Operating System Concepts

SunBeam Institute of Information & Technology, Hinjwadi, Pune & Karad.

# Scope of the Subject OS Concepts from CCAT point of view:

Section-B : Total 9 Questions

- Almost 95% questions are concept/theory oriented and GK about the subject.

Operating System Concepts

*** Introduction:**

- Why there is need of an OS?
- What is an OS?
- Booting process in brief
- Functions of an OS

*** UNIX System Architecture Design**

- Major subsystem of an UNIX system: File subsystem & Process Control subsystem.
- System Calls & its catagories
- Dual Mode Operation

*** Process Management**

- What is Process & PCB?
- States of the process
- CPU scheduling & CPU scheduling algorithms
- Inter Process Communication: Shared Memory Model & Message Passing Model

Operating System Concepts

*** Process Management**

- Process Synchronization/Co-ordination
- Deadlocks & deadlock handling methods

*** Memory Management**

- Swapping
- Memory Allocation Methods
- Internal Fragmentation & External Fragmentation
- Segmentation
- Paging
- Virtual Memory Management

*** File Management**

- What is file?
- What is filesystem & filesystem structure?
- Disk space allocation methods
- Disk scheduling algorithms

Q. Why there is a need of an OS?

- Computer is a **machine/hardware** does different tasks efficiently & accurately.

- Computer hardware mainly contains:

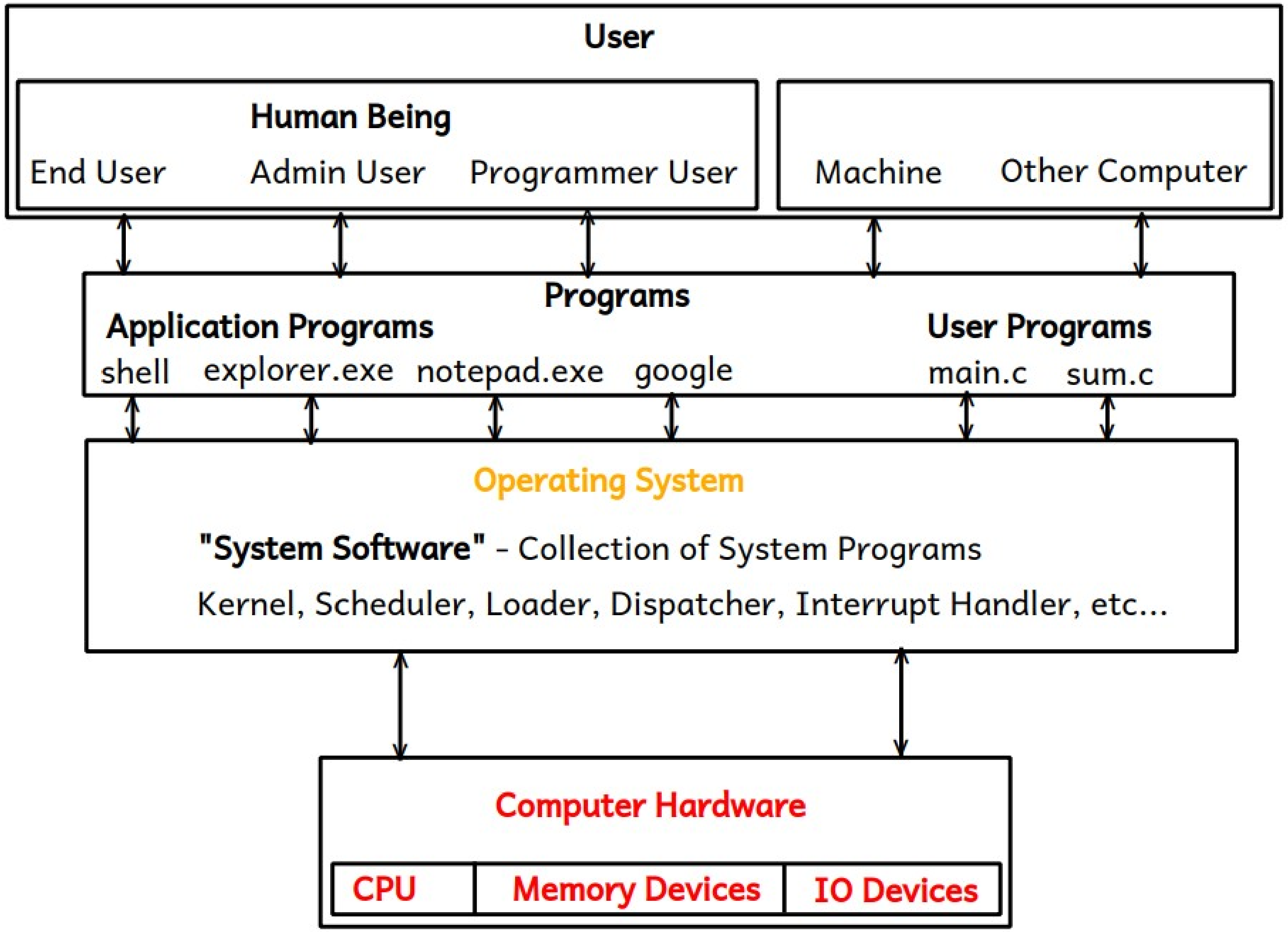
1. CPU/Processor

2. Memory Devices

3. I/O Devices

- As any user cannot communicates/interacts directly with any of the computer hardware components to do different tasks, hence there is need of some interface between user and hardware and to provide this interface is the job of an OS.

Operating System Concepts



Q. What is a Software?

- Software is a collection of programs.

Q. What is a Program?

- Program is a **finite set of instructions written in any programming language** (either low level or high level programming language) **given to the machine to do specific task.**

- 3 types of programs are there:

1. "user programs": programs defined by the programmer user/developers

e.g. main.c, hello.java, addition.cpp etc....

2. "application programs": programs which comes with an OS/can be installed later

e.g. MS Office, Notepad, Compiler, IDE's, Google Chrome, Mozilla Firefox, Calculator, Games etc....

3. "System Programs": programs which are inbuilt in an OS/part of an OS.

e.g. Kernel, Loader, Scheduler, Memory Manager etc...

Q. What is an IDE (Integrated Software Developement) ?

- It is an **application software** i.e. collection of tools/application programs like **source code editor, preprocessor, compiler, linker, debugger** etc... required for **faster software developement.**

e.g. VS code editor, MS Visual Studio, Netbeans, Android Studio, Turbo C etc....

Source Code – Program written in any programming langauge.

1. "Editor": it is an application program used to write a source code.

e.g. notepad, vi editor, gedit etc...

2. "Preprocessor": it is an application program gets executes before compilation and does two jobs - **it executes all preprocessor directives** and **removes all comments from the source code.**

e.g. cpp

3. "Compiler": it is an application program which **converts high level programming language code into low level programming language code i.e. human understandable language code into the machine understandable language code.**

e.g. gcc, tc, visual c etc...

Operating System Concepts

4. "Assembler": it is an application program which converts assembly language code into machine language code/object code.

e.g. masm, tasm etc...

