Assignment : Linux

**Submitted By :- Purv Patel**

**Module 1: Linux server - Understand and use essential tools**

**1. What is the minimum number of partitions you need to install Linux?**

> The minimum number of partitions required to install Linux depends on the distribution and installation method, but in general:

1. One partition – You can install Linux on a single partition (which includes everything: root (/), home, and swap). However, this is not recommended for most setups.
2. Two partitions – A more typical minimal setup includes:

Root (/) – Contains the operating system and user files.

Swap – Used for virtual memory (though it's optional, especially if you have sufficient RAM).

1. Three partitions (Recommended Minimum) – A more traditional approach includes:

Root (/) – Holds the OS.

Swap – Acts as virtual memory.

Boot (/boot) – Some systems require a separate /boot partition for booting.

**2. Explain About Chmod Command**

### > chmod Command in Linux

The chmod (change mode) command is used to change file and directory permissions in Linux.

1. File Permissions

Each file has three permission types:

* Read (r= 4) – View file contents
* Write (w = 2) – Modify file
* Execute (x = 1) – Run file

Permissions are assigned to:

* Owner (u)
* Group (g)
* Others (o)

2. Usage

Syntax:

chmod [options] mode file

Example Commands:

* chmod u+x file.sh → Add execute permission for owner
* chmod g-w file.txt → Remove write permission for group
* chmod 755 file → Owner (rwx), Group (r-x), Others (r-x)

Recursive Change:

chmod -R 755 folder/

3. Numeric Mode

Each permission has a numeric value:

* 7 = rwx (Read, Write, Execute)
* 6 = rw- (Read, Write)
* 5 = r-x (Read, Execute)
* 4 = r-- (Read-only)

Examples:

* chmod 777 file → Full access to all
* chmod 644 file → Owner (rw-), Others (r--)
* chmod 600 file → Only owner can read & write

4. Special Permissions

* SetUID (u+s) → Run as file owner
* SetGID (g+s) → Run as group
* Sticky Bit (+t) → Only owner can delete (e.g., /tmp)

5. Check Permissions

Use ls -l to verify changes.

🔹 Example Output:

-rwxr--r-- 1 user group 1234 file.txt

**3. How to check Linux memory utilization**

### > **Check Memory Utilization in Linux**

You can check memory usage in Linux using various commands:

### **1.** free **Command (Quick Overview)**

free -m

* Displays memory in MB.
* Example output:
* total used free shared buff/cache available
* Mem: 7952 4296 1234 456 2422 3156
* Swap: 2048 512 1536
* **Key Values:**
  + **Total**: Total RAM.
  + **Used**: Currently used RAM.
  + **Free**: Unused RAM.
  + **Available**: Usable RAM for new processes.

### **2.** top **Command (Real-Time Monitoring)**

top

* Shows memory and CPU usage.
* Look for the **"KiB Mem"** and **"KiB Swap"** sections.

### **3.** htop **(Interactive View)**

htop

* A better version of top with a graphical interface.
* If not installed, install it using:
* sudo apt install htop # Debian/Ubuntu
* sudo yum install htop # RHEL/CentOS

### **4.** vmstat **(System Performance)**

vmstat -s

* Displays memory statistics.

### **5.** cat /proc/meminfo **(Detailed Memory Info)**

cat /proc/meminfo

* Provides detailed RAM and swap usage.

### **6.** sar **(Historical Memory Usage)**

sar -r 5 10

* Displays memory usage every 5 seconds, 10 times.

### **7.** ps aux --sort=-%mem **(Top Memory-Consuming Processes)**

ps aux --sort=-%mem | head -10

* Lists top 10 memory-consuming processes.

**4. Use grep to search for specific patterns in files.**

### > **Using** grep **to Search for Patterns in Files**

The grep command is used to **search for specific text patterns in files**. It supports regular expressions, case-insensitive search, and more.

