

FILE ACCESS PERMISSION IN LINUX

File Permissions:

All the three owners (user owner, group, others) in the Linux system have three types of permissions defined. Nine characters denotes the three types of permissions.

1. **Read (r)** : The read permission allows you to open and read the content of a file. But you can't do any editing or modification in the file.
2. **Write (w)** : The write permission allows you to edit, remove or rename a file. For instance, if a file is present in a directory, and write permission is set on the file but not on the directory, then you can edit the content of the file but can't remove, or rename it.
3. **Execute (x)**: In Unix type system, you can't run or execute a program unless execute permission is set. But in Windows, there is no such permission available.

permission	on a file	on a directory
r (read)	read file content (cat)	read directory content (ls)
w (write)	change file content (vi)	create file in directory (touch)
x (execute)	execute the file	enter the directory (cd)

Permission Set:

```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ ls -l  
total 68  
-rw-rw-r-- 1 sssit sssit 64 Jun 27 14:14 acb.bz2  
drwxr-xr-x 4 sssit sssit 4096 Jun 29 12:28 Desktop
```

Look at the above snapshot, there are ten characters (-rw-rw-r--) before the user owner. We'll describe these ten characters here.

position	characters	ownership
1	-	denotes file type
2-4	rw-	permission for user

5-7	rw-	permission for group
8-10	r--	permission for other

File permissions for (-rw-rw-r--)

When you are the **User owner**, then the user owner permission applies to you. Other permissions are not relevant to you.

When you are the **Group** then the group permission applies to you. Other permissions are not relevant to you.

When you are the **Other**, then the other permission applies to you. User and group permissions are not relevant to you.

Permission Example

Now we'll show some examples how permissions can be seen for a file or directory.

```

sssit@JavaTpoint: ~
sssit@JavaTpoint:~$ ls -lh
total 68K
-rw-rw-r-- 1 sssit sssit  64 Jun 27 14:14 acb.bz2
drwxr-xr-x 4 sssit sssit 4.0K Jun 29 12:28 Desktop
drwx----- 4 sssit sssit 4.0K May 18 11:20 Disk1
drwxr-xr-x 4 sssit sssit 4.0K Jul  7 09:35 Documents
drwxr-xr-x 2 sssit sssit 4.0K Jun 29 18:06 Downloads
-rw-rw-r-- 1 sssit sssit  37 Jun 13 17:08 dupli.txt

```

Look at the above snapshot, different directories and files have different permissions.

First letter (-) or **d** represents the files and directories respectively.

Now, from remaining nine letters, **first** triplet represents the permission for **user owner**. Second triplet represents the permission for **group owner**. **Third** triplet represents the permission for **other** .

Setting Permissions With chmod:

You can change the permissions with chmod command accordingly to your need. Below are some examples to change the permissions for different groups.

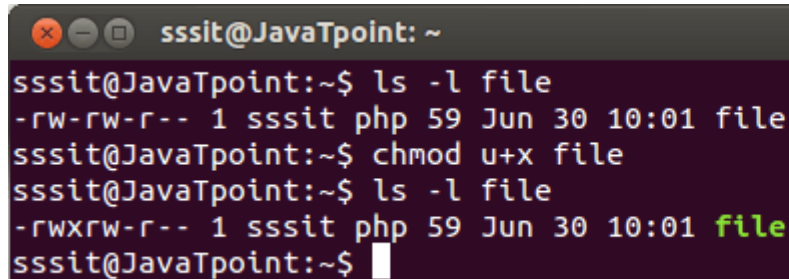
To add permissions to a group.

Syntax:

1. `chmod <groupName>+<permissionName> <fileName>`

Example:

1. `chmod u+x file`



```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ ls -l file  
-rw-rw-r-- 1 sssit php 59 Jun 30 10:01 file  
sssit@JavaTpoint:~$ chmod u+x file  
sssit@JavaTpoint:~$ ls -l file  
-rwxrw-r-- 1 sssit php 59 Jun 30 10:01 file  
sssit@JavaTpoint:~$
```

Look at the above snapshot, permission to execute is added to the user owner group.

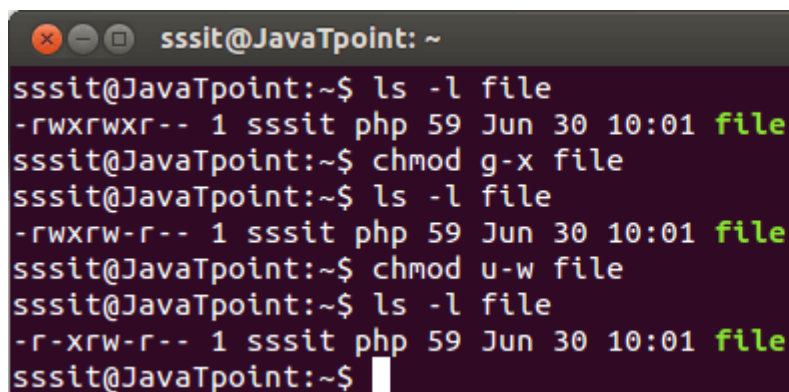
To remove permissions from a group:

Syntax:

1. `chmod <groupName>-<permissionName> <fileName>`

Example:

1. `chmod g-x file`
2. `chmod u-w file`



```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ ls -l file  
-rwxrwxr-- 1 sssit php 59 Jun 30 10:01 file  
sssit@JavaTpoint:~$ chmod g-x file  
sssit@JavaTpoint:~$ ls -l file  
-rwxrw-r-- 1 sssit php 59 Jun 30 10:01 file  
sssit@JavaTpoint:~$ chmod u-w file  
sssit@JavaTpoint:~$ ls -l file  
-r-xrw-r-- 1 sssit php 59 Jun 30 10:01 file  
sssit@JavaTpoint:~$
```

Look at the above snapshot, permission to execute is removed from the group and permission to write is removed from the user owner.

