**Day 6 Task: File Permissions and Access Control Lists**

Today is more on Reading, Learning and Implementing File permissions

The concept of Linux File permission and ownership is important in Linux. Here, we will be working on Linux permissions and ownership and will do tasks on both of them. Let us start with the Permissions.

**Create a simple file and do ls -ltr to see the details of the files refer to Notes**

Each of the three permissions are assigned to three defined categories of users. The categories are:

owner — The owner of the file or application.

"chown" is used to change the ownership permission of a file or directory.

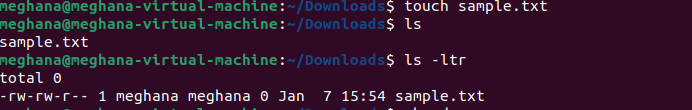
group — The group that owns the file or application.

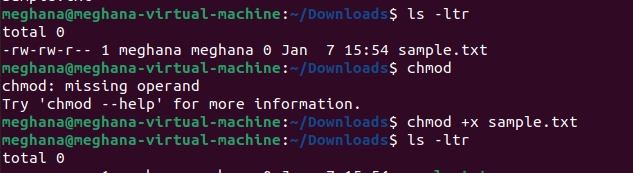
"chgrp" is used to change the gropu permission of a file or directory.

others — All users with access to the system. (outised the users are in a group)

"chmod" is used to change the other users permissions of a file or directory.

As a task, change the user permissions of the file and note the changes after ls –ltr





**Write an article about File Permissions based on your understanding from the notes.**

Linux as a multi-operating system sets permissions and ownership to ensure security for a file and directories of the users.

And it also allows to change and modify the permissions to a set of people as per the requirements.

Linux File Permissions

The basic Linux permissions model works by associating each system file with an owner and a group and assigning permission access rights for three different classes of users:

The file owner.

The group members.

Others (everybody else).

File ownership can be changed using the chown and chgrp commands.

Three file permissions types apply to each class of users:

The read permission.

The write permission.

The execute permission.

This concept allows you to control which users can read the file, write to the file, or execute the file.

To view the file permissions, use the ls command: ls -l file\_name

-rw-r--r-- 12 linuxize users 12.0K Apr 28 10:10 file\_name

|[-][-][-]- [------] [---]

| | | | | | |

| | | | | | +-----------> 7. Group

| | | | | +-------------------> 6. Owner

| | | | +--------------------------> 5. Alternate Access Method

| | | +----------------------------> 4. Others Permissions

| | +-------------------------------> 3. Group Permissions

| +----------------------------------> 2. Owner Permissions

+------------------------------------> 1. File Type

The first character indicates the file type. It can be a regular file (-), directory (d), a symbolic link (l),

or other special types of files. The following nine characters represent the file permissions,

three triplets of three characters each. The first triplet shows the owner permissions, the second one group permissions,

and the last triplet shows everybody else permissions.

In the example above (rw-r--r--) means that the file owner has read and write permissions (rw-),

the group and others have only read permissions (r--).

File permissions have a different meaning depending on the file type.

Each of the three permission triplets can be constructed of the following characters and have different effects,

depending on whether they are set to a file or to a directory:

Examples :

chmod +rwx filename to add permissions.

chmod -rwx directoryname to remove permissions.

chmod +x filename to allow executable permissions.

chmod -wx filename to take out write and executable permissions.

**Read about ACL and try out the commands getfacl and setfacl**

**setfacl** command in Linux is used to set access control lists (ACLs) of files and directories. ACL helps to create an additional, more flexible permission mechanism for the file system. It allows us to provide permission for any user or group to any disk resource.

**getfacl** command is used to get file access control lists. For each file, getfacl displays the file name, owner, the group, and the Access Control List (ACL). If a directory has a default ACL, getfacl also displays the default ACL.

