

Operating Structure & Architecture

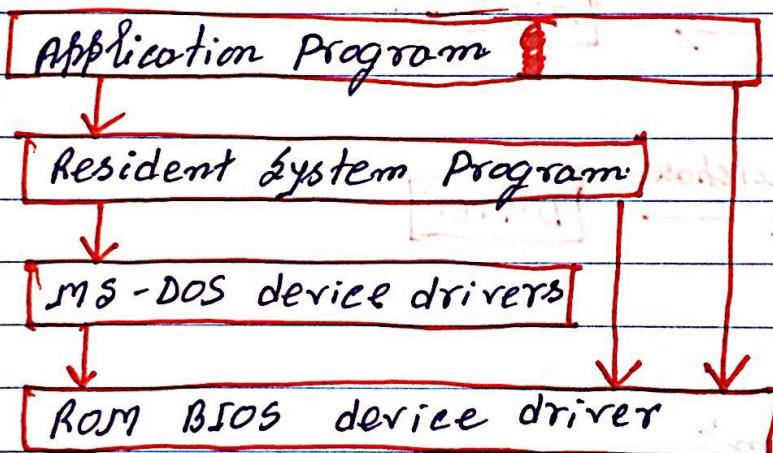
6-2

Structure:-

- Managing Complexity.
- Influenciable key features are robustness, efficiency.

Simple Structure:-

- NO CPU execution mode (User & Kernel).
- Errors in application cause the whole system crash.
- Eg. MS-DOS, UNIX.



Monolithic System:-

- No special accommodation for the special nature of OS.
- Functionality of the OS is simple function calls within Kernel
- Device drivers are loaded into Kernel.
- Eg. CP/M, DOS.

MicroKernel Systems:-

- Its target is robustness.
- Privileges of the individual parts of the OS are more restricted.
- Communication b/w parts relies on special communication mechanism.
- eg: MACH, OSX.
- provides more security & reliability.

Layered Systems:-

- Breaks up the OS into different layers.
- Allow developers to increase modularity.
- As long the external interface of the routine don't change, developers have more freedom to change the inner routine.
- Bottom layer is the h/w & highest layer is the user interface.
- privileged layer contain handling coding with interrupt handling & context switching.

Eg. MULTICS, Windows NT.

Advantages:-

- Simplicity of construction & debugging.

Difficulty:-

- Defining various layers.

Disadvantage:-

- OS tends to be less efficient than other implementation.

Operator

User Program

I/O Management

Device Driver

Memory Management

Process Allocation multiprogramming

Hardware

Virtualization:-

- Is a technique which is pre-installed or permanently loaded on the local device.
- No hard disk needed.

Type of virtualization:-

1) Private Virtual Disk:- Used by one client only.

2) Shared/Common Virtual Disk:- Used by multiple clients at the same time.

Advantages:-

- Imposes little or no overhead.
- Capable of live migration.