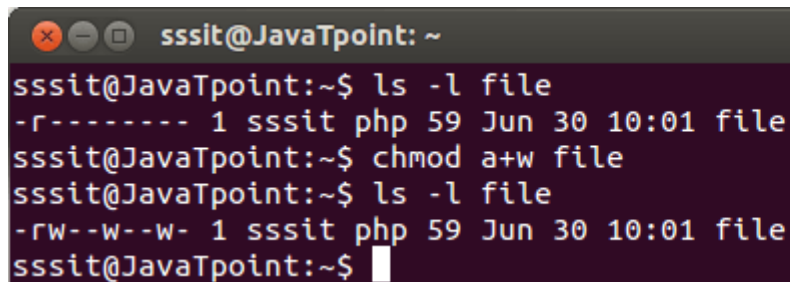
To add permission to all the groups together

Syntax:

1. `chmod a+<permissionName> <fileName>`

Example:

1. `chmod a+w file`

A terminal window titled 'sssit@JavaTpoint: ~' showing the following commands and output:

```
sssit@JavaTpoint:~$ ls -l file
-r----- 1 sssit php 59 Jun 30 10:01 file
sssit@JavaTpoint:~$ chmod a+w file
sssit@JavaTpoint:~$ ls -l file
-rw--w--w- 1 sssit php 59 Jun 30 10:01 file
sssit@JavaTpoint:~$
```

Look at the above snapshot, we have given permission to write for all the groups.

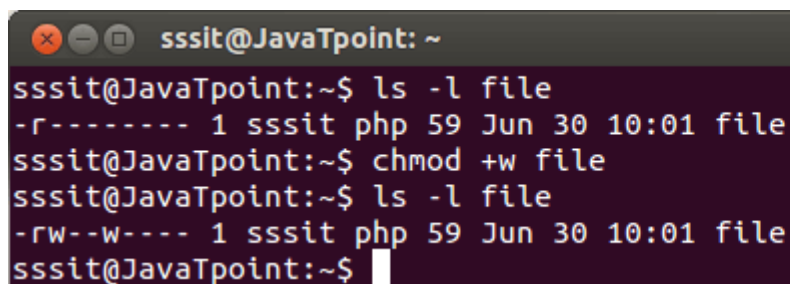
To add permission to all the groups without typing a

Syntax:

1. `chmod +<permissionName> <fileName>`

Example:

1. `chmod +w file`

A terminal window titled 'sssit@JavaTpoint: ~' showing the following commands and output:

```
sssit@JavaTpoint:~$ ls -l file
-r----- 1 sssit php 59 Jun 30 10:01 file
sssit@JavaTpoint:~$ chmod +w file
sssit@JavaTpoint:~$ ls -l file
-rw--w---- 1 sssit php 59 Jun 30 10:01 file
sssit@JavaTpoint:~$
```

Look at the above snapshot, this example is same as the earlier one only difference is that we haven't typed a in this.

To set explicit permission:

Syntax:

1. `chmod <groupName>=<permissions> <fileName>`

Example:

1. chmod **o=rw** file

```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ ls -l file  
-rw--w---- 1 sssit php 59 Jun 30 10:01 file  
sssit@JavaTpoint:~$ chmod o=rw file  
sssit@JavaTpoint:~$ ls -l file  
-rw--w-rw- 1 sssit php 59 Jun 30 10:01 file  
sssit@JavaTpoint:~$
```

Look at the above snapshot, we have set explicit permission to read and write for others.

To set explicit permissions for different groups:

Syntax:

1. chmod **<groupName>=<permissions> <fileName>**

Example:

1. chmod **u=rwx,g=rw,o=r** file

```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ ls -l file  
-rw--w-rw- 1 sssit php 59 Jun 30 10:01 file  
sssit@JavaTpoint:~$ chmod u=rwx,g=rw,o=r file  
sssit@JavaTpoint:~$ ls -l file  
-rwxrw-r-- 1 sssit php 59 Jun 30 10:01 file  
sssit@JavaTpoint:~$
```

Look at the above snapshot, we have set permissions for all the three groups.

Setting Octal Permissions:

Octal permissions can also be set for the groups.

For example, to set **r** octal will be **4**, to set **w** octal will be **2**, to set **x** octal will be **1**.

Octal Table:

binary	octal	permissions
000	0	---

001	1	--X
010	2	-W-
011	3	-WX
100	4	r--
101	5	r-X
110	6	rw-
111	7	rwX

From this we can conclude that,

1. **777** = **rwxrwxrwx**
2. **765** = **rwxrw-r-x**
3. **654** = **rw-r-xr--**

and so on.

```

sssit@JavaTpoint: ~
sssit@JavaTpoint:~$ chmod 777 file
sssit@JavaTpoint:~$ ls -l file
-rwxrwxrwx 1 sssit php 59 Jun 30 10:01 file
sssit@JavaTpoint:~$ chmod 274 file
sssit@JavaTpoint:~$ ls -l file
--w-rwxr-- 1 sssit php 59 Jun 30 10:01 file
sssit@JavaTpoint:~$ chmod 111 file
sssit@JavaTpoint:~$ ls -l file
---x--x--x 1 sssit php 59 Jun 30 10:01 file
sssit@JavaTpoint:~$

```

Look at the above snapshot, we have shown some random octal examples with the numbers **777**, **274** and **111**.