستون اول که بصورت **drwxr-xr-x** ویا **—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 میباشد و حرف **"d"** مخفف **directory** میباشد . همچنین این عبارت میگوید که این پوشه دارای مجوز **read** بطور اختصار **"r"** و **write** یا خواندن که بطور اختصار با **"w"** و همچنین مجوزاجرای یا **execute** که بطور اختصار با حرف **"x"** نمایش داده شده است برای خود کاربر **oracle** می باشد.

پس **rwx** در اولین ستون اجازه های 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 پوشه های داخل این مسیر بگیرد.

اما **r-x** ستون بعدی میگید که گروهی که این کاربر به آن متعلق است قادر است محتویات ای 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 آنجا که بجای حرف **"w"** حرف "-" در این قسمت گذاشته شده یعنی گروه این کاربر قادر به تغییر این فایلها در این پوشه نمی باشد .

اما **r-x** در قسمت آخر این عبارت میگوید افرادی بجز کاربر وگروهی که کاربر به آن متعلق است یعنی باقی کاربران سیستم چه اجازه ها 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 اینجای 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 پوشه ها چه کسی است و ستون چهارم میگوید گروهی که این کاربر به آن متعلق است چه گروهی است و ستون پنجم سایز فایل یا پوشه را ذکر میکند ستونهای **6** و **7** تاریخ و زمان ایجاد فایل یا پوشه را نمایش میدهند و ستون آخر نام فایل یا پوشه را نشان میدهد .

پس متوجه شدیم که با یک دستور ساده **"ls –l"** در یک مسیر خاص میتوانیم اطلاعات امنیتی اولیه مربوط به فایها و پوشه های موجود در آن مسیر را پیدا کنیم .

دراین قسمت میخواهم در مورد اجازه های دسترسی که در بالا صحبت کردیم بیشتر شرح دهم.

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

1- کاربر که میتواند اجازه خواندن ؛ نوشتن و یا اجرا را در سطح فایل یا پوشه داشته باشد . این اجازه ها را بصورت **read,write,execute** و با حروف اختصاری **"rwx"** نمایش میدهیم . اما از آنجا که سیستم عامل دارای زبان ماشین است و ماشین عدد میفهمد لذا این حروف به اعداد تبدیل میشوند و این اعداد غالبا در مبنای **2** می باشند .

اما چگونه؟

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

ستون دوم میشود 2^1=2

ستون سوم 2^2=4

این محاسبه برای اجازه های دسترسی در سطح گروه و دیگران نیز به همین صورت می باشد یعنی 1 و2 و4

بنابراین اگر فایل یا پوشه ای دارای همه مجوزهای خواندن ، نوشتن و اجرا باشد یعنی مجموع اعداد 4و2و1 را داراست که مجموع این ارقام یعنی 7 = 4+2+1 میباشد . بنابراین اگر فایل یا پوشه ای مجوز عددی 7 را دارا باشد یعنی هر کاری روی این فایل میشود انجام داد یا بعبارتی کلیه مجوزها را دارا میباشد .

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

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

همچنین فایل مجوزنوشتن 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 0 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 4 برای خواندن و 0 برای نوشتن و 1 برای اجرا می باشد که مجوع این ارقام یعنی 4+0+1=5 مقدار عددی مجوز این فایل در سطح کاربر را نمایش میدهد

