

LINUX

What is Linux?

- The Linux Operating System is a type of operating system that is similar to Unix, and it is built upon the Linux Kernel.
- The Linux Kernel is like the brain of the operating system because it manages how the computer interacts with its hardware and resources. It makes sure everything works smoothly and efficiently.
- But the Linux Kernel alone is not enough to make a complete operating system.

Key Characteristics:

- Open source.
- Portability.
- Security.
- Multi-user and multi-tasking.

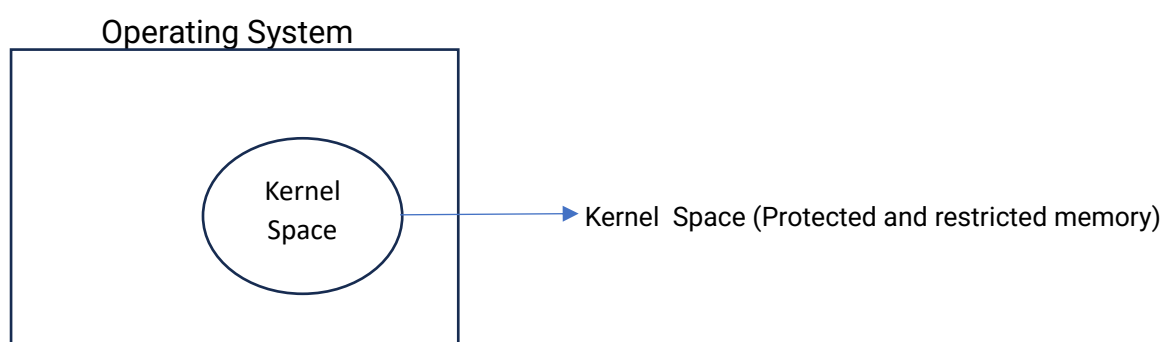
Kernel:

It is a special part of the operating system that controls the CPU Hardware, allocates memory, accesses data, schedules processes, runs the applications and protects them from each other.

Example:

First program loaded on the computer when the computer starts up. Most critical pieces of code in the kernel are loaded into protected areas of memory, so that they can't be overwritten.

Kernel Space & User Space



- It prevents the kernel from terminating and crashing the system.

User Space:

- An application or tool executes in user space.
- By running these applications separate from kernel space, they can't tamper with the kernel resources and cause the system to panic (crash).

Linux Distributions:

- A Linux distribution (or distro) combines the Linux kernel with various software packages to create a complete operating system.

General-Purpose Distributions :

- Ubuntu: User-friendly and beginner-friendly, widely used for desktops and servers.
- Fedora: A cutting-edge distro focused on innovation and open-source principles.
- Debian: Known for its stability and extensive software repository.
- Linux Mint: A popular choice for desktop users due to its ease of use and elegant interface.

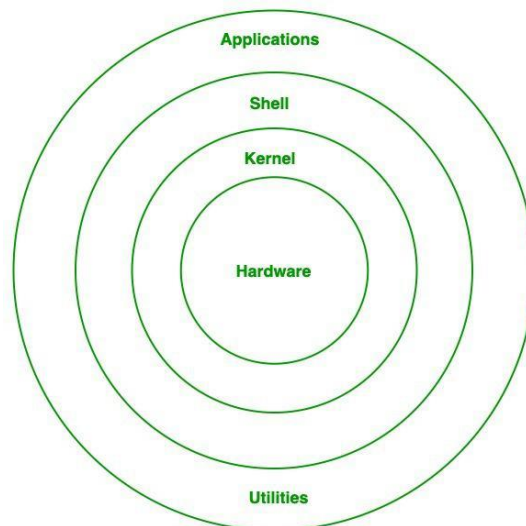
Server and Enterprise Distributions:

- Red Hat Enterprise Linux (RHEL): Widely used in enterprise environments, with commercial support.
- CentOS: A free, community-supported version of RHEL.
- SUSE Linux Enterprise Server (SLES): Focused on enterprise-grade stability and support.

