File permissions in Linux

Project description

The research team at my organization needed to review and update file and directory permissions within the projects directory to better support collaboration while maintaining system security. Some files required additional access for the group, while other files and directories needed more restrictive permissions to prevent unauthorized access. To address this, I reviewed the existing permissions and applied appropriate changes using Linux commands.

Check file and directory details

```
researcher2@4dbbe570b91f:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jan 28 04:43 .
drwxr-xr-x 3 researcher2 research_team 4096 Jan 28 05:07 ..
-rw--w---- 1 researcher2 research_team   46 Jan 28 04:43 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jan 28 04:43 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jan 28 04:43 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jan 28 04:43 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan 28 04:43 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan 28 04:43 project_t.txt
researcher2@4dbbe570b91f:~/projects$
```

The first line on the command use the `ls -la` command. Output of this command is to The output displays a hidden file named `.project_x.txt`, a directory called **drafts**, and several project files. The 10-character string in the first column represents the permissions set on each file or directory.

Describe the permissions string

The 10-character permission string explains who is allowed to access a file or directory and what actions they can perform. Each position in the string has a specific meaning:

- **1st character:** Indicates the type of item. A **d** means it is a directory, while a hyphen **(-)** means it is a regular file.

- **2nd–4th characters:** Represent the **read (r)**, **write (w)**, and **execute (x)** permissions assigned to the user (owner). If a hyphen **(-)** appears in any position, that permission is

not granted to the user.

- **5th–7th characters:** Represent the **read (r)**, **write (w)**, and **execute (x)** permissions for the group. A hyphen (-) indicates that the corresponding permission is not granted to the group.

- **8th–10th characters:** Represent the **read (r)**, **write (w)**, and **execute (x)** permissions for others, which includes all remaining users on the system who are not part of the user or group. A hyphen (-) indicates that the permission is not granted.

Change file permissions

The organization determined project_m.txt must have write access added for group.

The following code demonstrates how i used Linux commands to change the permissions :

```
researcher2@4dbbe570b91f:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jan 28 04:43 .
drwxr-xr-x 3 researcher2 research_team 4096 Jan 28 05:07 ..
-rw--w---- 1 researcher2 research_team   46 Jan 28 04:43 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jan 28 04:43 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jan 28 04:43 project_k.txt
-rw-rw---- 1 researcher2 research_team   46 Jan 28 04:43 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan 28 04:43 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan 28 04:43 project_t.txt
researcher2@4dbbe570b91f:~/projects$
```

The command to do this is using chmod, chmod command changes the permissions on files. Implementation of chmod command : chmod g+w project_m.txt to add write permissions on project_m.txt. After adding write permissions, I used ls -la command to review the updates I made

Change file permissions on a hidden file

The research team at the organization recently archived .project_x.txt. They don't want the group to have read access to this project, but the group should have read access.

The following code demonstrates how i used Linux commands to change the permissions :

```
researcher2@4dbbe570b91f:~/projects$ chmod g+r .project_x.txt
researcher2@4dbbe570b91f:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jan 28 04:43 .
drwxr-xr-x 3 researcher2 research_team 4096 Jan 28 05:07 ..
-rw-rw---- 1 researcher2 research_team   46 Jan 28 04:43 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jan 28 04:43 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jan 28 04:43 project_k.txt
-rw-rw---- 1 researcher2 research_team   46 Jan 28 04:43 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan 28 04:43 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan 28 04:43 project_t.txt
researcher2@4dbbe570b91f:~/projects$
```

The first two lines of the command I entered is adding read access using chmod command and ls -la command to see the update i made. Now .project_x.txt the user and group have read and write permissions to file.

Change directory permissions

My organization requires that only the researcher2 user can access the drafts directory and everything inside it. This means that no other users or groups should have execute permissions on this directory.

The following code demonstrates how i used Linux commands to change the permissions :

```
researcher2@4dbbe570b91f:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jan 28 04:43 .
drwxr-xr-x 3 researcher2 research_team 4096 Jan 28 05:07 ..
-rw-rw---- 1 researcher2 research_team   46 Jan 28 04:43 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Jan 28 04:43 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jan 28 04:43 project_k.txt
-rw-rw---- 1 researcher2 research_team   46 Jan 28 04:43 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan 28 04:43 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan 28 04:43 project_t.txt
researcher2@4dbbe570b91f:~/projects$
```

The output displays the permission list for multiple files and directories.The fourth line represents the drafts directory, which now has restricted permissions. Only the researcher2 user has execute access. The group previously had execute permissions, so I used the chmod g-x drafts command to remove them. Since researcher2 already had execute permissions, no additional permissions needed to be added.

Summary

The following Result :

```
researcher2@4dbbe570b91f:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jan 28 04:43 .
drwxr-xr-x 3 researcher2 research_team 4096 Jan 28 05:07 ..
-rw-rw---- 1 researcher2 research_team   46 Jan 28 04:43 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Jan 28 04:43 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jan 28 04:43 project_k.txt
-rw-rw---- 1 researcher2 research_team   46 Jan 28 04:43 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan 28 04:43 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jan 28 04:43 project_t.txt
researcher2@4dbbe570b91f:~/projects$
```

I updated several file and directory permissions within the projects directory to align with my organization's authorization requirements. I first used `ls -la` to review the existing permissions, which guided the changes I made. I then applied the `chmod` command to modify permissions on the relevant files and directories.