Devops

Here's the entire content formatted properly for your notes:

OPERATING SYSTEM (OS)

What is an Operating System? OS is a medium between software and hardware. OS is installed on top of hardware. As a user you are installing an app, app is talking to OS, OS is talking to hardware and vice versa. This is the way of communication.

Communication Flow: User/Application → Operating System (OS) → Hardware (CPU, RAM, Disk, etc.)

1. User installs Application
2. Application talks to OS
3. OS talks to Hardware
4. Hardware sends response back through OS to Application

Example: When you click "Save" in a document:

- App → OS: "Save this file"
- OS → Hardware: "Write data to disk"
- Hardware → OS: "Data written successfully"
- OS → App: "File saved"

WHY LINUX IS USED IN 80% OF DEPLOYMENTS

80% of the time Linux is used and apps are deployed on Linux OS.

4 Key Reasons:

1. Linux is free OS / Open Source Software - No licensing costs, community-driven
2. Very Secure - No antivirus installation required, robust security model
3. Different Distributions are provided - Ubuntu, CentOS, Debian, RedHat, etc.
4. Fast OS - Efficient resource usage, excellent performance

LINUX ARCHITECTURE

Layer Structure:

User Programs/Apps (Your applications) ↓ Compilers/User Processors (System software) ↓ System Libraries (Perform specific tasks) ↓ KERNEL (Heart of Linux) (Core OS functions) ↓ Hardware (CPU, RAM, Disk, etc.)

KERNEL - THE HEART OF LINUX OS

The kernel is the heart of Linux OS. It establishes communication between hardware and your software.

4 Important Aspects of Kernel:

1. Device Management - Controls hardware devices (keyboard, mouse, disk, network)
2. Memory Management - Allocates and manages RAM
3. Process Management - Runs and schedules applications/processes
4. System Calls - Responsible for handling system related calls

System Libraries: System libraries are responsible for performing a task.

Flow: Application → System Library → Kernel → Hardware

FUNDAMENTALS OF SHELL SCRIPTING

What is Shell? Shell is the way you talk to OS. We don't have graphical UI sometimes for our servers in organizations, we have to do it via command line way. This command line way through which we talk to our OS is shell commands.

Why Shell instead of GUI?

- Production systems don't have graphical UI because it comes with some weight
- We use shell commands to directly talk to Linux OS
- Lightweight - No overhead of graphics
- Faster - Direct commands
- Automation-friendly - Can write scripts
- Remote access - Works over SSH

Ubuntu Distribution: Ubuntu is one of the vendors who provide Linux OS.

ESSENTIAL LINUX SHELL COMMANDS

1. ls - List Files and Folders Command: ls Purpose: Used to list files and folders in directory Output: Shows all files and folders in current directory

2. pwd - Present Working Directory Command: pwd Purpose: Shows your current location in the file system Output: Shows the current directory path

3. cd - Change Directory Command: cd Sushmita/ Purpose: Changes directory, goes to Sushmita folder Result: Moves into the "Sushmita" folder Verify: If you do pwd, the value will be inside Sushmita directory

4. cd .. - Go Back One Directory Command: cd .. Purpose: Go back to parent directory Result: Goes back one level in directory structure

5. ls -ltr - Detailed List with Properties Command: ls -ltr Purpose: Provides detailed information about content Output: Shows info, size, when the file/folder is created and other properties

- If starts with 'd' = Directory
- If starts with '-' = File

Flags:

- l = Long format (detailed)
- t = Sort by time
- r = Reverse order (oldest first)

6. touch - Create Empty File Command: touch abc Purpose: Create a file using touch command Result: File called 'abc' is created

7. vi - Create and Edit File Command: vi test Purpose: Create file and write content inside file

Steps to write content:

1. Click on Esc
2. Click on 'i' (enters INSERT mode)
3. Start writing inside a file
4. Once you write, press Esc
5. Type :wq! to save the file
6. Press Enter

Commands:

- i = Insert mode (start editing)

- Esc = Exit insert mode
- :wq! = Write, quit, force (save and exit)
- :q! = Quit without saving

8. cat - Read File Content Command: cat filename Purpose: Used to read the content of the file Result: Displays content of the file

9. mkdir - Create Directory Command: mkdir directory_name Purpose: To create a directory

Example: mkdir Projects Result: Creates a directory named "Projects"

10. rm -r - Remove Directory Command: rm -r directory_name Purpose: Delete directory and all its contents Result: Permanently deletes the folder

Flags:

- r = Recursive (delete folder and everything inside)

Warning: Be careful! This permanently deletes files.

SYSTEM PERFORMANCE COMMANDS

Log in to Linux machine and check CPU and performance of OS:

1. free -g - Check Memory (RAM) Command: free -g Purpose: Tells memory of your server Output: Shows memory usage in gigabytes Displays: Total memory, used memory, free memory, available memory

2. nproc - Number of CPUs Command: nproc Purpose: Tells the number of CPUs Output: Shows number of CPU cores in your server

3. df -h - Disk Space Usage Command: df -h Purpose: Tells the disk size, how much % is used and remaining and all Output: Shows disk usage in human-readable format Displays: Total disk size, used space, available space, usage percentage

4. top - Real-Time System Monitor Command: top Purpose: If I want to see all of this in one place then write top command Output: Shows ALL system stats in one place (live updating)

Displays:

- CPU usage
- Memory usage
- Running processes

- Process IDs (PIDs)
- CPU and RAM per process

Exit: Press 'q'