ISLM-225

Unit-2
System Structure

Department of Information Science & Library Management (ISLM)
University of Rajshahi, Rajshahi-6205, Bangladesh

Unit-1: Operating System Overview: Introduction to OS. Operating system functions, evaluation of O.S., Different types of O.S.: batch, multi-programmed, time-sharing, real-time, distributed, parallel.

Unit-2: System Structure: Computer system operation, I/O structure, storage structure, storage hierarchy, different types of protections, operating system structure (simple, layered, virtual machine), O/S services, system calls.

Unit-3: Installing and Configuring OS: Introduction to Installation and Media Types, Performing a Custom OS Installation, Run Levels and the Startup/Shutdown Sequence, Logging In and Out of a Operating System.

Unit-4: Process Management: Processes- Concept of processes, process scheduling, operations on processes, co-operating processes, interprocess communication, Threads- overview, benefits of threads, user and kernel threads., CPU scheduling, process synchronization, deadlocks- system model, deadlock characterization, methods for handling deadlocks, deadlock prevention, deadlock avoidance, deadlock detection, recovery from deadlock.

Unit-6: Storage Management: Memory Management- background, logical vs. physical address space, swapping, contiguous memory allocation, paging, segmentation, segmentation with paging, Virtual Memory- background, demand paging, performance, page replacement, page replacement algorithms (FCFS, LRU), allocation of frames, thrashing, File Systems, I/O Management, Disk Management.

Unit-7: Distributed OS and File System: Motivation, Types of Network-based OS, Network structure, Distributed File System- Background, Naming and transparency, Remote File Access, State full and Stateless services. Distributed Synchronization: Event Ordering, Mutual Exclusion, Atomicity, Concurrency Control, Deadlock Handling, Election algorithm and Reaching agreement.

Computer System Operation

- **Every general-purpose computer** must have an OS to run other programs.
- OSs perform basic tasks, such as …….
  - recognizing input from the keyboard,
  - sending output to the display screen,
  - keeping track of files and directories on the disk, and
  - controlling peripheral devices such as disk drives and printers.

Computer System Operation

- **For large systems**, the OS has even greater responsibilities and powers.
  - It is like a **traffic cop** – it makes sure that different programs and users running at the same time do not interfere with each other.
  - The OS is also responsible for security, ensuring that unauthorized users do not access the system.

OS Services

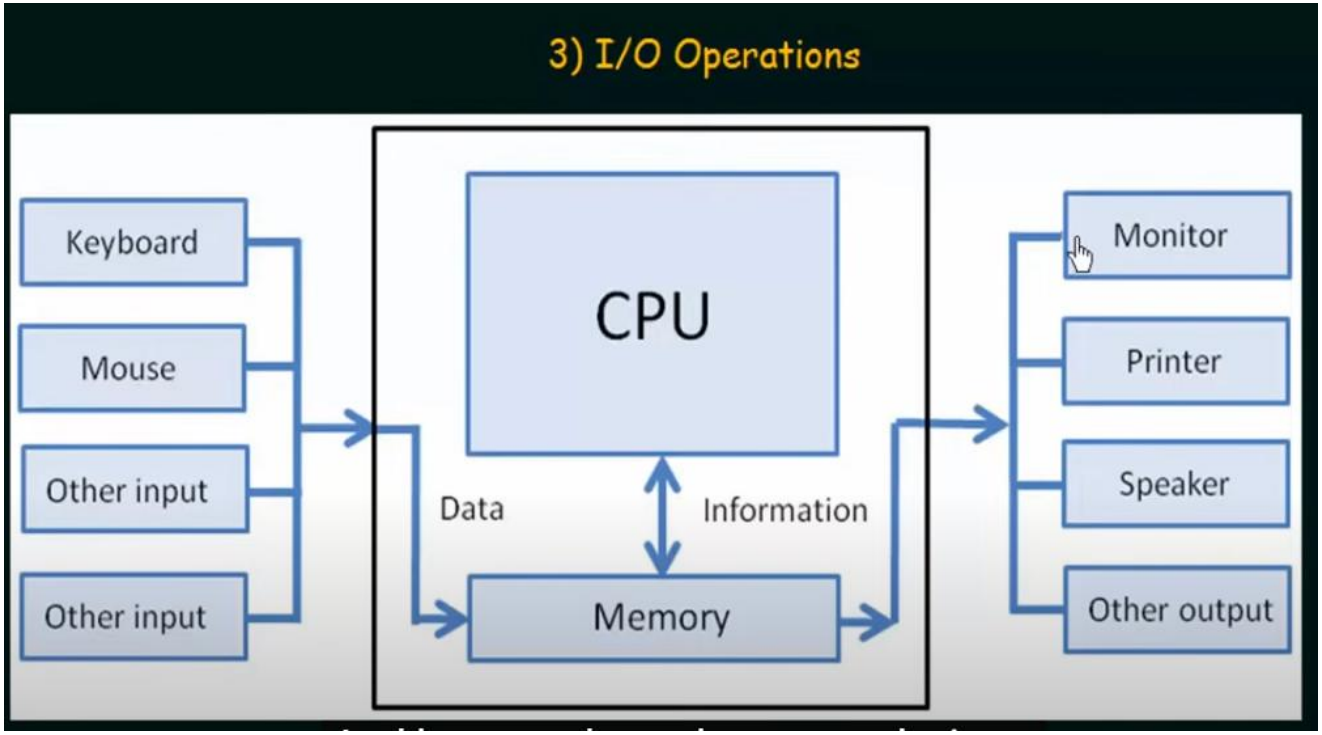
- **The various OS services are:**
  - Program Execution
  - I/O Operations
  - File System Manipulation
  - Communications
  - Error Detection
  - Resource Allocation
  - Accounting
  - Protection and Security

OS Services-Program Execution

- **The purpose of a computer systems is to allow the user to execute programs**. So the OS provides an environment where the user can conveniently run programs.
- **The user does not have to worry about the memory allocation or multitasking or anything.** These things are taken care of by the OSs.
- Running a program involves the allocating and deallocating memory, CPU scheduling in case of multiprocess.
- These functions cannot be given to the user-level programs. So **user-level programs cannot help the user to run programs independently without the help from OS.**

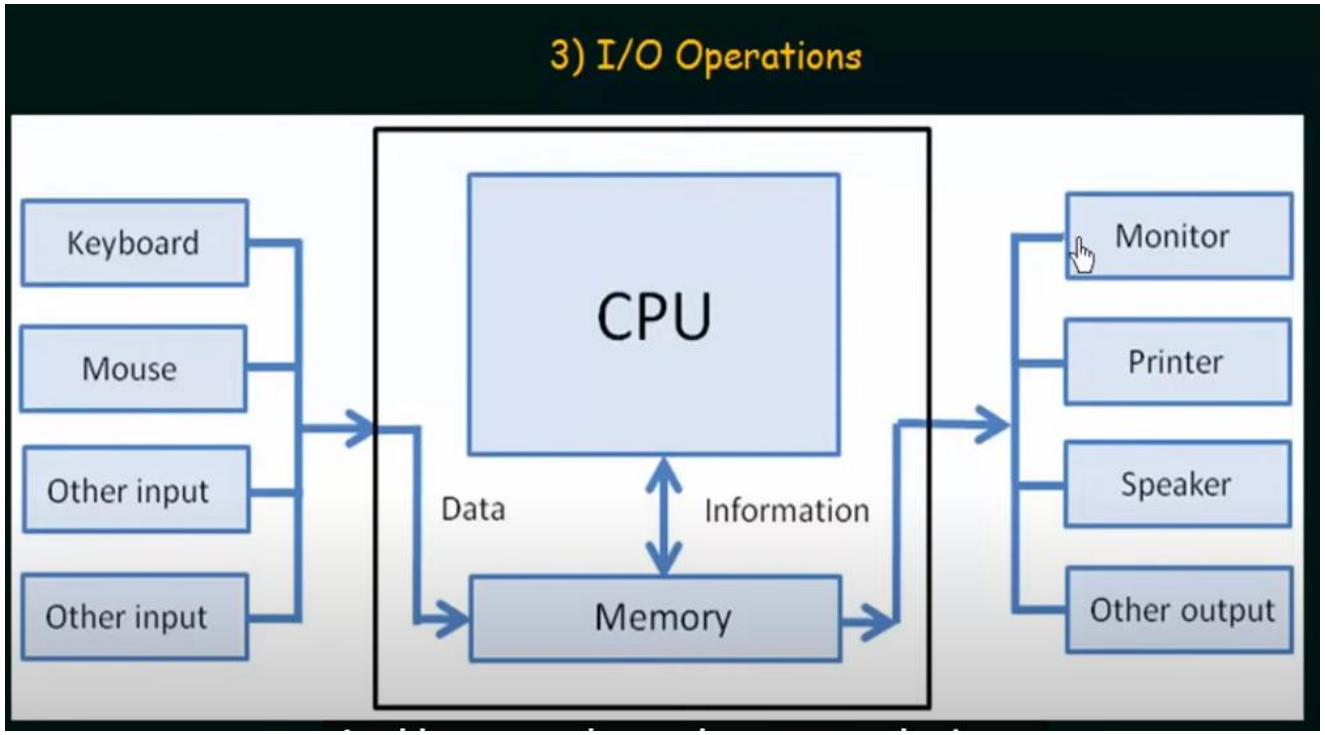
OS Services-I/O Operations

- Each program requires an input and produces output. This involves the use of I/O.
- The OSs hides the user the details of underlying hardware for the I/O.