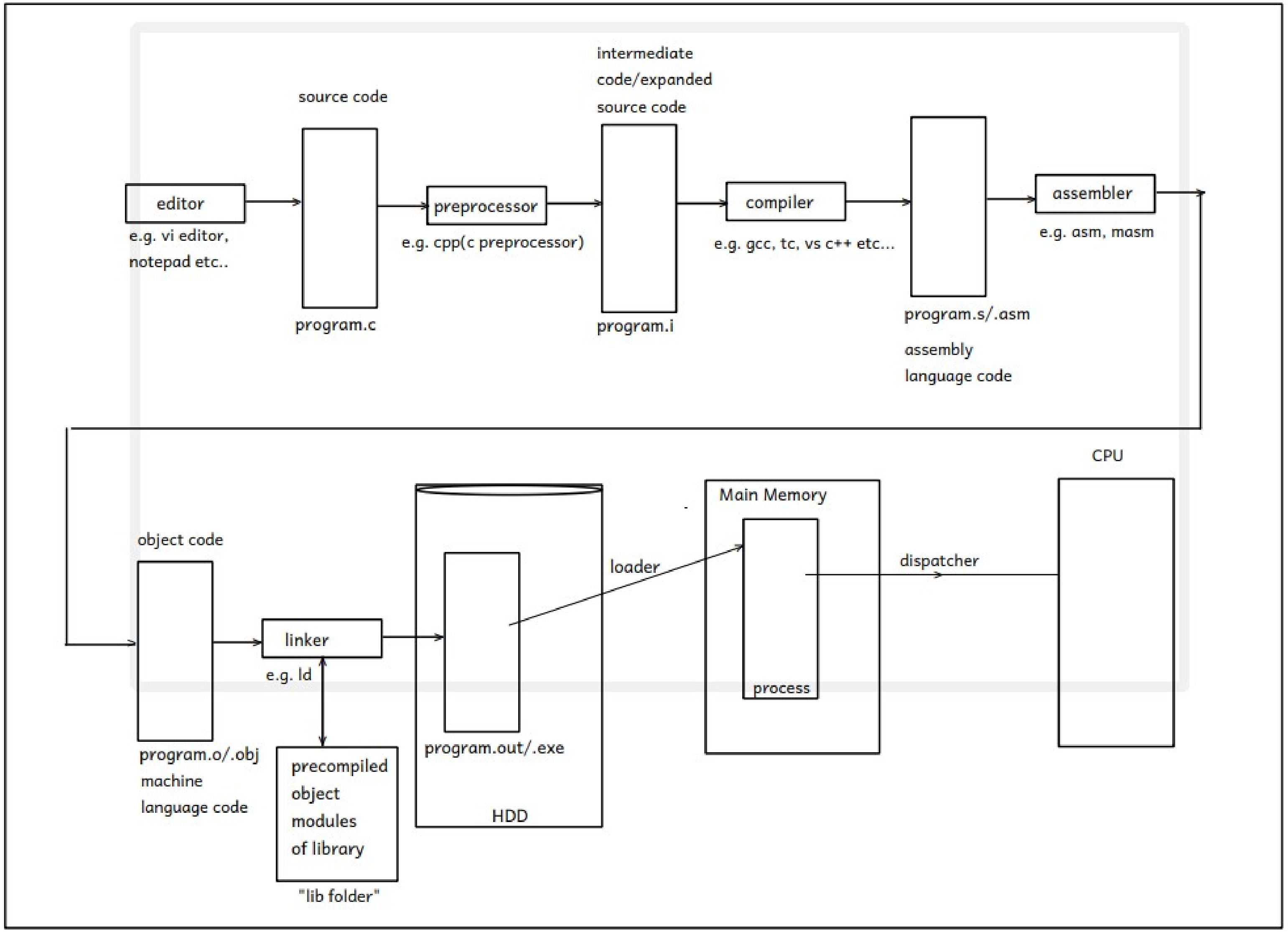
5. "Linker": it is system program which links object file/s of a program with precompiled object modules of library functions exists in a lib folder and creates final single executable file.

e.g. **ld**: link editor in Linux.

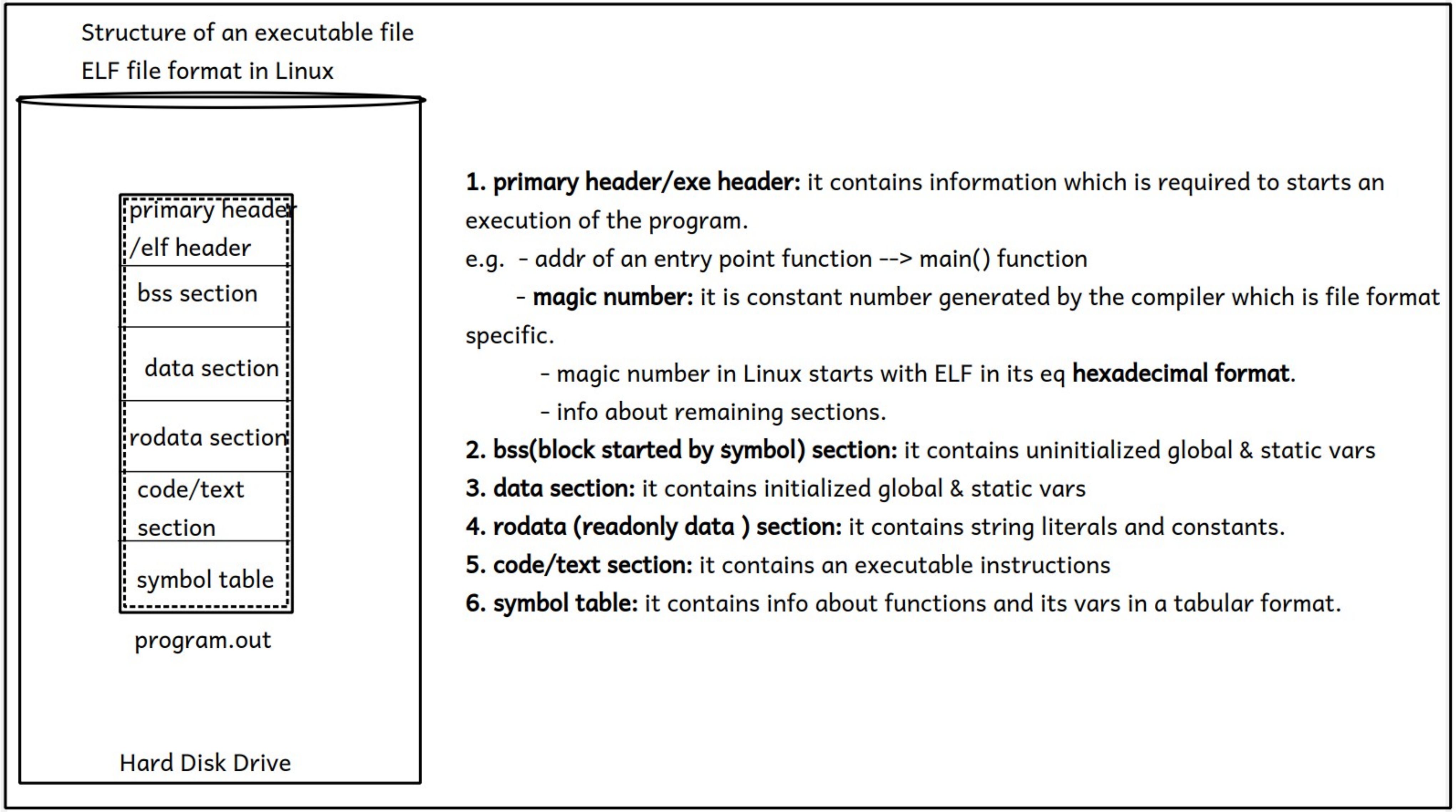
Loader : It is a system program (i.e. inbuilt program of an OS) which loads an executable file from **HDD** into the **main memory.**

