DAY 10 **DATE:09/05/2025**

NAME: ANNIE JOHN

USER ID:27739

Batch: 25VID0885_DC_Batch4

TITLE: FILE PERMISSIONS - TYPES AND LEVELS

- **FILE PERMISSION**: In Linux, file permissions are rules that determine who can access, modify, or execute files and directories. They are foundational to Linux security, ensuring that only authorized users or processes can interact with your data.
- **Three Basic Permissions: Files vs. Directories**

Permission	Effect on Files	Effect on Directories
Read (r)	Allows viewing the file's content (e.g. `cat`)	Allows listing files in the directory (`ls`)
Write (w)	Allows editing or modifying the file	Allows creating, deleting, or renaming files **in** the directory
Execute (x)	Allows running the file if it's a script/binary	Allows accessing (cd into) the directory and reading metadata

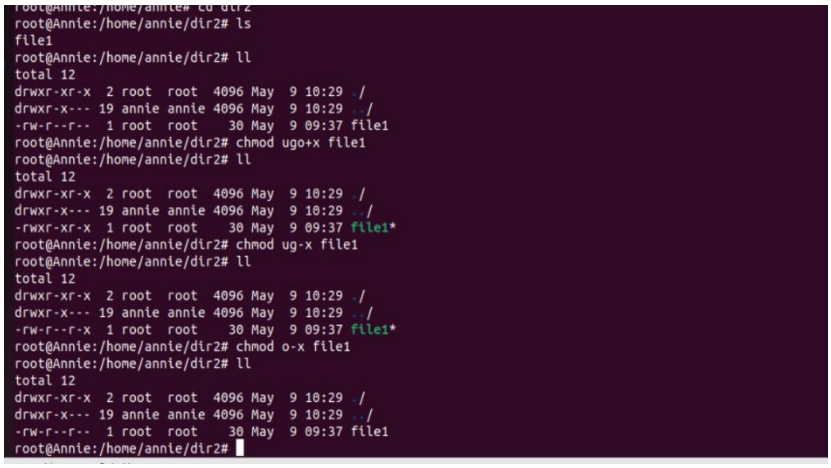
- **Three levels:**

  Permissions are assigned to three categories of users:
  - **User (Owner):** The person who created the file.
  - **Group:** Users belonging to a shared group (e.g., "developers" or "admins").
  - **Others:** Everyone else on the system.
- **chmod – Change Mode (Permissions)**
  - **Basic Syntax**

    chmod [options] [permissions] [file or directory]

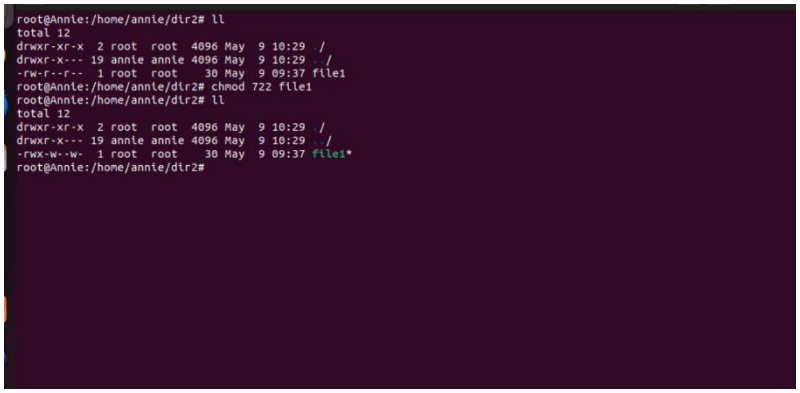
- **Two Methods**
  - Symbolic Method (Human-readable)

Part	Meaning
Who	u (user), g (group), o (others), a (all)
Operator	+ (add), - (remove), = (set exactly)
Permission	r (read), w (write), x (execute)



- # Absolute (Numeric) Method

  This uses **octal numbers** (0–7) to set exact permissions. The octal notation is used to represent file permission in Linux by using three user group by denoting 3 digits i.e. user, group and other users
  - Here's how to permissions are mapped:
    1. **Read (r) = 4**
    2. **Write (w) = 2**
    3. **Execute (x) = 1**



- **Comparison Table**

Feature	Symbolic Method	Absolute Method
Readability	Human-readable (`u+x`)	Concise but less readable (`755`)
Flexibility	Modify specific users/permissions	Sets all permissions at once
Use case	Minor changes, dynamic tweaks	Full setting of permissions

- **User Administration**

  A user is an entity, in a Linux operating system, that can manipulate files and perform several other operations. Each user is assigned an ID that is unique for each user in the operating system. In this post, we will learn about users and commands which are used to get information about the users. After installation of the operating system, the **ID 0 is assigned to the root user** and the IDs 1 to 999 (both inclusive) are assigned to the system users and hence the ids for local user begins from 1000 onwards.
  - In a single directory, we can create 60,000 users. Now we will discuss the important commands to manage users in Linux.
- **Types of users In Linux and their attributes:**

TYPE	EXAMPLE	USER ID (UID)	GROUP ID (GID)	HOME DIRECTORY	SHELL
Super User	Root	0	0	/root	/bin/bash
System User	ftp, ssh, Apache	1 to 499	1 to 499	/var/ftp, etc	/sbin/nologin
Normal User	Karan, Kumar, etc	1000 to 60000	1000 to60000	/home/username	/bin/bash

- **In Linux there are three types of users.**

  1. Super user or root user

  Super user or the root user is the most powerful user. He is the administrator user.

