# File permissions in Linux

## Project description

## To address the discrepancy in file permissions within the projects directory, the research team at my organization needs to update them to reflect the appropriate level of authorization. Ensuring these permissions accurately reflect the required access levels is crucial for maintaining system security. To accomplish this task, I conducted the following actions:

## 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.



## The first line of the screenshot shows the command I used, while the subsequent lines display the output. The command lists all contents of the projects directory. I utilized the ls command with the -la option to generate a detailed listing of the file contents, including hidden files. The output reveals one directory named drafts, one hidden file named .project\_x.txt, and five additional 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 has access to the file and their specific permissions. Here's what each character represents:

- 1st character: Indicates the file type. "d" signifies a directory, while "-" signifies a regular file.

- 2nd-4th characters: Denote read (r), write (w), and execute (x) permissions for the user. A "-" instead of a letter indicates the absence of that permission.

- 5th-7th characters: Represent read (r), write (w), and execute (x) permissions for the group. A "-" denotes the lack of that permission.

- 8th-10th characters: Indicate read (r), write (w), and execute (x) permissions for others. A "-" indicates the absence of that permission for other users on the system apart from the owner and group.

## For instance, let's take the file permissions for project\_t.txt: -rw-rw-r--. The first character, a hyphen (-), indicates that project\_t.txt is a file, not a directory. The second, fifth, and eighth characters are all "r," indicating that the user, group, and others have read permissions. The third and sixth characters are "w," indicating that only the user and group have write permissions. No one has execute permissions for project\_t.txt.

## Change file permissions

After reviewing the file permissions, I retrieved earlier, I identified project\_k.txt as needing modification. Specifically, I determined that the write access for others should be removed from this file to comply with the organization's security policies.

The following code demonstrates how I used Linux commands to do this:



## The first two lines of the screenshot show the commands I entered, and the subsequent lines display the output of the second command. The chmod command is used to modify permissions on files and directories. The first argument specifies the permissions to be changed, while the second argument specifies the file or directory. In this example, I removed write permissions from others for the project\_k.txt file. Following this, I used ls -la to review the updates I 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 permissions:



## The first two lines of the screenshot show the commands I entered, and the subsequent lines display the output of the second command. Recognizing that .project\_x.txt is a hidden file because it begins with a period (.). In this example, I removed write permissions from both the user and group while adding read permissions to the group. Specifically, I removed write permissions from the user with u-w. Then, I removed write permissions from the group with g-w, and finally, I 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 output here presents the permission listing for various files and directories. Line 1 denotes the current directory (projects), while line 2 represents the parent directory (home). Line 3 shows a regular file titled .project\_x.txt. Line 4 displays the directory (drafts) with restricted permissions, where only researcher2 has execute permissions. It was previously established that the group had execute permissions, so I utilized the chmod command to remove them. As researcher2 already possessed execute permissions, there was no need to add them.

## 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.