

گروه نرم افزارى سخت ۹۸

Soft98.iR

راهنمای کامل خط فرمان لینوکس

کتاب : راهنمای کامل خط فرمان لینوکس
نویسنده : سایت رسمی Techno-tux

Written By:
Techno\tux

Security Magazine ©
www.simorgh-ev.com

Ramin.samad@yahoo.com

ظاهر پوسته فرمان

در صورتی که لینوکس شما فاقد محیط گرافیکی است و یا اکنون محیط گرافیکی آن در حال اجرا نیست، شما باید دستورات خود را از طریق پوسته فرمان به سیستم عامل ارسال کنید. نخستین چیزی که در پوسته فرمان مشاهده میکنید، اعلان فرمان است که بصورت علامت \$ میباشد. اعلان فرمان برای کاربر ریشه بصورت # است. در بیشتر سیستمهای لینوکس قبل از اعلان فرمان نام کاربری شما و نام کامپیوترتان قرار میگیرد که بصورت زیر نشان داده میشود:

```
[alan@memphis home]$
```

امکان نمایش کاراکترهای مورد نیازتان بجای کاراکترهای فوق وجود دارد. چگونگی این کار بعدا شرح داده خواهد شد. محیط پوسته فرمان امکانات زیادی دارد.

تایپ دستورات در محیط پوسته فرمان بسیار آسان میباشد. برای اینکه با محیط پوسته فرمان آشنا شوید، سعی کنید با دستوراتی که در زیر بررسی میشوند، تمرین کنید.

نکته: در صورتی که هنگام راه اندازی سیستم، بجای پوسته فرمان محیط گرافیکی لینوکس اجرا میشود، برای تایپ فرامین پوسته باید از Terminal یا Konsole استفاده کنید. میتوانید در منوی run، فرمان xterm را نیز تایپ کنید.

در مثالهای زیر علامتهای \$ و # نشان دهنده اعلان فرمان میباشدند. پس تایپ هر فرمان باید کلید Ebter را فشار دهید و خروجی آن فرمان در خطوط پس از آن نمایش داده خواهد شد.

بررسی نشست ورود به سیستم

هنگامی که وارد سیستم لینوکس میشوید، برای سیستم دارای یک هویت خاص هستید. این هویت شامل نام کاربری شما، نام گروه شما، شماره کاربری شما و شماره گروه شماست. همچنین لینوکس اطلاعات زمان ورود به سیستم، مدت حضور، مدت بیکاری و محل ورود شما به سیستم را نگهداری میکند. (حواستان را جمع کنید!)

برای بدست آوردن اطلاعات در مورد هویت کاربری خودتان در جلوی اعلان فرمان دستور زیر را تایپ کنید. خروجی آن در زیر آن نشان داده شده است:

```
$ id
```

```
uid=500(Alan) gid=500(Alan) groups=500(Alan)
```

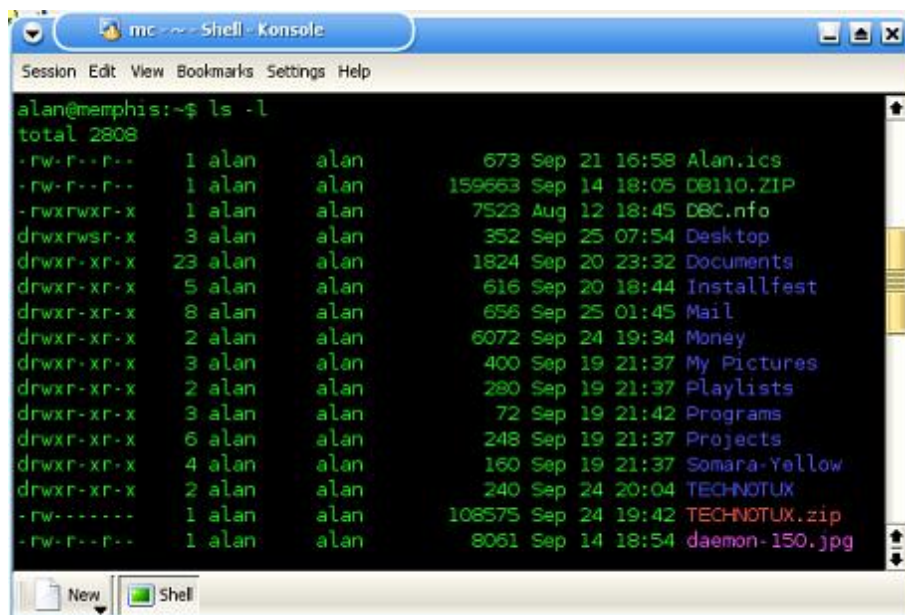
خروجی فرمان نشان میدهد که نام کاربر Alan بوده که عضو گروه Alan است و شماره های کاربری و گروه آن ۵۰۰ میباشد.

با استفاده از فرمان who میتوانید اطلاعاتی در مورد نشست جاری بدست آورید. در زیر این فرمان به همراه خروجی آن نشان داده شده است:

```
$ who
```

```
Alan :0 Apr 23 08:46
```

همچنان که می بینید، در خروجی نام کاربر جاری، زمان و تاریخ ورود به سیستم نمایش داده شده است.



تصویر ۱ برنامه Konsole محیط KDE

بررسی دایرکتوری ها و مجوزهای فایلها

در لینوکس مسیر جاری به مسیری گفته میشود که کاربر در آن لحظه در آن قرار دارد. هنگامی که وارد سیستم میشوید، لینوکس شما را در دایرکتوری خانگی تان قرار میدهد. هنگامی که دستور باز کردن یا ذخیره کردن فایلی را صادر میکنید، لینوکس مسیر جاری را بعنوان محل آن فایل فرض کرده و از آنجا آنرا باز کرده و یا ذخیره میکند. ساختار سیستم فایل لینوکس بعدا شرح داده خواهد شد و لازم نیست نگران آن باشید. برای نمایش دایرکتوری جاری فرمان زیر را جلوی خط فرمان تایپ کنید. خروجی آن در زیر آن نمایش داده شده است:

```
$ pwd
```

```
/usr/bin
```

در مثال بالا مسیر جاری usr/bin است. برای یافتن مسیر دایرکتوری خانگی خود، فرمان زیر را تایپ کنید:

```
$ echo $HOME
```

```
/home/Alan
```

همچنان که در خروجی ملاحظه میکنید، مسیر دایرکتوری خانگی شما نمایش داده شده است. برای اینکه به دایرکتوری خانگی خود باز گردید، کافی است به سادگی فرمان زیر را تایپ کنید:

```
$ cd
```

این فرمان، شما را به دایرکتوری خانگی تان باز می گرداند. خوب بد نیست ببینیم که چه چیزهایی در دایرکتوری خانگی وجود دارد. برای نمایش محتویات یک دایرکتوری، باید از فرمان ls استفاده نمایید. در صورتی که در دایرکتوری خانگی خود

قرار ندارید میتوانید مسیر کامل آنرا تایپ کنید. در صورتی که فرمان ls را بدون هرگونه دایرکتوری تایپ کنید، محتویات مسیر جاری نمایش داده خواهد شد. گزینه a تمام فایلها و مخفی را نمایش میدهد و گزینه l برای نمایش جزئیات کامل فایلها بکار میرود. هنگام تایپ یک فرمان میتوانید گزینه های متعدد آنرا کنار هم تایپ کنید. در زیر این دستور به همراه یک خروجی مثال نشان داده شده است:

```
$ ls -la /home/Alan
```

```
total 46740
```

```
drwx----- 47 Alan Alan 4096 Apr 23 11:09 .
```

```
drwxr-xr-x 8 root root 4096 Mar 12 17:51 ..
```

```
-rw----- 1 Alan Alan 616581 Apr 18 23:29 779-red_hat_linux_9.tar.gz
```

```
drwxr-xr-x 2 Alan Alan 4096 Mar 20 11:15 .acrobat
```

```
drwx----- 2 Alan Alan 4096 Mar 20 11:15 .adobe
```

```
drwx----- 2 Alan Alan 4096 Mar 12 17:04 .adonthell
```

```
drwxr-xr-x 2 Alan Alan 4096 Feb 14 13:19 .anjuta
```

```
-rw----- 1 Alan Alan 18325 Apr 23 00:36 .bash_history
```

```
-rw-r--r-- 1 Alan Alan 24 Aug 24 2002 .bash_logout
```

```
-rw-r--r-- 1 Alan Alan 191 Aug 24 2002 .bash_profile
```

هنگامی که از سوئیچ l برای نمایش جزئیات بیشتر استفاده میکنید، چیزی بیش از سایز فایلها و دایرکتوری ها نمایش داده میشود. دایرکتوری جاری (.) و دایرکتوری والد (..) در بالای لیست قرار می گیرند. یعنی در حقیقت نقطه نشان دهنده دایرکتوری /home/Alan و دونقطه نشاندهنده دایرکتوری /home است. بخش ابتدایی لیست نشاندهنده مجوزهای هر فایل است. سایر اطلاعات نمایش داده شده عبارتند از اندازه فایل به بایت و تاریخ و ساعتی که فایل برای آخرین بار تغییر کرده است.

بررسی فعالیت سیستم

لینوکس علاوه بر چندکاربره بودن، سیستم عاملی است چند وظیفه (multitasking). چند وظیفه بودن به این معنی است که برنامه های زیادی میتوانند در یک زمان اجرا شوند. هر برنامه در حال اجرا یک پروسه نامیده میشود. لینوکس فرامینی برای نمایش پروسه های در حال اجرا، نمایش استفاده از منابع سیستمی و متوقف کردن پروسه های در مواقع لزوم دارد.

