


Linux Summer Training 2025



- 
- > We have 5 session:
:Sunday & Wensday 3pm**
 - > Bring Your Lab with you**
 - > Should install linux before session2**
 - > Attend 5 Session include Evaluation**

The background features abstract, 3D-style orange geometric shapes. On the left, there are several overlapping rectangular blocks in different shades of orange, some with thin black outlines. On the right, there are more vertical and horizontal blocks, also in various shades of orange, creating a sense of depth and structure.

**What will we learn
in this session?**

Agenda

01

OS

02

The History of
Linux

03

Why Linux?

04

Intro to Linux

05

Basic commands



OSC

Open Source Community

Open Source Software



Closed Source Software



Free

Open Source





OS

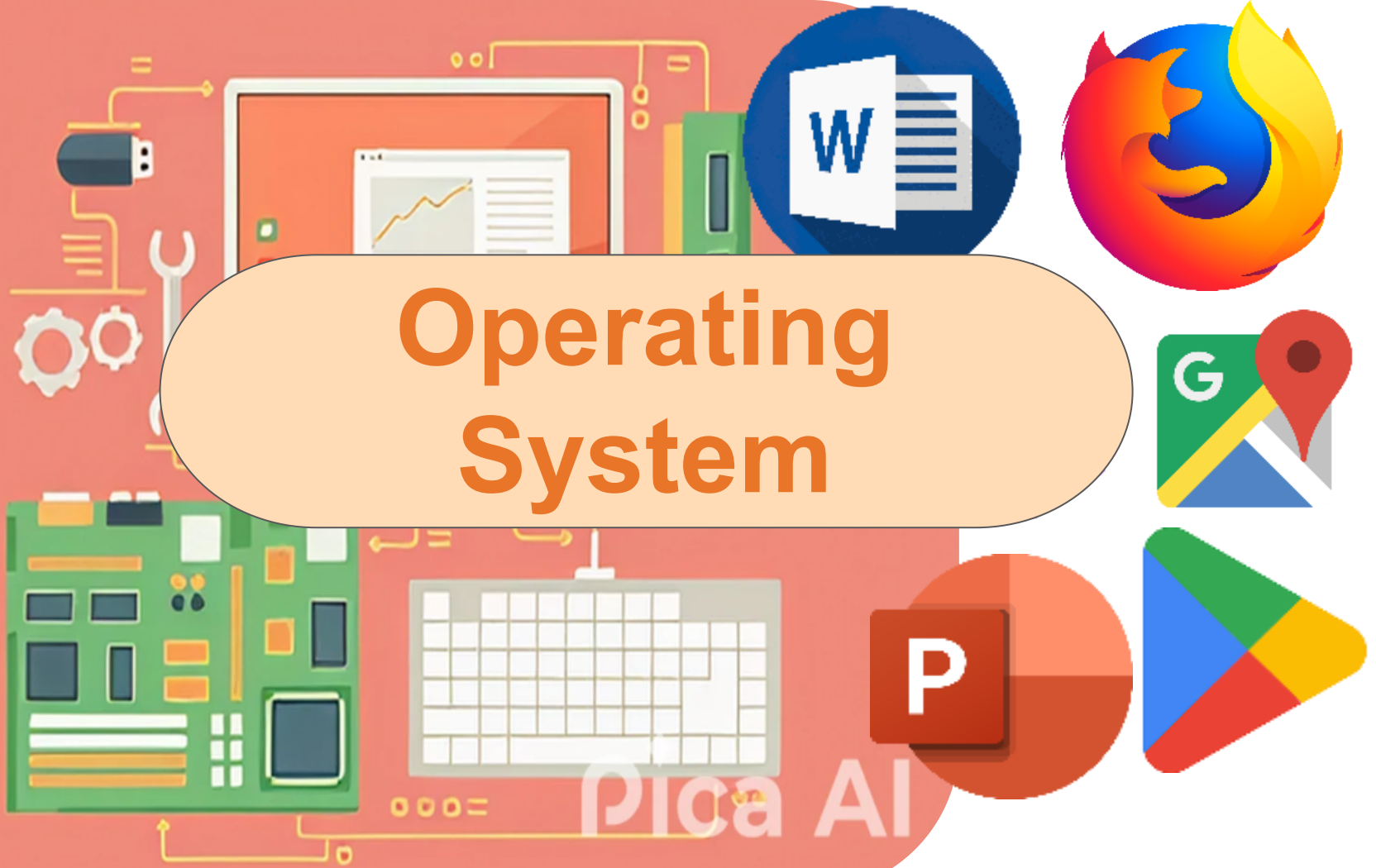
01



What is Linux ?



Operating System



What is an Operating System?

An operating System is the software that manages all the resources of a computer.

It is a fully integrated set of specialized programs that, together, acts as an interface between the software and the computer hardware.

Examples: Linux, macOS and Microsoft Windows.

Operating System Functions

Resource Management

Process management
Memory management
File management
Device management.

Security

Authorization and
protecting device from
malicious threats.

