

## جستجو با استفاده از دستور Find

از دستور find برای جستجوی فایل و دایرکتوری در لینوکس استفاده می شود.

شکل کلی دستور :

```
find [Path] [Expression] [Action]
```

این دستور دارای سوئیچ های متنوعی می باشد. در این نوشتار به بررسی این سوئیچ ها می پردازیم. در ابتدا با دستورات زیر، شرایط را برای محیط تست فراهم می کنیم:

```
# mkdir sample
```

```
# cd sample
```

```
# mkdir dir_test_1 dir_test_2 dir_test_3
```

```
# touch dir_test_1/file_test_1 dir_test_2/file_test_2 dir_test_3/file_test_3
```

معرفی سوئیچ های پرکاربرد دستور find :

استفاده از دستور find به تنهایی، سبب نمایش لیستی شامل آدرس محتویات مکان فعلی و زیر مجموعه آن می گردد.

```
# find
```

```
./dir_test_2
./dir_test_2/file_test_2
./dir_test_3
./dir_test_3/file_test_3
./dir_test_1
./dir_test_1/file_test_1
```

سوئیچ -name :

برای جستجو بر اساس نام از این سوئیچ استفاده می شود.

```
# find -name 'file_test_2'
```

```
/root/sample/dir_test_2/file_test_2
```

👉 با دستور بالا عمل جستجو در مکان فعلی و زیرمجموعه ی آن انجام می شود. جهت جستجو در مکانی خاص ، آدرس آن مکان به دستور اضافه می شود.

مثال : جستجوی فایل file\_test\_2 که در مسیر /root/sample قرار دارد :

```
# find /root/sample -name 'file_test_2'
```

```
/root/sample/dir_test_2/file_test_2
```

👉 دستور بالا در صورتی با خروجی همراه می شود که عبارت مورد نظر کاملا مطابق با نام فایل و یا دایرکتوری باشد. در نتیجه دستور پایین بدون خروجی می باشد :

```
# find /root/sample/ -name 'file_test'
```

برای سهولت در انجام جستجو می توان از کاراکترهای خاص (Wildcard) نیز بهره برد.

- \* = یک یا چند کاراکتر
- : = یک کاراکتر
- [] = کاراکترهای میان براکت

مثال : جستجوی فایل هایی که در مسیر /root/sample/ بوده و در نامشان عبارت test وجود دارد :

```
# find /root/sample/ -name '*test*'
```

```
/root/sample/dir_test_2  
/root/sample/dir_test_2/file_test_2  
/root/sample/dir_test_3  
/root/sample/dir_test_3/file_test_3  
/root/sample/dir_test_1  
/root/sample/dir_test_1/file_test_1
```

مثال : جستجوی فایل هایی که در مسیر /etc/ بوده و به عبارت .conf ختم می شوند(جستجوی فایل های تنظیمات) :

```
# find /etc/ -name '*.conf'
```

```
/etc/ld.so.conf.d/mysql-x86_64.conf  
/etc/ld.so.conf.d/kernelcap-2.6.9-89.0.0.1.EL.conf  
/etc/ld.so.conf.d/sblim-cmpi-syslog-x86_64.conf  
/etc/ld.so.conf.d/xorg-x11-x86_64.conf  
/etc/ld.so.conf.d/sblim-cmpi-network-x86_64.conf  
/etc/ld.so.conf.d/sblim-cmpi-nfsv3-x86_64.conf  
/etc/ld.so.conf.d/sblim-cmpi-nfsv4-x86_64.conf  
/etc/ld.so.conf.d/qt-i386.conf  
....
```

👉 در لینوکس وجود پسوند در نام فایل ها صرفاً برای شناخت بهتر کاربر می باشد.

مثال : جستجوی فایل هایی که نام آن ها با کاراکتر f شروع و به عدد 1 ختم می شود :

```
# find /root/sample -name 'f*1'
```

```
/root/sample/dir_test_1/file_test_1
```

مثال : جستجوی فایل های مسیر /etc که نام آن ها دارای 6 کاراکتر بوده و با m شروع و به f ختم می شود :

```
# find /etc -name 'm????f'
```

```
/etc/my.cnf
```

👉 در لینوکس هر فایل و یا دایرکتوری که نام آن با نقطه آغاز شود، بعنوان فایل و یا دایرکتوری مخفی می باشد.

