

**Hardware Networking**

**Understand and Use Essential Tools**

**SUBMITTED TO**

**KAMLESH SHUKLA SIR**

**SUBMITTED BY**

**MANAV BANDHANIA**

**1. Minimum Number of Partitions Needed to Install Linux**

The minimum number of partitions required to install Linux is **one**. However, for better organization and performance, at least two partitions are recommended:

* **Root (/) Partition:** The main partition where the OS and files are stored. It is necessary.
* **Swap Partition (Optional):** Used as virtual memory when RAM is full. It is recommended but not mandatory.

For a more structured system, additional partitions like /home, /boot, and /var can be used.

**2. Explanation of the chmod Command**

The chmod (Change Mode) command in Linux is used to change the **permissions** of files and directories.

**Syntax:**

chmod [options] mode file

**Usage:**

* **Using Numeric Mode:**
  + chmod 755 file.txt → Gives the owner full permissions and others read-execute permissions.
  + chmod 644 file.txt → Owner can read/write, others can only read.
* **Using Symbolic Mode:**
  + chmod u+x file.sh → Adds execute (x) permission to the user (u).
  + chmod g-w file.sh → Removes write (w) permission for the group (g).
* **Recursive Change:**
  + chmod -R 755 /path/to/directory → Changes permissions for all files inside a directory.

**3. How to Check Linux Memory Utilization**

You can check memory usage in Linux using the following commands:

1. **free -h** (Displays free and used memory in human-readable format)
2. free -h
3. **top or htop** (Shows real-time memory usage)
4. top
5. htop # (if installed)
6. **vmstat** (Shows memory statistics)
7. vmstat -s
8. **cat /proc/meminfo** (Detailed memory information)
9. cat /proc/meminfo

**4. Using grep to Search for Specific Patterns in Files**

The grep command is used to search for text patterns in files.

**Basic Syntax:**

grep [options] "pattern" filename

**Examples:**

1. **Search for "error" in a file:**
2. grep "error" logfile.txt
3. **Case-insensitive search:**
4. grep -i "error" logfile.txt
5. **Search in multiple files:**
6. grep "warning" \*.log
7. **Show line numbers:**
8. grep -n "failed" auth.log
9. **Recursive search in directories:**
10. grep -r "config" /etc/

**5. Connecting to a Linux Server via SSH**

SSH (Secure Shell) is used to remotely access and manage Linux servers securely over a network.

**Step 1: Install SSH Client and Server**

* **On Linux:** SSH client is usually pre-installed. To install SSH server:
* sudo apt update && sudo apt install openssh-server # (Debian/Ubuntu)
* sudo systemctl start ssh
* **On Windows:** Use tools like **PuTTY** or **PowerShell (ssh command)`**.

**Step 2: Connect to the Server**

To connect, use the following command:

ssh username@server\_ip

Example:

ssh root@192.168.1.10

If connecting for the first time, you may see a fingerprint verification message. Type **"yes"** to proceed.

**Step 3: Authenticate**

* If password authentication is enabled, enter the password when prompted.
* If using **SSH keys**, the system will authenticate using the private key.

**Step 4: Execute Commands on the Remote Server**

Once logged in, you can run Linux commands as if you were physically present at the server.

**Step 5: Disconnect**

To log out of the remote session, type:

exit

or press Ctrl + D.

**Additional SSH Features:**

* **Use a different port (default is 22):**
* ssh -p 2222 user@server\_ip
* **Copy files using scp (Secure Copy):**
* scp file.txt user@server\_ip:/home/user/
* **Use SSH key authentication (More Secure):**  
  Generate keys:
* ssh-keygen -t rsa

Copy the public key to the server:

ssh-copy-id user@server\_ip

**6. Create 5 Files in /tmp, then Bundle and Compress with tar and gzip**

**Step 1: Create 5 Files in /tmp**

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

**Step 2: Bundle the Files Using tar**

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

* c → Create a new archive
* v → Verbose mode (shows progress)
* f → Specify the file name

**Step 3: Compress the Tar Archive with gzip**

gzip /tmp/files.tar

This creates /tmp/files.tar.gz, a compressed version of the tar archive.

**Step 4: Verify the Archive**

ls -lh /tmp/files.tar.gz

**Step 5: Extract the Files (Optional)**

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

**7. Describe the Root Account**

The **root account** in Linux is the **superuser** with **full administrative privileges** over the system. It has unrestricted access to **all commands, files, and configurations**.