01

02

04

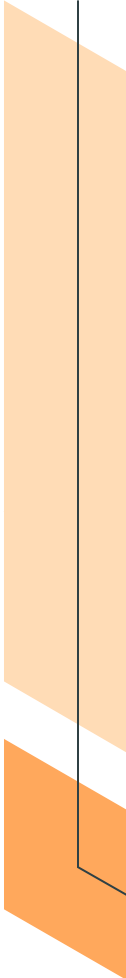
03

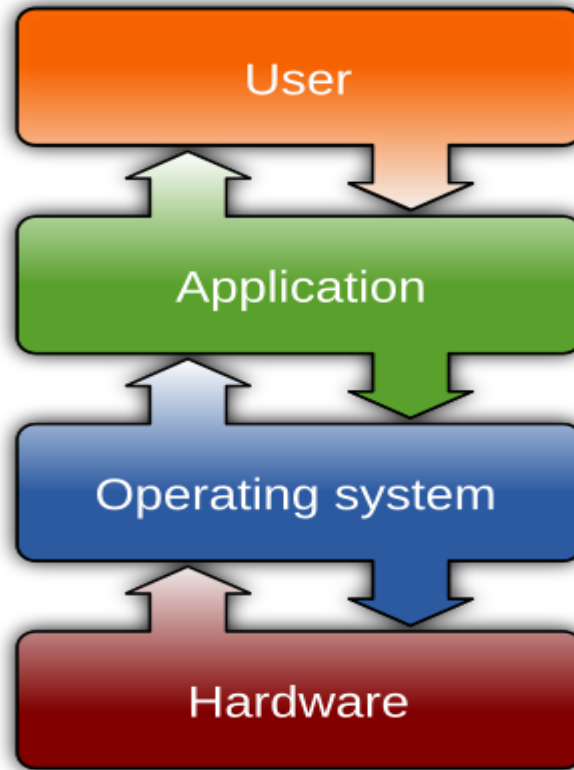
Abstraction

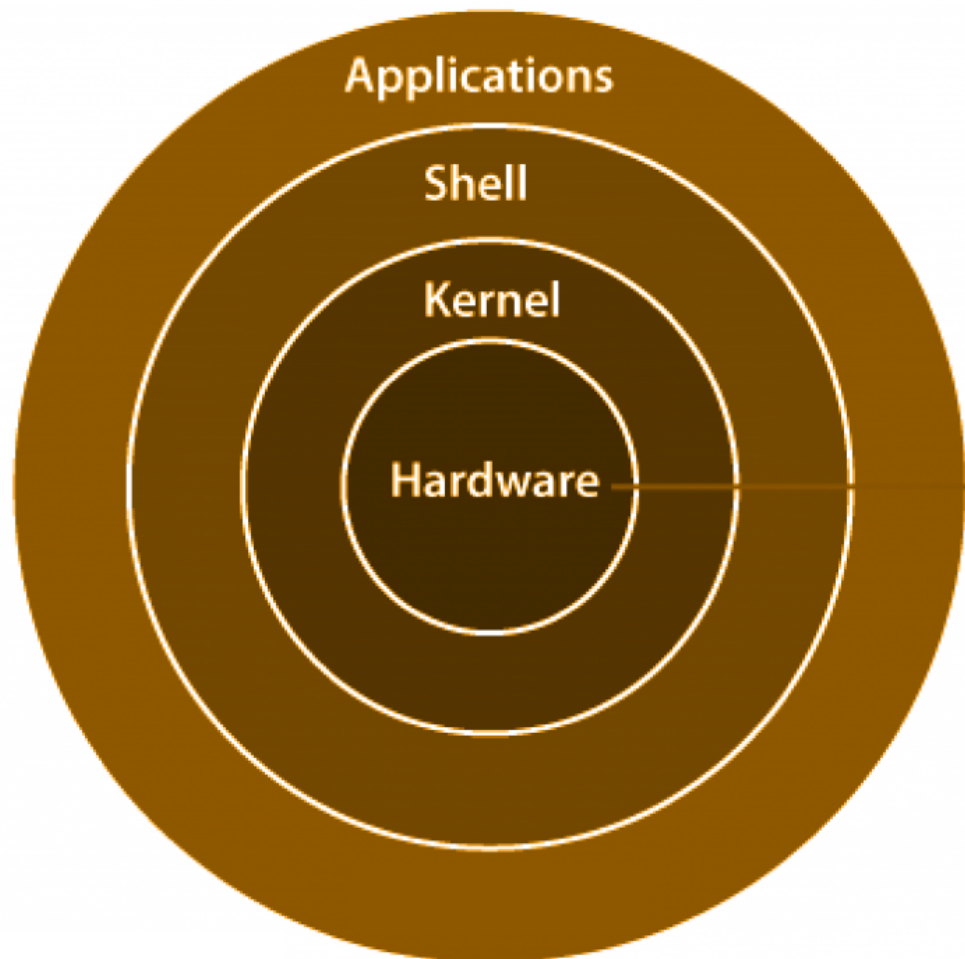
Provide simple
abstractions of physical
resources; you don't
need to worry about low-
level details.

