Operating System Concepts

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

Operating System Concepts

BATCH NAME : CRASH COURSE OC06

MODULE NAME : OPERATING SYSTEM CONCEPTS

NO. OF HRS. : 8 HRS

- Crash course is designed only for exam oriented preparation of CCAT, kind of revision sessions and only introduction of topics in brief.

Q. Why there is a need of an OS?

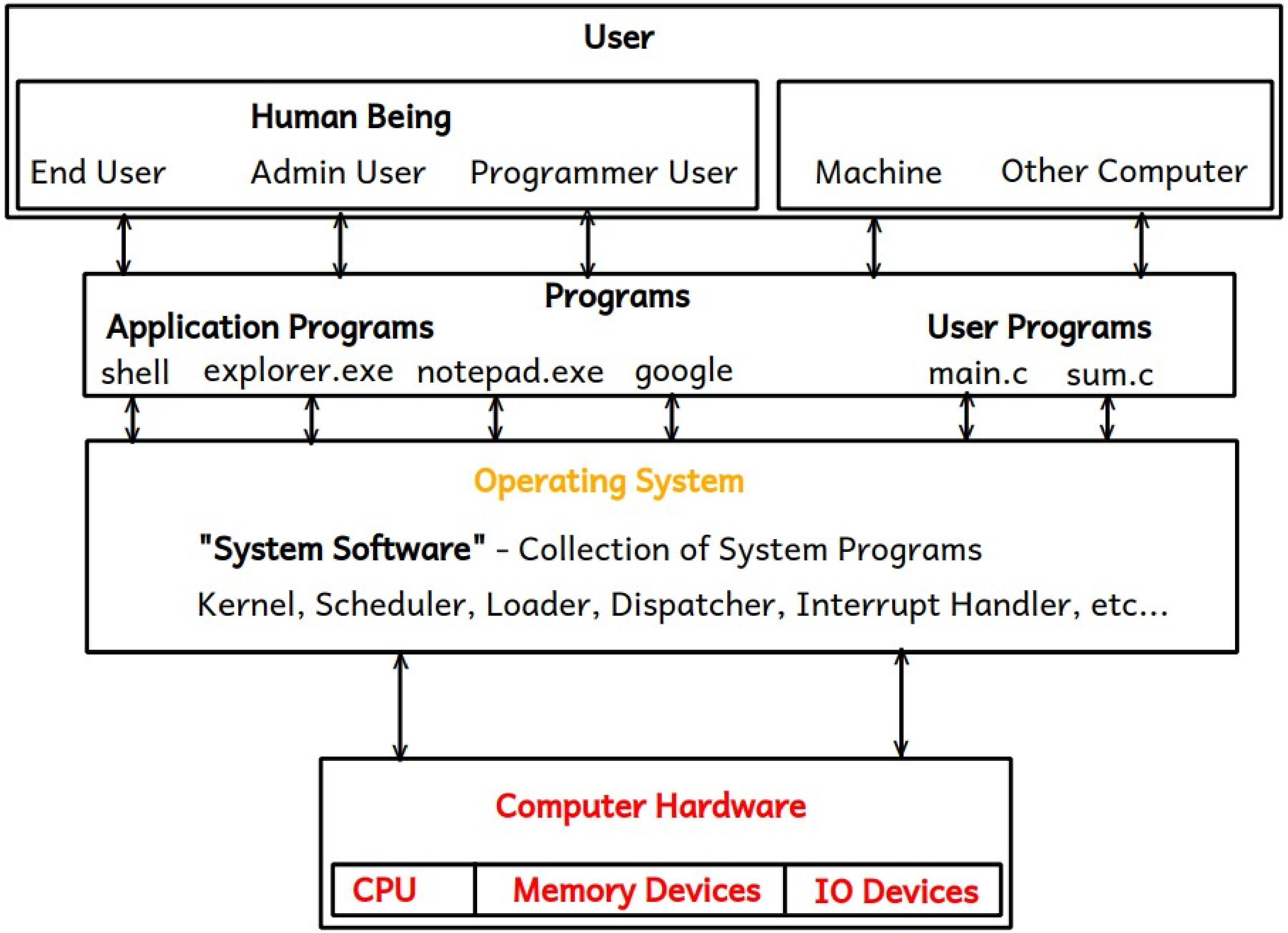
- Computer is a machine/hardware does different tasks efficiently & accurately.

