Operating System

DAY01

Mrs. Akshita. S. Chanchlani

akshita.chanchlani@sunbeaminfo.com

Operating System Concepts

- **Introduction:**
  - Why there is need of an OS?
  - What is an OS?
  - 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 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

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.**

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:

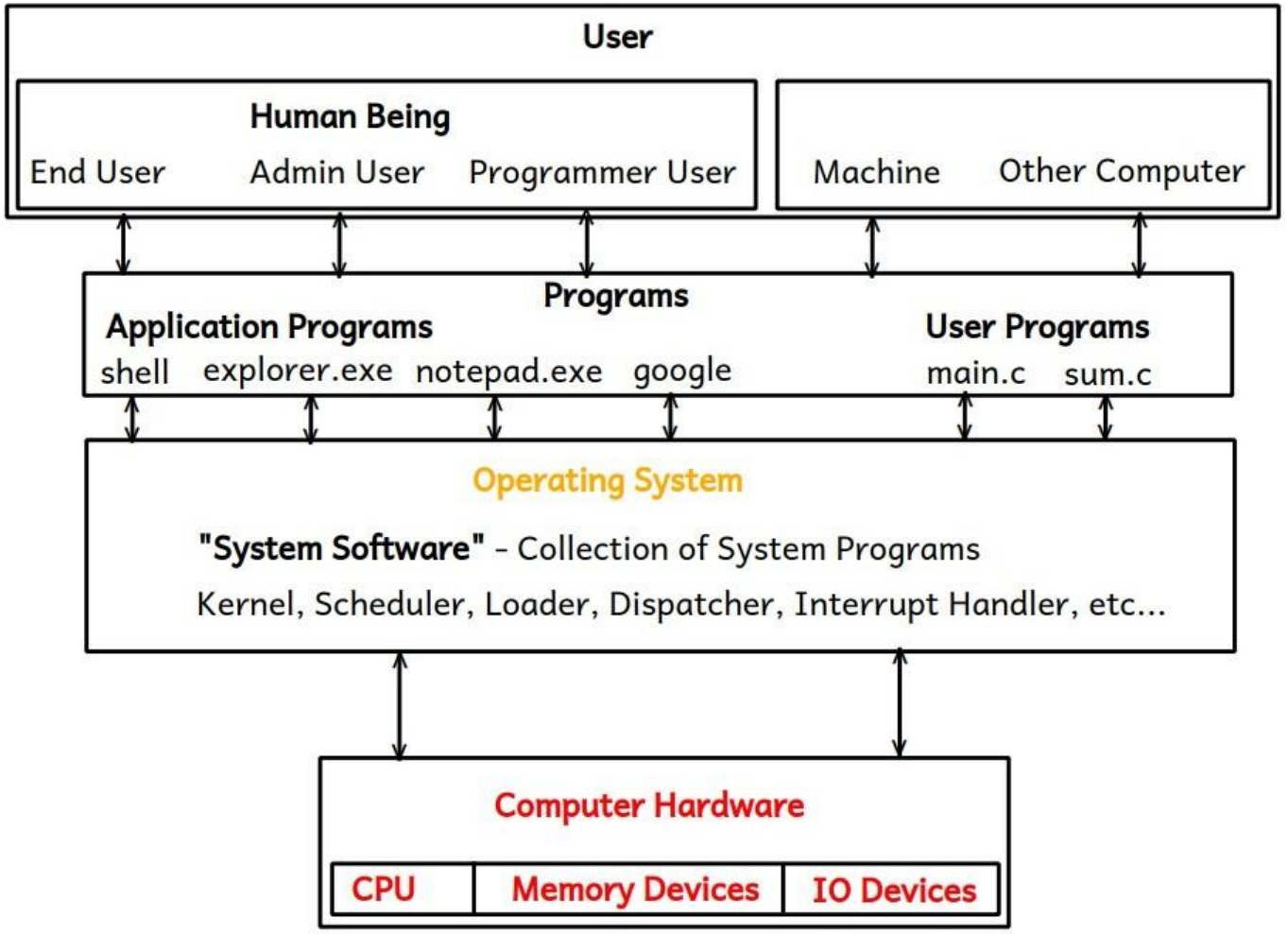
1. Kernel: It is a core program/part of an OS which runs continuosly into the main memory does basic minimal functionalities of it.

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...

Operating System Concepts



Need of 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.

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 interacts with an OS by means entering commands onto the terminal/command line in a text format.

