

---

# Operating System Concepts

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

**Trainer: Akshita Chanchlani**  
**Email: [akshita.chanchlani@sunbeaminfo.com](mailto:akshita.chanchlani@sunbeaminfo.com)**



## \* **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 categories
- 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



## \* **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.

- Basic functions of computer:

1. Data Storage

2. Data Processing

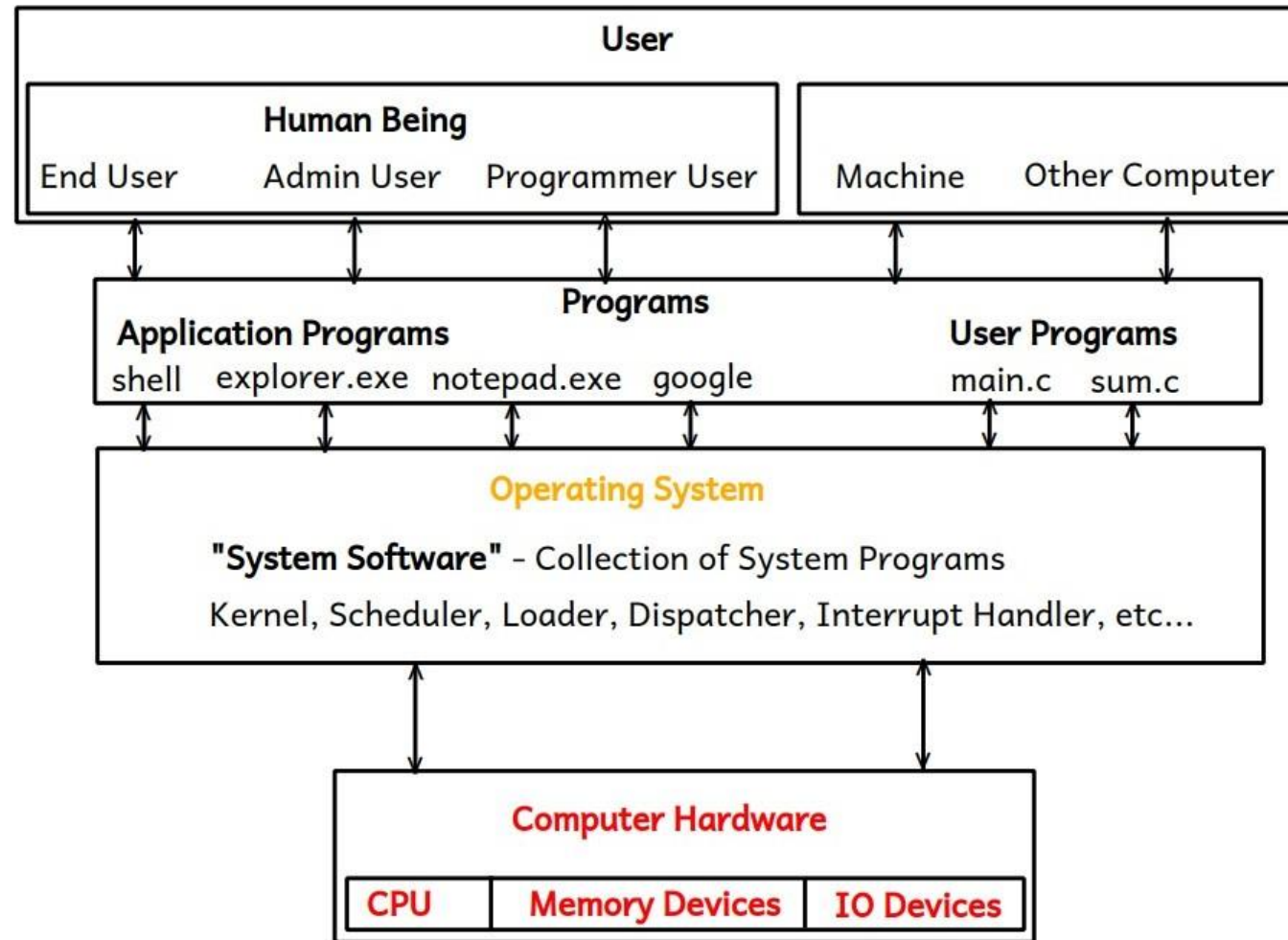
3. Data Movement

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



# 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...





# Operating System Concepts

---

## Q. What is an IDE (Integrated Software Development) ?

- It is an application software i.e. collection of tools/programs like **source code editor, preprocessor, compiler, linker, debugger** etc... required for **faster software development**.  
e.g. VS code editor, MS Visual Studio, Netbeans, Android Studio, Turbo C etc....

1. **"Editor"**: it is an application program used for 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 convert 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.

