



آزمایشگاه سیستم عامل

دستور کار ۲: مدیریت فایل

بخش اول:

برای اجرای دستورات لازم است ترمینال سیستم عامل لینوکس خود را باز کنید. برای این کار می‌توانید از کلید میانبر **Ctrl+Alt+T** نیز استفاده کنید.

در ادامه به معرفی دستورات مربوط به مدیریت فایل در سیستم عامل لینوکس پرداخته خواهد شد.

- **دستور ls:** از این دستور برای لیست کردن فایل‌ها و دایرکتوری‌ها استفاده می‌شود. برای تنظیم کردن این دستور سوییچ‌های مختلفی ارائه شده است که هر کدام کار خاص خود را انجام می‌دهند.

`ls [options]`

در زیر لیست برخی از سوییچ‌ها قابل مشاهده است.

لیست برخی از سوییچ‌های دستور ls	
-l	نشان دادن جزئیات بیشتر در لیست
-1	در هر خط فقط یک فایل لیست شود.
-t	بر اساس زمان تغییر یافتن مرتب می‌شود و آخرین تغییر در اول می‌آید.
-r	بر عکس کردن اصل مرتب‌سازی
-s	برای چاپ میزان حافظه مصرف شده برای هر فایل
-R	زیر دایرکتوری‌ها را به صورت بازگشتی لیست کند.

- **دستور cp:** این دستور برای کپی کردن فایل‌ها و دایرکتوری‌ها استفاده می‌شود.
`cp [options] source destination`
- **دستور mv:** این دستور برای جابه‌جا کردن یک فایل یا دایرکتوری و همچنین برای تغییر نام آن‌ها به کار می‌رود.
`mv [options] source destination`

لیست برخی از سویچ‌های دستورهای mv و cp	
-b	از فایل‌های رونویسی شده پشتیبان تهیه می‌کند.
-p	صفات را حفظ می‌کند.

- دستور **rm**: این دستور برای پاک کردن یک فایل یا دایرکتوری به کار می‌رود.

`rm [options] file`

لیست برخی از سویچ‌های دستور rm	
-R, -r	برای پاک کردن دایرکتوری‌ها و محتوای داخل آن به صورت بازگشتی
-f	حذف کردن به صورت اجباری
-i	قبل از هر حذف از کاربر سوال می‌کند.

- دستور **mkdir**: این دستور برای ساختن دایرکتوری‌ها استفاده می‌شود.

`mkdir [options] dir_name`

- دستور **rmdir**: این دستور برای پاک کردن دایرکتوری خالی به کار می‌رود.

`rmdir [options] dir_name`

اگر دایرکتوری خالی نباشد برای پاک کردن آن از دستور `rm -r dir_name` استفاده می‌شود.

- دستور **cd**: این دستور برای تغییر دایرکتوری جاری و رفتن به دایرکتوری دیگر استفاده می‌شود.

`cd dir_name`

✓ برای برگشتن به عقب به اندازه یک دایرکتوری از دستور `cd ..` استفاده می‌کنیم.

✓ برای رفتن به شاخه `root` از دستور `cd /` استفاده می‌کنیم.

✓ دستور `cd ~` برای رفتن به دایرکتوری خانگی استفاده می‌شود.

- علائم (**wildcards**) را می‌توان برای استفاده‌های متعددی که در یک مرحله کاربر روی تعداد زیادی فایل می‌خواهد انجام

شود استفاده کرد. برای مثال:

*	به معنی تمام رشته‌ها
?	به معنی تمامی تک حرف‌هاست
[ABC]	با A, B و C تطابق می‌دهد.
[a-k]	با حروف a تا k تطابق می‌دهد.
[0-9a-z]	همه ارقام و همه حروف را تطابق می‌دهد.
[!x]	به معنی هر حرف به جز x است.

هریک از این علائم را می‌توان در دستوراتی که تا الان معرفی شده، استفاده کرد. به عنوان مثال:

```
cp [a-t]*.* Documents
cd Documents/
```

```
cd ..
ls D*
ls t?
```

- دستور **touch** برای تغییر دادن تاریخ و زمان (Timestamp) به کار می‌رود و اگر فایل موجود نباشد آن را ایجاد می‌کند:

انواع تاریخ و زمان:

۱- Access time: آخرین زمانی که فایل خوانده شده است.

