# File permissions in Linux

## Project description

The research team at my organization needs to update the file permissions for selected files and directories within the projects directory. The permissions don’t currently show the level of authorization that should be permitted. Checking and updating these permissions will help keep the system up to date and secure. To do this, I performed the following tasks:

## Check file and directory details

Inputting the following code demonstrates how I was able to use the Linux commands to determine existing permissions set for a specific directory in the file system.



First line of the screenshot shows the command I entered, and the other lines display the output of the entered command. Code lists all contents of the projects directory. Using the ls command with the -la option to display a detailed list of file contents and hidden files. Output of my command indicated that there is one directory named drafts, as well as 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 authorization of file access and the specific permissions. The characters and what they mean are as follows:

* **1st character:** This character is either a d or hyphen (-) and indicates the file type. If its 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-10 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- -. 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 team determined that the other portion shouldn’t have write access to any of their files. To apply this, I referred to the file permissions that I previously inputted. I had determined that project\_k.txt must have the write access removed for other.

The following code demonstrates how the Linux command was used to do this:



The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. The chmod command changes the permissions on files and directories. The first argument indicates what permissions should be changed, and the second argument specifies the file or directory. In this example, write permissions from other for the project\_k.txt file were removed. After this, ls -la command was to used to review the updates made.

## Change 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.

The following code demonstrates how I used Linux commands to change the following permissions:



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.

## Change 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.

The following code demonstrates how I used Linux commands to change the permissions:



The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. I 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 the 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.