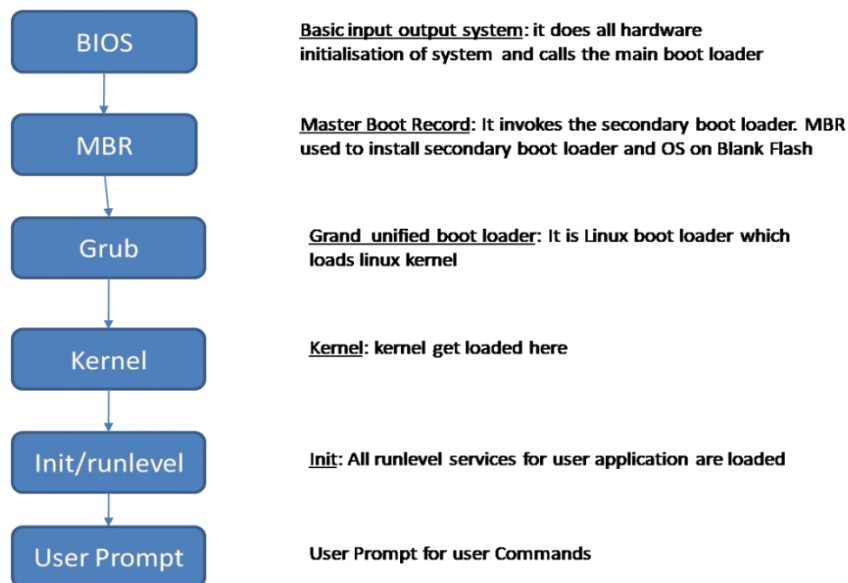
Specialized Distributions:

- Kali Linux: Designed for penetration testing and security auditing.
- Arch Linux: A minimalist distro for advanced users who prefer manual configuration.
- Raspberry Pi OS: Optimized for Raspberry Pi devices.

Linux Architecture:



The main components of Linux operating system are: Application, Shell, Kernel, Hardware, Utilities



- Hardware : It will start the hardware components present in the laptop.
- BIOS: It will Start all the hardware component.
- MBR (Master Boot Record): it will ask for user which os need to boot or start.
- Boot Loader: Selected os requirements will boot by this boot loader.
- Kernel: it manages how the computer interacts with its hardware and resources. It makes sure everything works smoothly and efficiently.
- Init: it will initialize the required application to start.

Commands:

1.To List the files

Ls :It will list the files and folder

Ls -l : it will show the permissions to respective files

Ls -R : lists the contents of directories recursively, meaning it displays the files and subdirectories inside all nested directories.

02 - Navigating the file system

Pwd : It will show present working directory

Cd : Change directory

. :current Directory

Cd .. : navigate to previous directory

3 - Creating and Removing Directories

mkdir – Make a new directory

- mkdir newfolder → Creates newfolder.
- mkdir -p dir1/dir2/dir3 → Creates **nested** directories.

rmdir – Remove empty directories

- rmdir folder_name → Deletes an **empty** folder.
- rmdir -p parent/child → Removes **nested** empty directories.

rm -r – Remove directories and files

- rm -r folder_name → Deletes **non-empty** directories.

4 - Copy, Move, and Delete Files

cp – Copy files and directories

- cp file1 file2 → Copies file1 to file2.
- cp file.txt dir/ → Copies file into dir/.
- cp -r dir1 dir2 → Copies directories **recursively**.

mv – Move or rename files

- mv file1 file2 → Renames file1 to file2.
- mv file.txt dir/ → Moves file into dir/.

rm – Remove files

- rm file.txt → Deletes file.txt.
- rm *.txt → Deletes **all .txt files**.
- rm -r folder → Deletes a **folder with contents**.

5 - Searching for Files

find – Search for files and directories

- find . -name "*.txt" → Finds all .txt files in **current directory**.
- find ~/Documents -name "*data*" → Finds files containing data in **Documents**.

6 - User Roles & Permissions

ls -l – View file permissions

- ls -l → Lists files with **permissions, owner, size, date**.

chmod – Change file permissions

- chmod 755 file.sh → Gives **execute** permission to everyone.

- `chmod u-r file.sh` → Removes **read** permission from the user.

chown – Change file ownership

- `sudo chown root file.txt` → Changes file owner to root.

sudo – Run commands as superuser

- `sudo ls /root` → Lists root directory (requires permission).
- `sudo -s` → Switch to **root user**.
- `exit` → Exit root mode.

7 - Linking Files

ln – Create hard links

- `ln file1 file2` → Creates a **hard link** to file1.

ln -s – Create symbolic (soft) links

- `ln -s file1 link1` → Creates a **soft link** to file1.

8 - Viewing and Searching Text

cat – Display file content

- `cat file.txt` → Shows full content of a file.

head – Show first lines of a file

- `head -n5 file.txt` → Displays **first 5 lines**.

tail – Show last lines of a file

- `tail -n3 file.txt` → Displays **last 3 lines**.

less – View large files page by page

- `less file.txt` → Opens file for **scrolling**.

grep – Search text in files

- `grep "word" file.txt` → Finds word in file.
- `grep -i "word" file.txt` → Case-insensitive search.
- `grep -n "word" file.txt` → Shows **line numbers**.

