Operating Systems

By
Meghana G Raj
Assistant Professor
IT Department

Introduction

- OS Functionalities
- Types of Operating System
- User OS Interface
- System Calls
- System Boot

What happens when you switch on a computer?(1/6)

- A computer without a program running is just an inert hunk of electronics.
- The first thing a computer has to do when it is turned on is start up a special program called an *operating system*.
- The operating system's job is to help other computer programs to work by controlling the computer's hardware (CPU, Input, Output devices, Memory).

What happens when you switch on a computer?(2/6)

- The process of bringing up the operating system is called *booting.*
- Computer knows how to boot because instructions for booting are built into one of its chips, the BIOS (or Basic Input/Output System) chip.
- The BIOS chip tells it to look in a fixed place, usually on the hard disk (the *boot disk*) for a special program called a *boot loader* (under Linux the boot loader is called Grub or LILO).
- The boot loader is pulled into memory and started. ***The boot loader's job is to start the real operating system*.**

What happens when you switch on a computer?(3/6)

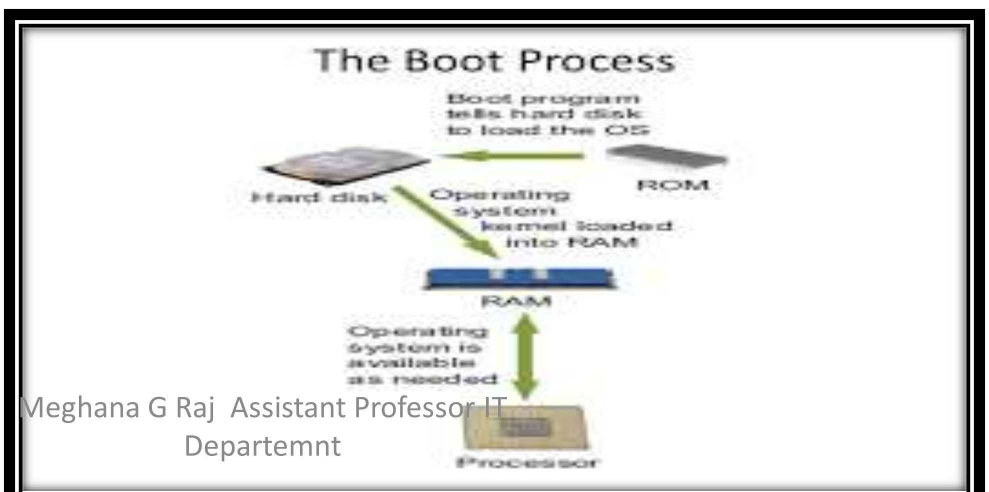
- The loader does this by looking for a *kernel*, loading it into memory, and starting it.
- In Linux you see "LILO" on the screen followed by a bunch of dots, it is loading the kernel. (Each dot means it has loaded another *disk block* of kernel code.
- Once the kernel starts, it has to look around, find the rest of the hardware, and get ready to run programs
- Kernel, has a lot of built-in knowledge about what it's likely to find where, and how controllers will respond if they're present. This process is called *autoprobing*.

What happens when you switch on a computer?(4/6)

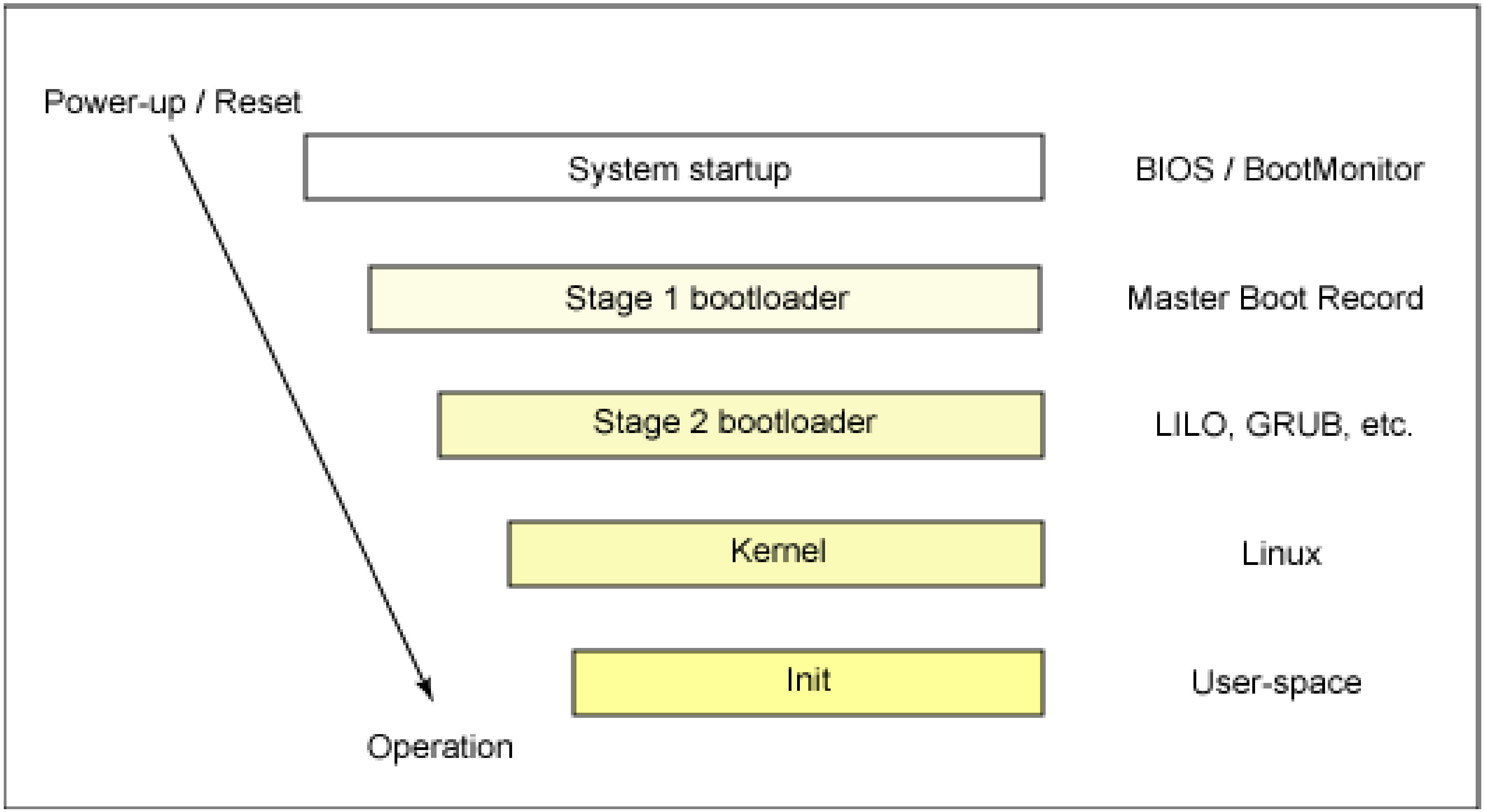
- But getting the kernel fully loaded and running isn't the end of the boot process; it's just the first stage (sometimes called *run level 1*).
- After this first stage, the kernel hands control to a special process called ‘init’ (Some recent Linuxes use a different program called ‘upstart’ that does similar things)
- The init process's first job is usually to check to make sure your disks are OK.
- Disk file systems are fragile things; if they've been damaged by a hardware failure or a sudden power outage, there are good reasons to take recovery steps before your Unix is all the way up.

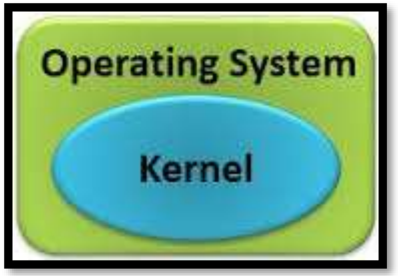
What happens when you switch on a computer?(5/6)

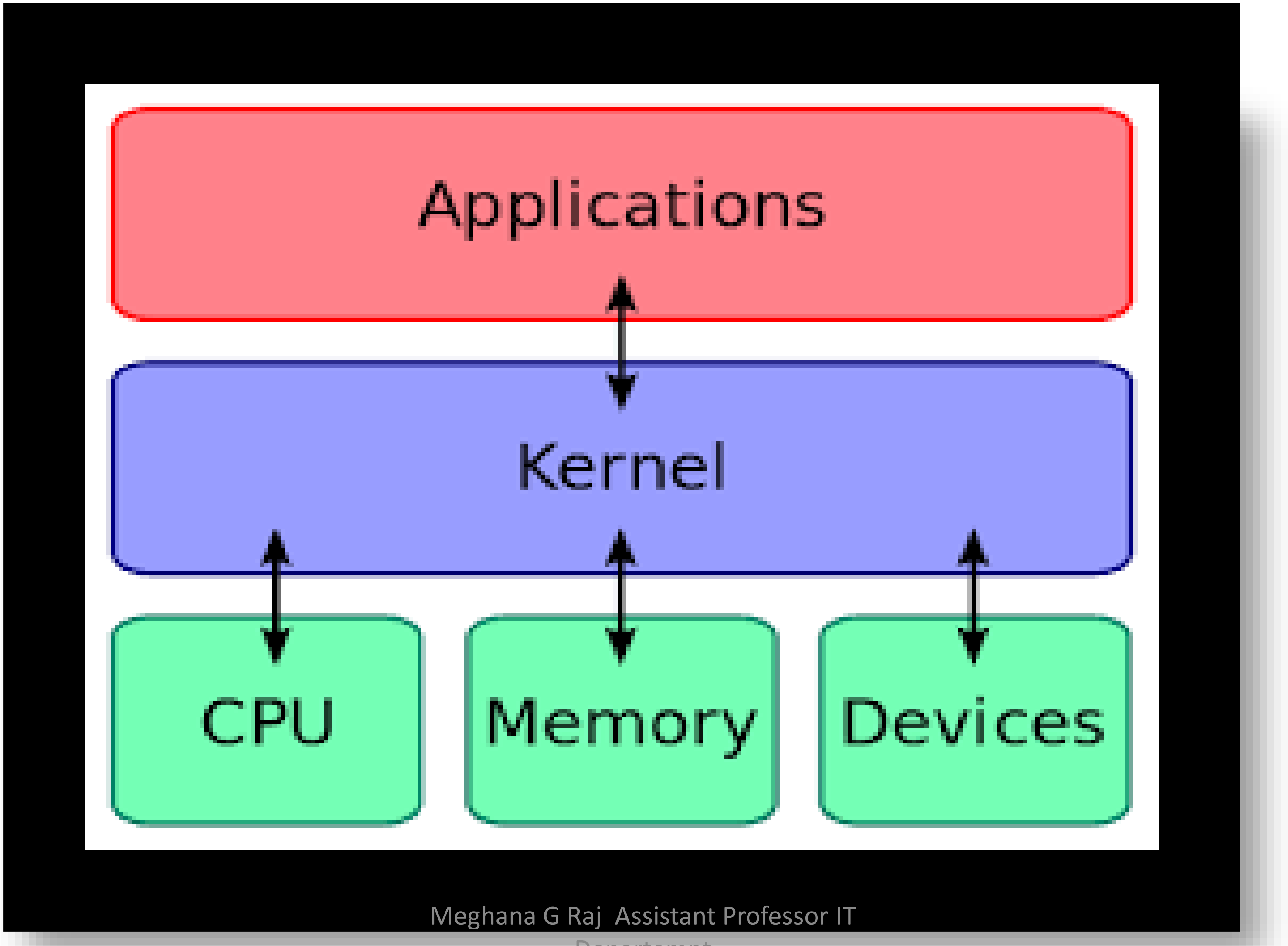
- Init's next step is to start several *daemons*.
- A daemon is a program like a mail listener or a WWW server that runs in the background, waiting for things to do.
- The next step is to prepare for users. Init starts a copy of a program called **getty** to watch your screen and keyboard.



