File permissions in Linux

Project description

In this project, I explored Linux file permissions and how they are applied to files and directories. By examining the `drafts` directory, I learned how to interpret permission strings and distinguish between owner, group, and others. This activity helped me understand the principle of least privilege and how to manage access control effectively in a Linux environment.

Check file and directory details

Use the code ls -la to display all the files in the directory including the hidden files

```
researcher2@6e8df72e0245:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Sep  5 05:56 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep  5 06:29 ..
-rw--w---- 1 researcher2 research_team   46 Sep  5 05:56 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep  5 05:56 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Sep  5 05:56 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep  5 05:56 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep  5 05:56 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep  5 05:56 project_t.txt
researcher2@6e8df72e0245:~/projects$
```

Describe the permissions string

Let's look at the **drafts** directory.

A typical file permission string looks like this: `drwxrwxrwx`.

- The first character indicates the type:
    - `d` = directory
    - `-` = regular file

The next nine characters represent permissions, divided into three groups:

1. **Owner (user)**
2. **Group**
3. **Others**

Each group has three possible permissions:

- `r` = read
- `w` = write
- `x` = execute

Now, for the **drafts** directory:

- The leading `d` means it is a directory.
- The owner has `rwx`, which means they can read, write, and execute (access) the directory.
- The group has `--x`, meaning they cannot read or write, but they can execute (enter) the directory.
- Others have `---`, meaning they have no permissions at all (no read, no write, no execute).

Change file permissions

The organization does not allow others to have write access to any files. The file project_k.txt has write permission. We used the command “chmod o-w project_k.txt” to remove the write permission to that file

```
researcher2@6e8df72e0245:~/projects$ chmod o-w project_k.txt
researcher2@6e8df72e0245:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Sep  5 05:56 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep  5 06:29 ..
-rw--w---- 1 researcher2 research_team   46 Sep  5 05:56 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep  5 05:56 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Sep  5 05:56 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep  5 05:56 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep  5 05:56 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep  5 05:56 project_t.txt
researcher2@6e8df72e0245:~/projects$ ~
```

The research team has archived **.project_x.txt**, which is why it's a hidden file. This file should not have write permissions for anyone, but the user and group should be able to read the file.

```
researcher2@6e8df72e0245:~/projects$ chmod u-w .project_x.txt
researcher2@6e8df72e0245:~/projects$ chmod g-w .project_x.txt
```

```
researcher2@6e8df72e0245:~/projects$ chmod g+r .project_x.txt
```

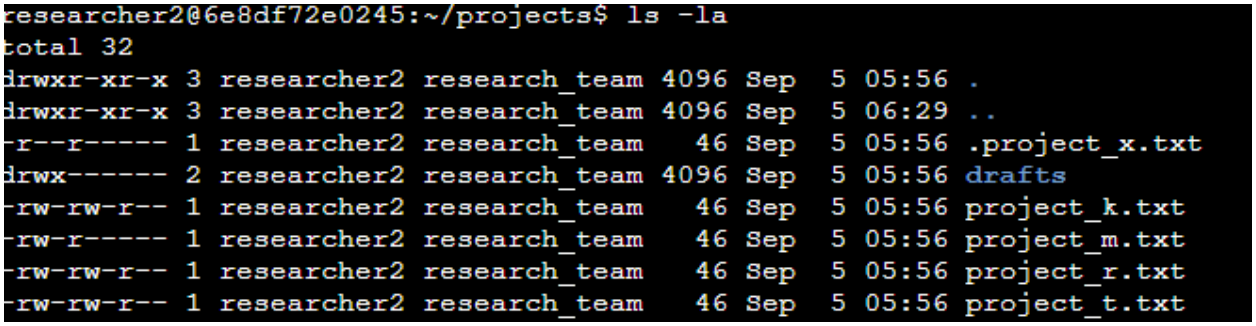
```
researcher2@6e8df72e0245:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Sep  5 05:56 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep  5 06:29 ..
-r--r----- 1 researcher2 research_team   46 Sep  5 05:56 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep  5 05:56 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Sep  5 05:56 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep  5 05:56 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep  5 05:56 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep  5 05:56 project_t.txt
researcher2@6e8df72e0245:~/projects$
```

The command 'chmod u-w' was used to remove the write permission for user
The command 'chmod g-w' was used to remove the write permission for group
The command 'chmod g+r' was used to add the read permission for group
By typing ls - la we can see all the files including the hidden files and directory of the dricetory

Change directory permissions

The files and directories in the projects directory belong to the **researcher2** user. Only **researcher2** should be allowed to access the **drafts** directory and its contents.

```
researcher2@6e8df72e0245:~/projects$ chmod g-x drafts
```



The command ‘chmod g-x’ was used to remove the write permission for group in the drafts directory

By typing ls - la we can see all the files including the hidden files and directory of the dricetory

Summary

In this activity, I reviewed how Linux permissions work and applied that knowledge to analyze the `drafts` directory. I identified the type of file, broke down the permission string, and explained what access the owner, group, and others have. This exercise demonstrated how permissions directly control who can read, write, or execute within a directory, which is essential for maintaining security and proper access in real-world scenarios.