### **Basic Syntax**

grep [options] "pattern" file

### **1. Basic Search**

Search for "error" in log.txt:

grep "error" log.txt

### **2. Case-Insensitive Search**

Use -i to ignore case:

grep -i "error" log.txt

### **3. Search in Multiple Files**

grep "warning" file1.txt file2.txt

### **4. Search Recursively in Directories**

grep -r "TODO" /path/to/dir

### **5. Display Line Numbers**

grep -n "failed" log.txt

### **6. Show Only Matching Word**

grep -o "success" file.txt

### **7. Invert Match (Show Non-Matching Lines)**

grep -v "error" log.txt

### **8. Use Regular Expressions**

Match lines starting with "Error":

grep "^Error" log.txt

Match lines ending with "failed":

grep "failed$" log.txt

### **9. Count Matches**

grep -c "error" log.txt

### **10. Highlight Matches**

grep --color "error" log.txt

### **11. Find Specific Word (Not Substring)**

grep -w "word" file.txt

### **12. Combine with** cat **or** less

cat file.txt | grep "pattern"

less file.txt | grep "pattern"

**5. Get Connecting on a linux server by ssh**

### > **Connect to a Linux Server via SSH**

#### **1. Basic SSH Login**

ssh username@server-ip

#### **2. Connect Using a Custom Port**

ssh -p 2222 username@server-ip

#### **3. SSH Key Authentication (Passwordless Login)**

* Generate a key:
* ssh-keygen -t rsa -b 4096
* Copy key to server:
* ssh-copy-id username@server-ip
* Connect without a password:
* ssh username@server-ip

#### **4. Run Commands Remotely**

ssh username@server-ip "ls -lah /home"

#### **5. Transfer Files with** scp

* **Local → Remote:**
* scp file.txt username@server-ip:/path/
* **Remote → Local:**
* scp username@server-ip:/path/file.txt .

#### **6. SSH Config for Quick Access**

Edit ~/.ssh/config:

Host myserver

HostName 192.168.1.100

User username

Port 2222

Now connect with:

ssh myserver

#### **7. Disconnect from SSH**

exit

or press **Ctrl + D**.

**6. Create 5 files in the /tmp directory, and then use tar and gzip to bundle and compress the files.**

### > **Create and Compress Files in** /tmp **Using** tar **and** gzip

#### **1. Create 5 Files in** /tmp

touch /tmp/file1.txt /tmp/file2.txt /tmp/file3.txt /tmp/file4.txt /tmp/file5.txt

#### **2. Bundle the Files into a tar Archive**

tar -cvf /tmp/files.tar /tmp/file1.txt /tmp/file2.txt /tmp/file3.txt /tmp/file4.txt /tmp/file5.txt

* -c → Create archive
* -v → Verbose (shows progress)
* -f → Specify filename

#### **3. Compress the tar Archive Using gzip**

gzip /tmp/files.tar

This creates /tmp/files.tar.gz.

#### **(Alternative) Create and Compress in One Step**

tar -cvzf /tmp/files.tar.gz /tmp/file1.txt /tmp/file2.txt /tmp/file3.txt /tmp/file4.txt /tmp/file5.txt

* -z → Compress with gzip

#### **4. Verify the Compressed File**

ls -lh /tmp/files.tar.gz

#### **5. Extract the Archive**

tar -xvzf /tmp/files.tar.gz -C /tmp

**7. Describe the root account**

### > **Root Account in Linux**

The **root account** is the **superuser** in Linux with full control over the system.

#### **Key Features:**

* Has **unrestricted** access to all files, commands, and configurations.
* Can **install, modify, and delete** any file or user.
* Identified by **UID 0** and the # prompt in the terminal.
* Must be used **cautiously** to avoid system damage.

#### **Usage:**

* Switch to root:
* su -
* Execute commands as root:
* sudo command
* Disable direct root login (for security):
* sudo passwd -l root

**8. What is shell?**

### > **What is a Shell in Linux?**

A **shell** is a command-line interface (CLI) that allows users to interact with the operating system by executing commands.

#### **Key Functions of a Shell:**

* Accepts **user input** (commands).
* Translates commands into **system actions**.
* Supports **scripting** for automation.

#### **Common Shells in Linux:**

* **Bash (**/bin/bash**)** → Default on most Linux systems.
* **Zsh (**/bin/zsh**)** → More customizable and user-friendly.
* **Sh (**/bin/sh**)** → Basic, lightweight shell.
* **C Shell (**csh**), Korn Shell (**ksh**), Fish Shell (**fish**)** → Alternative shells.

