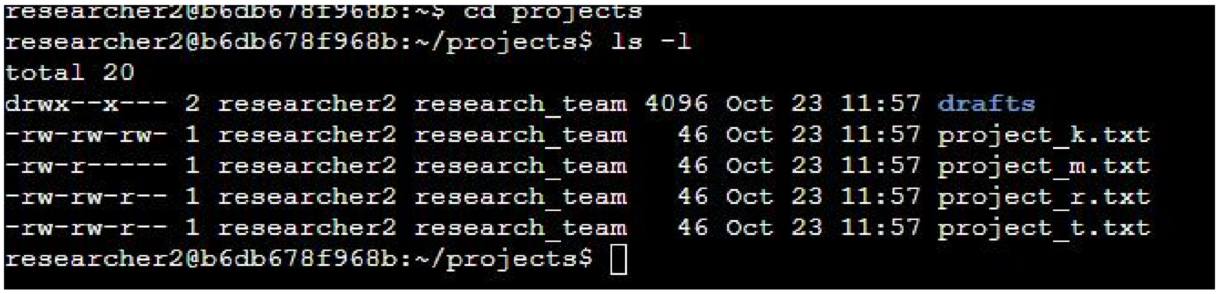
File permissions in Linux

Project description

In this project, I explore how to manage file and directory permissions in Linux. Through the use of various commands, I demonstrate how to check, change, and customize access levels for files and directories, including hidden files. Understanding Linux file permissions is crucial for controlling access to sensitive data and ensuring that only the appropriate users or groups can read, write, or execute specific files.

Check file and directory details

To begin, I checked the file and directory permissions, including hidden files, in the **`/home/researcher2/projects`** directory. By using the ls command with appropriate flags, I can view detailed permission information:



```
researcher2@b6db678f968b:~$ cd projects
researcher2@b6db678f968b:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Oct 23 11:57 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 23 11:57 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 23 11:57 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 23 11:57 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 23 11:57 project_t.txt
researcher2@b6db678f968b:~/projects$ 
```

· The -l flag displays detailed information about each file and directory, including permissions.
· The -a flag lists all files, including hidden ones (files starting with a dot .).

Describe the permissions string

The permissions string, for example **`drwx--xx---`**, provides a breakdown of access rights for the user, group, and others:

1. **File type**: The first character indicates whether it's a directory (d), a file (-), or a symbolic link (l).

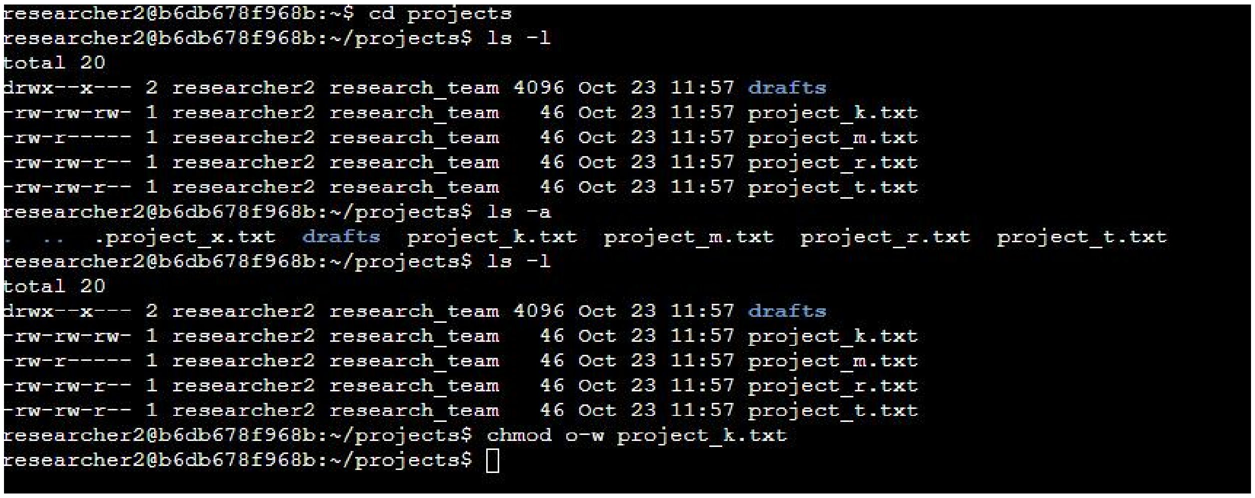
2. **User permissions**: The next three characters refer to the file owner (user):
    1. r (read)
    2. w (write)
    3. x (execute)
3. **Group permissions**: The second set of three characters refers to the group's access rights.
4. **Others' permissions**: The final three characters control access for everyone else.