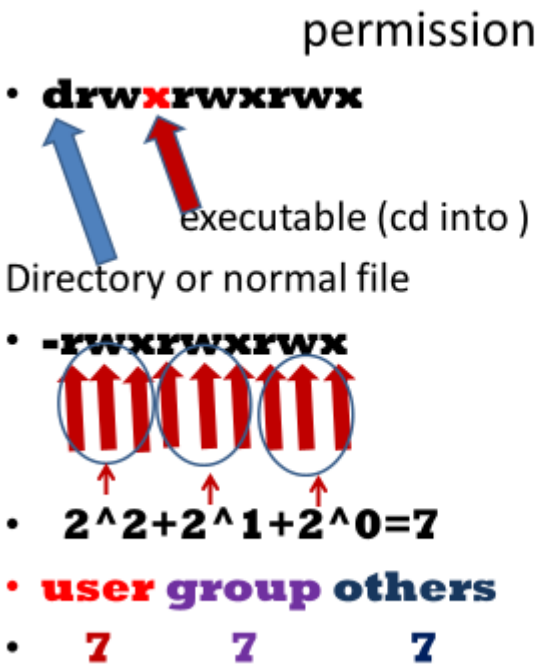
اگر این فایل در سطح گروه هم همین مجوزها را داشته باشد بنابراین درسطح گروه هم عدد 5 را دارا میباشد .

حالا اگر این فایل برای باقی کاربران فقط مجوز خواند داشته باشد مقدار آن چه میشود ؟

مجوز خواندن یا **read** یعنی **2^2=4** و چون مجوزهای دیگر را دارا نیست یعنی مقدار **0** 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 **7** برای کاربر و **5** برای گروه و **4** برای دیگران میباشد یعنی عدد **754**که بصورت متنی عبارت **"rwxr-xr—"** را برای فایل مشاهده خواهیم کرد.

شکل زیر گویای این شرح میباشد



اما شناخت این اعداد برای ما چه سودی دارد ؟

زمانیکه ما میخواهیم مجوزهای یک فایل را وارد کنیم دو راه در پیش رو داریم

1- بصورت مطلق این مجوز ها را به فایل بدهیم
2- بصورت نسبی این مجوزها را به فایل بدهیم

1- حالت مطلق

در حالت مطلق ما کاری به اینکه درحال حاضر مجوزهای فایل یا پوشه چه هستند نداریم و به یکباره مجوز مورد نظر را اعمال میکنیم یعنی اگر قبلا اجازه خواندن و نوشتن و اجرا را در سطح کاربر یا گروه یا دیگران داشته یا نداشته برای ما مهم نیست ما میخواهیم از الان به بعد مثلا مجوز فقط خواندن و نوشتن برای کاربر و فقط خواندن برای گروه و فقط اجرا برای دیگران داشته باشد .

2- حالا نسبی

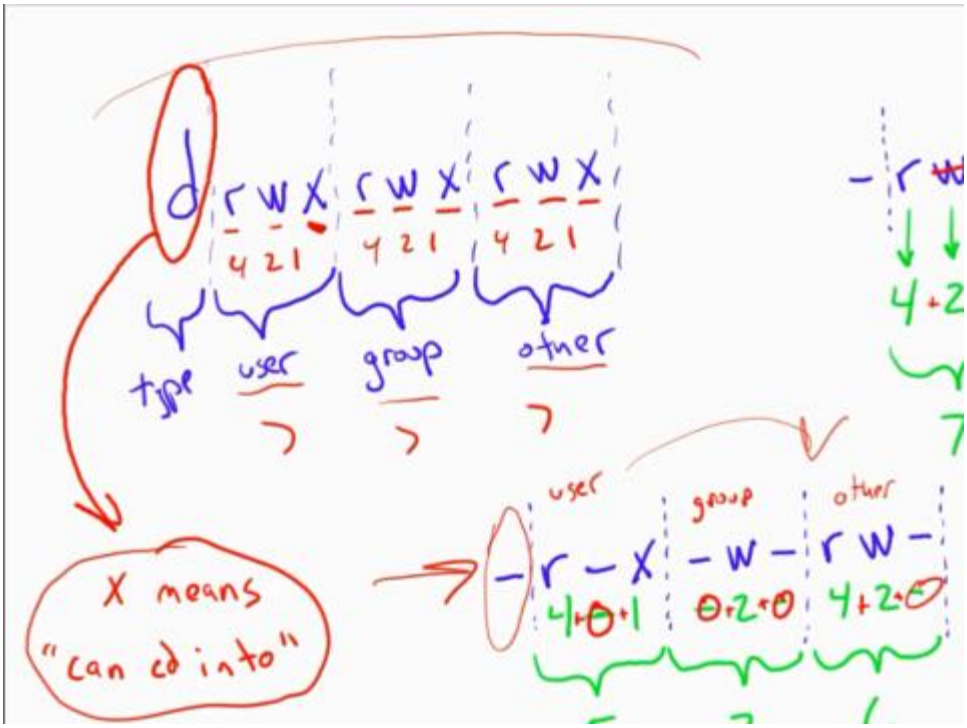
در این حالت ما مجوزهایی را به فایل یا پوشه نسبت به مجوزهایی فعلی آن میدهیم . یعنی میگوییم اگر فایل مجوز خواندن و اجرا دارد داشته باشد حالا مجوز نوشتن را به آن اضافه کن . یا به مجوز گروه و کاربران دیگر دست نزن و فقط مجوز کاربر را تغییر بده