User Interface

Look and feel of the
system.







The Kernel

- The kernel is the **core component** of an operating system.
- It is the **first program** of operating system that is loaded into the main memory to start the working of the system. It remains in the main memory till the system is shut down.
- Kernel acts as a bridge between **application software** and **hardware** of the system. It directly communicates with the hardware and informs it what the application software has requested.

The Shell

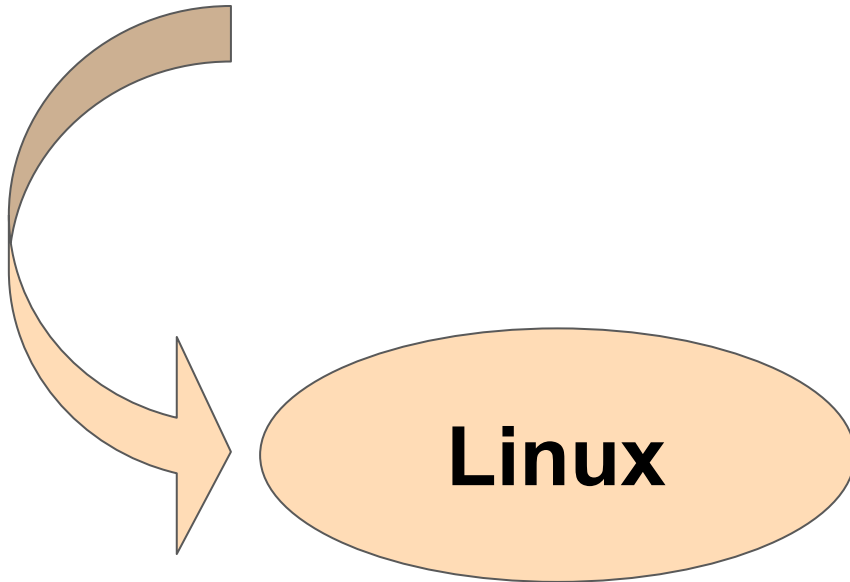
- The shell is a command-line interface (**CLI**) or graphical user interface (**GUI**) that allows users to interact with the operating system.
- It acts as a bridge between the user and the kernel, translating user commands into actions performed by the system.



02

The history of Linux

How did all of this start ?



A Little History

1970s – Unix

```
graph TD; A[1970s – Unix] --> B[Expensive Licensing]; A --> C[Closed source];
```

Expensive Licensing

Closed source

A Little History

Once upon a time, there was an operating system called **Unix**. It was known for being:

- Stable, secure, and reliable.
- The first OS to feature a **command line interface (CLI)**.

However, Unix had one big problem: it was **expensive** and not freely accessible to everyone.

A Little History

Richard Stallman

GNU Project (1983)

✗ Kernel

Linus Torvalds (1991)



The GNU Project

In the 1980s, **Richard Stallman** dreamed of creating a free version of Unix that anyone could use.

He started the **GNU Project**, which focused on developing free software like compilers, editors, and utilities.

But there was a missing piece: **the kernel** (the core of the operating system).

The Birth of Linux

In 1991, **Linus Torvalds**, a student in Finland, created the **Linux kernel**. He shared it with the world as an open-source project, inviting developers to improve it.

With the GNU project tools combined with Linus's kernel, they created a complete operating system called **GNU/Linux**, or simply **Linux**.



