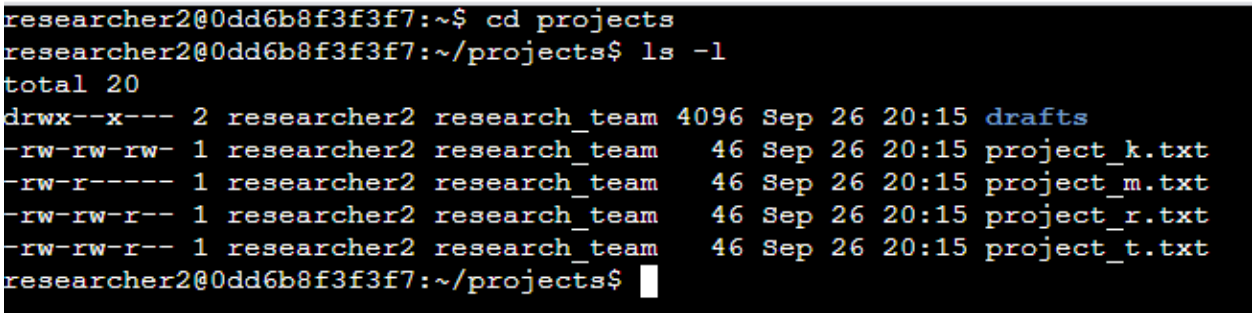
File permissions in Linux

Project description

In Linux, file permissions are crucial for maintaining system security and user access control. They determine what actions a user can perform on a file or directory, such as reading, writing, or executing. Permissions are defined for three types of users: the owner, the group, and others. With Linux commands, you can change these permissions here are some examples of what you can accomplish:

Check file and directory details

This is an example that demonstrates how I used Linux commands to access the permissions of the files in the `projects` directory:



The first command i used is the " cd projects" which moves the user in this case, me to the "projects" directory
The next command i used is the "ls -l" which lists files with details like permissions, owner, group, size, & modification date
Several files were then listed, each with different permissions indicating who can read, write, or execute them.

Describe the permissions string

a 10-character string begins each entry and indicates how the permissions on the file are set. For instance, a directory with full permissions for all owner types would be drwxrwxrwx.

The permissions string in this scenario follows the Linux file permission format. Here's a breakdown:

First character: Indicates the type of file. d means directory, - means file.

Next three characters: User (owner) permissions. r for read, w for write, x for execute.
Next three characters: Group permissions. Same r, w, x format.
Last three characters: Others (everyone else) permissions. Same r, w, x format.

drwxr-x—:
d: Directory
rwx: Owner has read, write, and execute permissions
r-x: Group has read and execute permissions
—: Others have no permissions
-rw-rw-r–:
-: Regular file
rw-: Owner has read and write permissions
rw-: Group has read and write permissions
r–: Others have read-only permission
-rw-r-----:
-: Regular file
rw-: Owner has read and write permissions
r–: Group has read-only permission
—: Others have no permissions
These permissions control who can read, write, or execute the files and directories.

Change file permissions

This is an example of how I used Linux to change file permissions. In this scenario none of the files should allow the other users to write to files

```
drwx--x--- 2 researcher2 research_team 4096 Sep 26 20:15 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Sep 26 20:15 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep 26 20:15 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 26 20:15 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 26 20:15 project_t.txt
researcher2@0dd6b8f3f3f7:~/projects$ chmod o-w project_k.txt
researcher2@0dd6b8f3f3f7:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Sep 26 20:15 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Sep 26 20:15 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep 26 20:15 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 26 20:15 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 26 20:15 project_t.txt
researcher2@0dd6b8f3f3f7:~/projects$
```

In this scenario I realized that “project_k.txt” has write permissions for other users.
To change this i ran the command chmod o-w project_k.txt

Change file permissions on a hidden file

This is an example that demonstrates how I used Linux commands how to change permissions on a hidden file:

```
researcher2@0dd6b8f3f3f7:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Sep 26 20:15 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep 26 21:39 ..
-rw--w---- 1 researcher2 research_team   46 Sep 26 20:15 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep 26 20:15 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Sep 26 20:15 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep 26 20:15 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 26 20:15 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 26 20:15 project_t.txt
researcher2@0dd6b8f3f3f7:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@0dd6b8f3f3f7:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Sep 26 20:15 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep 26 21:39 ..
-r--r----- 1 researcher2 research_team   46 Sep 26 20:15 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep 26 20:15 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Sep 26 20:15 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep 26 20:15 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 26 20:15 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 26 20:15 project_t.txt
researcher2@0dd6b8f3f3f7:~/projects$
```

The file .project_x.txt is a hidden file that has been archived and should not be written to by anyone. (The user and group should still be able to read this file.)
To change this I ran the command `chmod u-w,g-w,g+r .project_x.txt`
Notice when i run the command “ls -la” again the file .project_x.txt no longer has written permissions but the user and group still are able to read this file

Change directory permissions

This is an example that demonstrates how I used Linux commands to change directory permissions:

```
researcher2@4f7438a1062e:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Sep 26 21:00 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Sep 26 21:00 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep 26 21:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 26 21:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 26 21:00 project_t.txt
researcher2@4f7438a1062e:~/projects$ chmod g-x drafts
researcher2@4f7438a1062e:~/projects$ ls -l
total 20
drwx------ 2 researcher2 research_team 4096 Sep 26 21:00 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Sep 26 21:00 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep 26 21:00 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 26 21:00 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 26 21:00 project_t.txt
researcher2@4f7438a1062e:~/projects$
```

In this scenario the researcher2 is the only user that should be allowed to access the drafts directory and its contents. (This means that only researcher2 should have execute privileges.) I noticed that the group has execute permissions and therefore has access to the drafts directory.

To remove the execute permission for the group from the drafts directory, the command `chmod g-x drafts` was ran thus removing the execute permissions for the group and the `ls` command was used to verify changes were made

Summary

In Linux, file permissions dictate who can read, write, or execute files and directories. To check permissions, the `ls -l` command is used, displaying permissions in a three-group format representing the owner, group, and others. Changing permissions involves the `chmod` command, with numeric or symbolic values to set the desired access. For hidden files, which begin with a '.', the same `chmod` command applies. Directory permissions are altered similarly, but with the understanding that execute permission allows entry into the directory. Understanding and managing these permissions is crucial for system security and user management.