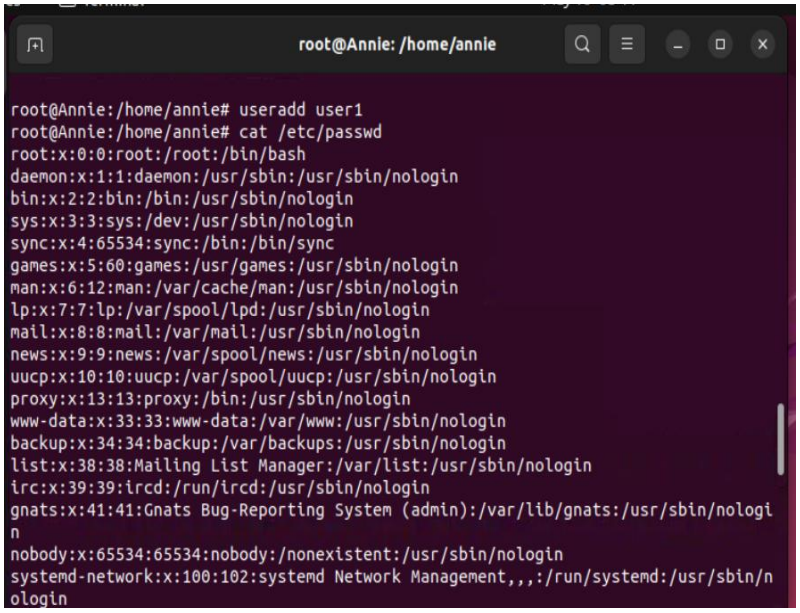
  2. System user

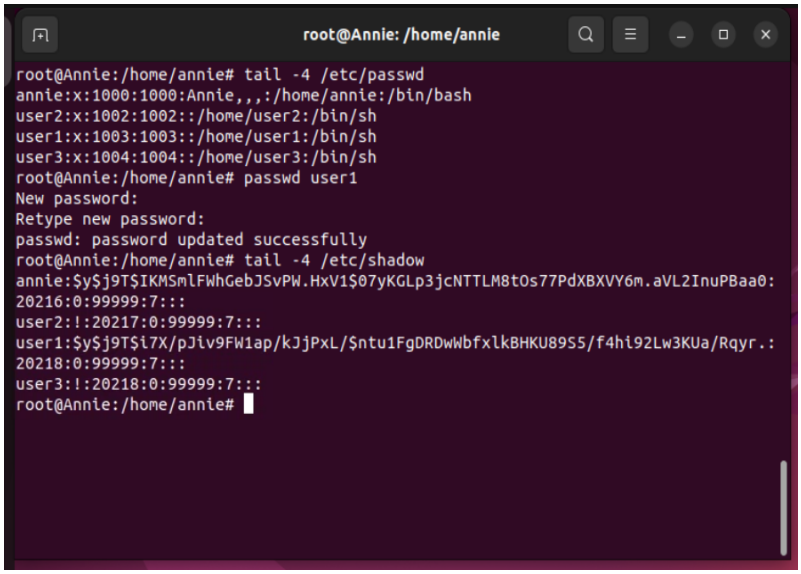
  System users are the users created by the software's or applications. For example, if we install Apache it will create a user Apache. These kinds of users are known as system users.

3. Normal user

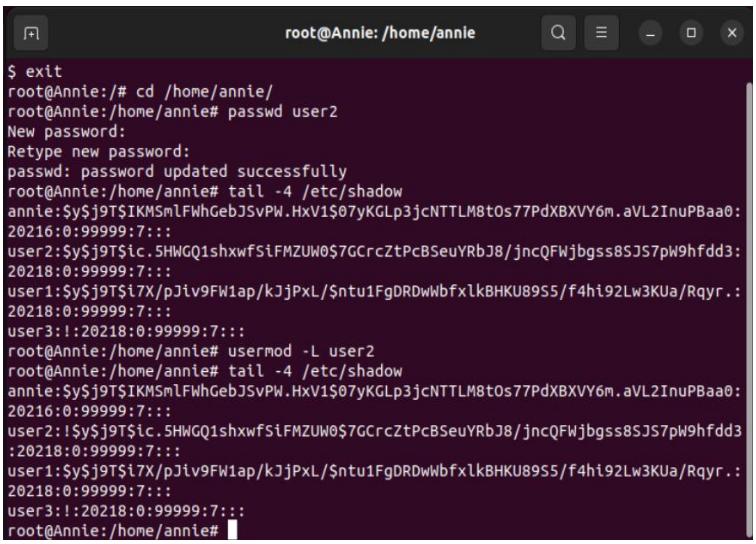
Normal users are the users created by root user. They are normal users like Kiran, Kumar etc. Only the root user has the permission to create or remove a user.

- **/etc/passwd – User Account Information File**
  Purpose: Stores user account details like username, UID, home directory, shell, etc. Readable by all users. Does not store actual passwords (only a placeholder x if shadow passwords are used).
- **/etc/shadow:** Contains **encrypted passwords** for each user. Also includes password aging, expiration, and inactivity policies.
- **Creating a user**
  - # useradd <option> <username> - creating a user
  - passwd <username> - creating password
  - login <username> - login





- usermod -L <username> - lock user



- usermod -U <username> - Unlock user