# Day 1-Linux Core Concepts Part 1

## What is a Shell?

Linux systems can operate through both a **Command-Line Interface (CLI)** and a **Graphical User Interface (GUI)**. The CLI is typically accessed via a **shell**.

---

A **shell** is a text-based interface that allows users to interact with the operating system by entering commands. It acts as a bridge between the user and the system's core (kernel), interpreting input and displaying output.

There are several types of shells, each with its syntax and features:

- **Bourne Shell (sh)**
- **C Shell (csh)**
- **Korn Shell (ksh)**
- **Z Shell (zsh)**
- **Bourne Again Shell (bash)** — the most commonly used shell on Linux systems

You can check which shell you're currently using by running:

`echo $SHELL`

This will return the path to the default shell for your user account, such as `/bin/bash` or `/bin/zsh`.

## Home Directory

When you log into a Linux system, you're placed in your home directory, which is your workspace on the system. It’s a secure area meant only for you; regular users don’t have access to each other’s home directories. This location is represented by the tilde symbol (~).

## Command Types

Linux commands are classified into two types:

### Internal commands

These are built-in to the system and come pre-installed.

Include commands like echo, cd, pwd, set, mkdir, etc.

### External commands

These are available via a binary or script.

Include commands like mv, date, cp, etc

### Some Common Linux Commands

#### Directory Navigation

- `pwd`
  Prints the **present working directory** (i.e., your current location in the file system).
- `ls`
  Lists the contents of the **current directory**.

- `ls <directory>`
  Lists the contents of the **specified directory**.
- `ls -l`
  Displays a **long listing** format, including file permissions, ownership, and timestamps.
- `ls -a`
  Lists **all files**, including **hidden files** (those starting with `.`).
- `ls -lt`
  Lists files sorted by **modification time**, newest first.
- `ls -ltr`
  Lists files sorted by **modification time**, but in **reverse order** (oldest first).

### Directory Creation

- `mkdir <directory>`
  Creates a **new directory**.
- `mkdir -p <parent>/<child>`
  Creates a **parent directory** and its **child** in one command. Useful when the parent directory doesn't exist yet.

### Directory Stack Navigation

- `pushd`
  **Saves** the current directory onto a **stack** and navigates to another directory. Useful for toggling between locations.
- `popd`
  **Returns** to the last directory saved with `pushd`.

### File Viewing (Pagers)

- `more <file>`
  Opens a file in a **pager** that displays one page at a time.
  Common keys:
  - `Enter`: Scroll down one line
  - `/pattern`: Search forward
- `less <file>`
  A more advanced pager with **bidirectional scrolling**.
  Common keys:
  - `↑ / ↓`: Scroll up/down
  - `/pattern`: Search
  - `q`: Quit

### Getting Help

- `whatis <command>`
  Displays a **one-line description** of a command.
- `apropos <keyword>`
  Searches the **man page database** for commands and descriptions related to the keyword.