- Basic functions of computer:

  1. Data Storage
  2. Data Processing
  3. Data Movement
  4. Control

- As any user cannot communicates/interacts directly with computer hardware to do different tasks, and hence there is need of some interface between user and hardware.

Operating System Concepts



User

Human Being

End User Admin User Programmer User

Machine Other Computer

Programs

Application Programs

shell explorer.exe notepad.exe google

User Programs

main.c sum.c

Operating System

"System Software" - Collection of System Programs

Kernel, Scheduler, Loader, Dispatcher, Interrupt Handler, etc...

Computer Hardware

CPU Memory Devices IO Devices

Operating System Concepts

- As any user cannot directly interacts with an OS, hence an OS gives two types of interfaces for the user in the form of programs:

1. CUI/CLI : Command User Interface/Command Line Interface

- in this type of interface, user can interacts with an OS by means of entering commands in a text format throgh command line.

e.g. cd, cp, gcc, ls, mv etc....

- In Linux name of the program which provides CUI is **shell/terminal**

- In Windows name of the program which provides CUI is **cmd.exe/command prompt**

2. GUI: Graphical User Interface

- in this type of interface, user can interacts with an OS by means of making an events like left click, right click, click on buttons, menu bar, menu list etc....

- In Linux name of the program which provides GUI is **GNOME/KDE**

- In Windows name of the program which provides GUI is **explorer.exe**

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.

- Three 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/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....

1. "Editor": it is an application program also used to write a source code in any programming lanaguage.

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(c preprocessor)

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...

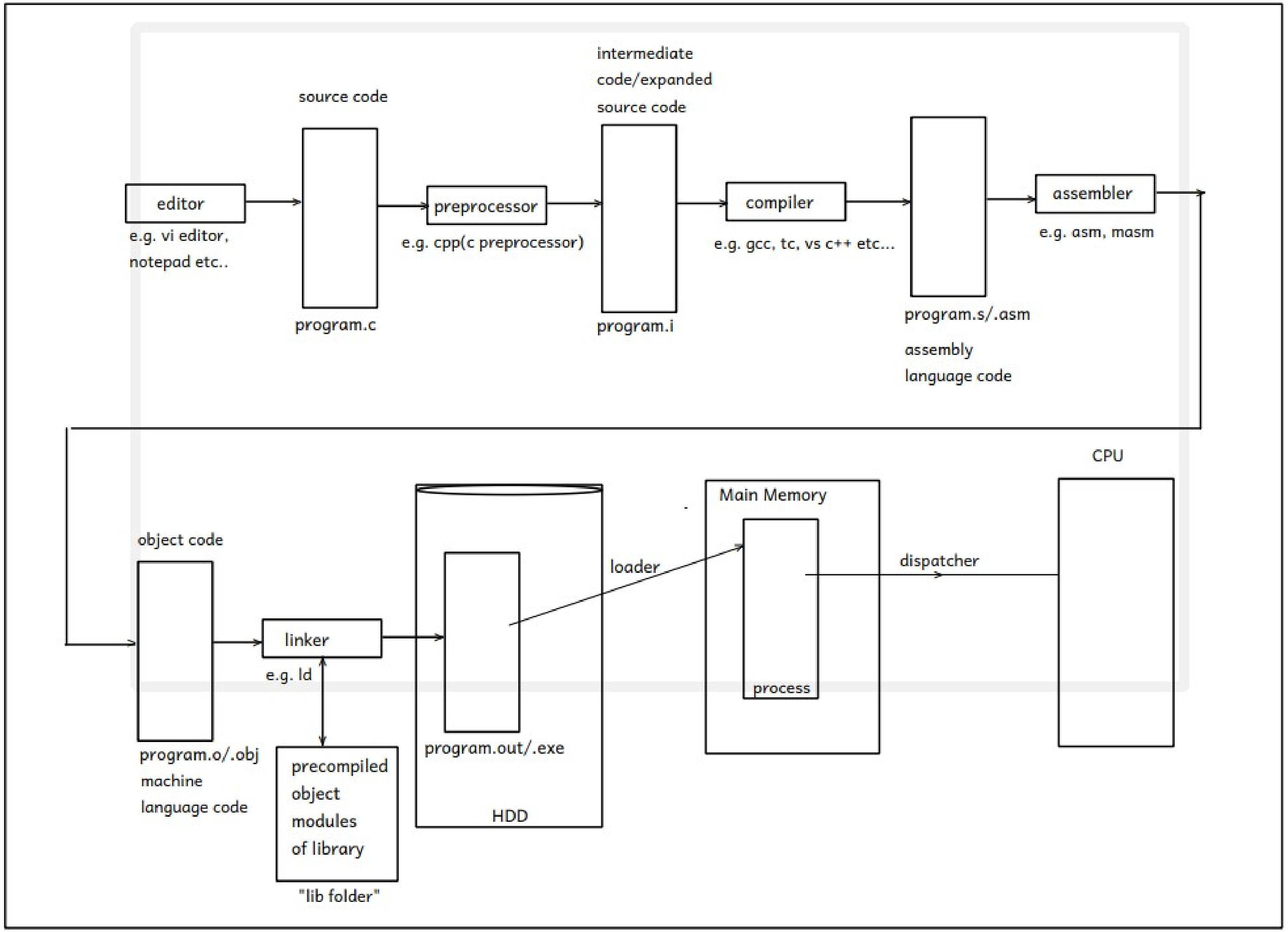
4. "Assembler": it is an application program which converts assembly language code into machine language code/object code.

