

## INFORMATION NOTE

From: H. A. Helms  
Date: 02.01.21  
Location: Apeldoorn

## THEME

**find files and directories** on Raspberry and MacBook-Pro-van-Heino by using Linux Terminal commands

## TAGS

Linux. Mac. Terminal commands,

## SUMMARY

1. Find Files with certain names
2. Find Files Based on their Permissions
3. Find Files Based On Owners and Groups
4. Find Files and Directories Based on Date and Time
5. Find Files and Directories Based based on Size

## DESCRIPTION

### 1. Find Files with certain names

1a. Find all the files whose name is tecmint.txt in a **current working directory**  
**find . -name tecmint.txt**

1b. Find all the **files** under /home directory with name tecmint.txt.  
**find /home -name tecmint.txt**

1c. Find all the **files** whose name is tecmint.txt and contains **both capital and small letters** in /home directory.  
**find /home -iname tecmint.txt**

1d. Find all **directories** whose name is **Tecmint** in / directory.  
**find / -type d -name Tecmint**

1e. Find all **php files** whose name is **tecmint.php** in a current working directory.  
**find . -type f -name tecmint.php**

### 2. Find Files Based on their Permissions

2a. Find all the files whose **permissions** are 777 <sup>2)</sup>  
**find . -type f -perm 0777 -print**

2b. Find all the SGID bit files whose **permissions** set to 644.  
**find / -perm 2644**

2c. Find all the Sticky Bit set files whose **permission** are 551.  
**find / -perm 1551**

2d. Find all **SUID** set files <sup>1)</sup>  
**find / -perm /u=s**

2e. Find all **SGID** set files <sup>5)</sup>  
**find / -perm /g=s**

2f. Find all Read Only files.  
**find / -perm /u=r**

2g. Find all Executable files  
**find / -perm /a=x**

2h. Find all 777 permission files **and** use chmod command to set permissions to 644  
**find / -type f -perm 0777 -print -exec chmod 644 {} \;** <sup>3) 4)</sup>

2i. Find all 777 permission directories and use chmod command to set permissions to 755

```
find / -type d -perm 777 -print -exec chmod 755 {} \;
```

2j. Find a single file called tecmint.txt and remove it

```
find . -type f -name "tecmint.txt" -exec rm -f {} \;
```

2k. find and remove multiple files such as .mp3 or .txt, then use

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

```
find . -type f -name "*.mp3" -exec rm -f {} \;
```

2l. find all **empty files** under a certain path.

```
find /tmp -type f -empty
```

2m. file all **empty directories** under a certain path

```
find /tmp -type d -empty
```

2n. find all **hidden files**

```
find /tmp -type f -name ".*"
```

### 3. Find Files Based On Owners and Groups

3a. find all or single file called tecmint.txt under / root directory of owner root.

```
find / -user root -name tecmint.txt
```

3b. find all files that belong to user Tecmint under /home directory

```
find /home -user tecmint
```

3c. find all files that belong to the group Developer under /home directory.

```
find /home -group developer
```

3d. find all .txt files of **user** Tecmint under /home directory

```
find /home -user tecmint -iname "*.txt"
```

### 4. Find Files and Directories Based on Date and Time

4a. find all the files which are modified 50 days back

```
find / -mtime 50
```

4b. find all the files which are accessed **50** days back.

```
find / -atime 50
```

4c. find all the files which are modified more than 50 days back and less than 100 days

```
find / -mtime +50 -mtime -100
```

4e. To find all the files which are changed in the last 1 hour

```
find / -cmin -60
```

4f. To find all the files which are modified in the last 1 hour

```
find / -mmin -60
```

4g. To find all the files which are accessed in the last 1 hour.

```
find / -amin -60
```

### 5. Find Files and Directories Based based on Size

5a. find all 50MB files, use

```
find / -size 50M
```

5b. find all the files which are greater than 50MB and less than 100MB.

```
find / -size +50M -size -100M
```

5c. find all **100MB** files and **delete** them using one single command.

```
find / -type f -size +100M -exec rm -f {} \;
```

5d. Find all **.mp3** files with more than 10MB and **delete** them using one single command.

```
find / -type f -name *.mp3 -size +10M -exec rm {} \;
```

## NOTES

<sup>1)</sup> **SUID** (Set owner **U**ser **I**D up on execution) is a special type of file permissions given to a file. Normally in Linux/Unix when a program runs, it inherit's access permissions from the logged in user. SUID is defined as giving temporary permissions to a user to run a program/file with the permissions of the file owner rather that the user who runs it. **In simple words users will get file owner's permissions as well as owner UID and GID when executing a file/program/command.**

<sup>2)</sup> **Permissions 777:**

0 (0+0+0) – No permission.

1 (0+0+1) – Only execute permission.

2 (0+2+0) – Only write permission.

3 (0+2+1) – Write and execute permissions.

4 (4+0+0) – Only read permission.

5 (4+0+1) – Read and execute permission.

6 (4+2+0) – Read and write permissions.

7 (4+2+1) – Read, write, and execute permission.

<sup>3)</sup> **-exec chmod 644 {} \;**

- The `\;` is a; fed to the program (find) by the `\` escape preventing it from be handled by the shell (normally would separate commands). The `\;` part is basically telling find: "okay, I'm done with the command I wanted to execute"

- The **-exec** argument interprets everything as a command up to that inserted `;` that ends the **-exec** stuff.

- `{}` means "the output of find". As in, "whatever find found". Find returns the path of the file you're looking for, right? So `{}` replaces it; it's a placeholder for each file that the find command locates.

**Example:** Let's say I'm in a directory full of .txt files. I then run: **find . -name '\*.txt' -exec cat {} \;**

- The first part, `find . -name *.txt`, returns a list of the .txt files.

- The second part, `-exec cat {} \;` will execute the `cat` command for every file found by find, so `cat file1.txt`, `cat file2.txt`, and so on.

<sup>4)</sup> **chmod 644**

```
chmod-R 644 folder_name
```

```
chmod -R a+rwx,u-x,g-wx,o-wx folder_name
```

<sup>5)</sup> <https://www.thegeekdiary.com/what-is-suid-sgid-and-sticky-bit/>