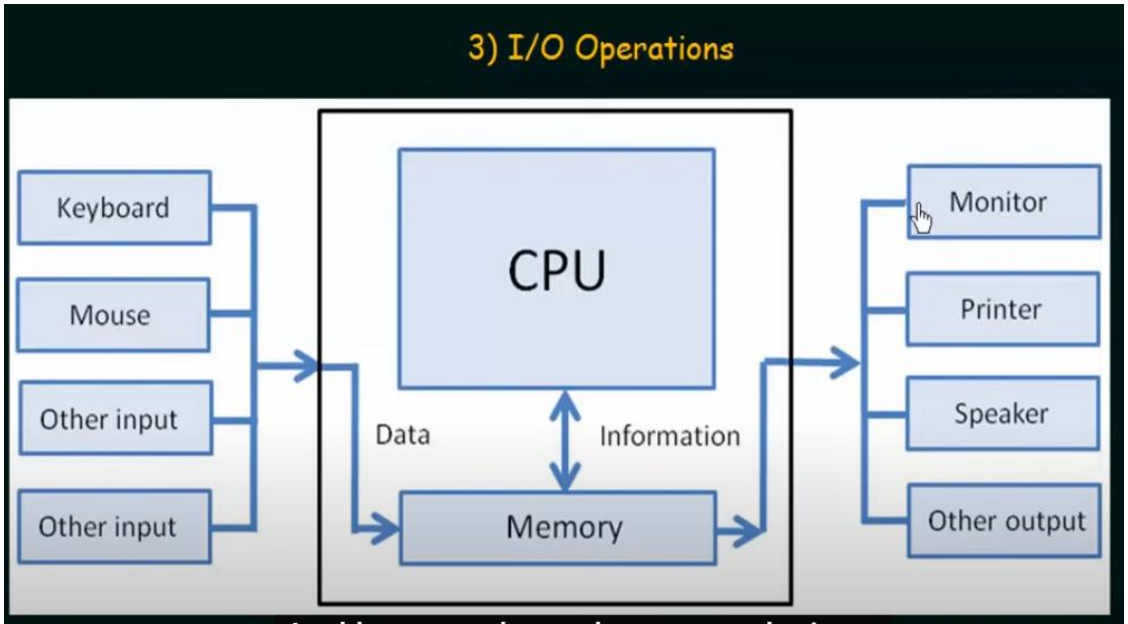
OS Services-I/O Operations

- All the user sees is that the I/O has been performed without any details.
- So the OSs by providing I/O makes it convenient for the users to run programs.



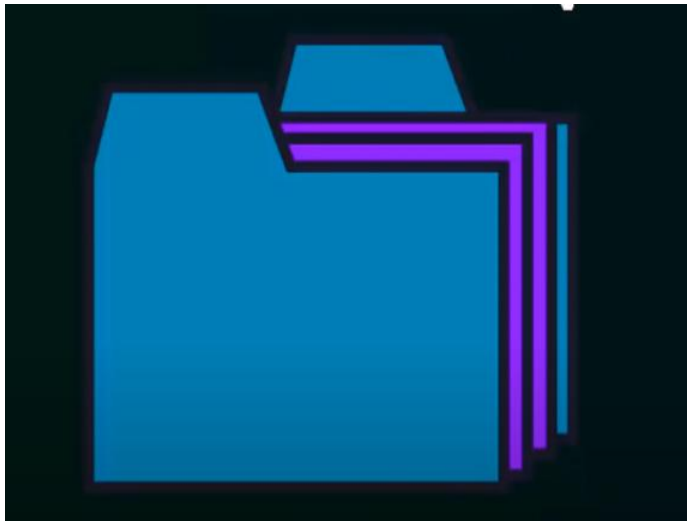
OS Services-I/O Operations

- A user can not directly control the I/O devices.
- Now you may think that when you are using your keyboards, and mouse, you feel like you are controlling it by yourself directly.
- But there is an OS between you and your system that actually controls the usage of the I/O devices.
- So, this I/O operation is also one of the most important operations.



OS Services-File System Manipulation

- Now the next services that is provided by the OS is known as File system manipulation.
- The output of a program may need to be written into new files or input taken from some files.
- There are many files and directories in our system that has to be used.
- Now the OS must control how this files are manipulated or managed.
- Some time we may have to create, delete, modify, or search for a given file
- So, all these are controlled by the OS.



OS Services-File System Manipulation

- And, also it controls the permission that is given to certain programs or users for the access of certain files.
- So, all the files that we have can not be allowed to be used by every program or by every user.
- There are access restriction.
- So, this restriction of access is also controlled by the OS.
- Thus, this is another important services that is provided by the OS.

OS Services-Communication

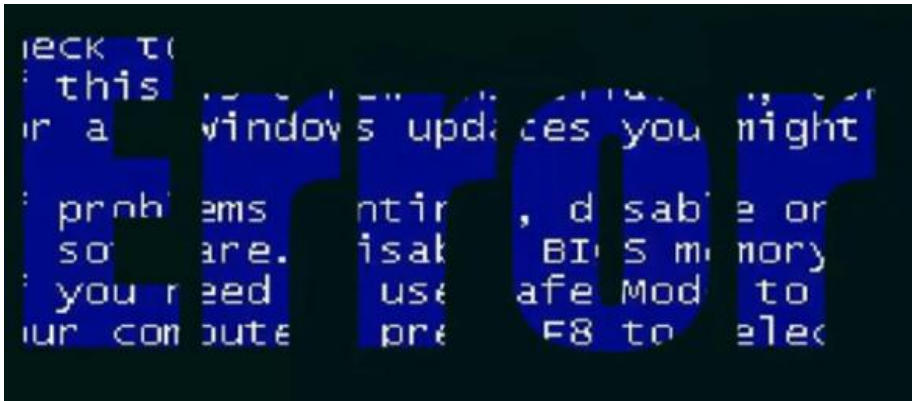
- The next services provided by the OS is communications.
- Now, from the name itself, you may already understand what is communication.
- But, here what we mean is communication between processes (a program that is in execution is known as a process).
- Now there are many processes that are present, and this processes often need to communicate with each other so that they can synchronize with each other and their execution can be done in an efficient way.
- So, there are many circumstances in which one process needs to exchange information with another process.

OS Services-Communication

- So, such communication may occur between processes that may be executing on **the same computer or between processes that are executing on different computers tied together by a network.**
- So, the processes that we have may be present between the same computer and they may have to communicate with each other or may be connected by a network and in this network they are connected to different computers and there are different processes present in different computers.
- So, the communication between these processes between the same computers or even between the different computers is controlled by the OS.
- So this is one of the important services that are provided by the OS.

OS Services-Error Detection

- Error can always occur in the courses of computing, and the OS needs to be constantly aware of possible errors that can occur.
- So, there are many kinds of errors that can occur, Like there are errors that can occur in ….
    - The CPU,
    - The memory,
    - The hardware,
    - The Input or output devices, and so on.
- So, there are many kinds of errors.

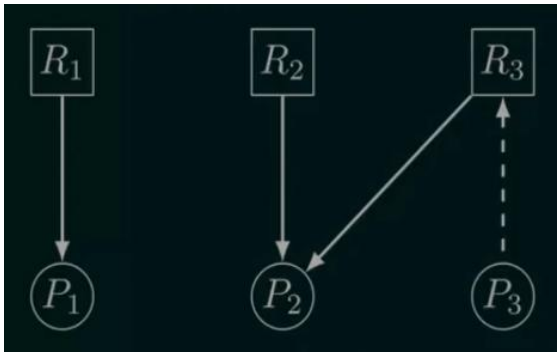


OS Services-Error Detection

- So, let me give you an example of an error…
- Here you have a printer and let's say that you are printing somethings, and suddenly the paper run out.
- Now, the printer is not able to print because there is no more paper available.
- This is also an error. And the OS must be able to handle this kind of error so that it ensures correct and consistent computing.
- Now what we mean by this is that when error occurs your system must not just break down completely, and it should not just seize your computing ability completely.
- The OS must have a way of in which it manages those errors so that your computing is consistent and it is still carried on even if some error encountered.
- So, error detection is very important services that is provided by the OS.

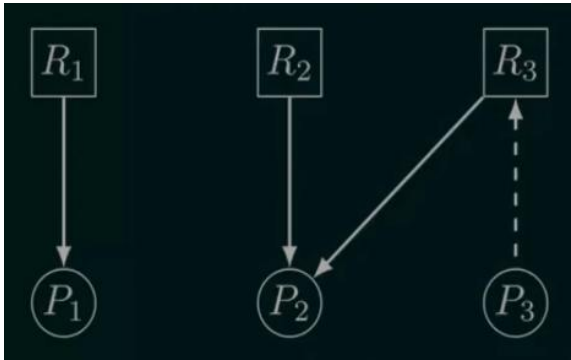
OS Services-Resource Allocation

- Resource allocation means allocating resources to different processes to different users. Now, what do you mean by resource?
- Resources can be of different types…CPU, files, I/O devices, main memory, and so on.
- Thus, there are so many resources that you have.
- And as there are many processes running in our system. And all the processes require certain resources at a certain point of time.
- Here, the OS must help in resource allocation. It means that it should allocate the required resource to the processes which are waiting or asking for the those resources.



OS Services-Resource Allocation

- And it must allocate them in an efficient way such that all the processes get the resources that they need and no process keeps waiting for the resource and never gets it.
- We should not have a scenario where a process keeps waiting for a resource but it never gets it.
- And also we should not have a scenario where a resource is held by particular process .



OS Services-Accounting

- We mean by accounting is that we want to keep track of which user use how much and what kind of resources and why is this required?
- This record keeping may be used for accounting or simply for accumulating usage statistics.
- So keeping an account of this or by having a statistics of this usage, it can be a valuable tool for researchers who wish to reconfigure the system or to improve the computing services.

OS Services-Accounting

- So, if you want to improve your computing services you need to know how the resources are used and how they actually work.
- So, for this purposes accounting is an important services.

OS Services-Protection and Security

- Now the next services provided by an OS is Protection and Security.
- What you mean by protection and security?
- So, our data should be secure, and everything what we do must be protected.
- **Now, we can talk about protection and security in different ways…**
  - When several processes are executing at the same time, it should not be possible for one process to interfere with the others or with the OS.
  - So, the processes should not interfere with each others operation, and protection involves ensuring that all access to system resources are controlled.
  - Thus, protection means that access to the system resources must be controlled

OS Services-Protection and Security

- **Now, we can talk about protection and security in different ways…**
  - We can talk about security in terms of outside access.
  - We should make sure that the system is not accessible to outsiders who are not allowed to access the system.
  - So this is very common in our daily lives when we used our system or when you used your accounts like your email accounts, or your social media accounts.
  - You make your user id and password and by doing that what you are doing is you are protecting your account.
  - You are authenticating yourself by providing the right password.