مثال : جستجوی فایل و دایرکتوری مخفی سیستم به صورت زیر قابل انجام می باشد :

```
# find / -name '.*'
```

## جستجو در چند مسیر :

برای انجام جستجو در دو یا چند مسیر، کافیست مسیرها پشت سر هم در دستور درج شوند.

مثال : جستجو فایلی با نام bashrc در مسیرهای /boot , /etc , /root بصورت زیر است :

```
# find /root /etc /boot -name 'bashrc'
```

```
/etc/bashrc
```

## سوئیچ -ls :

در صورت استفاده از این سوئیچ جزئیات بیشتری از فایل هایی که در نتیجه ی جستجو بدست آمده اند نمایش می یابد.

```
# find /root /etc /boot -name 'bashrc' -ls
```

```
5111829 4-rw-r--r-- 1 root root 2681 Oct 2 2013 /etc/bashrc
```

## سوئیچ -or یا -o :

این سوئیچ نقش عملگر OR را ایفا می کند. بدین معنی که خروجی هر قسمت، نمایش داده می شود. (در خروجی مقدار تکراری وجود ندارد)

مثال : جستجوی فایل هایی که نامشان برابر با عبارات `sort` , `date` , `fdisk` بوده و در مسیرهای `/bin` , `/sbin` قرار دارند:

```
# find /bin /sbin -name 'sort' -o -name 'date' -o -name 'fdisk'
```

```
/bin/date  
/bin/sort  
/sbin/fdisk
```

## سوئیچ -and یا -a :

این سوئیچ نقش عملگر AND را ایفا می کند. بدین معنی که خروجی باید تمامی فیلترهای مورد نظر را دارا باشد.

مثال : جستجوی فایل هایی که نام آن شامل عبارات `'dir'` , `'1'` باشد :

```
# find /root/sample -name '*1*' -a -name '*dir*'
```

```
/root/sample/dir_test_1
```

👉 دستور زیر نیز عملکردی مشابه با دستور بالا دارد.

```
# find /root/sample -name '*1*' '*dir*'
```

## سوئیچ -iname :

برای اینکه لینوکس بین حروف کوچک و بزرگ حساس نباشد از این سوئیچ استفاده می شود.

```
# find /root/sample/ -iname '*FILE*'
```

```
/root/sample/dir_test_2/file_test_2  
/root/sample/dir_test_3/file_test_3  
/root/sample/dir_test_1/file_test_1
```

حروف کوچک و بزرگ در لینوکس متمایز می باشند. 🖐

## سوئیچ -type :

در لینوکس همه چیز، حتی یک سخت افزار به شکل فایل دیده می شود. با این سوئیچ عمل جستجو بدنبال نوع خاصی از فایل محدود می شود. در صورت عدم استفاده از این سوئیچ تفاوتی در نوع فایل های خروجی ایجاد نمی شود. این سوئیچ کاربردی همراه با مقادیر زیر استفاده می شود.

### block (buffered) special ← b 🖐

معرف یک ابزار ذخیره سازی فیزیکی و یا منطقی می باشد. مانند Hard Disk. زمانی که قرار به خواندن دسته ایی داده ها باشد از آن استفاده می شود.

#### # file /dev/sda

/dev/sda: block special

### character (unbuffered) special ← c 🖐

یک فایل سخت افزاری است که زمان خواندن داده بصورت کاراکتر به کاراکتر استفاده می شود. مانند صفحه کلید

#### # file /dev/tty

/dev/tty: character special

### directory ← d 🖐

دایرکتوری در لینوکس به عنوان یک نوع فایل دیده می شود که دسته بندی سایر فایل ها را انجام می دهد.

### named pipe (FIFO) ← p 🖐

فایل ویژه ایی که می توان بصورت همزمان برای خواندن یا نوشتن بین چند پردازش به اشتراک گذاشته شود.

### regular file ← f 🖐

فایل معمولی می باشد. مانند فایل باینری یا عکس

### symbolic link ← l 🖐

منظور همان Soft Link بوده که به مانند Shortcut می باشد.

