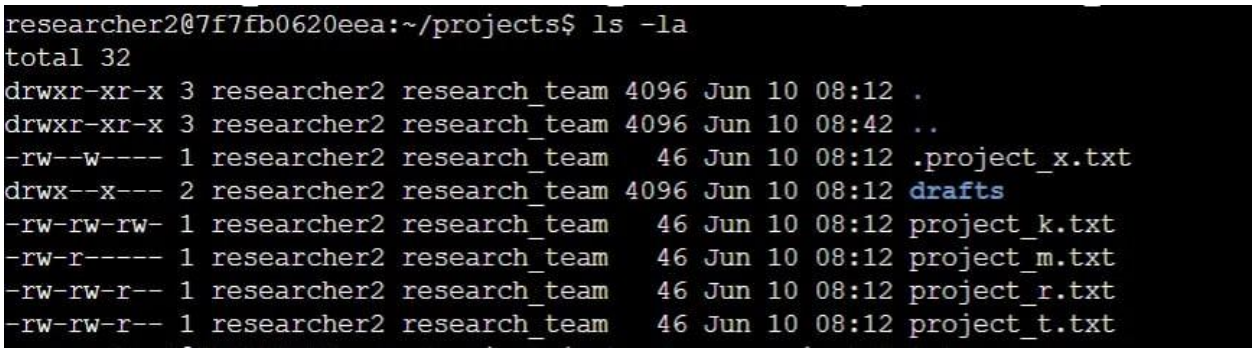
File permissions in Linux

Project description

My organization needed to update the permissions for certain files and directories within the projects directory. To complete this task, I performed following tasks:

Check file and directory details

Within the projects directory, I used the command **ls -la** to check the details about permissions in listed files and directories, including hidden files:



```
researcher2@7f7fb0620eea:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 10 08:12 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 10 08:42 ..
-rw--w---- 1 researcher2 research_team   46 Jun 10 08:12 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 10 08:12 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jun 10 08:12 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jun 10 08:12 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 10 08:12 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 10 08:12 project_t.txt
```

Describe the permissions string

The permission string shows what permissions are in use for three types of users: **user**, **group** and **other**. **User** is the account that created this file (in this case researcher2), and goes listed as first in the permission string. The second type of user is a **group** (research_team), and the last one is type **other,** meaning every other possible user.
The permission string shows who has read**(r)**, write**(w)**, or execute**(x)** permissions, in that order. On the beginning of the string, there is information if this is a file**(-)** or a directory**(d)**. For example, in project_t.txt, we can see that this is a file. The user has **rw-** permissions, same as the group. Other users can only read**(r--)** the file.

Change file permissions

My organization decided that no other users should have write permissions to any of their files in the projects directory. To change the permissions as needed, I used the **chmod** command:

```
researcher2@7f7fb0620eea:~/projects$ chmod o-w project_k.txt
researcher2@7f7fb0620eea:~/projects$ chmod g-r project_m.txt
researcher2@7f7fb0620eea:~/projects$ chmod u=r,g=r .project_x.txt
researcher2@7f7fb0620eea:~/projects$ chmod g-x drafts
researcher2@7f7fb0620eea:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 10 08:12 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 10 08:42 ..
-r--r----- 1 researcher2 research_team   46 Jun 10 08:12 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Jun 10 08:12 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jun 10 08:12 project_k.txt
-rw------- 1 researcher2 research_team   46 Jun 10 08:12 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 10 08:12 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 10 08:12 project_t.txt
researcher2@7f7fb0620eea:~/projects$
```

For project_k.txt, I removed the permission to write**(w)** for other users**(o)**. For project_m.txt, I removed a group**(g)** permission to read**(r)** the file. In the case of a hidden file, .project_x.txt, I allowed both user and group to have read access, removing all of the remaining permissions at the same time. As for the drafts directory, the organization asked me to remove the execute**(x)** permission for the group.

I used the **ls -la** command again to see the results.

Summary

I changed multiple permissions to match the level of authorization my company needed in the projects directory. In order to achieve that, I first had to use the **ls -la** command to see the files as well as what permissions are in place, then I managed them using the **chmod** command.