- user management is a fundamental administrative task used to control who can access the system and what actions they can perform.

1. Types of User Accounts :

- **Root User:** The "superuser" (UID 0) with unrestricted access to all files and commands.
- **System Users:** Accounts (typically UIDs 1–999) created to run specific background services like `mysql`

1. Adding and Removing Users

- **`useradd`**: The low-level utility to create a new user.
  - *Example:* `sudo useradd -m johndoe` (The `-m` flag creates a home directory).
- **`adduser`**: An interactive, more user-friendly script (common on Debian/Ubuntu) that asks for passwords and details automatically.
- **`userdel`**: Deletes a user.
  - *Tip:* Use `sudo userdel -r username` to remove their home directory and files too.

2. Modifying Users

- **`usermod`**:
  - **Add to a group:** `sudo usermod -aG sudo username` (The `-a` is crucial for *appending*, otherwise you'll remove them from all other groups!).
  - **Lock an account:** `sudo usermod -L username`.
- **`passwd`**: Changes a user's password.
  - *Example:* `sudo passwd username`.

3. Group Management

Groups allow you to manage permissions for multiple people at once.

- **`groupadd`**: Create a new group (e.g., `sudo groupadd developers`).
- **`groupdel`**: Remove a group.
- **`groups`**: See which groups a specific user belongs to.

Core commands:

Task	**Command**	**Example**
Add User	`useradd`	`sudo useradd -m username` (creates home directory)
Set Password	`passwd`	`sudo passwd username`
Modify User	`usermod`	`sudo usermod -aG sudo username` (adds to sudo group)
Delete User	`userdel`	`sudo userdel -r username` (removes home directory)
Identify User	`id`	`id username` (shows UID, GID, and groups)
Switch User	`su`	`su - username` (switches to the user's environment)

- file management is handled primarily through a tree-like hierarchical structure starting from the **root (/)** directory.

Core commands:

Task	**Command**	**Example**
List Files	`ls`	`ls -lh` (human-readable sizes and details)
Create File	`touch`	`touch report.txt` (creates a blank file)

Create Folder	`mkdir`	`mkdir -p projects/2024` (creates nested folders)
Copy	`cp`	`cp file.txt /backup/`
Move/Rename	`mv`	`mv old.txt new.txt` (renames or moves)
Delete	`rm`	`rm -rf folder_name` (deletes folder and contents)
View Content	`cat` / `less`	`less large_file.log` (scrollable view)

2.File System Hierarchy (FHS)

Linux organizes system files into standard directories defined by the Filesystem Hierarchy Standard:

- `/bin`: Essential user command binaries (e.g., `ls`, `cp`).
- `/etc`: System-wide configuration files (the "nerve centre" of Linux).
- `/home`: Personal directories for regular users.
- **`/tmp`**: Temporary files, often cleared on reboot.
- `/var`: Variable data like system logs (`/var/log`).
- `/dev`: Special files representing hardware devices.

3. Permissions and Ownership

Every file has an **Owner**, a **Group**, and **Others** (the rest of the world). You can see these by running `ls -l`.

- **Permissions:** Read (r), Write (w), and Execute (x).
- **Changing Permissions:** Use `chmod` (e.g., `chmod 755 script.sh` makes it executable).
- **Changing Ownership:** Use `chown` (e.g., `sudo chown user:group file.txt`).

4. Navigation Shortcuts

- `.` : Represents the **current** directory.
- `..` : Represents the **parent** directory (use `cd ..` to go up).
- `~` : Represents your **Home** directory.
- `pwd` : Prints your current working directory path so you don't get lost.

Process Management(systemctl , journalctl, units (service) :

- A **unit** is any resource systemd can manage, such as services, sockets, ormount points.

Managing Services with `systemctl`

The `systemctl` command is the primary interface for controlling the state of services.

Action	**Command**	**Description**
Start	`sudo systemctl start <name>`	Starts a service immediately.
Stop	`sudo systemctl stop <name>`	Stops a running service.
Restart	`sudo systemctl restart <name>`	Stops and then starts the service again.
Reload	`sudo systemctl reload <name>`	Reloads configuration without stopping the service.
Enable	`sudo systemctl enable <name>`	Sets service to start automatically at boot.
Disable	`sudo systemctl disable <name>`	Prevents service from starting at boot.
Status	`systemctl status <name>`	Shows if a service is running, its PID, and recent logs.
Mask	`sudo systemctl mask <name>`	Completely prevents a service from starting (stronger than disable).

- The **journal** is a centralized binary log that collects data from the kernel and all system services.
- **View all logs:** `journalctl` (starts from the oldest entries).
- **Filter by service:** `journalctl -u <service-name>` (most common for debugging).

- **Real-time logs:** `journalctl -f` (tails the log, showing new entries as they happen).

➢ Volume Management (LVM)

- **Concept:** Creates flexible virtual partitions that can span multiple disks.
- **Workflow:** Physical Volume (`pvcreate`) → Volume Group (`vgcreate`) → Logical Volume (`lvcreate`).
- **Key Benefit:** Easily resize partitions without formatting disks.

➢ The "Big Four" Tools

Tool	Primary Use	Example
`find`	**Locate files** by name, size, or time.	`find / -name "*.log"`
`grep`	**Search text** for specific patterns/words.	`grep "error" app.log`
`sed`	**Modify text** (find & replace) in a stream.	`sed 's/old/new/g' file.txt`
`awk`	**Extract columns** and process data fields.	`awk '{print $1}' data.csv`