# 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 steps:

## Check file and directory details

To list file, hidden files and directory details, I used the following command:



This command displays detailed information about files, including:

* File type
* Permissions
* Number of links
* Owner
* Group
* File size
* Last modification time
* File or directory name

The output of my command indicates that there is one directory named drafts, one hidden file named .project\_x.txt, and five other project files. The 10-character string in the first column represents the permissions set on each file or directory.

## 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. A hyphen (-) indicates that the permission is not granted.
* **5th-7th characters:** These characters indicate the read (r), write (w), and execute (x) permissions for the group. A hyphen (-) indicates that the permission is not granted.
* **8th-10th characters:** These characters indicate the read (r), write (w), and execute (x) permissions for others (all other users on the system). A hyphen (-) indicates that the permission is not granted.

#### **Example: project\_t.txt**

For the file project\_t.txt, the permissions string is -rw-rw-r--. This can be explained as:

* -: Indicates a regular file.
* rw-: The owner has read and write permissions.
* rw-: The group has read and write permissions.
* r--: Others have read-only permissions.

## Change file permissions

The organization determined that others should not have write access to any files. To comply with this, I referred to the file permissions and identified that project\_k.txt had write access for others.

To remove this access, I used the following command:



After running this command, I verified the changes using ls -la. The updated permissions for project\_k.txt are now:

* User: rw-
* Group: rw-
* Others: r--

## Change file permissions on a hidden file

The research team recently archived .project\_x.txt. This file should not have write permissions for anyone, but the user and group should have read access.

To achieve this, I used the following command:



This sets the permissions to:

* User: r--
* Group: r--
* Others: ---

I verified the updated permissions using ls -la. The output confirmed that .project\_x.txt is now read-only for the user and group, with no access for others.

## Change directory permissions

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

To remove execute permissions for the group, I used the following command:



The updated permissions for drafts are now:

* User: rwx
* Group: ---
* Others: ---

## 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 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 update permissions for specific files and directories, ensuring proper access control and system security.