e.g. masm, tasm etc...

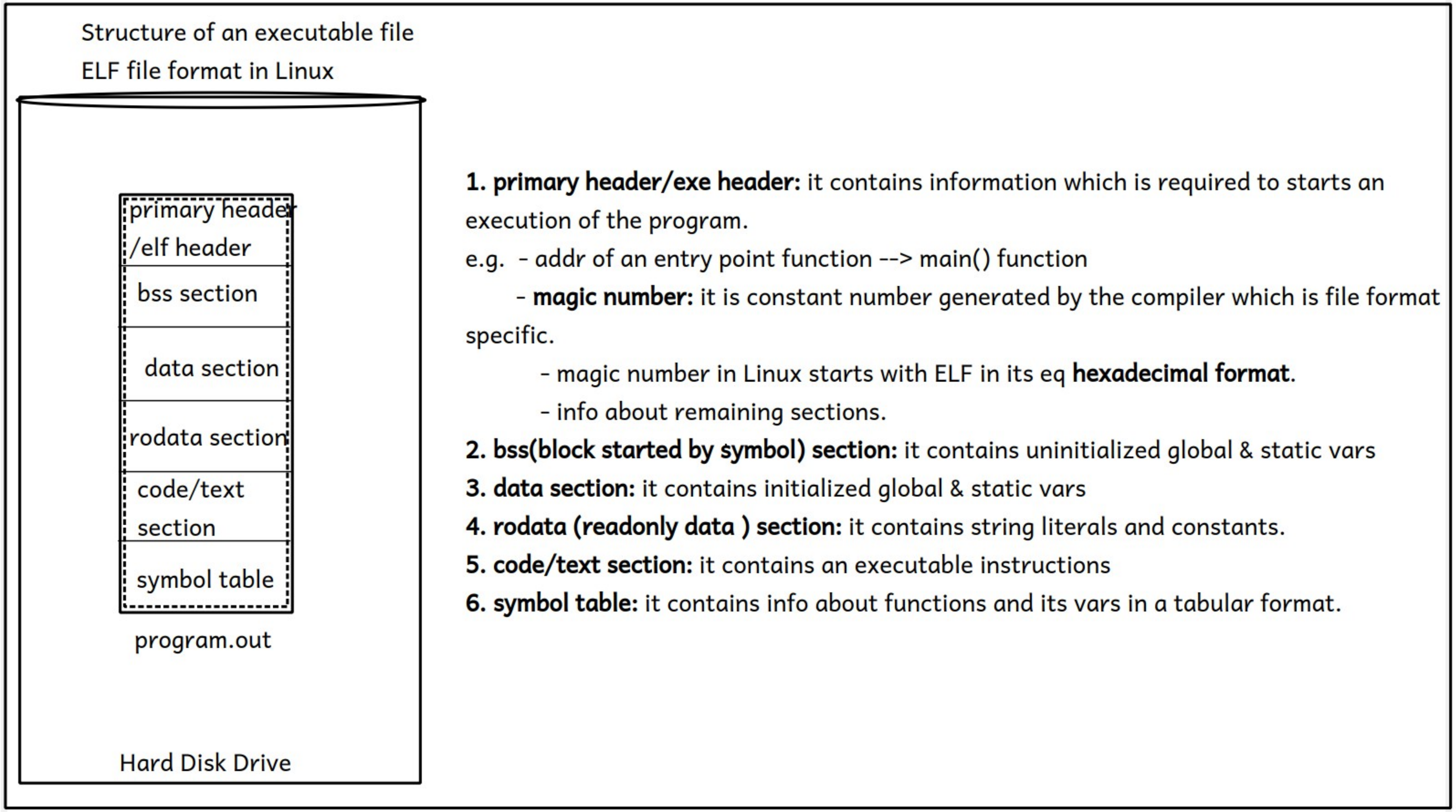
5. "Linker": it is an application program which links object file/s in a program with precompiled object modules of library functions exists in a lib folder and creates final single executable file.

