TERMINAL

A **terminal** is a program which allows you to call other programs using text as input.

Terminal, also known as command-line or console allows us to accomplish and automate tasks on a computer without the use of graphical user interface.

Terminal allows us to send simple text commands to our computer to do things like navigate through a directory or copy a file, and form the basis for many more complex automations and programming skills.

GUI: Graphical User Interface

CLI: Command Line Interface

Shell

Shell is a program that takes commands from the keyboard and gives them to the operating system to perform.

It is named a shell because it is the outermost layer around the operating system kernel.

The **kernel** is a computer program that is the core of a computer's operating system, with complete control over everything in the system. It allows the hardware to talk to the software.

bash

Command Processor: Inside the terminal there is another program to understand the text being typed within the terminal.

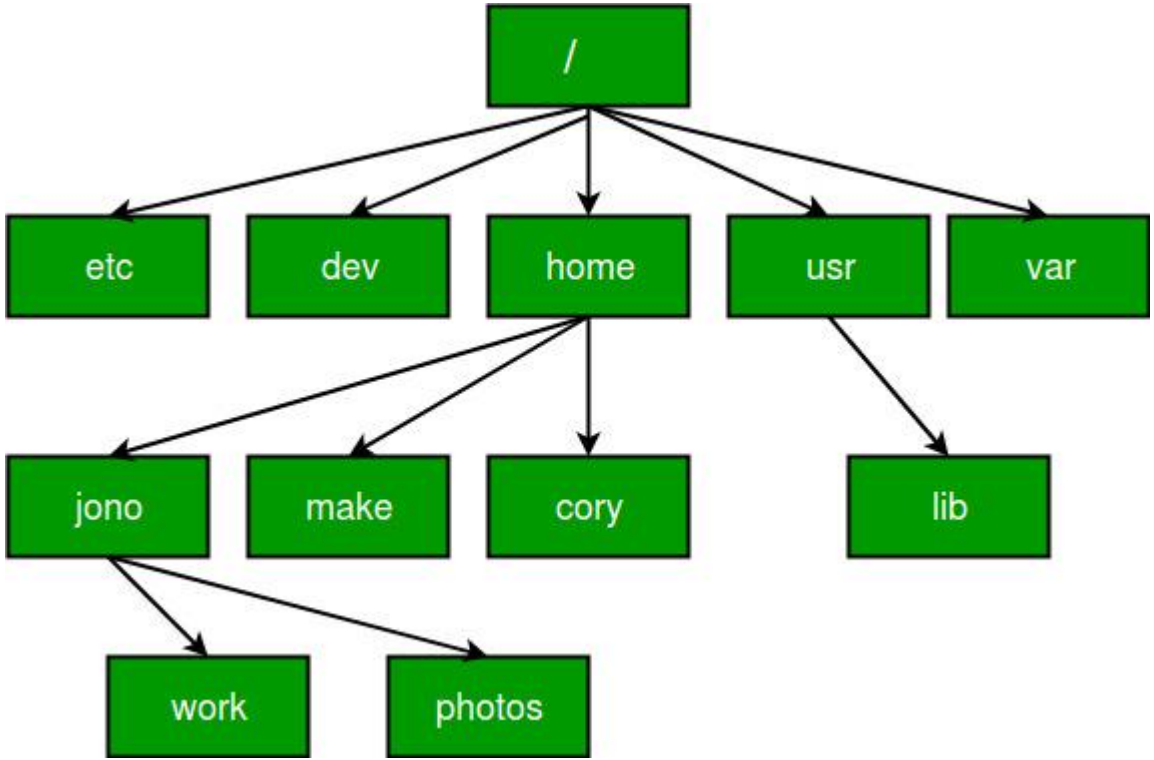
The command processor consumes the text and performs the actions which the user intends to do.

The default command processor in linux is bash(Bourne Again Shell).

There are many shells available such as bash, ksh, zsh, tsch, etc.

Bash(Bourne Again Shell): Almost all linux distributions default to bash shell.

Hierarchical Directory Tree



Hierarchical Directory Tree

- Every file is organized in a hierarchical order.
- The first directory in the filesystem is known as **root**.
- The root directory has many folders and files in which you can store more files and directories.
- The root directly is shown as */*
- The location of these files and directories are referred to as **path**.

