**Day 1 - Linux**

**Detailed Notes on Linux**

**1. Introduction to Linux**

**History of Linux**

* **1964:** Bell Labs (New Jersey) started working on an open-source operating system.
* **1969:** Dennis Ritchie and Ken Thompson developed **UNIX** (*UNIPLEXED INFORMATION & COMPUTING SERVICES*).
* **1975:** UNIX Version 6 was released, becoming widely popular.
* **1991:** Linus Torvalds, a university student, developed **Linux** based on **MINIX** (created by Andrew Tanenbaum).
* **Key Point:** Linux is **open-source**, meaning it is free to use and modify.

**Linux Distributions (Distros)**

Popular Linux distributions include:  
✅ **RHEL** (*Red Hat Enterprise Linux*)  
✅ **Fedora**  
✅ **Debian**  
✅ **Ubuntu** (*Most popular, third most used OS worldwide*)  
✅ **CentOS**  
✅ **Amazon Linux**  
✅ **Kali Linux** (*Used for security testing and penetration testing*)

**Linux vs. Windows**

| **Feature** | **Linux** | **Windows** |
| --- | --- | --- |
| **Speed** | Faster and lightweight | Slower and resource-heavy |
| **Security** | More secure, less prone to malware | More vulnerable to viruses |
| **Interface** | Primarily **CLI** (Command Line Interface) | Uses **GUI** (Graphical User Interface) |
| **License** | Open-source and free | Proprietary and paid |
| **Core** | Linux is a **kernel**, not a full OS | Windows is a complete OS |

**Important:**

* Linux **is just a kernel**—the Linux operating system is a combination of the **Linux kernel** and **GNU software**.

**2. Features of Linux**

🔹 **Open Source** – Free to use and modify.  
🔹 **Security** – More difficult to hack than Windows.  
🔹 **Multitasking & Multi-user** – Supports multiple users and tasks **simultaneously**.  
🔹 **Fast & Efficient** – Better memory and CPU resource management.  
🔹 **Simplified Updates** – Easy to update all installed software.  
🔹 **Lightweight** – Uses **less RAM** and **disk space** compared to Windows.

**3. Linux File System Hierarchy**

The **Linux file system** follows a hierarchical structure, starting from the **root directory (/)**.

| **Directory** | **Purpose** |
| --- | --- |
| /root | Home directory for the **root user** (Administrator) |
| /home | Home directory for **regular users** |
| /etc | Stores **configuration files** |
| /usr | Default location for **installed software** |
| /bin | Contains commands used by **all users**, including root |
| /sbin | Contains commands **only used by root users** |
| /boot | Contains **bootloader files** |
| /dev | Stores **device files** (e.g., USB, hard disk, keyboard) |
| /opt | Stores **optional software packages** |

**4. Creating Files in Linux**

| **Command** | **Purpose** |
| --- | --- |
| touch filename | Creates an **empty file** and updates the **timestamp** |
| cat filename | Displays file content |
| cat > filename | Creates a new file |
| cat file1 file2 > mergedfile | **Merges multiple files** into one |
| nano filename | Opens the **Nano text editor** to edit a file |
| vi filename | Opens the **VI text editor** to edit a file |
| tac filename | Displays file content in **reverse order** |

**5. Linux Kernel vs. Operating System**

* The **Linux Kernel** is the **core** part of the OS. It manages **hardware**, **memory**, and **processes**.
* The **Linux Operating System** is a combination of:  
  ✅ The **Linux Kernel** (core system)  
  ✅ **GNU software** (user utilities, libraries, and tools)

**6. Linux vs. Windows Architecture**

**Windows Architecture:**

👤 **User** → **Shell** → **OS** → **Hardware**

**Linux Architecture:**

👤 **User** → **Shell** → **Kernel** → **Hardware**

📌 **Key Difference:**

* Linux has fewer layers between the **user** and **hardware**, making it **faster and more efficient**.
* **Windows** uses **folders**, while **Linux** uses **directories**.