virtualBox/vmware/HyperV

Guest
OS

Guest
OS

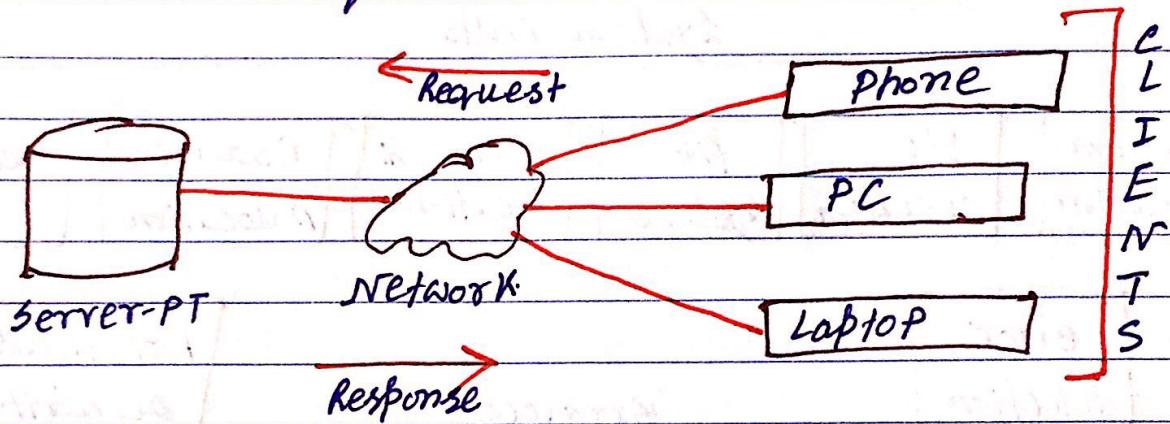
Guest
OS

Host Operating System

Host hardware

Client-Server Model:-

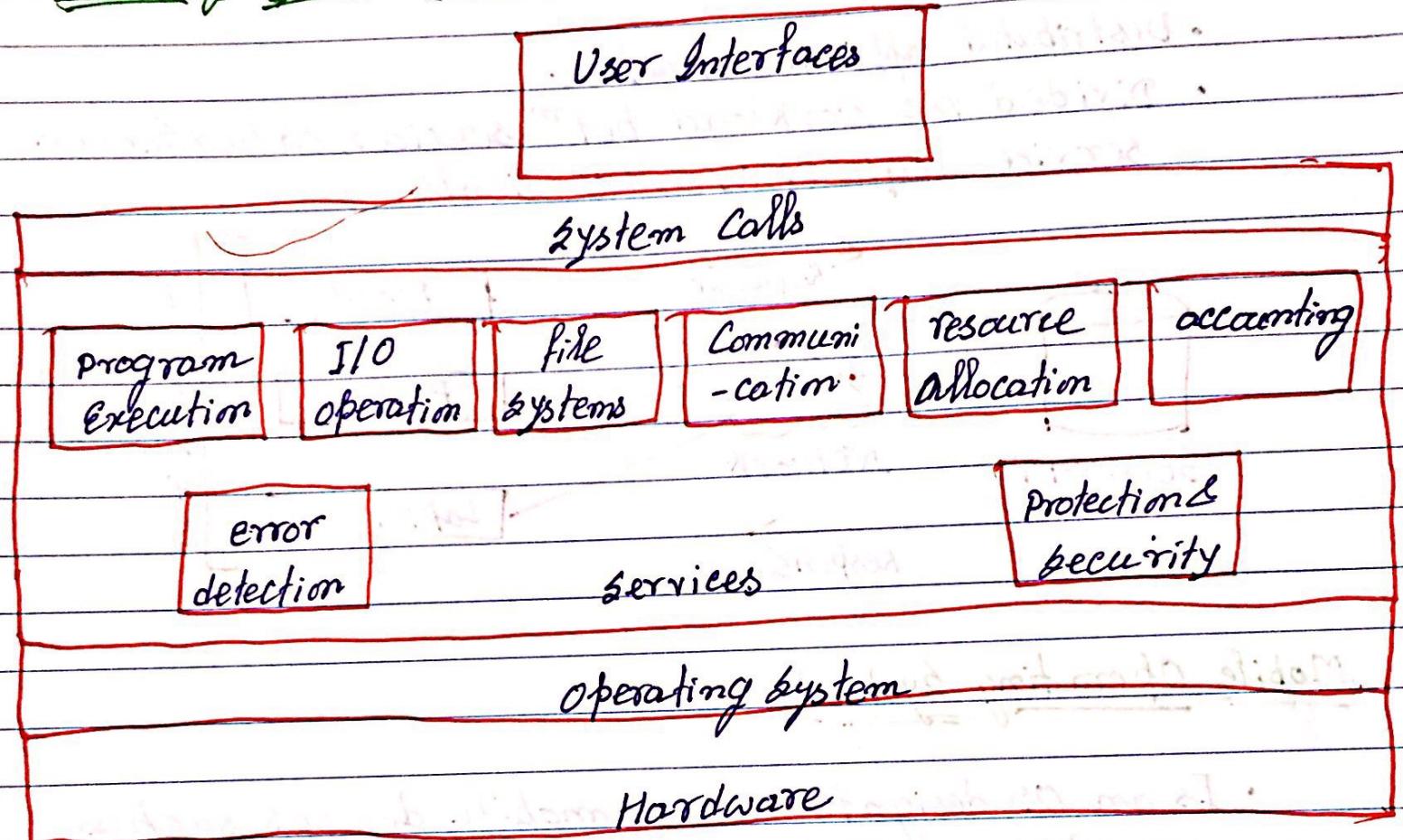
- Distributed application structure.
- Divided the workload betⁿ: services, called Server.
- service requesters called clients.