۲- Modification time: آخرین زمانی که محتوای فایل تغییر کرده است.

۳- Change time: آخرین زمانی که ابر داده (meta data) فایل (مانند permissions) تغییر کرده است.

لیست برخی از سویچ‌های دستور touch	
-a	فقط زمان دستیابی (Access Time) تغییر کند
-c	اگر فایل موجود نبود فایلی جدید تولید نکند.
-d	رشته‌ای که پس از آن می‌آید را پارس کرده و به جای زمان فعلی استفاده می‌کند.
-m	فقط زمان تغییر (Modification Time) تغییر کند.
-r	از زمان‌های فایل به جای زمان فعلی استفاده کند.
-t	فایلی با زمان مشخص تولید کند.

مثال:

```
touch filename
touch -d "1am" filename
touch -d "07:50" filename
touch -d "yesterday 9pm" filename
touch -r referenceFile targetFile
```

- ✓ برای مشاهده زمان دستیابی (Access Time) فایل‌ها از `ls -l` استفاده کنید.

- دستور **find**: برای جستجوی به صورت سلسله مراتبی استفاده می‌شود.

لیست برخی از سویچ‌های دستور find	
-name	دنبال الگوی نوشته شده پس از این سویچ در نام فایل‌ها می‌گردد.
-iname	مشابه سویچ قبل است اما به کوچک یا بزرگ بودن حروف حساس نیست.
-type d	جستجوی دایرکتوری
-type f	جستجوی فایل

برای جستجو بر اساس حجم فایل استفاده می‌شود. + به معنای بزرگتر از N و - به معنای کوچکتر از N است. اگر عدد خالی بیاید به معنی بلوک است و با استفاده از c برای کاراکتر، G برای گیگابایت و ... می‌توان حجم را معلوم کرد.	-size +N/-N
برای جستجوی فایل یا دایرکتوری خالی استفاده می‌شود.	-empty
برای جستجوی فایل‌هایی که n*24 ساعت قبل خوانده شده است.	-atime n
برای جستجوی فایل‌هایی که n*24 ساعت قبل متا دیتای تغییر کرده است.	-ctime n
برای جستجوی فایل‌هایی که n*24 ساعت قبل محتوای آن تغییر کرده است.	-mtime n
برای جستجوی فایل‌هایی که n دقیقه قبل خوانده شده است.	-amin n
برای جستجوی فایل‌هایی که n دقیقه قبل متا دیتای آن تغییر کرده است.	-cmin n
برای جستجوی فایل‌هایی که n دقیقه قبل محتوای آن تغییر کرده است.	-mmin n

مثال:

```
find .
find directory/
find . -name "f*"
find . -iname "f*"
find . -type f -iname "t*"
find . -type d -iname "t*"
find . -size 65c
find -size +5k
find -empty
find -mtime -1
find -amin -45 -type d
```

✓ اگر قرار باشد روی فایل‌هایی که با استفاده از دستور بالا پیدا شده است، عملی انجام شود، راه مناسب استفاده از `-exec` است که پس از این از سویچ `{ }` یا `'{ }'` برای اشاره به فایل‌ها و پس از پایان دستور از `;` باید استفاده کرد.

مثال:

```
find -mmin -1 -exec cat '{}' \;
find /etc/rc* -exec echo Arg: {} \;
```

- از دستور **file** برای مشاهده نوع فایل به طوری که برای بیننده واضح باشد می‌توان استفاده کرد.
- دستور **locate** پیدا کردن فایل به کار می‌رود. چنانچه فایل را پیدا نکند به خط فرمان باز می‌گردد و در صورت پیدا نمودن فایل، مسیر کامل آن را نمایش خواهد داد.

`locate{pattern}`

مثال:

```
locate "*.txt"
```

بخش دوم:

مالکیت و مجوزهای فایل

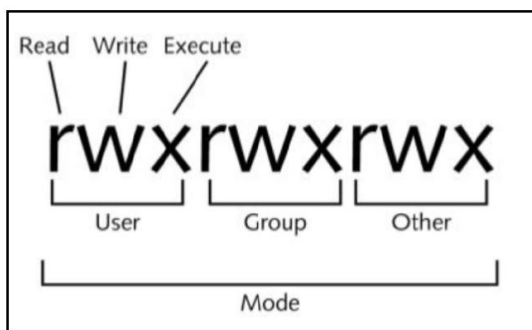
هر فایل شامل سه قسمت مجوز است:

الف) User permissions: مربوط به مالک فایل است.