What happens when you switch on a computer?(6/6)

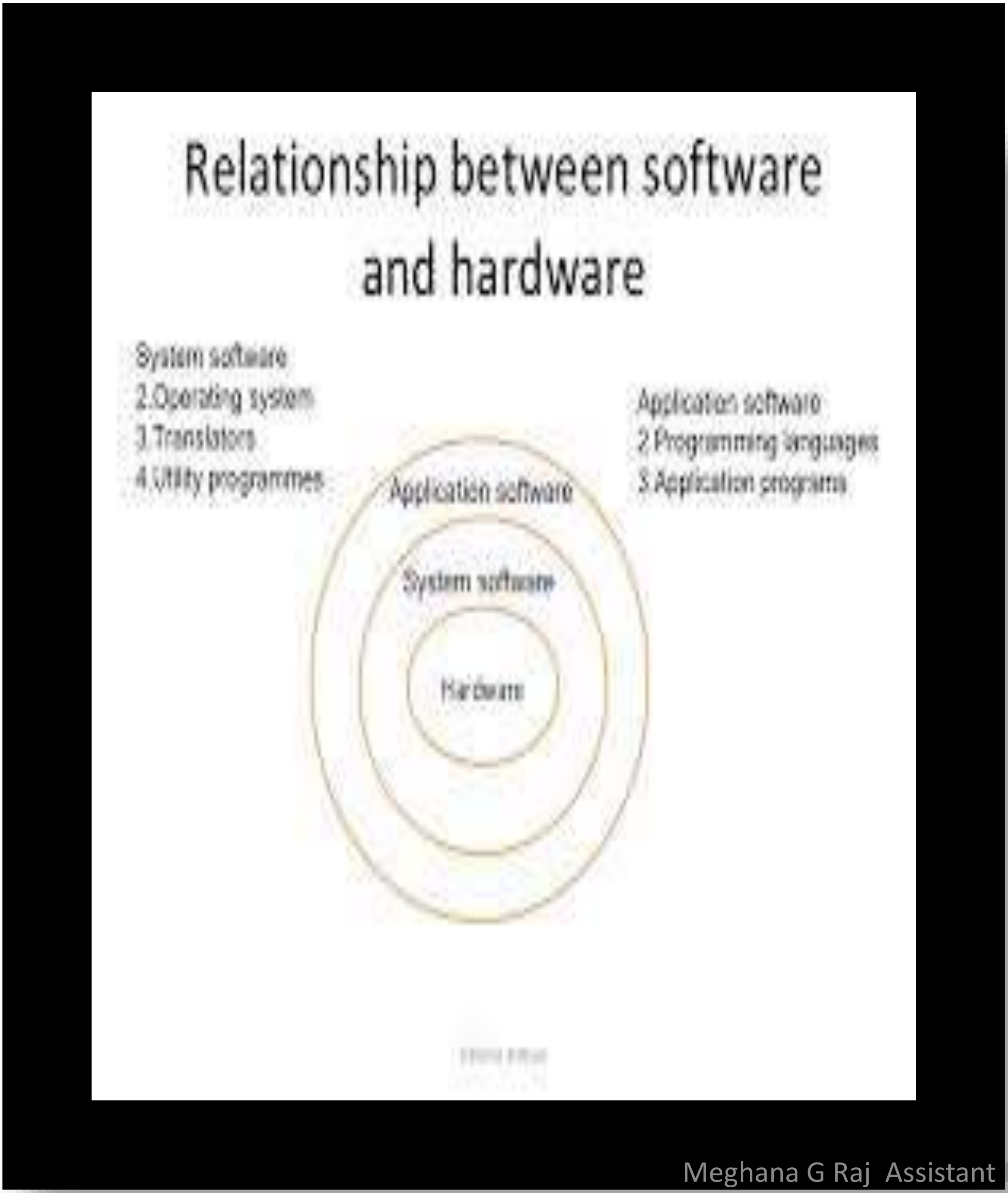



Operating System
Kernel


Applications
Kernel
CPU
Memory
Devices

Application Program and System Programs

- An **application program** (**app** or **application** for short) is a computer program, designed to perform tasks, or activities for the benefit of the user.
- **System programming** (or **systems programming**) is the activity of programming computer system software.
- The primary distinguishing characteristic of systems programming when compared to application programming is that application programming aims to produce software which provides services to the user directly (e.g. word processor), whereas systems programming aims to produce software and software platforms which provide services to other software (e.g. operating systems, game engines)



Relationship between software and hardware
System software
2.Operating system
3.Translators
4.Utility programmes
Application software
2.Programming languages
3.Application programs
Application software
System software
Hardware

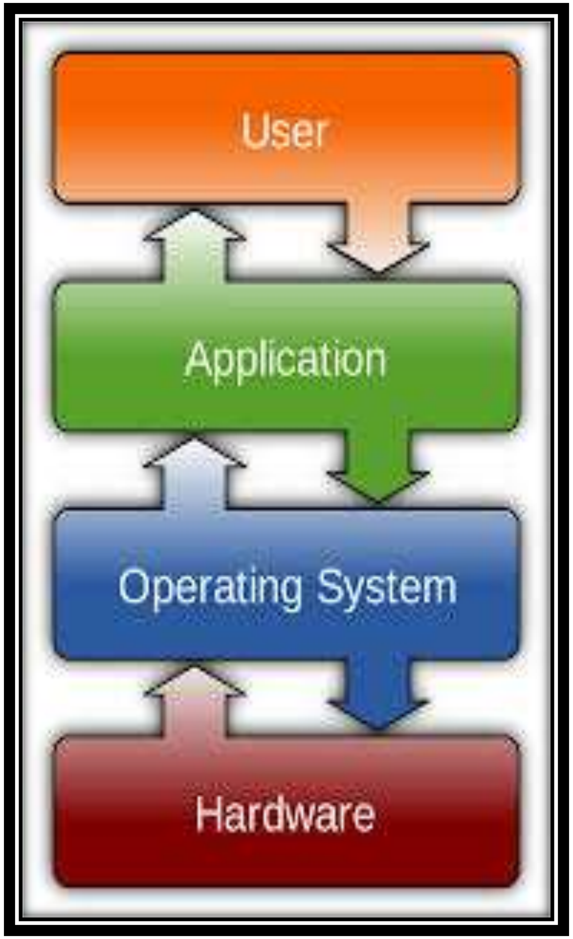
Application Programs: -
MS Office, Banking System
System Program: -
Editors, Compiler and Command Interpreter
Operating System
Physical Devices/Computer hardware

Objective

- To describe the basic organization of computer systems
- To walk through the major components of operating systems

What is an Operating System?

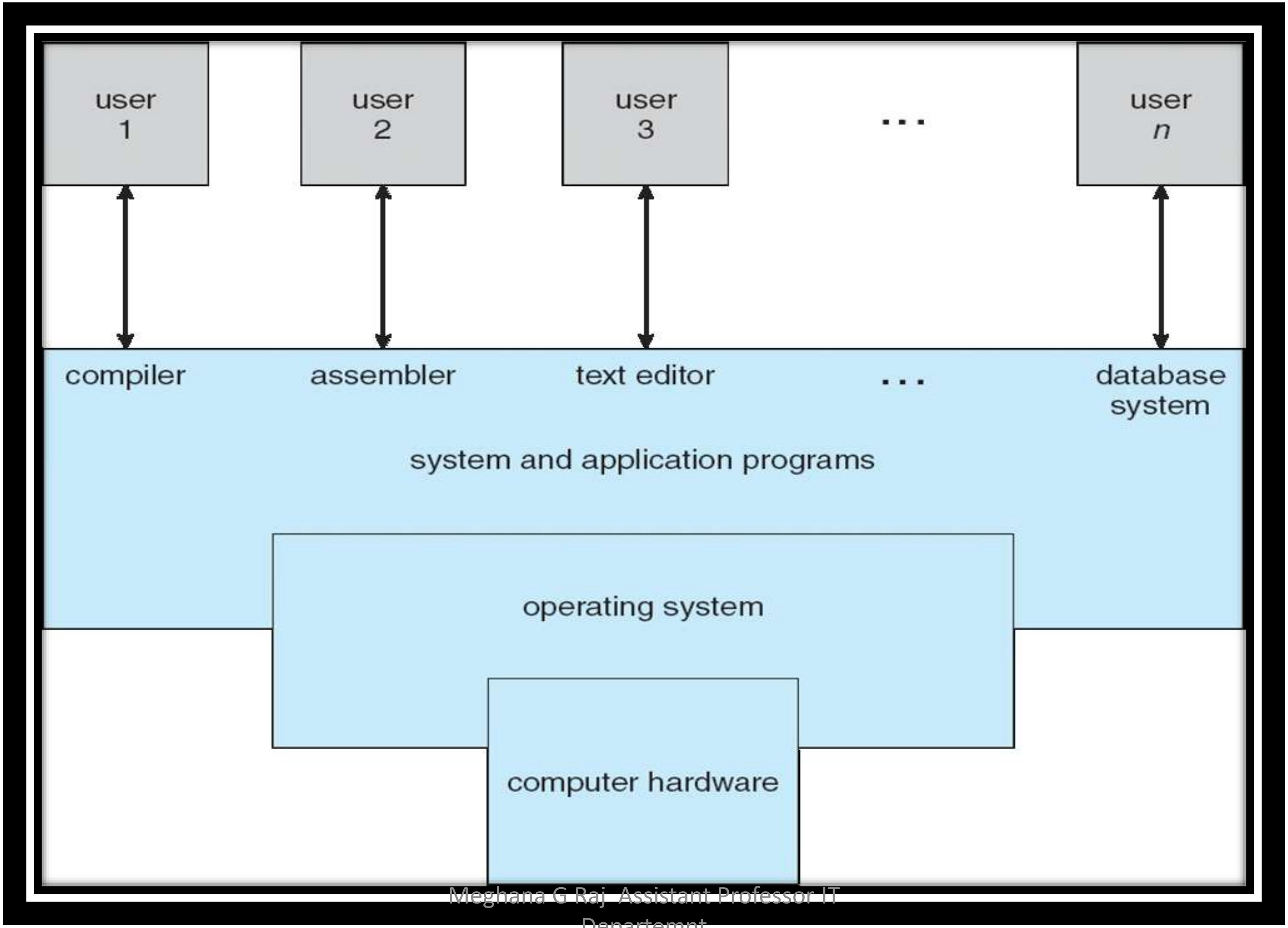
- A program that acts as an intermediary between a **user of a computer** and the **computer hardware.**
- *Operating system goals*:
  - Execute user programs and make solving user problems easier.
  - Make the computer system convenient to use.
  - Use the computer hardware in an efficient manner.



Computer System Structure

- Computer system can be divided into four components:
  - ❑ **Hardware**– provides basic computing resources
    - CPU, memory, I/O devices
  - ❑ **Operating system**
    - Controls and coordinates use of hardware among various applications and users
  - ❑ **Application programs** – define the ways in which the system resources are used to solve the computing problems of the users
    - Word processors, compilers, web browsers, database systems, video games
  - ❑ **Users**
    - People, machines, other computers

Four Components of a Computer System



Operating Systems Design

- An amazing aspect of "OS" is how it varies in accomplishing the task.
- Some OS designed to be

1. Convenient
2. Efficient

Operating Systems Design(1/4)

- **<u>Main frame and Mini Computers</u>**

  OS designed primarily to optimize the utilization of computer hardware.

User View

- Many users use the same computer through other terminal.

1. Sharing of resource and
2. Exchange of information

   takes place.

- **<u>OS designed</u>**

  To maximize **"Resource Utilization"**

  CPU, Memory, I/O, used by all users efficiently and that no individual user takes more than her fair share.

Operating Systems Design(2/4)

- **<u>Personal Computer</u>**

  OS designed to support complex games, business applications.

User View

- View differs on the interface used.
- Personal Computer designed for individual user to utilize the resource.

❑ **Goal**:

- Is to maximize the work that the user is performing.

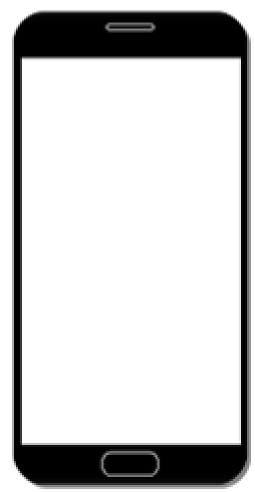
❑ **OS designed**

1. It is easy to use the hardware of the system.
2. Some attention paid towards performance.
3. None to resource utilization.
4. Example: Windows, Linux

Operating Systems Design(3/4)

❑ **<u>Mobile Computers</u>**

OS designed to provide an environment in which a user can easily interface with the computer to execute programs.



User View

- These are stand alone units for individual users
- Connected through wireless technologies.

