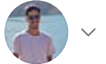


Open in app ↗

Get unlimited access



Search Medium



Fayssal Elaazouzi

Mar 22 · 3 min read · Listen



Save



Day 6 : File Permissions and Access Control Lists



This is #90DaysOfDevops challenge under the guidance of Shubham Londhe sir.

Day 6 TASK

check this for task:

90DaysOfDevOps/tasks.md at master · LondheShubham153/90DaysOfDevOps

This repository is a Challenge for the DevOps Community to get stronger in DevOps. This challenge starts on the 1st...

github.com

1. Change the user permissions of the file and note the changes after `ls -ltr`

```
fayssal@fayssal-VirtualBox:~$ ls
combined.sh  D1  D2  D3  day  Desktop  directory  Documents  Downloads  Music  my_directory  my_directory2  New  P
fayssal@fayssal-VirtualBox:~$ cd D1
fayssal@fayssal-VirtualBox:~/D1$ ls
fayssal@fayssal-VirtualBox:~/D1$ touch file1.txt
fayssal@fayssal-VirtualBox:~/D1$ ls
file1.txt
fayssal@fayssal-VirtualBox:~/D1$ chmod 777 file1.txt
fayssal@fayssal-VirtualBox:~/D1$ ls -ltr
total 0
-rwxrwxrwx 1 fayssal fayssal 0 16:58 22 ميس file1.txt
fayssal@fayssal-VirtualBox:~/D1$ chmod 754 file1.txt
fayssal@fayssal-VirtualBox:~/D1$ ls -ltr
total 0
-rwxr-xr-- 1 fayssal fayssal 0 16:58 22 ميس file1.txt
fayssal@fayssal-VirtualBox:~/D1$
```

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

In Linux, file permissions determine who can access and modify files and directories. There are three types of permissions that can be set for each file or directory:

1. Read permission (r): allows a user to read the contents of a file or view the names of files in a directory.
2. Write permission (w): allows a user to modify the contents of a file or create, delete, and rename files in a directory.
3. Execute permission (x): allows a user to execute a file as a program or access the contents of a directory.

There are three types of users who can be assigned permissions to a file or directory:


1. Owner: the user who created the file or directory.
2. Group: a collection of users who share common permissions.
3. Others: any user who is not the owner or a member of the group.

Each permission type is assigned a numerical value as follows:

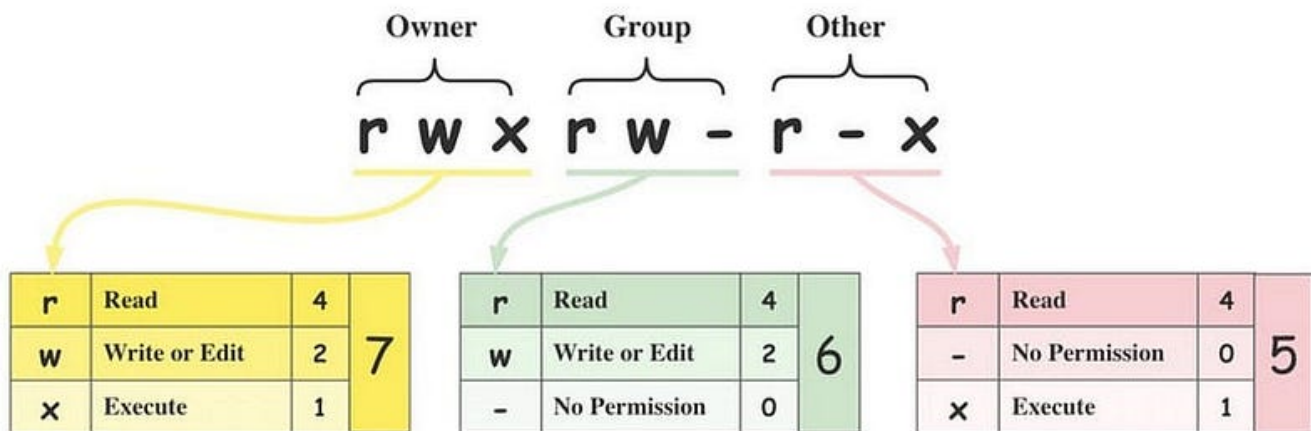
- Read permission (r): 4
- Write permission (w): 2
- Execute permission (x): 1

The sum of these values is used to represent the permission level for a user or group. For example, if a user has read and write permission but not execute permission, their permission level would be 6 (4+2).

Linux File Permissions

 blog.bytebytego.com

Binary	Octal	String Representation	Permissions
000	0 (0+0+0)	---	No Permission
001	1 (0+0+1)	--x	Execute
010	2 (0+2+0)	-w-	Write
011	3 (0+2+1)	-wx	Write + Execute
100	4 (4+0+0)	r--	Read
101	5 (4+0+1)	r-x	Read + Execute
110	6 (4+2+0)	rw-	Read + Write
111	7 (4+2+1)	rwX	Read + Write + Execute



To view and modify file permissions in Linux, you can use the `chmod` command. For example, to give the owner read, write, and execute permission, but only give others read permission, you can use the following command:

```
chmod 754 filename
```

Here, the first digit (7) represents the permission level for the owner, the second digit (5) represents the permission level for the group, and the third digit (4) represents the permission level for others.

3. Read about ACL and try out the commands `getfacl` and `setfacl`

ACL stands for Access Control Lists, which is a way of defining permissions on files and directories in Unix-like operating systems, including Linux. ACLs allow for more fine-grained control over file and directory permissions than traditional Unix permissions, which only provide three levels of permission (read, write, execute) for three categories of users (owner, group, others).

The `getfacl` command is used to view the ACLs of a file or directory, while the `setfacl` command is used to modify them. Here are some examples of how to use these commands:

To view the ACLs of a file, use the `getfacl` command followed by the filename:

```
getfacl filename
```

This will output a list of the permissions granted to each user or group.

To set ACLs on a file or directory, use the `setfacl` command followed by the filename and the options specifying the desired permissions:

```
setfacl -m u:user1:rwX filename
```

This command will grant the user `user1` read, write, and execute permissions on the file `filename`.

To remove a user's permission from a file, use the `-x` option:

```
setfacl -x u:user1 filename
```

This command will remove the user `user1` 's permissions from the file `filename` .

ACLs can be a powerful tool for managing file permissions in a Linux environment, but they can also be complex and difficult to manage. It's important to have a solid understanding of how they work before implementing them in a production environment.

If this post was helpful, please do follow and click the clap 🖐️ button below to show your support 😊

_ Thank you for reading ❤️

_Fayssal 👍

[Linux](#)[Ubuntu](#)[Acl](#)[DevOps](#)[AWS](#)