**Common Linux Interview Questions and Answers**

**1. What is Linux?**

**Answer:** Linux is an **open-source** operating system based on the **Linux kernel**. It is widely used for its **security, speed, and flexibility**, supporting **multitasking** and **multi-user environments**.

**2. What is the difference between Linux and Windows?**

**Answer:**

* Linux is **open-source, free, and faster** compared to Windows.
* Linux uses **CLI**, while Windows primarily uses **GUI**.
* **Linux is a kernel**, while Windows is a **full OS**.

**3. What are the main features of Linux?**

**Answer:**  
✔️ Open-source  
✔️ Secure and difficult to hack  
✔️ Supports multiple users & multitasking  
✔️ Fast and lightweight

**4. What is the Linux file system hierarchy?**

**Answer:**  
The Linux file system starts with the **root directory (/)**. Some key directories:

* /root → Home directory for the **root user**.
* /home → Home directory for **other users**.
* /etc → Stores **configuration files**.
* /usr → Default location for **installed software**.
* /bin → Contains **commands for all users**.
* /sbin → Contains **commands for root users**.
* /boot → Stores **bootable files**.

**5. How do you create a file in Linux?**

**Answer:** Use commands like:

* touch filename → Creates an **empty file**.
* cat > filename → Creates a **new file** and allows typing content.
* nano filename → Opens the **Nano text editor**.
* vi filename → Opens the **VI text editor**.

**6. What is the difference between /bin and /sbin?**

**Answer:**

* /bin contains **commands for all users**.
* /sbin contains **commands for the root user only**.

**7. How do you update the timestamp of a file in Linux?**

**Answer:** The touch command updates a file's **access, modify, and change times**.

**8. What is the Linux kernel?**

**Answer:** The **Linux kernel** is the core part of the OS that interacts **directly with the hardware** and manages system resources.

**9. What is the difference between a directory and a folder?**

**Answer:**

* In **Linux**, a **directory** is a structure for organizing files.
* In **Windows**, a similar structure is called a **folder**.

**10. How do you concatenate files in Linux?**

**Answer:** The cat command merges files:

cat file1 file2 > mergedfile

**🔹 Summary**

✅ Linux is a **powerful, open-source OS** with a rich history and multiple distributions.  
✅ It is **faster, more secure, and lightweight** compared to Windows.  
✅ The **Linux file system hierarchy** is well-structured, with **specific directories** for different purposes.  
✅ **File creation and manipulation** are done using commands like touch, cat, nano, and vi.  
✅ Understanding the **Linux kernel** is crucial for **DevOps** and **System Administration**.

These notes provide a **solid foundation** for learning Linux and preparing for interviews in **IT, DevOps, and System Administration**. 🚀

**Day 2- Linux**

**1️. Linux Overview**

**🔹 Linux Kernel vs. Operating System**

* **Linux is a kernel, not a full operating system.**
* **An Operating System (OS) is a combination of the Linux kernel and GNU software.**
* **Linux is open-source, secure, multitasking, fast, and lightweight.**

**🔹 Linux Directory Structure**

| **Directory** | **Purpose** |
| --- | --- |
| **/** | **Root directory (top-level directory).** |
| **/home** | **User home directories.** |
| **/etc** | **Configuration files for system settings.** |
| **/usr** | **Default location for installed software.** |
| **/bin** | **Contains essential commands for all users.** |
| **/sbin** | **Contains system administration commands (used by root).** |
| **/boot** | **Contains bootable system files.** |

**2️. File Management in Linux**

**🔹 Creating Files**

| **Command** | **Purpose** |
| --- | --- |
| **touch file1** | **Creates an empty file.** |
| **cat > file1** | **Creates a file with content (Type content & press Ctrl + D to save).** |

**🔹 Displaying File Content**

| **Command** | **Purpose** |
| --- | --- |
| **cat file1** | **Displays the entire content of file1.** |
| **tac file1** | **Displays the content in reverse order.** |
| **less file1** | **Opens the file for paged viewing (scroll with arrow keys).** |
| **head -n 5 file1** | **Displays the first 5 lines of a file.** |
| **tail -n 5 file1** | **Displays the last 5 lines of a file.** |

