Linux Essential commands

File Ownership and Permissions

By:Mostafa Mohamed Elsalakawy

July 22, 2025

Contents

1 File Ownership

Every file and directory has both user and group ownership.
A newly-created file will be owned by:

- The user who creates it.
- That user's primary group (unless the file is created in a set group ID (SGID) directory).

File ownership can be changed using # chown command.
Examples:

→ sudo chown user1 file1

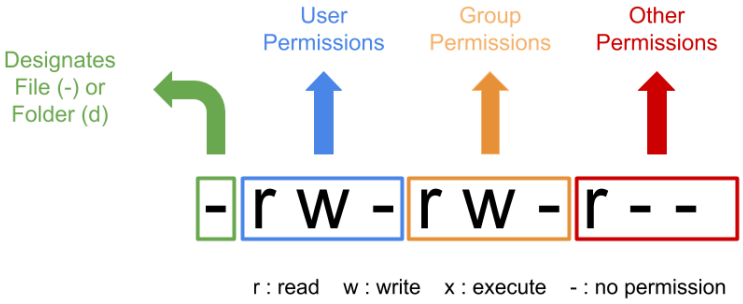
→ sudo chown user1:group1 file1

→ sudo chown :group1 file1

Note:you can change group ownership of a file only if you member in new group. In other hand you can't change user ownership of a file unless you are root or has root's authority

Remember:The # chgrp commod can also changes the group name that a file or directory belongs to.

2 Permissions



2.1 Changing the Permissions

you can changr file or directory permissions by:

→ chmod permission filename

Permissions are specified in either

- Symbolic mode
  - Who
    - u : Owner permissions
    - g : Group permissions.
    - o : Other permissions.
    - a : all permissions.
  - Operator
    - + : Add permissions.
    - − : Remove permissions.
    - + : Assign permissions absolutely.
  - Permissions
    - r : read
    - w : write.
    - x : execute.
    - s : Special Permasiona

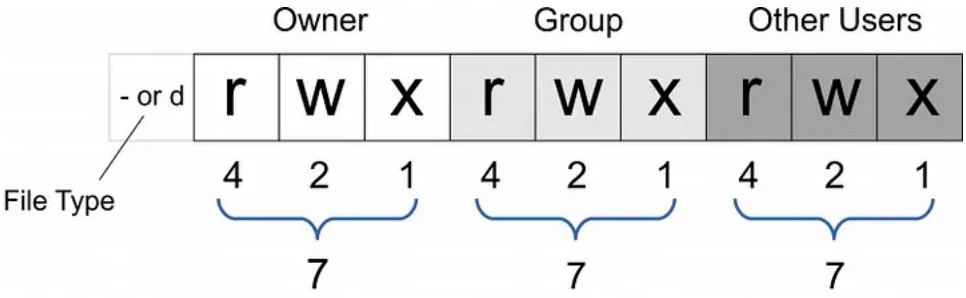
Examples:

command(1)

```
1s -1 filel
=> -rw-r--r-- 1 user1 staff 1319 Mar 22 14:51 file1
chmod o-r file1
1s -1 file1
=> -rw-r----- 1 user1 staff 1319 Mar 22 14:51 file1
chmod g-r file1
ls -1 filel
=> -rw------- 1 user1 staff 1319 Mar 22 14:51 file1
chmod u+x, go+r filel
ls -1 filel
=> -rwxr--r-- 1 user1 staff 1319 Mar 22 14:51 file1
chmod a=rw filel
ls -1 filel
=> -rw-rw-rw- 1 user1 staff 1319 Mar 22 14:51 file1
```

chmod a=rw filel : deletes any previous permissions then set read and write permissions to all

- Octal mode



Permission encoding used by umask		
Octal	Binary	Permissions
0	000	rwx
1	001	rw-
2	010	r-x
3	011	r--
4	100	-wx
5	101	-w-
6	110	--x
7	111	(none)

command(2)

```
chmod 740 file1
ls -1 file1
=> -rwxr- - - - - 1 iti dip 0 01:16 4 file1
chmod 317 file1
ls -l file1
=> --wx--xrwx 1 iti dip 0 01:16 4 file1
```

2.2 Special permissions

SPECIAL PERMISSION	EFFECT ON FILES	EFFECT ON DIRECTORIES
u+s (suid)	File executes as the user that owns the file, not the user that ran the file.	No effect.
g+s (sgid)	File executes as the group that owns the file.	Files newly created in the directory have their group owner set to match the group owner of the directory.
o+t (sticky)	No effect.	Users with write access to the directory can only remove files that they own; they cannot remove or force saves to files owned by other users.

we can also set this permissions using octal mode by adding most significant bit with values:

4 : SUID

2 : SGID

1 : Sticky Bits

2.3 Set user ID (SUID)

```
chmod u+s file1
ls -l file1
=> -rwSrw-r-- 1 iti dib 0 01:18 4 file1
chmod a+x file1
ls -l file1
=> -rwsrwxr-x 1 iti dib 0 01:18 4 file1
```

we see here that in first result **s** is capital this because file wasn't executable
This permission allows as to add password using # passwd command which write in passwd file without write permission to others.when we write #**passwd** command system see us as passwd file owner. And when we write #**passwd user1** the system also see us as passwd file owner so it does't protest but the command itself will return error as we try to change another account password

2.4 Set group ID

Like SUID this permission is used with executable files to make system see our groups as owner group.

If user1 in group **iti** create file in a directory its group is **dib** any created file in the directory will have group **iti**. If we want any created file in the directory has group of directory not owner directory we must set (SGID) permission to the directory.

2.5 Sticky Bits

It's permission for directories only. It used to make any user can create file or execute any file in this group.you may ask question why i don't use w,x permissions ? because this permissions make user can also delete or move the file but sticky bits permission don't. you can't delete or move any file unless you are the root or the owner of the file

2.6 Default Permissions

To set default permissions for created files and directory use :
umask octalcode this will mask given permissions and set other ones
Note : files by default aren't executable even if you unmask x permission

```
unmask 137
touch file1 ; mkdir dir1
ls -l file1 dir1/
=> drw-r----- 2 iti iti 0 01:18 4 dir1/
=> -rw-r----- 1 iti iti 0 01:18 4 file1
touch file2 ; mkdir dir2
ls -l file2 dir2/
=> drwxrwxrwx 2 iti iti 0 01:18 4 dir1/
=> -rw-rw-rw- 1 iti iti 0 01:18 4 file1
```

In last result we can see the files by default not executable