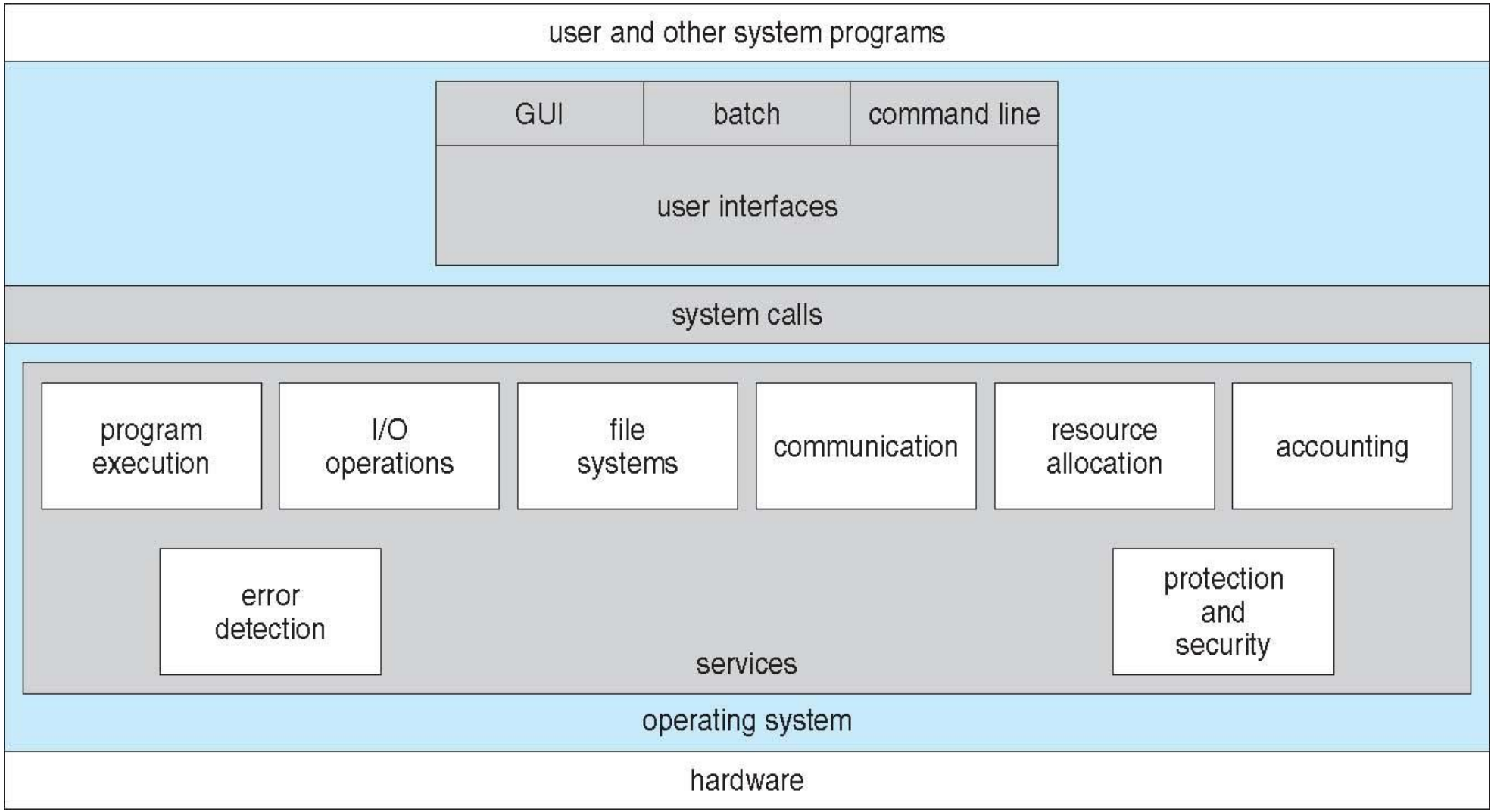
e.g. In Windows name of the program which provide CUI => cmd.exe command prompt
In Linux name of an application program which provides CUI => shell/terminal
In MSDOS name of the program which provides CUI => command.com (MicroSoft Disk Operating System).

2. GUI : Graphical User Interface

- by using this kind of interface user can interacts with an OS by means making an events like click on buttons, left click/rigyht click/double click, menu bar, menu list etc.....
- Windows = User friendly GUI.