OS Services

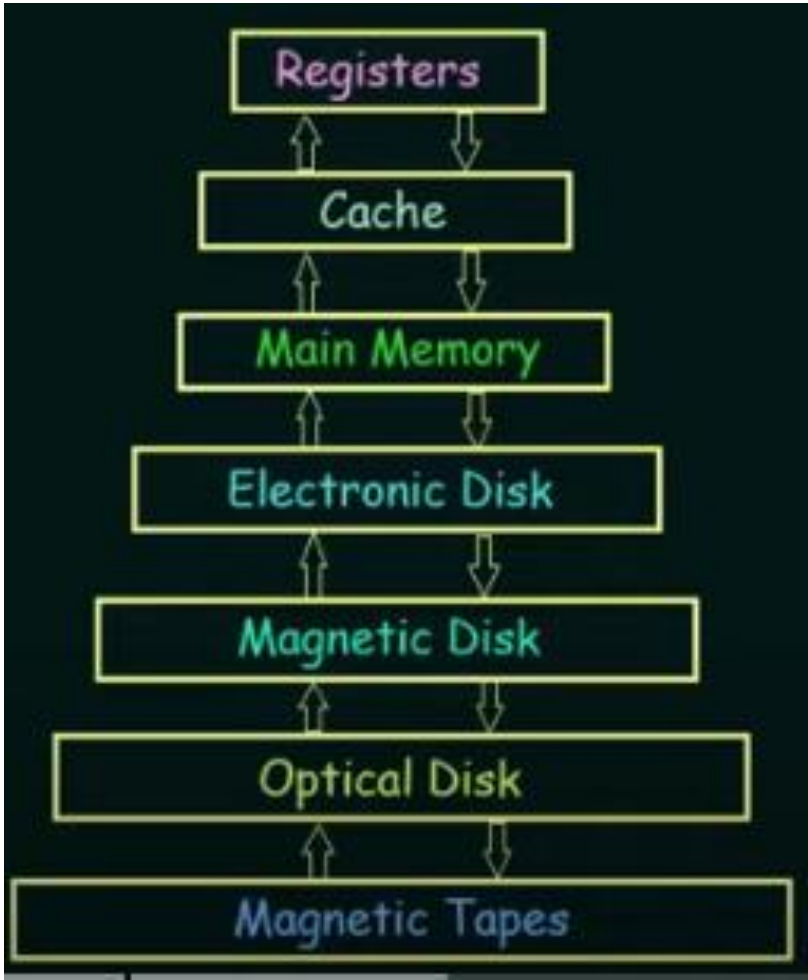
- **The various OS services that we have discussed:**
  - Program Execution
  - I/O Operations
  - File System Manipulation
  - Communications
  - Error Detection
  - Resource Allocation
  - Accounting
  - Protection and Security

Storage Structure and Hierarchy

•There are different storage devices structure that we have in our computer systems.

•We have a diagram given which represents the storage devices hierarchy.

•On the top, we have Registers, followed by Cache, the we have the main memory, and then the electronic disk. Then Magnetic disk, then optical disk and Magnetic tapes.



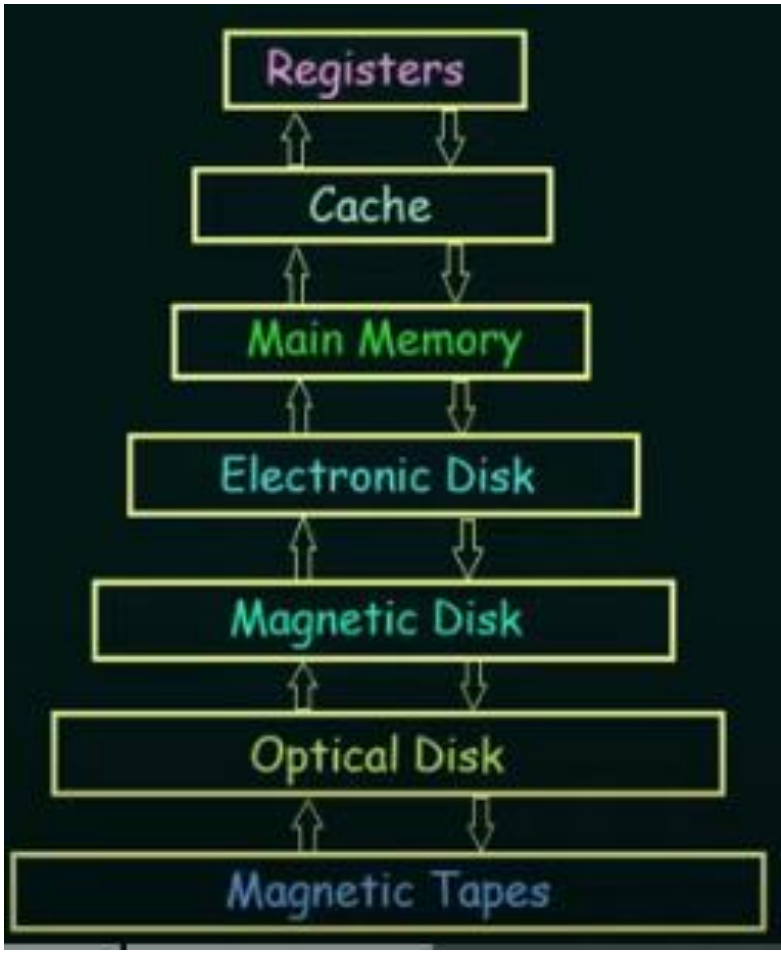
Storage Structure and Hierarchy

•So, if you look at these devices, starting from **Registers**.

• Registers are the smallest storage devices. They stores data in bits, that means in 0's or 1's.

•So they are the smallest devices and since they store only in bits.

•They can be accessed very quickly .



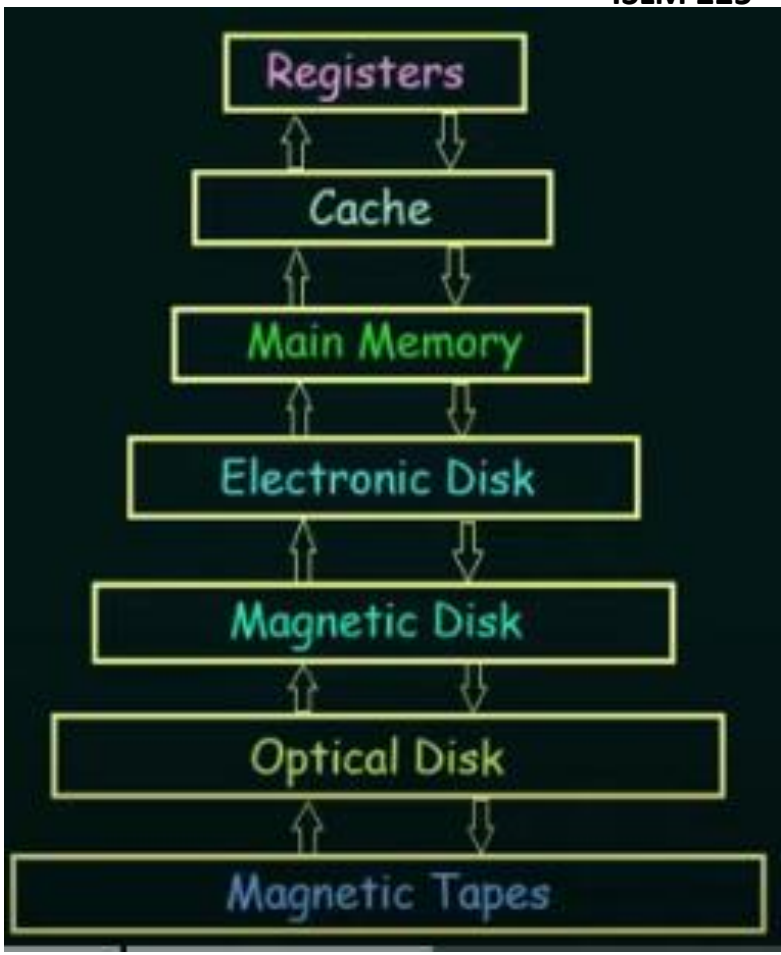
Storage Structure and Hierarchy

•The next one is Cache. The size of the cache is a little bigger than that of the registers.

• But its speed is a little slower as compared to the registers.

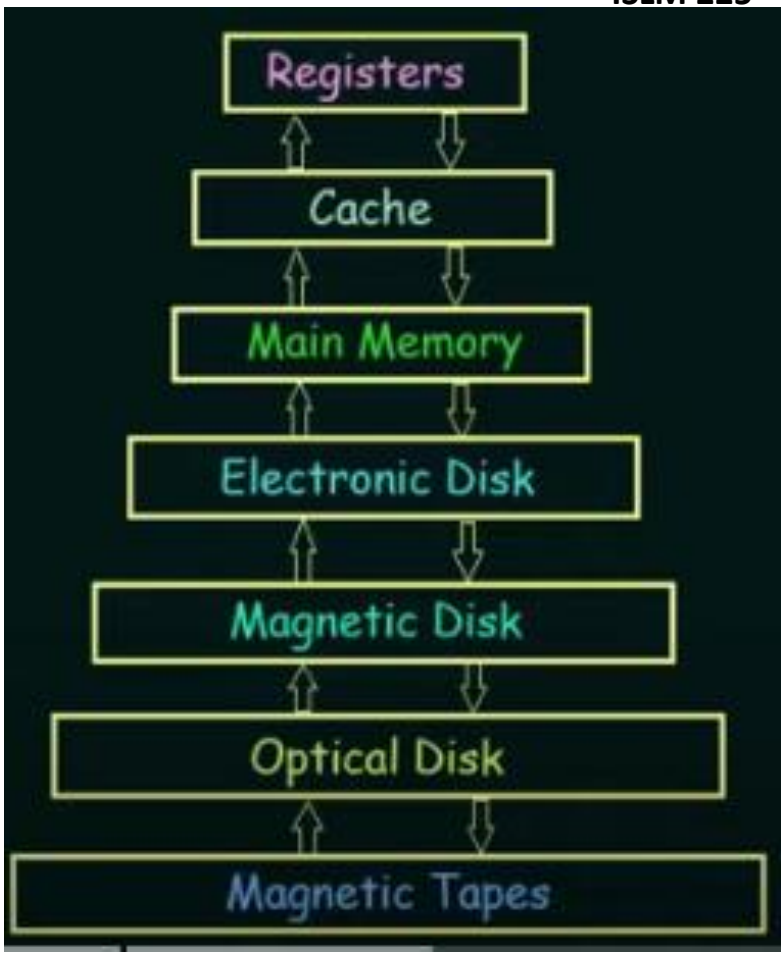
•After that, we have the main memory which is very important things that you need to know.

