

# File permissions in Linux

## Project description

[Describe what you accomplish through Linux commands.]

## Check file and directory details

The following code demonstrates how I used Linux commands to determine the existing permissions set for a specific directory in the file system.

```
researcher2@84051c957b90:/home$ cd /home/researcher2/projects
researcher2@84051c957b90:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Nov 23 20:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Nov 23 21:22 ..
-r--r---- 1 researcher2 research_team 46 Nov 23 20:31 .project_x.txt
drwx----- 2 researcher2 research_team 4096 Nov 23 20:31 drafts
-rw-rw-r-- 1 researcher2 research_team 46 Nov 23 20:31 project_k.txt
-rw----- 1 researcher2 research_team 46 Nov 23 20:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team 46 Nov 23 20:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team 46 Nov 23 20:31 project_t.txt
researcher2@84051c957b90:~/projects$
```

## Describe the permissions string

The 10-character string can be deconstructed to determine who is authorized to access the file and their specific permissions. The characters and what they represent are as follows:

- **1st character:** This character is either a `d` or hyphen (`-`) and indicates the file type. If it's a `d`, it's a directory. If it's a hyphen (`-`), it's a regular file.
- **2nd-4th characters:** These characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for the user. When one of these characters is a hyphen (`-`) instead, it indicates that this permission is not granted to the user.
- **5th-7th characters:** These characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for the group. When one of these characters is a hyphen (`-`) instead, it indicates that this permission is not granted for the group.
- **8th-10th characters:** These characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for other. This owner type consists of all other users on the system apart from the user and the group. When one of these characters is a hyphen (`-`) instead, that indicates that this permission is not granted for other.

For example, the file permissions for `project_t.txt` are `-rw-rw-r--`. Since the first character is a hyphen (`-`), this indicates that `project_t.txt` is a file, not a directory. The second, fifth, and eighth characters are all `r`, which indicates that user, group, and other all have read permissions. The third and sixth characters are `w`, which indicates that only the user and group have write permissions. No one has execute permissions for `project_t.txt`.

## Change file permissions

The organization requested that Others and Group should be allowed to read all files. The following shows how I did that using linux commands.

```
researcher2@84051c957b90:~/projects$ chmod g+r,o+r project_m.txt
researcher2@84051c957b90:~/projects$ chmod o+r .project_x.txt
researcher2@84051c957b90:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Nov 23 20:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Nov 23 21:22 ..
-r--r--r-- 1 researcher2 research_team 46 Nov 23 20:31 .project_x.txt
drwx----- 2 researcher2 research_team 4096 Nov 23 20:31 drafts
-rw-rw-r-- 1 researcher2 research_team 46 Nov 23 20:31 project_k.txt
-rw-r--r-- 1 researcher2 research_team 46 Nov 23 20:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team 46 Nov 23 20:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team 46 Nov 23 20:31 project_t.txt
researcher2@84051c957b90:~/projects$
```

## Change file permissions on a hidden file

The organization recently archived **.project\_x.txt** and wants to ensure only User has read permissions. I used the following linux commands to follow those requests.

```
researcher2@84051c957b90:~/projects$ chmod g-r,o-r .project_x.txt
researcher2@84051c957b90:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Nov 23 20:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Nov 23 21:22 ..
-r----- 1 researcher2 research_team 46 Nov 23 20:31 .project_x.txt
drwx----- 2 researcher2 research_team 4096 Nov 23 20:31 drafts
-rw-rw-r-- 1 researcher2 research_team 46 Nov 23 20:31 project_k.txt
-rw-r--r-- 1 researcher2 research_team 46 Nov 23 20:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team 46 Nov 23 20:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team 46 Nov 23 20:31 project_t.txt
researcher2@84051c957b90:~/projects$
```

Change

## directory permissions

The organization requested that the permission for the directory 'drafts' be changed to add read permissions for group and others.

```
researcher2@84051c957b90:~/projects$ sudo chmod g+r,o+r drafts/
researcher2@84051c957b90:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Nov 23 20:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Nov 23 21:22 ..
-r----- 1 researcher2 research_team 46 Nov 23 20:31 .project_x.txt
drwxr--r-- 2 researcher2 research_team 4096 Nov 23 20:31 drafts
-rw-rw-r-- 1 researcher2 research_team 46 Nov 23 20:31 project_k.txt
-rw-r--r-- 1 researcher2 research_team 46 Nov 23 20:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team 46 Nov 23 20:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team 46 Nov 23 20:31 project_t.txt
researcher2@84051c957b90:~/projects$
```

## Summary

I changed multiple permissions to match the level of authorization my organization wanted for files and directories in the 'projects' directory. The first step in this was using **ls -la** to check the permissions for the directory. This informed my decisions in the following steps. I then used the **chmod** command multiple times to change the permissions on files and directories.