

# Linux Operating System and Programming

## TOPIC # 11.7

### Exercising 'find' utility in Linux

#### Find – Locating files

How find operates:

find path\_list selection\_criteria action

First, it recursively examines all files in the directories specified in 'path\_list'

It then matches each file for one or more 'selection\_criteria'

Finally, it takes some 'action' on those selected files.

To begin, let's look at the basic structure of the find command:

```
find start_directory test options criteria_to_match  
action_to_perform_on_results
```

#### Goal # 1 : Finding a file with it's full or partial name (Note that extension is part of name in linux)

In the following command, find will start looking in the current directory, denoted by the ".", for any file with the "java" extension in its name:

```
find . -name "*.java"
```

```
find . -name \*.java
```

Both above commands will do the same thing. In either case, you need to escape the wildcard character to be sure it passes to the find command and is not interpreted by the shell. So, put your search string in quotes, or precede it with a backslash.

You may use any absolute path to find within instead just '.' (current directory).

```
[jpandya@JMP ~]$ mkdir learnfind  
[jpandya@JMP ~]$ cd learnfind  
[jpandya@JMP learnfind]$ pwd  
/home/jpandya/learnfind
```

There are no files or directories in present working directory. 'find' does not display anything if no match found:

```
[jpandya@JMP learnfind]$ tree ../learnfind  
../learnfind  
0 directories, 0 files
```

```
[jpandya@JMP learnfind]$ ls -l testfind.txt
ls: testfind.txt: No such file or directory
[jpandya@JMP learnfind]$ find . -name testfind.txt
```

Creating a file and finding with name:

```
[jpandya@JMP learnfind]$ touch testfind.txt
```

//The `-name` option makes the search case sensitive.

```
[jpandya@JMP learnfind]$ find . -name testfind.txt
./testfind.txt
```

To show absolute path from present working directory in find result, use `pwd` with backtick or `$PWD` environment variable:

```
[jpandya@JMP learnfind]$ find `pwd` -name testfind.txt
/home/jpandya/learnfind/testfind.txt
```

```
[jpandya@JMP learnfind]$ find $PWD -name testfind.txt
/home/jpandya/learnfind/testfind.txt
```

You may use `-iname` for case insensitive search:

```
[jpandya@JMP learnfind]$ find $PWD -iname TeStfInd.txt
/home/jpandya/learnfind/testfind.txt
```

## Goal # 2: Find file owned by the particular user

```
find /home -user joe
```

Below creates two files in learnfind directory owned by user1 and while searching it finds those files only. See that the logged in user name is jpandya.

```
[jpandya@JMP learnfind]$ pwd
```

```
/home/jpandya/learnfind
```

```
[jpandya@JMP learnfind]$ sudo su -
```

```
[sudo] password for jpandya:
```

```
[root@JMP ~]# mkdir -p /home/jpandya/learnfind/ddu/btech/sem1/losp/shellscripts
```

```
[root@JMP ~]# mkdir -p
/home/jpandya/learnfind/ddu/btech/sem1/losp/shellscripts/ce28
```

```
[root@JMP ~]# touch
/home/jpandya/learnfind/ddu/btech/sem1/losp/shellscripts/ce28/test.sh
```

```
[root@JMP ~]# chown user1:user1
```

```
/home/jpandya/learnfind/ddu/btech/sem1/losp/shellscripts/ce28/test.sh

[root@JMP ~]# touch
/home/jpandya/learnfind/ddu/btech/sem1/losp/shellscripts/user1_info.txt

[root@JMP ~]# chown user1:user1
/home/jpandya/learnfind/ddu/btech/sem1/losp/shellscripts/user1_info.txt

[root@JMP ~]# exit

logout
```

```
[jpandya@JMP learnfind]$ tree ../learnfind
```

```
../learnfind
|-- ddu
|   |-- btech
|       |-- sem1
|           |-- losp
|               |-- shellscripts
|                   |-- ce28
|                       |-- test.sh
|                           |-- user1_info.txt
|-- testfind.txt
```

6 directories, 3 files

```
[jpandya@JMP learnfind]$ find . -user user1
```

```
./ddu/btech/sem1/losp/shellscripts/user1_info.txt
```

```
./ddu/btech/sem1/losp/shellscripts/ce28/test.sh
```

```
[jpandya@JMP learnfind]$ ls -l `find . -user user1`
```

```
-rw-r--r-- 1 user1 user1 0 Sep 18 10:31
./ddu/btech/sem1/losp/shellscripts/ce28/test.sh
```

```
-rw-r--r-- 1 user1 user1 0 Sep 18 10:31
./ddu/btech/sem1/losp/shellscripts/user1_info.txt
```

```
[jpandya@JMP learnfind]$
```

See that testfind.txt was owned by logged in user jpandya.  
You may also use -group option in find, if wish to locate files by their groups.

