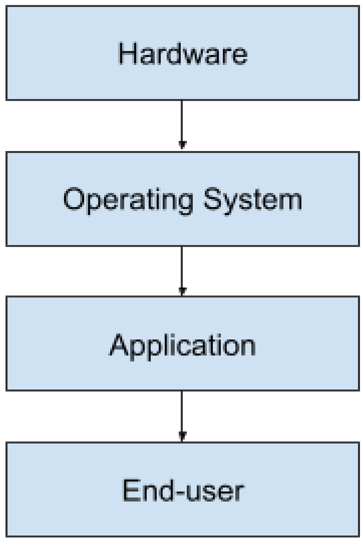
→ Linux

a. Linux is an OS whose kernel is distributed under open-source license
b. functionality like UNIX
c. released by Linus Torvalds on 17 sept 1991
d. Free & open-source
e. source code can be modified & re-distributed

→ Operating System (OS)

a. helps you to interact between hardware and software



→ Operating System

a. OS is an system program that manages computer hardware resources while kernel is the core part of the operating system that interacts directly with the computer hardware
b. In simpler words, OS is like a manager that controls all the resources of the computer whereas kernel works as worker, who actually performs the tasks on computer hardware
c. OS is responsible for managing user level tasks like managing applications. It provides an interface to the user to compute, manage the system resources, while the kernel is responsible for managing the system resources like CPU, memory, i/o devices. Kernel is also responsible for handling system calls, interrupts, etc.
d. OS is a large software program that includes kernel and other system level services which is responsible for managing the system resources, while kernel stays closer to hardware to handle system calls, interrupts, etc.

→ CPU

a. CPU is the brain of the computer, which is responsible for executing and controlling the operation of any system

→ RAM

a. RAM stands for Random Access Memory
b. It stores the data & program instructions temporarily

→ Input Devices

a. These devices are used to enter data or commands into the system
b. Examples
    i. keyboard, mouse, touchscreen, scanner, etc.

→ Output Devices

a. These devices are used to display the output from the system
b. Examples
    i. monitor, printer, speaker, etc.

→ Kernel Modules

a. Kernel Module is also known as Device Driver or Loadable Kernel
b. These are dynamic softwares or loadable components that can be loaded/unloaded into kernel to add support to move hardware
c. Kernel Module is basically written in C lang or Assembly lang.
d. They're loaded into memory when needed
e. We can unload the resources, when we don't need them
f. Examples
    i. Graphics card, network adapter

→ Kernel

a. Kernel is a core component of an operating system that manages system resources like memory i/o, CPU time
b. It provides essential services like process management, memory management, device drivers and system calls

→ System libraries

a. System libraries are pre-written codes, that provides common functionalities to programs and applications running on an operating system
b. system libraries are divided into
    i. <u>Standard Libraries</u>
        1. It is collection of libraries that provide common functionalities for programming languages such as C, C++, Java
    ii. <u>Platform Specific Libraries</u>
        1. These are designed to provide system level resources such as hardware devices, file systems, networks

→ User Utilities

a. It is a set of tools and commands that are used to manage user accounts and their associated permissions
b. Examples
    i. adduser, userdel, chgrp, chown, chmod, su, passwd, chsh, etc

→ User processes

a. It refers to programs/tasks that are initiated and manages by user when any user logs into any system, that typically assigns a user ID, and when the user starts any processes that is typically associated with the user ID
b. Each user process is assigned with a user ID, i.e. PID, which is used to identify & manage the processes
c. User processes typically runs in user mode, which means that they've limited access to system resources and cannot directly access hardware devices or kernel functions

→ System Processes

a. There are system processes that are initiated and managed by the operating system, these are typically run in kernel mode.
b. These kind of processes are responsible for managing system resources, handle system level functions such as, memory management, i/o operations, & process scheduling

→ System Softwares

a. There is a large variety of applications, that falls under this category
b. These are those softwares, which are designed to manage and control the operation of computer system
c. It includes softwares like device drivers, operating system, utility programs, etc.

→ Monolithic kernel

a. It is a type of kernel where all operating system services that operate in kernel space
b. This is an oldest type of kernel, where the entire operating system is composed of single large binary executable file that runs in kernel mode
c. It has direct access to all hardware resource and provides services such as memory management, process scheduling and device driver
d. Advantage
    i. It is very fast as it operates from kernel space
e. Disadvantage
    i. It has million lines of code, so anything gets corrupt, the whole system is affected/stopped
f. Examples
    i. Linux & UNIX

→ Micro kernel

a. This type of kernel provides essential services such as inter-process communication, basic memory management, and other services run as user mode process
b. It is more stable than monolithic kernel, as any service gets affected/corrupt, we don't have to re-install (it'll not affect the whole system)

→ Hybrid Kernel

a. It is a combination of monolithic kernel & micro kernel, combined in such a way where it avoids the non-essential services, like device driver into user mode, and it keeps the critical services in kernel mode
b. *Examples*
    i. WindowsOS, MacOS

→ Difference between Micro Kernel & Monolithic Kernels

Micro kernel	Monolithic kernel
User Services & Kernel Services are kept in separate spaces	Both User Services and Kernel Services are kept in same space (Kernel Space)
They're smaller in size	They're larger than micro kernels
It is easier to add any new functionalities	It is difficult to add new functionalities
Failure of one component does not affect the working of micro kernel	Failure of one component can affect the whole system
Execution speed is slower	Execution speed is faster
e.g. windows, MacOS	e.g. Linux, UNIX