### socket ← s 🖐

از این نوع فایل برای تبادل داده بین دو پردازش استفاده می شود.

مثال : جستجوی دایرکتوری های مسیر /root/sample/ که در نامشان عبارت test وجود داشته باشد :

```
# find /root/sample/ -name '*test*' -type d
```

```
/root/sample/dir_test_2  
/root/sample/dir_test_3  
/root/sample/dir_test_1
```

مثال : جستجوی فایل های مسیر /root/sample/ که در نامشان عبارت test وجود داشته باشد :

```
# find /root/sample/ -name '*test*' -type f
```

```
/root/sample/dir_test_2/file_test_2  
/root/sample/dir_test_3/file_test_3  
/root/sample/dir_test_1/file_test_1
```

سوئیچ -not یا علامت ! :


همواره با استفاده از این سوئیچ می توان عملکرد سوئیچ های دیگر را برعکس نمود.

```
# find /root/sample/ -name '*2*' -type f
```

```
/root/sample/dir_test_2/file_test_2
```

```
# find /root/sample/ -not -name '*2*' -type f
```

```
/root/sample/dir_test_3/file_test_3  
/root/sample/dir_test_1/file_test_1
```

همین طور می توان بجای -not از علامت ! نیز استفاده نمود. 

```
# find /root/sample/ ! -name '*2*' -type f
```

سوئیچ -empty :

با استفاده از این سوئیچ می توان فایل ها و دایرکتوری های خالی را پیدا کرد.

```
# find /root/sample/ -type f -empty
```

```
/root/sample/dir_test_1/file_test_1  
/root/sample/dir_test_3/file_test_3
```

## سوئیچ -path :

برای جستجو بر اساس مسیر دستیابی به فایل ها از این سوئیچ استفاده می شود.

```
# find /root/sample -type f -path '*2*'
```

```
/root/sample/dir_test_2/file_test_2
```

👉 با دستور بالا تمامی فایل هایی که آدرس دستیابی شان عبارت مورد نظر باشد نمایش می یابد.

## سوئیچ -user :

با کمک این سوئیچ می توان عمل جستجو را بر اساس مالک فایل انجام داد.

```
# find /home -type f -user oracle
```

```
/home/oracle/.bash_profile
```

```
/home/oracle/.bashrc
```

```
/home/oracle/.kshrc
```

```
....
```

👉 با دستور بالا تمامی فایل هایی که در مسیر /home قرار دارند و کاربر مالک آن ها oracle باشد نمایش می یابد.

## سوئیچ -uid :

با این سوئیچ می توان جستجو را بر اساس User ID انجام داد.

```
# find /home -type f -uid 1010
```

👉 با دستور بالا تمامی فایل هایی که در مسیر /home قرار دارند و ID کاربر مالک آن ها 1010 باشد نمایش می یابد.

## سوئیچ -group :

با کمک این سوئیچ می توان عمل جستجو را بر اساس گروه فایل انجام داد.

```
# find /home -type f -group school
```

👉 با دستور بالا تمامی فایل هایی که در مسیر /home قرار دارند و گروه آن ها school باشد نمایش می یابد.

## سوئیچ -gid :

با این سوئیچ می توان جستجو را بر اساس Group ID انجام داد.

```
# find /home -type f -gid 1000
```

👉 با دستور بالا تمامی فایل هایی که در مسیر /home قرار دارند و ID گروه آن ها برابر با 1000 باشد نمایش می یابد.

## سوئیچ -maxdepth , -mindepth :

از این سوئیچ برای تعیین سطح ابتدا و انتهای مسیر جستجو استفاده می شود.

- تعیین سطح ابتدا ← -mindepth
- تعیین سطح انتها ← -maxdepth

👉 در حالت عادی سطح صفر همان انتهای مسیر مشخص شده در دستور می باشد. بنابراین دستورات زیر خروجی یکسان دارند.

```
# find /root/sample = # find /root/sample/ -mindepth 0
```

```
/root/sample/  
/root/sample/empty_file  
/root/sample/dir_test_1  
/root/sample/dir_test_1/file_test_1  
/root/sample/dir_test_3
```



```
/root/sample/dir_test_3/file_test_3  
/root/sample/dir_test_2  
/root/sample/dir_test_2/file_test_2
```

👉 در دستور زیر فایل و دایرکتوری هایی در خروجی قرار می گیرند که سطح ابتدای آنها نسبت به `/root/sample` یک سطح بالاتر باشد. مقداری برای سطح انتها مشخص نشده است.

