

# File permissions in Linux

## Project description

In the Linux environment, interaction with the operating system often takes place through the terminal, using commands that allow users to navigate directories, view files, and modify permissions. Among the most commonly used commands are `pwd`, `ls`, and `chmod`, each with specific functionalities that facilitate system administration and file organization.

The `pwd` command displays the current directory, helping the user determine their exact location within the file system. The `ls` command, with its variations such as `ls -l`, `ls -a`, and `ls -la`, enables listing files and directories in different ways, including detailed information or hidden files. Additionally, the `chmod` command is essential for managing permissions, allowing users to define who can read, write, or execute a file or directory.

Understanding and correctly using these commands is essential for any Linux user, whether for personal or professional purposes. In this document, we will explore each of these commands in detail, demonstrating their practical applications and the most commonly used variations.

## Check file and directory details

```
researcher2@77329759e364:~$ pwd
/home/researcher2
researcher2@77329759e364:~$ ls
projects
researcher2@77329759e364:~$ cd projects/
researcher2@77329759e364:~/projects$ ls
drafts  project_k.txt  project_m.txt  project_r.txt  project_t.txt
```

In this process, we examine the founders and directories until we reach the target subdirectory: **/projects**. Subsequently, we reviewed its contents inside the subdirectory and it found one more founder and .txt files.

## Describe the permissions string

```
researcher2@77329759e364:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Mar 20 21:35 drafts
-rw-rw-rw- 1 researcher2 research_team  46 Mar 20 21:35 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Mar 20 21:35 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Mar 20 21:35 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Mar 20 21:35 project_t.txt
researcher2@77329759e364:~/projects$ ls -a
.   .project_x.txt  project_k.txt  project_r.txt
..  drafts         project_m.txt  project_t.txt
researcher2@77329759e364:~/projects$
```

In Linux, the **permissions string** is a sequence of ten characters that defines access rights to files and directories. The first character indicates the file type, which can be **-** for regular files, **d** for directories, or **l** for symbolic links. The remaining nine characters are divided into three groups of three, representing the permissions for the **owner (user)**, **group**, and **others**. Each group can contain **r** (read), **w** (write), and **x** (execute), or a **-** if the permission is not granted. For example, the string **-rwxr-xr--** means the owner has read, write, and execute permissions, the group can read and execute, and others can only read. These permissions can be modified using the **chmod** command and are essential for security and access control in the system.

## Change file permissions

```
researcher2@77329759e364:~/projects$ chmod o-w project_k.txt
researcher2@77329759e364:~/projects$ chmod g-r project_m.txt

researcher2@77329759e364:~$ ls -la projects/
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Mar 20 21:35 .
drwxr-xr-x 3 researcher2 research_team 4096 Mar 20 22:48 ..
-rw--w---- 1 researcher2 research_team  46 Mar 20 21:35 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Mar 20 21:35 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Mar 20 21:35 project_k.txt
-rw----- 1 researcher2 research_team  46 Mar 20 21:35 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Mar 20 21:35 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Mar 20 21:35 project_t.txt
```

In this process, I am modifying the permissions using the **chmod** command to ensure that others cannot write to **project\_k.txt**, and the group cannot read **project\_m.txt**. The changes can be verified in the image following the commands.

## Change file permissions on a hidden file

```
researcher2@77329759e364:~/projects$ ls
drafts project_k.txt project_m.txt project_r.txt project_t.txt
researcher2@77329759e364:~/projects$ chmod u=r,g=r .project_x.txt
researcher2@77329759e364:~/projects$ █

-r--r----- 1 researcher2 research_team 46 Mar 20 21:35 .project_x.txt
```

Thus, I identified the hidden file `.project_x.txt` on the list with the commands `ls -a` and modified its permissions, granting read-only access to users and the group.

## Change directory permissions

```
total 20
drwx--x--- 2 researcher2 research_team 4096 Mar 20 21:35 drafts

researcher2@77329759e364:~/projects$ chmod g-x drafts/
researcher2@77329759e364:~/projects$ ls -l
total 20
drwx----- 2 researcher2 research_team 4096 Mar 20 21:35 drafts
```

Finally, I changed the permissions of the `/drafts` directory to restrict execution access for the group.

## Summary

This document covers essential Linux commands such as `pwd`, `ls`, and `chmod`, which are fundamental for navigation, file listing, and permission management. We explored the **permissions string** structure, defining access for **user**, **group**, and **others**, and how to modify them using `chmod`. Changes were made to files, including hidden ones, restricting write and read permissions as needed. Finally, the `/drafts` directory permissions were adjusted to restrict execution for the group, ensuring better access control.