•Examples of your main memory are RAM.

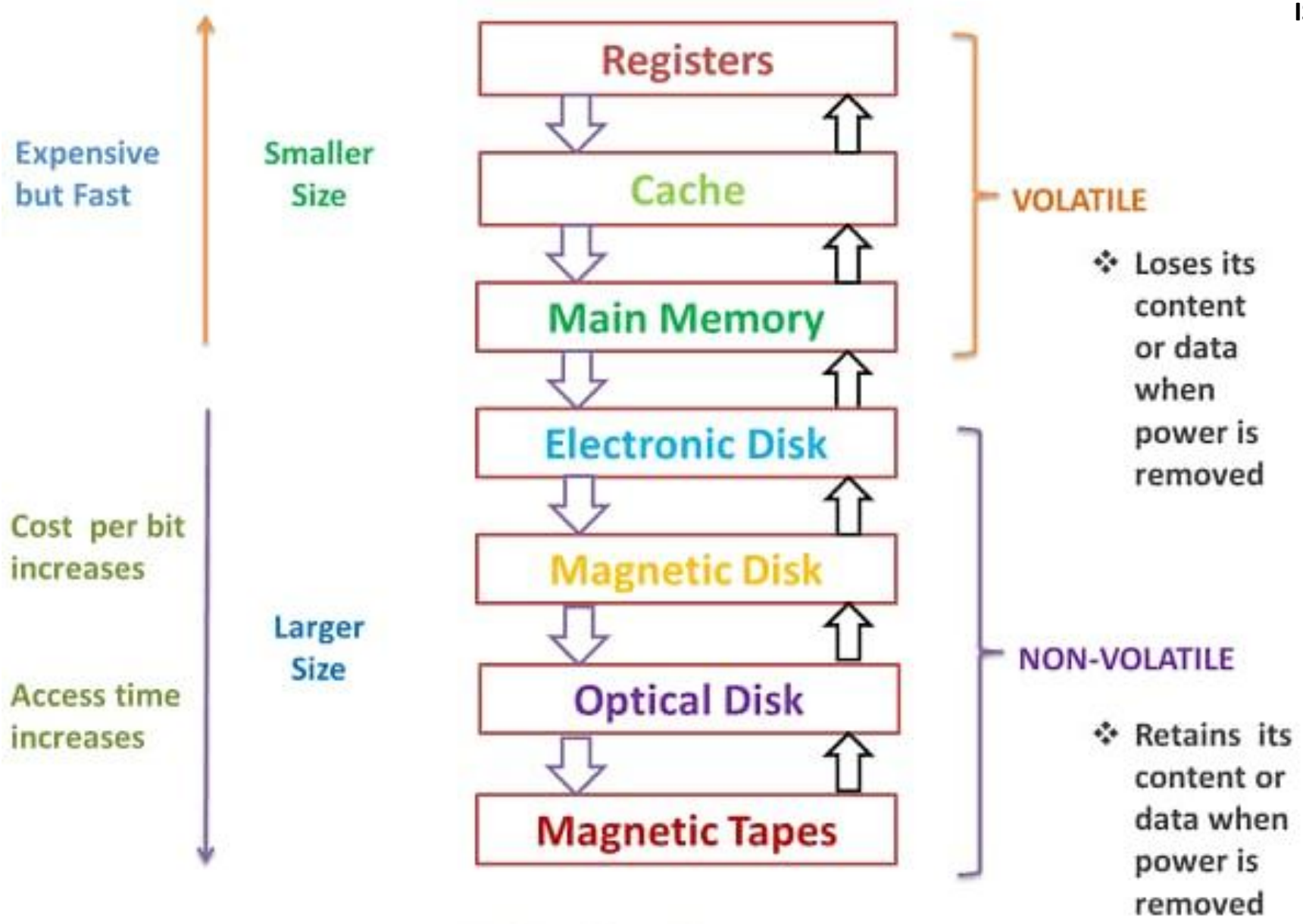


Storage Structure and Hierarchy

- Then we have electronic disk, magnetic disk, optical disk, and magnetic tapes .
- So, they are the secondary storage devices.



Storage Structure and Hierarchy



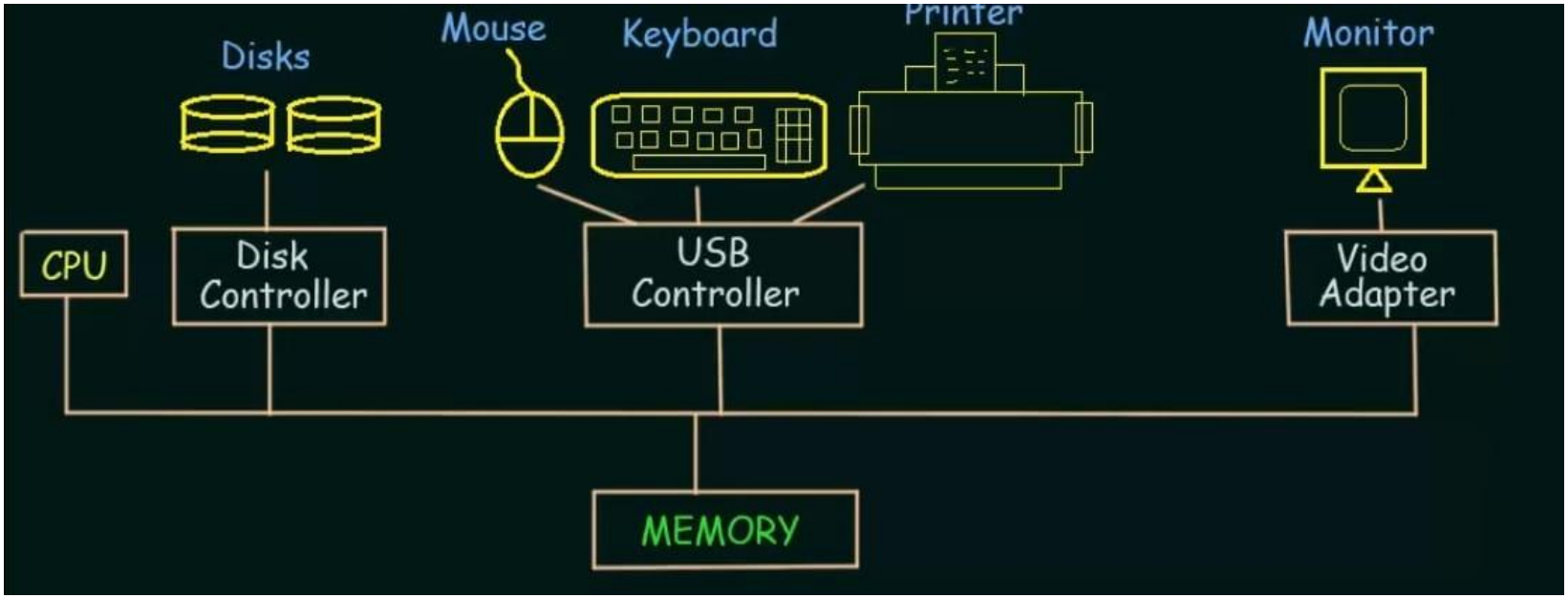
So, for explaining this, let me take a simple examples

I/O Structure

- So far, we have studied the storage structure under the basics of the OS.
- Storage is only one of many types of I/O devices within a PC.
- A large portion of OS code is dedicated to managing I/O, both because of its importance of the reliability and performance of a system and because of the varying nature of the I/O devices.

I/O Structure

•A general purpose PC consists of CPUs, and multiple device controllers that are connected through a common bus that provides access to shared memory.



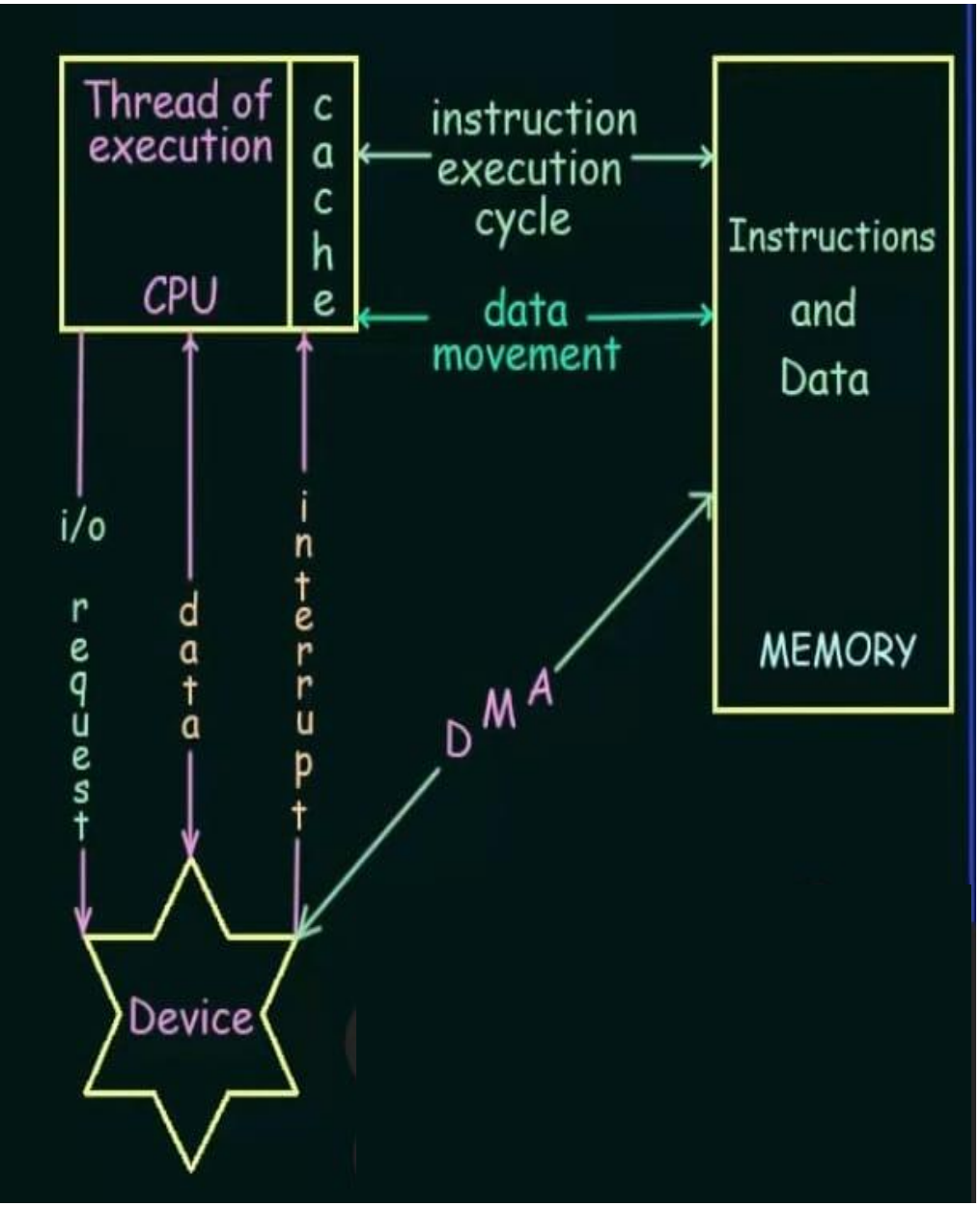
•Each device controller is in charge of a specific type of device.

