File permissions in Linux

Project description

Inside the /home/researcher2/projects directory I have changed the file and directories permissions found to what was asked to. Using the chmod and ls -la commands i was able to accomplish the tasks.

Check file and directory details

Move into the specified directory: cd /home/researcher2/projects
Check permissions: ls -la

Describe the permissions string

project_k.txt : -rw-rw-rw-
project_m.txt : -rw-r-----
project_r.txt : -rw-rw-r--
project_t.txt : -rw-rw-r--
project_k.txt : -rw--w----
drafts: drwx--x---
The first character shows if it is a directory, (drafts has d as it is a directory). The next 3 characters show the user permissions, in this case it can read, write and execute files. The next 3 is for the group, in this case they can only execute files inside the directory. Finally, the filan 3 character shows the permissions for other user groups, in this case they all are -, which means they don't have any permission in this directory.

Change file permissions

Inside the /home/researcher2/projects directory run the following command: chmod o-r ./*

Change file permissions on a hidden file

chmod u-w,g-r,g+r .project_x.txt

Change directory permissions

chmod u+rwx, g-rwx, o-rwx ./drafts

Summary

In this activity, I explored and managed file permissions within the `/home/researcher2/projects` directory to align with specific access requirements. Using the `ls -la` command, I reviewed the 10-character permission strings for each file and directory, gaining insight into how Linux distinguishes permissions for users, groups, and others. The command output revealed the read, write, and execute privileges, along with special considerations for hidden files such as `.project_x.txt`.

To enforce proper access controls, I utilized the `chmod` command to make targeted changes:

- I removed read access for others on all regular files with: `chmod o-r ./*`
- I modified hidden file permissions using: `chmod u-w,g-r,g+r .project_x.txt`
- I secured the `drafts` directory by restricting access to the user only with: `chmod u+rwx,g-rwx,o-rwx ./drafts`

These commands ensured compliance with the principle of least privilege—granting access only to those who need it. The resulting permission strings confirmed the intended configurations, such as `drwx--x---` for the `drafts` directory. Overall, this exercise demonstrated how `chmod` and `ls -la` work together to manage and audit Linux file system permissions effectively.