**Characteristics of the Root Account:**

* Can **install, update, or remove** software
* Can **modify system files** and **change user permissions**
* Can **start, stop, or modify system services**
* Has **access to all directories and files**, even those restricted to other users

**How to Use the Root Account?**

1. **Direct Login (Not Recommended):**
2. su -
3. **Using sudo (Recommended):**
4. sudo command

Example:

sudo apt update

**Why Not Use Root Directly?**

* High risk of **accidental system damage**
* Vulnerability to **security breaches**
* Best practice is to use **sudo** instead

**8. What is a Shell?**

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

**Functions of a Shell:**

* Accepts **commands** from users
* Executes **programs and scripts**
* Supports **variables, loops, and scripting**

**Types of Shells in Linux:**

* **Bash (Bourne Again Shell)** – Most common and default in Linux
* **Zsh (Z Shell)** – Advanced shell with additional features
* **Ksh (Korn Shell)** – Combines features of Bash and C Shell
* **Tcsh (TENEX C Shell)** – Based on C Shell

**Example Usage:**

echo "Hello, World!"

ls -l

pwd

**9. What is Linux?**

**Linux** is a **free and open-source operating system** based on the **Unix** architecture. It is widely used in **servers, desktops, mobile devices, and embedded systems**.

**Key Features of Linux:**

* **Open-source:** Anyone can modify and distribute it
* **Multi-user & Multi-tasking:** Multiple users and tasks run simultaneously
* **Highly Secure:** Built-in security features and access controls
* **Customizable:** Various distributions (Ubuntu, CentOS, Arch Linux, etc.)
* **Stability & Performance:** Used in enterprise environments

**Popular Linux Distributions:**

1. **Ubuntu** – User-friendly, popular for desktops
2. **Debian** – Stable and secure, used in servers
3. **Fedora** – Cutting-edge features, backed by Red Hat
4. **Arch Linux** – Minimal and customizable
5. **Kali Linux** – Security-focused, used for penetration testing

**10. What is Bash?**

**Bash (Bourne Again Shell)** is the default **command-line interpreter (CLI) and scripting language** in most Linux distributions. It is an enhanced version of the original **Bourne Shell (sh)** with more features.

**Features of Bash:**

* **Command Execution** – Runs Linux commands
* **Scripting** – Supports automation using scripts
* **Variables & Functions** – Stores and processes data
* **Command History** – Remembers previously executed commands
* **Job Control** – Runs processes in the background

**Basic Bash Commands:**

pwd # Print working directory

ls # List files in a directory

cd /home # Change directory

mkdir test # Create a new directory

rm file.txt # Remove a file

**Bash Scripting Example:**

#!/bin/bash

echo "Hello, this is a Bash script!"

Save it as script.sh, then run:

chmod +x script.sh

./script.sh

**Why is Bash Important?**

* Used for **system administration** and **automation**
* Supports **custom scripts** for repetitive tasks
* Default shell in **Linux and macOS**

**11. First Step When Using a New Empty Hard Drive for Linux**

When setting up a new empty hard drive for Linux, the **first step** is **partitioning and formatting** the drive.

**Step 1: Identify the New Drive**

Use the following command to list available disks:

lsblk

or

fdisk -l

Look for a new drive like /dev/sdb.

**Step 2: Partition the Disk**

Use the fdisk command to create a partition:

sudo fdisk /dev/sdb

Then, follow these steps:

* Press n (new partition)
* Press p (primary partition)
* Choose the partition number and size
* Press w (write changes and exit)

Alternatively, you can use parted or gparted (GUI tool).

**Step 3: Format the Partition**

sudo mkfs.ext4 /dev/sdb1

**Step 4: Mount the Partition**

sudo mount /dev/sdb1 /mnt

**Step 5: Add to /etc/fstab for Auto-Mounting**

Edit /etc/fstab and add:

/dev/sdb1 /mnt ext4 defaults 0 2

**12. Linux Command to Show the Current Working Directory**

pwd

This prints the **absolute path** of the current directory.

Example:

$ pwd

/home/user

**13. Linux Command to Get Help with Various Options**

**1. Using man (Manual Pages)**

To get help on any command:

man command

Example:

man ls

**2. Using --help Option**

For a quick summary of options:

command --help

Example:

ls –help

**3. Using info (More Detailed Docs)**

info command

Example:

info bash

**4. Using whatis (Short Description)**

whatis ls

**14. Linux Command to Display What All Users Are Currently Doing**

w

This shows the list of logged-in users along with their activity.