Dispatcher : It is a system program (i.e. inbuilt program of an OS) which loads data & instructions of a program which is in the **main memory** onto **the CPU** (i.e. into the CPU registers).

Operating System Concepts



Operating System Concepts



1. primary header/exe header: it contains information which is required to starts an execution of the program.

e.g. - addr of an entry point function --> main() function

- **magic number:** it is constant number generated by the compiler which is file format specific.

  - magic number in Linux starts with ELF in its eq **hexadecimal format**.
  - info about remaining sections.

2. bss(block started by symbol) section: it contains uninitialized global & static vars

3. data section: it contains initialized global & static vars

4. rodata (readonly data) section: it contains string literals and constants.

5. code/text section: it contains an executable instructions

6. symbol table: it contains info about functions and its vars in a tabular format.

Q. What is an Operating System?

- An OS is a **system software** (i.e. collection of system programs) which **acts as an interface between user and hardware.**

- An OS also **acts as an interface between programs(user & application programs) and hardware.**

- An OS allocates resources like **main memory, CPU time, i/o devices access** etc... to all running programs, hence it is also called as a **resource allocator.**

- An OS controls an execution of all programs, it also controls hardware devices which are connected to the computer system and hence it is also called as a **control program.**

Q. What is an Operating System?

- An OS manages limited available resources among all running programs, hence it is also called as a **resource manager.**

- An OS is a software (i.e. collection of thousands of system programs and utitlity programs in a binary format) comes either in CD/DVD/PD, has following main components:

1. Kernel: It is a **core program/part of an OS** which runs continuosly into the main memory does **basic minimal functionalities** of it.

e.g. Linux: **vmlinuz**, Windows: **ntoskrnl.exe**

2. Utility Programs - Softwares: e.g. disk manager, windows firewall, anti-virus software etc...

3. Application Programs - Softwares: e.g. google chrome, shell, notepad, msoffice etc...

Functions of an OS:

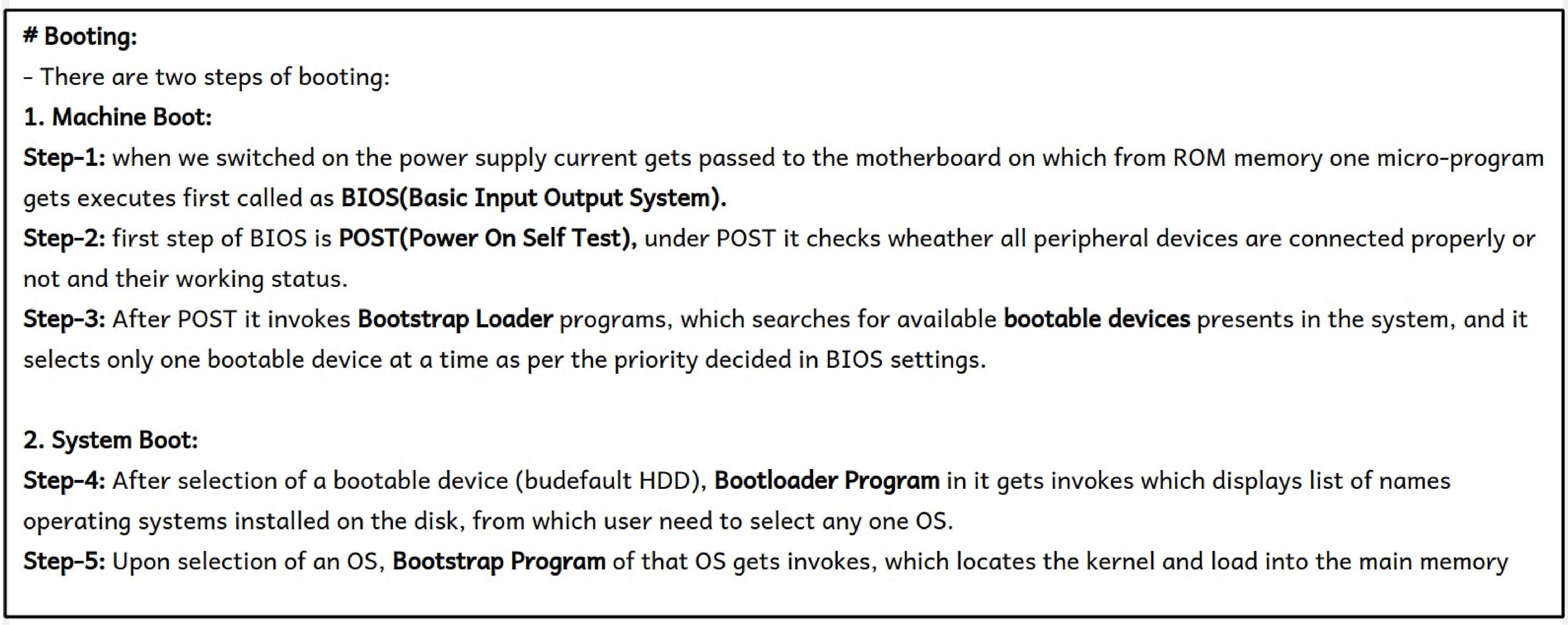
Basic minimal functionalities/Kernel functionalities – must be supported by any OS – compulsory functionalities :

1. Process Management
2. Memory Management
3. Hardware Abstraction
4. CPU Scheduling
5. File & IO Management

Extra utility functionalities/optional:

6. Protection & Security
7. User Interfacing
8. Networking

Operating System Concepts



# Booting:

- There are two steps of booting:

1. Machine Boot:

Step-1: when we switched on the power supply current gets passed to the motherboard on which from ROM memory one micro-program gets executes first called as **BIOS(Basic Input Output System).**

Step-2: first step of BIOS is **POST(Power On Self Test),** under POST it checks wheather all peripheral devices are connected properly or not and their working status.

Step-3: After POST it invokes **Bootstrap Loader** programs, which searches for available **bootable devices** presents in the system, and it selects only one bootable device at a time as per the priority decided in BIOS settings.

2. System Boot:

Step-4: After selection of a bootable device (budefault HDD), **Bootloader Program** in it gets invokes which displays list of names operating systems installed on the disk, from which user need to select any one OS.

Step-5: Upon selection of an OS, **Bootstrap Program** of that OS gets invokes, which locates the kernel and load into the main memory

# UNIX Operating System:

- UNIX: UNICS – **Uniplexed Information & Computing Services/System.**

- UNIX was developed at **AT&T Bell Labs** in US, in the decade of 1970's by **Ken Thompson, Denies Ritchie** and team.

