

~~jmfp~~

## ② find :-

The find cmd is used to searching for files & directories in the linux file system & it will display the full path.

Find is one of the most powerful & frequently used cmd. It is also one of the most extensive cmd with over 50 options & this makes it a bit confusing, specially when it is paired with the exec or xargs command.

Syntax - \$ find [path] [options] [filename]

\$ find . filename

### \* Options:-

- → current loc<sup>n</sup>
- .. → Parent loc<sup>n</sup>
- ? → home loc<sup>n</sup>
- ! → root loc<sup>n</sup>
- ~ → previous loc<sup>n</sup>

## ① Search files by name

- we can mention path  
by using absolute path, relative  
path & symbolic notation.

(path)

- \$ find dir -name "filename"

e.g. \$ find work -name "file001"

② \$ find /home/omkar/work -name  
"file001"

③ \$ find . -name "file001"

④ \$ find ~ -name "file001"

② Search files by name ignore  
case or case-insensitive case

→ By default, the find  
cmd is case-sensitive

file  
type ↓      link ↓

- \$ find ~ -type l -iname "slink"
- \$ find ~ -type l -iname "SLINK"
  
- \$ find . -iname file.txt

### \* wildcard characters :-

If we don't know the exact filename or dirname, that kind of files can be found or search or delete by matching with wildcard char. We can use it with any cmd like ls, cd, find, rm etc.

- \* → represent zero or more char
- ? → represent single char
- [ ] → represent a range of char

It is used to match the filename

emp.log → filename

ex - \$ find . - name "em?.log"

① \$ find . - name "???"

It will find all ~~files~~ files having 3 character.

② \$ find . - name "file???"

\$ find . - type f - name "file???"

\$ find . - type d - name "file???"

\* \$ find . - name "joboo?.sh"

\* \$ find . - name "\* .sh"

It will search all files having .sh extension

\$ find . - name " \*.txt "

\$ find . - name " \*001\* "

\$ find . - name " \* .?? "

It will search ash files

`$ find . -name "*.???"`

It will find all .txt files  
or any other files having extension  
with 3 char

### ③ Search files by its type

It will find all files  
by its type

f → file

d → directory

l → link

s → socket

b → block      etc

- `$ find . -type f -name "file.txt"`
- `$ find . -type d -name "dir"`
- `$ find . -type l -name "slink"`
- `$ find . -type f -name "DIR"`

ignoring case sensitivity

\$ find -type d -name dir

We can search under multiple  
folders / directories

→ find dir1 dir2 -name file.txt

• find dir1 dir2 -type f -name file

Q Find all files from specific  
directory & move or copy or  
remove it

→ ① \$ cd study

\$ ls

\$ mkdir study001

\$ cd study

\$ ls

\$ cp \* ~/work/study001

\$ rm \*file\*

copy all files  
from study dir  
& move it to  
study001 dir

④ Search for empty files & dir

→ The -empty option  
enables you to look for empty  
files & dir with the find cmd

- \$ find . -empty

To find all empty files &  
dir in current location

- We can specify the file type to  
look for empty files or dir

- \$ find . -empty -type f

- \$ find . -empty -type d

- \$ find . -type f -empty

- \$ find . -type d -empty

- We can also find all empty files  
or dir with ~~size~~ name

- \$ find . -type d -empty -name '\*file'

~~\$ find . -size +10K~~

~~NOTE →~~

we can combine multiple options with find cmd

~~- \$ find . -empty -type f -name  
"\*.txt"~~

• \$ find . -type f -size 0

\$ find . -type d -size 0

\$ find . -type f -empty -size 0

We can find empty file without using -empty option

We can use -size 0 option to find empty files bcoz empty files means having size 0.

⑤ Find files & directories based on file or dir size

→ It will find all files or dirs by exact size

-num → less than size

+num → greater than size

• \$ find . -size +10K

• \$ find . -size 10K

• \$ find : -size -10K

• \$ find . -type f -size 10K

\$ find . -type d -size 10K

• find files or dir sizes in range between 10K - 20K

• \$ find . -size +10K -size -20K

\$ find . -type f -size +10K  
-size -20K

We can mention name of file also in filter

\$ find . -type f -size 10K -name "file"

- look for dir larger than 20K
- \$ find . -type d -size +20K

#### ⑥ find files by its extension

- \$ find . -name \*.txt

\$ find . -name type f -name "\*.txt"