**🔹 Concatenating and Appending Files**

| **Command** | **Purpose** |
| --- | --- |
| **cat file1 file2 > file3** | **Combines file1 and file2 into file3.** |
| **cat >> file1** | **Appends new content to file1 without overwriting.** |

**🔹 Updating & Checking File Timestamps**

| **Command** | **Purpose** |
| --- | --- |
| **touch file1** | **Updates the access and modification timestamps.** |
| **stat file1** | **Displays detailed metadata (access, modify, and change times).** |

**3️. AWS EC2 Instances**

**🔹 What is EC2?**

* **EC2 (Elastic Compute Cloud) is a virtual machine in AWS.**
* **Used for hosting applications, running scripts, and deploying services.**

**🔹 Key Concepts**

| **Concept** | **Description** |
| --- | --- |
| **Key Pairs** | **Used for secure SSH access.** |
| **.pem File** | **Used for direct SSH login (Linux/macOS).** |
| **.ppk File** | **Used for PuTTY on Windows.** |
| **AMI (Amazon Machine Image)** | **Pre-configured OS images (e.g., Ubuntu, Amazon Linux).** |

**🔹 Connecting to an EC2 Instance**

| **Method** | **Command** |
| --- | --- |
| **Using SSH (Linux/macOS)** | **ssh -i key.pem ubuntu@public-ip** |
| **Using PuTTY (Windows)** | **Convert .pem to .ppk, then connect via PuTTY.** |

**4️. Text Editors in Linux**

**🔹 vi Editor (Powerful, Command-Based)**

| **Command** | **Description** |
| --- | --- |
| **vi file1** | **Opens file1 in the vi editor.** |
| **i** | **Insert mode (for typing text).** |
| **Esc** | **Exit insert mode.** |
| **:wq** | **Save and exit.** |
| **:q!** | **Exit without saving.** |

**🔹 nano Editor (User-Friendly)**

| **Command** | **Description** |
| --- | --- |
| **nano file1** | **Opens file1 in the nano editor.** |
| **Ctrl + X** | **Exit nano.** |
| **Ctrl + O** | **Save changes.** |
| **Ctrl + K** | **Cut a line.** |
| **Ctrl + U** | **Paste a line.** |

**5️. Practical Linux Tips**

| **Task** | **Command** |
| --- | --- |
| **Clear the screen** | **clear or Ctrl + L** |
| **Check file timestamps** | **stat file1** |
| **Concatenate files** | **cat file1 file2 > file3** |
| **Show system info** | **uname -a** |
| **Check disk usage** | **df -h** |

**📝 Interview Questions & Answers**

**1️. What is the difference between Linux and Unix?**

**Answer:**

* **Linux is open-source, while Unix is proprietary.**
* **Linux is widely used for cloud, servers, and embedded systems.**

**2️. Explain the Linux directory structure.**

**Answer:**

* **/root: Admin’s home directory.**
* **/home: User directories.**
* **/etc: Configuration files.**
* **/bin: Essential user commands.**
* **/sbin: System admin commands.**
* **/usr: Installed software.**

**3️. How do you create and display a file in Linux?**

**Answer:**

* **Create: touch file1 (empty) or cat > file1 (with content).**
* **Display: cat file1.**

**4️. What is the difference between > and >> in Linux?**

**Answer:**

* **> overwrites a file (e.g., cat file1 > file2).**
* **>> appends content to a file (e.g., cat file1 >> file2).**

**5️. What is the purpose of the touch command?**

**Answer:**

* **Creates an empty file.**
* **Updates the timestamps of an existing file.**

**✅ Conclusion**

* **Linux is a powerful, open-source OS widely used in servers, cloud computing, and DevOps.**
* **AWS EC2 provides scalable virtual machines, with SSH access via key pairs.**
* **Mastering basic Linux commands and file operations is essential for DevOps and cloud engineers.**
* **Practicing these commands will help prepare for technical interviews in DevOps & cloud roles.**