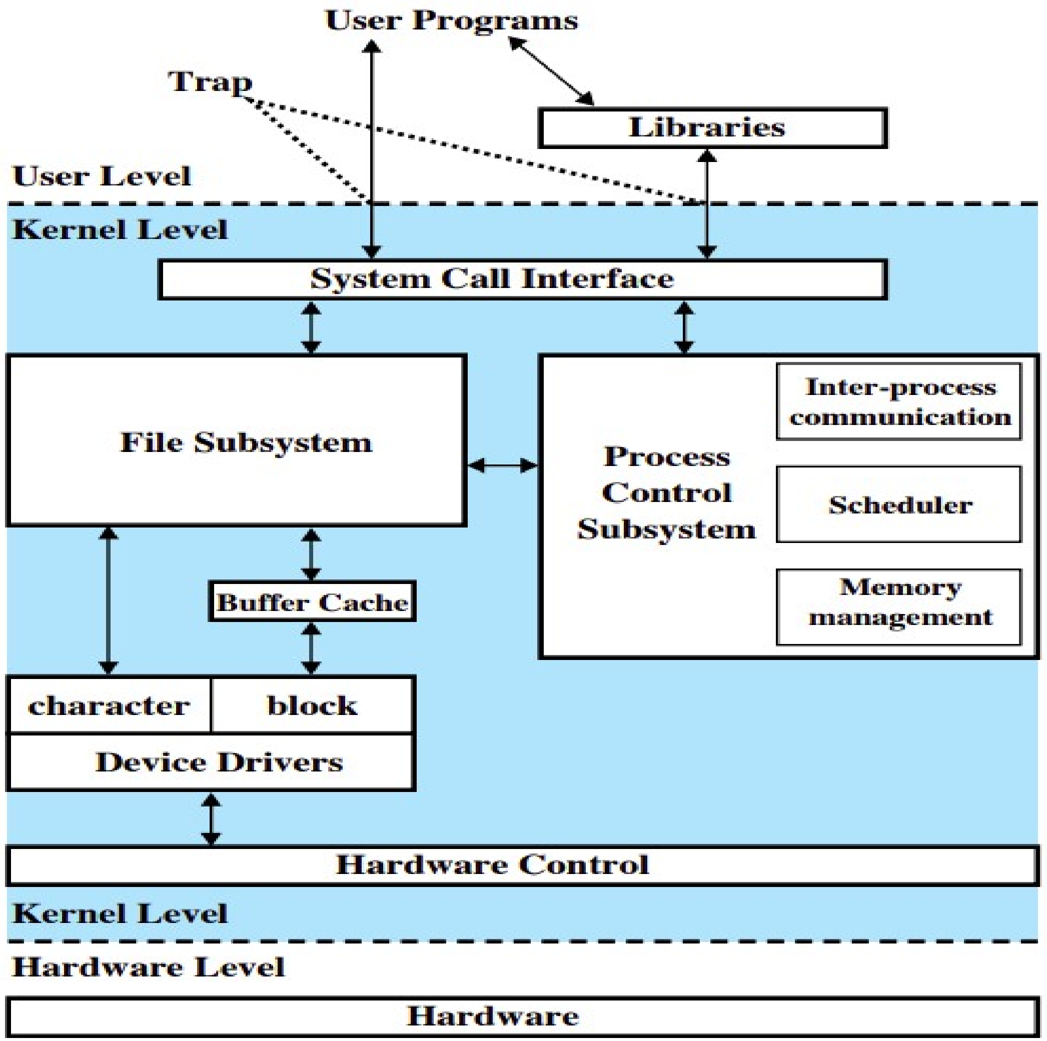
- It was first run on a machine **DEC-PDP-7** (Digital Equipment Corporation – Programmable Data Processing-7).

- UNIX is the first **multi-user, multi-programming** & **multi-tasking** operating system.

- UNIX was specially designed for developers by developers

- System architecture design of UNIX is followed by all modern OS's like Windows, Linux, MAC OS X, Android etc..., and hence UNIX is referred as **mother of all modern operating systems.**

Operating System Concepts



Operating System Concepts

- Kernel acts as an interface between programs and hardware.

- Operating System has subsystems like **System Call Interface Block, File Subsystem Block, Process Control Subsystem Block (which contains IPC, Memory Management & CPU Scheduling), Device Driver, Hardware Control/Hardware Abstraction Layer.**

- There are two major subsystems:

1. Process Control Subsystem

2. File Subsystem

- In UNIX, whatever is that can be stored is considered as a file and whatever is active is reffered as a process.

- File has space & Process has life.

Operating System Concepts

- From UNIX point of view all devices are considered as a file

- In UNIX, devices are catagorises into two catagories:

1. Character Devices: Devices from which data gets transfered character by character --> character special device file

e.g. keyboard, mouse, printer, monitor etc...

2. Block Devices: Devices from which data gets transfered block by block --> block special device file

e.g. all storage devices.

- **Device Driver:**It is a program/set of programs enable one or more hardware devices to communicate with the computer’s operating system.