مرسوم ترین ابزار برای بررسی پروسه های در حال اجرا، دستور ps است. با این دستور، میتوانید بررسی کنید که چه برنامه هایی در حال اجرا هستند ، از چه منابعی استفاده میکنند و چه کسی در حال اجرای آنهاست. در زیر یک خروجی مثال از این فرمان نشان داده شده است :

```
$ ps au
```

```
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
```

```
Alan 1152 0.0 0.5 4476 1348 pts/0 S 17:39 0:00 bash
```

```
Alan 1831 0.0 0.2 2580 664 pts/0 R 18:14 0:00 ps au
```

در مثال بالا، گزینه a، برای نمایش تمام پروسه هایی که به ترمینال فعلی شما مربوط است و گزینه u برای نمایش نام کاربری و زمانی که برنامه آغاز به کار کرده است، بکار می‌رود. مفهوم ترمینال به زمانهای قدیم باز میگردد. در آن زمان کاربران روی ترمینالهای مبتنی بر متن کار میکردند و هر ترمینال نشان دهنده یک نفر بود. اکنون شما میتوانید روی صفحه مانیتور خود تعداد زیادی ترمینال داشته باشید. این کار بوسیله باز کردن تعدادی پنجره ترمینال امکان پذیر است.

در مثال بالا، چیز خاصی اتفاق نیفتاده است. خروجی نشان میدهد که کاربری به نام Alan، از برنامه های bash و ps در حال استفاده است. ستون TTY یا ترمینال، نشان دهنده ترمینال است که کاربر با آن به سیستم وارد شده است و ستون STAT نشاندهنده وضعیت پروسه است. R نشاندهنده پروسه در حال اجرا و S نشاندهنده پروسه در حال خواب میباشد.

ستون USER نام کاربری که پروسه را شروع کرده نمایش میدهد. هر پروسه توسط یک عدد پکتا به نام شماره پروسه (Process ID) مشخص میشود. از این شماره هنگام از میان بردن یا اصطلاحاً kill کردن پروسه استفاده میشود. ستونهای CPU% و MEM% نشاندهنده مقدار پردازنده و حافظه ای هستند که پروسه ها استفاده کرده اند. ستون VSZ یا Virtual Set Size نشاندهنده سایز پروسه image به کیلوبایت و RSS یا Resident Set Size نشاندهنده سایز پروسه در حافظه است. ستون START نشاندهنده زمان آغاز پروسه و ستون TIME نشاندهنده زمان سیستم استفاده شده برای پروسه است.

بسیاری از پروسه هایی که در کامپیوتر در حال اجرا هستند، به یک ترمینال خاص مربوط نیستند. یک سیستم عادی مبتنی بر لینوکس، دارای پروسه هایی فراوانی است که در پس زمینه اجرا میشوند. پروسه های پس زمینه پروسه هایی هستند که اعمالی مانند ثبت فعالیتهای سیستم یا گوش کردن به پورتهای برای اطلاعات واصله از شبکه را انجام میدهند. این پروسه ها هنگام بوت شدن سیستم آغاز به کار کرده و هنگام خاموش کردن سیستم، به کار خود پایان میدهند. برای نمایش تمام پروسه های در حال اجرا بر روی کامپیوترتان باید از فرمان زیر استفاده کنید:

```
$ ps aux | less
```

قسمت less | به این دلیل به فرمان اضافه شده است که در صورتی که تعداد پروسه ها از یک صفحه بیشتر شد، امکان نمایش صفحه به صفحه آن وجود داشته باشد. به این فرایند لوله بندی (pipe) فرمان گویند که به معنی هدایت خروجی یک فرمان برای ورودی فرمان دیگر است.

خروج از پوسته فرمان

هنگامی که کارهای خود را انجام دادید و مایل بودید از پوسته فرمان خارج شوید، کافی است که کلیدهای Ctrl+D را فشار دهید. در صورتی که در حالت متنی لینوکس را بوت کرده اید، کافی است فرمان logout یا exit را تایپ کنید.

خوب، تا اینجا با چند فرمان که به شما کمک میکند از سیستمتان اطلاعات لازم را به دست آورید، آشنا شدید. صدها فرمان دیگر نیز وجود دارند که میتوانید آنها را آزمایش کنید. این فرامین در مسیرهای usr/bin و bin قرار دارند. همچنین فرامین مدیریت سیستم در مسیرهای usr/sbin و sbin قرار دارند. بیشتر این فرامین در ادامه این فصل توضیح داده خواهند شد.

درک دقیق تر پوسته فرمان لینوکس

قبل از اینکه آیکونها و پنجره ها روی صفحه کامپیوترها پدیدار شوند، کاربران برای کار کردن با کامپیوترها باید فرمانهایی را تایپ میکردند. در سیستمهای مبتنی بر یونیکس که لینوکس هم یکی از آنهاست، برنامه ای که برای تفسیر و مدیریت فرمانها ایجاد شده است، پوسته فرمان (Command Shell) نام دارد.

پوسته فرمان راهی برای اجراکردن برنامه ها، کارکردن با فایلها، کامپایل کردن برنامه ها و مدیریت کامپیوتر ایجاد میکند.

با اینکه کارکردن با ابزارهای گرافیکی آسان تر از کار کردن با پوسته فرمان است، ولی بیشتر کاربران حرفه ای لینوکس ترجیح میدهند تا بجای ابزارهای گرافیکی از پوسته فرمان استفاده کنند. زیرا برای انجام بسیاری از کارها مانند پیکربندی های سیستم، پوسته فرمان بسیار قدرتمند تر از ابزارهای گرافیکی است. حتی برخی کاربران قدیمی یونیکس و لینوکس به ندرت از محیطهای گرافیکی برای انجام کارهایشان استفاده میکنند.

پوسته فرمانی که در این راهنما توضیح داده خواهد شد، bash نام دارد. نام آن برگرفته از Bourne Again Shell است. پوسته bash از نخستین پوسته سیستمهای یونیکس که sh یا Bourne Shell نام داشت، ایجاد شده است و یکی از پر کاربرد ترین پوسته های فرمان به شمار میرود. البته پوسته های دیگری نیز وجود دارند که از آنها استفاده میشود که میتواند از آنها csh یا C Shell که در سیستمهای یونیکس BSD استفاده میشود و ksh یا Korn Shell که بیشتر در Unix System V استفاده میشود، نام برد. لینوکس همچنین دارای پوسته های ash و tcsh نیز میباشد.

هنگامی که استفاده از یک پوسته فرمان را در لینوکس فرا بگیرید، به آسانی میتوانید پوسته های دیگر را نیز یاد بگیرید. در صورتی که هرگونه مشکل یا سوالی داشتید، میتوانید به صفحه manual آن پوسته مراجعه کنید.

نکته : برای نمایش صفحه manual هر فرمان کافی است در خط فرمان لینوکس دستور زیر را تایپ کنید:

```
$ man <command>
```

در لینوکس ، پوسته bash کاملا سازگار با پوسته فرمان sh میباشد.

استفاده از پوسته فرمان در لینوکس

هنگامی که یک فرمان را در پوسته فرمان تایپ میکنید، میتوانید به آن کاراکترهای دیگری اضافه کنید تا چگونگی کارکرد دستور مورد نظر را تغییر دهید. علاوه بر خود دستور، موارد دیگری که میتوانید در خط فرمان تایپ کنید عبارتند از :

-گزینه ها (Options) : اکثر فرامین دارای یک یا چند گزینه هستند که با اضافه کردن و بکار بردن این گزینه ها میتوانید نحوه رفتار فرمان را تغییر دهید. برای مثال همانطور که قبلا هم دیدید، در فرمان ls -la گزینه l برای نمایش لیست مشروح فایلها و دایرکتوری ها و گزینه a برای نمایش فایل های مخفی که با نقطه شروع میشوند، بکار رفت. ضمنا گزینه هایی که مخفف یک کلمه هستند با یک - شروع میشوند در صورتی که گزینه هایی که یک کلمه کامل هستند با -- شروع میشوند . برای مثال ls --help .

- آرگومان ها (Arguments) : بسیاری از فرامین، علاوه بر گزینه ها ، آرگومانهایی را نیز قبول میکنند. یک آرگومان یک بخش شامل نوعی اطلاعات مانند مسیر یا نام فایل میباشد. برای مثال در فرمان ls -la /home بخش home آرگومان فرمان ls به شمار میرود.

- متغیر های محیطی (Variables Environment) : خود پوسته اطلاعاتی را در بر دارد که برای کاربر مفید است. به این اطلاعات متغیرهای محیطی می گویند. برای مثال متغیر SHELL نمایانگر نوع پوسته مورد استفاده ، SP1 نشاندهنده اعلان فرمان و MAIL نشاندهنده محل صندوق پستی شما است :

```
$ echo $SHELL
```

```
/bin/bash
```

```
$ echo $MAIL
```

```
/var/spool/mail/Alan
```

توجه داشته باشید که برای فراخوانی متغیر ها به ابتدای آنها علامت \$ اضافه میشود.

نکته : برای نمایش تمام متغیرهای محیطی میتوانید از دستور declare استفاده کنید. برای نمایش یک متغیر خاص میتوانید همانند بالا از دستور echo استفاده کنید.

- کاراکترهای ویژه (Metacharacters) : کاراکترهایی وجود دارند که دارای معنای خاصی برای پوسته فرمان هستند. این کاراکترها میتوانند برای هدایت خروجی یک فرمان به یک فایل ، لوله بندی خروجی یک فرمان و یا اجرای فرمان در پس زمینه استفاده شوند. کاراکترهای ویژه در این فصل توضیح داده خواهند شد.