❑ **<u>User Interface</u>**:

- User interacts with the system through "Touch Screen"

❑ **<u>OS designed</u>**

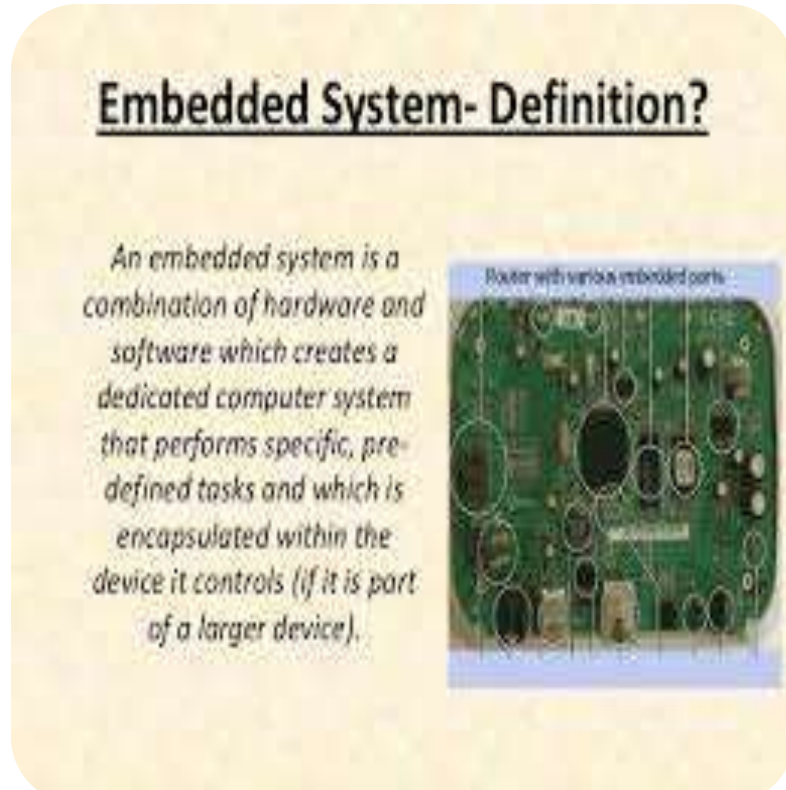
1. Easy to Interact with the system.
2. Some attention paid towards the performance
3. Performance per unit of battery life.
4. None towards the resource utilization.
5. Example: Android, IOS

Operating Systems Design(4/4)

- **<u>Embedded Computers</u>**
- An **embedded system** is a computer system with a dedicated function within a larger mechanical or electrical system.
- It has little or no user view.(microwave, washing machine), Numeric keypads may turn the indicator lights on or off to show the status.

❑ **<u>OS designed</u>**

- Primarily to run without user intervention.



System View(1/2)

- **Resource Allocator**

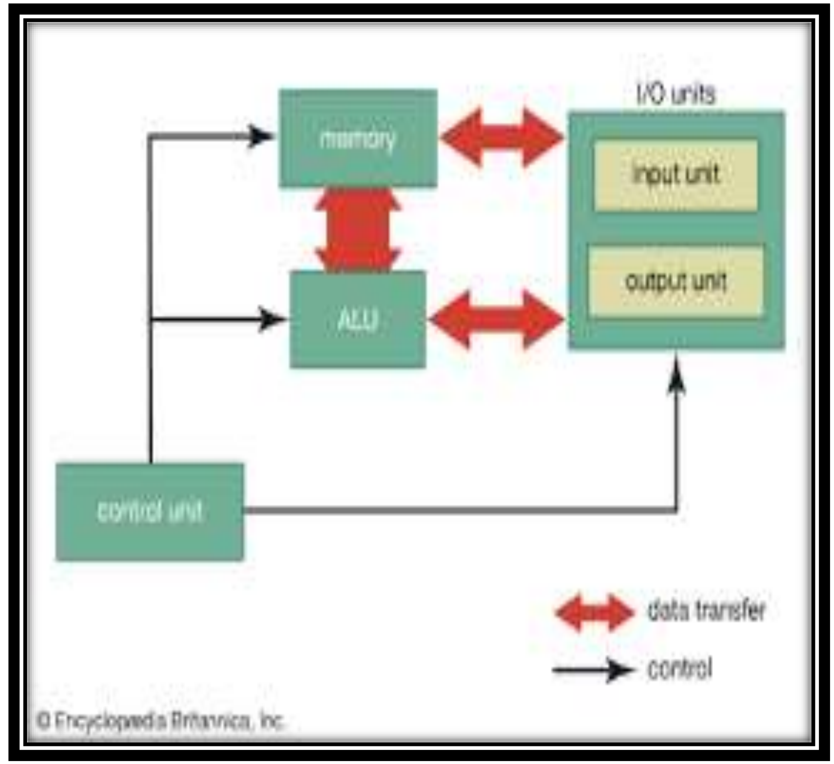
1. Computer system has many resources that may be required to solve a problem.
2. If there are numerous and possibly conflicting requests for resources, OS decides how to allocate them to specific application program, so that it can cater to user needs efficiently and fairly.

Example: Mainframe and Mini computers here resource allocation plays a major role.

System View(2/2)

❑ **Control Program**

1. Controls various user programs and Input/ output devices.
2. Manages execution of user programs to prevent errors and improper use of computer.
3. Specially designed with the operation and control of input/output device.



Operating System Definition

- No universally accepted definition
- "Everything a vendor ships when you order an operating system" is a good approximation
- "The one program running at all times on the computer" is the **kernel**.
- Everything else is either
  - a system program (ships with the operating system) , or
  - an application program.

Computer System Organization

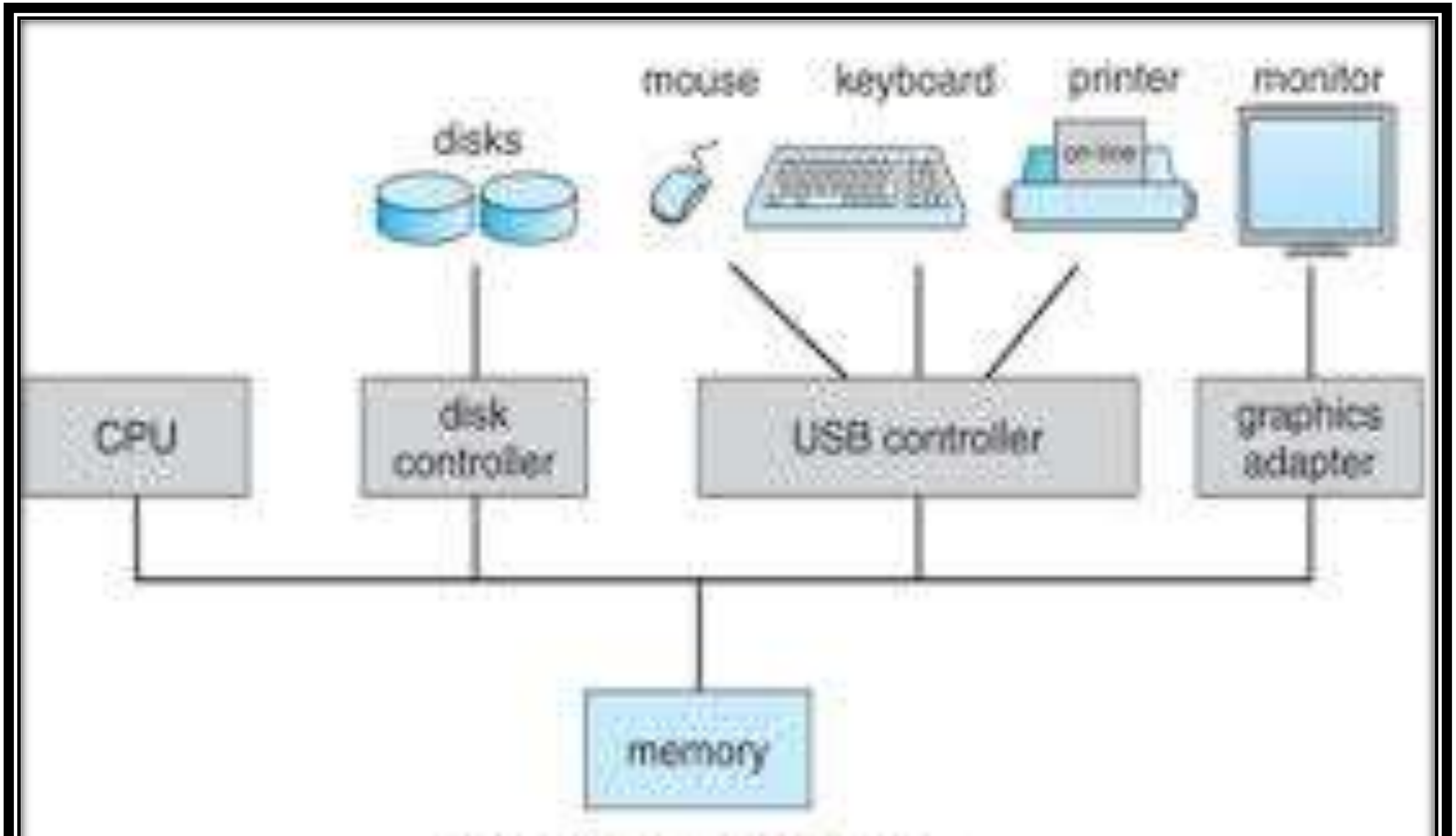
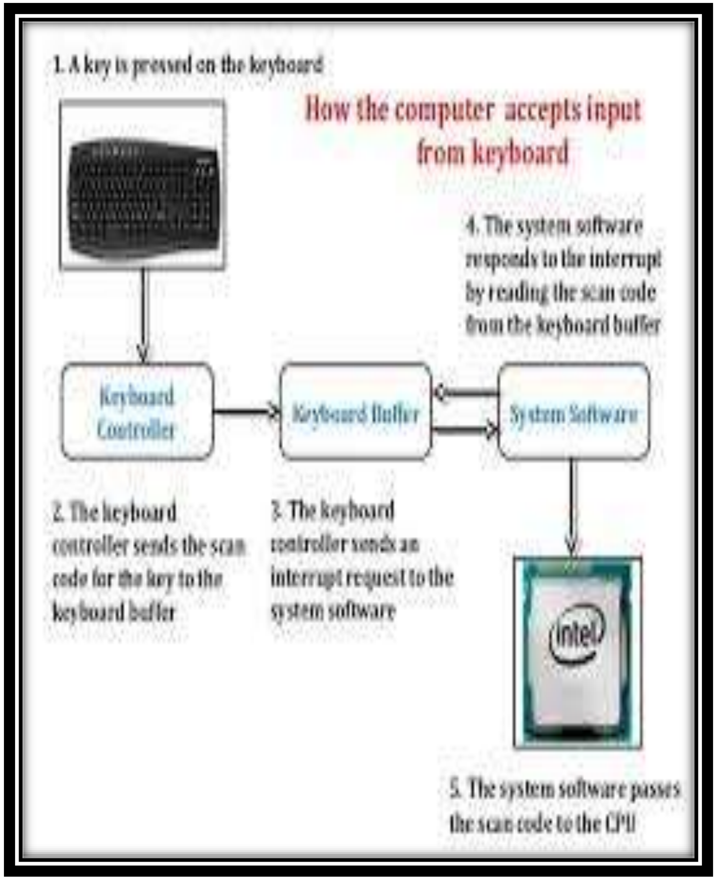


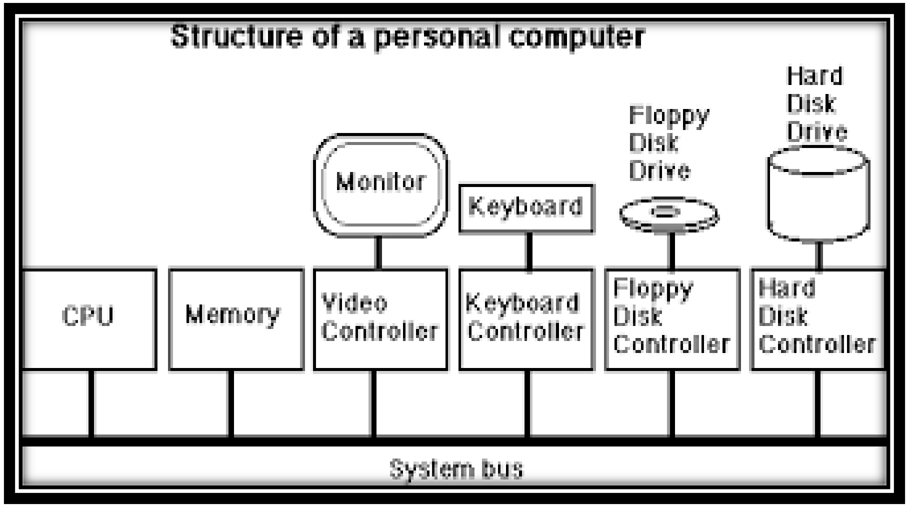
Figure 1.2 A modern computer system.

