File permissions in Linux

Project description

In this project, I will be checking the permissions on files and directories in the Linux operating system. I will also be updating the permissions based on instructions received from a faux organization. Updating and checking permissions should happen regularly at an organization and helps to keep systems and information secure. This project is four pages and shows print screens of my ability to execute permission tasks using Linux.

Check file and directory details

Using the `ls -la` command in the `projects` directory, I am able to list all permissions including those on hidden files.

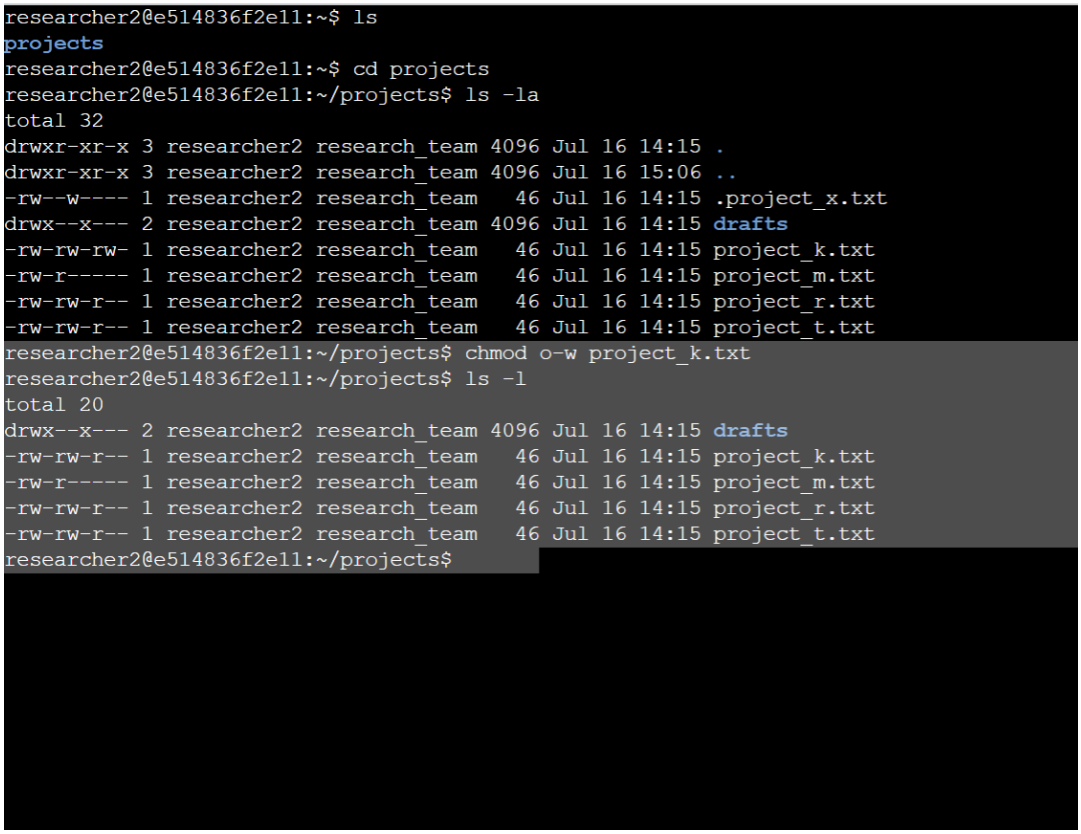
```
researcher2@e514836f2e11:~$ ls
projects
researcher2@e514836f2e11:~$ cd projects
researcher2@e514836f2e11:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jul 16 14:15 .
drwxr-xr-x 3 researcher2 research_team 4096 Jul 16 15:06 ..
-rw--w---- 1 researcher2 research_team   46 Jul 16 14:15 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jul 16 14:15 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jul 16 14:15 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jul 16 14:15 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 16 14:15 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jul 16 14:15 project_t.txt
researcher2@e514836f2e11:~/projects$
```

Describe the permissions string

The permissions string is a 10 character string that describes the permissions of a file or directory. Let us describe the permissions on `project_k.txt`. The first hyphen shows this is a file and not a directory. The 2nd, 3rd, and 4th spaces show the user permissions using the character r (read), w (write), and x (execute). If the user is not authorized to perform one or more of these permissions, the character will be replaced with a hyphen. In our example, the user only had read and write permissions. The 5th-7th characters show permissions for the group owner, while the 8th-10th characters show permissions for the “other” owner type.