**🚀 Keep practicing, and good luck with your journey into Linux and AWS!**

**Day 3 – Linux**

**📌 Directory Creation and Navigation**

**🔹 mkdir Command (Make Directory)**

* **Used to create directories in Linux.**
* **Example:**
* **mkdir dir1**

**Creates a directory named dir1.**

**✅ Nested Directory Creation:**

* **Use mkdir -p to create multiple directories in a nested structure.**
* **Example:**
* **mkdir -p dir2/dir3/dir4**

**Creates dir2, dir3, and dir4 in a single command.**

**🔹 cd Command (Change Directory)**

* **Used to navigate between directories.**
* **Examples:**
* **cd dir1** # Moves into dir1
* **cd ..** # Moves up one directory level
* **cd ~** # Moves to the home directory
* **cd -** # Switches to the previous directory

**🔹 pwd Command (Print Working Directory)**

* **Displays the full path of the current working directory.**
* **Example:**
* **pwd**

**Output: /home/user/dir1**

**📌 🔴 Terminal Color Changes:**

* **Normal directories appear in blue.**
* **Executable files appear in green.**
* **Compressed files (e.g., .zip, .tar) appear in red.**
* **Symbolic links appear in cyan.**

**📌 Listing Files and Directories**

**🔹 ls Command (List Files & Directories)**

* **Used to list files and directories in the current directory.**
* **Common options:**
* **ls -l** # Long format with details (permissions, owner, size, date)
* **ls -a** # Lists all files, including hidden files
* **ls -lh** # Human-readable file sizes
* **ls -lt** # Sorts by modification time (latest first)

**📌 🔴 Terminal Color Changes:**

* **Hidden files (.filename) appear in gray.**

**📌 File and Directory Deletion**

**🔹 rmdir Command (Remove Directory)**

* **Removes empty directories only.**
* **Example:**
* **rmdir dir1**

**Deletes dir1 if it is empty.**

**🔹 rm Command (Remove)**

* **Used to delete files and directories.**
* **Common options:**
* **rm file1** # Deletes a file
* **rm -r dir1** # Deletes a directory and its contents
* **rm -rf dir2** # Forcefully deletes `dir2` and all its contents

**📌 ⚠️ Caution:**

* **rm -rf \*/ will delete the entire system. Use with extreme caution!**

**📌 Checking File and Directory Size**

**🔹 du Command (Disk Usage)**

* **Displays the size of a file or directory.**
* **Example:**
* **du -sh dir1**

**Shows the size of dir1 in a human-readable format.**

**🔹 df Command (Disk Free Space)**

* **Displays available disk space on file systems.**
* **Example:**
* **df -h**

**Shows free and used disk space in a human-readable format.**

**📌 File and Directory Copying & Moving**

**🔹 cp Command (Copy Files)**

* **Copies files and directories.**
* **Examples:**
* **cp file1 file2** # Copies file1 to file2
* **cp -r dir1 dir2** # Recursively copies `dir1` to `dir2`

**🔹 mv Command (Move/Rename)**

* **Moves or renames files and directories.**
* **Examples:**
* **mv file1 dir1/** # Moves `file1` into `dir1`
* **mv file1 file2** # Renames `file1` to `file2`

**📌 File Permissions**

**🔹 chmod Command (Change Mode)**

* **Modifies file permissions.**
* **Permissions Representation:** 
  + **r = Read (4)**
  + **w = Write (2)**
  + **x = Execute (1)**

**📌 Examples:**

**chmod 777 file1** # Gives full permissions to everyone

**chmod u+x file1** # Adds execute permission to the user (`u`)

**📌 🔴 Terminal Color Changes:**

* **Executable files appear in green.**

**📌 Concatenation and Redirection**

**🔹 cat Command (Concatenate & Display)**

* **Displays, creates, and merges files.**

**📌 Redirection Operators:**

* **> Overwrites file content.**
* **>> Appends content to a file.**

**📌 Example:**

**cat file1 > file2** # Overwrites file2 with file1's content

**cat file1 >> file2** # Appends file1’s content to file2

**📌 Special Navigation Example**

**Understanding . and ..**

* **. represents the current directory.**
* **.. represents the parent directory.**
* **../../ moves up multiple levels.**