9 - Editing Text

nano – Open Nano text editor

- `nano file.txt` → Edits file.txt.

vi – Open Vim text editor

- `vi file.txt` → Opens file.txt in Vim.

10 - Working with Archives

tar – Create and extract archives (It will only Archives file but not Compress)

- tar -cvf archive.tar folder/ → Creates a .tar file.
- tar -xvf archive.tar → Extracts .tar file.
- tar -caf archive.tar.gz folder/ → Creates a **compressed .tar.gz**.
- tar -czvf archive.tar.gz dir # Create gzip-compressed tar archive
- tar -xzvf archive.tar.gz # Extract gzip-compressed tar archive
- tar -xf archive.tar.gz -C destination/ → Extracts .tar.gz to destination/.

zip & unzip – Compress and extract zip files (It will compress and Archive)

- zip -r archive.zip folder/ → Creates .zip file.
- unzip archive.zip -d folder/ → Extracts .zip to folder/.
- gzip file # Compress files
- gunzip file.gz # Decompress files

11 - Redirection & Piping

> and >> – Redirect output to a file

- ls > file.txt → Saves output to file (overwrite).
- ls >> file.txt → Appends output to file.

| (Pipe Operator) – Connect commands

- ls | wc -l → Counts files in a directory.
- cat file.txt | grep "word" → Finds "word" in file.

12 - Environment Variables & System Info

env – Display environment variables

- echo \$PATH → Shows system paths.

which – Locate a command

- which ls → Shows location of ls.

uname – Show system details

- uname -a → Displays OS details.
- uname -r → Shows kernel version.

df -h – Disk usage

- df -h → Displays disk usage in **human-readable** format.
- du -h file → Show file size in human-readable format.

- `mount /dev/sdX /mnt # Mount a disk`
- `umount /mnt # Unmount a disk`

free -h – Check memory usage

- `free -h` → Shows RAM usage.

lscpu – Display CPU info

- `lscpu` → Shows processor details.

13 - Package Management (APT for Debian/Ubuntu)

apt – Install & update software

- `sudo apt update` → Updates package lists.
- `sudo apt upgrade` → Upgrades installed packages.
- `sudo apt install tree` → Installs the tree package.
- `apt search package_name` → Searches for a package.

14 - VI Editor Shortcuts

`i` # Enter insert mode

`Esc` # Exit insert mode

`:w` # Save file

`:q` # Quit

`:wq` # Save and quit

`:q!` # Quit without saving

`dd` # Delete a line

`yy` # Copy a line

`p` # Paste copied line

`u` # Undo last change

`/word` # Search for "word"

15 - Network Monitoring Commands

Netstat Command

The netstat command helps monitor network connections, listening services, and routing tables.

- `netstat -a` → Show all active connections and listening ports
- `netstat -t` → Display only TCP connections
- `netstat -u` → Display only UDP connections

- `netstat -l` → Show only listening ports
- `netstat -n` → Display numerical addresses (no DNS resolution)
- `netstat -p` → Show process IDs (PIDs) associated with connections
- `netstat -r` → Show the routing table
- `netstat -i` → Display network interface statistics
- `netstat -s` → Show summary statistics for each protocol
- `netstat -c` → Continuously display network connections (updates dynamically)

Memory Monitoring Commands

Free Command

The `free` command is used to check system memory usage.

- `free` → Show basic memory usage
- `free -h` → Show memory in a human-readable format (MB/GB)
- `free -m` → Display memory in MB
- `free -g` → Display memory in GB
- `free -k` → Display memory in KB (default)
- `free -t` → Show total memory (RAM + swap)
- `free -s 5` → Refresh memory stats every 5 seconds

locate Command

What is locate?

- The `locate` command in Linux is used to find files quickly by searching a pre-built database (`mlocate.db`).
- It is much faster than `find` because it doesn't scan the file system in real-time.

Why use locate?

- It is faster than `find` since it searches an indexed database.
- Useful when you need to find files quickly by name.
- Helps locate files across the entire system without scanning directories manually.

How to use locate?

Basic Syntax:

`locate <filename>`

Examples:

1. Find a file named `test.txt`


```
locate test.txt
```

2. **Count the number of results**

```
locate -c test.txt
```

3. **Ignore case sensitivity**

```
locate -i Test.txt
```

4. **Update the database (if a new file is not found)**

```
sudo updatedb
```

- updatedb updates the locate database, as it doesn't index files in real-time.

sudo Command

What is sudo?