برای صرفه جویی در مقدار تایپ و آسانتر شدن کار ، پوسته فرمان دارای ویژگیهایی است که دستورات قبلی تایپ شده را نگهداری میکند. همچنین شما میتوانید برای آسانتر شدن، نامهای مستعار برای دستورات ایجاد کنید. پوسته فرمان

دستوراتی که قبلاً وارد کرده اید ذخیره میکند و میتوانید بجای تایپ مجدد دستورات ، دستورات قبلی را فراخوانی نمایید. این موضوع نیز جلوتر بررسی خواهد شد.

در صورتی که پوسته فرمان را تغییر داده نباشید، پوسته bash پوسته ای است که همراه با لینوکس استفاده میکنید. پوسته bash از نظر امکانات و قابلیتها قویتر از انواع دیگر پوسته های فرمان است. در این فصل بیشتر قابلیت های پوسته فرمان bash بررسی خواهند شد. ولی در صورتی که نیاز به اطلاعات بیشتری داشتید، میتوانید از دستور man bash برای نمایش راهنمای پوسته bash استفاده کنید.

یافتن فرمانهای لینوکس

در صورتی که بدانید که یک دستور در کجای سیستم فایل لینوکس قرار دارد، میتوانید آنرا با تایپ مسیر کامل اجرا نمایید. برای مثال برای اجرای دستور date :

```
$ /bin/date
```

البته در صورتی که دستوری در مسیرهای سخت و طولانی قرار داشته باشد ، این کار دشوار خواهد بود. بهترین راه حل این مشکل، نگهداری فرامین در یک دایرکتوری خاص است. سپس میتوانید این دایرکتوری را به مسیر جستجوی پوسته فرمان خود اضافه کنید تا هنگام تایپ یک فرمان، خود پوسته بطور خودکار دایرکتوری فوق را برای وجود فرمان کاوش کند

```
$ echo $PATH
```

```
/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:/home/Alan/bin
```

خروجی فرمان فوق مسیرهای تعریف شده برای پوسته فرمان را برای یک کاربر خاص نشان میدهد. همانطور که می بینید دایرکتوری ها توسط یک کلون از هم جدا شده اند. بیشتر دستوراتی که همراه با لینوکس ارائه میشوند، در دایرکتوری های bin ، usr/bin ، یا usr/local/bin قرار دارند. دستورات گرافیکی که با محیط های گرافیکی استفاده میشوند در مسیرهای usr/X11R6/bin و usr/bin/X11 قرار دارند. آخرین دایرکتوری نشان داده شده در خروجی فرمان، در دایرکتوری خانگی کاربر قرار دارد.

نکته : در صورتی که مایلید دستوراتی که خود ایجاد میکنید مستقیماً در خط فرمان اجرا شوند، میتوانید یک دایرکتوری به نام bin در دایرکتوری خانگی خود ایجاد کنید و این دستورات را در آنجا ذخیره کنید. لینوکس این دایرکتوری را بطور خودکار به مسیرهای تعریف شده اضافه میکند.



در صورتی که شما کاربر ریشه هستید، دستورات مربوط به مدیریت سیستم در دایرکتوری های sbin و usr/sbin قرار دارند.

ترتیب دایرکتوری های موجود در مسیرهای تعریف شده نیز مهم است. این دایرکتوری ها از چپ به راست بررسی میشوند. بنابراین اگر دستوری به نام foo هم در دایرکتوری usr/bin و هم در دایرکتوری bin قرار داشته باشد، اولی اجرا خواهد شد. برای اجرای دستور دوم foo باید مسیر کامل آنرا تایپ کنید و یا مسیرهای تعریف شده را تغییر دهید. چگونگی این کار جلوتر توضیح داده خواهد شد.

تمام فرامینی که تایپ میکنید، در دایرکتوری های مسیرهای تعریف شده شما قرار ندارند. برخی فرامین بصورت درونی در پوسته فرمان گنجانده شده اند. در صورتی که برای یک فرمان خاص یک نام مستعار همراه با گزینه ها و آرگومانهای خاص ایجاد کنید، ابتدا آن اجرا میشود. همچنین راههایی برای ایجاد توابعی که شامل چند فرمان هستند نیز وجود دارد. ترتیب بررسی محلهای مختلفی که پوسته فرمان برای پیدا کردن یک دستور انجام میدهد به شرح زیر است :

- **نامهای مستعار :** نامهایی که با دستور alias ایجاد شده اند و نشانگر یک دستور به همراه گزینه ها و آرگومانهای احتمالی میباشند.

- **کلمات رزرو شده پوسته فرمان :** کلماتی هستند که برای استفاده های مخصوص رزرو شده اند. بیشتر این کلمات دستوراتی هستند که معمولاً در زبانهای برنامه نویسی استفاده میشوند مانند do ، while ، case و غیره.

- **توابع :** دسته ای از دستورات که همراه هم در پوسته فرمان اجرا میشوند.

- **دستورات درونی** : دستوراتی که درون خود پوسته فرمان گنجانده شده اند.

- **دستورات سیستم فایل** : دستورات معمولی که بصورت فایل‌هایی در سیستم فایل لینوکس قرار دارند. مسیرهای این دستورات در متغیر محیطی PATH گنجانده شده است.

نکته : برای نمایش لیستی از فرامین درونی bash و گزینه های آن میتوانید از دستور help استفاده کنید. برای نمایش اطلاعات بیشتر در مورد دستور مورد نظر از دستور info علاوه نام دستور مورد نظر استفاده کنید.

برای اینکه بفهمید که یک دستور در کجا قرار دارد، میتوانید از دستور type برای این منظور استفاده کنید. برای مثال :

```
$type bash
```

```
bash is /bin/bash
```

از دستور بالا برای یافتن محل فرامیت دیگری مانند which، case و ... استفاده کنید. در صورتی که دستوری در چندین دایرکتوری قرار دارد، میتوانید با اضافه کردن گزینه a به دستور type، تمام محل‌های وجود آنرا چاپ کنید.

نکته : گاهی اوقات هنگام اجرای یک فرمان با خطاهایی مانند "این فرمان پیدا نشد" و یا "شما مجوز استفاده از این فرمان را ندارید" مواجه میشوید. برای مورد اول بررسی کنید که دستور را صحیح تایپ کرده اید و مسیر آن در مسیر PATH شما قرار داشته باشد. ممکن است فرمان مورد نظر اجرایی نباشد. در بخش کارکردن با فایل‌ها، چگونگی اجرایی کردن یک فایل تشریح خواهد شد.

اجرای مجدد یک فرمان

تصور کنید یک فرمان بسیار طولانی را تایپ کرده اید و پس از اجرای آن متوجه میشوید که مرتکب اشتباه شده اید. مطمئناً چیزی درآوردن تر از این وجود ندارد! پوسته فرمان دارای قابلیتهایی است که میتوانید بوسیله آن دستوراتی که قبلاً اجرا کرده اید فراخوانی کرده و در صورت لزوم پس از اصلاح یا تغییر حتی بدون تغییر آنها را مجدداً اجرا کنید.

پوسته فرمان دارای قسمتی به نام تاریخچه (History) است که فرامینی که قبلاً وارد کرده اید را نگهداری میکند. شما میتوانید این فرامین را از تاریخچه فراخوانی کرده و استفاده کنید.

ویرایش خط فرمان

در صورتی که در تایپ یک دستور مرتکب اشتباه شده اید، میتوانید به آسانی آنرا فراخوانی کرده و مجدداً پس از ویرایش ، آنرا اجرا کنید. میتوانید از برخی کلیدهای میانبر برای راحت تر کردن این کار استفاده کنید. مثلاً کلیدهای Ctrl+a اشاره گر را به ابتدای فرمان و Ctrl+E به انتهای فرمان حرکت می دهد. همین کار را کلیدهای Home و End نیز انجام میدهند. ویرایش کردن فرمان مانند کارکردن در ویرایش گرهای متنی است و بسیار ساده است. پس اتمام ویرایش دستور، کافی است کلید Enter را برای اجرای آن فشار دهید.

کامل کردن خودکار فرمان

برای اینکه مقدار تایپ شما به حداقل برسد، پوسته فرمان فرمان ناقص شما را به روشهایی کامل میکند. . برای بکارگیری این قابلیت کافی است که ابتدا چند حرف اول فرمان مورد نظر را تایپ کرده و کلید tab را فشار دهید. در زیر برخی موارد را که میتوانید ناقص تایپ کنید می بینید :

- **متغیر های محیطی** : در صورتی که متن با یک علامت دلار شروع شود، با فشردن کلید tab ، پوسته فرمان آنرا با یک متغیر محیطی کامل خواهد کرد.

- **نام کاربری** : در صورتی که متن بوسیله یک کاراکتر ~ شروع شود، پوسته فرمان آن را بوسیله یک نام کاربری کامل خواهد کرد.

- **دستورات، نامهای مستعار یا توابع** : در صورتی که متن با یک کاراکتر عادی شروع شود، پوسته فرمان آنرا بوسیله یک دستور، نام مستعار یا تابع کامل خواهد کرد.

- **نام میزبان** : در صورتی که متن با یک علامت @ شروع شود، پوسته فرمان آنرا بوسیله یک نام میزبان که از فایل etc/hosts می خواند، کامل میکند.