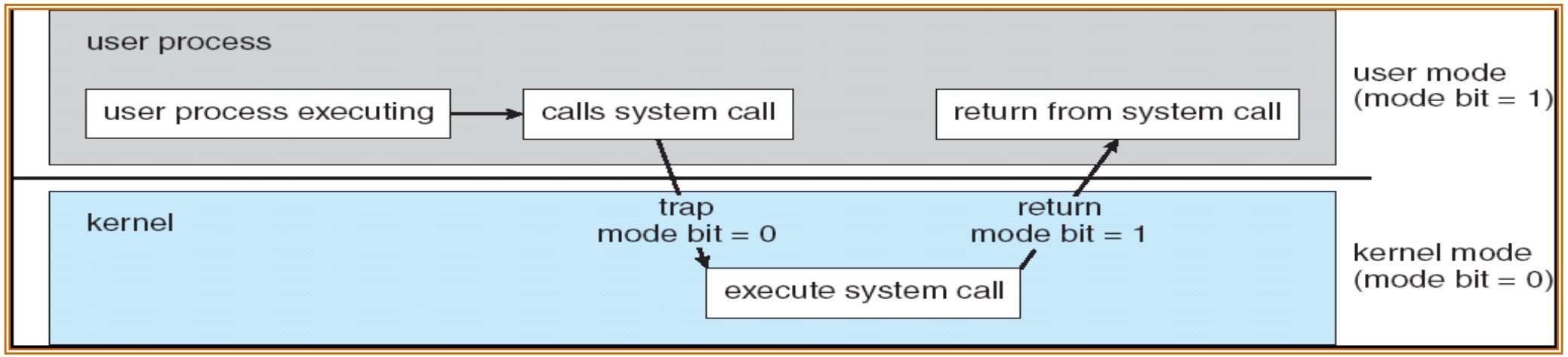
e.g. In Windows name of an application program which provides GUI => explorer.exe
In Linux name of an application program which provides GUI => GNOME/KDE (GNU Network Object Model Environment / Common Desktop Environment).

Operating System View

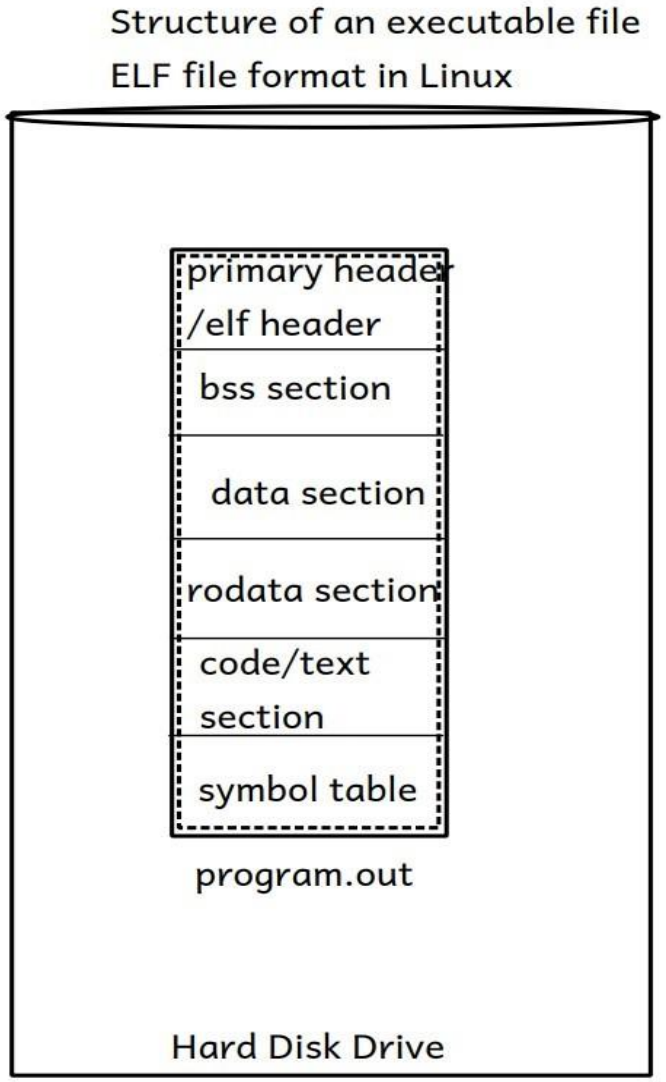


Dual Mode Operation

- Allows OS to protect itself and other system components
  - **User mode** and **kernel mode**
  - **Mode bit** provided by hardware
    - Provides ability to distinguish when system is running user code or kernel code
    - Some instructions designated as **privileged**, only executable in kernel mode
- To perform privileged operations, must transit into OS through well defined interfaces
  - System calls
  - Interrupt handlers



System View of File



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.

