Linux Commands Tutorial (Theory + Practical)

This tutorial is designed **for beginners**. Each section explains **what**, **why**, and **how**, followed by **simple examples** you can try on a Linux terminal.

1. Introduction to Linux

What is Linux?

Linux is an **open-source operating system** used in servers, cloud platforms, DevOps, cybersecurity, and development environments.

Why learn Linux?

- Used in **servers & cloud (AWS, Azure, GCP)**
- Strong **security & permissions**
- Powerful **command-line tools**

Accessing Linux

- Native Linux OS (Ubuntu, Fedora)
- Windows: **WSL**, **Git Bash**, **VMware**, **VirtualBox**

2. File Operations (Theory + Practical)

2.1 Viewing Current Location

Command: pwd

Purpose: Shows your current directory

```
pwd
```

Output example:

```
/home/user
```

2.2 Listing Files and Directories

Command: `ls`

```
ls
```

Common options:

- `ls -l` → long listing (permissions, owner, size)
- `ls -a` → shows hidden files

```
ls -la
```

2.3 Creating Files

Command: `touch`

```
touch file1.txt
```

Creates an empty file.

2.4 Creating Directories

Command: `mkdir`

```
mkdir myfolder
```

Create multiple directories:

```
mkdir dir1 dir2 dir3
```

2.5 Copy Files & Directories

Command: `cp`

```
cp file1.txt file2.txt
```

Copy directory:

```
cp -r myfolder backupfolder
```

2.6 Move / Rename Files

Command: `mv`

```
mv file1.txt newfile.txt
```

Move file to directory:

```
mv newfile.txt myfolder/
```

2.7 Delete Files & Directories

Command: `rm`

```
rm file1.txt
```

Delete directory:

```
rm -r myfolder
```

⚠ **Caution:** Deleted files cannot be recovered easily.

2.8 Viewing File Content

Commands: `cat`, `less`, `more`

```
cat file1.txt
less file1.txt
```

3. File Permissions (Theory + Practical)

File permissions control **who can access a file** and **what actions they can perform**. This is one of the most important security features in Linux.

3.1 Why File Permissions Are Important

- Prevent **unauthorized access**
- Protect system files from accidental deletion
- Allow **multi-user environments** to work safely

Example scenario:

- A student should **read** notes
- A trainer should **edit** notes
- Others should not access them at all

Linux permissions make this possible.

3.2 Understanding Permission Structure

Run:

```
ls -l
```

Sample output:

```
-rwxr-xr-- 1 user staff 1024 file.sh
```

Breakdown:

Part	Meaning
`-`	File type
`rwx`	Owner permissions
`r-x`	Group permissions
`r--`	Others permissions

3.3 File Types (First Character)

Symbol	Type
-	Regular file
d	Directory
`l`	Symbolic link

Example:

```
drwxr-xr-x 2 user user 4096 myfolder
```

3.4 Permission Symbols Explained

Symbol	Meaning
r	Read – view file contents
w	Write – modify/delete
x	Execute – run as program

3.5 Read, Write, Execute on Files vs Directories

On Files:

- r → open and read content
- w → edit or delete content
- x → run file as a program

On Directories:

- r → list files (ls)
- w → create/delete files
- x → enter directory (cd)

⚠ Without x, you **cannot enter a directory**, even if r is present.

3.6 Permission Categories

Category	Symbol	Meaning
Owner	u	File creator
Group	g	Group members
Others	o	Everyone else
All	a	u + g + o

3.7 Changing Permissions – chmod (Symbolic Method)

Syntax:

```
chmod [who][operator][permission] filename
```

Operators:

- + add permission
- - remove permission
- = assign permission

Examples:

Give execute permission to owner:

```
chmod u+x file.sh
```

Remove write permission from others:

```
chmod o-w file.txt
```

Give read permission to all:

```
chmod a+r file.txt
```

3.8 Changing Permissions – chmod (Numeric Method)

Numeric values:

Number	Permission
4	Read
2	Write
1	Execute

Add values:

- 7 → 4+2+1 → rwx
- 6 → 4+2 → rw-
- 5 → 4+1 → r-x

Example:

```
chmod 755 file.sh
```

Meaning:

- Owner: rwx
- Group: r-x
- Others: r-x

Common permission sets:

Permission	Use case
777	Full access (unsafe)
755	Scripts, directories
644	Text files

3.9 Recursive Permission Change

Apply permissions to directory and all files inside:

```
chmod -R 755 myfolder
```

3.10 Ownership and Group (chown & chgrp)

Check owner:

```
ls -l
```

Change owner:

```
sudo chown user file.txt
```

Change owner and group:

```
sudo chown user:group file.txt
```

Change group only:

```
chgrp group file.txt
```

3.11 Special Permissions (Basic Awareness)

Permission	Meaning
SUID	Runs with owner privileges
SGID	Runs with group privileges
Sticky bit	Only owner can delete files

Example (sticky bit):

```
chmod +t /shared
```

3.12 Common Beginner Mistakes

❌ Using 777 everywhere

❌ Forgetting execute permission on directories

❌ Running `chmod -R /` (dangerous!)

3.13 Practice Exercises

1. Create a file and check permissions
2. Remove write access for others
3. Make a script executable
4. Change ownership (using sudo)
5. Try accessing without permission

3.2 Checking Permissions

```
ls -l
```

3.3 Changing Permissions (chmod)

Symbolic mode:

```
chmod u+x file.sh
```

Numeric mode:

```
chmod 755 file.sh
```

Meaning of 755:

- Owner: 7 (rwx)
- Group: 5 (r-x)
- Others: 5 (r-x)

3.4 Changing Ownership

```
chown user file.txt
```

Change owner and group:

```
chown user:group file.txt
```

4. System Administration Basics

4.1 User Management

Create user:

```
sudo useradd username
```

Set password:

```
sudo passwd username
```

Delete user:

```
sudo userdel username
```

4.2 Disk Usage

Check disk space:

```
df -h
```

Check folder size:

```
du -sh myfolder
```

4.3 System Information

```
uname -a
top
free -h
```

5. Process Management (Theory + Practical)

5.1 What is a Process?

A process is a **running program** in the system.

5.2 Viewing Processes

```
ps
ps -ef
```

Live view:

```
top
```

5.3 Killing a Process

```
kill PID
```

Force kill:

```
kill -9 PID
```

5.4 Background & Foreground

Run in background:

```
command &
```

View jobs:

```
jobs
```

Bring to foreground:

```
fg
```

6. Archival & Compression

6.1 What is Archival?

Combining multiple files into a single file.

6.2 Creating Archive (tar)

```
tar -cvf archive.tar myfolder
```

6.3 Extract Archive

```
tar -xvf archive.tar
```

6.4 Compression with gzip

Create compressed archive:

```
tar -czvf archive.tar.gz myfolder
```

Extract compressed archive:

```
tar -xzvf archive.tar.gz
```

7. Practice Tasks for Beginners

1. Create a directory and add 3 files
2. Change permissions to read-only
3. Compress the directory
4. Extract it back
5. Check running processes