•Each device controller maintains Local Buffer storage and Set of special purpose Registers.

I/O Structure

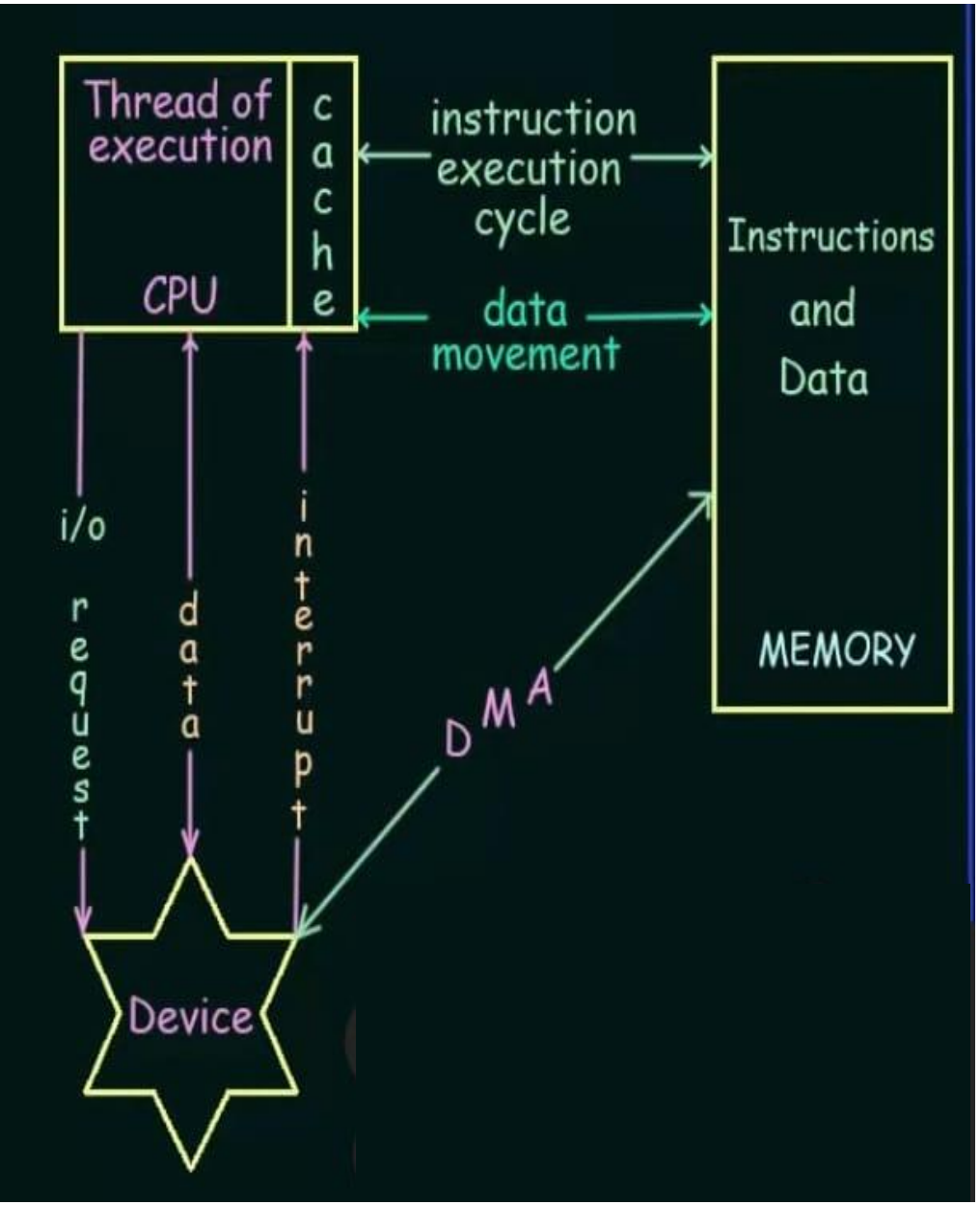
- Typically, OS have a device driver for each device controller.
- This device driver understands the device controller and presents a uniform interface to the device to the rest of the OS.
- **Lets see the working principles of an I/O operation.**

Working principle of I/O Operation



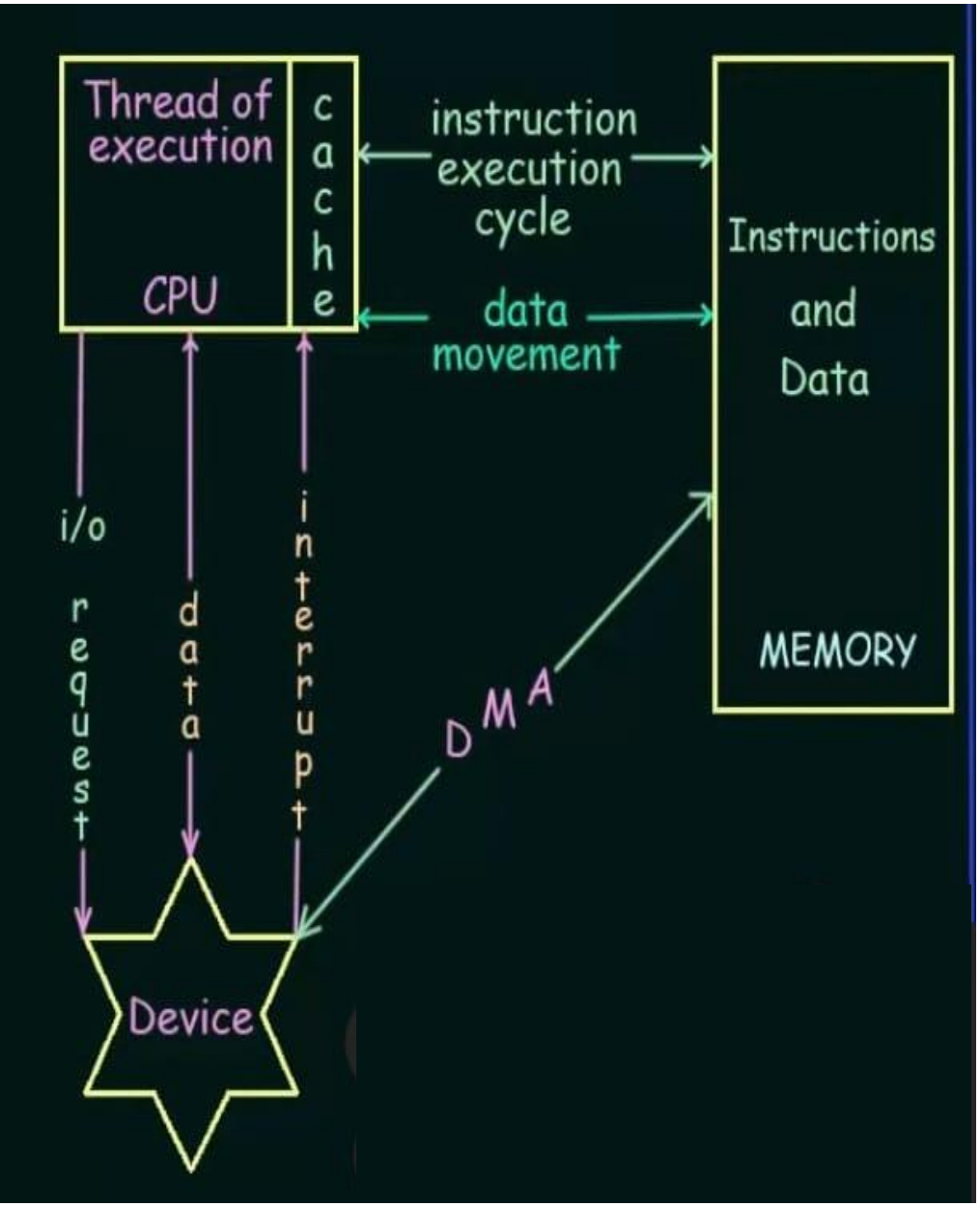
- To start an I/O operation, the devices driver loads the appropriate registers within the device controllers
- **The device controller, in turn, examines the contents of these registers to determine what action to take.**