موقعی وجود دارد که برای کامل کردن یک فرمان چندین گزینه وجود دارد . مثلاً چندین متغیر محیطی وجود دارد که با حرف P شروع میشوند. در این موارد در صورتی که شما دوبار کلید Tab را فشار دهید و یا کلیدهای +Esc را فشار دهید، تمام حالتها ممکن به شما نشان داده میشود :

```
$ echo $P<tab><tab> or <Esc+?>
```

```
$PATH $PPID $PS1 $PS4
```

```
$PIPESTATUS $PROMPT_COMMAND $PS2 $PWD
```

فراخوانی مجدد یک فرمان

پس از اینکه یک دستور را تایپ کردید، همانطوری که قبلاً گفتیم این دستور بطور کامل در تاریخچه پوسته فرمان ذخیره میشود. برای نمایش محتویات تاریخچه پوسته فرمان میتوانید از دستور history استفاده کنید. در صورتی که پس از آن یک عدد اضافه کنید، به تعداد آن عدد دستورات تایپ شده را نشان خواهد داد :

```
$ history 5
```

```
1023 ls
```

```
1024 cd Fonts/
```

```
1025 man more
```

```
1026 date
```

```
1027 history 5
```

برای فراخوانی دستورات تایپ شده میتوانید از روشهای زیر استفاده کنید:

- **کلیدهای مکان نما** : از کلیدهای بالا و پایین مکان نما میتوانید برای حرکت کردن در لیست تاریخچه استفاده کنید. بجای آن از کلیدهای Ctrl+n و Ctrl+p نیز میتوانید استفاده کنید.

- **کلیدهای Ctrl+r** : برای جستجوی آخر به اول یک رشته در تاریخچه استفاده میشود. برای مثال با تایپ یک یا چند حرف، دستوری که دارای آن حروف است نمایش داده میشود.

- **کلیدهای Ctrl+s** : مشابه بالا ولی جستجو بصورت اول به آخر صورت میگیرد.

روش دیگری که میتوانید از آن برای کار کردن با فرامین استفاده کنید، دستور fc است. با استفاده از این دستور، که پس از آن میتوانید شماره دستور مورد نظر در تاریخچه یا بازه ای از شماره ها را ذکر کنید، این دستورات در یک ویرایشگر متنی باز میشوند که میتوانید آنها را ویرایش کرده و خارج شوید. برای مثال دستور زیر دستورات ۱۰۰ ام تا ۱۵۰ ام تاریخچه را در ویرایشگر باز خواهد کرد :

```
$ fc 100 150
```

لیست تاریخچه در فایلی به نام .bash_history که در دایرکتوری خانگی شما قرار دارد، ذخیره میشود و در آن تا ۱۰۰۰ دستور نگهداری میشود.

اتصال و گسترش فرامین

یکی از قابلیت‌های واقعا قدرتمند پوسته فرمان، قابلیت هدایت خروجی یا ورودی یک فرمان به فرامین دیگر است. برای این منظور، همانطور که قبلا اشاره شد، از کاراکترهای ویژه استفاده میشود.

لوله بندی فرامین (Piping Commands)

کاراکتر ویژه لوله بندی کاراکتر (|) است. این کاراکتر، خروجی یک فرمان را به ورودی فرمان دیگر هدایت میکند. برای مثال :

```
$ cat /etc/passwd | sort | more
```

```
adm:x:3:4:adm:/var/adm:/sbin/nologin
```

```
Alan:x:500:500:Alan Bachumian,7852020:/home/Alan:/bin/bash
```

```
apache:x:48:48:Apache:/var/www:/sbin/nologin
```

```
bin:x:1:1:bin:/bin:/sbin/nologin
```

```
Linet:x:501:501:Linet Minasian:/home/Linet:/bin/bash
```

```
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
```

```
--More--
```

این فرمان محتویات فایل `etc/passwd` را خوانده و خروجی را به فرمان `sort` هدایت میکند. این فرمان، کاراکتر ابتدای هر سطر را گرفته و خروجی را بصورت الفبایی مرتب کرده و خروجی را به دستور `more` میفرستد و این دستور نیز خروجی را بصورت صفحه به صفحه نمایش میدهد.

قابلیت لوله بندی نمایش خوبی است از اینکه چگونه یونیکس، پدر لینوکس بر اساس قطعات مختلف نرم افزاری شکل گرفته است. مثلا در یونیکس ابزارهای مختلف را طوری به هم وصل میکردن که کارهای مختلفی بتوان با آنها انجام داد. مثال خوبی که در این مورد میشود زد: سالها پیش که واژه پردازهای گرافیکی و راحت مانند اکنون وجود نداشتند، کاربران باید ابتدا سند خود را بصورت متنی ایجاد کرده و سپس آنرا بوسیله ماکروهای خاصی فرمت بندی میکردند و بعد باید بررسی میکردند که چطور از آب در آمده است . برای این کار از فرمانی مانند زیر استفاده میشد:

```
$ nroff -man grep.1 | lpr
```

در دستور بالا از `nroff` برای فرمت کردن فایل `grep.1` با استفاده از ماکروی `man` استفاده شده و حاصل کار با استفاده از لوله بندی به خروجی چاپگر که `lpr` است فرستاده شده است.

دستورات متوالی

برخی اوقات نیاز دارید که چند فرمان بصورت همزمان با استفاده از یک فرمان اجرا شوند. این کار به سادگی امکان پذیر است. کافی است پس از اتمام هر فرمان آنرا از فرمان بعدی بوسیله یک کاراکتر سم کالن (;) جدا کنید. برای مثال :

```
$ date; troff -me mytext | lpr; ls /home
```

فرامین پس زمینه

برخی دستورات برای تمام شدن نیاز به زمان دارند. برخی اوقات مایل نیستید که پوسته فرمانتان را معطل باقی بگذارید تا دستور به اتمام برسد. برای این منظور میتوانید دستور مورد نظر را با استفاده از کاراکتر آمپرسند (&) در پس زمینه اجرا کنید. برای مثال :

```
$ troff -me mytext &
```

راههایی برای مدیریت پروسه های پیش زمینه و پس زمینه وجود دارد که جلوتر درباره آنها صحبت خواهیم کرد .

توسعه فرامین

به وسیله قابلیت جانشینی فرامین میتوانید پوسته فرمان را وادار کنید تا خروجی یک فرمان را خودش تفسیر کند، بجای اینکه این کار به خود فرمان واگذار شود. در این مورد شما میتوانید خروجی استاندارد یک فرمان را بصورت آرگومان یک دستور دیگر تعیین کنید. دو شکل قابلیت جانشینی فرامین بصورت زیر است :

```
$(sommand) or 'command'
```

برای فهمیدن این قابلیت به مثال زیر توجه کنید:

```
$ vi $(find / -print | grep xyzzy)
```

در این دستور، قبل از اجرای vi جانشینی فرامین صورت می گیرد. ابتدا دستور find از دایرکتوری ریشه شروع به کار کرده و نام تمام فایلها و دایرکتوری ها را چاپ میکند. خروجی این دستور به grep ارسال میشود و این دستور تمام آنها را که فاقد رشته xyzzy هستند را فیلتر میکند. سپس vi تمام فایلهایی را که دارای رشته xyzzy هستند را باز میکند.

توسعه عبارات حسابی

موارد زیادی وجود دارد که شما مایلید که نتایج یک جمله محاسباتی را به یک فرمان ارسال کنید . دو راه برای انجام آن وجود دارد:

```
$(expression) or $((expression))
```

برای روشن شدن مطلب به مثال زیر توجه کنید :

```
$ echo "Iam $[2003-1978] years old."
```

```
Iam 25 years old.
```

در مثال بالا، پوسته فرمان ابتدا عبارت حسابی را انجام داده و سپس نتیجه را به فرمان echo ارسال میکند.

توسعه متغیرهای محیطی

همانطور که قبلا گفتیم، متغیرهای محیطی اطلاعاتی را در مورد پوسته فرمان در بر دارند. هنگامی که یک متغیر محیطی را در یک دستور قرار میدهید، بجای اینکه نام آن چاپ شود، محتویات آن چاپ میشود :

```
$ ls -l $BASH
```

```
-rwxr-xr-x 1 root root 626188 Aug 24 2002 /bin/bash
```

در مثال بالا، دستور ls با استفاده از متغیر محیطی BASH محل آنرا چاپ میکند. در این مورد بیشتر توضیح خواهیم داد.

استفاده از متغیرهای محیطی

متغیرهای محیطی برای ذخیره اطلاعاتی مانند محل فایل‌های پیکربندی، صندوق‌های پستی و مسیر دایرکتوری‌ها بکار میروند. همچنین این متغیرها دارای مقادیری برای شکل اعلان فرمان، اندازه تاریخچه و نوع سیستم عامل نیز هستند.

برای نمایش متغیرهایی که اکنون به پوسته فرمان شما اختصاص داده شده‌اند، باید از دستور declare استفاده کنید.

برای نمایش محتویات هر یک، کافی است یک علامت دلار جلوی آن قرار داده و آن را در دستورات خط فرمان استفاده کنید:

```
$ echo $USER
```

```
Alan
```

همانطور که می‌بینید، فرمان بالا نام کاربر فعلی سیستم را نمایش میدهد.