**📌 Example Navigation Command:**

**cd ../../..** # Moves three levels up

**📝 Interview Questions & Answers**

**1️. What is the difference between rmdir and rm -rf?**

**Answer:**

* **rmdir deletes empty directories only.**
* **rm -rf forcefully removes non-empty directories without confirmation.**

**2️. How do you create a nested directory structure?**

**Answer:**

**mkdir -p dir1/dir2/dir3**

**3️. What is the difference between > and >>?**

**Answer:**

* **> overwrites file content.**
* **>> appends content to a file.**

**4️. How do you check the size of a directory?**

**Answer:**

**du -sh dir1**

**5️. How do you list hidden files?**

**Answer:**

**ls -a**

**6️. What is the purpose of the chmod command?**

**Answer:**

* **It changes file permissions (read, write, execute).**

**✅ Conclusion**

* **Mastering Linux file system navigation and command-line tools is essential for DevOps.**
* **Regular practice builds confidence in managing directories, permissions, and file handling.**
* **Understanding Linux fundamentals will greatly help in real-world DevOps roles and interviews.**

**🚀 Keep practicing, and you'll become a Linux expert in no time!**

**Day 4- Linux**

**📌 Links in Linux**

**🔹 What is a Link in Linux?**

* **A link in Linux is a shortcut to a file.**
* **Any changes made to the original file will be reflected in the link.**

**🔹 Types of Links**

**✅ Soft Link (Symbolic Link)**

* **A shortcut to the original file (similar to a Windows shortcut).**
* **Changes in the original file reflect in the soft link.**
* **If the original file is deleted, the soft link becomes broken.**

**📌 Syntax:**

**ln -s <file-name> <link-name>**

**📌 Practical Example:**

**touch file001** # Create a file

**ln -s file001 softlink\_file** # Create a soft link

**echo "Today is Tuesday" >> file001** # Add content to the original file

**cat softlink\_file** # View the content via the soft link

**rm -rf file001** # Delete the original file

**cat softlink\_file** # The soft link is now broken

📌 **🔴 Terminal Color Changes:**

* **Soft links appear in cyan/light blue.**

**✅ Hard Link**

* **A duplicate entry pointing to the same inode (file structure in the filesystem).**
* **Changes in the original file reflect in the hard link.**
* **Even if the original file is deleted, the hard link will still work.**

**📌 Syntax:**

**ln <file-name> <link-name>**

**📌 Practical Example:**

**touch file001** # Create a file

**ln file001 hardlink\_file** # Create a hard link

**rm -rf file001** # Delete the original file

**cat hardlink\_file** # The hard link still works

**📌 find Command**

**🔹 What is the find Command?**

* **The find command is used to search for files and directories in a directory hierarchy.**

**🔹 Syntax:**

**find <directory> -type <file-type> -name <file-name>**

* **<directory>** → The directory to search in (. for the current directory).
* **<file-type>** → f for files, d for directories.
* **<file-name>** → The name of the file or directory to search for.

