Sanjana thamke

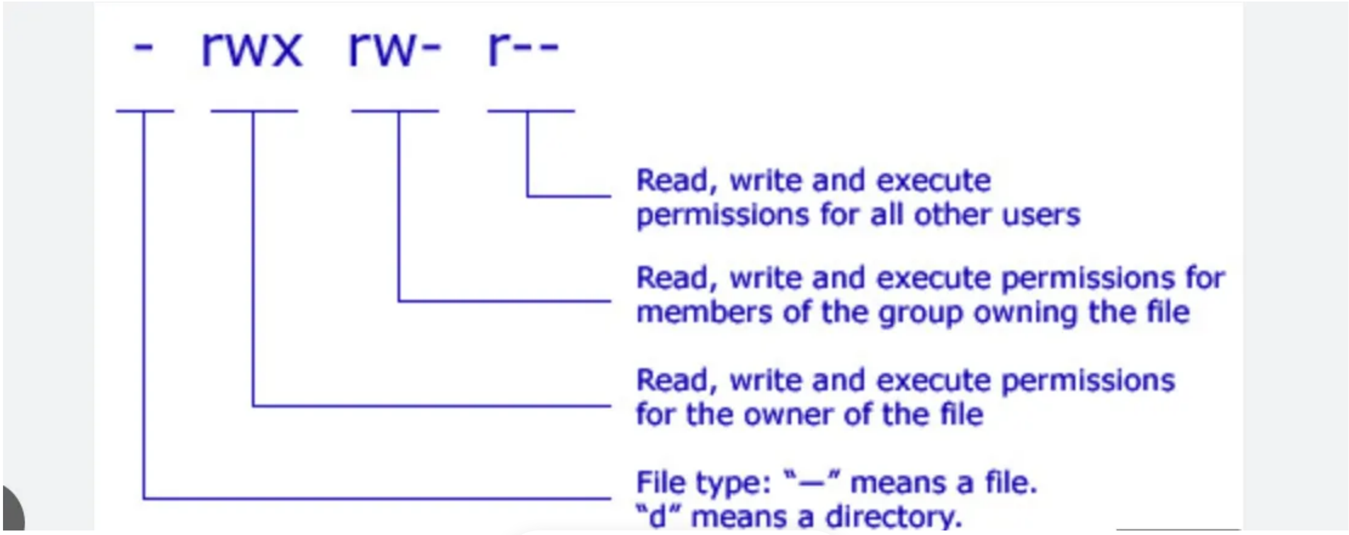
Feb 17 · 3 min read · Listen

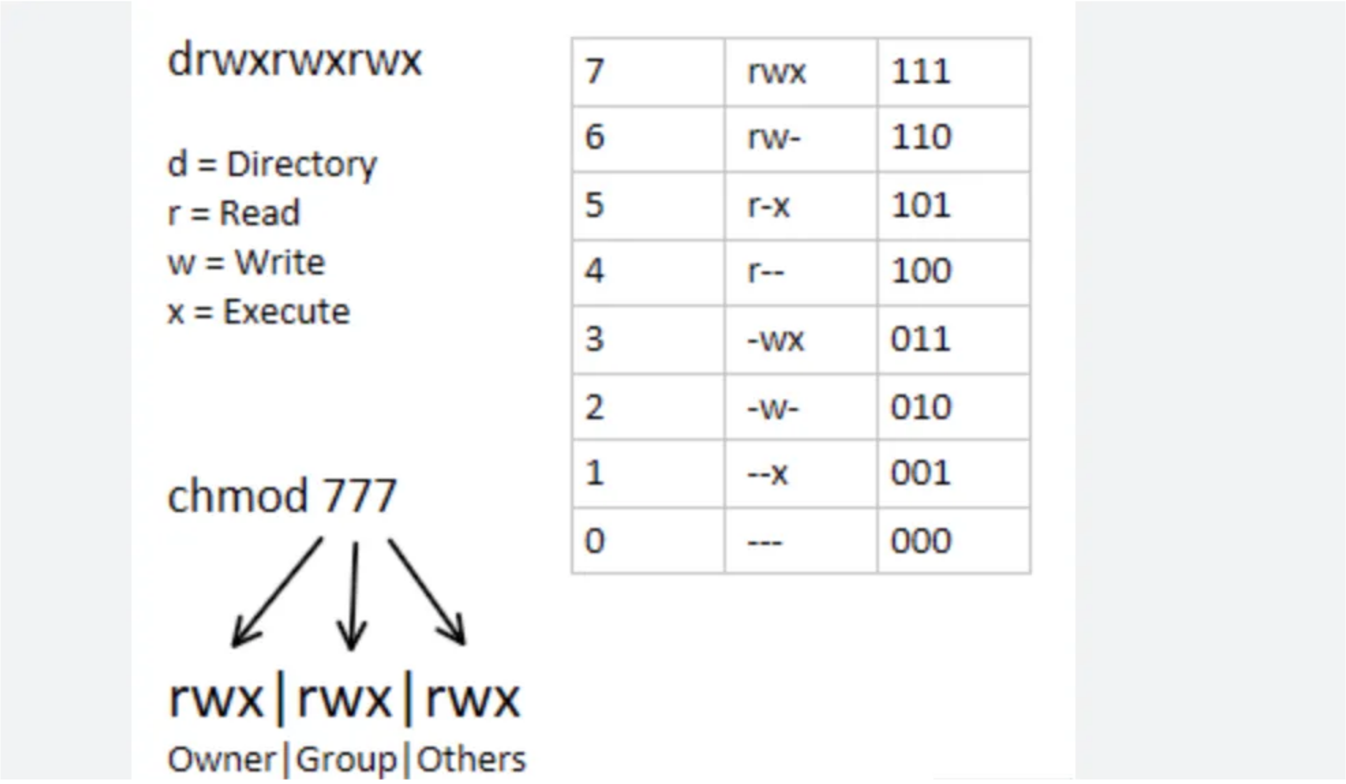
Day 6 Task: File Permissions and Access Control Lists

1. **Write an article about File Permissions based on your understanding from the notes.**

a) All the three owners (user owner, group, others) in the Linux system have three types of permissions defined. File permissions can be categorized in these three categories :- **read**, **write** and **execute**.

> Read (r):- It is the permission by which user will be able to view content of the file.
>
> Write(w) : It is the permission by which user can edit or delete the file
>
> Execute(x) :- It is the permission by which user will be able to execute the script (Like bash scripts).


- rwx rw- r--
Read, write and execute permissions for all other users
Read, write and execute permissions for members of the group owning the file
Read, write and execute permissions for the owner of the file
File type: "—" means a file. "d" means a directory.



