

Day 6 Task: File Permissions and Access Control Lists

The concept of Linux File permission and ownership is important in Linux. Here, we will be working on Linux permissions and ownership and will do tasks on both of them. Let us start with the Permissions.

Permission : Linux is a multi-user operating system, so it has security to prevent people from accessing each other's confidential files. Every file, directory, and other system objects in Linux are assigned an owner, a group and an otheruser. This is the most basic, yet essential, part of system security that protects users from each other. Owners, users belonging to a group, and all others may be granted different types of access to read from, write to, or execute files. This is generally referred to as file permissions in Linux.

Types of Permissions Groups:

Owners :- The owner's permissions will only apply to owner of the file or directory, determine what actions the owner of the file can perform on the file and will not affect the other groups.

Groups :- The group's permissions you can assign a group of users specific permission, which will only impact users within the group.

Other users :- The permissions for others indicate what action all other users can perform on the file.

Types of file permissions:

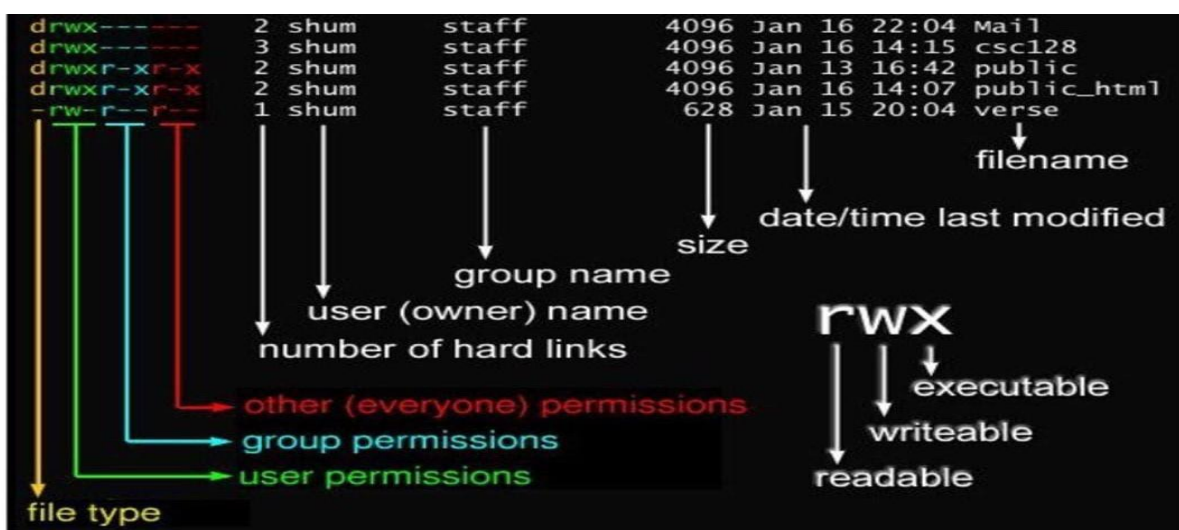
Read permission :- Allows a user or group to read/view the contents of the file

Write permission :- Allows a user or group to modify the contents of the file

Execute permission :- Allows a user or group to execute or run a file as a program.

1. Create a simple file and do ls -l command to see the details of the files

```
[ec2-user@ip-172-31-80-226 day-6]$ ls -l
total 0
-rw-rw-r-- 1 ec2-user ec2-user 0 Jan 10 06:00 file_permission.txt
drwxrwxr-x 2 ec2-user ec2-user 6 Jan 10 06:03 files
```



Some command to change the file permission

- **Chown command** : Used to Change the ownership permission of a file or directory

Syntax : 1. Sudo chown root filename

```
[ec2-user@ip-172-31-80-226 day-6]$ sudo chown root file_permission.txt
[ec2-user@ip-172-31-80-226 day-6]$ ls -l
total 0
-rw-rw-r-- 1 root      ec2-user 0 Jan 10 06:00 file_permission.txt
drwxrwxr-x 2 ec2-user ec2-user 6 Jan 10 06:03 files
```

Syntax : 2. Sudo chow root:root filename

```
[ec2-user@ip-172-31-80-226 day-6]$ sudo chown root:root file_permission.txt
[ec2-user@ip-172-31-80-226 day-6]$ ls -l
total 0
-rw-rw-r-- 1 root      root      0 Jan 10 06:00 file_permission.txt
drwxrwxr-x 2 ec2-user ec2-user 6 Jan 10 06:03 files
```

- **Chmod command** : Used to change the users permission of file/directory

Syntax :

sudo chmod +r filename = Owner & groups get only reading access.

sudo chmod +w filename = Owner & groups get only writing access.

sudo chmod +x filename = Owner & groups get only executing access.

Number	Octal Permission Representation	Ref
0	No permission	---
1	Execute permission	--x
2	Write permission	-w-
3	Execute and write permission: 1 (execute) + 2 (write) = 3	-wx
4	Read permission	r--
5	Read and execute permission: 4 (read) + 1 (execute) = 5	r-x
6	Read and write permission: 4 (read) + 2 (write) = 6	rw-
7	All permissions: 4 (read) + 2 (write) + 1 (execute) = 7	rwX

```
[ec2-user@ip-172-31-80-226 day-6]$ ls -l
total 0
-rw-rw-r-- 1 root      root      0 Jan 10 06:00 file_permission.txt
drwxrwxr-x 2 ec2-user ec2-user 6 Jan 10 06:03 files
[ec2-user@ip-172-31-80-226 day-6]$ sudo chmod +x file_permission.txt
[ec2-user@ip-172-31-80-226 day-6]$ ls -l
total 0
-rwxrwxr-x 1 root      root      0 Jan 10 06:00 file_permission.txt
drwxrwxr-x 2 ec2-user ec2-user 6 Jan 10 06:03 files
```

What is Access Control List (ACL)

ACLs allow us to apply a more specific set of permissions to a file or directory without (necessarily) changing the base ownership and permissions. They let us "tack on" access for other users or groups.

Some Commands: setfacl and getfacl are used for setting up ACL and showing ACL respectively.

- **getfacl**- Displays access permissions of files and directories with filename, owner, group and the ACL's.

Syntax: getfacl filename/directory name

```
[ec2-user@ip-172-31-80-226 day-6]$ getfacl file_permission.txt
# file: file_permission.txt
# owner: root
# group: root
user::rwx
group::rwx
other::r-x
```

- **setfacl**- Helps in adding permissions to user and groups.

Syntax: setfacl -m u/g:username/groupname:permissions /path of file