**🔹 Practical Examples:**

**find . -type f -name file001** # Find a file named 'file001' in the current directory

**find . -type d -name folder1** # Find a directory named 'folder1'

**📌 User Management in Linux**

**🔹 Creating a New User**

**Command:**

**useradd -m <username>**

**Example:**

**useradd -m abhishek**

**🔹 Setting a Password for a User**

**Command:**

**passwd <username>**

**Example:**

**passwd abhishek**

**🔹 Switching Users**

**Command:**

**su <username>**

**Example:**

**su abhishek**

**🔹 Switching to Root User**

**Commands:**

**sudo su**

**sudo -i**

📌 **🔴 Terminal Color Changes:**

* **The root user prompt changes from $ to #.**

**📌 Checking Linux Version & System Information**

**🔹 Check Linux Version**

**Command:**

**uname -v**

**🔹 Check System Details**

**Command:**

**uname -a**

**🔹 Display Server Name**

**Command:**

**hostname**

**🔹 Display IP Address**

**Command:**

**hostname -i**

**🔹 Display Complete Server Name**

**Command:**

**hostname -f**

**📌 Passwordless SSH Connection**

**🔹 Steps to Set Up Passwordless SSH**

**1️. Generate SSH keys on the main server (Linux1)**

**ssh-keygen**

**2️. Copy the public key to the target server (Linux2)**

**ssh-copy-id root@<Linux2-IP>**

**3️. Test the connection**

**ssh root@<Linux2-IP>**

📌 **🔴 Terminal Color Changes:**

* **SSH connections display a different prompt, indicating a remote session.**

**📌 Common Port Numbers**

| **Service** | **Port Number** |
| --- | --- |
| SSH | **22** |
| HTTP | **80** |
| HTTPS | **443** |
| DNS | **53** |
| SMTP | **25** |
| FTP | **21** |

**📝 Interview Questions & Answers**

**1️. What is the difference between a Soft Link and a Hard Link?**

**Answer:**

* **Soft Link:**
  + Points to the original file.
  + Becomes broken if the original file is deleted.
  + **Syntax:** **ln -s <file-name> <link-name>**
* **Hard Link:**
  + Points to the same inode as the original file.
  + Still works even if the original file is deleted.
  + **Syntax:** **ln <file-name> <link-name>**

**2️. How do you find a file in Linux?**

**Answer:**

* Use the find command.
* **Example:**
* find /home -type f -name "file001"

**3️. How do you create a user in Linux?**

**Answer:**

useradd -m abhishek

**4️. How do you set up a passwordless SSH connection?**

**Answer:**

1. **Generate SSH keys:** ssh-keygen
2. **Copy the public key:** ssh-copy-id root@<target-server-IP>
3. **Test connection:** ssh root@<target-server-IP>

**5️. What is the default port number for SSH?**

**Answer:**

* **Port 22**

**6️. How do you check the Linux version?**

**Answer:**

uname -v

**7️. How do you switch users in Linux?**

**Answer:**

su <username>

* **Switch to root user:**
* sudo su

**✅ Conclusion**

* **Understanding Linux commands is crucial for DevOps and system administration.**
* **Practicing user management, file linking, and SSH connections improves server management skills.**
* **Prepare for interviews by reviewing common questions and practicing commands in a real Linux environment.**

🚀 **Keep learning and practicing Linux commands to become a DevOps expert!**

**Extras**

**📌 Viewing File Content (head, tail, cat, tac)**

**🔹 head Command (View First N Lines of a File)**

* **Displays the first 10 lines of a file by default.**
* **Syntax:**
* **head <file-name>**
* **Options:** 
  + **-n <number> → Specifies the number of lines to display.**
* **Example:**
* **head -5 file.txt** # Displays the first 5 lines of file.txt

**🔹 tail Command (View Last N Lines of a File)**

* **Displays the last 10 lines of a file by default.**
* **Syntax:**
* **tail <file-name>**
* **Options:** 
  + **-n <number> → Specifies the number of lines to display.**
  + **-f → Follows a file in real-time (useful for monitoring logs).**
* **Examples:**
* **tail -5 file.txt** # Displays the last 5 lines of file.txt
* **tail -f /var/log/syslog** # Continuously monitors the syslog file

**🔹 cat Command (Concatenate & View File Content)**

* **Displays file content in one go.**
* **Syntax:**
* **cat <file-name>**
* **Options:** 
  + **> file2 → Overwrites an existing file.**
  + **>> file2 → Appends content to an existing file.**
* **Examples:**
* **cat file1 > file2** # Overwrites file2 with file1's content
* **cat file1 >> file2** # Appends file1’s content to file2

**🔹 tac Command (Display File in Reverse Order)**

* **Works like cat, but displays content from bottom to top.**
* **Example:**
* **tac file.txt**

