# 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 is required. Checking and updating these permissions will help the system remain secure.

## Check file and directory details

The following screenshot demonstrates how I utilized Linux commands to determine the existing permission set for the directory in the file system.



The screenshot showcases all the contents in the “projects” directory. The Linux command shown in the first line [ ls -la ] displayed a detailed listing of the file contents that also showcased a hidden file. Amongst the listings, only one directory appeared named “drafts” and one hidden file named “.project\_x”. Additionally, the 10-character string represents the permissions set for each file or directory shown.

## Describe the permissions string

The 10-character string can be deconstructed to determine who is authorized to access each particular file and their associated permissions. The breakdown is as follows:

* **1st character** : is shown as either a “-” or a “d”. The “-” indicates a regular file, while the “d” indicates a directory.
* **2nd-4th characters**: are shown as either “-”, “r”, “w”, “x”, with each of these permissions belonging to the **user class**. The “-” indicates the lack of permissions to the user, while the “r” indicates the ability for the user to read. Additionally, the “w” indicates that the user can write to the file. Finally, the “x” indicates that the user can execute the file (if applicable).
* **5th-7th characters**: follows the same parameters as the previous 3 characters, but the permissions belong to the **group class**. This group is usually limited in size, but tends to either have matching permissions to the user or the second highest.
* **8th-10th characters**: follows the same parameters as the previous 3 characters, but the permissions belong to the **other class**. This group tends to have the lowest amount of permissions, if any are granted at all.

## Change file permissions

The organization determined that the **other class** should not have write access to any of their files. The only file that needed to be changed was the permissions related to the file: “project\_k.txt”.



The screenshot shows the updated permissions after executing the command in the top line. The command “chmod” is the editing command for permissions in Linux with the “o-w” specifying that the write command “w” should be taken from the **other class** for permissions. The command “ls -la” shows the updated permissions.

## Change file permissions on a hidden file

The research team at my organization recently archived “project\_x.txt”. They want no write access to this project, however, they want both the **user** and **group** **classes** to have read access. Both the **user class** and **group class** need to be modified.



Again, utilizing the “chmod” command, the permissions for both the **user class** and the **group class** were revoked with their respective commands “u-w” and “g-w”, while read access was added to the group class with the command “g+r”. Again, the updated permissions are shown with “ls -la”.

## Change directory permissions

My organization only wants the “researcher 2” user to have access to the “drafts” directory and its content. The **group class** needs to have their executable permission revoked.



Following similar methods, “chmod g-x” is utilized to take permissions away from the **group class** with only permissions existing for the **user class**. Once again, “ls -la” is used to show the updated permissions.

## Summary

I changed multiple permissions to match the desired level of authorization that my organization wanted for all the files and directories within the “projects” directory. My decisions were informed from the original permissions outlined in the first “ls -la” command and checked with each subsequent one after the implementation of my “chmod” commands to refer back to the desired outcomes.