**192311035**

**M.RAVI SAI VINAY**

# EXPERIMENT-40

**Illustrate the various File Access Permission and different types of users in Linux.**

## AIM:-

The aim is to illustrate the various file access permissions and different types of users in **Linux**, along with how they are implemented to control file access and security.

## ALGORITHM:-

1. Types of Users in Linux:
    - Owner (User): The user who owns the file.
    - Group: A set of users that share the same access rights to the file.
    - Others: All users who are not the owner and do not belong to the file's group.
2. File Permissions:
    - Read (r): The ability to read the contents of the file.
    - Write (w): The ability to modify the file's contents.
    - Execute (x): The ability to run the file if it is executable.
3. Permission Representation:
    - Symbolic Representation: A string of 10 characters such as -rwxr-xr--.
        - First character: File type (- for regular file, d for directory).
        - Next three characters: Owner permissions.
        - Next three characters: Group permissions.
        - Last three characters: Other users' permissions.
    - Numeric (Octal) Representation: A three-digit number (e.g., 755) representing the permissions for owner, group, and others.

- Read = 4
- Write = 2
- Execute = 1
- Combine these values for each user category to set permissions.

4. Changing Permissions:
    - chmod command is used to change file permissions.
        - chmod 755 file.txt: Sets read, write, and execute permissions for the owner, and read and execute permissions for the group and others.
    - chown command is used to change file owner and group.
5. Listing Permissions:
    - ls -l shows detailed file information, including file permissions and user/group ownership.

## PROCEDURE:-

1. Identify the type of users in Linux: **Owner (User), Group, and Others**.
2. Understand the file access permissions for **read (r), write (w), and execute (x)**.
3. Use the **chmod** command to set appropriate permissions.
4. Use **ls -l** to display file permissions and owner details.
5. Use **chown** to change ownership of files and groups.

## CODE:-

```
#!/bin/bash


# Step 1: Create a file and add content

echo "==================================="

echo "Step 1: Creating file 'example.txt' with content..."
```

```
touch example.txt

echo "Sample content for permissions testing" > example.txt


# Step 2: View the initial file permissions

echo "==================================="

echo "Step 2: Initial file permissions of 'example.txt'..."

ls -l example.txt


# Step 3: Change file permissions to 754

echo "==================================="

echo "Step 3: Changing permissions of 'example.txt' to 754..."

chmod 754 example.txt

ls -l example.txt


# Step 4: Create a new user (ensure no name conflicts)

NEW_USER="testuser"

echo "==================================="

echo "Step 4: Adding a new user '$NEW_USER'..."

sudo userdel -r $NEW_USER 2>/dev/null  # Clean up if user exists

sudo adduser $NEW_USER


# Step 5: Switch to the new user and test file access

echo "==================================="

echo "Step 5: Testing file access as '$NEW_USER'..."
```

```
su - $NEW_USER -c "ls -l /home/$USER/example.txt && cat /home/$USER/example.txt"


# Step 6: Cleanup (optional) - Remove the new user and file

echo "==================================="

echo "Step 6: Cleaning up..."

rm -f example.txt

sudo userdel -r $NEW_USER


echo "==================================="

echo "Demo completed."
```

## OUTPUT:-

```
Changing the user information for testuser
Enter the new value, or press ENTER for the default
        Full Name []: RAVI
        Room Number []: 321
        Work Phone []: 901******4
        Home Phone []: 970******1
        Other []: specials
Is the information correct? [Y/n] y
info: Adding new user `testuser' to supplemental / extra groups `users' ...
info: Adding user `testuser' to group `users' ...
==================================
Step 5: Testing file access as 'testuser'...
ls: cannot access '/home/root/example.txt': No such file or directory
==================================
Step 6: Cleaning up...
userdel: user testuser is currently used by process 498
==================================
Demo completed.
root@RAVISAIVINAY:~#
```

## RESULT:-

permissions and user access.This set of observations demonstrates how file permissions (rwx for owner, group, and others) control access to files in Linux, as well as how users interact with those permissions.