### # find /root/sample -mindepth 1

```
/root/sample/empty_file  
/root/sample/dir_test_1  
/root/sample/dir_test_1/file_test_1  
/root/sample/dir_test_3  
/root/sample/dir_test_3/file_test_3  
/root/sample/dir_test_2  
/root/sample/dir_test_2/file_test_2
```

👉 در دستور زیر فایل و دایرکتوری هایی در خروجی قرار می گیرند که سطح انتهای آنها نسبت به `/root/sample` یک سطح بالاتر باشد. مقداری برای سطح ابتدا مشخص نشده است.

### # find /root/sample -maxdepth 1

```
/root/sample/  
/root/sample/empty_file  
/root/sample/dir_test_1  
/root/sample/dir_test_3  
/root/sample/dir_test_2
```

👉 در دستور زیر فایل و دایرکتوری هایی در خروجی قرار می گیرند که سطح ابتدا و انتهای آنها نسبت به `/root/sample` یک سطح بالاتر باشد.

### # find /root/sample -mindepth 1 -maxdepth 1

```
/root/sample/empty_file  
/root/sample/dir_test_1  
/root/sample/dir_test_3  
/root/sample/dir_test_2
```

👉 در دستور زیر عمل جستجو میان فایل و دایرکتوری هایی که در نامشان عبارت `lvm` وجود دارد و در محدوده ی تعیین شده قرار دارند انجام می شود.

### # find / -name '\*lvm\*' -mindepth 2 -maxdepth 3

```
/etc/lvm
```

```
/sbin/lvmconf
/sbin/lvm
/var/run/lvm
/var/lock/lvm
/usr/sbin/system-config-lvm
/usr/bin/system-config-lvm
/etc/lvm/lvm.conf
/etc/pam.d/system-config-lvm
/lib64/device-mapper/libdevmapper-event-lvm2raid.so
/lib64/device-mapper/libdevmapper-event-lvm2mirror.so
....
```

## سوئیچ -size :

با این گزینه می توان جستجو را بر اساس اندازه ی فایل ها با مقیاس های زیر انجام داد.

- c = بایت
- k = کیلوبایت
- M = مگابایت
- G = گیگابایت

مثال : جستجوی فایل هایی 200 مگابایتی که در مسیر /root/sample قرار دارند :

```
# find /root/sample -type f -size 200M
```

مثال : جستجوی فایل های کوچکتر از 4 گیگابایت که در مسیر /root/sample قرار دارند :

```
# find /root/sample -type f -size -4G
```

مثال : جستجوی فایل های بزرگتر از 500 کیلوبایت که در مسیر /root/sample قرار دارند :

```
# find /root/sample -type f -size +500k
```

مثال : جستجوی فایل های بزرگتر از 100 مگابایت و کوچکتر از 2 گیگابایت که در مسیر /root/sample قرار دارند :

```
# find /root/sample -type f -size +100M -size -2G
```

## سوئیچ -delete :

جهت حذف فایل هایی که در خروجی دستور جستجو هستند از این سوئیچ استفاده می شود.

```
# find /root/sample/ -size 0k
```

```
/root/sample/dir_test_1/file_test_1
```

```
/root/sample/dir_test_3/file_test_3
```

```
# find /root/sample/ -size 0k -delete
```

## سوئیچ -fls :

برای ذخیره ی نتیجه دستور جستجو در یک فایل از این سوئیچ استفاده می شود.

```
# find /root/sample/ -name '*2*' -type f -fls /root/search_list
```

👉 در صورت استفاده از این سوئیچ بعد از زدن Enter بلافاصله Prompt نمایش می یابد و خروجی در فایل مشخص شده قرار می گیرد.

👉 برای اینکه خروجی در یک فایل ذخیره شود از کاراکتر > نیز می توان استفاده کرد.

```
# find /root/sample/ -name '*2*' -type f > /root/search_list
```

## سوئیچ -exec :

از این سوئیچ برای اجرای دستور بر روی فایل هایی که در نتیجه جستجو قرار دارند، استفاده می شود.

