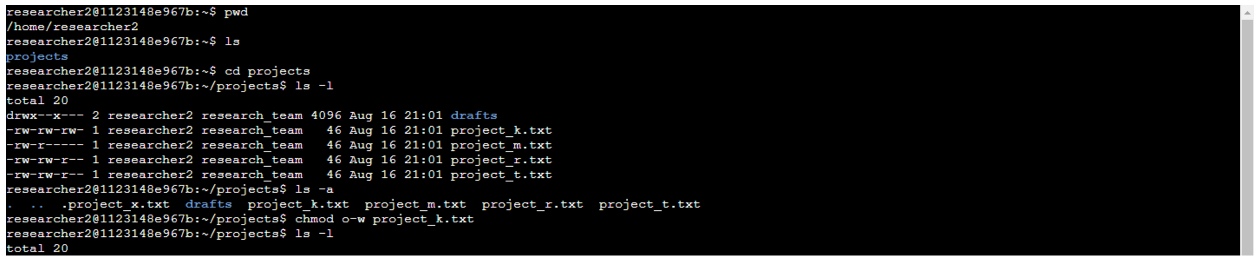
File permissions in Linux

Project description

The organization's research team needs to change the file permissions for a few specific files and folders in the project's directory. The permissions do not presently correspond to the appropriate level of authorisation. Their system will remain safe if these permissions are checked and updated. I did the following things to finish this task

Check file and directory details

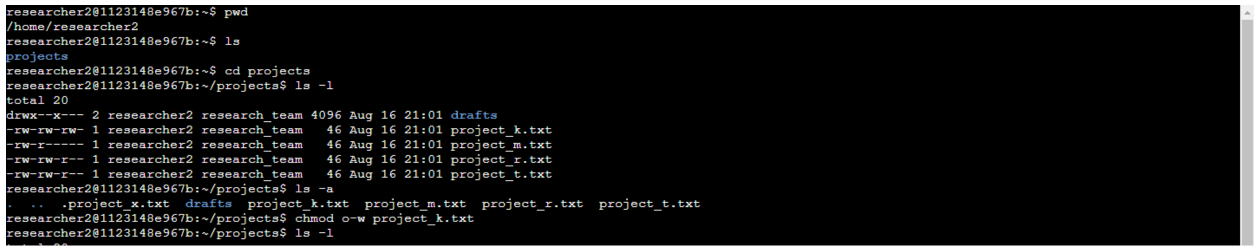
With the use of the ls -l command on Linux I can review the current user permissions in the given directory and in this case, it is the " /home/researcher2/projects" directory to change or modify current permissions for different users. As shown in the image below



```
researcher2@1123148e967b:~$ pwd
/home/researcher2
researcher2@1123148e967b:~$ ls
projects
researcher2@1123148e967b:~$ cd projects
researcher2@1123148e967b:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Aug 16 21:01 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Aug 16 21:01 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug 16 21:01 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 16 21:01 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 16 21:01 project_t.txt
researcher2@1123148e967b:~/projects$ ls -a
.  ..  .project_x.txt  drafts  project_k.txt  project_m.txt  project_r.txt  project_t.txt
researcher2@1123148e967b:~/projects$ chmod o-w project_k.txt
researcher2@1123148e967b:~/projects$ ls -l
total 20
```

Describe the permissions string

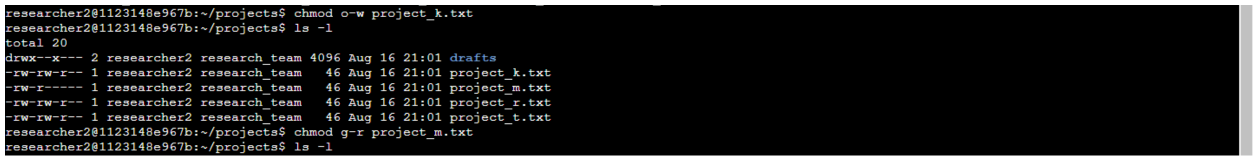
For example, the drwx - - x - - - permission string from the directory in this file contains the permissions for the User, Group of users and others. The d in this instance stands for the directory, and the second, third and fourth characters, in this case, rwx, represent the user's permissions, r meaning read, w meaning write and x meaning execute. The fifth, sixth and seventh characters in this instance, - - x means the permissions of the group users and the last 3 characters represent the permissions for the other users. The (-) characters stand for restricted permissions. To the left of the image below is an example of permission strings.



```
researcher2@1123148e967b:~$ pwd
/home/researcher2
researcher2@1123148e967b:~$ ls
projects
researcher2@1123148e967b:~$ cd projects
researcher2@1123148e967b:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Aug 16 21:01 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Aug 16 21:01 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug 16 21:01 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 16 21:01 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 16 21:01 project_t.txt
researcher2@1123148e967b:~/projects$ ls -a
.  ..  .project_x.txt  drafts  project_k.txt  project_m.txt  project_r.txt  project_t.txt
researcher2@1123148e967b:~/projects$ chmod o-w project_k.txt
researcher2@1123148e967b:~/projects$ ls -l
```

Change file permissions

The company decided that no one else should be able to write to any of its files. I referenced back to the file permissions that I had previously returned in order to comply with this. I came to the conclusion that other users should not have write access to project_k.txt.



