

File permissions →

Three file permissions:-

read: permitted to read the contents of file.

write: permitted to write to the file.

execute: permitted to execute the file as a program/script.

Three directory permissions:

read: permitted to read the contents of directory (view files and sub-directories in that directory).

write: permitted to write in to the directory. (create files and sub-directories in that directory)

execute: permitted to enter into that directory.

Chmod command →

Use the chmod command to set file permissions.

Relative Permissions →

Detecting File Permissions—

You can use the ls command with the -l option to show the file permissions set. For example, for apple.txt, I can do this:

```
$ ls -l apple.txt
```

```
-rwxr--r-- 1 december december    81 Feb 12 12:45 apple.txt
```

The sequence -rwxr--r-- tells the permissions set for the file apple.txt. The first - tells that apple.txt is a file. The next three letters, rwx, show that the owner has read, write, and execute permissions. Then the next three symbols, r--, show that the group permissions are read only. The final three symbols, r--, show that the world permissions are read only.

u Sets permissions for the owner of the file, e.g.: "u+w" allows the owner to write to the file

g Sets permissions for the group (to which owner belongs), e.g. "g-x" suppresses the execution of the file by the group

o Sets permissions for other users (that are not in group), e.g.: "o=r" allows others only to read the file

a Sets permissions for all (owner, group and others), e.g.: "a-w" disables write access to the file for everyone

= Assigns the permissions, e.g. "a=rw", sets read and write permissions and disables execution for all

- Removes certain thing[s] from the permissions, keeping all other (not involved) permissions. E.g. "a-x" disables execution of the file for everyone, this example doesn't touch read and write permissions.

+ Adds certain thing[s] to the permissions, keeping all other (not involved) permissions. E.g. "a+x" allows execution of the file for everyone, this example doesn't touch read and write permissions.

chmod Examples →

[1] Give read, write and execute to everybody (user, group, and others)

read, write and execute = 4 + 2 + 1 = 7.

\$ chmod 777 file.txt

(or)

\$ chmod ugo+rwx file.txt

[3] chmod u=rwx,g=rx,o=rx script.sh

[4] chmod u=rwx,go=rx script.sh

[5] chmod u+rwx,g+rx,g-w,o+rx,o-w script.sh

[6] chmod u+rwx,go+rx,go-w script.sh

[7] chmod o+r data

This grants other read permission to the file data. The command

```
[8]chmod +x data
```

grants everyone (user, group and other) execute permission, and the command

```
[9]chmod g+rx data
```

gives category group read, write and execute permission.

```
$ ls -l socktest.pl
```

```
-rwxr-xr-x 1 nick users 1874 Jan 19 10:23 socktest.pl*
```

```
$ chmod a-x socktest.pl
```

```
$ ls -l socktest.pl
```

```
-rw-r--r-- 1 nick users 1874 Jan 19 10:23 socktest.pl
```

```
$ chmod g+w socktest.pl
```

```
$ ls -l socktest.pl
```

```
-rw-rw-r-- 1 nick users 1874 Jan 19 10:23 socktest.pl
```

```
$ chmod ug+x socktest.pl
```

```
$ ls -l socktest.pl
```

```
-rwxrwxr-- 1 nick users 1874 Jan 19 10:23 socktest.pl*
```

```
$ chmod ug-wx socktest.pl
```

```
$ ls -l socktest.pl
```

Absolute Permissions:-

The chmod command uses a three-digit code as an argument.

The three digits of the chmod code set permissions for these groups in this order:

- Owner (you)
- Group (a group of other users that you set up)
- World (anyone else browsing around on the file system)

Each digit of this code sets permissions for one of these groups as follows.

- Read is 4
- Write is 2
- Execute is 1

The sums of these numbers give combinations of these permissions:-

(00) 0 = no permissions whatsoever; this person cannot read, write, or execute the file

(001) 1 = execute only

(010) 2 = write only 3 = write and execute (1+2)

(100) 4 = read only

(101) 5 = read and execute (4+1)

(110) 6 = read and write (4+2)

(111) 7 = read and write and execute (4+2+1)

Chmod commands on file apple.txt (use wildcards to include more files)

Command	Purpose
---------	---------

chmod 700 apple.txt	Only you can read, write to, or execute apple.txt
---------------------	---

chmod 777 apple.txt	Everybody can read, write to, or execute apple.txt
---------------------	--

chmod 744 apple.txt Only you can read, write to, or execute apple.txt Everybody can read apple.txt;

chmod 444 apple.txt You can only read apple.txt, as everyone else.

Chown command →

The chown command lets you change the owner of a file. Only the superuser can change the owner of a file.

The chown command has the form:

```
chown owner filelist
```

owner :-

The file's new owner; specify the owner by name or by decimal UID.

filelist :-

The list of files whose owner you are changing.

Some more permission commands

```
chmod -R a+x d1
```

```
chmod -R 755 .    works on hidden file also
```

```
chmod -R a+x *    leaves out hidden file
```

Changing file and group ownership together:

```
chown sharma:dba f1          ownweship to sharma group to dba
```

changing group owner:chgrp

```
chgrp dba f1
```

NOTE: see chgrp and chown properly in chapter 7