**Example Output:**

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

john pts/1 192.168.1.5 10:05 0.00s 0.10s 0.02s sshd

alex tty1 :0 09:50 1:15m 0.20s 0.10s bash

* USER → Name of the logged-in user
* TTY → Terminal they are using
* FROM → Source IP (if remote login)
* LOGIN@ → Time when logged in
* WHAT → Current command the user is executing

**Alternative Commands:**

* who → Shows only the logged-in users
* users → Lists usernames of logged-in users
* uptime → Shows system uptime and logged-in users

**15. Linux Command to Get Information About the Operating System**

To check OS details, use the following commands:

**1. Using uname (Basic OS Information)**

uname -a

Example output:

Linux ubuntu 5.15.0-56-generic #62-Ubuntu SMP x86\_64 GNU/Linux

* Linux → Kernel name
* ubuntu → Hostname
* 5.15.0-56-generic → Kernel version
* x86\_64 → Architecture

**2. Using /etc/os-release (Detailed OS Info)**

cat /etc/os-release

Example output:

NAME="Ubuntu"

VERSION="22.04 LTS"

ID=ubuntu

ID\_LIKE=debian

**3. Using lsb\_release (For Debian-based Distros)**

lsb\_release -a

Example output:

Distributor ID: Ubuntu

Description: Ubuntu 22.04 LTS

Release: 22.04

Codename: jammy

**4. Using hostnamectl (For System Details)**

hostnamectl

Example output:

Static hostname: ubuntu-server

Operating System: Ubuntu 22.04 LTS

Kernel: Linux 5.15.0-56-generic

Architecture: x86-64

**5. Using df -h (Check File System & OS Disk Info)**

df -h

Would you like more detailed explanations on any of these? 🚀

**16. Linux Command to Create a Hard Link of a File**

A **hard link** is a duplicate entry of a file that points to the same inode. Changes to one reflect in the other.

**Syntax:**

ln source\_file hard\_link

**Example:**

ln file1.txt hardlink\_file1.txt

This creates hardlink\_file1.txt, which is a hard link to file1.txt. Both share the same inode.

**Verify Hard Links:**

ls -li file1.txt hardlink\_file1.txt

Both files will have the same inode number.

**17. Linux Command to Create a Soft Link of a File and Directory**

A **soft link (symbolic link)** is a pointer to the original file or directory. If the original is deleted, the soft link breaks.

**Creating a Soft Link for a File:**

ln -s file1.txt softlink\_file1.txt

This creates softlink\_file1.txt as a symbolic link to file1.txt.

**Creating a Soft Link for a Directory:**

ln -s /home/user/Documents mydocs

This creates mydocs, a soft link to /home/user/Documents.

**Verify Soft Links:**

ls -l

Output:

lrwxrwxrwx 1 user user 10 Mar 2 12:30 softlink\_file1.txt -> file1.txt

The l at the beginning indicates a symbolic link.

**18. Linux Command to Search for a Specific Pattern in a File**

Use the grep command to search for a pattern inside a file.

**Syntax:**

grep "pattern" filename

**Example:**

grep "error" /var/log/syslog

This searches for the word "error" in /var/log/syslog.

**Additional Options:**

* **Case-insensitive search:**
* grep -i "error" logfile.txt
* **Display line numbers:**
* grep -n "failed" auth.log
* **Search recursively in directories:**
* grep -r "config" /etc/

**19. Using Basic Regular Expressions with grep (Detailed Explanation)**

Regular expressions (regex) help match patterns in text files. The grep command supports regex for advanced searching.

**1. Match Lines Starting with a Word (^)**

grep "^root" /etc/passwd

Finds lines where root appears at the **beginning**.

**2. Match Lines Ending with a Word ($)**

grep "bash$" /etc/passwd

Finds lines where bash appears at the **end**.

**3. Match Any Single Character (.)**

grep "h.t" words.txt

Finds words like hat, hot, hit.

**4. Match Any Character from a Set ([ ])**

grep "f[aeiou]n" words.txt

Finds fan, fin, fun, etc.

**5. Match Any Word with Multiple Characters (\*)**

grep "go\*d" words.txt

Finds gd, god, good, goood, etc.

**6. Match Repeated Patterns (\+ for one or more, \? for zero or one)**

grep -E "fo+bar" words.txt

Finds foobar, foooooobar, etc.

**7. Escape Special Characters (\)**

grep "10\.0\.0\.1" network.log

Finds exact IP 10.0.0.1.

**8. Match Whole Words (\b)**

grep -E "\berror\b" logfile.txt

Finds exact word error without matching errors or errorlog.

**9. Using grep with egrep (-E for Extended Regex)**

grep -E "error|fail|critical" logfile.txt

Finds lines containing error, fail, or critical.