Linux Timeline

1970s

Unix is the most popular OS for servers and HPC

Early 1980s

Richard Stallman starts the GNU project to replace Unix

1990

Every part of GNU is complete except the kernel

Mid 1980s

Windows and Macintosh systems appear

1991

Linus Torvalds creates Linux and adds it to GNU, creating GNU/Linux






GNU-Linux

Now I have over 500

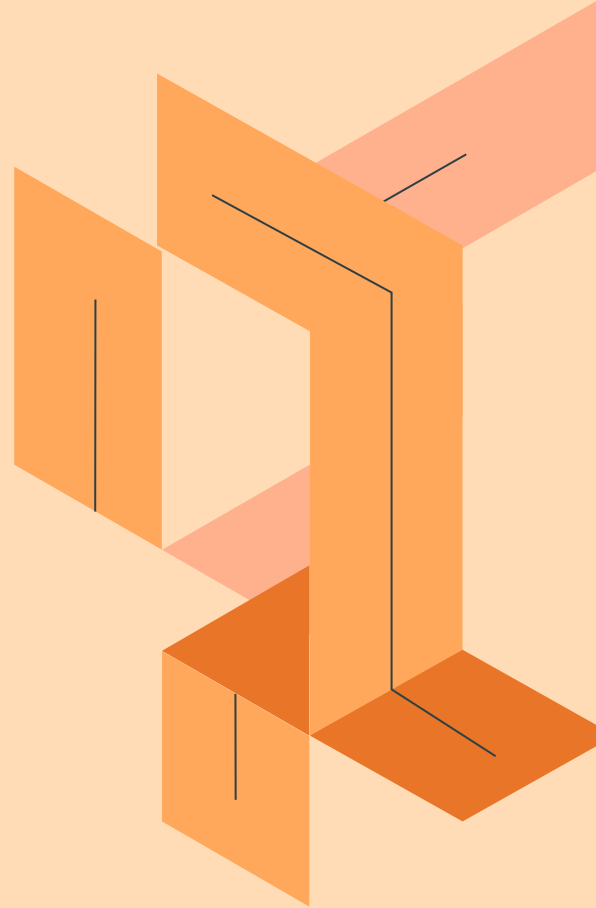




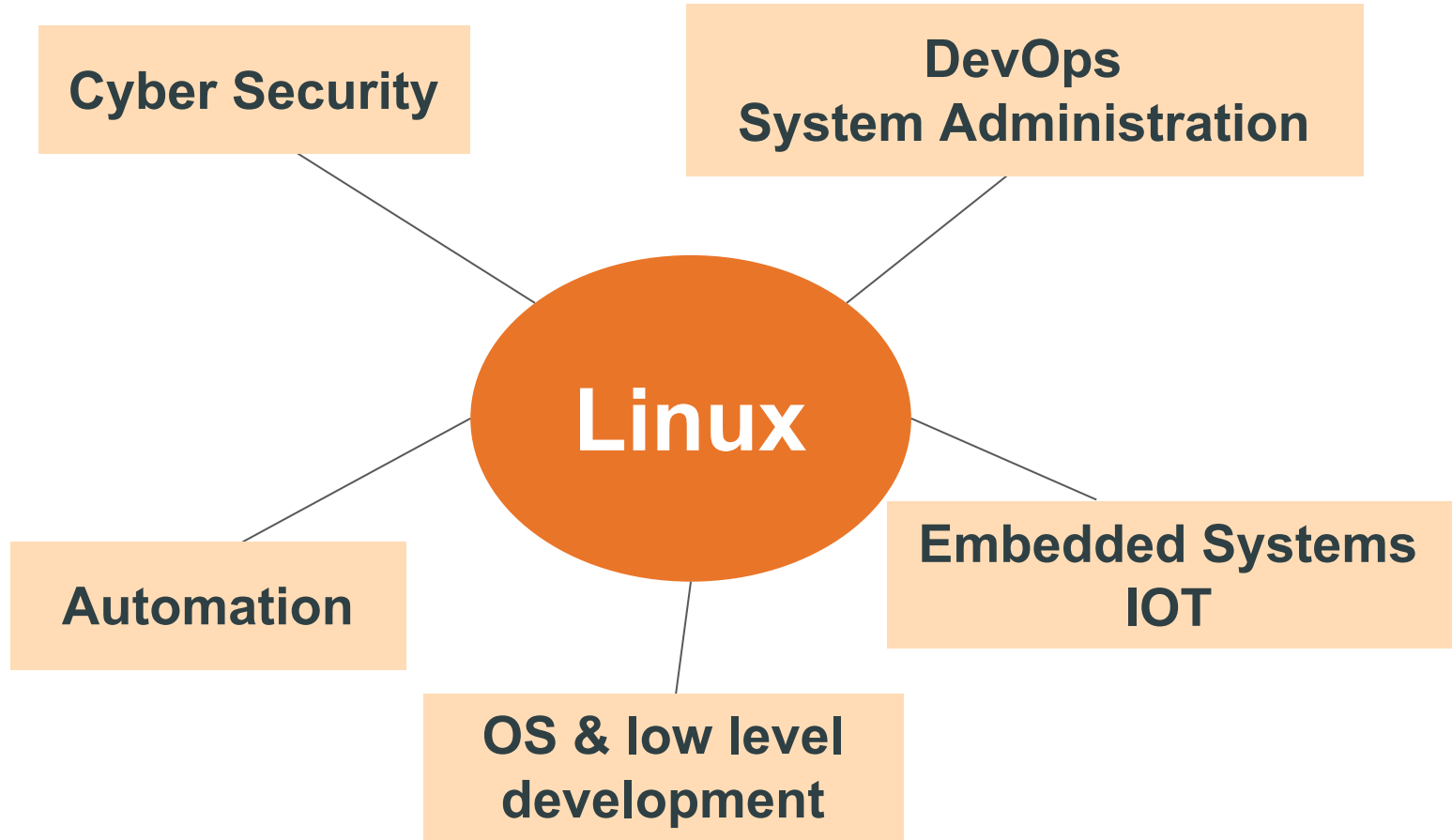
Distribution:

- "Distro" is a pre-packaged set of software that includes the Linux kernel and various other components needed to run an operating system.
 - There are different Linux distros to suit any type of user, from new users to hard-core users.
- 

Why Linux ???



- **Free and Open source**
- **Customizable**
- **Stable**
- **Performance**
- **Learn about your computer**
- **Community and Documentation**
- **Secure**



What are the options?

1. **Primary OS**
2. **Dual (or multi) boot**
3. **Live boot**
4. **Virtual machine**

And more, but we will focus on these four.