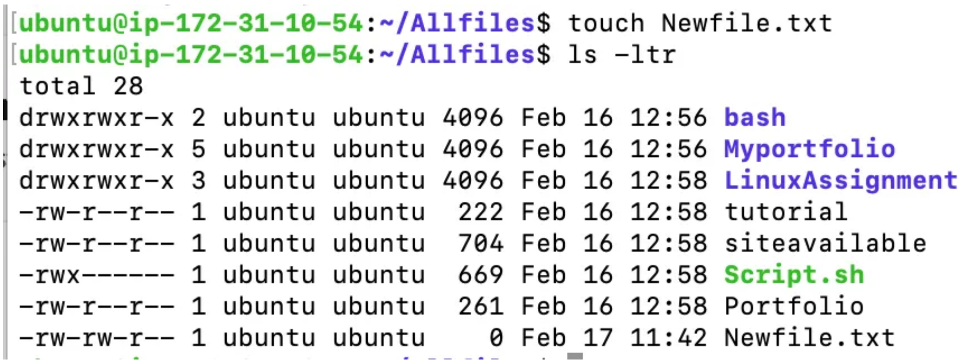
drwxrwxrwx

d = Directory
r = Read
w = Write
x = Execute

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

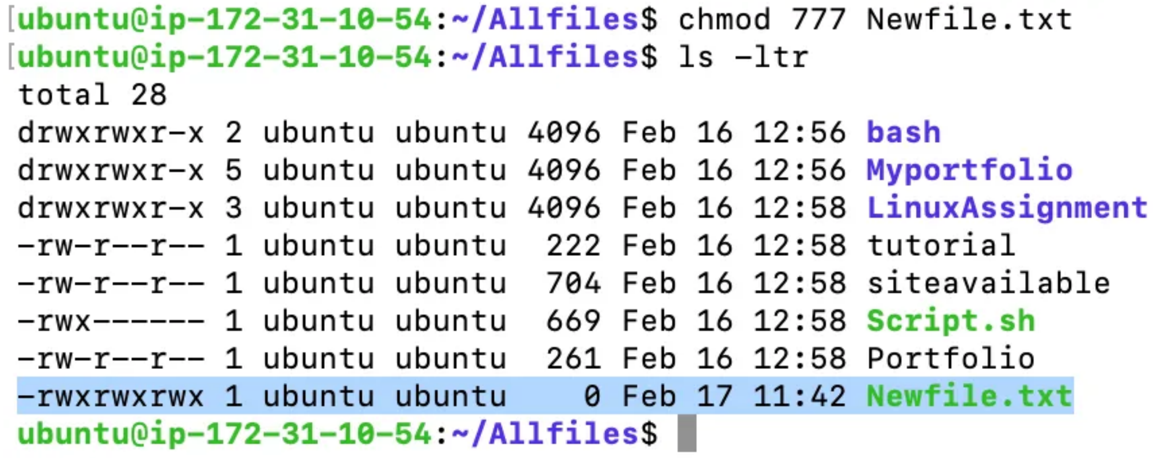
Example:

1. Create a any file : touch Newfile.txt
2. Run command : *ls- ltr*
3. Observe in first column file permissions are visible
4. Permission of Newfile.txt are -rw-rw-r — : means file with read ,write permission for owner and group . read permission only to others



```
ubuntu@ip-172-31-10-54:~/Allfiles$ touch Newfile.txt
ubuntu@ip-172-31-10-54:~/Allfiles$ ls -ltr
total 28
drwxrwxr-x 2 ubuntu ubuntu 4096 Feb 16 12:56 bash
drwxrwxr-x 5 ubuntu ubuntu 4096 Feb 16 12:56 Myportfolio
drwxrwxr-x 3 ubuntu ubuntu 4096 Feb 16 12:58 LinuxAssignment
-rw-r--r-- 1 ubuntu ubuntu  222 Feb 16 12:58 tutorial
-rw-r--r-- 1 ubuntu ubuntu  704 Feb 16 12:58 siteavailable
-rwx------ 1 ubuntu ubuntu  669 Feb 16 12:58 Script.sh
-rw-r--r-- 1 ubuntu ubuntu  261 Feb 16 12:58 Portfolio
-rw-rw-r-- 1 ubuntu ubuntu    0 Feb 17 11:42 Newfile.txt
```

5. Run *Chmod 777 Newfile.txt* and observe file permission get modified



```
[ubuntu@ip-172-31-10-54:~/Allfiles$ chmod 777 Newfile.txt
[ubuntu@ip-172-31-10-54:~/Allfiles$ ls -ltr
total 28
drwxrwxr-x 2 ubuntu ubuntu 4096 Feb 16 12:56 bash
drwxrwxr-x 5 ubuntu ubuntu 4096 Feb 16 12:56 Myportfolio
drwxrwxr-x 3 ubuntu ubuntu 4096 Feb 16 12:58 LinuxAssignment
-rw-r--r-- 1 ubuntu ubuntu  222 Feb 16 12:58 tutorial
-rw-r--r-- 1 ubuntu ubuntu  704 Feb 16 12:58 siteavailable
-rwx------ 1 ubuntu ubuntu  669 Feb 16 12:58 Script.sh
-rw-r--r-- 1 ubuntu ubuntu  261 Feb 16 12:58 Portfolio
-rwxrwxrwx 1 ubuntu ubuntu    0 Feb 17 11:42 Newfile.txt
ubuntu@ip-172-31-10-54:~/Allfiles$ 
```

