# File permissions in Linux

## Project description

This project focuses on managing file and directory permissions in a Linux system. The goal is to ensure that the correct permissions are granted or revoked to authorized users, preventing unauthorized access and maintaining system security. Through the use of Linux commands, I examined, modified, and documented the permissions for several files and directories within a research team's working environment.

## Check file and directory details

To check the file and directory details in Linux, I used the `ls -la` command, which displays a detailed list of files and their permissions, including hidden files. The command output includes file names, permission strings, owner, group, file size, and the last modified date.

Here’s an example of the command I used:

|  |
| --- |
| ls -la |

Permissions for the files in the `/home/researcher2/projects` directory were as follows:

|  |
| --- |
| -rw-rw-rw- project\_k.txt  -rw-r----- project\_m.txt  -rw-rw-r-- project\_r.txt  -rw-rw-r-- project\_t.txt  -rw--w---- .project\_x.txt  drwx--x--- drafts |

## Describe the permissions string

The 10-character permission string follows this format: `rwxrwxrwx`, where `rwx` stands for read (r), write (w), and execute (x) permissions for user, group, and others, respectively. Here’s a breakdown of the permission string components:

* **User (owner)**: First set of three characters
* **Group**:Second set of three characters
* **Others**: Final set of three characters

For example, the file `project\_k.txt` has a permission string `-rw-rw-rw-`. This means:

* User has read and write permissions (`rw-`)
* Group has read and write permissions (`rw-`)
* Others have read and write permissions (`rw-`)

## Change file permissions

In the lab, I modified the permissions to secure the files by removing write access for others and restricting group access. I used the `chmod` command to achieve this. For instance:

* To remove write access for others on `project\_k.txt`, I ran the command:

|  |
| --- |
| chmod o-w project\_k.txt |

* I also removed read permissions for the group on `project\_m.txt` using:

|  |
| --- |
| chmod g-r project\_m.txt |

## Change file permissions on a hidden file

I updated the permissions of the hidden file `.project\_x.txt` to make it read-only for the user and provide read permissions to the group using the following command:

|  |
| --- |
| chmod u-w,g=r .project\_x.txt |

This command ensured that the owner could only read the file, while the group had read access. The resulting permission string for `.project\_x.txt` was `-r--r-----`.

## Change directory permissions

To modify the permissions for the `drafts` directory, I removed the group’s execute permissions. This was done to restrict the group from accessing the directory’s contents. I used the following command:

|  |
| --- |
| chmod g-x drafts |

After running this command, the permission string for the `drafts` directory changed from `drwx--x---` to `drwx------`.

## Summary

In this activity, I utilized various Linux commands to check and manage file permissions in a secure way. I reviewed the current permissions using `ls -la` and made appropriate changes using the `chmod` command to restrict unauthorized access. These actions help maintain the security of sensitive files and directories in a multi-user environment, demonstrating the importance of proper file permission management in Linux systems.