متغیرهای محیطی عمومی

هنگامی که یک پوسته فرمان باز میکنید، متغیرهایی وجود دارند که مقادیر آنها قبلاً تخصیص داده شده است. در زیر برخی از این متغیرها نشان داده شده‌اند:

- 0 BASH : محتوی مسیر کامل برنامه پوسته فرمان است. به طور معمول bin/bash.
- 0 BASH_VERSION : شماره نسخه برنامه پوسته فرمان را نشان میدهد.
- 0 EUID : شماره شناسایی موثر کاربر فعلی را نمایش میدهد. این مقدار هنگامی که پوسته شروع میشود، تخصیص داده میشود.
- 0 HISTFILE : محل فایل تاریخچه فرامین را نمایش میدهد.
- 0 HISTFILESIZE : تعداد فرامینی که تاریخچه در خود نگهداری میکند. معمولاً ۱۰۰۰ است.
- 0 HISTCMD : شماره فرمان جاری را در تاریخچه نشان میدهد.
- 0 HOME : دایرکتوری خانگی کاربر جاری را نشان میدهد.
- 0 HOSTTYPE : نوع معماری پردازنده کامپیوتر را نشان میدهد.
- 0 MAIL : مسیر صندوق پستی کاربر جاری را نشان میدهد. معمولاً به نام شما در var/spool/mail/ قرار دارد.
- 0 OLDPWD : مسیر قبل از دایرکتوری جاری فعلی را نشان میدهد.
- 0 OSTYPE : نوع سیستم عامل را نشان میدهد. در مورد ما خروجی به صورت linux-gnu خواهد بود.
- 0 PATH : لیست دایرکتوری‌های معرفی شده را نشان میدهد. برای اجرای یک فرمان در این دایرکتوری‌ها جستجو صورت میگیرد.
- 0 PPID : شماره پروسه‌ای که پوسته فرمان را شروع کرده است، نمایش میدهد.
- 0 PROMPT_COMMAND : دستوری را که هربار پیش از نمایش اعلان فرمان اجرا میشود را نشان میدهد.
- 0 PS1 : مقدار اعلان فرمان را تخصیص میدهد. مقادیر زیادی وجود دارند که آنها را میتوانید در اعلان فرمان خود بگنجانید مانند تاریخ، زمان، نام کاربر، نام کامپیوتر و ... برخی اوقات یک فرمان به اعلان‌های بیشتری نیاز دارد که میتوانید از متغیرهای PS2 یا PS3 برای این کار استفاده کنید. در این مورد بیشتر توضیح خواهیم داد.
- 0 PWD : دایرکتوری جاری را نشان میدهد.
- 0 RANDOM : با مراجعه به این متغیر یک شماره تصادفی بین ۰ و ۹۹۹۹ تولید میشود.
- 0 SECONDS : تعداد ثانیه‌ای که پوسته فرمان آغاز به کار کرده است.
- 0 UID : شماره شناسایی اصلی کاربر فعلی را نمایش میدهد. این شماره در فایل etc/passwd ذخیره شده است.

ایجاد متغیرهای محیطی خاص

از متغیرهای محیطی میتوانید برای ذخیره اطلاعاتی که معمولاً در پوسته فرمان استفاده میکنید، بهره‌برداری کنید. شما میتوانید هر گونه متغیر محیطی به دلخواه خود ایجاد کنید. برای ایجاد موقت یک متغیر محیطی میتوانید نام متغیر و مقدار آن را جلوی اعلان فرمان تایپ کنید:

```
$ AB=/usr/locl/documents; export AB
```

مثال بالا مسیر یک دایرکتوری را به یک متغیر به نام AB اختصاص میدهد. دستور export این متغیر را به پوسته فرمان صادر میکند. بنابراین در صورتی که پوسته های فرمان دیگری نیز اجرا شوند، این متغیر در آنها موجود خواهد بود.

نکته : ممکن است توجه کرده باشید که تمام متغیرهای محیطی با حروف بزرگ تعریف شده اند. این کار یک رسم است نه یک الزام. یعنی در صورتی که نام متغیری را با حروف کوچک تعیین کنید، باز هم کار خواهد کرد. البته توجه داشته باشید که متغیر xyz با XYZ یکی نیست.

مشکلی که در ایجاد این گونه متغیرهای محیطی وجود دارد این است که موقت بوده و با خروج از پنجره پوسته ای که این متغیر در آن تخصیص داده شده است، این متغیر پاک خواهد شد. برای اختصاص دائمی این متغیرها، باید آنها را به فایلهای پیکربندی پوسته فرمان اضافه کنید. این موضوع جلوتر توضیح داده خواهد شد.

در صورتی که مایلید متنی درست جلوی مقدار یک متغیر محیطی قرار گیرد، کافی است که متغیر را در دو پرانتز قرار داده و متن مورد نظر را جلوی آن قرار دهید. برای مثال :

```
$ echo ${HOME} /Documents
```

```
/home/Alan/Documents
```

به خاطر داشته باشید که برای استفاده از متغیرها یا باید آنها را export کنید و یا به فایل پیکربندی پوسته فرمان اضافه نمایید. دستور export بسیار قابل انعطاف است. مثلا میتوانید در هنگام صادر کردن متغیر، مقدار آنرا هم تخصیص دهید :

```
$ export XYZ=/home/Alan/Documents
```

و یا میتوانید با حفظ مقادیر قبلی، مقداری را به یک متغیر اضافه نمایید :

```
$ export PATH=$PATH:/home/Alan/Documents
```

در مثال بالا، دایرکتوری home/Alan/Documents به طور موقت به متغیر PATH اضافه شده است.

در صورتی که احساس کردید دیگر به یک متغیر نیازی ندارید، میتوانید با استفاده از دستور unset آنرا پاک کنید:

```
$ unset XYZ
```

همانطور که دیدید، برای پاک کردن متغیر نیازی به علامت دلار نیست.

مدیریت پروسه های پس زمینه و پیش زمینه

در صورتی که از لینوکس در محیط شبکه ای و با استفاده از یک ترمینال متنی استفاده میکنید، پوسته فرمان تنها چیزی است که میتوانید از آن استفاده کنید و از محیطهای گرافیکی خبری نخواهد بود. در صورتی که نیاز داشته باشید در آن واحد با چندین برنامه کار کنید، این مسئله بسیار محدود کننده خواهد بود.

با اینکه پوسته فرمان محیطی گرافیکی برای اجرای برنامه ها ندارد، ولی قابلیتی دارد که با استفاده از آن میتوانید برنامه های فعال را بین پس زمینه و پیش زمینه جابجا نمایید. با این وسیله میتوانید تعداد زیادی برنامه را در یک زمان در حال اجرا داشته باشید و بین آنها حرکت کنید.

راههای گوناگونی برای قرار دادن یک برنامه در پس زمینه وجود دارد. قبلا اشاره کردیم که با اجرای برنامه ای که به آخر آن یک کاراکتر آمپرسند (&) اضافه شده است، در پس زمینه قرار میگیرد. روش دیگر استفاده از دستور at برای اجرای برنامه ها بصورتی که به پوسته متصل نباشند، است.

برای توقف اجرای یک فرمان و قرار دادن آن در پس زمینه، از کلیدهای Ctrl+z استفاده کنید. پس از اینکه اجرای دستور متوقف شد، با استفاده از دستور fg می‌توانید آنرا به پیش زمینه آورده، استفاده کنید و یا با دستور bg آنرا در پس زمینه بکار بگیرید.

شروع پروسه های پس زمینه

در صورتی که برنامه هایی دارید که مایلید در هنگام کار کردن شما در پس زمینه اجرا شوند، پس از دستور ، یک علامت آمپرسند (&) در پایان آن اضافه کنید. برای مثال :

```
$ find /usr -print > /home/Alan/usrfiles &
```

این دستور تمام فایل‌های موجود در دایرکتوری /usr لینوکس شما را در فایلی به نام usrfiles ذخیره میکند. علامت آمپرساند باعث میشود که این فرمان در پس زمینه اجرا شود. برای دیدن اینکه چه برنامه هایی در پس زمینه در حال اجرا هستند، از دستور jobs استفاده کنید:

```
$ jobs
```

```
[1]- Stopped mc
```

```
[2]+ Stopped vi
```

```
[3] Running find /usr -print >usrfiles &
```

همانطور که در خروجی فرمان بالا مشاهده میکنید، سه برنامه mc ، vi و دستور find در حال اجرا در پس زمینه هستند. علامت مثبت در کنار برنامه دوم نشان میدهد که این آخرین پروسه ای است که در حالت پس زمینه اجرا شده است و علامت منفی نشاندهنده پروسه ای است که قبل از آخرین پروسه، در پس زمینه قرار داده شده است. بعلاوه اینکه برنامه های اول و دوم برای کارکرد به خروجی ترمینال نیاز دارند تا زمانی که در حالت پیش زمینه اجرا شوند، متوقف باقی خواهند ماند. ولی برنامه find که به خروجی ترمینال نیازی ندارد، در حال اجرا میباشد.

نکته : برای نمایش شماره پروسه برنامه های پس زمینه ، می‌توانید گزینه l را به فرمان jobs اضافه نمایید. در صورتی که از دستور ps برای نمایش پروسه های فعال استفاده کنید، می‌توانید ببینید که کدامیک از آنها دستوری است که در پس زمینه در حال اجراست.

استفاده از فرامین پس زمینه و پیش زمینه

در ادامه مثالی که در بالا ذکر شد، برای برگرداندن برنامه vi به پیش زمینه می‌توانید از دستور زیر استفاده کنید:

```
$ fg %2
```

با این دستور، برنامه vi مجدداً روی پوسته فرمان نمایش داده خواهد شد. با فشردن کلیدهای Ctrl+z می‌توانید مجدداً آنرا به پس زمینه بفرستید.

هشدار : قبل از اینکه یک برنامه واژه پرداز و یا برنامه ای که اطلاعات ذخیره نشده دارد را به پس زمینه ارسال کنید، اطلاعات آنرا ذخیره نمایید. برنامه های پس زمینه به سادگی فراموش میشوند و ممکن است اطلاعات خود را از دست بدهید.