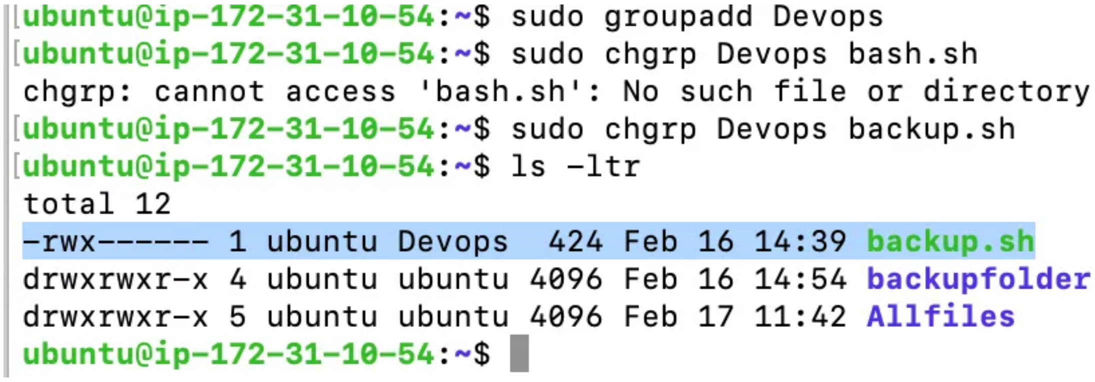
b) chgrp command is used to change the group ownership of the file or directory.

1. create a group by

> *# sudo groupadd Devops*

2. Now lets change the group ownership of backup.sh file using

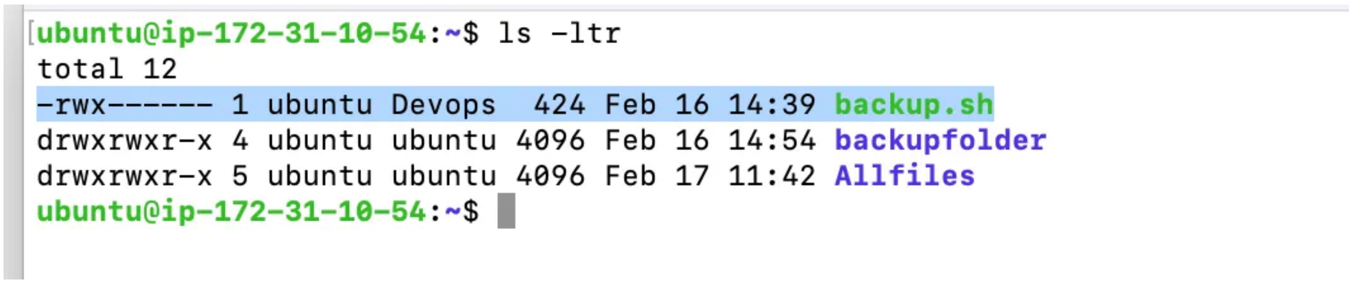
> *# sudo chgrp Devops backup.sh*



```
[ubuntu@ip-172-31-10-54:~$ sudo groupadd Devops
[ubuntu@ip-172-31-10-54:~$ sudo chgrp Devops bash.sh
chgrp: cannot access 'bash.sh': No such file or directory
[ubuntu@ip-172-31-10-54:~$ sudo chgrp Devops backup.sh
[ubuntu@ip-172-31-10-54:~$ ls -ltr
total 12
-rwx------ 1 ubuntu Devops  424 Feb 16 14:39 backup.sh
drwxrwxr-x 4 ubuntu ubuntu 4096 Feb 16 14:54 backupfolder
drwxrwxr-x 5 ubuntu ubuntu 4096 Feb 17 11:42 Allfiles
ubuntu@ip-172-31-10-54:~$ 
```

c) To change the user ownership of a file we can use **chown** command.

