File permissions in Linux

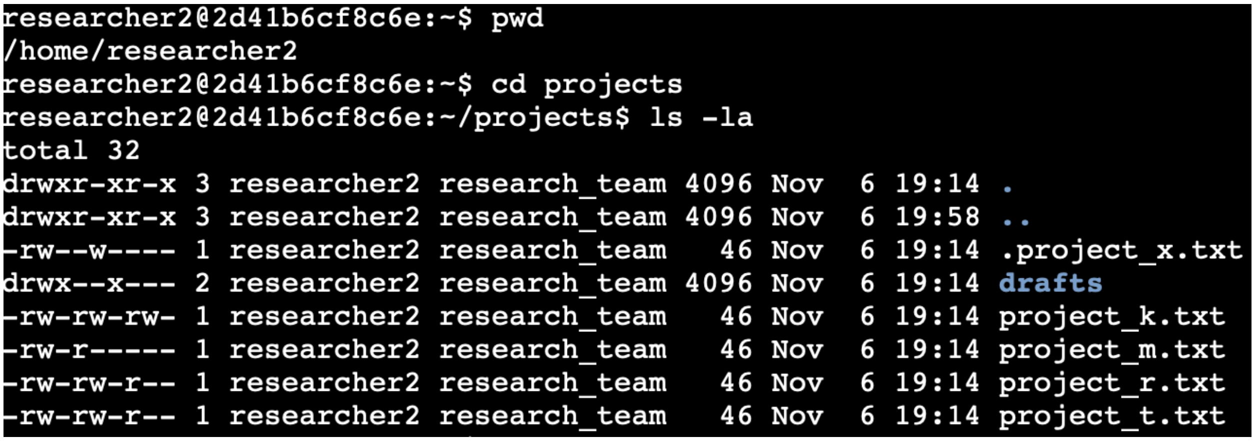
Project description

[Describe what you accomplish through Linux commands.]

Check file and directory details

First, use **pwd** to find what directory you are currently working in. To check permissions in the **/projects** directory, including hidden files, you must ensure you are in the correct working directory, then use the **cd** command to make the change.

ls -la simultaneously displays the contents of the current directory while revealing the permissions, including hidden files of each item.



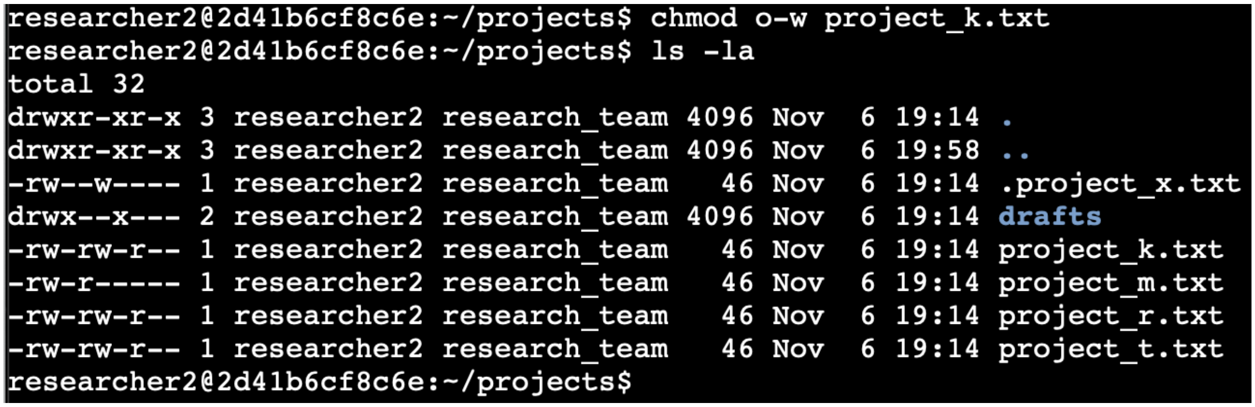
```
researcher2@2d41b6cf8c6e:~$ pwd
/home/researcher2
researcher2@2d41b6cf8c6e:~$ cd projects
researcher2@2d41b6cf8c6e:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Nov  6 19:14 .
drwxr-xr-x 3 researcher2 research_team 4096 Nov  6 19:58 ..
-rw--w---- 1 researcher2 research_team   46 Nov  6 19:14 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Nov  6 19:14 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Nov  6 19:14 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Nov  6 19:14 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov  6 19:14 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov  6 19:14 project_t.txt
```

Describe the permissions string

The 10-character string displayed when viewing permissions in Linux describes whether the item is a directory or a file, and what the current permissions are for the user, the group, and other.
The first character can either be **d** (directory) or **-** (file).
Characters 2-4 represent the read/write/execute permissions for the user (researcher2).
Characters 5-7 represent the read/write/execute permissions for the group (research_team).
Characters 8-10 represent the read/write/execute permissions for other(s).

Change file permissions

Other(s) should not have write access permissions to any files within the organization. The project file **project_k.txt** currently has write permissions for other and this should be changed. The command **chmod** is used along with **-** or **+** to add or remove **r, w, x** permissions for **u, g,** or **o**.



```
researcher2@2d41b6cf8c6e:~/projects$ chmod o-w project_k.txt
researcher2@2d41b6cf8c6e:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Nov  6 19:14 .
drwxr-xr-x 3 researcher2 research_team 4096 Nov  6 19:58 ..
-rw--w---- 1 researcher2 research_team   46 Nov  6 19:14 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Nov  6 19:14 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Nov  6 19:14 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Nov  6 19:14 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov  6 19:14 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov  6 19:14 project_t.txt
researcher2@2d41b6cf8c6e:~/projects$
```

Change file permissions on a hidden file

The archived file **.project_x.txt** is hidden and should not have write permissions for anyone, but the user and group should be able to read it. Using **chmod** with **=** can overwrite previous permissions and assign new ones.

Change directory permissions

chmod g-x drafts

Summary

It is important to ensure permissions for directories and files are kept up to date to maintain the principles of least privilege when it comes to the security of an organization. By ensuring only the right people can access certain files with varying degrees of confidentiality, the risks involved in giving too much access to unauthorized individuals is greatly reduced.