### **Goal # 3 : Find based on modified time period**

Find every file under the directory /var/spool that was modified more than 60 days ago.

```
find /var/spool -mtime +60
```

See that the timestamp is 2013-09-18 for the file. We will update the timestamp to a back date and try locating that file using find utility.

```
[jpandya@JMP learnfind]$ date
Wed Sep 18 10:49:37 IST 2013
[jpandya@JMP learnfind]$ stat testfind.txt
  File: `testfind.txt'
  Size: 0                Blocks: 8                IO Block: 4096   regular empty file
Device: fd00h/64768d    Inode: 22184606       Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 501/ jpandya)   Gid: ( 502/ jpandya)
Access: 2013-09-18 09:59:19.000000000 +0530
Modify: 2013-09-18 09:59:19.000000000 +0530
Change: 2013-09-18 09:59:19.000000000 +0530
[jpandya@JMP learnfind]$ touch -m -t 201302270220 testfind.txt
[jpandya@JMP learnfind]$ stat testfind.txt
  File: `testfind.txt'
  Size: 0                Blocks: 8                IO Block: 4096   regular empty file
Device: fd00h/64768d    Inode: 22184606       Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 501/ jpandya)   Gid: ( 502/ jpandya)
Access: 2013-09-18 09:59:19.000000000 +0530
Modify: 2013-02-27 02:20:00.000000000 +0530
Change: 2013-09-18 10:48:12.000000000 +0530
[jpandya@JMP learnfind]$ find . -mtime +60
./testfind.txt
[jpandya@JMP learnfind]$
```

You may also use option -atime for locating file based on access time of files or directories.

### **Goal # 4 : Find based on type of file (like regular file, directory, device, pipe, softlink, socket, etc.)**

Creating a soft link as below:

```
[jpandya@JMP learnfind]$ ln -s ./ddu/btech/sem1/losp/shellscripts myscripts
[jpandya@JMP learnfind]$ tree ../learnfind
../learnfind
|-- ddu
|   |-- btech
|       |-- sem1
```

```
|      |-- losp
|      |-- shellscripts
|          |-- ce28
|          |-- test.sh
|          |-- user1_info.txt
|-- myscripts -> ./ddu/btech/sem1/losp/shellscripts
|-- testfind.txt
```

7 directories, 3 files

See that 'l' in the very first bit indicates that its a type of soft link:

```
[jpandya@JMP learnfind]$ ls -l myscripts
```

```
lrwxrwxrwx 1 jpandya jpandya 34 Sep 18 10:57 myscripts -> ./ddu/btech/sem1/losp/shellscripts
```

Searching for all file types of soft link in current directory:

```
[jpandya@JMP learnfind]$ find . -type l
```

```
./myscripts
```

```
[jpandya@JMP learnfind]$
```

### Goal # 5 : Search for files based on their permissions

Using option -perm and specifying permissions, locate files having similar match.

```
find . -perm /220
```

```
find . -perm /u+w,g+w
```

```
find . -perm /u=w,g=w
```

Creating a shell script file and giving 774 permission and then locate using find -perm:

```
[jpandya@JMP learnfind]$ touch ddu/btech/sem1/losp/shellscripts/demo.sh
```

```
[jpandya@JMP learnfind]$ chmod 774 ddu/btech/sem1/losp/shellscripts/demo.sh
```

```
[jpandya@JMP learnfind]$ find . -perm 774
```

```
./ddu/btech/sem1/losp/shellscripts/demo.sh
```

```
[jpandya@JMP learnfind]$
```

### Goal # 6 : Search file using inode number

In some cases where filenames or paths having special character then you may like to refer using inode-number.

```
[jpandya@JMP learnfind]$ ls -ai ddu/btech/sem1/losp/shellscripts/demo.sh
```

```
22184620 ddu/btech/sem1/losp/shellscripts/demo.sh
```

```
[jpandya@JMP learnfind]$ find . -inum 22184620
```

```
./ddu/btech/sem1/losp/shellscripts/demo.sh
```

```
[jpandya@JMP learnfind]$
```

## Goal # 7 : How to find files based on size in Unix and Linux

Following find example shows how you can use find -size option to find files based upon certain size. This will find all files in current directory and sub-directory, greater than some size using find command in Unix:

```
find . -size +1000c -exec ls -l {} \;
```

Always use a c after the number, and specify the size in bytes, otherwise you will get confuse because find -size list files based on size of disk block. to find files using a range of file sizes, a minus or plus sign can be specified before the number. The minus sign means "less than," and the plus sign means "greater than." Suppose if you want to find all the files within a range you can use find command as in below example of find:

```
find . -size +10000c -size -50000c -print
```

This [find example](#) lists all files that are greater than 10,000 bytes, but less than 50,000 bytes:

## The find operators

(!, -o, -a)

There are three operators used with find.