مثال : برای جستجو و حذف فایل و دایرکتوری های خالی از دستور زیر استفاده می شود :

```
# find /root/sample/ -empty -exec rm -rf {} \;
```

مثال : با استفاده از دستور زیر می توان فایل هایی که دارای سطح دسترسی 777 هستند را پیدا کرده و سطح دسترسی را به 755 تبدیل نمود :

```
# find /root/sample/ -perm 777 -exec chmod 755 {} \;
```

مثال : نمایش 5 فایل که در مسیر /boot/ بوده و دارای کمترین اندازه هستند :

```
# find /boot -type f -exec du -sh {} + | sort -h | head -5
```

```
4.0K /boot/grub/device.map
```

```
4.0K /boot/grub/grub.conf
4.0K /boot/grub/grub.conf.orabackup
4.0K /boot/grub/splash.xpm.gz
4.0K /boot/grub/stage1
```

با دستور بالا فایل ها توسط دستورات `du` و `sort` بر اساس اندازه مرتب می شوند و دستور `head` سبب نمایش 5 فایل ابتدایی می گردد.

مثال : نمایش 5 فایل که در مسیر `/boot` بوده و دارای بیشترین اندازه هستند :

```
# find /boot -type f -exec du -sh {} + |sort -h -r | head -5
```

```
22M /boot/initramfs-3.8.13-16.2.1.el6uek.x86_64.img
22M /boot/initramfs-3.8.13-16.2.1.el6uek.x86_64.debug.img
17M /boot/initramfs-2.6.32-431.el6.x86_64.img
4.7M /boot/vmlinuz-3.8.13-16.2.1.el6uek.x86_64.debug
4.4M /boot/vmlinuz-3.8.13-16.2.1.el6uek.x86_64
```

## سوئیچ `-executable` :

توسط این سوئیچ می توان فایل های اجرایی را جستجو نمود.

```
# find /root/sample/ -type f -executable
```

با این سوئیچ کفایت که خاصیت اجرا فقط برای هر کدام از کاربر مالک، گروه و یا سایر کاربران فعال باشد. (یعنی فایلی با سطح دسترسی `X-----` نیز در خروجی نمایش می یابد).

## عمل جستجو بر اساس زمان :

قبل از آشنایی با دستورات این قسمت به بررسی فاکتور زمان برای یک فایل می پردازیم. فاکتور زمان برای هر فایل دارای سه حالت می باشد.

### 1. Access Time :

این حالت با اعمالی که در زیر مشخص شده است Update می شود.

- ایجاد فایل = ✓
- خواندن فایل = ✓

- ✓ = تغییر در محتوای فایل
- ✓ = کپی فایل
- ✕ = تغییر نام فایل
- ✕ = تغییر مکان فایل
- ✕ = تغییر سطح دسترسی

👉 اگر به فایلی که محتوای آن ثابت است دسترسی متوالی ایجاد نمود، زمان Access Time فقط در مرتبه ی اول Update می شود.

👉 در صورتیکه از دستور echo برای تغییر محتوای یک فایل استفاده شود در Access Time تغییری ایجاد نمی شود.

👉 در فرآیند کپی، زمان Access Time برای هر دو فایل مبدا و مقصد Update می شود.

👉 برای دیدن زمان Access Time می توان از دو دستور زیر استفاده نمود.

#### # ls -lu

```
total 16
drwxr-xr-x. 2 root root 4096 Nov 11 13:36 dir_test_1
drwxr-xr-x. 2 root root 4096 Nov 11 13:34 dir_test_2
drwxr-xr-x. 2 root root 4096 Nov 11 13:34 dir_test_3
-rwxrwxrwx. 1 root root 0 Nov 12 10:14 empty_file
```

#### # ls -l --time=atime

```
total 16
drwxr-xr-x. 2 root root 4096 Nov 11 13:36 dir_test_1
drwxr-xr-x. 2 root root 4096 Nov 11 13:34 dir_test_2
drwxr-xr-x. 2 root root 4096 Nov 11 13:34 dir_test_3
-rwxrwxrwx. 1 root root 0 Nov 12 10:14 empty_file
```

## 2. Modification Time :

این حالت با انجام اعمال زیر بر روی یک فایل Update می شود.

