# File permissions in Linux

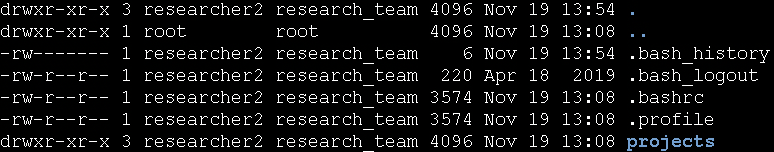
## Project description

My task is to examine existing permissions on the file system. I need to determine if the permissions match the authorization that should be given. If they do not match, I need to modify the permissions to authorize the appropriate users and remove any unauthorized access.

## Check file and directory details

## 

The ls command is used for listing files in the current directory. Option -l is used for including the permissions and option -a for including hidden files (files starting with dot).



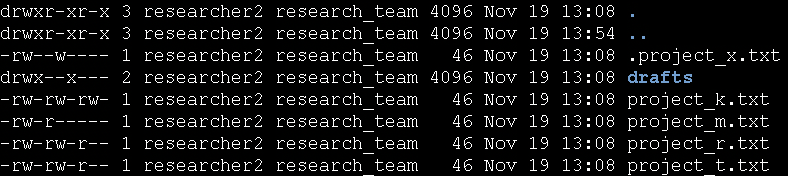
## Describe the permissions string

The first column in the output contains 10-character strings that represent permissions for user (owner), group and others for individual files in the listed directory. The first character indicates the file type. If it contains the letter ‘d’, then it is a directory, otherwise it is a regular file. The next 3 characters represent permissions for the user or the owner of the file. The letter ‘r’ stands for read, ‘w’ stands for write and ‘x’ stands for execute. If we look at the file named ‘projects’ we can see that it is a directory and the owner of this directory can read it, write to it and execute it. The next 3 characters represent permission for the group the owner belongs to. The projects directory can be read and executed by the group. The hyphen means that the action is not allowed. In this case writing to a directory is not allowed. The last three characters represent permissions for other users. The permissions for other users are the same as for the group. So other users are able to read the content of the directory and execute it.

## 

## Change file permissions

The organization does not allow others to have write access to any files in the projects directory. Again I’ll use the ls -la command to list all files in the projects directory and their permissions.

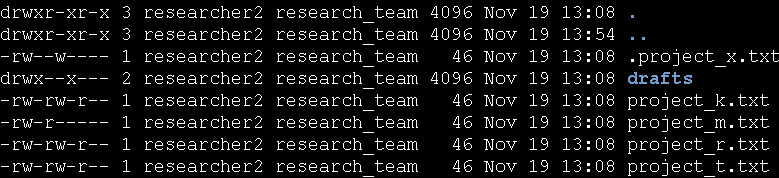


As we can see, there is only one file where others have the write permission. It is the project\_k.txt file. We can easily change the permissions by executing the following command.



Chmod stands for change mode and is used for changing permissions. The ‘o’ represents others, ‘-’ is used for removing permission (‘+’ on the other hand is used for adding permission) and ‘w’ stands for the write permission. The last argument of this command is the file name we want to change.

Now if we list the files again, we can see that there are no file where others have the write permission.



## Change file permissions on a hidden file

The research team has archived .project\_x.txt, which is why it’s a hidden file. This file should not have write permissions for anyone, but the user and group should be able to read the file. Right now the user has read and write permission, the group has only write permission and others have no permissions. To change these permissions, I’ll use the following command:



‘u-w’ removes the write permission for the user.

‘g-w’ removes the write permission for the group.

‘g+r’ adds the read permission for the group.

Now the permissions look like this:



## Change directory permissions

The files and directories in the projects directory belong to the researcher2 user. Only this user should be allowed to access the drafts directory and its contents.



As we can see the group also has access to execute the directory. We can change this by running the following command:



‘g-x’ removes the execute permission for the group.



## Summary

Using different Linux commands, I inspected and adjusted file permissions within the projects directory. Identified files or directories, modified them based on the organization’s policies, ensured appropriate permissions for hidden files like ‘.project\_x.txt’ and restricted access to the ‘drafts’ directory exclusively for the ‘researcher2’ user.