File permissions in Linux

Project description

I will check for permission on each file in the `projects` directory and determine if they are correct. If not, I will adjust them accordingly using Linux CLI commands.

Check file and directory details

I will first change to the `projects` directory using the `cd` command. Afterwards, I identify which files are in the directory. I use `ls -la` to check permissions on each file.

```
researcher2@37786ab29543:~$ cd projects
researcher2@37786ab29543:~/projects$ ls
drafts  project_k.txt  project_m.txt  project_r.txt  project_t.txt
researcher2@37786ab29543:~/projects$ ls la
ls: cannot access 'la': No such file or directory
researcher2@37786ab29543:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Nov 28 00:40 .
drwxr-xr-x 3 researcher2 research_team 4096 Nov 28 01:18 ..
-rw--w---- 1 researcher2 research_team   46 Nov 28 00:40 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Nov 28 00:40 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Nov 28 00:40 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Nov 28 00:40 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 28 00:40 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 28 00:40 project_t.txt
researcher2@37786ab29543:~/projects$
```

There is one directory named `drafts` and a hidden file named `.project_x`.

Describe the permissions string

The breakdown for the permissions are as follows:

- The first space identifies the file type. If there is a "d" then it means that the file is a directory
- The three spaces afterward identify the permissions for the **user** category. The "r" would indicate **read** permissions, the "w" indicates **write** permissions and finally, the "x" indicates **execute** permissions.
- The next three spaces identify the permissions for the **group** category

- The final three spaces identify the permissions for the **other** category

For example, for the file labelled ".", the "d" stands for **directory** which is the file type. The user has **read, write**, and **execute** permission. The group has **read** and **execute** permissions but not **write**. The other has **read** and **execute** permissions, but not **write**.

Change file permissions

The organization for this project indicated that there should not be **write** files for the **other** category on any of the files. If there is a "w" on any of the file permissions in the last three digits of the permission spaces, then they will need to be removed.

As the file "project_k.txt" has write permissions for the **other** category, I used the following

```
researcher2@37786ab29543:~/projects$ chmod o-w project_k.txt
```

Here, using the ls -la command again, it is clear that the file will no longer have the "w" permission for the **user** category.

```
researcher2@37786ab29543:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Nov 28 00:40 .
drwxr-xr-x 3 researcher2 research_team 4096 Nov 28 01:18 ..
-rw--w---- 1 researcher2 research_team   46 Nov 28 00:40 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Nov 28 00:40 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Nov 28 00:40 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Nov 28 00:40 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 28 00:40 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 28 00:40 project_t.txt
```

Change file permissions on a hidden file

The organization in this project does not want anyone to have write permissions on the hidden file, .project_x.txt. However, they want the user and group to have read permissions for it.

The .project_x.txt still has **write** permissions for both **user** and **group**. It is missing read permissions for the **group**. I used the following command to remove write permissions for the user and the group, but added read permissions to the group:

```
researcher2@37786ab29543:~/projects$ chmod u-w,g-w,g+r .project_x.txt
```

Afterwards, I used `ls -la` command once again to verify that `.project_x.txt` is readable to the **user** and **group** but not writable to anyone.

```
drwxr-xr-x 3 researcher2 research_team 4096 Nov 28 00:40 .
drwxr-xr-x 3 researcher2 research_team 4096 Nov 28 01:18 ..
-r--r----- 1 researcher2 research_team   46 Nov 28 00:40 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Nov 28 00:40 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Nov 28 00:40 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Nov 28 00:40 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 28 00:40 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 28 00:40 project_t.txt
```

Change directory permissions

The organization wants the **user**, `researcher2`, to be the only one to have **execute** permissions on the `draft` directory.

However, the group still has **execute** permissions for it. As such, I used the following command to remove those **execute** permissions:

```
researcher2@37786ab29543:~/projects$ chmod g-x drafts
```

Now, only the user has **read, write**, and **execute** permissions for `drafts`.

```
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Nov 28 00:40 .
drwxr-xr-x 3 researcher2 research_team 4096 Nov 28 01:18 ..
-r--r----- 1 researcher2 research_team   46 Nov 28 00:40 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Nov 28 00:40 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Nov 28 00:40 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Nov 28 00:40 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 28 00:40 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 28 00:40 project_t.txt
```

Summary

Using the CLI Linux commands, I navigated to the `projects` directory then changed permissions on various files as according to the organization's needs.

The organization indicated that they did not want **write** permissions for the **other** category.

- The `project_k.txt` file has **write** permissions
- Using `chmod o-w` command, I removed the **write** permissions on **other**

- Afterwards, I used `ls -la` to verify that the permissions are gone

The organization specified that the hidden file, `.project_x.txt`, must have no **write** files but can be readable for **user** and **group**.

- The `.project_x.txt` still has **write** permissions for both **user** and **group**
- However, it does not have **read** permissions for the **group**
- Using `chmod u-w, g-w, g+r .project_x.txt`
- Using `ls -la`, I verified that the **write** permissions have been removed from the **user** and **group** and **read** permissions have been added to the **group**

The organization wants the user, researcher2, to be the only one with **execute** permissions for the drafts directory.

- The **group** still has **execute** permissions for the `drafts` directory
- Using the `chmod g-x drafts` command, I removed the **execute** permission from the **group**
- Using `ls -la`, I verified that the execute permissions have been removed from the **group**