#### **Check Your Shell:**

echo $SHELL

**9. What is Linux?**

### > **What is Linux?**

**Linux** is a free, open-source **operating system (OS)** based on **Unix**. It is widely used for servers, desktops, and embedded systems.

### **Key Features of Linux:**

✅ **Open-source** – Free to use, modify, and distribute.  
✅ **Multi-user & multitasking** – Supports multiple users and processes simultaneously.  
✅ **Secure & stable** – Less prone to malware and crashes.  
✅ **Lightweight & customizable** – Choose from various distributions and desktop environments.  
✅ **Shell & scripting support** – Automate tasks using command-line scripting.

### **Popular Linux Distributions (Distros):**

🐧 **Ubuntu** – User-friendly, great for beginners.  
🐧 **Debian** – Stable and widely used.  
🐧 **CentOS/RHEL** – Enterprise-focused.  
🐧 **Arch Linux** – Minimal and highly customizable.  
🐧 **Kali Linux** – Used for ethical hacking and security testing.

### **Basic Linux Commands:**

* Check OS version:
* uname -a
* List files:
* ls
* Check system status:
* top

**10. What is Bash?**

### > **What is Bash?**

**Bash (Bourne Again Shell)** is the default **command-line interpreter (shell)** for most Linux distributions. It allows users to execute commands, run scripts, and automate tasks.

### **Key Features of Bash:**

✅ **Command Execution** – Runs Linux commands interactively.  
✅ **Scripting** – Supports shell scripting for automation.  
✅ **History & Aliases** – Saves command history and allows shortcuts.  
✅ **Job Control** – Manage background and foreground processes.  
✅ **Wildcards & Variables** – Supports \*, $VAR, and command substitution.

### **Basic Bash Commands:**

* **Check Bash version:**
* bash --version
* **Print working directory:**
* pwd
* **List files:**
* ls
* **Create & run a Bash script:**
* echo -e '#!/bin/bash\necho "Hello, World!"' > script.sh
* chmod +x script.sh
* ./script.sh

**11. You have a new empty hard drive that you will use for Linux. What is the first step you use.**

### > **First Step to Use a New Empty Hard Drive for Linux**

The first step is **partitioning the disk** to prepare it for installation or storage.

#### **1. Identify the New Hard Drive**

Run the following command to list available drives:

lsblk

or

fdisk -l

Look for the new drive (e.g., /dev/sdb).

#### **2. Partition the Disk**

Use fdisk or parted to create partitions:

* Open the disk for partitioning:
* sudo fdisk /dev/sdb
* Create a new partition:
  + Press n (new partition)
  + Select **Primary** or **Extended**
  + Choose partition size
  + Press w (write changes)

#### **3. Format the Partition**

For an **ext4** filesystem (common for Linux):

sudo mkfs.ext4 /dev/sdb1

#### **4. Mount the Drive**

Create a mount point and mount the disk:

sudo mkdir /mnt/newdisk

sudo mount /dev/sdb1 /mnt/newdisk

#### **5. (Optional) Make the Mount Permanent**

Edit /etc/fstab to mount automatically at boot:

echo "/dev/sdb1 /mnt/newdisk ext4 defaults 0 2" | sudo tee -a /etc/fstab

**12. Write the Linux command to show the current working directory.**

**>** The Linux command to show the **current working directory** is:

PWD

**13. write the Linux command to get help with various options.**

> You can use the following Linux commands to get help with various options:

1. Using man (Manual Pages)

man command\_name

🔹 Example:

man ls

(Shows the manual for the ls command.)

2. Using --help (Quick Help)

command\_name --help

🔹 Example:

ls --help

(Displays available options for ls.)

3. Using info (Detailed Documentation)

info command\_name

🔹 Example:

info grep

(Provides detailed documentation for grep.)

4. Using apropos (Search Commands by Keyword)

apropos keyword

🔹 Example:

apropos disk

(Finds commands related to "disk.")