Mobile Operating System:-

- Is an OS designed to run on mobile devices such as smartphone, PDA, tablet etc.
- eg. Android, IOS.

Operating System Services:-



User Interfaces:-

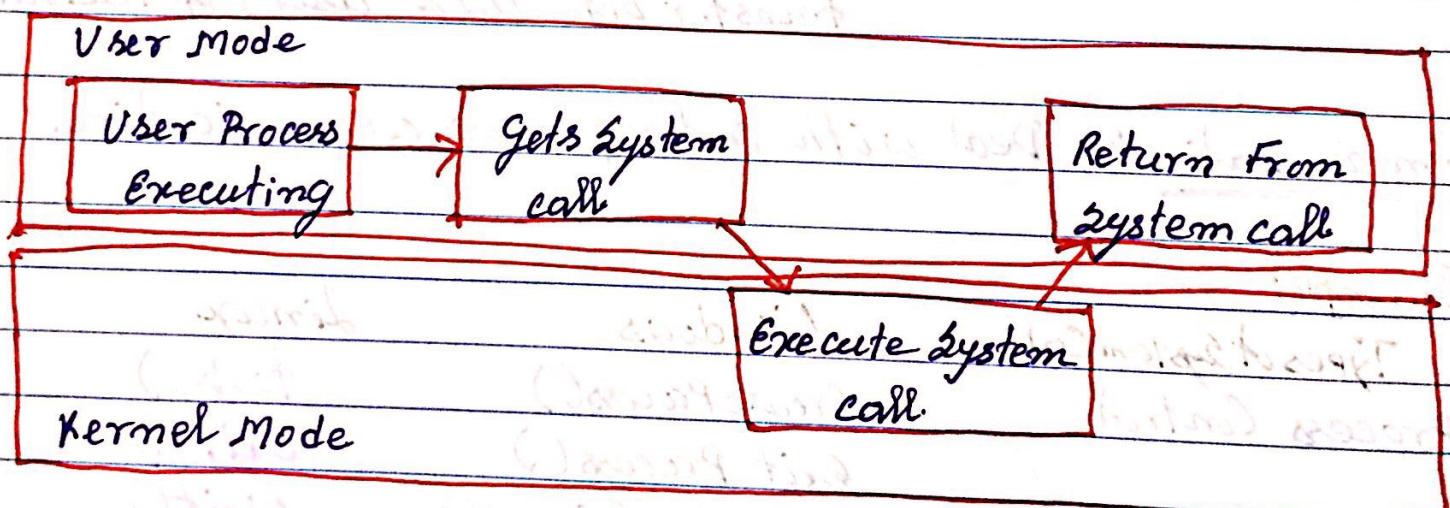
- User sends command to the system
- Different Command line user interfaces
 - Command line Interfaces
 - GUI interfaces
 - Batch Command System.

System Calls:-

- provide a path for user to call the services of OS.
- like a protected procedure call
- similar to procedure call

When system call requires:-

- creation or deletion of a file in a file system.
- read & write from files.
- create & manage of new processes.
- network connection, sending & receiving packets.
- access of h/w devices.



Types of System Calls:-

Process Control:- Deal with process, like process creation, process termination.