For example, in **`drwxr-xr-x`:**

- d: Directory.
- rwx: The user can read, write, and execute.
- r-x: The group can read and execute but not write.
- r-x: Others can also read and execute but not write.

Change file permissions

Linux provides the **`chmod`** command to change permissions. The numeric or symbolic representation can be used. For instance, changing a file's permissions so that only the owner can read and write, while the group and others can only read, is done as follows:

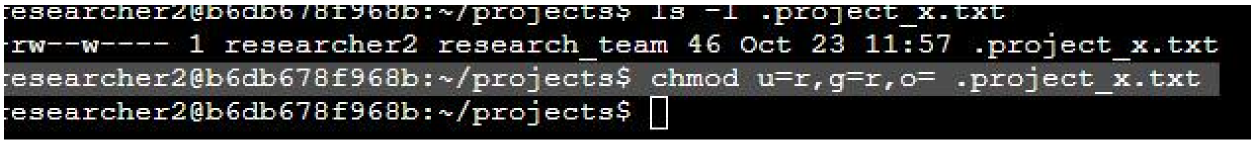


```
researcher2@b6db678f968b:~$ cd projects
researcher2@b6db678f968b:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Oct 23 11:57 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 23 11:57 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 23 11:57 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 23 11:57 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 23 11:57 project_t.txt
researcher2@b6db678f968b:~/projects$ ls -a
.  ..  .project_x.txt  drafts  project_k.txt  project_m.txt  project_r.txt  project_t.txt
researcher2@b6db678f968b:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Oct 23 11:57 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 23 11:57 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 23 11:57 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 23 11:57 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 23 11:57 project_t.txt
researcher2@b6db678f968b:~/projects$ chmod o-w project_k.txt
researcher2@b6db678f968b:~/projects$ 
```

- `u=rw`: User has read and write permissions.
- `g=r`: Group has read-only permission.
- `o=r`: Others have read-only permission.

Change file permissions on a hidden file

To change permissions on a hidden file (e.g., `.project_x.txt`), I was using the same approach as with regular files. For instance, to remove write access for everyone except the file owner:

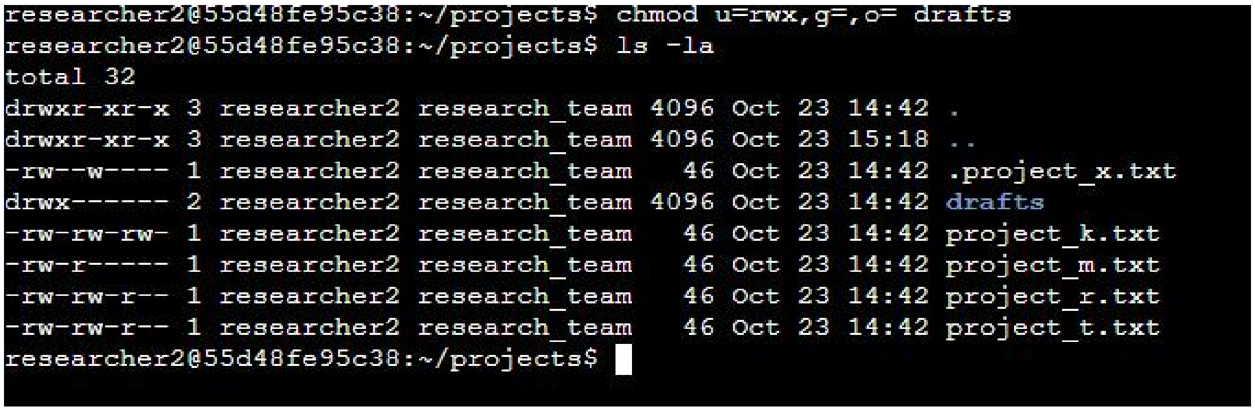


```
esearcher2@b6db678f968b:~/projects$ ls -l .project_x.txt
-rw--w---- 1 researcher2 research_team 46 Oct 23 11:57 .project_x.txt
esearcher2@b6db678f968b:~/projects$ chmod u=r,g=r,o= .project_x.txt
esearcher2@b6db678f968b:~/projects$ 
```

