

# File permissions in Linux

## Project description

The research team at my organization needs to update the file permissions for certain files and directories within the projects directory. The permissions do not currently reflect the level of authorization that should be given. Checking and updating these permissions will help keep their system secure. To complete this task, I performed the following tasks:

## Checking file and directory details

- **ls** : Displays the names of files in directories in the current working directory.
- **ls -a**: Displays hidden files. Hidden files start with a period (.) at the beginning.
- **ls -l**: Displays permissions to files and directories. Also displays other additional information, including owner name, group, file size, and the time of last modification.
- **ls -la**: Displays permissions to files and directories, including hidden files. This is a combination of the other two options.

```
researcher2@d1b1330b40db:~$ ls
projects
researcher2@d1b1330b40db:~$ cd projects
researcher2@d1b1330b40db:~/projects$ ls
drafts project_k.txt project_m.txt project_r.txt project_t.txt
researcher2@d1b1330b40db:~/projects$ ls -a
. .project_x.txt project_k.txt project_r.txt
.. drafts project_m.txt project_t.txt
researcher2@d1b1330b40db:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug 30 14:00 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug 30 14:17 ..
-rw----- 1 researcher2 research_team  46 Aug 30 14:00 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug 30 14:00 drafts
-rw-rw-rw- 1 researcher2 research_team  46 Aug 30 14:00 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Aug 30 14:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Aug 30 14:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Aug 30 14:00 project_t.txt
```

## Description of the permissions string

Each character in the 10-character string conveys different information about these permissions. The following table describes the purpose of each character:

Character	Example	Meaning
1st	<b>d</b> rw-rw-rw-	file type <ul style="list-style-type: none"><li>• <b>d</b> for directory</li><li>• - for a regular file</li></ul>
2nd	d <b>r</b> w-rw-rw-	read permissions for the user <ul style="list-style-type: none"><li>• <b>r</b> if the user has read permissions</li><li>• - if the user lacks read permissions</li></ul>
3rd	d <b>r</b> <b>w</b> -rw-rw-	write permissions for the user <ul style="list-style-type: none"><li>• <b>w</b> if the user has write permissions</li><li>• - if the user lacks write permissions</li></ul>
4th	d <b>r</b> <b>w</b> <b>x</b> -rw-rw-	execute permissions for the user <ul style="list-style-type: none"><li>• <b>x</b> if the user has execute permissions</li><li>• - if the user lacks execute permissions</li></ul>
5th	d <b>r</b> <b>w</b> -r <b>w</b> -rw-	read permissions for the group <ul style="list-style-type: none"><li>• <b>r</b> if the group has read permissions</li><li>• - if the group lacks read permissions</li></ul>
6th	d <b>r</b> <b>w</b> -r <b>w</b> <b>w</b> -rw-	write permissions for the group <ul style="list-style-type: none"><li>• <b>w</b> if the group has write permissions</li><li>• - if the group lacks write permissions</li></ul>
7th	d <b>r</b> <b>w</b> -r <b>w</b> <b>w</b> <b>x</b> -rw-	execute permissions for the group <ul style="list-style-type: none"><li>• <b>x</b> if the group has execute permissions</li><li>• - if the group lacks execute permissions</li></ul>

8th	drwxrwxrwx	read permissions for other <ul style="list-style-type: none"> <li>• <b>r</b> if the other owner type has read permissions</li> <li>• - if the other owner type lacks read permissions</li> </ul>
9th	drwxrwxrwx	write permissions for other <ul style="list-style-type: none"> <li>• <b>w</b> if the other owner type has write permissions</li> <li>• - if the other owner type lacks write permissions</li> </ul>
10th	drwxrwxrwx	execute permissions for other <ul style="list-style-type: none"> <li>• <b>x</b> if the other owner type has execute permissions</li> <li>• - if the other owner type lacks execute permissions</li> </ul>

## Changing file permissions

The organization determined that other shouldn't have write access to any of their files. To comply with this, I referred to the file permissions that I previously returned. I determined project\_k.txt must have the write access removed for other.

The **chmod** command changes permissions on files and directories.

Syntax: **chmod u/g/o "+" / "-" file\_name** // chmod u+w file.txt (will give write permission for user)

The following table reviews how each character is used within the first argument of **chmod**:

Character	Description
<b>u</b>	indicates changes will be made to user permissions
<b>g</b>	indicates changes will be made to group permissions
<b>o</b>	indicates changes will be made to other permissions
<b>+</b>	adds permissions to the user, group, or other
<b>-</b>	removes permissions from the user, group, or other
<b>=</b>	assigns permissions for the user, group, or other

```
researcher2@ed841b71d081:~/projects$ chmod o-w project_k.txt
researcher2@ed841b71d081:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Aug 30 11:08 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Aug 30 11:08 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Aug 30 11:08 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Aug 30 11:08 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Aug 30 11:08 project_t.txt
researcher2@ed841b71d081:~/projects$
```

Here, we removed the write permission of “other” in project\_k.

## Changing file permissions on a hidden file

The research team at my organization recently archived project\_x.txt. They do not want anyone to have write access to this project, but the user and group should have read access.

We can use the `ls -la` command to list permissions of files and directories including hidden ones.

```
researcher2@d1b1330b40db:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@d1b1330b40db:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug 30 14:00 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug 30 14:17 ..
-r--r----- 1 researcher2 research_team  46 Aug 30 14:00 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug 30 14:00 drafts
-rw-rw-rw- 1 researcher2 research_team  46 Aug 30 14:00 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Aug 30 14:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Aug 30 14:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Aug 30 14:00 project_t.txt
```

The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. I know .project\_x.txt is a hidden file because it starts with a period (.). In this example, I removed write permissions from the user and group, and added read permissions to the group. I removed write permissions from the user with u-w. Then, I removed write permissions from the group with g-w, and added read permissions to the group with g+r.

## Changing directory permissions

My organization only wants the researcher2 user to have access to the drafts directory and its contents. This means that no one other than researcher2 should have execute permissions.

Changing the permissions of a directory is similar to that of changing a file, just input directory name in the place of file name.

Syntax: `chmod u/g/o "+"/"-" directory_name`

```
researcher2@ed841b71d081:~/projects$ chmod g-x drafts
researcher2@ed841b71d081:~/projects$ ls -l
total 20
drwx----- 2 researcher2 research_team 4096 Aug 30 11:08 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Aug 30 11:08 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Aug 30 11:08 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Aug 30 11:08 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Aug 30 11:08 project_t.txt
```

Line 4 is the directory (drafts) with restricted permissions. Here you can see that only researcher2 has execute permissions. It was previously determined that the group had execute permissions, so I used the chmod command to remove them. The researcher2 user already had execute permissions, so they did not need to be added.

## 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 including hidden files. This informed my decisions in the following steps. I then used the `chmod` command multiple times to change the permissions on files and directories.