Working principle of I/O Operation



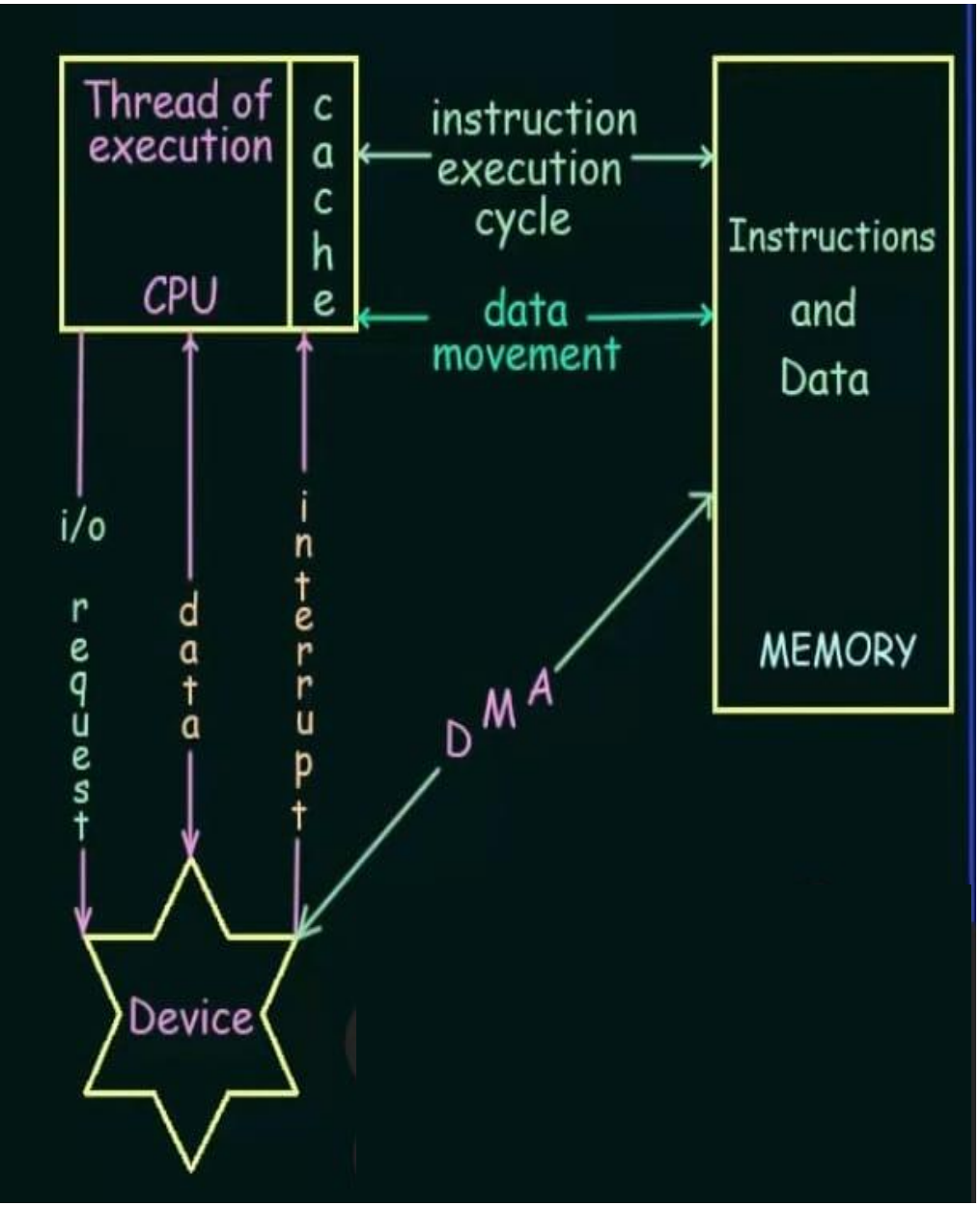
- The controller starts the transfer of data from the device to its local buffer.
- Once the transfer of data is complete, the device controller informs the device driver via an interrupt that it has finished its operation.
- The device driver then returns control to the operating system.

Working principle of I/O Operation



- This form of interrupt-driven I/O is fine for moving small amount s of data but can produce high overhead when used for bulk data movement.
- To solve this problem, Direct Memory Access (DMA) is used.

Working principle of I/O Operation



- After setting up buffers, pointers and counters for the I/O device, the device controller transfers an entire block of data directly to or from its own buffer storage to memory with no intervention by the CPU.

System Calls

- System calls provide an interface between a running program and OS.
- System calls are generally available as assembly language instructions.
- Several higher level languages such as C also allow to make system calls directly.
- There are **THREE** general methods that are used to pass information (parameters) between a running program and the OS.
  - *One method is to store information (parameters)in registers.*
  - *Another is to store information (parameters) in a table in memory and pass the address of table to the registers.*
  - *The third method is to push information (parameters) on stack and allow OS to pop the information (parameters) off the stack.*

Types of System Calls

- System calls can be grouped roughly into FIVE major categories:
  - Process Control
  - File Manipulation
  - Devices Management
  - Information Maintenance
  - Communications
- **Now lets see what are the meaning of these five categories one by one….**

Types of System Calls

•Process Control

- These types of system calls are used to control the processes.
- We know that there are various processes running in our system and all these processes are need to be controlled.
- Some examples are, when a process is running, it has to be run and it has to be terminate or halt either normally or abnormally.
- So by normally what is meant that the process should come to halt when it completes its execution.
- Otherwise, there might also be errors occurring in the process.
- So, if an error occurs in the execution of a process then that process has to be halted or aborted.
- So, what is that we mean by halting abnormally.

Types of System Calls

•Process Control (Cont…)

- So, the system calls sets are used for controlling the processes.
- Some example of Process control system calls sets……
  - end, abort,
  - load, execute,
  - create process, terminate process,
  - Get process attributes, set process attributes,
  - Wait for time
  - Wait event, signal event
  - Allocate and free memory etc.

Types of System Calls

•Process Control (Control…)

- Some example of Process control system calls……
  - **end, abort**:
    - *So, by 'end' we mean that to end the process in a normal way when it completes its execution.*
    - *And, by 'Abort' we mean to halt the process when it encounters an error.*
  - **load, execute:**
    - *So, by 'Load' we mean to load a process and by 'Execute' we mean to execute a process.*
  - **Create process, terminate process:**
    - *For creating a process or for terminating a process also, we need system calls.*
  - **Get process attributes, set process attributes:**
    - *For getting and setting the attributes of our processes .*

Types of System Calls

- **Process Control**
  - **Wait for Time:**
    - *This means that some of the processes need to wait for their time to come so that they can begin their executions . Thus, a process may need to wait for its time of execution.*
    - *For that also we need a system call.*
  - **Wait event, signal event:**
    - *So a process may need to wait for an event or need to signal an event.*
  - **Allocate and free memory :**
    - *When a process is going to be executed it may need access to the memory.*
    - *So, we may have to allocate the memory for a process.*
- **So, these are all examples of system calls that we use for controlling the process.**
- **They all fall under the categories of process control system calls.**

Types of System Calls

•File Manipulation

•These are the system calls that are used for manipulations, or managing your files.

•Some examples of file manipulation system calls……

- •Create files, delete files,
- •Open, close
- •Read, write , reposition,
- •Get file attribute, set file attributes.

Types of System Calls

•Device Manipulation

•The third one is device manipulation.

•Device manipulation or device management are the system calls that are used for managing or manipulating devices.

•Device means the I/O devices and other devices in your system.

•Some examples of device manipulation system calls……

- •Request devices, release device
- •Read, write, reposition
- •Get device attributes, set device attributes
- •Logically attach or detach devices (??????????????).

Types of System Calls

•Information Maintenance

•The fourth one is information maintenance.

•So, by information maintenance what we mean is that all the information that we have about our system must be maintained and updated.

•Some examples of Information Maintenance system calls……

- •Get time or date , set time or date
- •Get system data, set system data
- •Get process file, or device attributes
- •Set process, file, or device attributes

Types of System Calls

•Communication

•The sixth one is Communication system calls.

•So, communication system calls are those calls are those system calls which are used for communication between processes between different devices.

• Some examples of communication system calls……

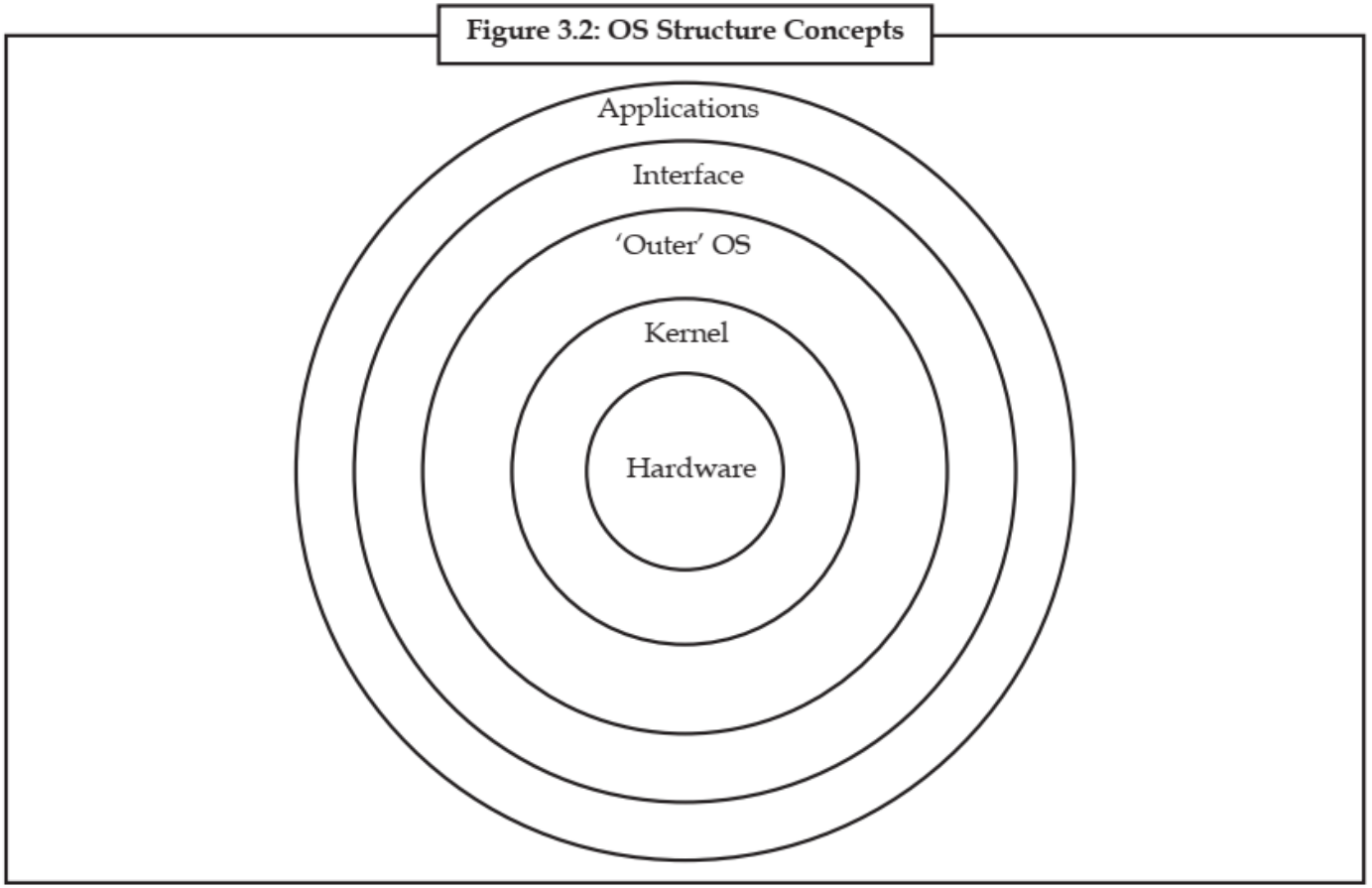
- •Create, delete communication connection
- •Send, receive messages
- •Transfer message information
- •Attach or detach remote device.

