**Experiment No: 02**

**Experiment Name: Managing File Permissions in Unix/Linux**

**Objective:** To understand and practice checking and changing file permissions in Unix/Linux.

**Intended Learning Outcome**

- Gain knowledge about file permissions and ownership in Unix/Linux.
- Learn how to view, interpret, and modify file permissions using both symbolic and absolute modes.

**Expected Skills**

- Basic familiarity with the Ubuntu Operating System.
- Ability to create files and execute basic commands in the Ubuntu terminal.

**Tools Required**

- Ubuntu Operating System

**Theory of this Experiment:**

In Unix/Linux, file permissions and ownership provide secure file access. Each file or directory has associated permissions for three types of users:

1. **Owner**: Permissions for the file's owner.
2. **Group**: Permissions for users in the group associated with the file.
3. **Other** (World): Permissions for all other users.

Permissions are grouped in sets of three:

- **Read (r)**: Permission to view the file's contents.
- **Write (w)**: Permission to modify the file.

- **Execute (x)**: Permission to run the file as a program (for executable files).

**Step-01: Preliminary Commands for Checking Permissions**

- **View Current File Permissions:**

  *$ ls -l*

  **Sample Output:**

  -rwxr-xr-- 1 kawsar users 1024 Oct 27 10:10 myfile

  drwxr-xr--- 1 kawsar users 1024 Oct 27 10:10 mydir

- **Owner** (rwx): Has read, write, and execute permissions.
- **Group** (r-x): Has read and execute permissions.
- **Other** (r--): Has read-only permission.

**Step-02: Changing Permissions**

Permissions can be modified using the **chmod** command in two modes: symbolic and absolute.

**Using chmod in Symbolic Mode**

Symbolic mode allows adding, removing, or setting specific permissions using symbols.

- **Category**: u (user/owner), g (group), o (others), a (all).
- **Operation**: + (add), - (remove), = (set exact permissions).
- **Permissions**: r (read), w (write), x (execute).

**Examples**

- **Remove write and execute permissions from the user:**

  *$ chmod u-wx myfile*

- **Add read and write permissions for the group:**

  $ chmod g+rw myfile

- **Set read, write, and execute permissions for the group:**

  $ chmod g=rwx myfile

**Step-03: Using chmod in Absolute (Octal) Mode**

- **Octal mode represents permissions numerically:**

  r = 4, w = 2, x = 1. Add up values for each category (Owner, Group, Others).

**Examples**

1. **Assign permissions 764 to file1:**

   $ chmod 764 file1

- **Owner** (7): All permissions (rwx).
- **Group** (6): Read and write (rw-).
- **Others** (4): Read only (r--).

2. **Assign permissions 750 to mydir:**

   $ chmod 750 mydir

- **Owner** (7): All permissions (rwx).
- **Group** (5): Read and execute (r-x).
- **Others** (0): No permissions (---).

**Lab Work:**

i. **Create files and directories**
ii. **Check initial permissions**
iii. **Use symbolic mode to change permissions ( Remove the write permission)**
iv. **Use absolute mode to set permissions**