

File permissions in Linux

Project description

Given the scenario of modifying permissions for files and directories within a `project` directory on a Linux system, here's how I approached this task using the Linux command-line interface:

Check file and directory details

To begin with, I navigated to the `projects` directory with the `cd projects` command. I executed the command `ls` to display the available directories. The result showed that `project` was the only directory listed. Next, I used the command `ls -la` to display the contents of the directory, including hidden files.

The output revealed the following:

1. There is one hidden file within the `project` directory named `.project_x.txt`
2. Four project files
3. One directory named `drafts`

```
researcher2@65b9e874f26d:~$ ls
projects
researcher2@65b9e874f26d:~$ cd projects
researcher2@65b9e874f26d:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 23 19:34 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 23 19:41 ..
-rw--w---- 1 researcher2 research_team   46 Oct 23 19:34 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct 23 19:34 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 23 19:34 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 23 19:34 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 23 19:34 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 23 19:34 project_t.txt
researcher2@65b9e874f26d:~/projects$
```

Describe the permission string

A sequence of ten characters governs the rights and privileges for interacting with files. Each character in this sequence has a distinct meaning related to access control. Let's examine the initial entry from the illustration provided. I will referring to the 6th row `drwxr-xr-x` from the image above to explain the permission characters.

The first character in a file listing indicates the file type:

- `d` : This character indicates that the item is a directory
- `-` : A hyphen in this position indicates that the item is a regular file
- `.` : A period shows the file is hidden

Let's break down what each character means:

Characters 2-4 `rw-`: Owner permissions

- `r`: The owner has read permission
- `w` : The owner has write permission
- `-` : The owner does not have execute permission for this directory

Characters 5-7 `xr-` Group permissions

- `x` : Group members have execute permission (can access the directory)
- `r`: Group members have read permission
- `-`: Group members do not have write permission

Characters 8-10 `x`: Others permissions

- `x` : Others (everyone else) have execute permission (can access the directory)
- `-` : Others do not have read permission (implied by the missing `r`)
- `-` : Others do not have write permission (implied by the missing `w`)

This permission set `drw-xr-x` means:

- It's a directory
- The owner can read and write in the directory, but cannot access its contents (no execute permission)
- Group members can read the directory contents and access it, but cannot modify it
- Others can only access the directory, but cannot read its contents or modify it

Change file permissions

The `chmod` (change mode) command is a powerful tool in Linux systems that allows users to alter file and directory permissions. When using `chmod`, it's important to consider the following key points:

The `chmod` command allows files and directories to be modified, here are some useful commands.

Add user permissions:

- `chmod u+[r/w/x] filename`

Remove user permissions:

- `chmod u-[r/w/x] filename`

Add group permissions:

- `chmod g+[r/w/x] filename`

Remove group permissions:

- `chmod g-[r/w/x] filename`

Add permissions for others:

- `chmod o+[r/w/x] filename`

Remove permissions for others:

- `chmod o-[r/w/x] filename`

In these commands:

1. `u`: user (owner)
2. `g`: group
3. `o`: others
4. `r`: read
5. `w`: write
6. `x`: execute

Modifications I made included:

Removing write permissions for others:

- I executed the command `chmod o-w project_k.txt` to revoke write permissions for the "others" category on the file named `project_k.txt`. This means that users who are not the owner and not in the file's group can no longer modify this file.

Removing read permission for the group:

- I used the command `chmod g-r project_m.txt` to remove read permissions for the group associated with the file `project_m.txt`. As a result, members of the file's group can no longer view the contents of this file.

```
researcher2@65b9e874f26d:~/projects$ chmod o-w project_k.txt
researcher2@65b9e874f26d:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 23 19:34 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 23 19:41 ..
-rw--w---- 1 researcher2 research_team  46 Oct 23 19:34 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct 23 19:34 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Oct 23 19:34 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Oct 23 19:34 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 23 19:34 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 23 19:34 project_t.txt
researcher2@65b9e874f26d:~/projects$ chmod g-r project_m.txt
researcher2@65b9e874f26d:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 23 19:34 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 23 19:41 ..
-rw--w---- 1 researcher2 research_team  46 Oct 23 19:34 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct 23 19:34 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Oct 23 19:34 project_k.txt
-rw----- 1 researcher2 research_team  46 Oct 23 19:34 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 23 19:34 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 23 19:34 project_t.txt
researcher2@65b9e874f26d:~/projects$
```

Change file permissions on a hidden file

The `chmod` command can also modify permissions for hidden files in Linux. For the hidden file `.project_x.txt`, we want to:

1. Remove write permissions for both the owner and the group
2. Ensure the group retains read permissions

By using this compact command, `chmod u-w,g=r .project_x.txt` we efficiently adjust the permissions of the hidden file `.project_x.txt` to meet our specific requirements in one operation

```
researcher2@65b9e874f26d:~/projects$ chmod u-w,g=r .project_x.txt
researcher2@65b9e874f26d:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 23 19:34 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 23 19:41 ..
-r--r----- 1 researcher2 research_team  46 Oct 23 19:34 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct 23 19:34 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Oct 23 19:34 project_k.txt
-rw----- 1 researcher2 research_team  46 Oct 23 19:34 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 23 19:34 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 23 19:34 project_t.txt
researcher2@65b9e874f26d:~/projects$
```

Change directory permissions

The command `chmod g-x drafts` modifies the permissions of the `drafts` directory by removing the execute permission for the group

```
researcher2@65b9e874f26d:~/projects$ chmod g-x drafts
researcher2@65b9e874f26d:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 23 19:34 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 23 19:41 ..
-r--r----- 1 researcher2 research_team  46 Oct 23 19:34 .project_x.txt
drwx----- 2 researcher2 research_team 4096 Oct 23 19:34 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Oct 23 19:34 project_k.txt
-rw----- 1 researcher2 research_team  46 Oct 23 19:34 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 23 19:34 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 23 19:34 project_t.txt
researcher2@65b9e874f26d:~/projects$
```

Summary

This scenario highlights my ability to manage file and directory permissions in the project directory according to my organization's authorization policies. I effectively use the `ls -la` command to display all files, including hidden ones, and employ the `chmod` command to modify permissions for users, groups, and others. By adjusting these permissions, I ensure proper access control and enhance data security within a Linux environment.