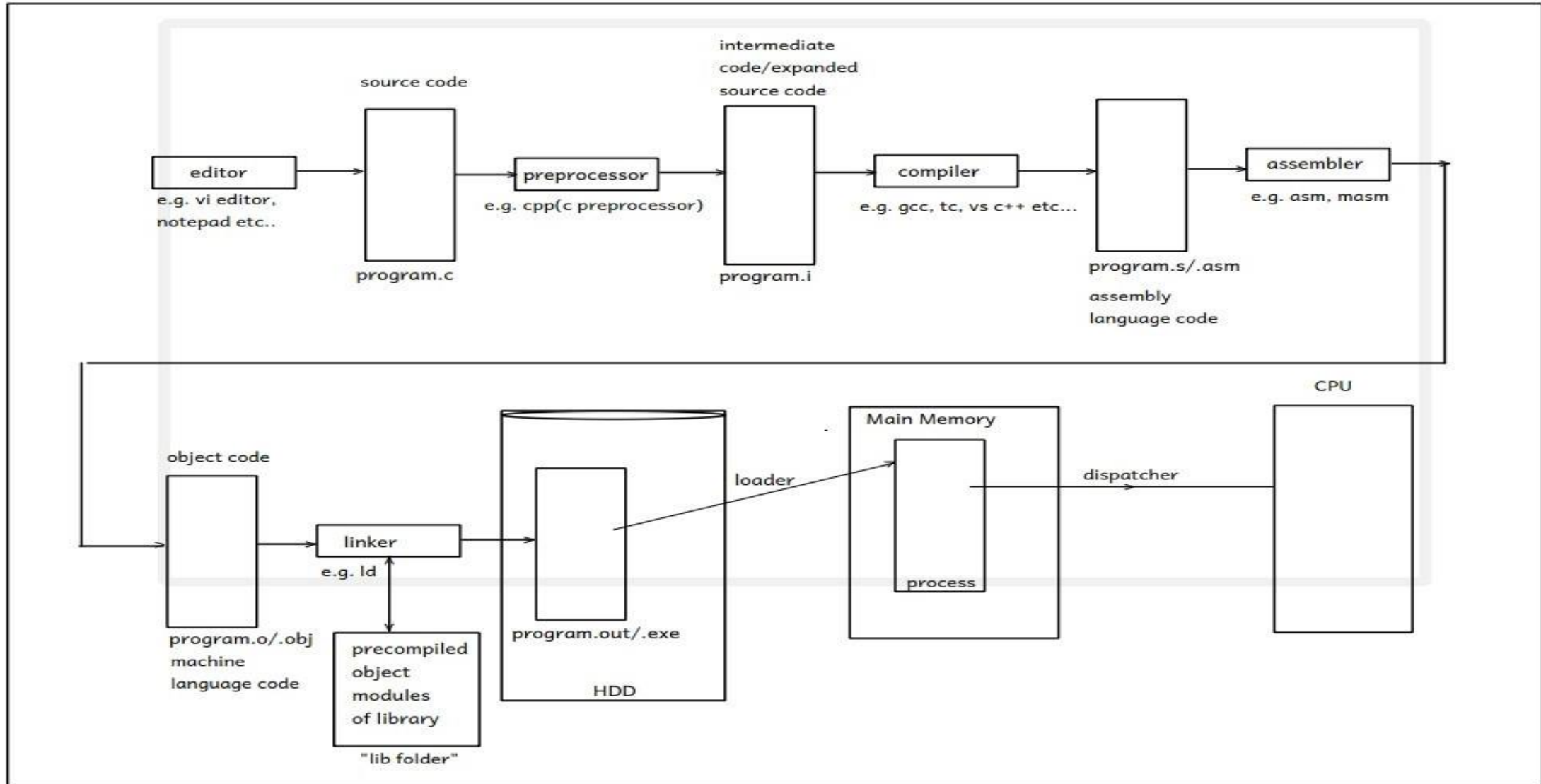
e.g. masm, tasm etc...

- Program written in any programming language is called as a **"source code"**.

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



# Interaction with an OS : Two Types of Interface (CUI and GUI)

---

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

- by using this kind of interface user can interact with an OS by means entering commands onto the terminal/command line in a text format.

e.g. In Windows name of the program which provides CUI => cmd.exe command prompt

In Linux name of an application program which provides CUI => shell/terminal

In MSDOS name of the program which provides CUI => command.com (Microsoft Disk Operating System).

## 2. GUI : Graphical User Interface

- by using this kind of interface user can interact with an OS by means making an event like click on buttons, left click/right click/double click, menu bar, menu list etc.....

- Windows = User friendly GUI.

e.g. In Windows name of an application program which provides GUI => explorer.exe

In Linux name of an application program which provides GUI => GNOME/KDE (GNU Network Object Model Environment / Common Desktop Environment).