File Management:- Deal with File manipulation. Like create, read, write of files.

Device Management:- Deal with device manipulation like reading from device, buffers, write to devices.

Information Maintenance Handle information & information transfer betⁿ: OS & user application.

Communication:- Deal with interprocess communication.

Example:-

Types of System Call

Process Control

Windows

CreateProcess()

Exit Process()

WaitForSingleObject()

Linux

fork()

exit()

wait()

File Management

CreateFile()

Open()

ReadFile()

read()

WriteFile()

write()

CloseHandle()

close()

Type of System Calls

Windows

Linux

Device Management

SetConsoleMode()

ioctl()

ReadConsole()

read()

WriteConsole()

write()

Information Maintenance

GetCurrentProcessID()

getpid()

SetTimer()

alarm()

Sleep

sleep()

Communication

CreatePipe()

pipe()

CreateFileMapping()

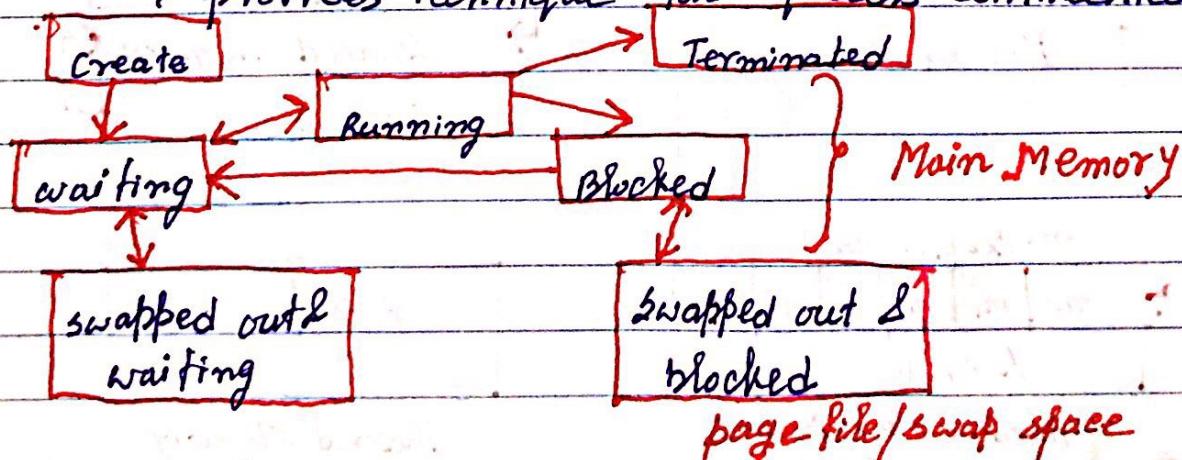
shmat()

MapViewOfFile()

mmap()

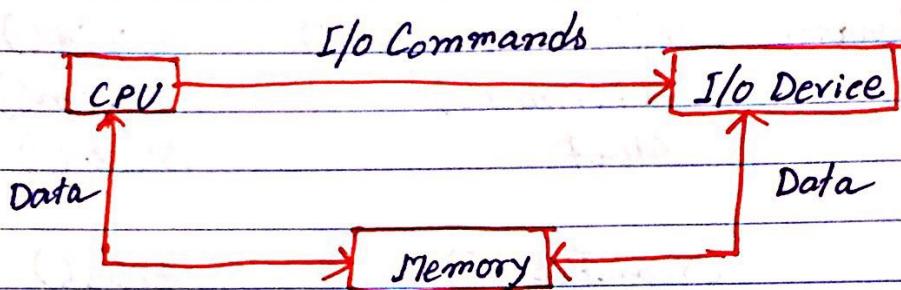
Program Execution:-

- load various activities into the memory & run it.
- program execution end either normally or forcefully.
- steps :-
 - Load program into memory.
 - Executes the program.
 - Handles the program execution
 - provides technique for process synchronization
 - provides technique for process communication



I/O Operation:-

- Responsible for transferring data to & from I/O devices.
- e.g. Keyboard, printers etc.
- I/O file or I/O devices involved.
- read & write are the I/O operations.

