

Lab- 03 Manual

Lab Instructor: Muhammad Haroon

Department of Software Engineering

Email: muhammad_haroon@umt.edu.pk

Lab 03: File Permissions

1. Objective:

Learn how to view, modify, and understand file permissions in Linux using terminal commands.

2. Introduction to File Permissions

In Linux, every file has associated permissions that define who can read, write, and execute the file. Permissions are represented in three categories:

- **Owner (u):** The user who owns the file.
- **Group (g):** The group associated with the file.
- **Others (o):** Other users.
- **All(a):** All Users.

Permission Types:

- **Read (r):** Permission to view the contents of the file.
- **Write (w):** Permission to modify the file.
- **Execute (x):** Permission to execute the file as a program.

Example of File Permissions:

```
-rwxr-xr--
```

- **rwX** (Owner): Read, Write, Execute
- **r-x** (Group): Read, Execute
- **r--** (Others): Read only

Viewing File Permissions

To view the permissions of a file or directory, use the `ls -l` command.

Lab Manual: Operating System

```
ls -l <file/directory_name>
```

Example:

```
ls -l sample.txt
```

Modifying File Permissions

To modify permissions, we use the `chmod` command.

Syntax:

```
chmod [permissions] <file_name>
```

Using chmod with Symbolic and Numeric Methods

a. Symbolic Method

The symbolic method uses `u` (user), `g` (group), and `o` (others) to assign permissions.

Examples:

- Grant execute permission to the owner:

```
chmod u+x sample.txt
```

- Remove read permission from others:

```
chmod o-r sample.txt
```

b. Numeric Method

Permissions can also be represented by numbers, where:

- Read = 4
- Write = 2
- Execute = 1

Each permission set (owner, group, others) is the sum of these numbers.

Example:

```
chmod 754 sample.txt
```

Explanation:

- **7** (Owner): Read, Write, Execute

Lab Manual: Operating System

- 5 (Group): Read, Execute
- 4 (Others): Read only

Changing Ownership with `chown`

The `chown` command changes the ownership of a file or directory.

```
chown <owner>:<group> <file_name>
```

Example:

```
chown user1:usergroup sample.txt
```

Tasks

1. Create a new file named `testfile1.txt` and display its current permissions.
2. Grant execute permission to the owner of `testfile1.txt` using symbolic mode.
3. Set the permissions of `testfile1.txt` to 644 using numeric mode.
4. Create a directory named `testdir` and set its permissions to 755.
5. Change the group ownership of `testfile1.txt` to `developers`.
6. Remove write permission from others for `testfile2.txt` using symbolic mode.
7. Change both the owner and group of `testfile2.txt` to `user2:developers`.
8. Revoke execute permission for group and others on `testfile1.txt`. What permissions remain for each category?
9. Create a file named `shared.txt`, set its group to `developers`, and give read/write permissions to the group, with no permissions for others. What are the final permissions?