- search for files with multiple extensions

- \$ find . -type f -name "\*.txt" "\*.py"

- \$ find . -type f (-name "\*.txt" -o -name "\*.pdf" -o -name "\*.doc")

- ⑦ Find files by using inode no  
→ check inode no of file  
or dir by using stat filename  
or ls -l command

\$ find . -inum inode number

\$ find . -inum 15078072

- ⑧ Find files having number of links

- \$ find . -links num

\$ find . -links 5

We can provide number  
as you convenient. It will check  
the number of links to file

For cross verify of the op we  
stat cmd

- Search files by type having  
2 links

\$ find . -type f -links 52

\$ find . -links 7

\$ find . -links 5

\$ find . -links 4

min or max

• search file links more than 4

\$ find . -links +4

\$ find . -links +5

\$ find . -links -4

\$ find . -links -5

\$ find . -type d -links +2

\$ find . -type d -links -2

\$ find . -type f -links +2

\$ find .. -links +7 -links -9

\$ find . -type f -links +7 -links -9

\$ find .. -type d -links +7 -links -10

\$ find . -inum 15079887

## ⑤ Actions on the result of find

\* Search all empty files in current loc<sup>n</sup> and delete

→ we need to perform operation on find cmd o/p by using -exec option , by using we can perform multiple operations with find cmd o/p like copy , move , remove etc

① find all empty files and remove interactively

→  
\$ find . -type f -empty -exec  
rm -i {} \;

we can perform multiple cmd operation with find cmd o/p . (cat , ls , rm , cp , ---etc)

① find all empty files and copy it to one directory

→  

```
$ find . -type f -empty -exec
cp {} dirname \;
```

```
$ find . -type f -empty -exec
cp -i {} dirname \;
```

Note - permission denied files can't be moved for that use -cp -f

② find all empty files & forcefully copy to empty folder /dir

→  

```
$ find . -type f -empty -exec
cp -f {} dirname \;
```

③ → Find all empty files and delete from specified dir

→  

```
$ find dirname -type f -empty
-exec rm {} \;
```

⑤ find all empty files by dirname  
and delete interactively



\$ find . -type f -name "dirname"  
-exec rm -i {} \;

⑥ find all empty files and delete  
directly

→ \$ find . -type f -empty -delete

\* -exec vs -delete

- we can perform multiple op using -exec option
- delete option can perform only one op using delete opt

⑦ find . -type d -empty -exec  
rm -rf {} \;

ix) find multiple files using name  
and long list the search result

→

```
$ find . -type f -name "*.txt"  
      -exec ls -l {};
```

x) find Rename all find results  
with .old extension

```
→ $ find . -type f -name "*.txt"  
      -exec mv {} {} .old \;
```

(xi) find files & change permission

```
→ $ find dirname -name "*.php"  
      -type F -exec chmod 755 {} \;
```

(xii) find & change file & directory  
permissions

```
→ $ find . -type f -perm 644  
      -exec chmod 655 {} \;
```

```
$ find . -type d -perm 644  
-exec chmod 755 {} \;
```

(xii) find & copy a specific file to a directory

→ \$ find -iname file.txt -exec  
cp {} dirname \;

\$ find -iname file.txt -exec  
cp {} ~tmp/images \;

(xi) find & copy one type of files to a directory

→ \$ find -iname "\*.jpg" -exec  
cp {} ~tmp/images \;

(x) find & copy one file to many directories

→ \$ find ~tmp/dir1 ~tmp/dir2/  
\$HOME/3/ -maxdepth 0 -exec cp

~ / numeric / hei { } |;

- (xvi) find & move files to different directory

→ \$ find . -type f -name "file.txt"  
-exec mv {} dirname |;

- (xvii) Search & move files with a certain extension to a diff' dir

→ \$ find . -type f -name ".txt"  
-exec mv {} dirname |;

- (xviii) find certain files & move to a specific diff' folder

→ \$ find . -type f -name unit.\*  
-exec mv {} dirname |;

- (xix) find & delete files & directories

→ \$ find . -type f -name "filename"  
-exec rm -f {} |;

\$ find . -type d -name "filename"  
 -exec rm -rf {} ;

(19) Delete files by extensions

→ \$ find . -type f -name "\*.txt"  
 -exec rm -f {} ;

\$ find . -type f -name "\*.txt"  
 -exec rm -i {} ;

(20) Find & delete files older than n days

→ \$ find . -type f -name "\*.bak"  
 -mtime +20 -exec rm {} ;

\* Alternate command option  
 for -exec is -ok

→ -ok option will ask  
 permission while removing  
 files or dir as yes or no choices

• \$ find dirname -empty -ok  
rm {} ;

Note - There is no rollback option  
for removed files or dir in  
linux , so care be careful while  
using 'rm' command.