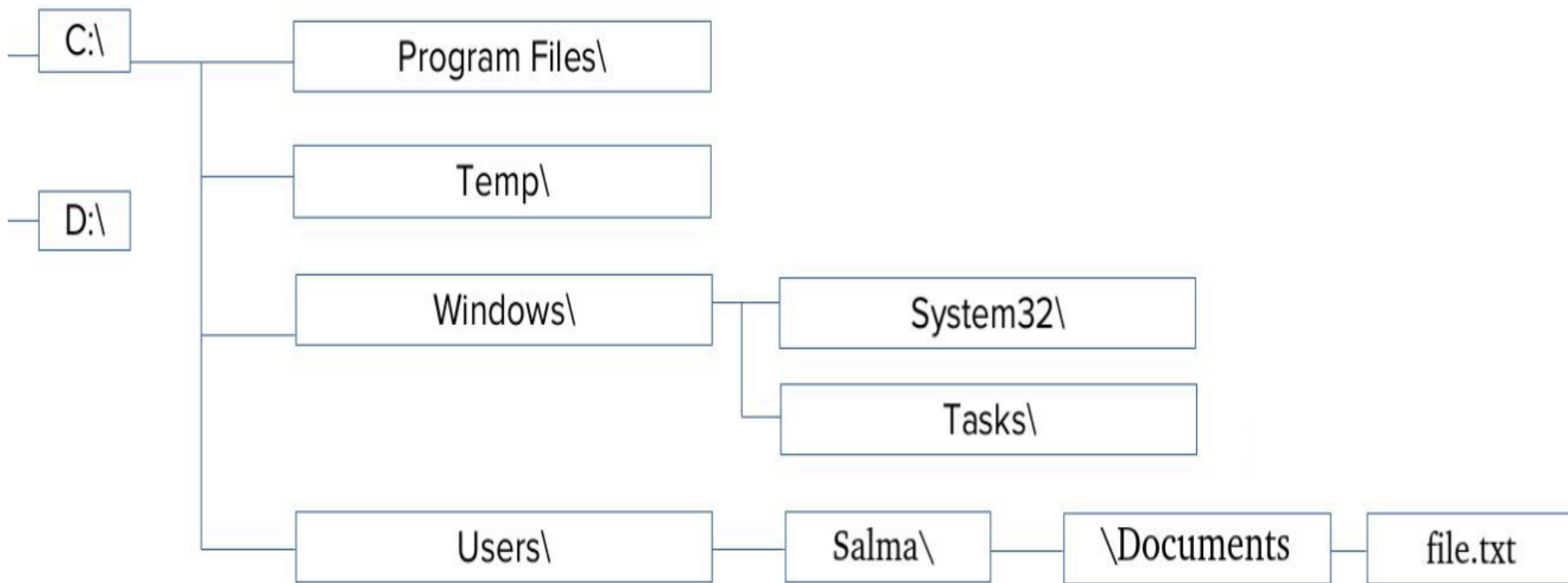
What are the options?

- **Virtual Machine:** Running multiple operating systems by sharing resources(A computer inside a computer).
- **Dual boot:** Refers to the process of installing and running multiple operating systems on a single computer.
- **Live USB:** Burn Linux ISO image to a flash drive and boot directly from it