- ✓ = ایجاد فایل
- ✓ = تغییر در محتوای فایل
- ✓ = کپی فایل
- ✕ = خواندن فایل
- ✕ = تغییر نام فایل
- ✕ = تغییر مکان فایل

- تغییر سطح دسترسی = \*

👉 در فرآیند کپی، زمان Modification Time فقط برای فایل مقصد Update می شود.

👉 برای دیدن زمان Modification Time می توان از دستور زیر استفاده نمود.

### # ls -l

```
total 16
drwxr-xr-x. 2 root root 4096 Nov 11 13:36 dir_test_1
drwxr-xr-x. 2 root root 4096 Nov 4 19:42 dir_test_2
drwxr-xr-x. 2 root root 4096 Nov 4 13:40 dir_test_3
-rwxrwxrwx. 1 root root 0 Nov 12 16:47 empty_file
```

### 3. Change Time :

این حالت با انجام اعمال زیر بر روی یک فایل Update می شود.

- ایجاد فایل = ✓
- تغییر در محتوای فایل = ✓
- کپی فایل = ✓
- تغییر نام فایل = ✓
- تغییر مکان فایل = ✓
- تغییر سطح دسترسی = ✓
- خواندن فایل = \*

👉 در فرآیند کپی زمان Change Time فقط برای فایل مقصد Update می شود.

👉 برای دیدن زمان Change Time می توان از دو دستور زیر استفاده نمود.

### # ls -lc

```
total 16
drwxr-xr-x. 2 root root 4096 Nov 11 13:36 dir_test_1
drwxr-xr-x. 2 root root 4096 Nov 4 19:42 dir_test_2
drwxr-xr-x. 2 root root 4096 Nov 4 13:40 dir_test_3
-rwxrwxrwx. 1 root root 0 Nov 12 16:47 empty_file
```

### # ls -l --time=ctime

```
total 16
drwxr-xr-x. 2 root root 4096 Nov 11 13:36 dir_test_1
drwxr-xr-x. 2 root root 4096 Nov 4 19:42 dir_test_2
drwxr-xr-x. 2 root root 4096 Nov 4 13:40 dir_test_3
-rwxrwxrwx. 1 root root 0 Nov 12 16:47 empty_file
```

پس از بررسی سه حالت زمان و چگونگی تغییر آنها، اکنون به سراغ استفاده از این حالت در دستور find می پردازیم.

### سوئیچ -amin :

با این سوئیچ می توان فایل ها را بر اساس Access Time با مقیاس دقیقه جستجو نمود.

**# find [path] -amin n**

روش محاسبه ی مدت زمان n :

- n = در دقیقه n م
- -n = کمتر از n دقیقه
- +n = بیشتر از n دقیقه

مثال : جستجوی فایل هایی که Access Time آنها در 30 دقیقه ی اخیر بوده است :

**# find /root/sample -type f -amin -30**

مثال : اگر الان ساعت 9 صبح باشد برای جستجوی فایل های مسیر /root/sample که Access Time آنها در بازه ی ساعت 8 الی 30:8 واقع شده باشد از چه دستوری استفاده می شود :

**# find /root/sample -type f -amin +30 -amin -60**

### سوئیچ -atime :

با این سوئیچ می توان فایل ها را بر اساس Access Time در مقیاس روز جستجو نمود.

مثال : جستجوی فایل هایی که Access Time آنها مربوط به 7 روز اخیر بوده است :

**# find /root/sample -type f -atime -7**

### سوئیچ -anewer :

از این سوئیچ برای جستجوی فایل هایی که Access Time آنها به روزتر از یک فایل خاص باشد، استفاده می شود.

مثال : جستجو در مسیر /root/sample جهت یافتن فایل هایی که Access Time آنها به روزتر از فایل /tmp/file\_2 باشد:

```
# find /root/sample -type f -anewer /tmp/file_2
```

سوئیچ -mmin :

با این سوئیچ می توان فایل ها را بر اساس Modification Time با مقیاس دقیقه جستجو نمود.