همانطوری که دیدید برای نمایش یک برنامه پس زمینه از علامت درصد و شماره آن که در دستور jobs مشخص شده بود استفاده شد. علاوه بر شماره، می‌توانید بجای آن نام برنامه و یا قسمتی از نام برنامه که ابتدای آن علامت سوال قرار داده شده استفاده کنید. این کار هنگامی که دو برنامه مشابه به همراه دو فایل متفاوت باز هستند، به شما کمک خواهد کرد. برای روشن شدن مطلب به مثال زیر توجه کنید :

\$ jobs

[2] Stopped vi

[3]- Stopped mc

[4]+ Stopped vi ./mytext

\$ fg %?my

با تایپ دستور fg %?my برنامه vi که در حال ویرایش فایل mytext است، در پوسته فرمان نمایش داده خواهد شد.

پیکربندی پوسته فرمان

برای اینکه بتوانید بطور موثرتری از پوسته فرمان خود استفاده کنید، میتوانید آنرا بنا به خواسته خود تنظیم کنید. برای این منظور باید فایل‌های پیکربندی پوسته فرمان خود را ویرایش کنید.

تعدادی فایل پیکربندی وجود دارد که نحوه رفتار پوسته فرمان شما را تعیین میکند. برخی از این فایل‌ها برای تمام کاربران و پوسته‌ها مشترک بوده و برخی مخصوص یک کاربر خاص هستند. فایل‌های پیکربندی زیر فایل‌هایی هستند که هر کاربر پوسته فرمان در لینوکس از آنها استفاده میکند :

- 0 etc/profile : این فایل اطلاعات محیط کاربری هر کاربر را ذخیره میکند. این فایل هنگامی اجرا میشود که شما به سیستم وارد شده و پوسته فرمان آغاز به کار میکند. این فایل مقادیر پیش‌گزینه مسیری، شکل اعلان فرمان، حداکثر تعداد فایلی که شما میتوانید ایجاد کنید و مجوز‌های پیش‌گزینه برای فایل‌هایی که ایجاد میکنید را تعیین میکند. همچنین این فایل متغیرهای محیطی مانند محل صندوق پستی و اندازه فایل‌های تاریخچه را تنظیم میکند.
- 0 etc/bashrc : این فایل برای هر کاربری که پوسته bash را اجرا میکند، اجرا میشود. این فایل حالت اعلان فرمان را تنظیم میکند. مقادیر این فایل میتواند توسط فایل bashrc که در دایرکتوری خانگی هر کاربر وجود دارد، تحت تأثیر قرار گیرد.
- 0 ~/.bashrc : این فایل حاوی اطلاعات مربوط به bash هر کاربر میباشد. این فایل هنگامی خوانده میشود که به سیستم وارد میشود و هر گاه که یک پوسته جدید باز میکنید. اینجا بهترین مکان برای ذخیره متغیرهای محیطی و فرمان‌های مستعار خاص خودتان است.
- 0 ~/.bash_profile : این فایل برای وارد کردن اطلاعات خاصی که هر کاربر در استفاده از پوسته بکار میبرد میباشد. این فایل تنها یکبار اجرا میشود. هنگامی که کاربر به سیستم وارد میشود. این فایل تعدادی از متغیرهای محیطی را مقداردهی کرده و فایل bashrc مربوط به کاربر را اجرا میکند.
- 0 ~/.bash_logout : این فایل هر گاه که شما از سیستم خارج میشوید اجرا میشود. این فایل فقط صفحه نمایش را پاک میکند.

برای تغییر فایل‌های etc/profile و etc/bashrc باید با کاربر ریشه وارد سیستم شده باشید. هر کاربر میتواند اطلاعات موجود در فایل‌های bashrc، bash_profile و bash_logout موجود در دایرکتوری‌های خود را تغییر دهد.

در قسمت زیر با برخی تنظیمات فایل‌های پیکربندی پوسته فرمان آشنا میشوید. در بیشتر موارد، تغییرات در فایل bashrc موجود در دایرکتوری خانگی صورت میگیرد. هرچند در صورتی که شما یک مدیر سیستم باشید، ممکن است این تنظیمات را برای کل کاربران خود اعمال کنید.

تنظیم اعلان فرمان

اعلان فرمان شما از تعدادی کاراکتر تشکیل شده است که هر گاه که به نمایش در می‌آید، معنی آن این است که پوسته فرمان آماده دریافت فرمان جدید است. محتویات اعلان فرمان در متغیر محیطی PS1 قرار دارد. در صورتی که پوسته فرمان شما به ورودی بیشتری نیاز داشته باشد، از مقادیر PS2، PS3 و PS4 نیز استفاده خواهد شد.

هنگامی که سیستم لینوکس شما نصب میشود، اعلان فرمان طوری تنظیم میشود که حاوی اطلاعات زیر باشد: نام کاربری شما، نام کامپیوتر شما و نام دایرکتوری که اکنون در آن قرار دارید. این اطلاعات در میان دو براکت قرار گرفته و در انتهای آن برای کاربران عادی یک علامت دلار (\$) و برای کاربر ریشه علامت پوند (#) قرار دارد. در زیر مثالی از یک اعلان فرمان را ببینید:

[alan@Memphis alan]\$

این امکان وجود دارد تا اطلاعات مختلفی را به اعلان فرمانتان اضافه کنید. این اطلاعات میتواند شامل شماره ترمینال، تاریخ، زمان و اطلاعات دیگر باشد. برای مثال :

- \! : شماره فعلی تاریخچه فرمان را نشان میدهد.
- \# : شماره دستور آخرین دستور را نشان میدهد.
- \\$: اعلان فرمان استاندارد را نشان میدهد.
- \W : فقط دایرکتوری کاری جاری را نشان میدهد.
- \l : فقط یک بک اسلش نشان داده میشود.
- \d : روز، ماه و شماره روز را نمایش میدهد. مثلا : ۲۳ Sat Jan
- \h : نام کامپیوتر میزبان را نشان میدهد.
- \n : یک خط جدید باز میکند.
- \s : نام پوسته فرمان را نشان میدهد. مثلا bash
- \t : زمان را بصورت ساعت، دقیقه و ثانیه نمایش میدهد. برای مثال : ۱۰:۱۴:۴۰
- \u : نام کاربر را نمایش میدهد.
- \W : مسیر کامل دایرکتوری جاری را نمایش میدهد.

 **نکته :** در صورتی که اعلان فرمان خود را به صورت موقت با تایپ مقادیر مربوطه در پوسته فرمان تغییر میدهید، باید مقادیر PS1 را بین دو گیومه قرار دهید . مثلا دستور:

```
export PS1="[ \t \w] \$"
```

میدهد اعلان فرمان را به صورت زیر نشان

```
[20:25:40 /var/spool/mail]$
```

برای ایجاد تغییرات دائمی در اعلان فرمان، باید مقدار PS1 را به فایل bashrc موجود در دایرکتوری خانگی خود اضافه کنید. معمولا این مقدار قبلا وجود دارد و کافی است آنرا تغییر دهید.

تنظیم اسامی مستعار

در لینوکس این امکان وجود دارد تا برای آسانی بیشتر، اسامی مستعاری را بجای فرمان اصلی تعیین کنید. برای اضافه کردن اسامی مستعار باید از دستور alias استفاده کنید. به مثالهای زیر توجه کنید:

```
$ alias p='pwd; ls -CF'
```

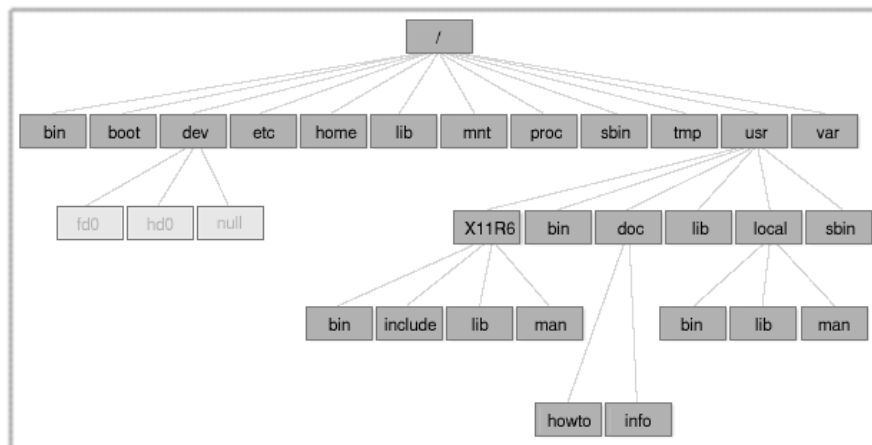
```
$ alias rm='rm -i'
```

در مثال نخست حرف p دستور pwd را اجرا کرده و پس از آن دستور ls -CF اجرا خواهد شد که محتویات دایرکتوری جاری را چاپ خواهد کرد. در مثال دوم، دستور rm طوری تنظیم شده است تا فقط با گزینه i اجرا شود.

در صورتی که دستور alias را به تنهایی تایپ کنید، لیستی از اسامی مستعاری که تنظیم کرده اید نمایش داده میشود. توجه داشته باشید که اسامی مستعار در یک فایل پیکربندی ذخیره شده و با بستن پوسته فرمان از بین نمی روند.

کار کردن با سیستم فایل لینوکس

سیستم فایل لینوکس ساختاری است که اطلاعات شما را در کامپیوتر ذخیره میکند. فایلها در یک ساختار درختی از دایرکتوری ها ذخیره میشوند. هر دایرکتوری میتواند حاوی فایلها و دایرکتوری های دیگری باشد. در صورتی که بخواهید ساختار سیستم فایل لینوکس را دقیق تر توصیف کنید، آن بیشتر شبیه یک درخت وارونه است. در بالاترین نقطه، دایرکتوری ریشه قرار دارد که بوسیله یک اسلش تنها نشان داده میشود. در زیر آن دایرکتوری های عمومی و سیستمی سیستم عامل لینوکس قرار میگیرند. مانند bin, dev, home و tmp. هر کدام از این دایرکتوری ها محتوی دایرکتوری های دیگری هستند. تصویر ۲ ساختار درختی سیستم فایل لینوکس را نشان میدهد.