Note - We can't use pipe symbol with  
find o/p oper<sup>n</sup> .

• \$ find study -empty | rm

## ⑩ maxdepth and mindepth

It is like a tree level structure

- maxdepth num → It will search upto max dir level from tree level structure
- mindepth num - minimum dir level wise searching from binary tree level structure

Depth is a levels of a dir or files as tree structure

(1, 2, 3, 4, ... )(dir1/dir11/dir2/dir3)

ex

Depth 1 → Download / xyzool

Depth 2 → Download / emp002 / abc

Depth 3 → Download / pqr / abc

Depth 4 → Download / pqr / abc / abc

:

It is a tree structure

use

`ls -R` → recursively displaying all files & directory

① Find files having max depth 4  
 $\rightarrow \$ \text{find} . -\text{maxdepth } 4 -\text{type } f$

② find files having min depth 4  
 $\rightarrow \$ \text{find} . -\text{mindepth } 4 -\text{type } f$

Note - Use 'cd' cmd to perform operation as per location

• ~~\$ mkdir /dir {A, B} / dir {1, 2, 3}~~

• ~~\$ touch dir {A, B} / dir {1, 2, 3}~~

$\quad \quad \quad | \text{file } \{A, B\}$

• lets create a tree structure

① ~~\$ mkdir dir1~~

~~\$ cd dir1~~

~~\$ touch file1~~

~~\$ mkdir dirA dirB~~

\$ touch dirB/file{0..5}

\$ touch dirA/file{0..5}

\$ ls -R

\$ mkdir dir{A,B}/dir{1,2,3}

\$ touch dir{A,B}/dir{1,2,3}/file{A,B}

ex - \$ find . - maxdepth 1

\$ find . - mindepth 1

\$ find . - mindepth 6 - maxdepth 5

\$ find . - mindepth 5 - maxdepth 6

\$ find . - mindepth 3 - maxdepth 5

- type d - empty

⑪ find user related file and directories

→ \$ find . -user root (username)

\$ find . -maxdepth 1 -user root

\$ find . -maxdepth 2 -user root

\$ find . -mindepth 1 -user root

⑫ Find group related files & dir.

→ \$ find . -group groupname

\$ find . -group akshay

\$ find . -group abc

\$ find . -maxdepth 2 -group abc

- find files under home dir
- \$ find /home -name "filename"

PAGE NO.:

DATE: / /

- (13) find files by permissions

Attribute

→ \$ find . -perm 644

\$ find . -perm 777

- (14) find files & directories by using Access time (last accessed)

→ \$ find . -atime 4

- atime - It stands for access time. This timestamp tells you when was the last time the file was accessed

- (1) find files accessed 4 days back

- \$ find . -atime 4

- (2) \$ find . -atime -4

4 → exact 4 days

-4 → upto 4 days / less than 4

+4 → before 4 days / 4 days ago

• \$ find . -atime 4

\$ find . -atime -4

\$ find . -atime +4

① find files accessed 4 days ago &  
upto 10 days

• \$ find . -atime +4 -atime -10

② 19 01/20/2017 09:31:11 31/10/2017 09:31:11  
files

→ \$ find . -ctime +19 -ctime -49  
display 1 month files

days - time  
hours - min

PAGE NO.:

DATE: / /

- mtime -

mtime stands for modify time. This timestamp tells you when was the last time the file was modified.

- ctime -

ctime stands for status change time. This timestamp tells you when was the last time the property of the file were changed.

ctime {

mtime } days

atime }

cmin {

mmin }

amin }

hours

- \$ find . -amin 4

- \$ find . -amin -4

\$ find . - cmin +4 -ctime 10.

\$ find . - atime +4 -atime -10

\$ find . - amin -4 -atime -10

\$ find . - ctime +19 -ctime -49

\$ find . - mtime +19 -mtime -49

\$ find . - atime +19 -atime -49