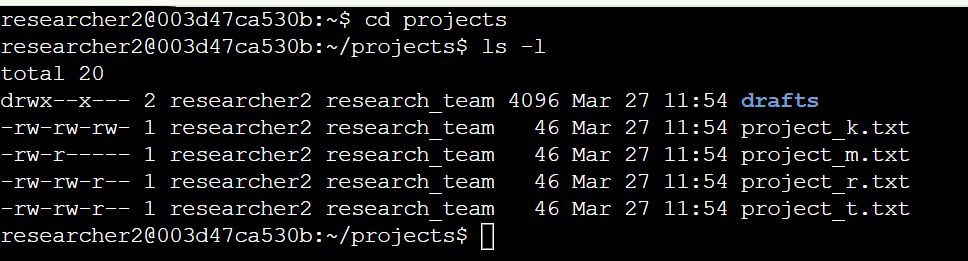
File permissions in Linux

# File permissions in Linux

## Project description

The project is to work for a research team in an organization. My work is to ensure that users have authorized permissions to access files and examine existing permissions to the file system. If the user has rights to access the file, I need to give permission for authorized users, and if they don't have rights to access then need to remove the authorization for that user. Checking and updating these permissions will help keep their system secure. To complete this task, i performed the following tasks:

## Check file and directory details



## The first line of the screenshot shows the command I entered, while the following lines display its output. The command lists all the contents of the **projects** directory. I used the **ls** command with the **-la** option to generate a detailed listing, which also includes hidden files. The output reveals that the directory contains one folder named **drafts**, a hidden file called **.project\_x.txt**, and five other project files. The 10-character string in the first column represents the permission settings for each file or directory.

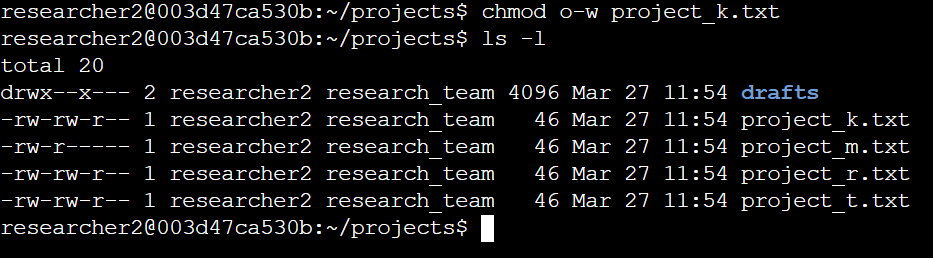
## Describe the permissions string

* The 1st character indicates the file type. The d indicates it’s a directory. When this character is a hyphen (-), it's a regular file.
* The 2nd-4th 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.
* The 5th-7th 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.
* The 8th-10th characters indicate the read (r), write (w), and execute (x) permissions for the owner type of 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 others.

The file permissions for project\_t.txt are -rw-rw-r--. The first character, a hyphen (-), indicates that project\_t.txt is a regular file and not a directory. The next three characters (rw-) represent the permissions for the file owner, where r (read) allows the owner to view the file's contents, w (write) permits modifications, and the absence of x (execute) means the owner cannot execute the file. Similarly, the next three characters (rw-) define the group permissions, granting read and write access to members of the group but not execution rights. The last three characters (r--) specify the permissions for others, allowing only read access while restricting both writing and execution. Since there are no execute permissions assigned to any user category, no one can execute this file.

## Change file permissions

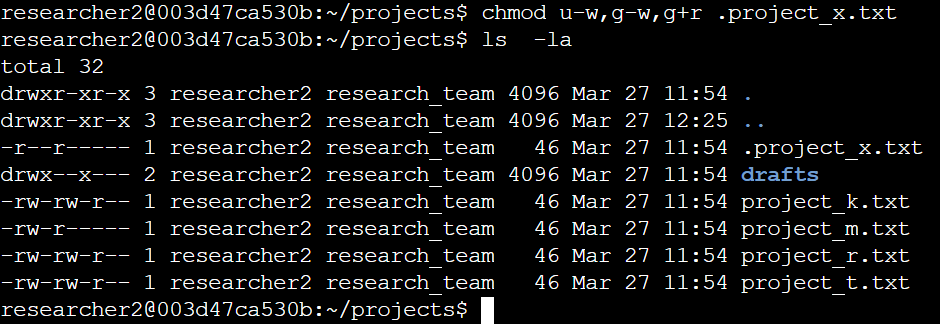
The first two lines of the screenshot show the commands I entered, while the remaining lines display the output of the second command. The chmod command is used to modify file and directory permissions. The first argument defines the permission changes, and the second argument specifies the target file or directory. In this example, I removed write permissions for others on the project\_k.txt file. After making this change, I used ls -la to verify the updated permissions.



## Change file permissions on a hidden file

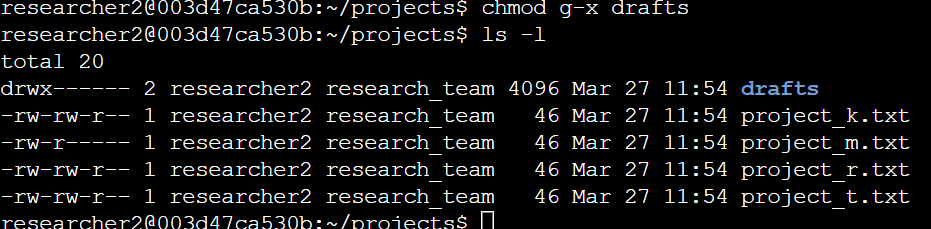
The research team at my organization recently archived project\_x.txt. They do not want anyone to have write access to this project, but the user and group should have read access.

The first two lines of the screenshot show the commands I entered, while the remaining lines display the output of the second command. The file .project\_x.txt is a hidden file, as its name begins with a period (.). In this example, I modified its permissions by removing write access for both the user and group, while granting read access to the group. First, I removed the user's write permission using u-w. Then, I removed the group's write permission with g-w and added read access for the group using 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 first two lines of the screenshot show the commands I entered, while the remaining lines display the output of the second command. Earlier, I identified that the **group** had execute permissions, so I used the **chmod** command to remove them. The **researcher2** user already had execute permissions, so no changes were needed for that user.

## Summary

I modified multiple permissions to align with the authorization levels required by my organization for files and directories within the **projects** directory. To begin, I used the **ls -la** command to review the existing permissions, which helped guide my decisions in the next steps. Then, I executed the **chmod** command multiple times to adjust the permissions on specific files and directories as needed.