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.

1. **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 group permission of a file or directory.
- others — All users with access to the system. (outside 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`

DAY 6 ASSIGNMENT

```
ubuntu@ip-172-31-34-30:~$ touch sample.txt
ubuntu@ip-172-31-34-30:~$ ls -lrth sample.txt
-rwxr-xr-x 1 ubuntu ubuntu 22 Mar 24 15:59 sample.txt
ubuntu@ip-172-31-34-30:~$
ubuntu@ip-172-31-34-30:~$
ubuntu@ip-172-31-34-30:~$
ubuntu@ip-172-31-34-30:~$ chmod 755 sample.txt
ubuntu@ip-172-31-34-30:~$
ubuntu@ip-172-31-34-30:~$ ls -lrth sample.txt
-rwxr-xr-x 1 ubuntu ubuntu 22 Mar 24 15:59 sample.txt
ubuntu@ip-172-31-34-30:~$
```

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

→ File permissions are a fundamental aspect of Unix-like operating systems, such as Linux. They determine who can access files, and what actions they can perform on them. Understanding file permissions is crucial for system administrators, developers, and users alike to ensure the security and integrity of their systems and data.

File Permission Basics:

- Each file and directory in Unix-like systems has three sets of permissions: one for the owner, one for the group, and one for others.
- Permissions are represented by a 10-character string: the first character indicates the file type, and the next nine characters represent the permissions for the owner, group, and others.
- The permissions consist of three parts: read (`r`), write (`w`), and execute (`x`).

File Types:

- The first character of the permission string represents the file type.
- Common file types include `-` for regular files, `d` for directories, `l` for symbolic links, and more.

Permission Notations:

- Permissions can be represented in symbolic notation (e.g., `rwx`) or numeric notation (e.g., `755`).
- Symbolic notation includes `r` for read, `w` for write, and `x` for execute, along with special characters like `-` for no permission.

- Numeric notation assigns a value to each permission (read = 4, write = 2, execute = 1) and sums them up to form a three-digit number.

Changing File Permissions:

- File permissions can be changed using the `chmod` command.
- The `chmod` command accepts symbolic or numeric notation to modify permissions.
- For example, `chmod u+x file.txt` adds execute permission for the owner, while `chmod 644 file.txt` sets read and write permissions for the owner and read-only permissions for the group and others.

Viewing File Permissions:

- File permissions can be viewed using the `ls` command with the `-l` option.
- The `ls -l` command displays detailed information about files and directories, including permissions, ownership, size, and modification date.
-

3. **Read about ACL and try out the commands `getfacl` and `setfacl`**

→ Access Control Lists (ACLs) extend the standard UNIX file permissions and provide more fine-grained control over access to files and directories. They allow you to set permissions for multiple users and groups beyond the owner and group associated with the file.

#Note:- before using use below install the package using below command

sudo apt install acl

`getfacl`: This command is used to view the ACL (Access Control List) entries for files and directories. It displays the permissions associated with the file or directory, including any additional ACL entries.

```
ubuntu@ip-172-31-34-30:~$ getfacl sample.txt
# file: sample.txt
# owner: ubuntu
# group: ubuntu
user::rwx
group::r-x
other::r-x
```

DAY 6 ASSIGNMENT

`setfacl`: This command is used to set ACL entries for files and directories. It allows you to define specific permissions for users and groups, in addition to the standard owner, group, and other permissions.

```
setfacl: Option -m: Invalid argument near character 3
ubuntu@ip-172-31-34-30:~$ setfacl -m u:ubuntu:rwx sample.txt
ubuntu@ip-172-31-34-30:~$ getfacl sample.txt
# file: sample.txt
# owner: ubuntu
# group: ubuntu
user::rwx
user:ubuntu:rwx
group::r-x
mask::rwx
other::r-x
```