1. Introduction to Linux

What is Linux?

- Linux is an open-source operating system kernel.
- Most “Linux distributions” (Ubuntu, Debian, Fedora, etc.) package:
  - Linux kernel
  - GNU tools
  - Package manager
  - Desktop environment (optional)

Why Learn Linux?

- Used heavily in:
  - Servers
  - Cloud infrastructure
  - DevOps
  - Cybersecurity
  - Programming environments
- Powerful command-line interface (CLI)
- Highly customizable and scriptable

2. Terminal & Shell

Terminal vs Shell

- **Terminal**: Interface to interact with the system.
- **Shell**: Program that interprets commands (e.g., bash).

Basic Command Structure

Code

```
command [options] [arguments]
```

Example:

Code

```
ls -l /home
```

Important Concepts

- Case-sensitive
- Tab auto-completion
- Command history (↑ arrow)
- Clear screen: `clear`

3. Filesystem Structure

Linux Filesystem Hierarchy

Root directory:

Code

```
/
```

Important directories:

Directory	Purpose
/	Root
/home	User files
/bin	Essential binaries
/etc	Configuration files

Directory	Purpose
/var	Logs & variable data
/usr	User-installed programs
/tmp	Temporary files

Navigation Commands

- `pwd` → Print working directory
- `ls` → List files
- `cd directory` → Change directory
- `cd ..` → Go up one level
- `cd ~` → Go to home directory

4. Working with Files & Directories

Create

- `touch file.txt`
- `mkdir folder`

Delete

- `rm file.txt`
- `rm -r folder`

Copy

- `cp file1 file2`
- `cp -r folder1 folder2`

Move / Rename

- `mv oldname newname`

View Files

- `cat file.txt`
- `less file.txt`
- `head file.txt`
- `tail file.txt`

5. File Permissions

Each file has:

- Owner
- Group
- Others

Permission types:

- r → read
- w → write
- x → execute

Example:

Code

```
-rwxr-xr--
```

Changing Permissions

Numeric method:

Number	Permission
7	rwx
6	rw-

Number	Permission
5	r-x
4	r--

Example:

Code

```
chmod 755 file.sh
```

Change Ownership

Code

```
chown user file
chgrp group file
```

6. Users & Sudo

Root User

- Superuser with full control.
- Use carefully.

sudo

- Temporarily run commands as root.

Code

```
sudo apt update
```

7. Programs & Processes

Running Programs

- Just type program name.

Code

```
python script.py
```

PATH Variable

- Stores directories where executables are searched.

Code

```
echo $PATH
```

Process Management

- `ps` → List processes
- `top` → Real-time process monitor
- `kill PID` → Stop process

8. Input, Output & Redirection

Standard Streams

- stdin (0)
- stdout (1)
- stderr (2)

Redirect Output

Code

```
command > file
command >> file
```

Redirect Errors

Code

```
command 2> error.txt
```

Pipes

Code

```
command1 | command2
```

Example:

Code

```
ls -l | grep ".txt"
```

9. Searching & Text Processing

grep

Search inside files:

Code

```
grep "word" file.txt
```

find

Search files in directories:

Code

```
find /home -name "*.txt"
```

wc

Word count:

Code

```
wc file.txt
```

10. Package Management

Depends on distribution.

Debian/Ubuntu (apt)

Code

```
sudo apt update
sudo apt install package
sudo apt remove package
```

Red Hat/Fedora (dnf/yum)

Code

```
sudo dnf install package
```

Package managers:

- Download software
- Handle dependencies
- Keep system updated

11. Environment Variables

View:

Code

```
printenv
```

Common variables:

- PATH
- HOME
- USER
- SHELL

Set temporary variable:

Code

```
export VAR=value
```

12. Exit Codes

Every command returns a status:

- 0 → Success
- Non-zero → Error

Check last exit code:

Code

```
echo $?
```

13. Command Efficiency & Philosophy

Linux philosophy:

- Small tools
- Do one thing well
- Combine tools with pipes
- Text-based configuration
- Automation-friendly

Core Skills Gained from Course

After completing the course, you should:

- Confidently navigate the Linux filesystem
- Create, delete, and manipulate files
- Understand permissions deeply
- Use pipes and redirection
- Manage software packages
- Understand processes and environment variables
- Work efficiently in terminal

If you'd like, I can now give you:

- A **condensed exam-revision version**
- A **practice lab exercise sheet**
- Or a **Linux command cheat sheet (1-page printable style)**