Types of System Calls

- System calls can be grouped roughly into FIVE major categories:
  - Process Control
  - File Manipulation
  - Devices Management
  - Information Maintenance
  - Communications

OS Structure

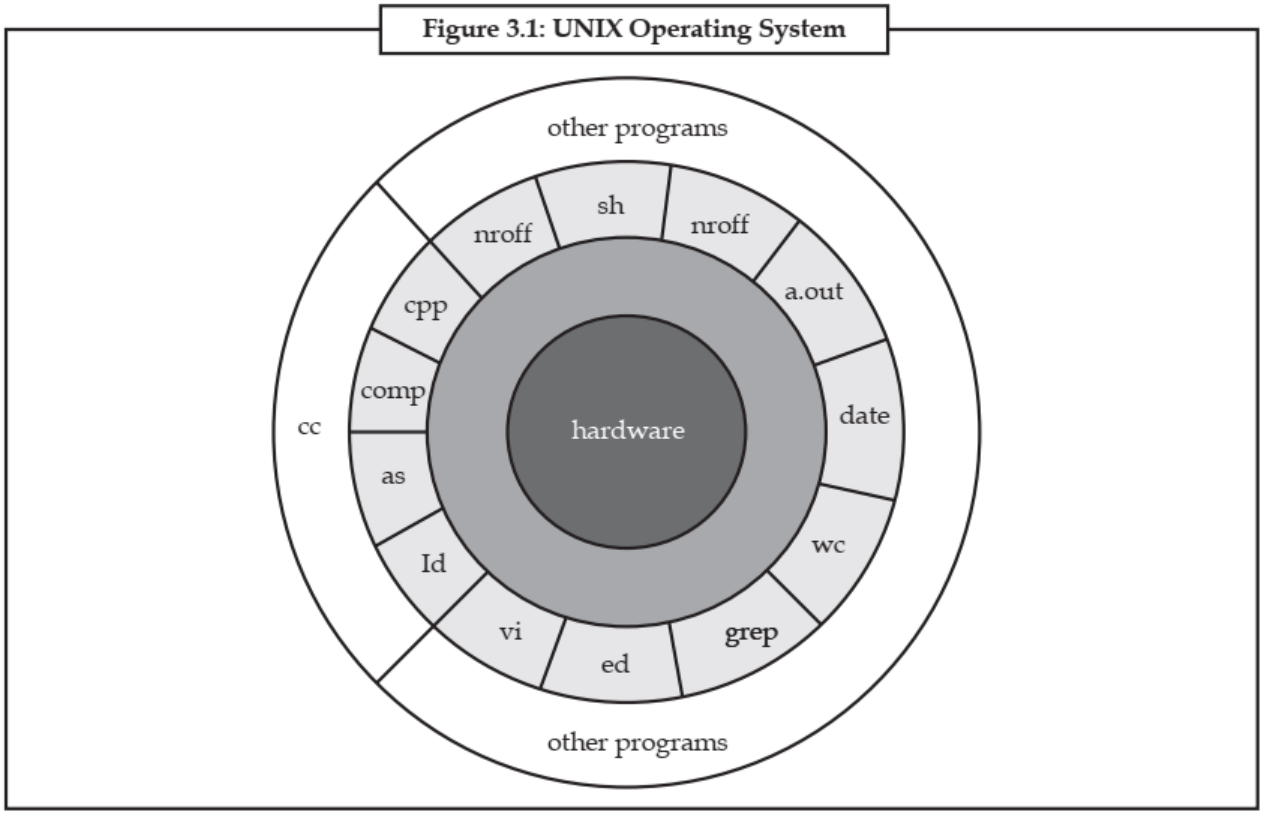
•It's very common to find pictures like Figure below that describe the basic structure of an operating system.



- You might find that some versions of this have different numbers of rings. What does each part represent?

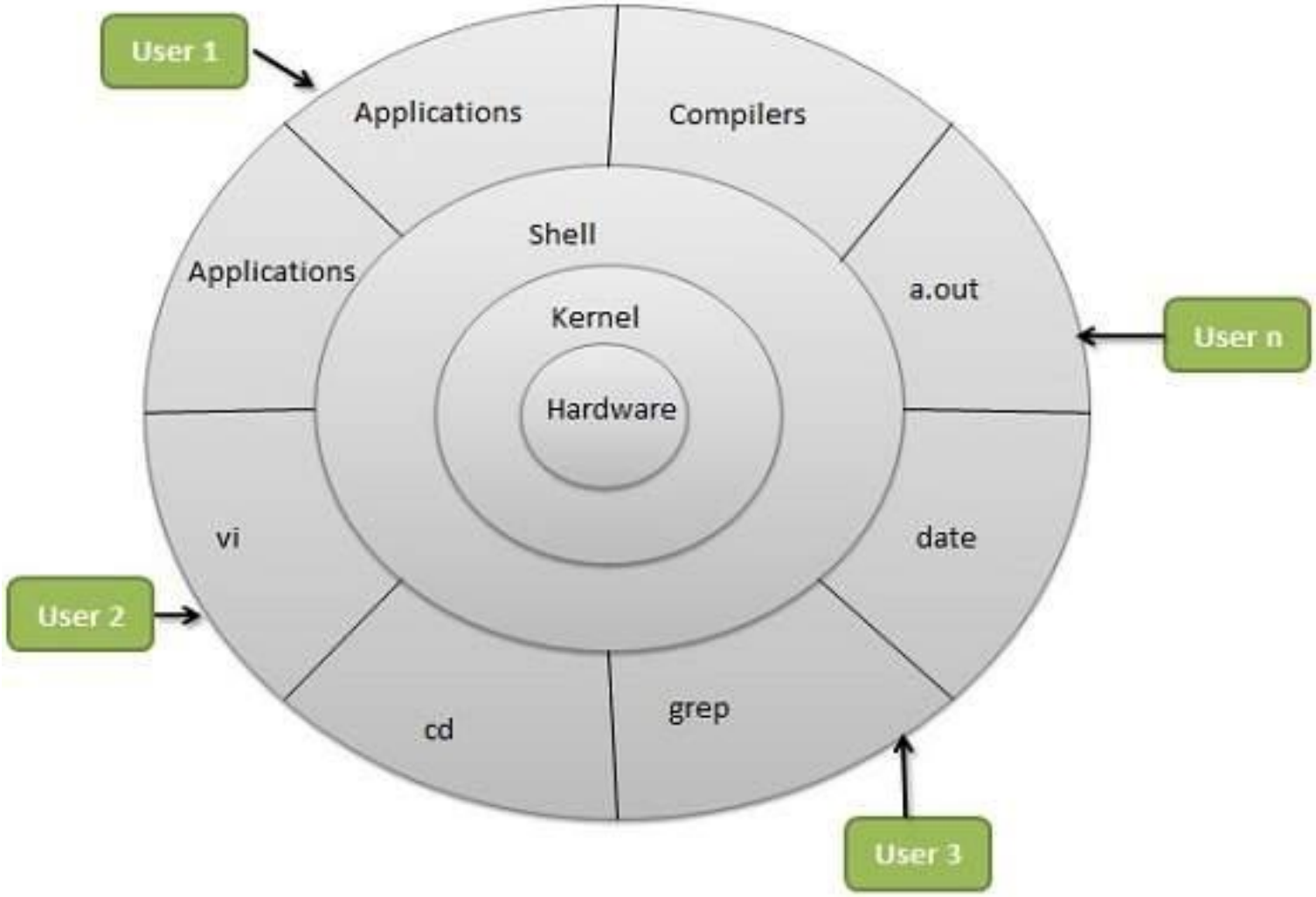
OS Structure

- The types and functions provided by the operating system substructures are meant to present a model for handling these resources that is largely independent of the operating system.
- There are different types of structure as described in Figure 3.1 for **UNIX OS**.



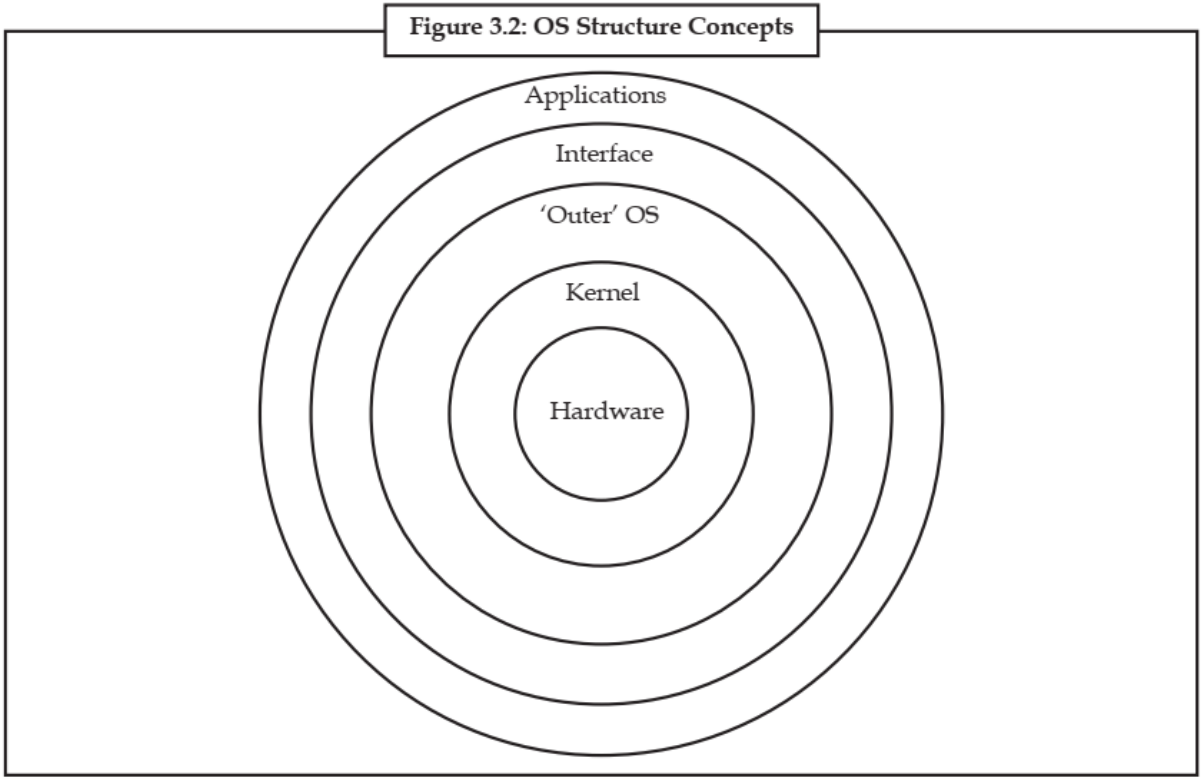
OS Structure

•There are different types of structure as described in Figure 3.1 for **Linux OS**.