برای تغییر مجوز روی فایل و پوشه ها از دستور **chmod** استفاده میکنیم

بدین صورت که میتوانیم از حالت مطلق بصورت **"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 ودیگران عطا 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 مثلا کاری نداریم که مجوزروی افراد غیر از گروه چه هست و فقط میگوییم مجوز نوشتن را از گروه بگیر یعنی بصورت نسبی تغییر اجازه دسترسی میدهیم

در شکل زیر مثال دیگری آور ده شده است :



مثلا وقتی میگویم **"chmod go-w"** یعنی مجوز نوشتن یا تغییر دادن فایل را از گروه و دیگران صلب کرده ایم اما به باقی مجوزها کاری نداریم .

یا وقتی میگوییم **"chmod ugo-w"** یعنی از کاربر وگروه و دیگران مجوز نوشتن را بگیر .

علامت "-" یعنی گرفتن یک اجازه

و علامت "+" یعنی دادن یک اجازه

مثلا 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 بکار میرود . البته چون اجازه های دسترسی و امنیت فایلها با این روش به خطر می افتد لذا تنها کاربر **root** و یا کاربرانی که قبلا با قدرت **root** در سیستم تعریف شده باشند قادر به اجرای این دستور میباشند .

با این دستور میتوانیم مالک یک فایل را تغییر دهیم مثلا **chown root /home/oracle** یعنی میگویم که مالک مسیر **/home/oracle** کاربر **root** باشد . اگر بخواهیم عمل تغییر مالک را بصورت تو در تو در داخل یک مسیر وتمام زیر مسیرهای آن و روی تمام فایلهای داخل مسیرهای زیرین آن انجام دهیم از گزینه **"-R"** استفاده میکنیم .

مثلا اگر بخواهیم در مسیر **/u01** مالک کل مسیر و تمام زیر مسیر و فایلهای آن بنام کاربر **oracle** شود از دستور

"chown -R oracle /u01" استفاده میکنیم

دستور **chgrp**

این دستور برای تغییر گروه یک مسیر وفایل های یک مسیر میباشد و مانند **chwon** باید اجازه **root** را دارا باشد

مثلا اگر میخواهیم گروه مسیر **/u01** را به **oinstall** تغییر دهیم با اجرای دستور **"chgrp -R oinstall /u01"** این عمل را انجام میدهیم.

نکته – در دستور **chown** میتوانیم هم زمان 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 زیر پوشه ها و فایلهای آن انجام دهیم از گزینه **"-R"** بصورت **"chown -R oracle:oinstall /u01"** استفاده میکنیم . علامت **":"** برای جدا سازی مالک از گروه میباشد

ارائه شده توسط وبسایت ایرانیان دانلود

مرجع دانلود کتاب الکترونیکی

www.ir-dl.com