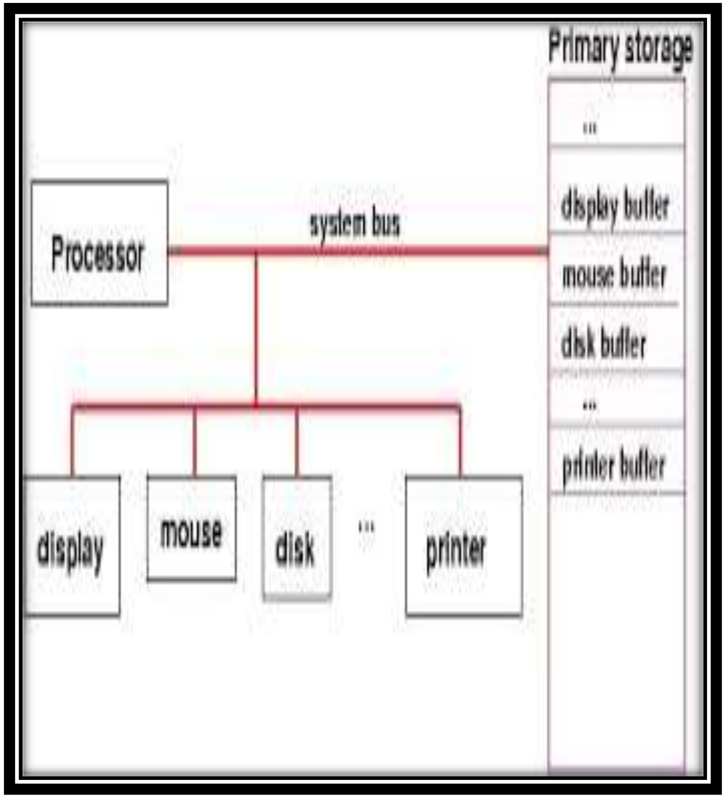
Computer System Organization

- **Computer-system operation**
  1. One or more CPUs, device controllers connect through common bus providing access to shared memory
  2. Concurrent execution of CPUs and devices competing for memory cycles
  3. Device controller is in charge of particular device type.
  4. Each device controller will have a local buffer.
  5. CPU moves data from and to the main memory and to and from local buffer.
  6. Input/output is from the device to local buffer of the controller.
  7. Device controller informs CPU that it has finished its operation by causing an "**INTERRUPT**".



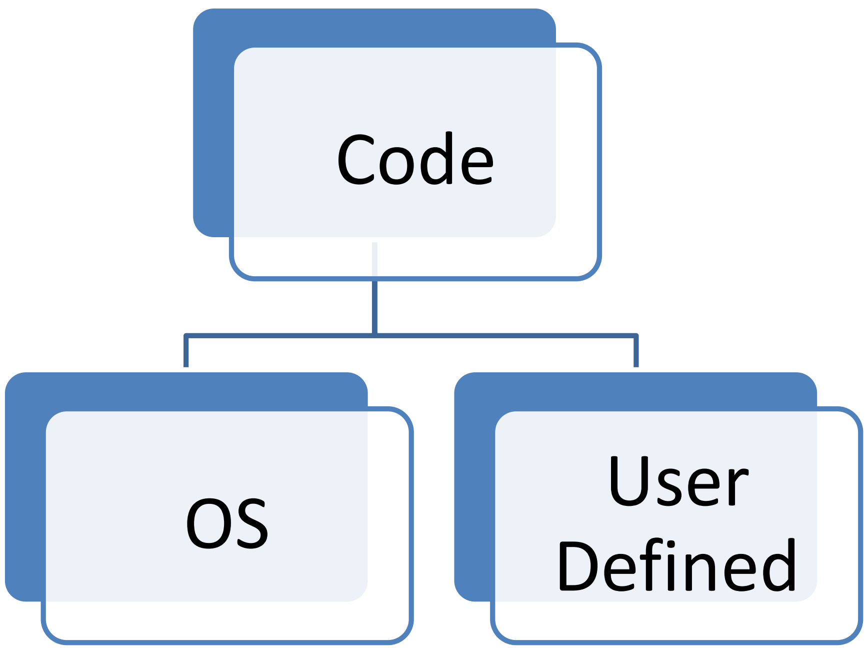
Computer System Organization



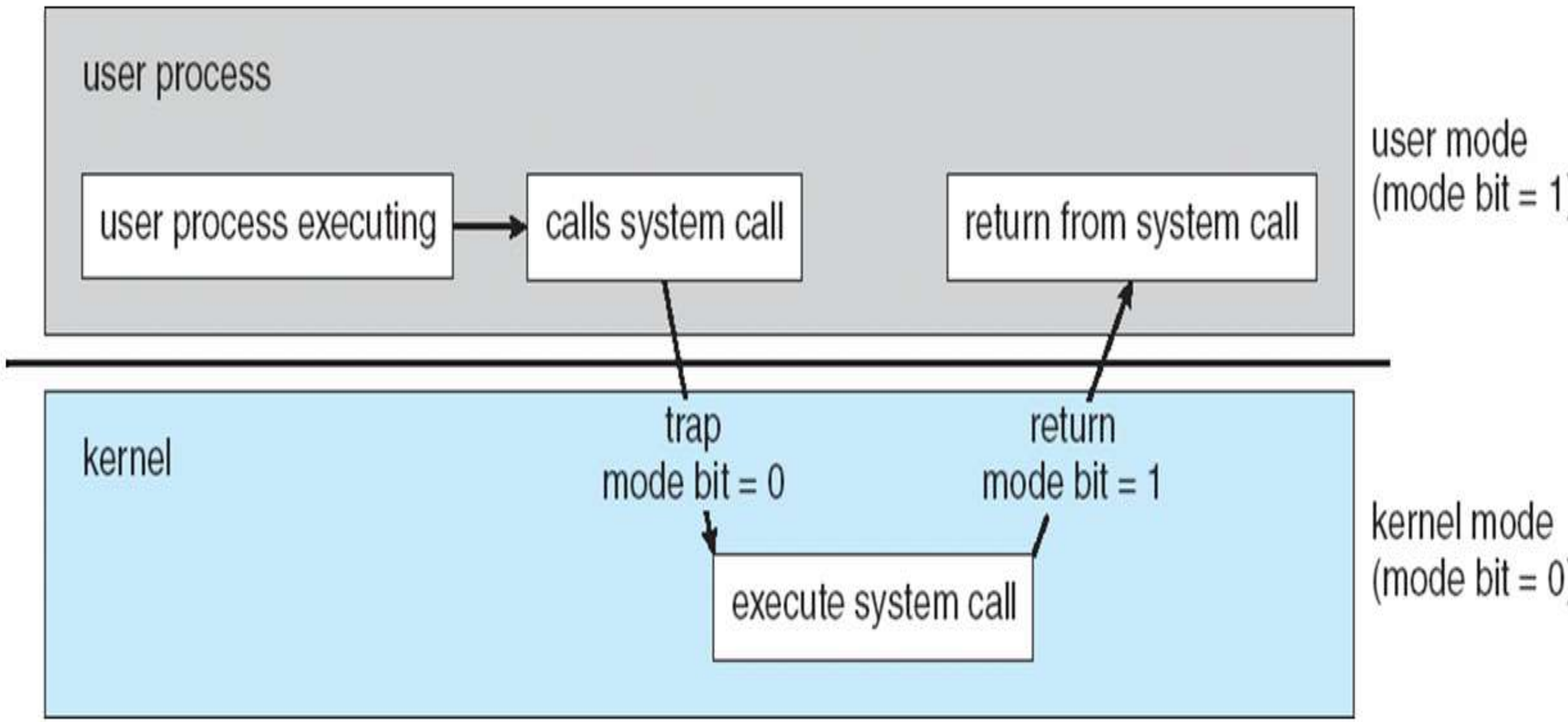


Transition from User to Kernel Mode

- In order to ensure proper execution of the OS, we must be able to distinguish between



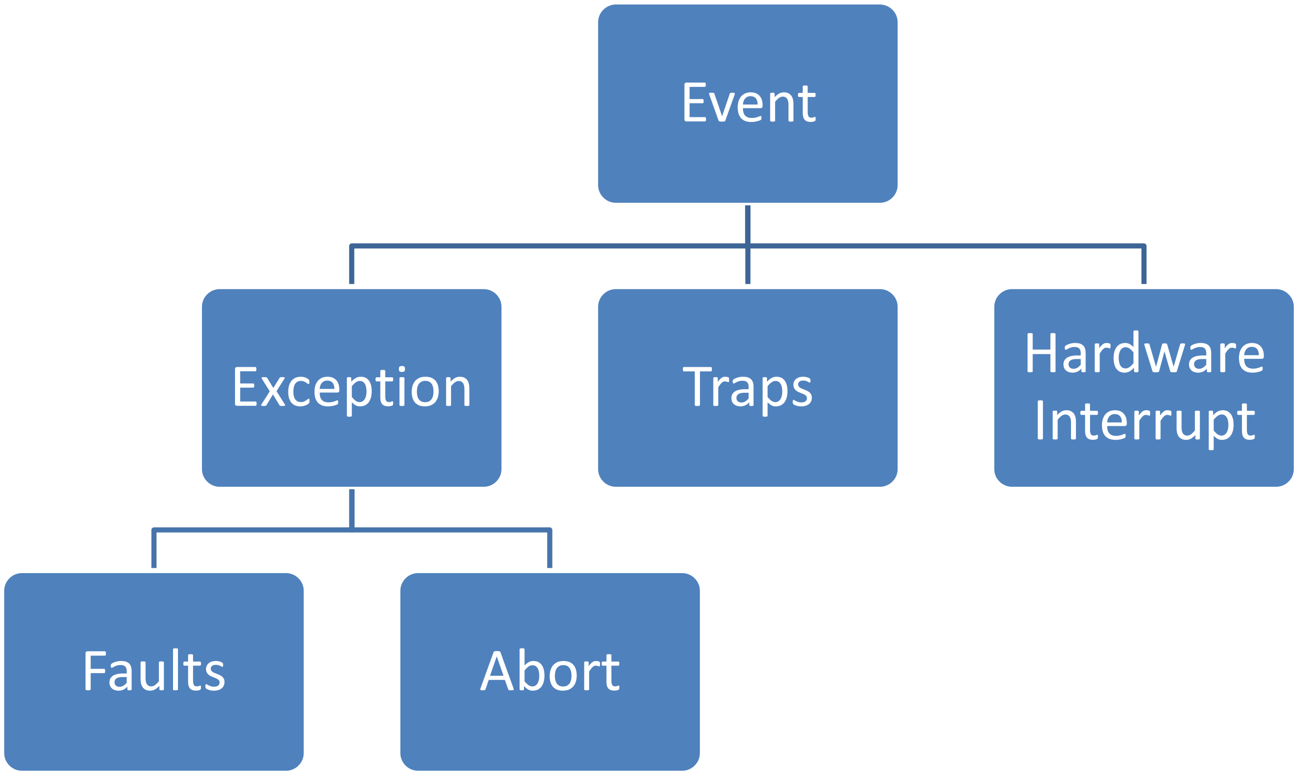
Transition from User to Kernel Mode



Transition from User to Kernel Mode

- **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
    - System call changes mode to kernel, return from call resets it to user

Events(1/4)



Events(2/4)

- **Hardware Interrupt**

1. As the name suggests raised by hardware device(example: USB plugged in)
2. They are asynchronous and can occur at any time.

Events(3/4)

- **Traps**

1. They are software interrupts.
2. Raised by user program, to invoke OS functionalities.
3. Example: print some statement on the monitor, the program makes a system call/ trap to OS, to print on the monitor.

Events(4/4)

- **<u>Exceptions</u>**

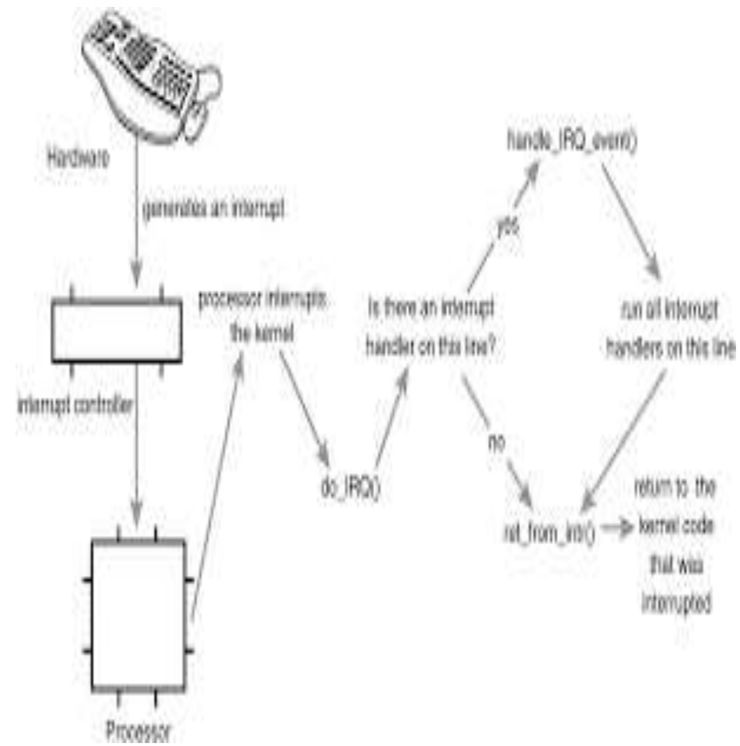
1. Generated automatically by the processor itself as a result of an illegal instructions.