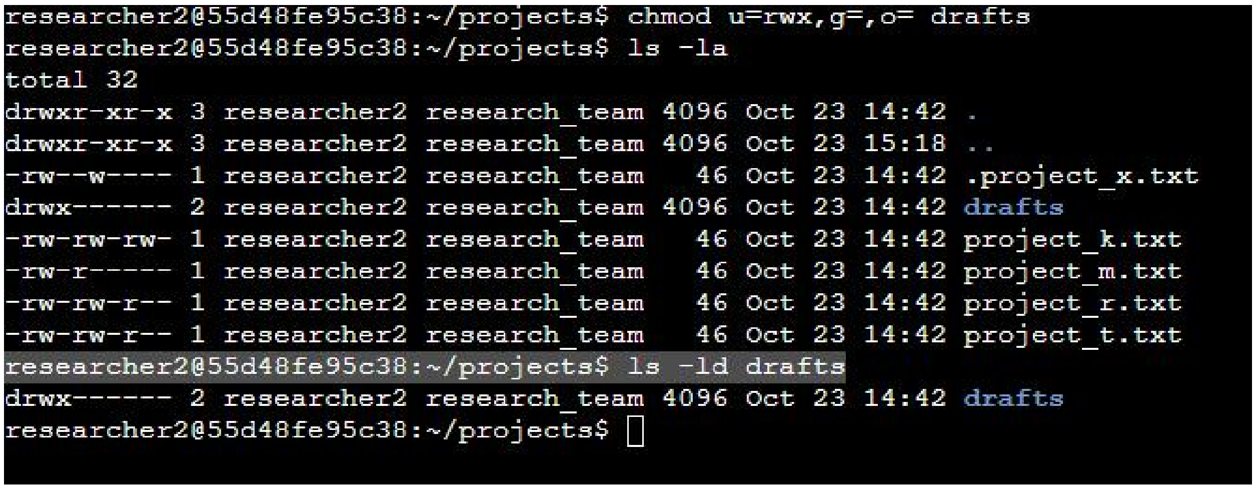
- This sets the file so only the owner can read and write, and the group and others have no permissions.

Change directory permissions

Directory permissions differ slightly in that the execute permission is crucial for allowing users to enter the directory. For example, if I want only the owner **(`researcher2`)** to have full access to the drafts directory, and no one else should have any access, I use:



```
researcher2@55d48fe95c38:~/projects$ chmod u=rwx,g=,o= drafts
researcher2@55d48fe95c38:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 23 14:42 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 23 15:18 ..
-rw--w---- 1 researcher2 research_team   46 Oct 23 14:42 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Oct 23 14:42 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 23 14:42 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 23 14:42 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 23 14:42 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 23 14:42 project_t.txt
researcher2@55d48fe95c38:~/projects$ 
```



```
researcher2@55d48fe95c38:~/projects$ chmod u=rwx,g=,o= drafts
researcher2@55d48fe95c38:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 23 14:42 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 23 15:18 ..
-rw--w---- 1 researcher2 research_team   46 Oct 23 14:42 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Oct 23 14:42 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 23 14:42 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 23 14:42 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 23 14:42 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 23 14:42 project_t.txt
researcher2@55d48fe95c38:~/projects$ ls -ld drafts
drwx------ 2 researcher2 research_team 4096 Oct 23 14:42 drafts
researcher2@55d48fe95c38:~/projects$ 
```

This ensures only **`researcher2`** can enter, view, or modify the drafts directory.

Summary

In this project, I demonstrated how to check and modify file and directory permissions in Linux using the ls, **`chmod`**, and related commands. These skills are essential for managing access control in Linux environments. By learning how to use both symbolic and numeric representations, I ensured precise permission settings, protecting sensitive files and directories from unauthorized access.

Understanding Linux file permissions enables better security practices, especially in multi-user environments, and ensures that the right people have the right level of access to important system resources.