Path

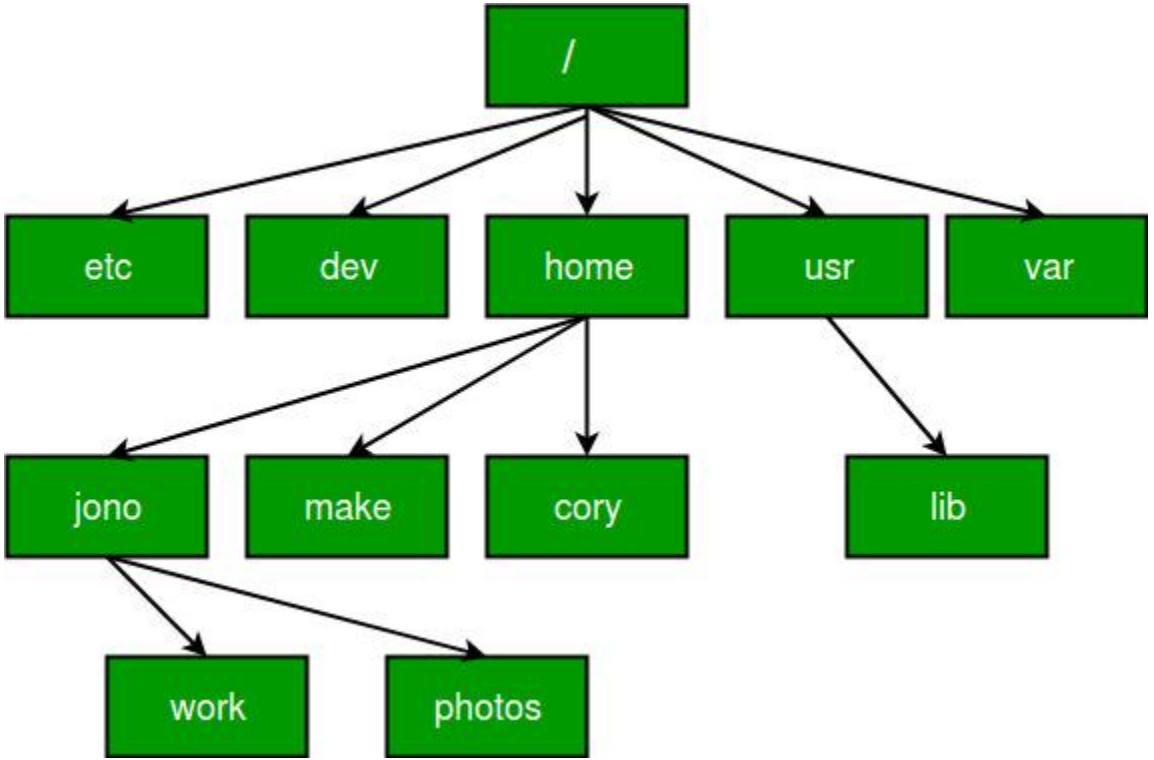
A **path** is a unique location to a file or a folder in a file system of an operating system(OS).

- **Absolute Path**: An absolute path is defined as specifying the location of a file or directory from the root directory(/).

  To write an absolute path-name:

  - Start at the root directory (/) and work down.
  - Write a slash (/) after every directory name (last one is optional)

- **Relative Path:** Relative path is defined as the path related to the present working directly. It starts at your current directory and **never starts with a /** .



Working with Files and Directories

- **pwd:** print the full path of the current working directory. It stands for "Print Working Directory"
- **cd**: change directory. To change to another directory, you can use the **cd** command.
- **ls**: The **ls** command lists the files in the current directory.
- **cat:** to read the content of a file
- **mkdir**: The **mkdir** command makes a new directory.
- **touch**: creates an empty file if that file doesn’t already exist.
- **history:** shows the history of commands you previously entered

Updating and Deleting

- **cp**: The **cp** command copies a file from one location to another.
- **mv**: The **mv** command moves a file from one location to another. It works exactly like the cp command above, but moves the file instead of creating a copy. mv can also be used to rename files.
- **rm** – The **rm** command removes a file.