مثال : جستجوی فایل هایی که Modification Time آنها بیش از 30 دقیقه ی اخیر بوده است :

```
# find /root/sample -type f -mmin +30
```

سوئیچ -mtime :

با این سوئیچ می توان فایل ها را بر اساس Modification Time با مقیاس روز جستجو نمود.

مثال : با استفاده از دستور زیر فایل هایی که Modification Time آنها مربوط به 4 روز اخیر بوده است نمایش می یابد.

```
# find /root/sample -mtime -4 -type f
```

سوئیچ -newer :

این سوئیچ برای جستجوی فایل هایی که Modification Time آنها به روزتر از یک فایل خاص باشد، استفاده می شود.

مثال : جستجو در مسیر /root/sample جهت یافتن فایل هایی که Modification Time آنها به روزتر از فایل /tmp/file\_2

باشد :

```
# find /root/sample -type f -newer /tmp/file_2
```

سوئیچ -cmin :

با این سوئیچ می توان فایل ها را بر اساس Change Time با مقیاس دقیقه جستجو نمود.



مثال : جستجوی فایل هایی که Change Time آنها در 40 دقیقه ی اخیر بوده است :

```
# find /root/sample -type f -cmin -40
```

سوئیچ -ctime :

با این سوئیچ می توان فایل ها را بر اساس Change Time با مقیاس روز جستجو نمود.

مثال : جستجوی فایل هایی که Change Time آنها مربوط به 8 روز اخیر بوده است :

```
# find /root/sample -type f -ctime -8
```

سوئیچ -cnewer :

از این سوئیچ برای جستجوی فایل هایی که Change Time آنها به روزتر از یک فایل خاص باشد، استفاده می شود.

مثال : جستجو در مسیر /root/sample جهت یافتن فایل هایی که Change Time آنها به روزتر از فایل /tmp/file\_2

باشد :

```
# find /root/sample -type f -cnewer /tmp/file_2
```

سوئیچ -newerXY :

برای اینکه عمل جستجو بر اساس تاریخ دلخواه صورت گیرد، می توان از این سوئیچ استفاده نمود.

👉 بجای XY مقادیر زیر قرار می گیرد.

- Access Time = at
- Modification Time = mt
- Change Time = ct

مثال : جستجو در مسیر /root/sample جهت یافتن فایل هایی که Modification Time آنها بعد از تاریخ 2018/04/25 باشد :

```
# find /root/sample -type f -newermt 2018-04-25
```

مثال : جستجو در مسیر /root/sample جهت یافتن فایل هایی که Modification Time آنها قبل از تاریخ 2018/04/25 باشد :

```
# find /root/sample -type f -newermt 2018-04-25
```

مثال : جستجو در مسیر /root/sample جهت یافتن فایل هایی که Modification Time آنها از تاریخ 2018/02/20 تا تاریخ 2018/04/10 باشد :

```
# find /root/sample -type f -newermt 2018-02-20 ! -newermt 2018-04-10
```

## سوئیچ -perm :

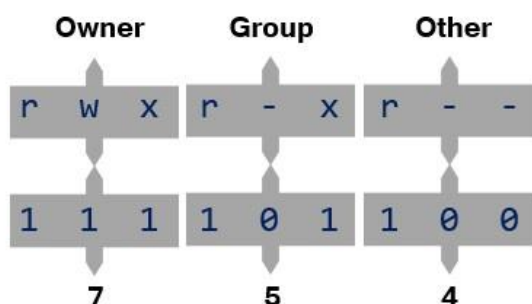
از این سوئیچ برای جستجو بر اساس Permission استفاده می شود.

👉 Permission یا سطح دسترسی :

مجوزهای دسترسی که بترتیب مشخص کننده سطح دسترسی کاربر مالک، گروه و سایر کاربران است در ۹ کاراکتر نمایش داده می شود. در روش محاسبه دودویی هر بیت می تواند یک یا صفر باشد. عدد ۱ به معنی داشتن و ۰ به معنی نداشتن آن مجوز می باشد.

- `rwxr-x--x` → Owner , Group , Other

- **u = Owner , g = Group , o = Other**
- **rwX → r = Read , w = Write , x = Execute**
- **rwX = 111 = 7 , r-x = 101 = 5 , r-- = 100 = 4 , --x = 001 = 1**
- **rwXr-xr-- = 111101100 = 754 → Owner = Full , Group = Read & Execute , Other = Read**



👉 در جستجو بر اساس سطح دسترسی می توان به دلخواه از اعداد یا حروف استفاده نمود.

