File Permissions

File Permissions:

Every file or directory within Linux has a set of permissions that control who may read, write, and execute the contents. Each of these permissions is represented by an abbreviation (r, w, or x) and has an octal value (see table 1 below).

There are three types of owners in Linux .

1. **User** — who create file
2. **Group** — multiple users and all users of a group have similar permission
3. **Others** — Apart from user or group who has access to the file.

File Types

There are two types of files

1. User Defined

2. System Defined

User Defined files are

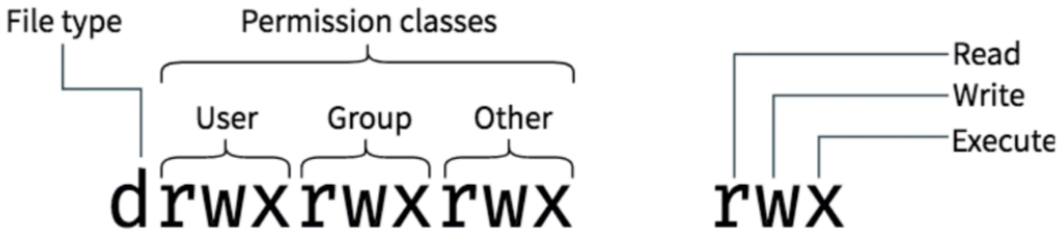
1. **normal denoted by (-)**
2. **directory denoted by (d)**
3. **link (l)**

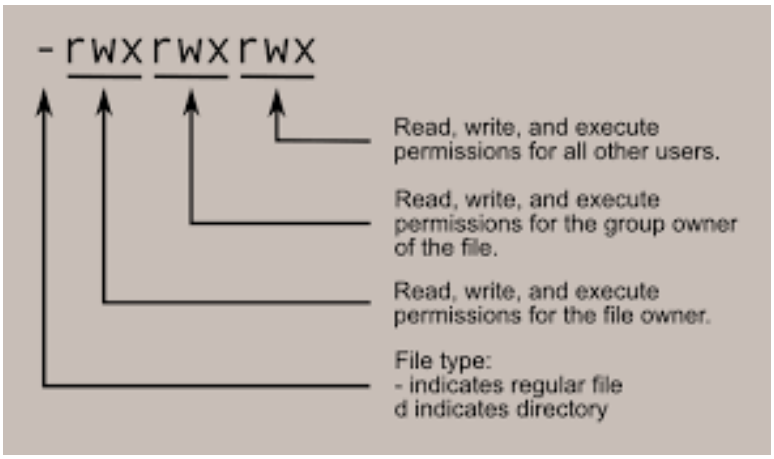
System Defined file are located in /dev

1. **block denoted by (b)**
2. **character denoted by (c)**
3. **pipe denoted by (p)**
4. **socket denoted by (s)**

There are 3 file permission :

READ(r), WRITE(w) , EXECUTE(e)





we can also use number instead of alphabets

Octal	Binary	File Mode
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwx

How can we change permissions of the files ?

'+' to give or assign permission

'-' to remove or delete the permission

'=' to overwrite the permission

Full Permission

1. directory - 777
2. file - 666

Default Permission

1. directory - 755
2. file - 644

Permission is assigned to users , groups, others

- ***users denoted by u***
- ***groups denoted by g***
- ***others denoted by o***

We can attach permission in two ways

1. Numeric
2. Alphabetic

Syntax: chmod <permission> filename

- **Numeric**

r - read = 4

w - write = 2

x -execute = 1

```
[root@localhost ~]# ll
total 4
-rw-------. 1 root root 1425 Apr  6 06:52 anaconda-ks.cfg
-rw-r--r--. 1 root root    0 Apr  6 13:37 file1
drwxr-xr-x. 2 root root   21 Apr  6 06:55 swati
[root@localhost ~]# _
```

1. ***ex: Assign read permission only to the users***

chmod 400 file1

```
[root@localhost ~]# chmod 400 file1
[root@localhost ~]# ll
total 4
-rw-------. 1 root root 1425 Apr  6 06:52 anaconda-ks.cfg
-r--------. 1 root root    0 Apr  6 13:37 file1
drwxr-xr-x. 2 root root   21 Apr  6 06:55 swati
[root@localhost ~]#
```