## 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 continuously 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



**Basic Minimal Functionalities/Core  
Functionalities => "Kernel"**

Process Management

Memory Management

CPU Scheduling

File & IO Management

Hardware Abstraction

Functions of  
OS

**Optional Functionalities/Extra utility  
Functionalities => "Utility Softwares"**

User Interfacing

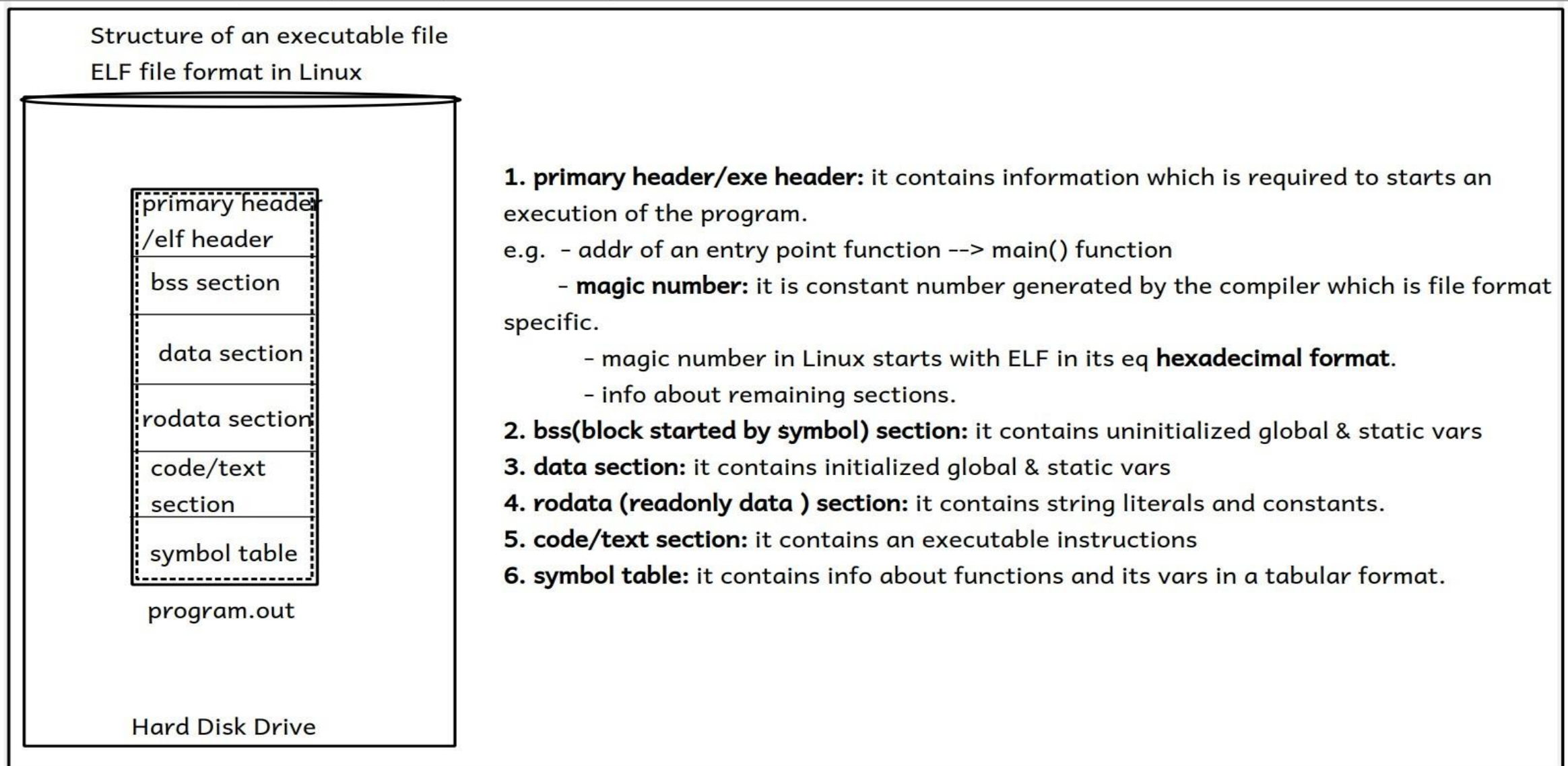
Networking

Protection & Security





# Operating System Concepts



# File Format

---

- file format of an executable file in Windows is PE (Portable Executable), whereas file format of an executable file in Linux is **ELF (Executable & Linkable Format)**.
- file format is a specific way to store data & instructions of a program inside an executable file, and it is different in diff OS.
- in Linux file format of an executable file is ELF:
- ELF file format divides an executable file logically into sections and inside each section specific contents can be kept in an organized manner:
  1. elf header
  2. bss section (block started by symbol)
  3. data section
  4. rodata (read only data )section
  5. code/text section
  6. symbol table



---

# Thank you

Email: [akshita.chanchlani@sunbeaminfo.com](mailto:akshita.chanchlani@sunbeaminfo.com)