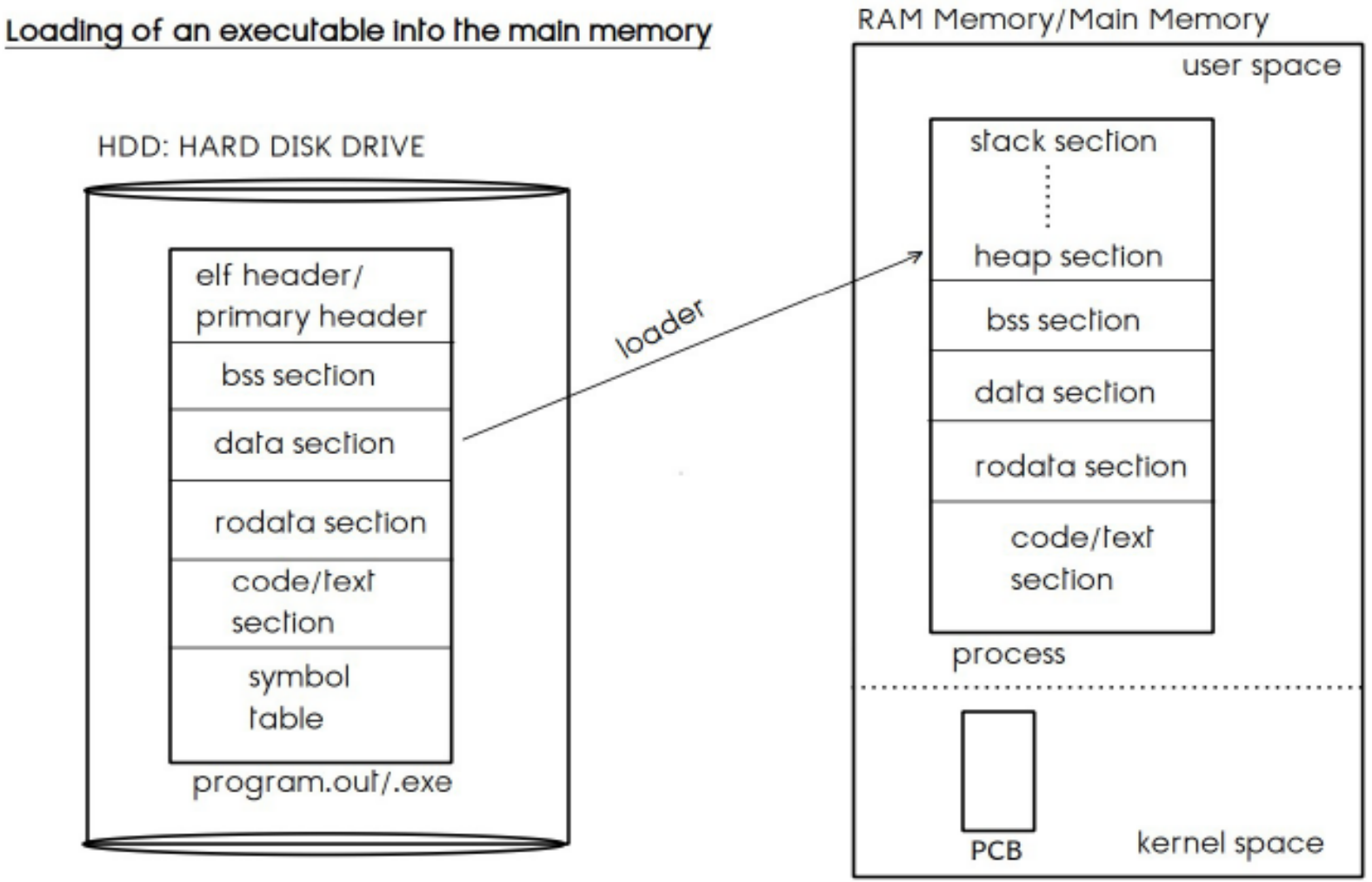
2. **bss(block started by symbol) section:** it contains uninitialized global & static vars
3. **data section:** it contains initialized global & static vars
4. **rodata (readonly data) section:** it contains string literals and constants.
5. **code/text section:** it contains an executable instructions
6. **symbol table:** it contains info about functions and its vars in a tabular format.

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

Process Loaded into the main memory



Thank You