- **Fault**: Recoverable errors (page faults)
- **Abort**: Difficult to recover (divide by 0)

Hardware Interrupt(1/3)

- The kernel is responsible for servicing the request of hardwares.
- The CPU must process the request from the hardware.
- Since the CPU frequency and the hardware frequency is not the same(hardware is slower) so the the hardwares can't send the data/request to the CPU synchronously.
- There are two ways in which CPU can check about the request from a hardware-

  1. **Polling**
  2. **Interrupt**



Hardware Interrupt(2/3)

- In polling the CPU keeps on checking all the hardwares of the availablilty of any request.
- In interrupt the CPU takes care of the hardware only when the hardware requests for some service.
- Polling is an expensive job as it requires a grater overhead.
- The better way is to use interrupt as the hardware will request the CPU only when it has some request to be serviced.
- Different devices are given different interrupt values called IRQ (interrupt request) lines.
- For ex. IRQ zero is the timer interrupt and IRQ one is the keyboard interrupt.
- An interrupt is physically produced by electronic signals originating from hardware devices and directed into input pins on an interrupt controller.
- Some interrupt numbers are static and some interrupts are dynamically assigned.
- Be it static or dynamic, the kernel must know which interrupt number is associated with which hardware.

Hardware Interrupt(3/3)

- The interrupt controller, in turn, sends a signal to the processor. The processor detects this signal and interrupts its current execution to handle the interrupt.
- The processor can then notify the operating system that an interrupt has occurred, and the operating system can handle the interrupt appropriately.
- Interrupt handlers in Linux need not be reentrant. When a given interrupt handler is executing, the corresponding interrupt line is masked out on all processors, preventing another interrupt on the same line from being received. Normally all other interrupts are enabled, so other interrupts are serviced, but the current line is always disabled.

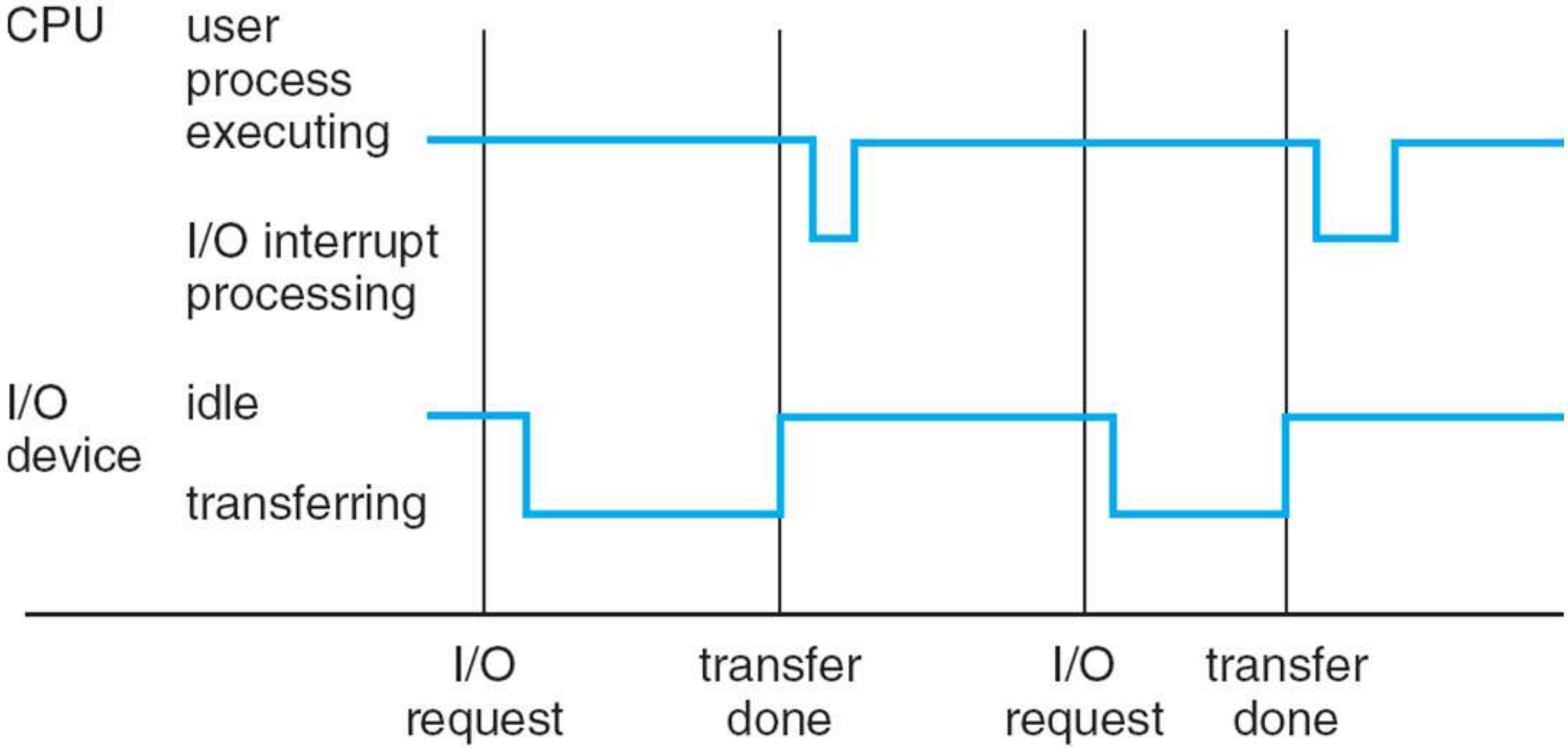
Interrupt Handler

- These are the C functions that get executed when an interrupt comes.
- Each interrupt is associated with a particular interrupt handler.
- Interrupt handler is also known as interrupt service routine (ISR).
- Since interrupts can come any time therefore interrupt handlers has to be short and quick.
- At least the interrupt handler has to acknowledge the hardware and rest of the work can be done at a later time.
-

Top Halves Versus Bottom Halves

- There are two goals that an interrupt handler needs to
- 1. execute quickly and
- 2. perform a large amount of work .
- Because of these conflicting goals, the processing of interrupts is split into two parts, or halves.
- The interrupt handler is the top half.
- It is run immediately upon receipt of the interrupt and performs only the work that is time critical, such as acknowledging receipt of the interrupt or resetting the hardware.
- Work that can be performed later is delayed until the bottom half.
- The bottom half runs in the future, at a more convenient time, with all interrupts enabled.
- Let us consider a case where we need to collect the data form a data card and then process it.
- The most important job is to collect the data from data card to the memory and free the card for incoming data and this is done in top half.
- The rest part which deals with the processing of data is done in the bottom half.

Interrupt Timeline



Computer System Architecture

- A computer system can be categorized based on number of processors used.

Uni- processor

Multi-processor

Processor and Core

❑Processor

- Processor (often referred as a micro processor, CPU - Central Processing Unit, brain of the computer) is the basic computation unit of a computer.
- The CPU of a computer is responsible for handling all the instructions it receives from the hardware and the software running on a computer.
- They are very important part of a computer as they are responsible for the functioning of the computer.
- Every desktop, Smartphone, tablet, smart watch, or any device running an operating system consists of a processor. The processors can be of small or big in efficiency and size but no device can work without a processor.
- The number of operations the computer can handle are decided by the number of cores of a computer

❑Core

- Cores are the parts of a processor which receives instructions, works on it and gives the results. In simple words, it is the basic computer unit of a computer .
- Every processor has core(s) which make it to function.
- It can run a single program context (or multiple ones) maintaining the correct program state, registers, and correct execution order, and performing the operations through ALUs. They are the parts of a processor which do simultaneous processing of multiple tasks at a given time.

Uni- Processor

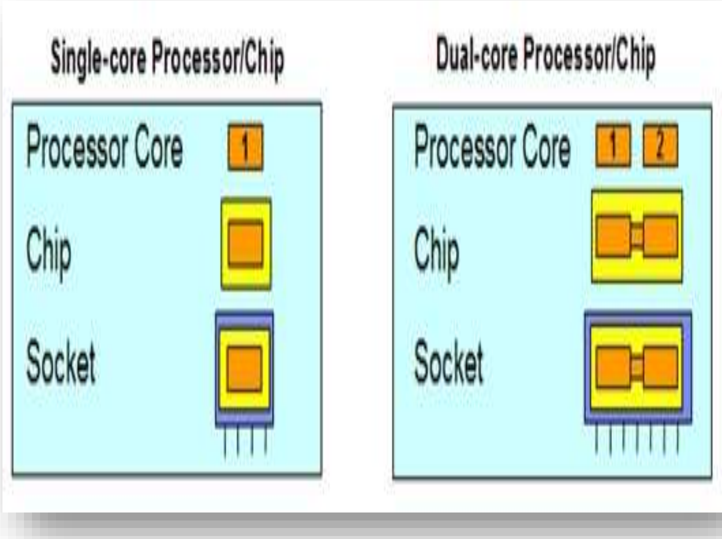
1. A system that has one main CPU, and is capable of executing a general purpose instruction set, including instructions from user processes.
2. Some systems have special purpose processor to perform specific task.
3. These special purpose processors have limited instruction set and do not run user processes.
4. They are managed by the OS by sending information about the next task and monitor their status.
5. Example: PCs contain a microprocessor in the keyboard to convert the keystrokes into codes to be sent to the CPU.
6. These processors do not convert a single processor system into multiprocessor system.

Multi-processor

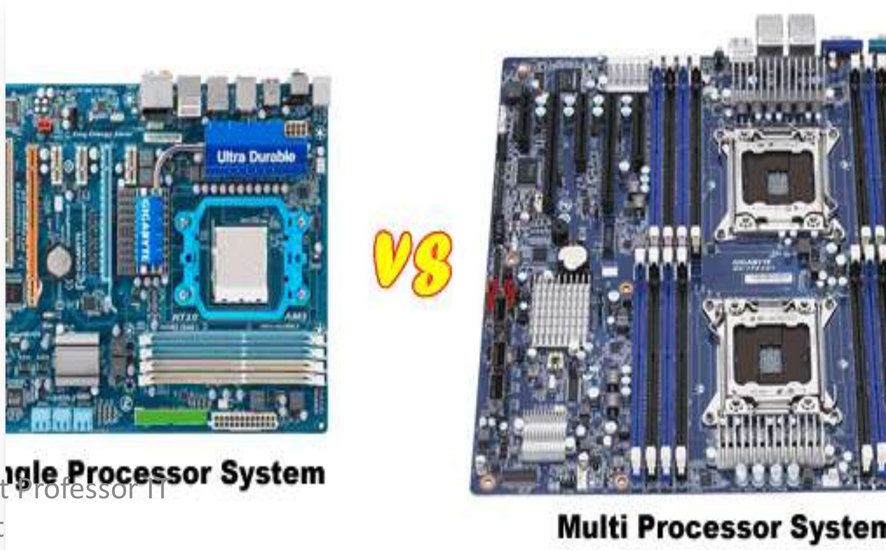
1. Also known as parallel or tightly coupled systems as they can run multiple process in parallel to each other efficiently.
2. Two or more processors will be in close communication with each other with shared memory, storage and power supply.
3. In these days rather than using multi processor systems, people tend to buy single processor with multiple cores. This can be more efficient as internal on-chip communication is faster than communication between different processors.
   Multi-core processors also use less power than that of multiple single core processors.

Multi-processor Advantages

- **Increased throughput**
  - As there are a number of processors, more work can be done in less time. These multiple processors run parallel to each other increasing the performance of the system.



- **Reliability and failure-free**
  - Failure of any processor will not affect the functionality of the system, as there are a number of processors. We can expect failure free service from multi-processor system.

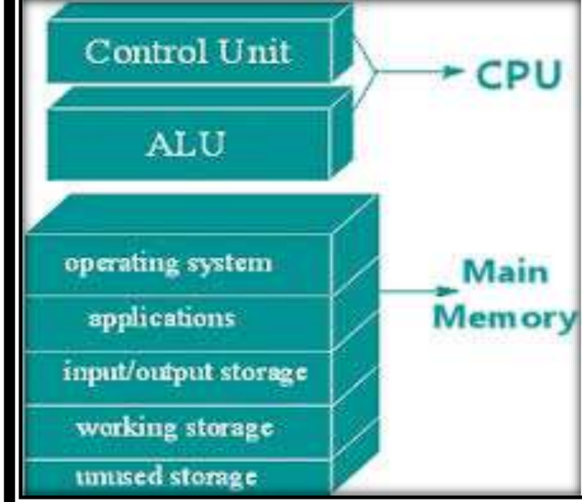


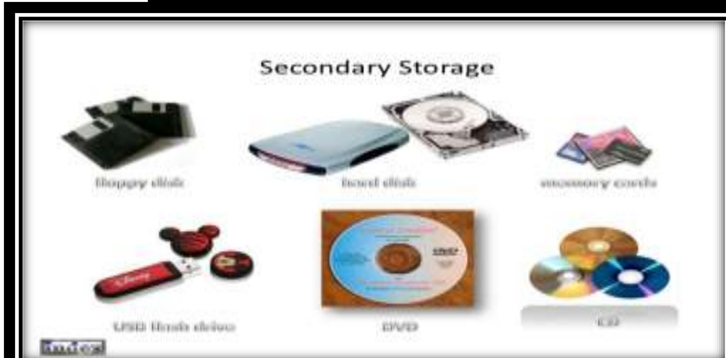
- **Economy of scale**
  - Multi Processor Systems cost less than a number of individual single processor system. In the case of multi processor system expenditure for system cabinet, memory power supply, accessories are saved as these systems share resources like power supply, memory and also space.