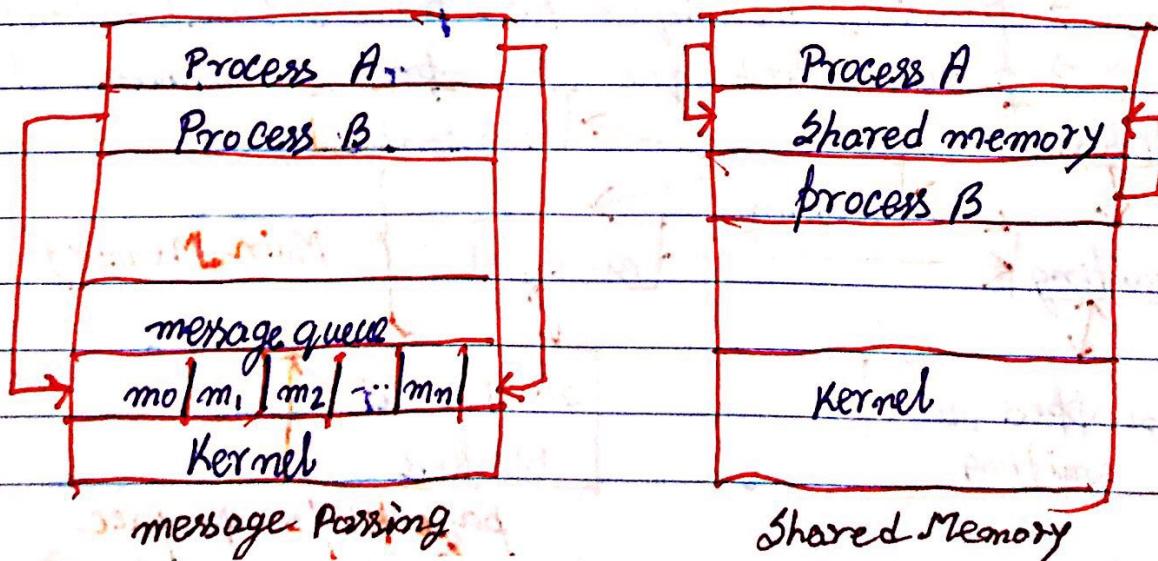


File System:-

- maintaining directory & subdirectory structures.
- mapping files name to specific blocks of storage
- performing operation on file.
- read & write a file.
- create / delete a file/directories

Communication:-

- Communication betⁿ: processes running on the same processor, separate processor, or separate machine.



Error Detection:-

- H/w & s/w error detected & handled appropriately.
- Error can occur anywhere anytime, like CPU, I/O devices.
- OS always checks for the possible errors.
-

Resource Allocation:-

- manages all kinds of resources using schedulers.
- CPU scheduling algorithms is used for better utilization of CPU.

Accounting:-

- Keeping track of resource usage & activity.