e.g. ld: link editor in Linux.

Operating System Concepts



Operating System Concepts


Structure of an executable file
ELF file format in Linux
primary header
/elf header
bss section
data section
rodata section
code/text
section
symbol table
program.out
Hard Disk Drive

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 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 and 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.**

- From End User: An OS is a software (i.e. collection of programs) comes either in CD/DVD, 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 Softwares: e.g. disk manager, windows firewall, anti-virus software etc...

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

Functions of an OS:

Basic minimal functionalities/Kernel 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

OS installation: to install any OS onto the machine, is noting but to store **OS software** (i.e. collection of thousands of system programs and application programs which are in a binary format) onto the HDD.

Booting:

- If an OS wants to becomes active, then first atleast its core program i.e. kernel must be laoded into the main memory.

- Loading of kernel program of an OS from HDD into the main memory is called as booting.

- **Bootstrap program** is responsible to load kernel from the HDD 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.

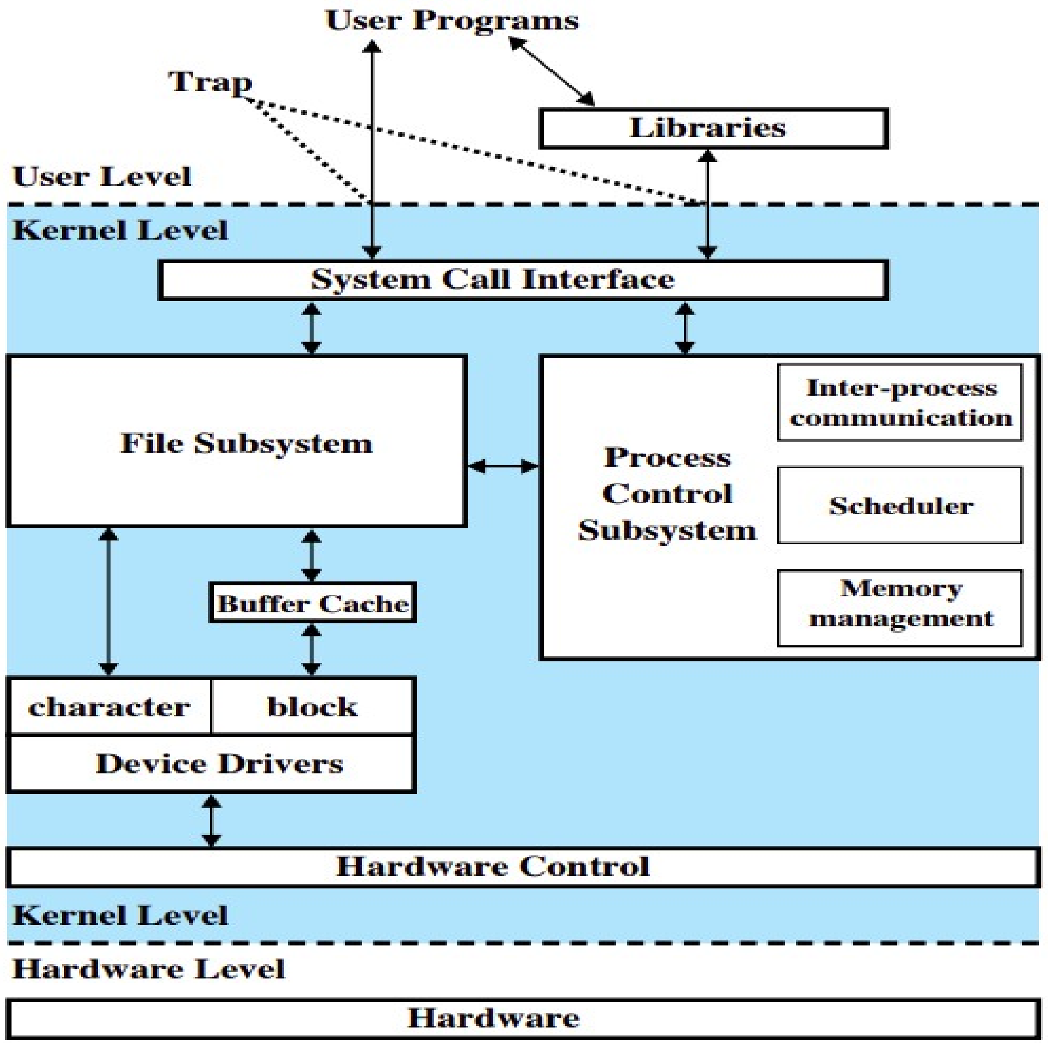
- It was first run on a machine **DEC-PDP-7** (Digital Equipment Corporation – Programmable Data Processing-7).

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

- UNIX was basically 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