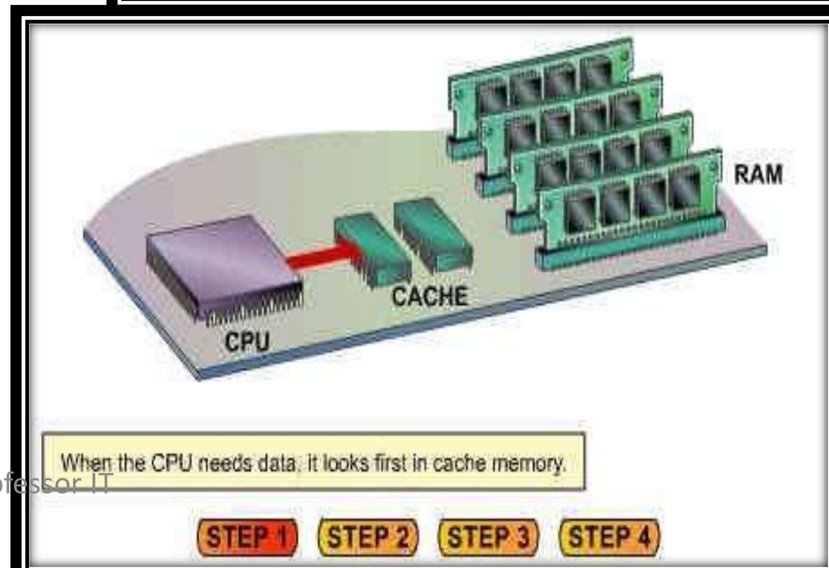
Storage Structure

- **MAIN MEMORY** is the **large storage** media that the **CPU can access directly.**

1. It is very small and volatile storage device.
2. Computer systems have secondary storage that provides large non-volatile storage capacity.
3. Magnetic disks are the common secondary storage devices. Other storage devices are cache memory , cd-rom,magnetic tapes and so on.
4. Storage system is differentiated based on their speed, cost and volatility.

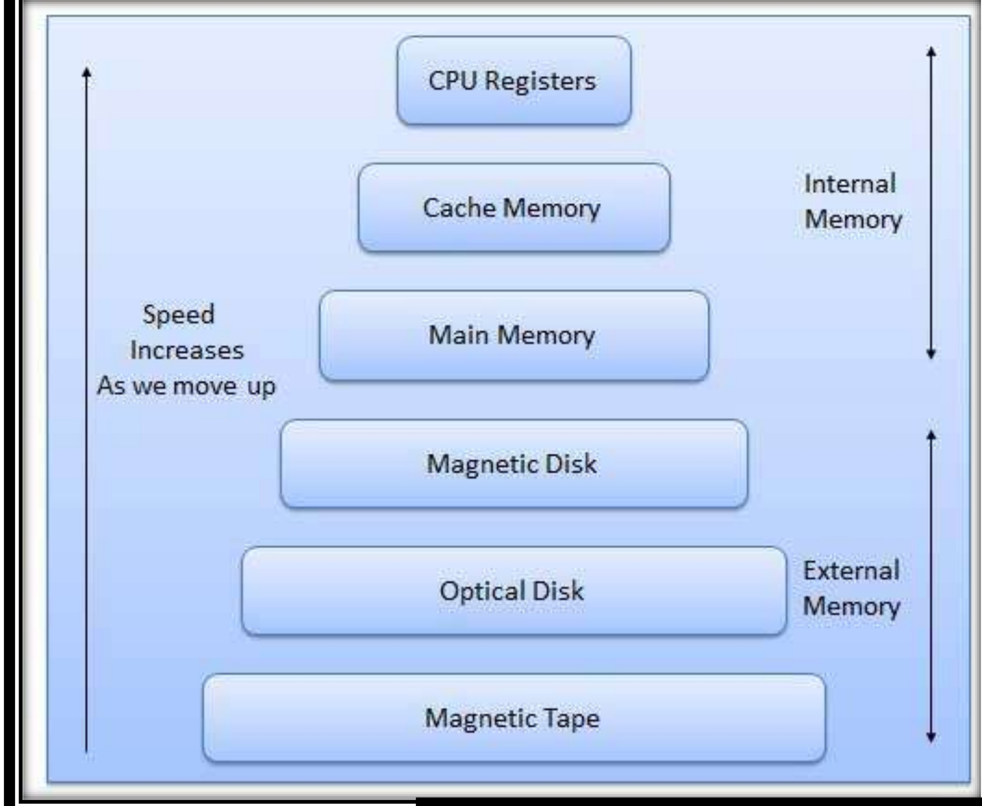






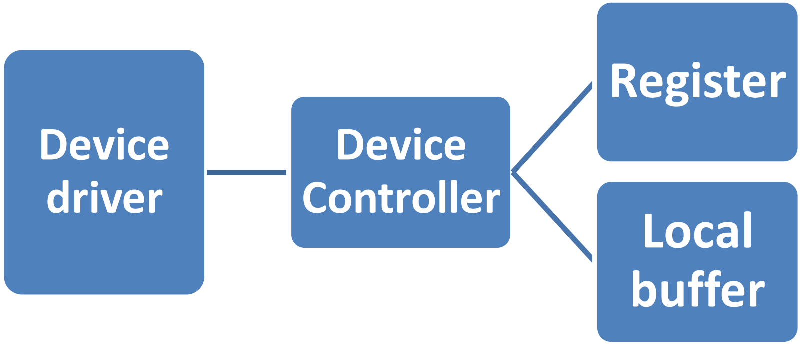
Storage Structure

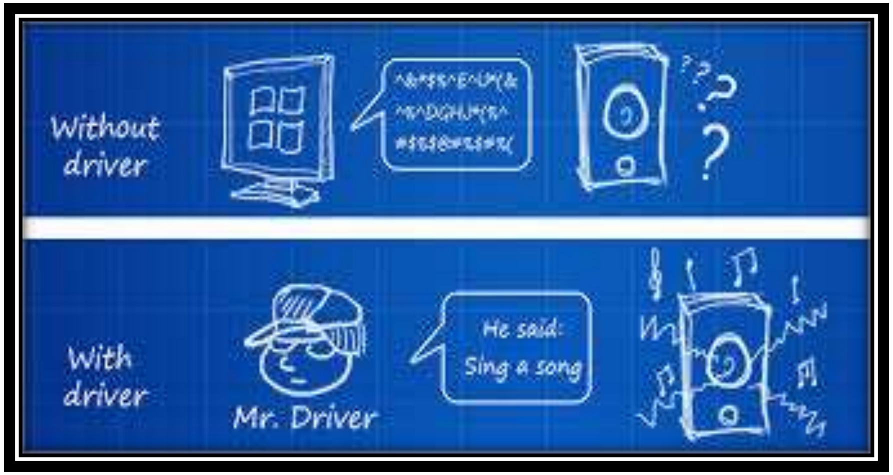
- Storage system can be arranged in hierarchy.
- Flash memory (a kind of memory that retains data in the absence of a power supply.)
- NVRAM(non-volatile RAM) is a DRAM with battery backup.



Input/output Structure

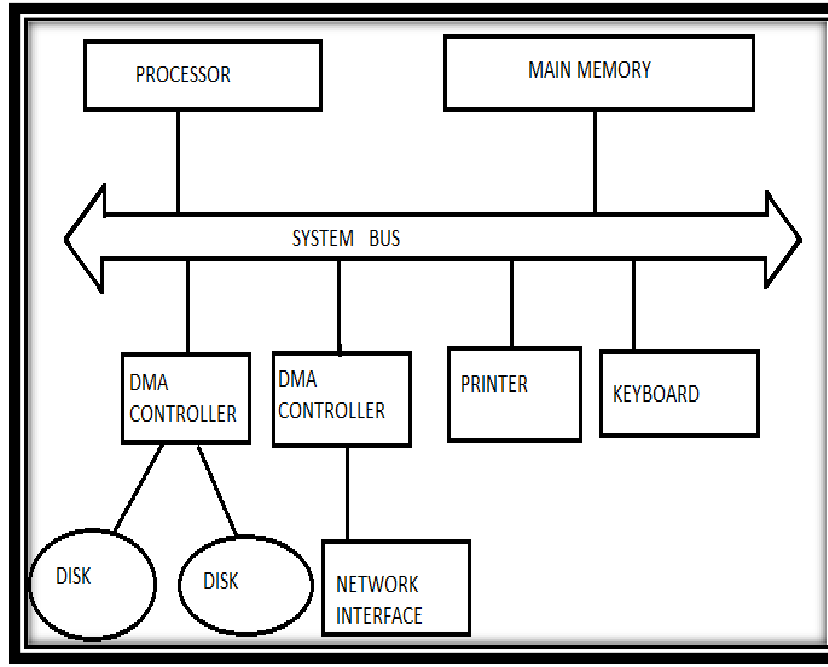
1. Storage is one form of I/O devices.
2. OS spends considerable amount of time in managing I/O device because they have an impact on the performance and reliability of the system.
3. There is one device driver for each device controller.
4. Each device controller maintains a buffer and a register.
5. The device driver understands the device controller and presents a uniform interface to the device and to the rest of the system.
6. When an I/O operation is started, the device driver loads the appropriate registers with in the device controller.
7. The device controller examines the content of the register and determines what action must be taken.

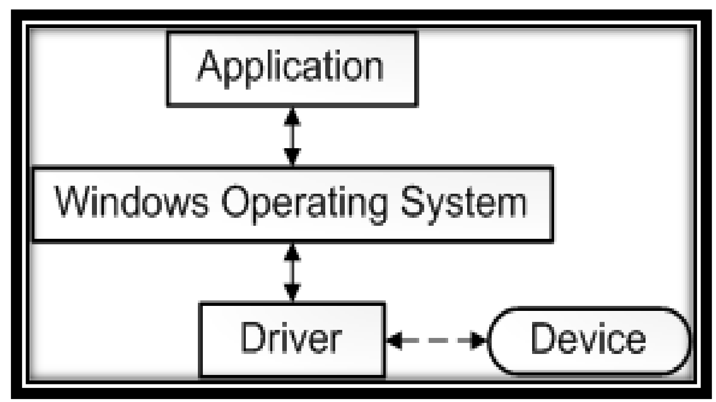




Input/output Structure

1. Once the transfer is complete the device controller informs the device driver via an interrupt that the operation is complete.
2. The device driver then returns control to the OS through an interrupt.
3. This form of interrupt driven I/O works well for moving small amounts of data but it is a big overhead when bulk transfer is required.
4. To solve this problem, DMA(direct memory access) is used.
5. **DMA is used for high speed I/O devices.**
6. The device controller transfers entire block of data to or from buffer storage to main memory without **CPU Intervention.**
7. Only one interrupt is generated per block, rather than one interrupt per byte

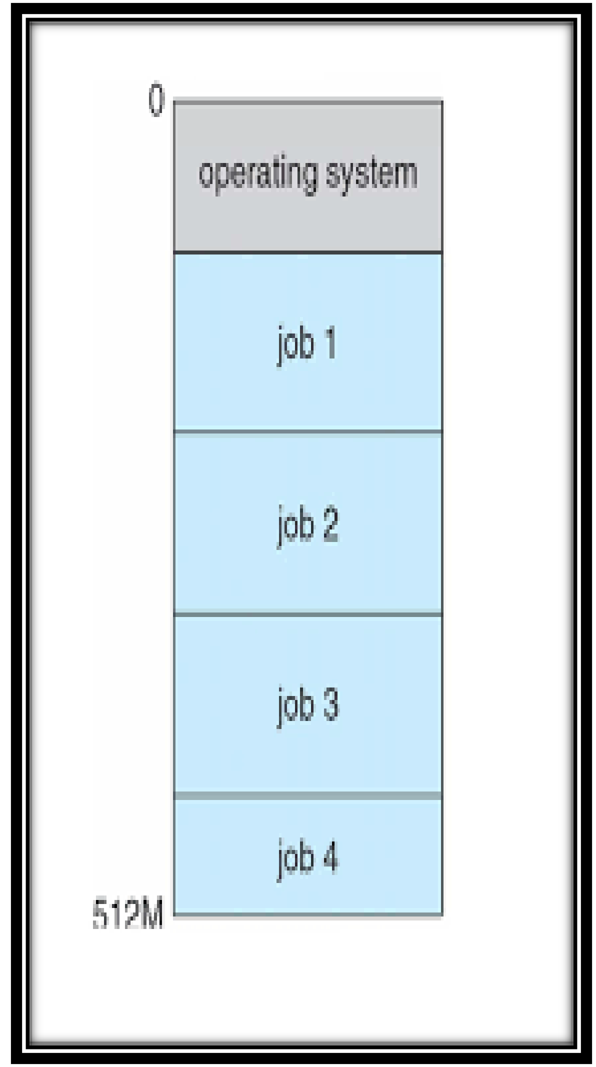




Operating System Structures(1/5)