→ Device Management

a. Device management refers to the management of hardware devices by kernel
b. It involves handling device drivers, allocating & freeing resources, such as i/o, etc.
c. The few tasks that are performed in device management are
    i. It loads & initializes the device drivers at system boot time and unload them when they're no longer required
    ii. The kernel detects & configures the hardware devices that are connected to the system
    iii. It also allocates system resources such as memory, i/o ports to each device
    iv. It also ensures that different devices and applications don't conflict with each other when accessing shared resources

    v. It imposes security policies that prevent unauthorized access to sensitive data or devices

→ Memory Management

a. Memory Management in kernel refers to the management of system memory by operating system kernel
b. It involves, allocating & de-allocating memories for different applications
c. Following are the tasks that are performed under memory management:
    i. The kernel manages the allocation & de-allocation of memory
    ii. It keeps track of available memory
    iii. Allocate the memory to process when requested
    iv. The kernel provides memory protection to ensure that processes doesn't interfere with each other
    v. It provides the mechanism of sharing memory between processes
    vi. The kernel uses Paging & Virtual Memory to manage the system memory efficiently

→ Process Management

a. Process Management in kernel refers to management of process, which includes, creation, scheduling & termination of any processes
b. Following are the tasks that are performed under process management:
    i. Kernel creates process when a program is executed by user or other process
    ii. Each process have unique process ID, and are allocated resources such as memory & file descriptor
    iii. Kernel schedules processes for execution on the CPU using scheduling algorithm
    iv. Kernel provides synchronization mechanism such that multiple processes can access shared resources without interfering with each other
    v. Kernel terminates the process when they've finished execution or when they're terminated by the user

→ Shell

a. Shell is a command line interface that allows user to interact with the operating system
b. It is a program that interprets user input and execute the command
c. Following are the tasks that are performed by the shell:
    i. Ee can execute the commands
    ii. The shell provides i/o redirection which allows the user to redirect the input & output of the command to-and-fro from the file
    iii. The shell amanages the system and environment variables, these are those variables that stores the information about the system environment
    iv. User can modify the system variables using shell

        v. The shell allows the user to write a script which is a collection of commands that can be executed as a single unit

    d. Shells are of two types:
        i. <u>Graphical Shell</u>
            1. These shells specify the manipulation of program using graphical interface that provides operations like moving, closing, re-sizing, switching between different applications
        ii. <u>Command Line Shell</u>
            1. It is a program that provides a command line interface for interacting with operating systems. It allows user to enter any command on command line & execute them

→ bash

a. Stands for bourne shell
b. extension for bash is '.sh'
c. This is usually installed in /bin/sh
d. Default prompt for root user is '#'
e. Default prompt for normal user is '$'

→ Linux File System

a. In Linux , files are ordered in tree structure, where root is considered as start of file system, and root is denoted by `'/'`
b. There are different types of files:
    i. Regular Files
        1. Contains files like images, programs, text, configuration files
        2. Represented by `'-'`
    ii. Directories
        1. These are special types of files, that contains files & directories
        2. Represented by `'d'`
    iii. Special Files
        1. Symbolic Links
            a. Represented by `'l'`
        2. Block Device file / Devices
            a. These are special files that represent physical & virtual devices in a system such as printer, hard drive, CD ROM, etc.
            b. Represented by `'b'`
        3. Character device file
            a. Represented by `'c'`
        4. Socket file
            a. Represented by `'s'`
        5. Named pipe file or pipe file
            a. Represented by `'p'`

→ Types of users in Linux

There are different types of users in linux:

a. Regular Users
    i. It is created when we install linux.
    ii. All files & directories are stored in the home directory of the user.
    iii. These kind of users don't have access to directory / files of any other user
b. Administrative / Root Users
    i. Root user is the super user, who has access to all the restricted files and have admin privileges
c. Service Users
    i. Linux is widely used as a server and services like email and other applications have their own service account.

Note: WinOS has Admin, Child and Guest account

→ File System in Linux

a. In linux, directories are created in /home
b. If you create any user in linux,their files & directories will be saved in `/home/<user>` (`/home/sdevsinx`)
c. In windows, all the program files are usually stored in C: drive, while in linux, the system & program files are stored in different directories like boot files are stored in /boot dir, all the program files are stored in /bin , etc.

→ Difference in Windows & Linux

Windows	Linux
Windows uses different drives with letters like C: , D:	Linux uses tree like structure for file system
In Windows , all the peripheral devices are considered as devices	In Linux , all the peripherals are considered as files
In Windows , there are five different kind of users like admin, child, guest, etc	In Linux , we have three different types of users, like regular, root & service

→ Components of command line

a. `'$'` Root User
b. `'#'` Regular User
c. `'~'` Home Directory

→ Path

a. Path is the location of any file or directory in the file system
b. There are two different types of paths:
    i. Absolute Path
        1. Defines the location of any file or directory from the root directory
        2. Example
            a. `/home/user/filename`
    ii. Relative Path
        1. Defines the path related to pwd, it starts at your current directory
        2. Example
            a. `'./'`(current directory)

→ Linux File Structure

a. In Linux , files are stored in tree structure
b. On the top, we have root directory, and under root directory, we have other directories and their sub-directories
    i. `/etc`
        1. It contains all the configuration files used by the system services
        2. This contains startup & shutdown, that are used to start & shut down individual programs
    ii. `/boot`
        1. It contains all the files needed to start the booting process
    iii. `/usr`
        1. Contains all the shared libraries, installed softwares & read-only data
    iv. `/hom`
        1. This stores all the program files by the user
        2. It contains all the user stored data, personal configuration, personal data such as, documents, music, videos, etc.
    v. `/bin`
        1. This contains all the user commands in binary format, like ls, cp, pwd and others
    vi. `/dev`
        1. This contains all the device files used to access the hardware
    vii. `/tmp`
        1. This stores the temporary files