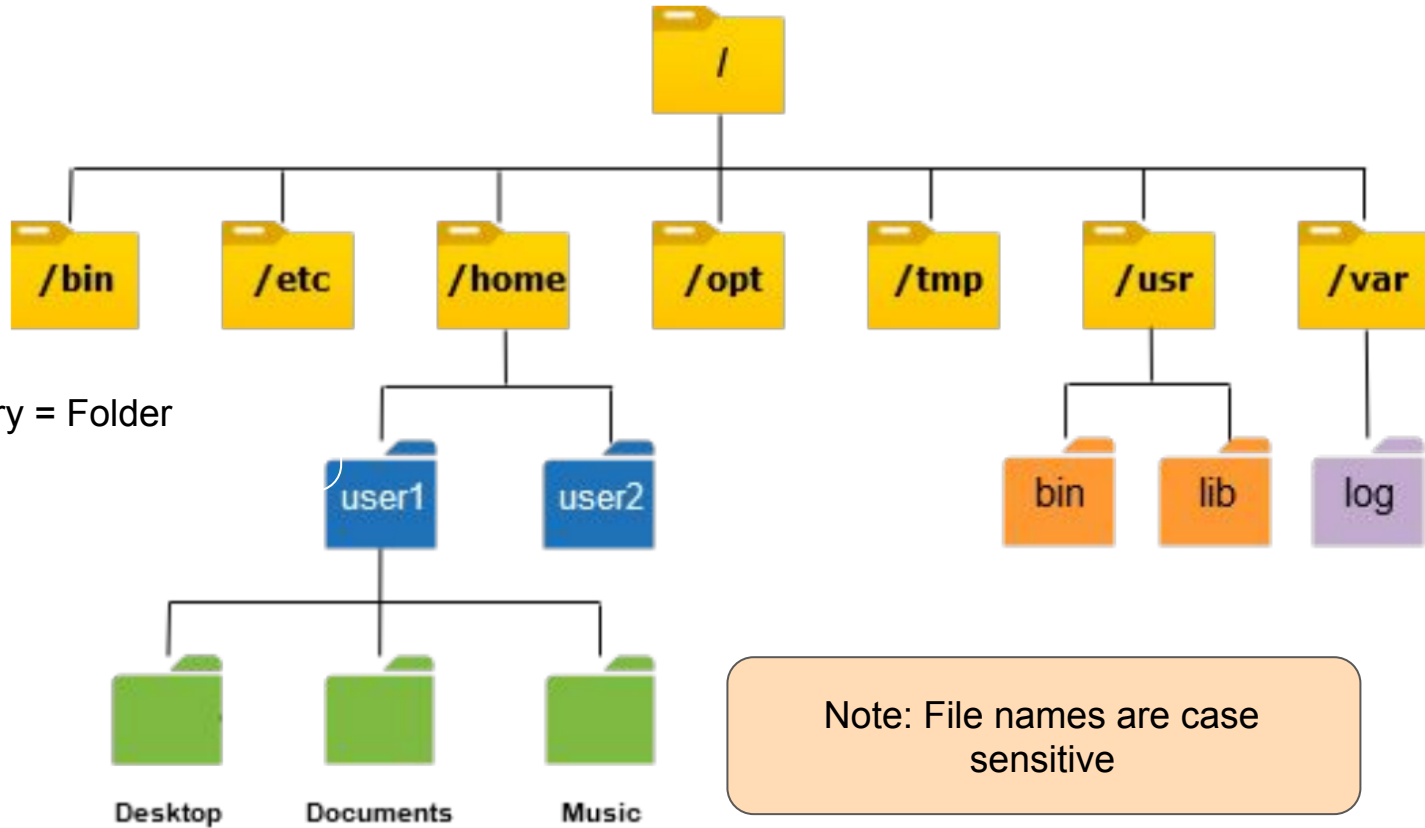


04

Intro to Linux



Directory = Folder



Linux Directory Structure

Relative and Absolute path



Relative and Absolute path

Jack's route to college daily is **Home->Bus Stop->Abbassia->FCIS ASU**.

If he met someone at Abbassia and asked him: "Where are you going?", Jack's response will be "**FCIS ASU**" only, because that's the next step. If someone asked Jack "What's your full route to college?", Jack's response would be **Home->Bus Stop->Abbassia->FCIS ASU**

Note that his route from Abbassia is shorter because it is relative to Abbassia.

The same thing applies in Linux for directories and files.

- **Absolute Path:** The total path leading to the directory.
- **Relative Path:** The path relative to the working directory.

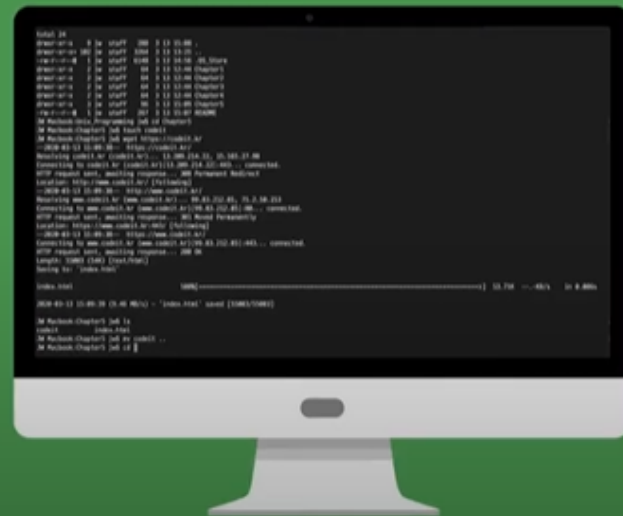
How to interact with the system?

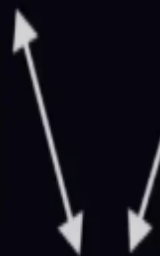
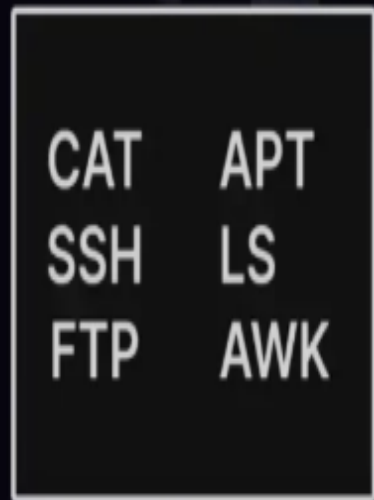