**📌 🔴 Terminal Color Changes:**

* **Normal text files appear in white.**

**📌 Searching for Files & Content (find, grep, locate)**

**🔹 find Command (Search for Files & Directories)**

* **Used to locate files and directories within a directory hierarchy.**
* **Syntax:**
* **find <directory> -type <file-type> -name <file-name>**
* **Options:** 
  + **-type f → Search for files.**
  + **-type d → Search for directories.**
  + **-iname → Case-insensitive search.**
  + **-exec <command> {} \; → Executes a command on found files.**
* **Examples:**
* **find /home -type f -name "file.txt"** # Find a file named 'file.txt' in /home
* **find . -type d -name "folder1"** # Find a directory named 'folder1'
* **find /var/log -type f -name "\*.log"** # Find all log files
* **find . -type f -name "\*.sh" -exec chmod +x {} \;** # Find all shell scripts and make them executable

**🔹 grep Command (Search for Patterns in a File)**

* **Finds specific words or patterns inside files.**
* **Syntax:**
* **grep <pattern> <file-name>**
* **Options:** 
  + **-i → Case-insensitive search.**
  + **-v → Invert match (show lines that don’t contain the pattern).**
  + **-r → Search in multiple files recursively.**
  + **-n → Show line numbers.**
* **Examples:**
* **grep "error" log.txt** # Search for "error" in log.txt
* **grep -i "Warning" log.txt** # Case-insensitive search
* **grep -v "Failed" log.txt** # Show lines \*\*without\*\* "Failed"
* **grep -r "keyword" /var/log/** # Search for "keyword" in all logs

**📌 🔴 Terminal Color Changes:**

* **Matching words appear in red by default.**

**🔹 locate Command (Find Files Quickly)**

* **Uses a pre-built database for fast file searching.**
* **Syntax:**
* **locate <file-name>**
* **Example:**
* **locate file.txt** # Quickly finds the file path

**📌 Ignoring Files & Directories (ignore, .gitignore)**

**🔹 ignore Command**

* **Used to ignore files while searching using grep or find.**
* **Example (Ignore Case Sensitivity in grep):**
* **grep -i "error" log.txt** # Ignore case differences

**🔹 .gitignore File**

* **Prevents specific files from being tracked in Git.**
* **Example (.gitignore file content):**
* **\*.log**
* **node\_modules/**
* **secret.txt**

**📌 Monitoring System & Processes**

**🔹 top Command (View Running Processes)**

* **Displays real-time CPU & memory usage.**
* **Example:**
* **top**

**🔹 ps Command (List Processes)**

* **Shows currently running processes.**
* **Example:**
* **ps aux**

**🔹 kill Command (Stop a Process)**

* **Stops a running process using its process ID (PID).**
* **Example:**
* **kill -9 <PID>**

**📌 Common Port Numbers**

| **Service** | **Port Number** |
| --- | --- |
| **SSH** | **22** |
| **HTTP** | **80** |
| **HTTPS** | **443** |
| **DNS** | **53** |
| **SMTP** | **25** |
| **FTP** | **21** |

**📝 Interview Questions & Answers**

**1️. What is the difference between find and grep?**

**Answer:**

* **find → Searches for files and directories.**
* **grep → Searches for text patterns inside files.**

**2️. How do you view the last 10 lines of a log file in real-time?**

**Answer:**

**tail -f /var/log/syslog**

**3️. How do you find all .log files in /var/log?**

**Answer:**

**find /var/log -type f -name "\*.log"**

**4️. How do you search for all lines in a file that do not contain a word?**

**Answer:**

**grep -v "Failed" log.txt**

**5️. How do you exclude files from Git tracking?**

**Answer:**

* **Use a .gitignore file**

**6️. What is the purpose of the kill -9 command?**

**Answer:**

* **Forces a process to stop immediately.**

**✅ Conclusion**

* **Mastering Linux file searching (find, grep), file viewing (head, tail), and process monitoring (top, ps) is crucial for DevOps.**
* **Regular practice of these commands will improve efficiency in managing servers and debugging issues.**
* **Understanding .gitignore helps in proper Git version control.**

**🚀 Keep learning and practicing these commands to become a Linux expert!**