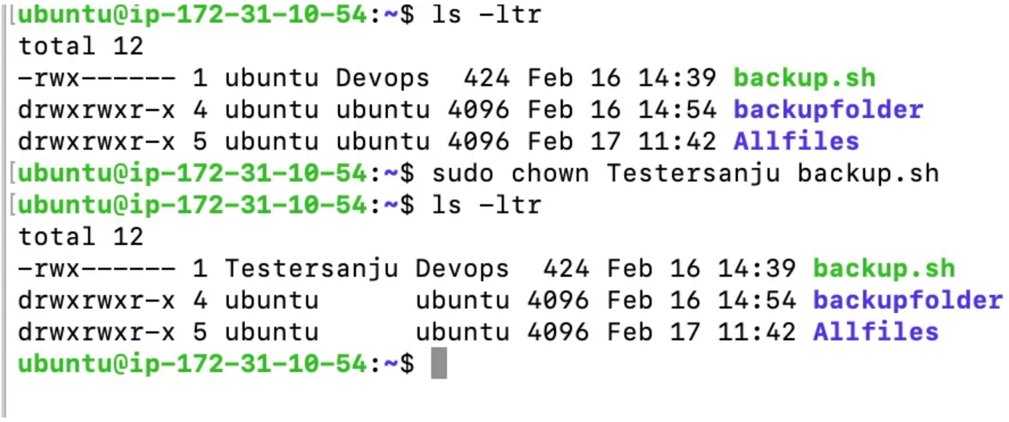
Lets first see the current owner of the file by using ls -ltr command, this is the type of output you should get :-



```
[ubuntu@ip-172-31-10-54:~$ ls -ltr
total 12
-rwx------ 1 ubuntu Devops  424 Feb 16 14:39 backup.sh
drwxrwxr-x 4 ubuntu ubuntu 4096 Feb 16 14:54 backupfolder
drwxrwxr-x 5 ubuntu ubuntu 4096 Feb 17 11:42 Allfiles
ubuntu@ip-172-31-10-54:~$ 
```

Lets change the user ownership of the file

> *#sudo chown Testersanju backup.sh*



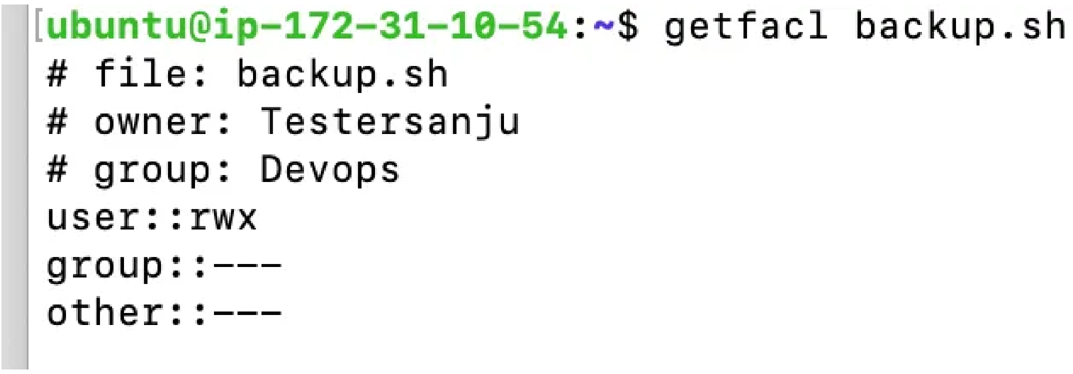
```
ubuntu@ip-172-31-10-54:~$ ls -ltr
total 12
-rwx------ 1 ubuntu Devops  424 Feb 16 14:39 backup.sh
drwxrwxr-x 4 ubuntu ubuntu 4096 Feb 16 14:54 backupfolder
drwxrwxr-x 5 ubuntu ubuntu 4096 Feb 17 11:42 Allfiles
ubuntu@ip-172-31-10-54:~$ sudo chown Testersanju backup.sh
ubuntu@ip-172-31-10-54:~$ ls -ltr
total 12
-rwx------ 1 Testersanju Devops  424 Feb 16 14:39 backup.sh
drwxrwxr-x 4 ubuntu      ubuntu 4096 Feb 16 14:54 backupfolder
drwxrwxr-x 5 ubuntu      ubuntu 4096 Feb 17 11:42 Allfiles
ubuntu@ip-172-31-10-54:~$ 
```

d)Access Control Lists (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

Case :What if you have an accounting intern (Kenny) who needs to be able to read certain files (or even just the files owned by Fred, his manager)? Or maybe people in the sales department also need access to the `accounting` owner's files to create invoices for Fred's team in order to bill customers, but you don't want the sales team to see the other reports that Fred's team generates. This situation can be tricky because, with regular permissions, each file and directory can have only one user and group owner at a time. This type of situation is what Linux Access Control Lists (ACLs) were intended to resolve.