- Kernel acts as an interface between programs and hardware.

- Operating System has subsystems like **System Call Interface, File subsystem, Process Control Subsystem(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 that can be stored is considered as a **file** and whatever is in a 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.

Operating System Concepts

- Hardware Control Layer/Block does communication with control logic block i.e. controller of a hardware.

# System Calls: are the functions defined in a C, C++ & Assembly languages, which provides interface of services made available by the kernel for the user (programmer user).

- If programmers want to use kernel services in their programs, it can be called directly through system calls or indirectly through set of libary functions provided by that programming language.

- There are 6 catagories of system calls:

1. Process Control System Calls: e.g. fork(), _exit(), wait() etc...

2. File Operations System Calls: e.g. open(), read(), write(), close() etc...

3. Device Control System Calls: e.g. open(), read(), write(), ioctl() etc...

4. Accounting Information System Calls: e.g. getpid(), getppid(), stat() etc...

5. Protection & Security System Calls: e.g. chmod(), chown() etc...

6. Inter Process Communication System Calls: e.g. pipe(), signal(), msgget() etc...

- In UNIX 64 system calls are there.

- In Linux more than 300 system calls are there

- In Windows more than 3000 system calls are there

- When system call gets called the CPU switched from user defined code to system defined code, and hence system calls are also called as **software interrupts/trap.**