(1)

**!' operator is used before an option to negate its meaning.**

So, find . ! -name "\*.c" -print

Selects all but c program. Means all other files then having c extension.

```
[jpandya@JMP learnfind]$ mkdir -p ddu/btech/sem2/cp/
[jpandya@JMP learnfind]$ touch ddu/btech/sem2/cp/helloworld.c
[jpandya@JMP learnfind]$ find . -name "*.c"
./ddu/btech/sem2/cp/helloworld.c
[jpandya@JMP learnfind]$ find . ! -name "*.c"
.
./ddu
./ddu/btech
./ddu/btech/sem1
./ddu/btech/sem1/losp
./ddu/btech/sem1/losp/shellscripts
./ddu/btech/sem1/losp/shellscripts/demo.sh
./ddu/btech/sem1/losp/shellscripts/user1_info.txt
./ddu/btech/sem1/losp/shellscripts/ce28
./ddu/btech/sem1/losp/shellscripts/ce28/test.sh
./ddu/btech/sem2
./ddu/btech/sem2/cp
./testfind.txt
```

```
./myscripts  
[jpandya@JMP learnfind]$
```

**(2) -o operator represents OR condition.**

To look for both shell and perl scripts:

```
find /home \( -name "*.sh" -o -name "*.pl" \) -print
```

**(3)**

**The -a operator represents the AND condition and is implied by default, whenever selection criteria placed together.**

**Miscellaneous**

**Displaying long list of files found via find command using -ls :**

```
[jpandya@JMP learnfind]$ find . -type f -ls  
22183989  4 -rwxrwxr--  1 jpandya  jpandya      0 Sep 18 12:04  
./ddu/btech/sem1/losp/shellscrip/demo.sh.bak  
22184531  4 -rw-r--r--  1 503    505      0 Sep 18 10:31  
./ddu/btech/sem1/losp/shellscrip/user1_info.txt  
22184530  4 -rw-r--r--  1 503    505      0 Sep 18 10:31  
./ddu/btech/sem1/losp/shellscrip/ce28/test.sh  
22183986  4 -rw-rw-r--  1 jpandya  jpandya      0 Sep 18 11:58 ./ddu/btech/sem2/cp/helloworld.c  
22184606  4 -rw-rw-r--  1 jpandya  jpandya      0 Feb 27 2013 ./testfind.txt  
[jpandya@JMP learnfind]$
```

**Taking action on selected files:**

i.e. remove a file using inode number

```
[jpandya@JMP learnfind]$ find . -inum 22184620 -exec rm {} \;  
[jpandya@JMP learnfind]$ ll ddu/btech/sem1/losp/shellscrip/demo.sh  
ls: ddu/btech/sem1/losp/shellscrip/demo.sh: No such file or directory  
[jpandya@JMP learnfind]$
```

```
find / -name core | xargs /bin/rm -f  
find / -name core -exec /bin/rm -f '{}' \; # same thing  
find / -name core -delete                  # same if using Gnu find
```

xargs command is designed to construct argument lists and invoke other utility. xargs reads items from the standard input or pipes, delimited by blanks or newlines, and executes the command one or more times with any initial-arguments followed by items read from standard input.

**Controlling find:**

The -maxdepth and -mindepth options allow you to specify how far down the directory tree you want

find to search. If you want find to look in just one level of the directory, you would use the maxdepth option.

```
[jpandya@JMP learnfind]$ find . -maxdepth 1 -name "*c"
[jpandya@JMP learnfind]$ find . -maxdepth 4 -name "*c"
[jpandya@JMP learnfind]$ find . -maxdepth 5 -name "*c"
./ddu/btech/sem2/cp/helloworld.c
[jpandya@JMP learnfind]$
```

So this way if you wish to search only in current directory and not in any subdirectories set the maxdepth to 1.

To avoid seeing error messages of not having permissions, you may redirect errors to /dev/null as below:

```
find . -perm +220 -exec ls -l {} \; 2> /dev/null
```

Courtesy:

<http://www.oracle.com/technetwork/articles/calish-find-087766.html>

[http://linux.about.com/od/commands/a/blcmdl1\\_findx.htm](http://linux.about.com/od/commands/a/blcmdl1_findx.htm)

<http://www.tutorialized.com/tutorial/10-examples-of-using-find-command-in-Linux/67264>

Text book Unix Concepts and Applications by Sumitabha Das.

Contributors (CE Department, DDU):

- Prof. Jigar M. Pandya