تصویر ۲ ساختار سیستم فایل لینوکس

برخی از دایرکتوری های مهم سیستم فایل لینوکس در زیر توضیح داده شده اند.

- bin : فرامین عمومی سیستم عامل لینوکس در این دایرکتوری قرار دارند. مانند ls, sort و chmod.
- dev : حاوی نقاط دسترسی به ابزارهای سخت افزاری کامپیوتر شما است. مانند ترمینالها (tty)، دیسکهای فلاپی (fd)، دیسکهای سخت (hd) و ... کاربران بطور معمول برای دستیابی به این دستگاهها از نامهای آنها استفاده میکنند.
- etc : حاوی برخی فایلهای پیکربندی سیستم است.
- home : دایرکتوری های کاربران یک سیستم لینوکس در این دایرکتوری قرار میگیرد.
- mnt : محلی را برای متصل کردن ابزارها و دیسکها مانند فلاپی، CD-ROM و درایوهای شبکه ایجاد میکند.
- root : دایرکتوری خانگی کاربر ریشه است.
- sbin : دستورات مدیریتی سیستم در این دایرکتوری قرار میگیرند.
- tmp : محل قرارگیری فایلها موقت.
- usr : محل قرارگیری مستندات سیستم، بازی ها، فایلهای گرافیکی، کتابخانه ها و بسیاری چیزهای دیگر.

ساختار سیستم فایل در داس و ویندوز با ساختار آن در لینوکس متفاوت هستند. با وجودی که شباهت هایی نیز در این میان دیده میشود ولی تفاوت های عمده به شرح زیر هستند:

- در داس و ویندوز برای دسترسی به ابزارهای ذخیره سازی مختلف و پارتیشن های مختلف دیسک سخت از حروفی که به نام درایو موسوم بودند استفاده میکردید. مانند A برای فلاپی، C برای دیسک سخت و ... در لینوکس تمام ابزارهای ذخیره سازی در دل سیستم فایل باهم ادغام شده اند. مثلاً محتویات یک فلاپی دیسک در مسیر mnt/floppy قرار میگیرد و ... ممکن است در ابتدای کار اصلاً به این سیستم عادت نداشته باشید ولی پس از مدتی به آن عادت خواهید کرد. محلی که شما برای ذخیره فایلها و اطلاعات خود استفاده خواهید کرد، همان دایرکتوری خانگی شماست.
- در سیستم فایل داس و ویندوز برای جدا کردن پوشه ها و مسیرها از یک اسلش استفاده میشود در حالی که در لینوکس از اسلش استفاده میشود.
- نام فایلها در داس و ویندوز همیشه دارای یک پسوند بوده اند. مانند txt برای فایلهای متنی و ... پسوند فایلها برای لینوکس و یونیکس لازم نیستند. سیستم فایل لینوکس بدون توجه به پسوند، نوع فایل را تشخیص میدهد.
- هر فایل و دایرکتوری در لینوکس دارای مجوزها و خصوصیات است که از دسترسی کاربران غیر مجاز به آن جلوگیری کرده و یا این دسترسی را محدود میکند. در بیشتر سیستمهای داس و ویندوز از این مجوزها خبری نیست زیرا این سیستمها در ابتدا بصورت سیستمهای تک کاربره طراحی و پیاده سازی شده اند. در سیستمهای ویندوز، سیستمهای مبتنی بر ویندوز NT که بصورت چند کاربره هستند این مجوزها پیاده سازی شده است.

ایجاد فایلها و دایرکتوری ها

به عنوان یک کاربر لینوکس ، همانطور که در بالا اشاره کردم، بیشتر فایلها و دایرکتوری های خود را در دایرکتوری خانگی خود ایجاد و ذخیره خواهید کرد. در اینجا با چند دستور که در این زمینه مفید هستند آشنا میشوید:

- `cd` : این دستور مسیر جاری را به مسیر دیگری که تعیین میکنید تغییر میدهد حتما با مشابه این دستور در داس کار کرده اید.
- `pwd` : این دستور مسیر دایرکتوری فعلی را چاپ میکند.
- `mkdir` : این دستور یک دایرکتوری ایجاد میکند.
- `chmod` : این دستور برای تغییر مجوزهای فایل و دایرکتوری بکار میرود.
- `ls` : این دستور محتویات یک دایرکتوری یا مسیر را چاپ میکند. مشابه دستور `dir` در داس.

خوب اکنون به کمی تمرین برای بکارگیری این دستورات می پردازیم. در صورتی که در حالت گرافیکی هستید، همانطور که قبلا گفته شد، یک پنجره ترمینال باز کنید.

۱. برای حرکت به دایرکتوری خانگی خود از هر جا ، کافی است دستور `cd` را تایپ کنید.

۲. برای حصول اطمینان از قرارگیری در دایرکتوری خانگی خود، دستور `pwd` را تایپ کنید:

```
$ pwd
```

```
/home/alan
```

۳. با استفاده از دستور `mkdir` یک دایرکتوری به نام `test` ایجاد کنید:

```
$ mkdir test
```

۴. مجوزهای دایرکتوری ایجاد شده را با استفاده از دستور `ls` بررسی کنید:

```
$ ls -ld test
```

```
drwxrwxr-x 3 alan alan 4096 May 17 20:14 test
```

خروجی فرمان نشان میدهد که `test` یک دایرکتوری بوده و مالک آن کاربری به نام `alan` است که به گروه `alan` تعلق داشته و آخرین بار در ۱۷ ام می در ساعت ۱۴:۳۰ دقیقه تغییر کرده است. تصور کنید میخواهید مجوزهای این دایرکتوری را طوری تنظیم کنید که افراد دیگری که از این کامپیوتر استفاده میکنند نتوانند محتویات دایرکتوری شما را دیده و استفاده کنند. در این مورد بیشتر توضیح خواهم داد.

۵. اکنون دستور زیر را تایپ کنید:

```
$ chmod 700 test
```

این دستور به شما تمام مجوزهای استفاده و تغییر دایرکتوری را میدهد در حالی که به دیگران اجازه حتی مشاهده محتویات این دایرکتوری نیز داده نخواهد شد. اگر مجدداً دستور `ls` که در بالا تایپ کردید را بکار ببرید، این بار مجوزها بصورت `-----drwx` نمایش داده خواهد شد.

۶. در این مرحله با استفاده از دستور `cd` به دایرکتوری `test` وارد شوید:

```
$ cd test
```

هنگامی که نیاز داشتید تا بدانید دایرکتوری خانگی شما در چه مسیری قرار دارد میتوانید از یکی از دو راه زیر استفاده کنید :

- متغیر محیطی `HOME`
- علامت `~`

با تایپ یکی از موارد بالا مقابل اعلان فرمان ، مسیر دایرکتوری خانگی شما نمایش داده میشود:

\$ ~

/home/alan

برای نمایش دایرکتوری خانگی یک کاربر دیگر کافی است به صورت زیر عمل کنید:

\$ ~ chris

/home/chris

در حرکت بین دایرکتوری ها و کارکردن در آنها فرامین دیگری نیز وجود دارند که بسیار مفید هستند:

- یک نقطه : نشاندهنده مسیر جاری است. مثلا :

\$ cp /usr/local/mygame .

دستور بالا فایل mygame را به مسیر جاری (که دایرکتوری خانگی تان بود) کپی میکند.

- دو نقطه : نشاندهنده مسیر ماقبل است. مثلا :

\$ mv mygame ..

دستور بالا فایل mygame را به مسیر بالاتر دایرکتوری خانگی تان (دایرکتوری home) منتقل میکند.

- متغیر محیطی OLDPWD : نشاندهنده دایرکتوری جاری قبل از دایرکتوری فعلی است.

استفاده از کاراکترهای ویژه و عملگرهای خط فرمان

برای استفاده کارآمد تر از پوسته فرمان ، کاراکترهای مخصوصی وجود دارند که به کاراکترهای ویژه و عملگرها موسوم هستند. با کاراکترهای مخصوص می‌توانید در تایپ کامل نام یک یا چند فایل صرفه جویی کرده و با استفاده از عملگرها اطلاعاتی را از یک فایل یا دستور به یک دستور یا فایل دیگر هدایت کنید.

استفاده از کاراکترهای ویژه مخصوص نام فایلها

برای کم کردن مقدار تایپ و انتخاب آسانتر دسته ای از فایلها ، پوسته فرمان به شما امکان استفاده از کاراکترهای ویژه را میدهد. کاراکترهای ویژه ای که از آنها می‌توانید بین نام فایلها استفاده کنید عبارتند از :

- علامت ستاره (*) : می‌تواند بجای هر تعدادی از کاراکترها قرار گیرد.
- علامت سوال (?) : می‌تواند بجای یک کاراکتر قرار گیرد.
- علامت دو براکت ([...]) : تمام کاراکتر های ذکر شده در براکت در انتخاب فایلها اثر می‌گذارند.

