File permissions in Linux

Project description

In this scenario, you must examine and manage the permissions on the files in the `/home/researcher2/projects` directory for the `researcher2` user. The `researcher2` user is part of the `research_team` group.

You must check the permissions for all files in the directory, including any hidden files, to make sure that permissions align with the authorization that should be given. When it doesn't, you must change the permissions.

Check file and directory details

```
researcher2@e6d99f6a64f0:~$ ls
drafts  project_k.txt  project_m.txt  project_r.txt  project_t.txt
```

```
researcher2@e6d99f6a64f0:~/projects$ ls -l
drwx--x--- 2 researcher2 research_team 4096 Jun 10 05:32 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jun 10 05:32 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jun 10 05:32 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 10 05:32 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 10 05:32 project_t.txt
```

Describe the permissions string

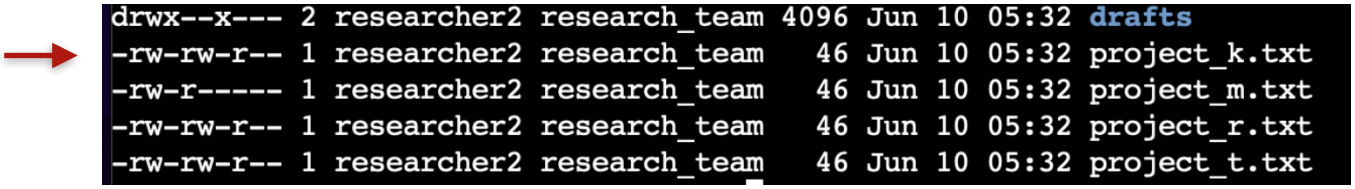
1. **`drwx--x---`**:
    - `d`: Indicates it is a directory.
    - `rwx`: The owner has read, write, and execute permissions.
    - `---`: The group and others have no permissions.
2. **`-rw-rw-rw-`**:
    - `-`: Indicates it is a regular file (not a directory).
    - `rw-`: The owner has read and write permissions.
    - `rw-`: The group has read and write permissions.
    - `rw-`: Others have read and write permissions.
3. **`-rw-r-----`**:
    - `-`: Indicates it is a regular file.
    - `rw-`: The owner has read and write permissions.
    - `r--`: The group has read-only permissions.
    - `---`: Others have no permissions.
4. **`-rw-rw-r--`**:
    - `-`: Indicates it is a regular file.
    - `rw-`: The owner has read and write permissions.

- `rw-`: The group has read and write permissions.
- `r--`: Others have read-only permissions.

5. **`-rw-rw-r--`**:
   - `-`: Indicates it is a regular file.
   - `rw-`: The owner has read and write permissions.
   - `rw-`: The group has read and write permissions.
   - `r--`: Others have read-only permissions.

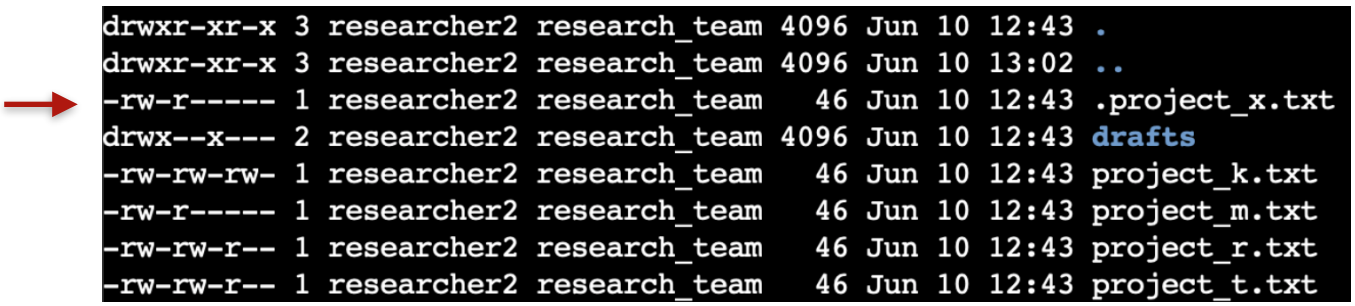
Change file permissions

```
researcher2@e6d99f6a64f0:~/projects$ chmod o-w project_k.txt
researcher2@e6d99f6a64f0:~/projects$ ls -l
```

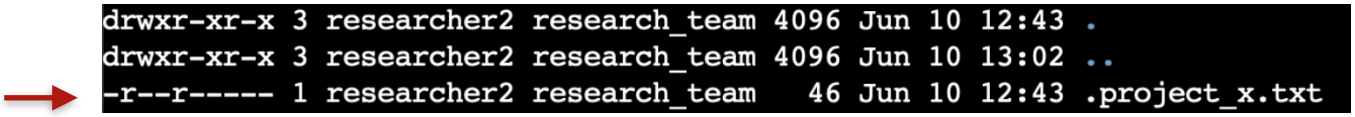


Change file permissions on a hidden file

```
researcher2@b9a5d64044c8:~/projects$ ls -la
```




```
researcher2@b9a5d64044c8:~/projects$ chmod u=r,g=r .project_x.txt
researcher2@b9a5d64044c8:~/projects$ ls -la
```



Change directory permissions

```
researcher2@b9a5d64044c8:~/projects/drafts$ chmod g-x .
researcher2@b9a5d64044c8:~/projects/drafts$ chmod g-x ..
```

```
drwx------ 2 researcher2 research_team 4096 Jun 10 12:43 .
drwxr--r-x 3 researcher2 research_team 4096 Jun 10 12:43 ..
```

Summary

File permissions in Linux determine the access levels for files and directories. There are three types of permissions: read (**r**), write (**w**), and execute (**x**).
The permissions are represented by characters in sets of three: owner-group-others.

The **chmod** command is used to modify permissions, and **ls -l** shows the permissions of files and directories.

I changed multiple permissions to match the level of authorization my organization wanted for files and directories in the `projects` directory. The first step in this was using `ls -la` to check the permissions for the directory. This informed my decisions in the following steps. I then used the `chmod` command multiple times to change the permissions on files and directories.