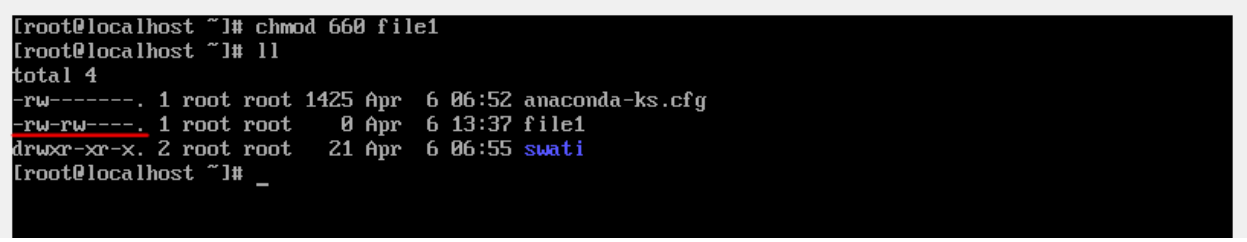
2. ex: Assign write permission only to the groups

chmod 020 file1

```
[root@localhost ~]# chmod 020 file1
[root@localhost ~]# ll
total 4
-rw-------. 1 root root 1425 Apr  6 06:52 anaconda-ks.cfg
----w----. 1 root root    0 Apr  6 13:37 file1
drwxr-xr-x. 2 root root   21 Apr  6 06:55 swati
[root@localhost ~]#
```

3. ***ex: Assign read & write permission only to the user & groups***

chmod 660 file1



- ***Alphabetic***

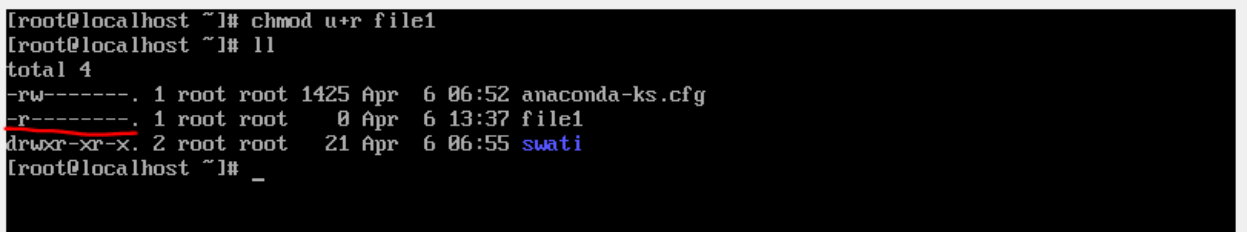
r - read

w - write

x -execute

1. **ex. Assign read permission only to the users**

chmod u+r file1



2. **ex . Assign read, write & execute to all users**

chmod ugo+rwx file1

```
[root@localhost ~]# chmod ugo+rwx file1
[root@localhost ~]# ll
total 4
-rw-------. 1 root root 1425 Apr  6 06:52 anaconda-ks.cfg
-rwxrwxrwx. 1 root root    0 Apr  6 13:37 file1
drwxr-xr-x. 2 root root   21 Apr  6 06:55 swati
[root@localhost ~]#
```

Or

chmod a+rwx file1

```
[root@localhost ~]# chmod a+rwx file1
[root@localhost ~]# ll
total 4
-rw-------. 1 root root 1425 Apr  6 06:52 anaconda-ks.cfg
-rwxrwxrwx. 1 root root    0 Apr  6 13:37 file1
drwxr-xr-x. 2 root root   21 Apr  6 06:55 swati
[root@localhost ~]# _
```

3. **ex. Remove read permission for the users**

chmod u-r file1

```
[root@localhost ~]# chmod u-r file1
[root@localhost ~]# ll
total 4
-rw-------. 1 root root 1425 Apr  6 06:52 anaconda-ks.cfg
--wxrwxrwx. 1 root root    0 Apr  6 13:37 file1
drwxr-xr-x. 2 root root   21 Apr  6 06:55 swati
[root@localhost ~]# _
```

UMASK

Formula for default permission

umask = full permission - default permission

umask (user file creation mode mask)

umask is responsible for default permission of the file or directory .

umask (default)	root user	local user
root - 022 644	file - 644	file -
local - 002 directory -775	directory - 755	

To set a new temporary umask

syntax : umask 044

To change umask permanently

- ***You need make changes into the bashrc file which is located in /etc folder***

etc/ bashrc

- ***After editing the bashrc file you need to update the file***

source bashrc

Difference between soft-link and hard-link

Hardlink	Softlink
• Inode number is same.	Inode number is different.
• Link count increases.	Link count remain same .

• If we delete the original file data will remain same.	If we delete the original file data will not be available.
• Path will be denoted	link & pointer denoted for softlink.
• Hardlink cannot be used for directory.	we can use for both file & directory.