برای تمرین بکارگیری این کاراکترها به یک دایرکتوری خالی (مانند دایرکتوری test که قبلا ایجاد کردید) رفته و با استفاده از دستور زیر دسته ای از فایلهای خالی را ایجاد کنید :

\$ touch apple banana grape grapefruit watermelon

حال برای درک بهتر چگونگی عملکرد کاراکترهای ویژه از دستور ls استفاده میکنیم. به خروجی هر فرمان توجه کنید :

\$ ls a*

apple

\$ ls g *

grape

grapefruit

\$ ls g * t

grapefruit

\$ ls * e *

apple grape grapefruit watermelon

\$ ls * n *

banana watermelon

مثال نخست هر فایلی را که با کاراکتر a شروع میشود را نمایش میدهد. مثال بعدی تمام فایلهایی را که با g شروع میشوند نمایش میدهد. در مثال بعدی فایلهایی که با g شروع شده و به t ختم میشوند نمایش داده میشوند و در دو مثال بعدی فایلهایی که حاوی e و n هستند نمایش داده میشوند.

به چند مثال هم در مورد کاراکتر علامت سوال توجه کنید:

\$ ls ???e

apple grape

\$ ls g???e *

grape grapefruit

در مثال اول فایلهایی که دارای ۵ حرف بوده و حرف آخر آنها e است نمایش داده میشوند. در مثال دوم فایلهایی که با g شروع شده و کاراکتر پنجم آنها e است را نمایش میدهد.

حال مثالهایی در مورد براکتها:

\$ ls [abw] *

allpe banana watermelon

\$ ls [agw] * [ne]

apple grape watermelon

در مثال نخست تمام فایلهایی که با a، b و w شروع میشوند نمایش داده میشود. در مثال دوم تمام فایلهایی که با a، g و w شروع شده و به n یا e ختم میشوند، نمایش داده میشوند.

استفاده از کاراکترهای ویژه مخصوص هدایت فایلها

دستورات ورودی خود را از ورودی استاندارد دریافت کرده و روی خروجی استاندارد نمایش میدهند. با استفاده از لوله بندی که قبلا شرح داده شد، میتوانستیم خروجی یک دستور را به ورودی دستور دیگر متصل کنیم. با فایلها میتوانیم از کاراکترهای کوچکتر از (<) و بزرگتر از (>) برای هدایت داده ها از/به فایلها استفاده کنیم. این کاراکترها عبارتند از:

- کاراکتر < : محتویات یک فایل را به یک دستور هدایت میکند.
- کاراکتر > : خروجی یک فرمان را به یک فایل هدایت کرده و در صورتی که فایلی به همان نام وجود داشته باشد، آنرا پاک میکند.
- کاراکتر >> : خروجی یک دستور را به یک فایل هدایت کرده و در صورتی که فایلی به همان نام وجود داشته باشد، اطلاعات به آخر آن اضافه خواهد شد.

برای درک بهتر به مثالهای زیر توجه کنید:

```
$ mail root < ~/.bashrc
```

```
$ nroff -man /usr/share/man/man1/chmod.1* > /tmp/chmod
```

```
$ echo "I finished the project on $(date)" >> ~/projects
```

در مثال نخست محتویات فایل .bashrc در دایرکتوری خانگی، در یک پیام پست الکترونیک به کاربر root کامپیوتر ارسال میشود. در مثال دوم، صفحه کمک دستور chmod با استفاده از دستور nroff فرمت بندی شده و خروجی به فایل tmp/chmod ارسال میشود. مثال آخر نیز باعث خواهد شد تا خط زیر به فایل projects که در دایرکتوری خانگی کاربر وجود دارد، اضافه شود:

```
I finished the project on Sun May 25 14:25:36 IRST 2003
```

درک مجوزهای فایلها (File Permissions)

پس از اینکه مدتی با لینوکس کار کردید، مطمئنا به پیامهایی مانند Permission Denied برخورد خواهید کرد. مجوزهای فایلها و دایرکتوری ها در لینوکس به این علت ایجاد شده اند که از دسترسی کاربران به فایلها و اطلاعات خصوصی کاربران دیگر جلوگیری به عمل آورده و از فایلهای سیستمی در مقابل آسیب دیدگی حفاظت کنند. به این علت به هر فایل ۹ بیت اضافه میشود که معرف چگونگی دسترسی شما و دیگران به آن فایل خواهد بود. این بیتها بصورت rwxrwxrwx نمایش داده میشوند. نخستین سه بیت تعیین کننده دسترسی مالک فایل است. سه بیت بعدی برای گروه مالک و سه بیت بعدی برای تعیین نحوه دسترسی دیگران است. ۲ نشانگر خواندن، w نشانگر نوشتن و x نشانگر اجازه اجرا هستند. در صورتی که بجای یکی از این حروف علامت دش (-) نمایش داده شود، به این معنی است که این اجازه غیر فعال است. برای نمایش مجوزهای هر فایل یا دایرکتوری میتوانید از دستور ls -ld استفاده کنید. به مثال زیر توجه کنید :

```
$ ls -ld ch3 test
```

```
-rw-rw-r-- 3 alan alan 4096 May 22 15:11 ch3
```

```
drwxr-xr-x 3 alan alan 4096 May 17 20:14 test
```

خط نخست فایلی را نشان میدهد که دارای مجوز خواندن و نوشتن برای مالک و گروه است. سایر کاربران فقط اجازه خواندن فایل را دارا هستند. این به این معنی است که آنها میتوانند فایل را ببینند ولی هیچ تغییری نمی توانند در آن اعمال کنند. خط دوم یک دایرکتوری است. دقت کنید که مجوزها با حرف d که به معنی دایرکتوری است آغاز شده است. مالک دایرکتوری دارای اجازه خواندن، نوشتن و اجرا است. در نتیجه تنها مالک میتواند فایلها را در این دایرکتوری اضافه کرده، تغییر داده و پاک کند. بقیه کاربران تنها اجازه خواندن دارند. یعنی میتوانند به این دایرکتوری وارد شده و محتویات آنرا ببینند.

در صورتی که شما مالک یک فایل باشید، میتوانید مجوزهای آنرا مطابق نیاز خودتان تنظیم کنید. این کار بوسیله دستور chmod امکان پذیر است. برای هریک از مجوزهای خواندن، نوشتن و اجرا عددی در نظر گرفته شده است. خواندن ۴، نوشتن ۲ و اجرا ۱. بنابراین برای اینکه تمام مجوزها را به خودتان بدهید، مقدار سه بیت نخست باید ۷ تعیین شود. (۴+۲+۱). برای گروه و سایرین نیز میتوانید بنا به نیازشان مجوز تعیین کنید. مجوزها بین ۷ (مجوز کامل) و ۰ (هیچ مجوزی!) متغیر هستند. برای روشن شدن بهتر مطلب به مثالهای زیر توجه کنید :

```
$ chmod 777 files = rwxrwxrwx
```

```
$ chmod 755 files = rwxr-xr-x
```

```
$ chmod 644 files = rw-r--r--
```

```
$ chmod 000 files = -----
```

هنگامی که یک فایل ایجاد میکنید، مجوز پیش گزیده آن ۶۴۴ خواهد بود. در مورد دایرکتوری این مجوز ۷۵۵ است. این مقادیر پیش گزیده توسط دستور umask تعیین میشود. برای نمایش مقدار umask دستور زیر را تایپ کنید:

```
$ umask
```

```
022
```

کافی است اعدادی که در دستور umask مشاهده میکنید، از ۷ کم کنید. با این کار مقادیر پیش گزیده را برای دایرکتوری مشاهده خواهید کرد. در مورد فایلها باید این اعداد را از ۶ کم کنید. زیرا در مورد فایلها به طور پیش گزیده مجوز اجرا (با مقدار ۱) غیر فعال است.

نکته:  برای تغییر تعداد زیادی از فایلها در یک زمان باید از گزینه R دستور chmod استفاده کنید. این امکان وجود دارد که با یک فرمان مجوزهای تمام فایلها و دایرکتورهای درون یک ساختار دایرکتوری را تغییر دهد. برای مثال برای تغییر مجوزهای تمام فایلها و دایرکتورهای موجود در مسیر tmp/test میتوانید دستور زیر را تایپ کنید:

```
$ chmod -R 777 /tmp/test
```

هشدار:  گزینه R دستور chmod هنگام اعطا مجوزهای کامل و اعطا مجوز اجرا بسیار خوب است. ولی در صورتی که دستور بالا را بجای مقدار ۷۷۷ با مقدار ۶۴۴ اجرا کنید دیگر نمی توانید به هیچ یک از دایرکتورهای موجود در آن مسیر وارد شوید.

انتقال، کپی و پاک کردن فایلها

کپی، انتقال و پاک کردن فایلها بسیار آسان است. برای انتقال یک فایل باید از دستور mv استفاده کنید. برای کپی کردن فایلها دستور cp وجود دارد و برای پاک کردن فایلها نیز دستور rm قابل استفاده است. به مثالهای زیر توجه کنید:

```
$ mv abc def
```

```
$ mv abc ~
```

```
$ cp abc def
```

```
$ cp abc ~
```

```
$ rm abc
```

```
$ rm *
```

دستور نخست نام فایل abc را به def تغییر میدهد. دستور دوم این فایل را به دایرکتوری خانگی کاربر (~) منتقل میکند. دستور سوم، فایل abc را به فایل def کپی کرده و دستور چهارم آنرا در دایرکتوری خانگی کاربر کپی میکند. دستور پنجم فایل abc را پاک میکند در حالی که دستور ششم تمام محتویات دایرکتوری جاری را پاک خواهد کرد.

نکته:  برای کاربر ریشه، دستور rm به کمک دستور alias طوری تنظیم شده است که برای پاک کردن فایلها حتما از کاربر ریشه سوال شود. این اقدام از پاک شدن تصادفی تعداد زیادی از فایلها در اثر اشتباه جلوگیری به عمل میآورد.

والسلام