OS Structure

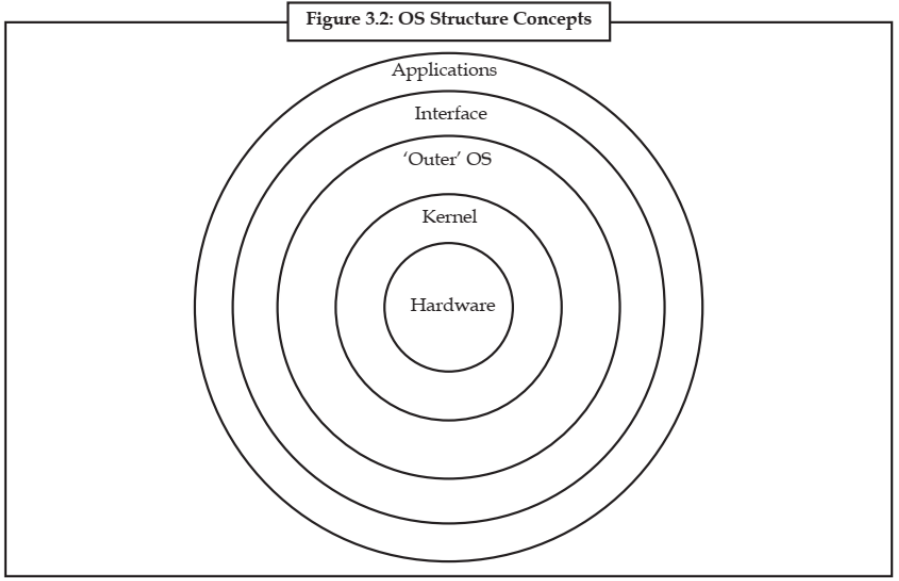
- It's very common to find pictures like Figure 3.2 below that describe the basic structure of an operating system.



- ***Hardware:*** The hardware is, obviously, the physical hardware and not particularly interesting to us in this module.

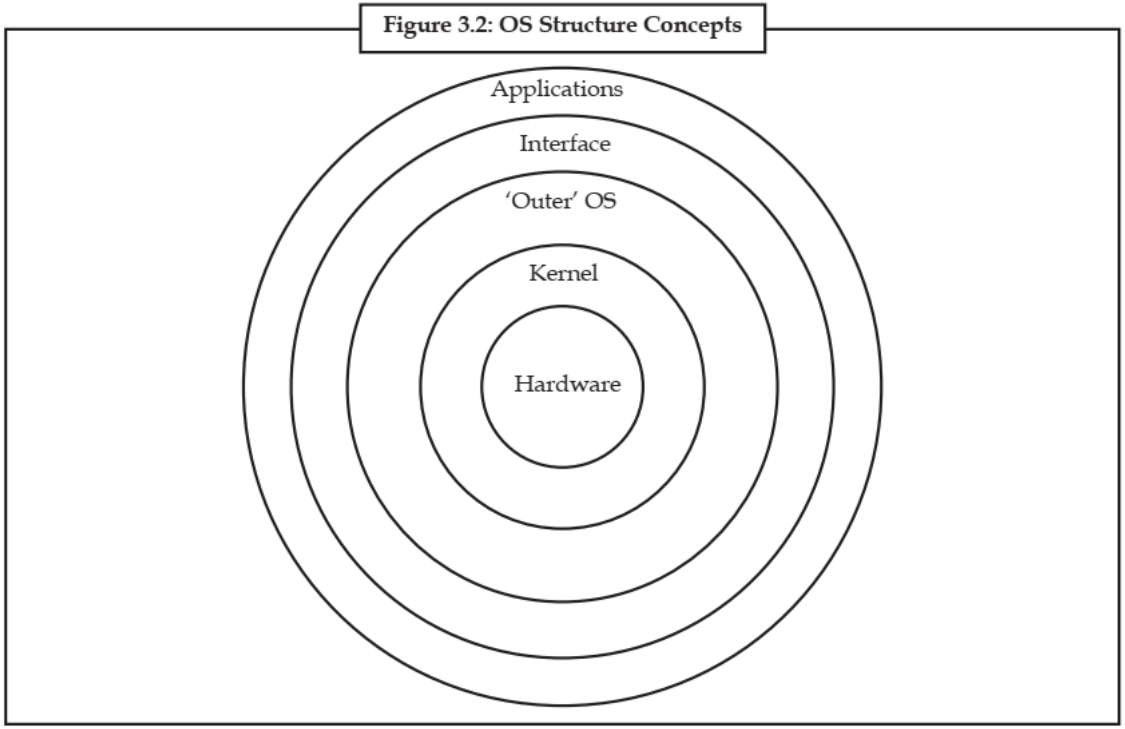
OS Structure

- **Kernel** is the heart of Linux OS.
- It manages resource of Linux OS.
- Resources means facilities available in OS. For e.g.
  - Facility to store data,
  - print data on printer,
  - memory,
  - File management etc.



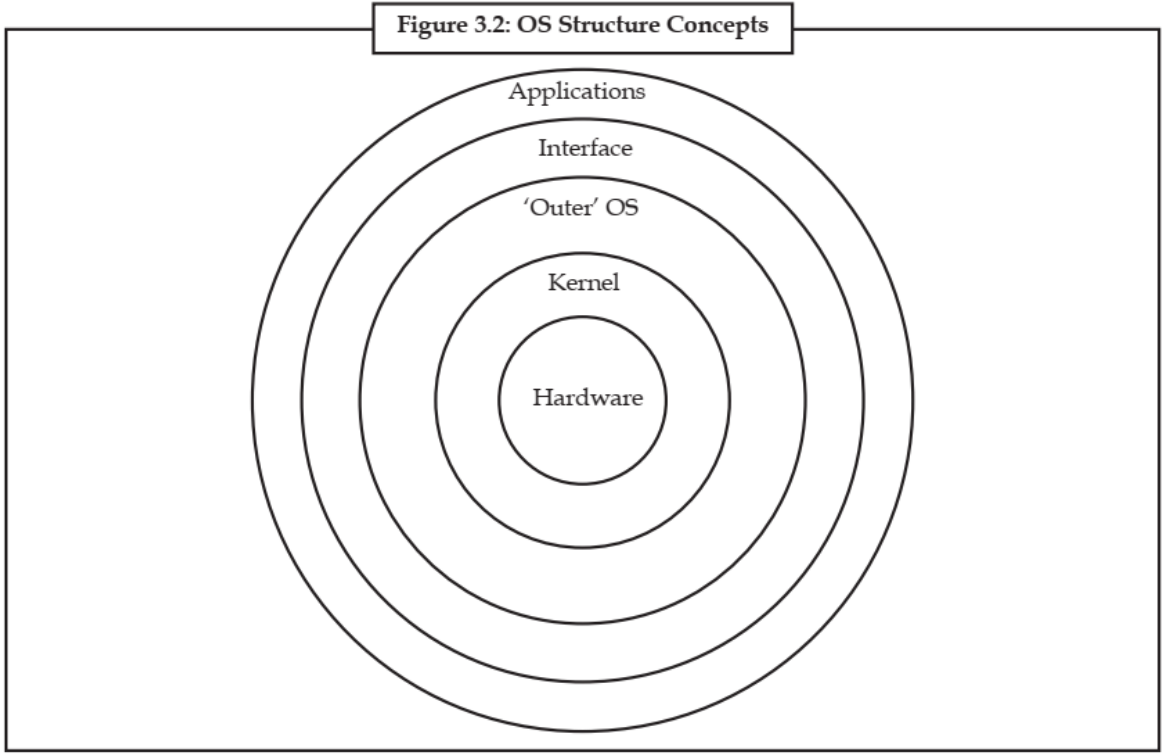
- Kernel decides who will use this resource, for how long and when.
- It runs your programs (or set up to execute binary files),
- The kernel acts as an intermediary between the computer hardware and various programs / application / shell.

OS Structure



- ***Outer OS:*** Surrounding the kernel are other parts of the operating system. These perform less critical functions - for example, the graphics system which is ultimately responsible for what you see on the screen.

OS Structure



- ***Interface:*** The interface provides a mechanism for you to interact with the computer.
- ***Applications:*** There are what do the actual work - they can be complex (for example Office) or simple (for example the is command commonly found on unix and Linux systems that lists files in a directory (or folder)

OS Structure- Monolithic Structure

- The monolithic operating system is a very basic operating system in which file management, memory management, device management, and process management are directly controlled within the kernel.
- The kernel can access all the resources present in the system.
- In monolithic systems, each component of the operating system is contained within the kernel.
- Operating systems that use monolithic architecture were first time used in the 1970s.

OS Structure- Monolithic Structure

- The monolithic operating system is also known as the monolithic kernel.
- This is an old operating system used to perform small tasks like time-sharing tasks in banks. The monolithic kernel acts as a virtual machine that controls all hardware parts.

OS Structure- Microkernel Structure

- ***Microkernel*** *OS* has limited tasks.
- A microkernel is divided into two parts, *kernel space*, and *user space*.
- Both parts communicate with each other through IPC (Inter-process communication).
- Microkernel's advantage is that if one server fails, then the other server takes control of it.
- Microkernel is slower but more secure and reliable than monolithic kernel.