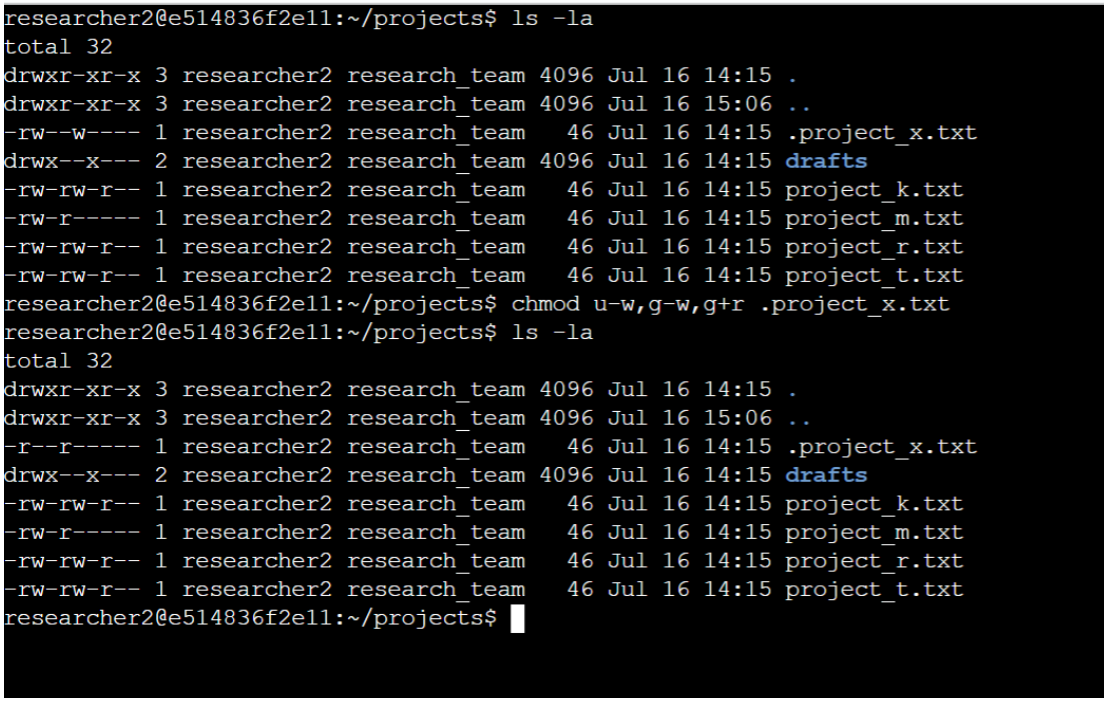
Change file permissions

Below is an example of using the `chmod` command to change permissions on `project_k.txt` for the “other” owner type. The command `chmod o-w project_k.txt` removed writing permissions for this owner. I then used the command `ls -l` to list current permissions and assure the change was completed. You can see this in the highlighted portion of the screenshot.



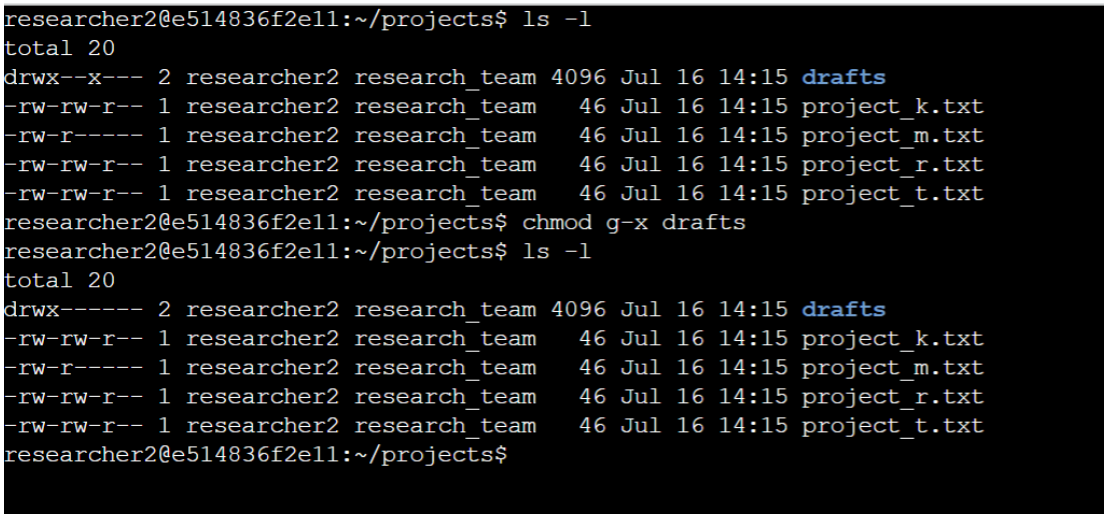
Change file permissions on a hidden file

In the example below, I have changed the permissions for the hidden file `.project_x.txt`. I first used `ls -la` to list all permissions including those for hidden files. I then used the `chmod` command to remove write permissions for both the user and group. I then added read permissions for the group. This command is `chmod u-w,g-w,g+r .project_x.txt`. Lastly I checked the permissions again to assure the changes occurred properly.



Change directory permissions

In the image below, you can see that I have changed the permissions for the drafts subdirectory. In this case, only the user should have access to read, write, and execute this directory. Using the `ls -l` command, I was able to see that the group owner also had access to execute the directory. Using the `chmod g-x` command, I was able to remove this permission from the group owner. I then used `ls -l` to assure the change was successful.



Summary

In this project, I was able to demonstrate my knowledge of owner types and permission commands using the Linux OS. Permissions were listed for the files and subdirectories within the “projects” directory. Reading the permissions string, I was able to assess the permissions allowed for the different owners and add or remove permissions as necessary.