- sudo (short for "superuser do") allows a permitted user to execute a command as another user, typically the root (administrator) user.

Why use sudo?

- Grants temporary superuser privileges without switching users.
- Helps manage system configurations securely.
- Prevents accidental system modifications by restricting root access.

How to use sudo?

Basic Syntax:

```
sudo <command>
```

Examples:

1. **Install a package (Example: curl)**

```
sudo apt install curl # Debian/Ubuntu
```

```
sudo yum install curl # CentOS/RHEL
```

2. **Edit a system file**

```
sudo nano /etc/hosts
```

3. **Restart a service**

```
sudo systemctl restart apache2
```

4. **Switch to the root user**

```
sudo su
```

Security Measures of sudo

- Logs all commands executed using `sudo` for auditing.
- Users must be added to the `sudoers` file (`/etc/sudoers`) to use `sudo`.
- Uses password authentication for security.

Common `sudo` Options and Subcommands:

Command	Description
<code>sudo <command></code>	Runs a command as a superuser.
<code>sudo -i</code>	Opens an interactive root shell (like logging in as root).
<code>sudo -s</code>	Opens a root shell but keeps the current user's environment.
<code>sudo -u <user> <command></code>	Runs a command as another user.
<code>sudo -k</code>	Invalidates the user's cached credentials, forcing re-authentication.
<code>sudo -l</code>	Lists commands the user is allowed to run with <code>sudo</code> .
<code>sudo -v</code>	Refreshes the user's cached credentials without executing a command.
<code>sudo !!</code>	Runs the last command with <code>sudo</code> (useful if you forgot <code>sudo</code>).
<code>sudo visudo</code>	Edits the <code>sudoers</code> file in a safe way.
<code>sudo groupadd <group></code>	Creates a new user group with superuser privileges.
<code>sudo reboot</code>	Reboots the system.

Difference Between `su` and `sudo`

Feature	<code>su</code>	<code>sudo</code>
Full root access?	Yes, if switched to root.	No, only per-command access.
Logs user actions?	No	Yes
Requires root password?	Yes	No, uses user's password
Security risk?	High (full root shell)	Lower (restricted access)

linux Security

- **Firewall Configuration:**
 - `ufw enable` → Enable Uncomplicated Firewall (Ubuntu)
 - `sudo ufw allow 22` → Allow SSH
 - `sudo firewall-cmd --list-all` → View firewall rules (CentOS)

- **SELinux & AppArmor:**
 - Security-enhanced Linux for mandatory access control.
 - getenforce → Check SELinux status
- **System Logs & Auditing:**
 - journalctl -xe → View system logs
 - sudo cat /var/log/auth.log → Check authentication logs

2. User and Group Management

- **Creating Users & Groups**
 - sudo useradd username → Create a new user
 - sudo passwd username → Set password
 - sudo groupadd devteam → Create a group
 - sudo usermod -aG devteam username → Add user to a group
- **Switching Users**
 - su - username → Switch to another user

3. Interprocess Communication (IPC)

- **Types of IPC:**
 - Pipes (| operator)
 - Message Queues (ipcs -q)
 - Shared Memory (ipcs -m)
 - Semaphores (ipcs -s)
- **Examples:**
 - mkfifo mypipe → Create a named pipe

4. Working with Processes

- **Foreground & Background Processes:**
 - top → Show running processes
 - ps aux → View process details
 - kill -9 PID → Terminate a process
- **Process Priorities:**
 - nice -n 10 myprogram → Run with lower priority
 - renice -5 PID → Change priority

5. Basics of Networking

- **Check Network Interfaces**
 - ip a or ifconfig

- **Check Network Connection:**
 - ping google.com
 - traceroute google.com
- **Network Configuration:**
 - /etc/network/interfaces (Debian-based)
 - /etc/sysconfig/network-scripts/ (RHEL-based)

6. Remote Connections (SSH, SCP, SFTP)

- **Connecting to a Remote Server:**
 - ssh user@server-ip
 - ssh -i key.pem user@server-ip
- **Copying Files Securely:**
 - scp file.txt user@server:/home/user/

8. Important Bash Variables

- \$HOME → User's home directory
- \$PATH → Search path for executables
- \$USER → Current logged-in user
- \$SHELL → User's shell

9. WSL 2 (Windows Subsystem for Linux)

- **Installation & Setup:**
 - wsl --install → Install WSL
 - wsl -l -v → List installed distros
- **Running Basic Commands in WSL:**
 - lsb_release -a → Check Linux version
 - wsl --update → Update WSL