👉 بعد از سوئیچ -perm مقدار مورد نظر جهت جستجو قرار می گیرد که برای معرفی آن از سه روش استفاده می شود.

### 1. بدون استفاده از پیشوند :

در این روش فایل هایی که سطح دسترسی شان دقیقاً مطابق با آن عدد و یا حروف باشد نمایش می یابد.

مثال : جستجوی فایل هایی مسیر /root/sample با سطح دسترسی 644 و یا rw-r--r-- :

```
# find /root/sample -type f -perm 644
```

Or

```
# find /root/sample -type f -perm u=rw,g=r,o=r
```

مثال : جستجوی فایل های مسیر /root/sample که سطح دسترسی شان 644 نیست :

```
# find /root/sample -type f ! -perm 644
```

مثال : جستجوی کل سیستم برای یافتن فایل هایی که دارای دسترسی کامل هستند :

```
# find / -type f -perm 777
```

Or

```
# find / -type -perm ugo=rwx
```

Or

```
# find / -type -perm a=rwx
```

مثال : جستجوی فایل های مسیر /root/sample که سطح دسترسی شان 751 یا 700 باشد :

```
# find /root/sample/ -type f -perm 751 && find /root/sample/ -type f -perm 700;
```

## 2. با استفاده از پیشوند - :

سطح دسترسی بصورت جداگانه برای هر قسمت (مالک، گروه و سایر کاربران) بررسی شده و آن فایل هایی که سطح دسترسی شان مساوی و بیشتر از عدد هر قسمت باشد نمایش می یابد.

مثال : جستجوی فایل هایی که سطح دسترسی شان حداقل برابر با 422 باشد :

```
# find / -type f -perm -422
```

👉 در خروجی فایل هایی که دارای سطح دسترسی زیر باشند نمایش می یابد.

- Owner => 4 & Group => 2 & Other => 2

👉 بطور مثال در خروجی فایلی با سطح دسترسی 717 وجود نخواهد داشت.

مثال : جستجو فایل هایی که گروه حداقل دارای سطح دسترسی نوشتن و سطح دسترسی سایر کاربران بی اهمیت باشد :

```
# find / -type f -perm -g=w
```

Or

```
# find / -type f -perm -020
```

مثال : جستجوی فایل هایی که کاربر مالک دارای سطح دسترسی کامل می باشد :

```
# find /root/sample/ -type f -perm -u=rwx
```

Or

```
# find /root/sample/ -type f -perm -700
```

👉 در این حالت فقط سطح دسترسی کاربر مالک بررسی می شود و سطح دسترسی سایر قسمت ها (گروه و سایر کاربران) تاثیری در نتیجه جستجو ندارد.

مثال : جستجوی فایل هایی که کاربر مالک حداقل دارای سطح دسترسی خواندن و نوشتن و گروه حداقل دارای دسترسی خواندن باشد :

```
# find /root/sample/ -type f -perm -u=rw,g=r
```

3. با استفاده پیشوند / :

وجود یک تشابه در هر کدام از قسمت ها (مالک، گروه و سایر کاربران) سبب قرار گرفتن آن فایل در خروجی می گردد.

```
# find / -perm /777 -type f
```

👉 با دستور بالا اگر فایلی دارای سطح دسترسی 001 باشد باز هم در خروجی قرار می گیرد.

سوئیچ -version :

سبب نمایش نسخه ی دستور find می شود:

```
#find -version
```

find (GNU findutils) 4.4.2

Copyright (C) 2007 Free Software Foundation, Inc.

License GPLv3+: GNU GPL version 3 or later <<http://gnu.org/licenses/gpl.html><

This is free software: you are free to change and redistribute it.

There is NO WARRANTY, to the extent permitted by law.

Written by Eric B. Decker, James Youngman, and Kevin Dalley.

Built using GNU glibc version e5573b1bad88bfabcda181b9e0125fb0c52b7d3b

Features enabled: D\_TYPE O\_NOFOLLOW(enabled) LEAF\_OPTIMISATION SELINUX FTS() CBO(level=0)