Protection & Security:-

- Secure the system from harmful attack.
 - virus
 - Worm
 - Denial of service

Methods:-

Authentication: Username / password

User Key

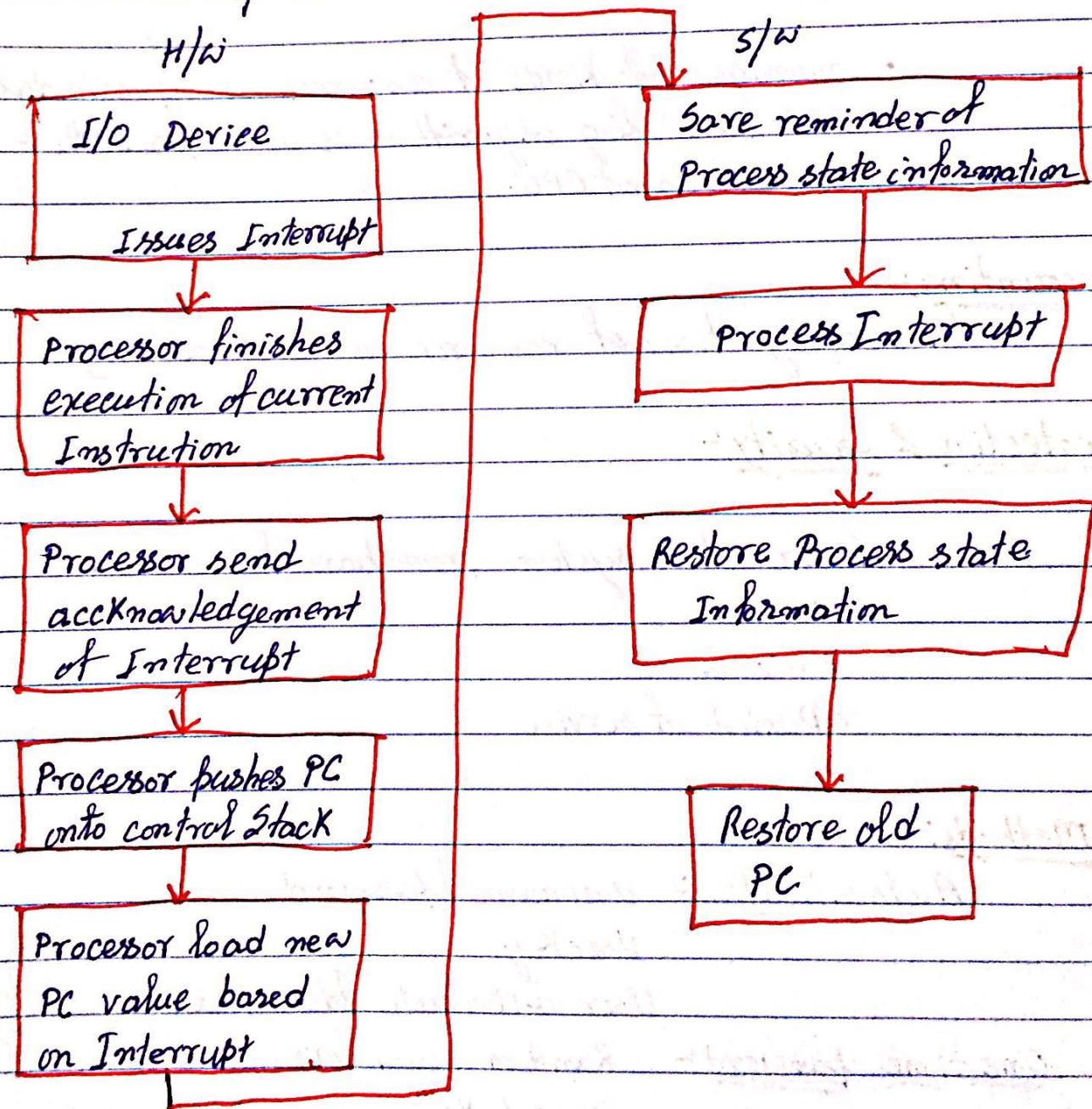
User attribute Identification

One time password: Random numbers

secret Key

Interrupt:-

- Input device issues the interrupts.
- The processor execute the instruction before responding to the interrupt.
- H/w device send signal to OS when interrupt occur.
- OS preempt the running process to handle the interrupt.



PC → Program Counter

Booting sequence:-

- In which order computer searches the nonvolatile data storage where program code ~~loaded~~ stored previously to upload the OS.

Boot Process:-

- PC turned on & the BIOS initializes the h/w
- BIOS calls code, stored in MBR at the start of disk0.
- MBR loads code from the bootsector of the active partition.
- Bootsector loads & run the bootloader from filesystem

Stages:-

1. ROM Code
2. SPL
3. U-BOOT
4. Linux Kernel