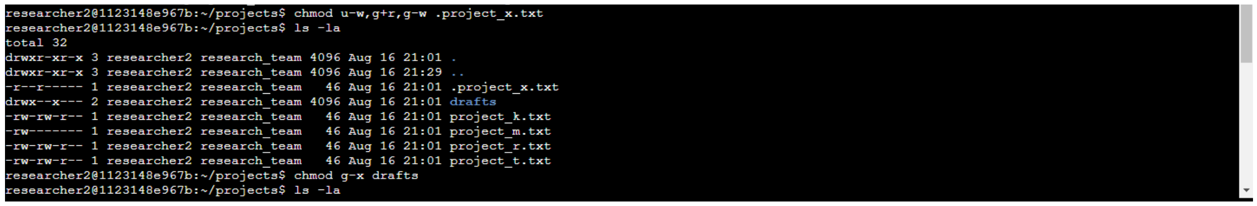
```
researcher2@1123148e967b:~/projects$ chmod o-w project_k.txt
researcher2@1123148e967b:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Aug 16 21:01 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Aug 16 21:01 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug 16 21:01 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 16 21:01 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 16 21:01 project_t.txt
researcher2@1123148e967b:~/projects$ chmod g-r project_m.txt
researcher2@1123148e967b:~/projects$ ls -l
```

The code above is how I executed this specific task.
The screenshot's first two lines show the instructions I typed in, and the following lines show the second command's results. The chmod command modifies a file's or directory's permissions. The second parameter provides the file or directory, while the first argument specifies which permissions should be altered. Here, I eliminated the word "write"

other people's permissions for the project_k.txt file. I then ran ls -l to look through the adjustments I had made.

Change file permissions on a hidden file

.Project_x.txt was recently archived by the organization's research team. The user and group should only be able to view this project; they do not want anybody to have write access.
The image below displays the code on a Linux interface on how I executed the request

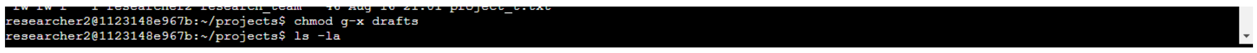


```
researcher2@1123148e967b:~/projects$ chmod u-w,g+r,g-w .project_x.txt
researcher2@1123148e967b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug 16 21:01 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug 16 21:29 ..
-r--r----- 1 researcher2 research_team   46 Aug 16 21:01 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug 16 21:01 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Aug 16 21:01 project_k.txt
-rw------- 1 researcher2 research_team   46 Aug 16 21:01 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 16 21:01 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 16 21:01 project_t.txt
researcher2@1123148e967b:~/projects$ chmod g-x drafts
researcher2@1123148e967b:~/projects$ ls -la
```

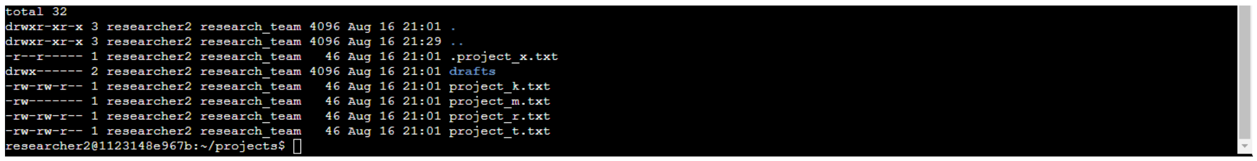
The screenshot's first two lines show the instructions I typed in, and the following lines show the second command's results. I know because it begins with a period (.), project_x.txt is a secret file. In this illustration, I gave the group read permissions while removing write permissions from the user and group. I used u-w to take the user's write permissions away. I then added read access to the group with g+r and deleted write permissions with g-w.

Change directory permissions

Only the researcher2 user should have access to the draughts directory and its contents, according to the organization. This indicated that only researcher2 should be granted execute permissions.



```
researcher2@1123148e967b:~/projects$ chmod g-x drafts
researcher2@1123148e967b:~/projects$ ls -la
```



```
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug 16 21:01 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug 16 21:29 ..
-r--r----- 1 researcher2 research_team   46 Aug 16 21:01 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Aug 16 21:01 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Aug 16 21:01 project_k.txt
-rw------- 1 researcher2 research_team   46 Aug 16 21:01 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 16 21:01 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 16 21:01 project_t.txt
researcher2@1123148e967b:~/projects$ 
```

The code above indicates how I used the Linux interface to execute the task.
The screenshot's first two lines show the instructions I typed in, and the following lines show the second command's results.
I used the chmod command to remove the execute rights after previously determining that the group possessed them. It was not necessary to add execute rights because the researcher2 user already had them.

Summary

Multiple user permissions were adjusted to the right amount of access required for the dedicated user, user group or other users. The use of ls -la command on the Linux OS demonstrates the adjustments made were saved and are active.