# File permissions in Linux

## Project description

In this project, I demonstrated essential file and directory management tasks in a Linux operating system. I showcased how to check detailed information about files and directories using different command. Additionally, I explained the permissions string, which represents the access rights for users, groups, and others. The project also covered how to change file and directory permissions using symbolic representations. Furthermore, I highlighted that the process of changing permissions for hidden files is no different from regular files. Overall, the project provided a concise and practical guide to efficiently manage file and directory permissions in Linux.

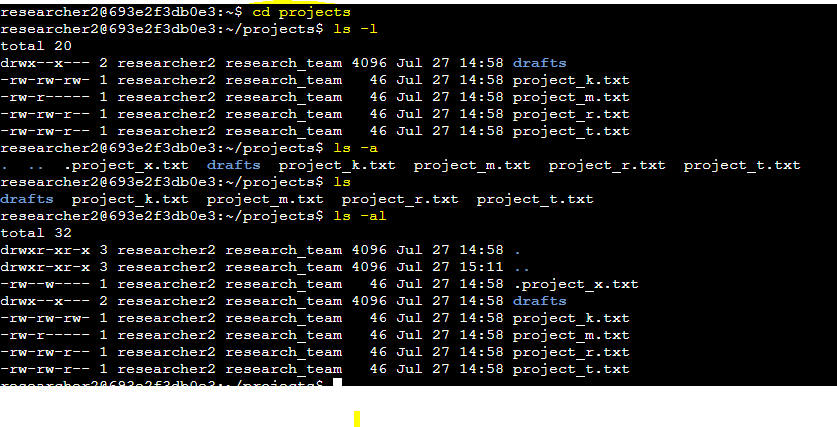
Check file and directory details.

To check the details of a file or directory in a Linux operating system, we you can use the “ls” command. Open a terminal or command prompt and navigate to the directory where the file or directory is located. Then, use the following commands:

1. To list the files and directories in the current directory: ls

## 2. To list all files and directories (including hidden ones) in the current directory: ls -a

## 3. To list the detailed information (permissions, owner, size, modification date, etc.) of a specific file or directory: ls -l



## Describe the permissions string.

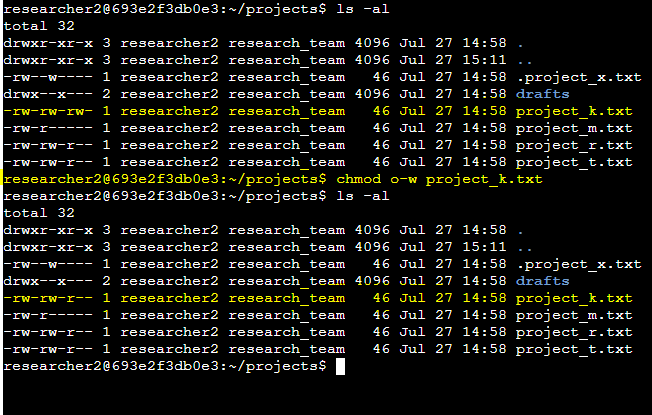
## In Linux systems, each file and directory has a set of permissions associated with it. These permissions define who can read, write, and execute the file or directory. The permissions string consists of ten characters, divided into four parts:

## File type: The first character indicates the type of the file. Common values include - for regular files, d for directories, l for symbolic links, and more.

## User permissions: The next three characters (rwx in the example) represent the permissions for the owner of the file. r stands for read, w for write, and x for execute. If the permission is granted, the corresponding letter appears; otherwise, a - is shown.

## Group permissions: The following three characters (r-x in the example) indicate the permissions for the group the file belongs to.

## Other permissions: The last three characters (r-- in the example) represent the permissions for all other users on the system who are not the owner or part of the group.

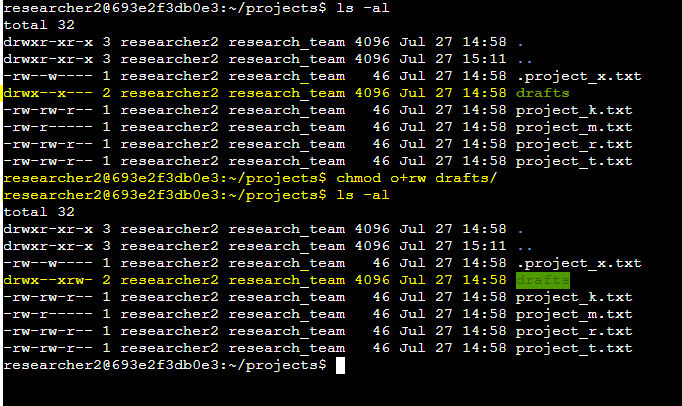


## Change file permissions

## To change file permissions, you can use the chmod command, followed by a three-digit number or a symbolic representation. The three-digit number represents the permission in octal format, while the symbolic representation is more intuitive.

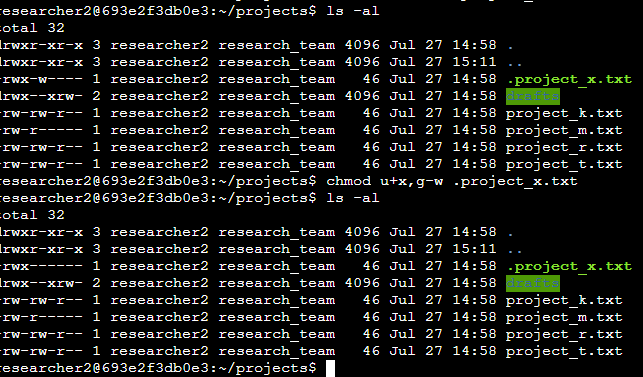
Using symbolic representation: chmod u+rwx,g+rw,o+r file.txt

This above command grants read, write, and execute permissions to the owner, read and write permissions to the group, and read-only permissions to others.



## Change file permissions on a hidden file

## Changing permissions on a hidden file is the same as changing permissions on any other file. Hidden files are just regular files with names starting with a dot (e.g., .hiddenfile). To modify the permissions of a hidden file, use the chmod command as shown in the previous section.



## Change directory permissions

## Changing directory permissions is also done using the chmod command. The process is the same as changing file permissions, but you apply the command to the directory instead.

example: chmod <permissions> <directoryname>

This command grants read, write, and execute permissions to the owner of the directory and read and execute permissions to the group and others.

## Summary

In this Linux project, I showcased fundamental file and directory management tasks. I demonstrated how to effectively check and display detailed information about files and directories using the 'ls' command. Furthermore, I explained the permissions string, which signifies the access rights for users, groups, and others, allowing for a clear understanding of security settings. The project also included instructions on how to modify file and directory permissions using both numeric and symbolic representations, providing users with versatile options. Additionally, I emphasized that changing permissions for hidden files is no different from regular files, making the process consistent and straightforward. Overall, this project serves as a concise and practical guide to mastering file and directory permissions in the Linux operating system.