ب) Group permissions: مربوط به گروه‌های تعریف شده در سیستم است.

ج) Other permission: مربوط به سایر افراد استفاده کننده از سیستم است.

هر دسته می‌توانند نوع مجوزهای خاص خود را داشته باشند.



- دستور **chown** برای تغییر مالکیت فایل و دایرکتوری استفاده می‌شود. فقط کاربر اصلی می‌تواند این کار را در لینوکس انجام دهد. مثال:

```
sudo chown root:root hello.sh
```

- دستور **chgrp** برای تغییر مالکیت گروهی فایل و دایرکتوری استفاده می‌شود. فقط کاربر اصلی می‌تواند این کار را در لینوکس انجام دهد. مثال:

```
chgrp adm hello.sh
```

- دستور **chmod** برای تغییر اجازه‌ها برای فایل و دایرکتوری استفاده می‌شود.

chmod symbolic-mode filename

دسته‌بندی‌هایی که با آن‌ها کار می‌شود:

الف) u=user ب) g=group ج) o=others د) a=all

عملیات:

الف) (=) set ب) (-) remove ج) (+) give

مجوزها:

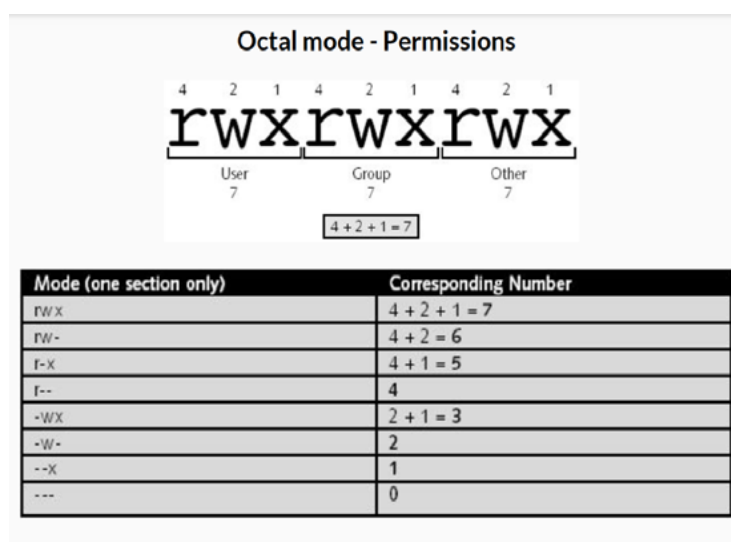
الف) (r) read ب) (w) write ج) (x) execute

- دستور **groups**: نام گروه‌هایی که کاربر جاری عضو آن‌ها است را نمایش می‌دهد. اولین گروه **primary group** است.

مثال:

```
chmod u+x filename
chmod ug-x filename
chmod o-r filename
chmod o=wrx filename
chmod o=r, g=r, u=wrx filename
```

در مدلی دیگر می‌توان مجوز فایل‌ها را تعیین کرد:



مثال:

```
chmod 755 filename
755: rwx r-x r-x
744: rwx r-- r--
777: rwx rwx rwx
666: rw- rw- rw-
```

تمرین:

۱- یک دایرکتوری داخل میزکاري (Desktop) بسازید و تمامی مجوزهای آن را به گونه‌ای تغییر دهید که فقط شما و اعضای گروه بتوانند بنویسند، بخوانند و در آن جستجو کنند.

۲- گروه‌هایی که شما در آن عضو هستید، را لیست کنید، سپس مالکیت فایل قبلی را به یکی دیگر از گروه‌های خود بدهید.

۳- یک فایل ایجاد کنید و زمان **Access time** آن را به ساعت ۹ شب تغییر دهید (در قالب یک دستور باید نوشته شود)، سپس زمان تغییر یافته را نشان دهید.

۴- درون کل دایرکتوری‌های موجود، فایل‌های خالی را پیدا کرده و پاک کنید. (این کار باید در یک خط دستور انجام شود).

۵- این دستور چه کاری انجام می‌دهد؟

```
chmod 4664 file.txt
```