GUI



CLI

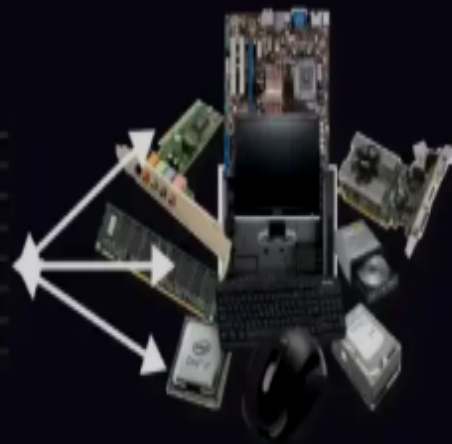
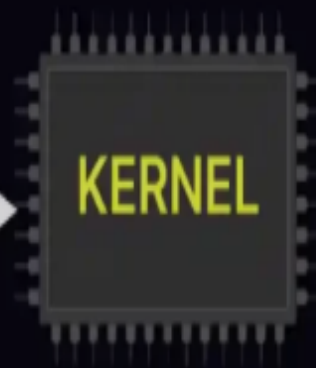
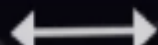




TERMINAL



SHELL



HARDWARE

CLI?

More Powerful and Flexible (ex: copy , rename files)

Uses Less System Resources

Greater Control

Universally Available (Servers)

Username@Hostname:Working_Directory(\$/#)



Username

The username of the current logged-in user.



Hostname

The name of the computer running (Name of the host)



Directory

The working directory, the directory that the terminal is working in right now. (Note: ~ sign is the user home directory)



(\$/#)

\$ states that you are logged as a regular user, while # state that you are logged as System administrator (root).

Break



Revision Time!



Useful commands

pwd

Print Working Directory

Tell where this terminal is working.

cd

Change Directory

Change the working directory to specific argument.

ls

List

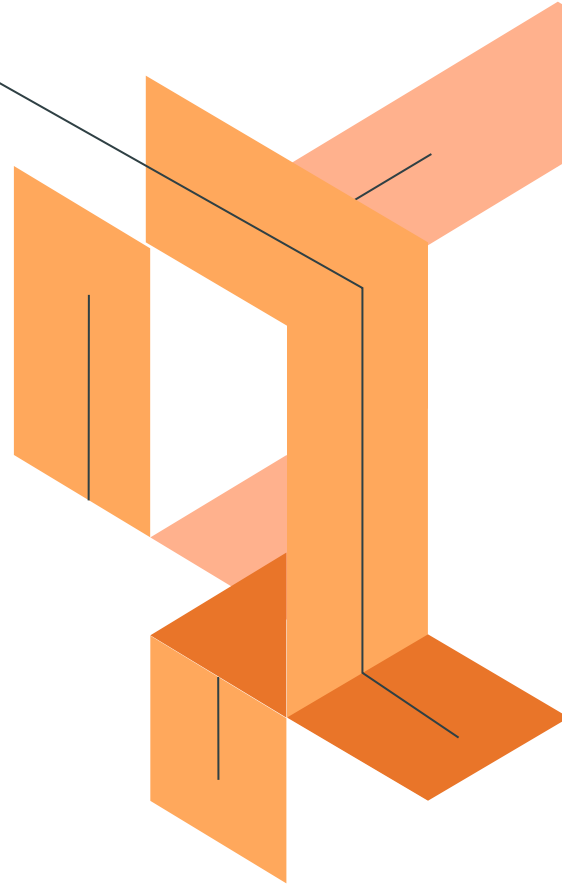
Used to list content of a directory.
-l (long list)

Hidden files

They are files that start with “.”
They don't appear in file content unless you add **-a** to **ls** command.

Examples:

- . it points into this directory.
- .. it points into parent directory.



Command line syntax

Name of the program you run.
i.e. ls, rm, cp, mv ...etc

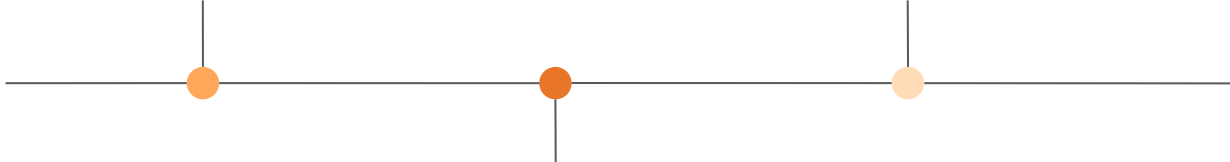
Commands may be followed by one or more arguments, which often indicate a target that the command

Command

Arguments

Option

The command may be followed by one or more options, which adjust the command behavior.
i.e. -h, -f, --all ...etc



File Creation

Command	Functionality
touch	It is used to make new empty file .
mkdir	The Command is used to make new empty directory .
echo	It is used to print on the terminal.

File Movement

Command	Functionality
mv	<ul style="list-style-type: none">• The Command mv is used to move files. mv file new_path• The Command mv also is used to move directories with all content inside. mv directory new_path

Important Note

- Using the "**mv**" command without a name/path of a directory will either:
 - Rename the first file to the name of the second file (if the second file doesn't exist).
 - If the second file already exists, then it will be deleted and the first file will be renamed to the name of the second file.