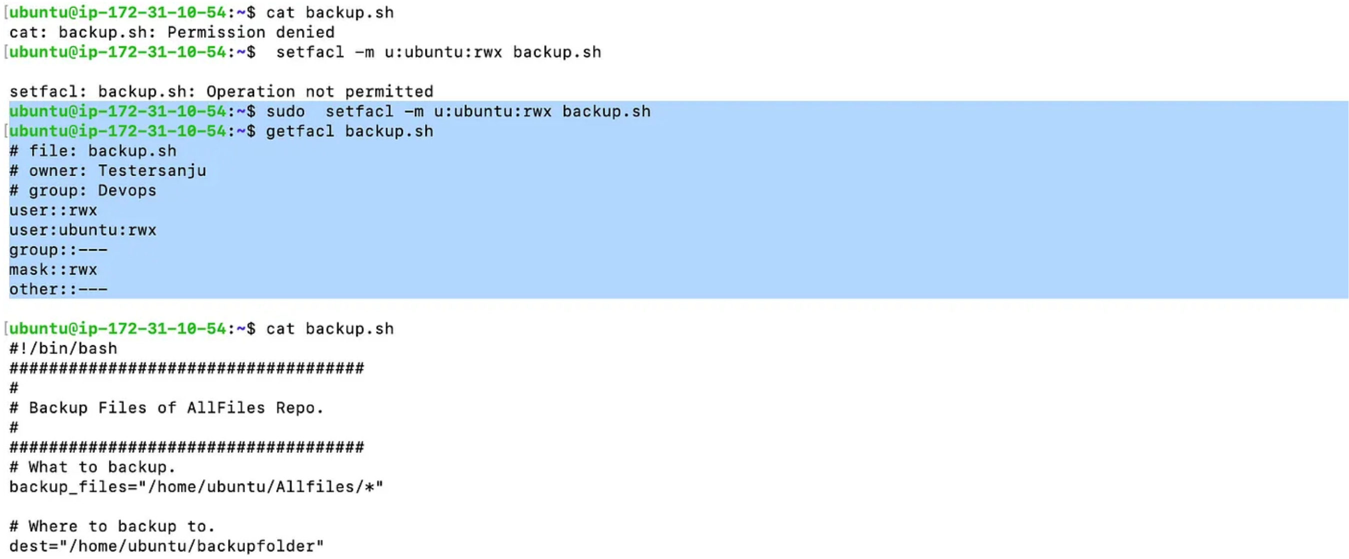
We can view the current ACL using the `getfacl` command:



```
ubuntu@ip-172-31-10-54:~$ getfacl backup.sh
# file: backup.sh
# owner: Testersanju
# group: Devops
user::rwx
group::---
other::---
```

As per above information backup.sh file can read ,write ,execute by Testersanju owner only

Now by using below *setfacl* command , user ubuntu (owner) is able to read ,write ,execute in backup.sh file



```
[ubuntu@ip-172-31-10-54:~$ cat backup.sh
cat: backup.sh: Permission denied
[ubuntu@ip-172-31-10-54:~$  setfacl -m u:ubuntu:rwx backup.sh

setfacl: backup.sh: Operation not permitted
ubuntu@ip-172-31-10-54:~$ sudo  setfacl -m u:ubuntu:rwx backup.sh
[ubuntu@ip-172-31-10-54:~$ getfacl backup.sh
# file: backup.sh
# owner: Testersanju
# group: Devops
user::rwx
user:ubuntu:rwx
group::---
mask::rwx
other::---

[ubuntu@ip-172-31-10-54:~$ cat backup.sh
#!/bin/bash
####################################
#
# Backup Files of AllFiles Repo.
#
####################################
# What to backup.
backup_files="/home/ubuntu/Allfiles/*"

# Where to backup to.
dest="/home/ubuntu/backupfolder"
```

_Thank you

_Sanjana ✌🏼

Linux Filé DevOps Learning