**14. Write the linux comman! to display what all users are currently doing.**

> The Linux command to display what all users are currently doing is:

w

✅ w shows logged-in users and their activities, including:

* Username
* Login time
* Active processes
* CPU usage
* Current command being executed

### **Example Output:**

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

john pts/0 192.168.1.10 10:30 5:00 0.10s 0.02s bash

alice pts/1 192.168.1.12 10:45 1:20 0.05s 0.01s vim file.txt

### **Alternative Commands:**

1. who – Shows logged-in users:
2. who
3. users – Lists usernames of logged-in users:
4. users
5. uptime – Shows system uptime and active users:
6. uptime

**15. write the Linux command to get information about the operating system.**

> You can use the following Linux commands to get information about the operating system:

1. Check OS Name and Version

cat /etc/os-release

🔹 Displays details like OS name, version, and ID.

2. Check Kernel Version

uname -r

🔹 Shows the Linux kernel version.

3. Get Full System Information

uname -a

🔹 Displays kernel name, version, system architecture, and more.

4. Check System Hardware and OS Info

hostnamectl

🔹 Provides OS name, kernel, architecture, and hostname.

5. Display Detailed OS and Kernel Info

lsb\_release -a

🔹 Works on Debian-based systems like Ubuntu.

**16. Write the Linux command to create a hard link of a file.**

> The Linux command to create a hard link of a file is:

ln source\_file hard\_link

Example:

ln file.txt hardlink.txt

🔹 This creates a hard link hardlink.txt pointing to file.txt.

Verify the Hard Link:

1️⃣ Check inode numbers (should be the same):

ls -li file.txt hardlink.txt

2️⃣ Both files share the same data until deleted.

**17. Write the Linux command to create a soft link of a file as well as Directory.**

> The Linux command to create a soft link (symbolic link) for a file or directory is:

1. Create a Soft Link for a File

ln -s source\_file soft\_link

🔹 Example:

ln -s file.txt symlink.txt

(This creates symlink.txt, pointing to file.txt.)

2. Create a Soft Link for a Directory

ln -s source\_directory soft\_link

🔹 Example:

ln -s /home/user/documents docs\_link

(This creates docs\_link, pointing to /home/user/documents.)

Verify the Soft Link

ls -l

🔹 Output will show the link with an arrow (->), e.g.:

lrwxrwxrwx 1 user user 9 Mar 3 12:00 symlink.txt -> file.txt

**18. Write the Linux command! to search for specific pattern in a file.**

> The Linux command to search for a specific pattern in a file is:

grep "pattern" filename

Example:

grep "error" log.txt

🔹 This searches for the word "error" in log.txt.

More Useful grep Options:

1️ Ignore case:

grep -i "pattern" filename

2️ Search in multiple files:

grep "pattern" file1.txt file2.txt

3️ Show line numbers:

grep -n "pattern" filename

4️ Search recursively in directories:

grep -r "pattern" /path/to/directory

5️ Show only matching words:

grep -o "pattern" filename

**19. Write the Linux command to show the use of basic regular expressions using grep command.**

### > Using Basic Regular Expressions with grep

The grep command supports Basic Regular Expressions (BRE) to match patterns in files.

1. Match Exact Word

grep "error" file.txt

🔹 Finds the word "error" in file.txt.

2. Match Lines Starting with a Pattern (^)

grep "^Hello" file.txt

🔹 Finds lines that start with "Hello".

3. Match Lines Ending with a Pattern ($)

grep "success$" file.txt

🔹 Finds lines that end with "success".

4. Match Any Single Character (.)

grep "c.t" file.txt

🔹 Matches "cat", "cot", "cut", etc.

5. Match One or More Occurrences (\*)

grep "go\*d" file.txt

🔹 Matches "gd", "god", "good", "goood", etc.

6. Match Zero or One Occurrence (\?)

grep "colou\?r" file.txt

🔹 Matches both "color" and "colour".

7. Match a Rage of Characters ([])

grep "[aeiou]" file.txt

🔹 Finds any vowel in file.txt.

8. Match Digits ([0-9])

grep "[0-9]" file.txt

🔹 Finds any number.

9. Match a Specific Set of Words (\|)

grep "error\|warning\|fail" file.txt

🔹 Matches "error", "warning", or "fail".

10. Escape Special Characters (\)

grep "price\$" file.txt

🔹 Matches the word "price$", treating $ as a normal character.