Copy File

Command	Functionality
cp	<p>File</p> <p>cp is used to copy files. cp file new_path</p> <p>Directory</p> <p>cp with -r option is used to copy directories with all content inside. cp -r directory new_path</p>

Delete File

Command	Functionality
rm	<p>File</p> <p>rm is used to remove files. rm file</p> <p>Directory</p> <ul style="list-style-type: none">• rmdir is used to remove empty directories.• The Command rm with option -r to remove directories recursively. <p>rmdir directory rm -r director</p>

The background features an abstract geometric design composed of various orange and peach-colored shapes, including triangles and polygons, arranged in a way that suggests depth and perspective. These shapes are primarily located on the left and right sides of the frame, framing the central text.

Hands On

Cheat Sheet

Command	Functionality
cd	Navigate between Directories
ls	List content
touch	Create Files
mkdir	Create Directory
cp	Copy File "-r" option with Directory
rm	Remove file "-r" option to remove mutli files

Hands On

- Create a directory named "Summer Training" (with a space) .
- Move inside the newly created directory.
- Create a file named "hello.txt".
- Move outside the directory.
- Copy the "Summer Training" directory along with its contents, and name it "Backup"
- Delete the "Summer Training" directory along with its contents

Hands On Solution

```
mkdir "Summer Training"  
cd "Summer Training"  
touch hello.txt  
cd ..  
cp -r "Summer Training" Backup  
rm -r "Summer Training"
```

Handling multiple files with these commands

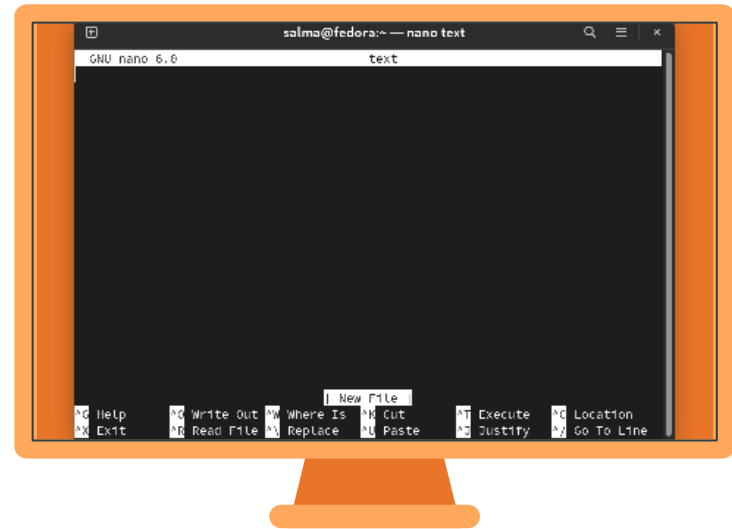
You can use these commands to modify more than one file at the same time.

- **mkdir** dir1 dir2 dir3
- **touch** file1 file2 file3
- **cp -r** dir1 file2 dir3 target
- **mv** file1 file2 file3 target
- **rm -r** file1 dir2 file3
- **rmdir** dir1 dir2 dir3

Nano

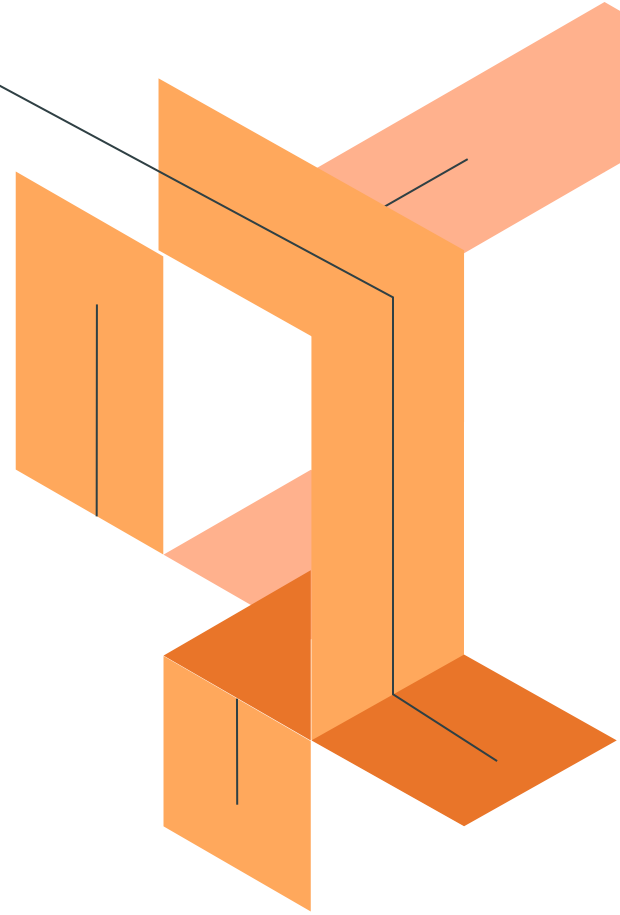
Nano is a Text editor used to edit file content.

- nano file



Cat

- Cat is a command used to display file content.
 - `cat file`
- Can be used to display multiple files.
 - `cat file1 file2`





Thanks!