- **Multiprogramming System**

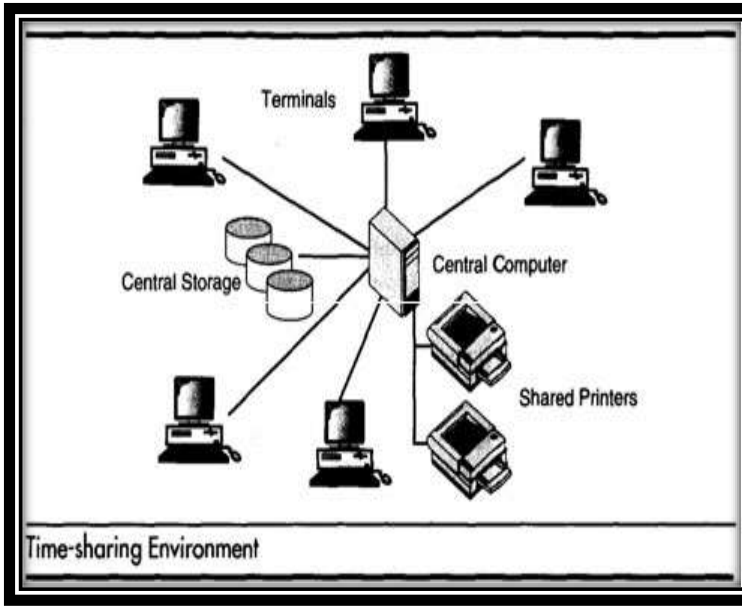
1. Single user cannot keep the CPU and I/O device busy at all times.
2. Multiprogramming increases CPU utilization by organizing jobs so that CPU always has one to execute.
3. The OS has several jobs in memory simultaneously.
4. The OS picks up and starts executing one of the jobs.
5. Eventually if this job may not need the CPU due to some I/O operation to complete, then in **"NON_MULTIPROGRAMMED" System CPU WOULD SIT IDLE.**
6. **But in a multi programmed system instead of having the CPU idle the OS switches to the next job in the memory**.



Operating System Structures(2/5)

- **Time Sharing(Multi-Tasking)**

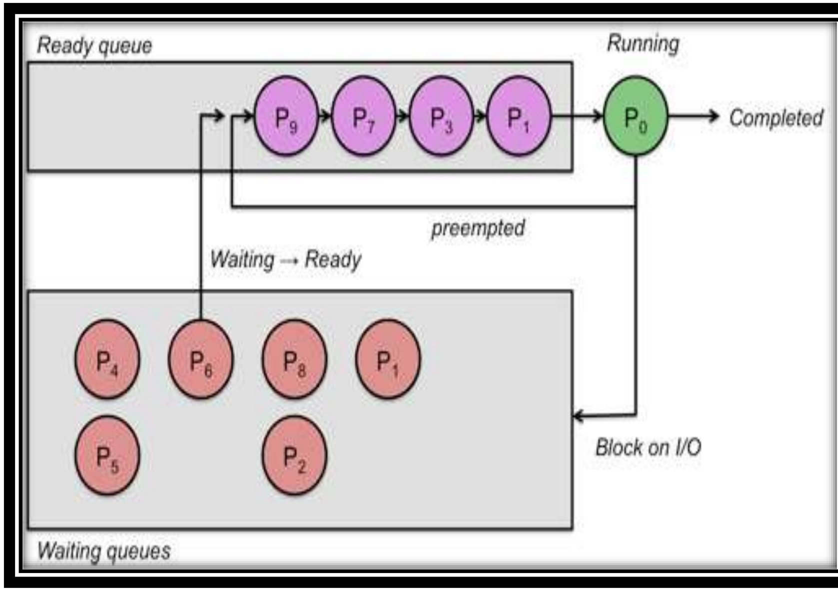
1. It is a logical extension of multi-programming in which CPU switches among jobs so frequently that users can interact with each job while it is running, creating interactive computing.
2. Allows many users to share the computer simultaneously.
3. It require an **" INTERACTIVE SYSTEM"** that provides direct communication between the user and the system.
4. Response time should be less than 1 second.
5. Each user has at least one program executing in memory.
6. A program loaded into memory and executing is called **process.**
7. Timesharing and multiprogramming requires several jobs to be kept simultaneously in memory.



Operating System Structures(3/5)

❑ Job Scheduling

1. A job pool consists of all processes residing on disk and awaiting allocation of main memory.
2. If several jobs are ready to be brought into memory and if there is not enough room for them, then the system must choose among them.
3. Making decision is **"Job scheduling".**



Operating System Structures(4/5)

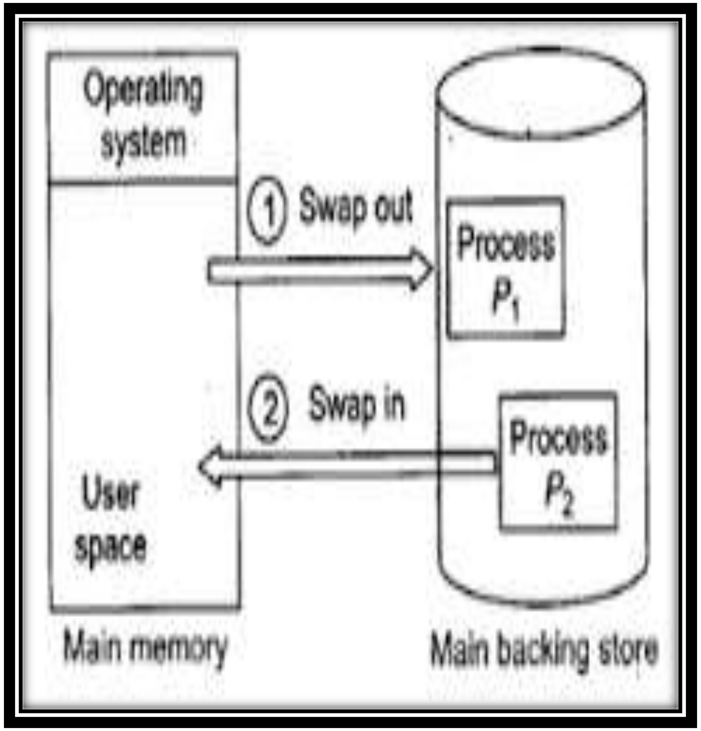
- **<u>CPU Scheduling</u>**

1. If several jobs are ready to run at the same time, then the system must choose among the process, which to execute?
2. Making this decision is **"CPU SCHEDULING"**

Operating System Structures(5/5)

❑ **Swapping**

1. In time shared system processes are swapped in and out of main memory into the disk to ensure reasonable response time.
2. A common method for achieving this is **"VIRTUAL MEMORY".**
3. The main advantage of this is that it enables users to run programs that are larger than actual physical memory.



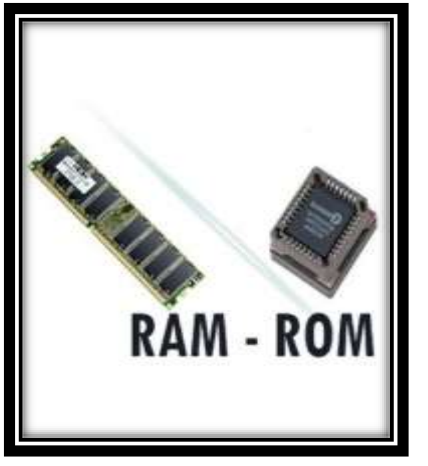
Process Management

- A process is a program in execution.
- Example: A word processing program run by an individual user on a PC is a process.
- A process requires certain resources like CPU time, Memory, I/O devices to complete its task.
- When the process terminates, the OS reclaims all the reusable resources.
- A program in a file stored on the disk is a passive entity, where as a process is an active entity located in the memory.
- System consists of collection of processes, some of which are system processes and some are user processes. All these processes can execute concurrently.
- OS responsible for the following activities of the process management.

1. Creating and deleting of the user and system processes.
2. Suspending and resuming processes.
3. Providing mechanism for process synchronization(shared memory)
4. Providing mechanism for process communication(IPC)
5. Providing deadlock handling mechanism.

Memory Management

- Main Memory is the central to the operation of the computer system.
- The CPU reads the instruction from main memory during **instruction fetch cycle** and during the **data fetch cycle** it reads and writes the data.
- The main memory is the only storage device in which the CPU is able to address and access directly.
- For a program to be executed, it must be loaded into memory and mapped to absolute address.
- When the program terminates, all available memory will be returned back.
- To improve the utilization of CPU and the response time several programs will be kept in memory.
- Several memory management schemes are available and selection depends on the HARDWARE design of the system.
- The OS is responsible for the following activities:

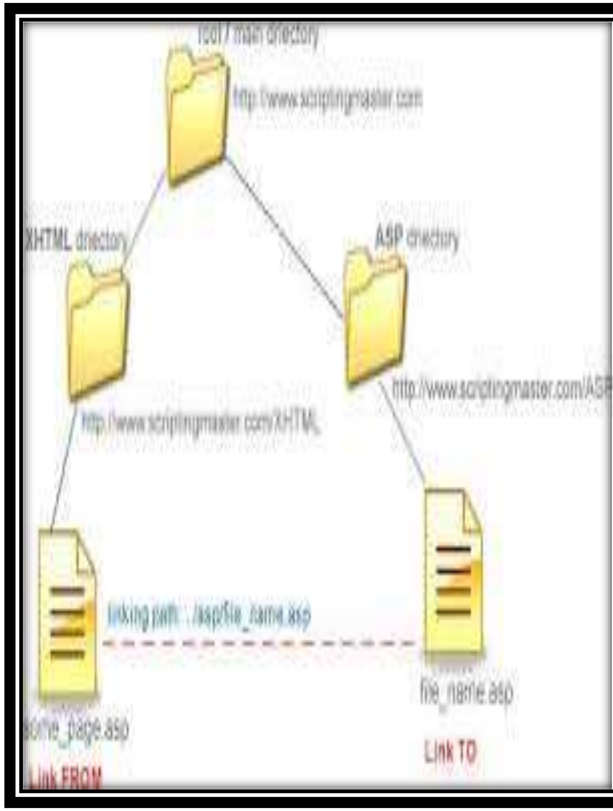


1. Keeping track of which parts of the memory are used and by whom
2. Deciding which process and data to move into and out of memory.
3. Allocating and de allocating memory space as needed.

Storage Management

- File Management is one of the visible components of OS.
- Computer can store information on different types of physical media like magnetic disks, magnetic tapes, optical disks, etc.
- These devices have their own unique characteristics like access speed, capacity, data transfer rate, and access methods(sequential or random)
- A file is a collection of related information defined by its creator. They commonly represent programs(source ad object) and data.
- Data files may be numeric or alphanumeric.
- Files can be organized into directories to make them easier to use.
- The OS is responsible for the following activities,

1. Creating and deleting files
2. Creating and deleting directories
3. Supporting primitives for manipulating files and directories
4. Mapping files onto secondary storage
5. Backing up files on stable(non-volatile) storage media.



Mass storage Management

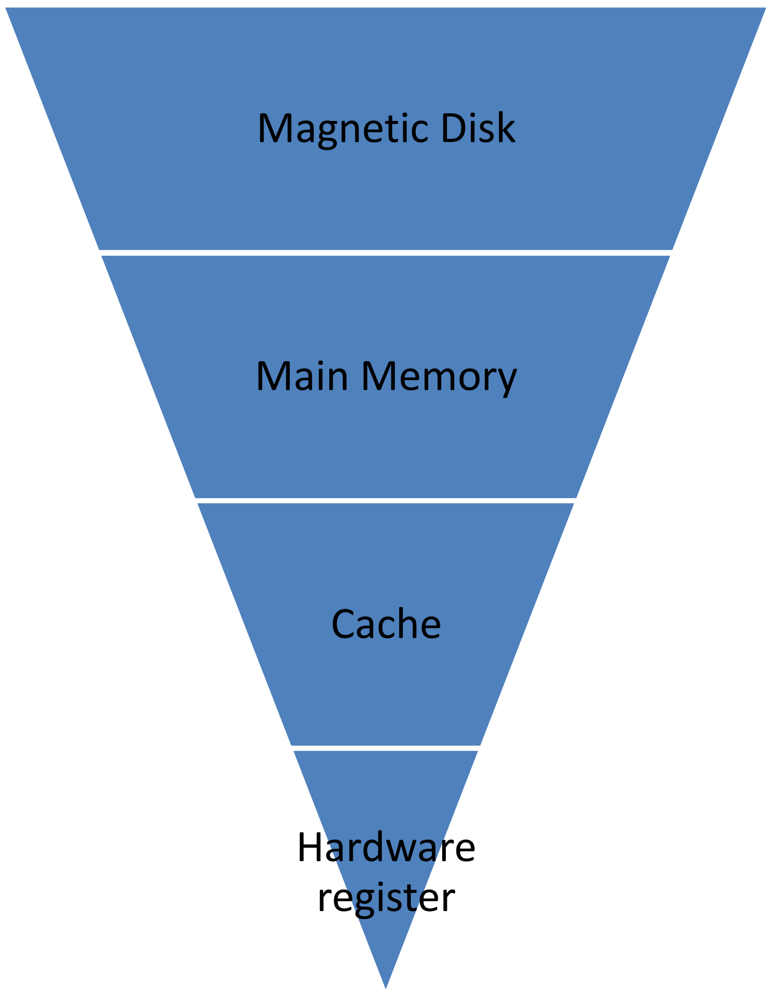
- Computer system must provide secondary storage to back up main memory because
1. It is too small to accommodate all data and programs
2. Data held in this memory is lost when the power goes off.
- Most programs including compilers, assemblers, word processors, editors etc are stored on the disk until loaded into the memory and then use disk as both source and destination of processing.
- Hence proper management of disk storage is very important.
- The OS responsible for the following activities:
1. Free space management
2. Storage allocation
3. Disk scheduling
- Slow and low cost but high capacity backup storage devices are called "ternary devices" which are used for backup of the regular disk data.
- Example: magnetic tapes and their drives, CD and DVD drives.

Caching(1/3)

- It is important principle of a computer system and is fast memory which is used for storing information on a temporary basis.
- First, the cache is searched when a particular piece of information is required during processing. If the information is already available, then it is directly used from the cache, otherwise we use information from source, putting a copy in the cache, under assumption that we will need the information again very soon.
- Internal registers like "Index Registers" can be used as high speed cache for main memory.
- Cache have limited size and thus cache management is an important design problem.

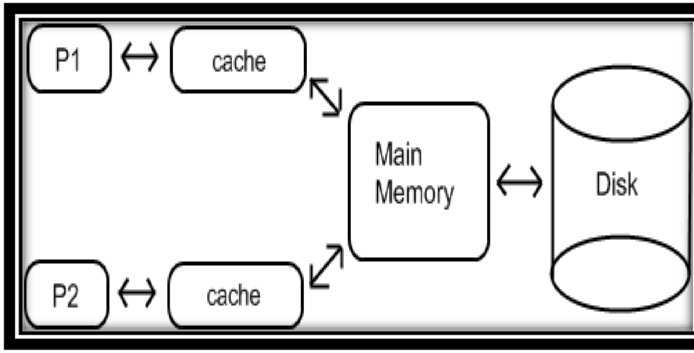
Caching(2/3)

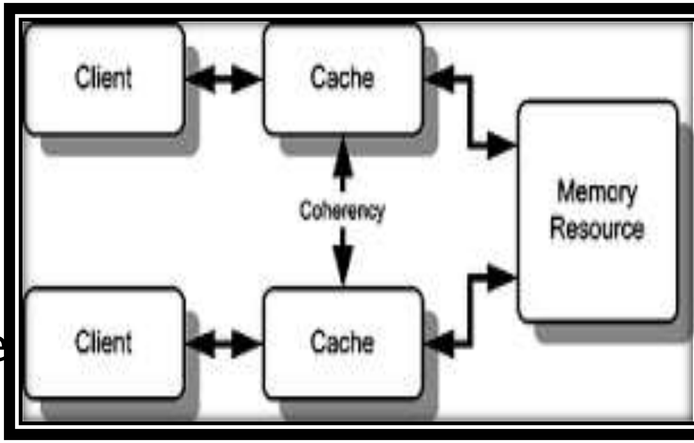
- In a hierarchical storage structure, the same data may appear in different levels of storage system.
- Example: Suppose that an integer A is to be incremented by 1 is located in file B which resides on disk, the migration of integer A from disk to register is shown below:



Caching(3/3)

- Once the increment to A takes place in the internal registers, the value of A differs in various storage system. The value of A becomes same only after the new value of A is written from the internal register back to the disk.
- In **Multi-tasking environments**, extreme care must be taken to use most recent value, not matter where it is stored in the storage hierarchy
- The situation becomes more complicated in **multi-processor environment**, where each CPU is associated with local cache. A care must be taken to make sure that an update to the value of A in one cache is immediately reflected in all other caches.
- This situation is called as "**cache coherency**".
- The situation becomes even more complex in a **distributed environment.** Several copies of the same file can be kept on different computers. Since the various replica may be accessed and updated concurrently, some distributed systems ensure that, when a replica is updated in one place all other replicas are updated as soon as possible.





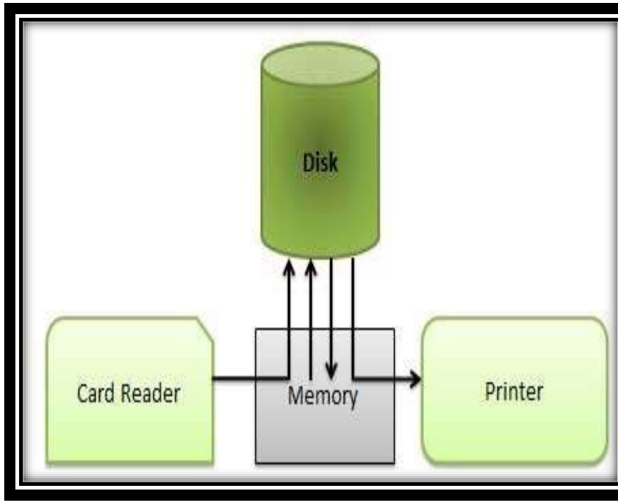
Input/output systems

- OS hides peculiarities of hardware device from the user.
- I/O subsystem consists of several components like,

1. The memory management components that include buffering, caching and spooling.
2. A general device driver interface
3. Drivers for specific hardware devices.

- Only the device driver knows about the peculiarities of each of the devices.

Spooling



- Spooling is an acronym for simultaneous peripheral operations on line. Spooling refers to putting data of various I/O jobs in a buffer. This buffer is a special area in memory or hard disk which is accessible to I/O devices.
- An operating system does the following activities related to distributed environment –
- Handles I/O device data spooling as devices have different data access rates.
- Maintains the spooling buffer which provides a waiting station where data can rest while the slower device catches up.
- Maintains parallel computation because of spooling process as a computer can perform I/O in parallel fashion. It becomes possible to have the computer read data from a tape, write data to disk and to write out to a tape printer while it is doing its computing task.

Difference between spooling, buffering and caching.

- **Spooling**

1. Acronym of "Simultaneous Peripheral Operation On-Line".
2. Its a process of placing data in temporary working area for another program to process.
3. E.g: Print spooling and Mail spools etc.
4. When there is a resource (like printer) to be accessed by two or more processes(or devices), there spooling comes handy to schedule the tasks. Data from each process is put on the spool (print queue) and processed in FIFO(first in first out) manner.
5. With spooling all process can access the resource without waiting.
6. After writing the data on spool, process can perform other tasks. And printing process operates separately.
7. Without spooling, process would be tied up until the printing finished.
8. **Spooling is useful for the devices which have differing data access rate. Used mainly when processes share some resource and needed to have synchronization.**

- **Buffering**

1. Preloading data into a reserved area of memory (the buffer).
2. It temporarily stores input or output data in an attempt to better match the speeds of two devices such as a fast CPU and a slow disk drive.
3. Buffer may be used in between when moving data between two processes within a computer. Data is stored in buffer as it is retrieved from one processes or just before it is sent to another process.
4. With spooling, the disk is used as a very large buffer. Usually complete jobs are queued on disk to be completed later.
5. **It is mostly used for input, output, and sometimes temporary storage of data either when transfer of data takes place or data that may be modified in a non-sequential manner.**

Difference between spooling, buffering and caching.

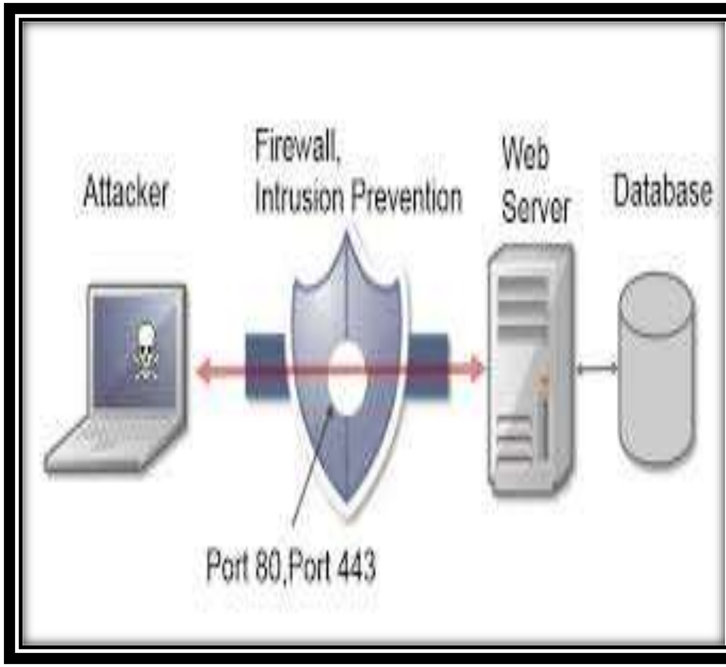
- **Caching**

1. Caching transparently stores data in component called Cache, so that future request for that data can be served faster.
2. A special high-speed storage mechanism. It can be either a reserved section of main memory or an independent high-speed storage device.
3. The data that is stored within a cache might be values that have been computed earlier or duplicates of original values that are stored elsewhere.
4. E.g: Memory Caching, Disk Caching, Web Caching(used in browser), Database Caching etc.
5. **A cache's sole purpose is to reduce accesses to the underlying slower storage.**

Protection

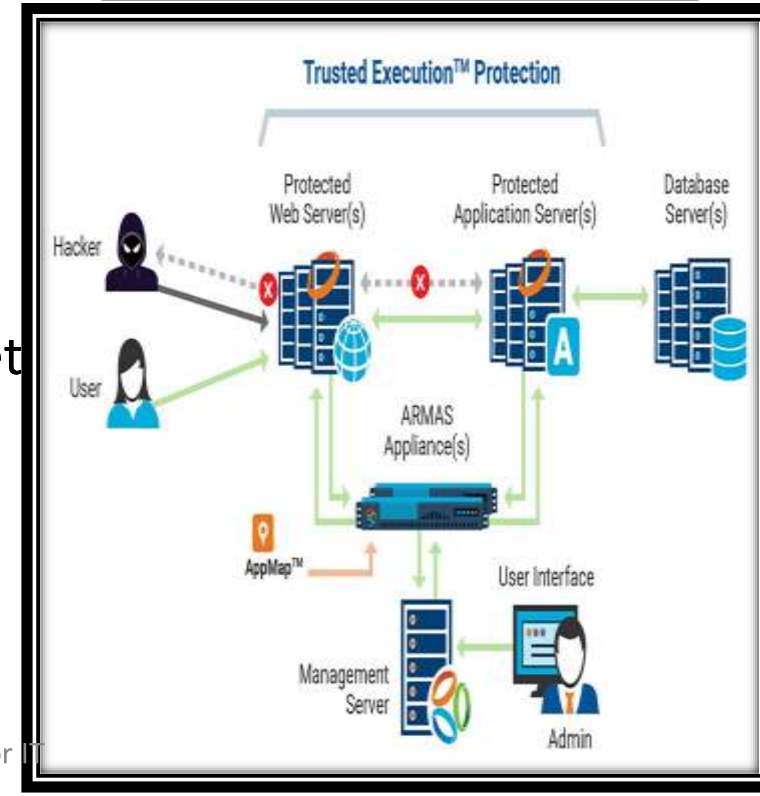
- If a computer system has multiple users and allows the concurrent execution of multiple processes, the a protection mechanism is required to regulate access to data.
- System resources like files, memory segments, CPU etc. are made available to only those processes which have gained authorization from OS.
- **Protection**: is a mechanism for controlling the access or users to the resources defined by a computer system.
- The advantages of providing protection are:

1. It can improve reliability by detecting latent errors at the interface between component subsystems and early detection of interface errors can prevent corruption of good subsystems by another malfunctioning subsystem.
2. Protection can prevent misuse by an unauthorized or incompetent user.



Security

- Security system must defend the system from external and internal attacks.
- **Attacks** can be of various types like virus, worms, theft of service etc.
- Example: If a users authentication information is stolen then the owners data can be stolen, corrupted or deleted.
- The mechanism of protection and security must be able to distinguish among all its users. This is possible because the system maintains a list of **user ids**. These ids are unique per user.
- When it is required to distinguish among a set of users rather than individual users then group functionality is implemented.
- A system wide list of group names and group ids are stored.
- If a user needs to escalate privileges for gaining extra permission then different methods are provided by OS.



THANK YOU