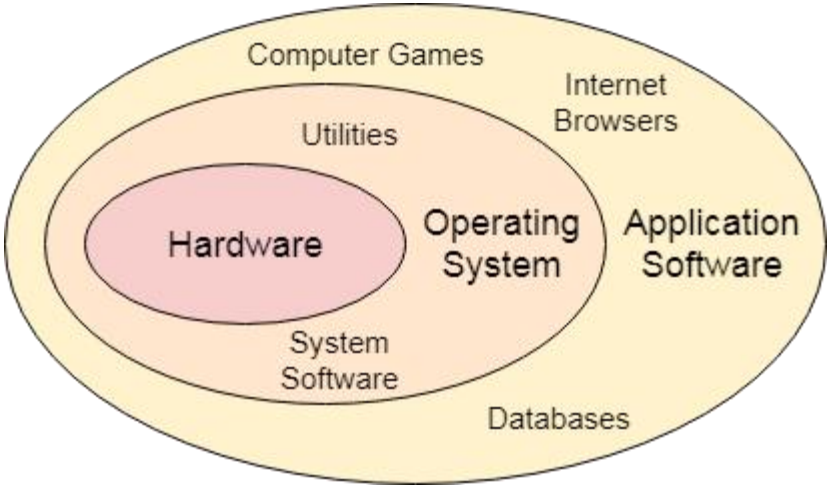
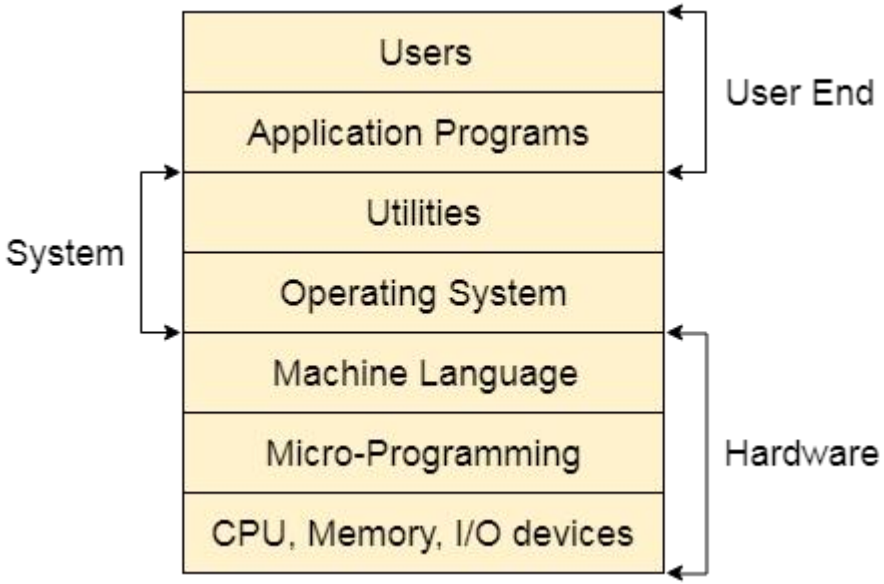
Unit 1

Introduction to Operating System: Introduction and Need of operating system, Layered Architecture/Logical Structure of Operating system, Type of OS(Multiprogramming , Time Sharing, Real Time ,Networked, Distributed, Clustered, Hand Held), Operating system as Resource Manager and Virtual Machine, OS Services, BIOS, System Calls/Monitor Calls, Firmware- BIOS, Boot Strap Loader. Threads- processes versus threads, threading, concepts, models, kernel & user level threads, thread usage, benefits, multithreading models.

- Operating System Definition and Function
- In the Computer System (comprises of Hardware and software), Hardware can only understand machine code (in the form of 0 and 1) which doesn't make any sense to a naive user.
- We need a system which can act as an intermediary and manage all the processes and resources present in the system.



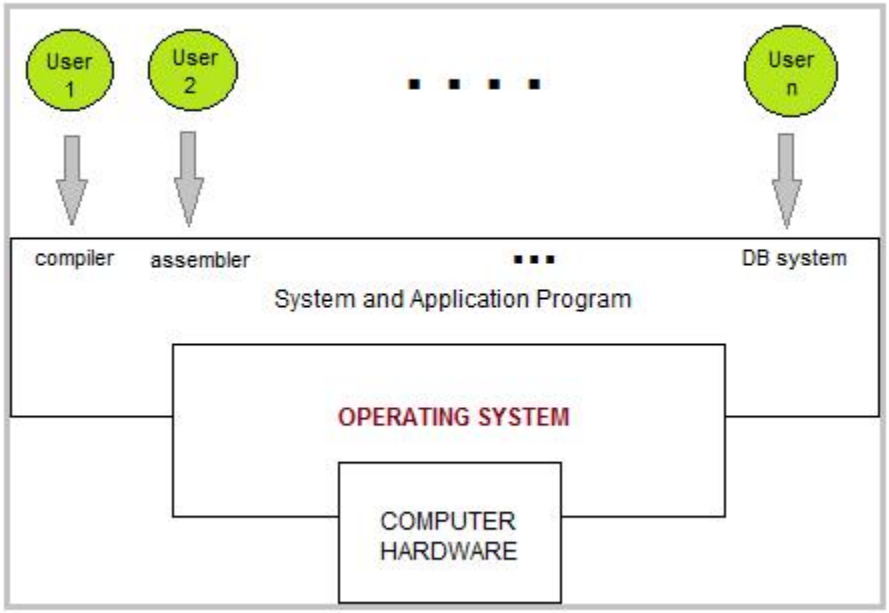
- An **Operating System** can be defined as an **interface between user and hardware**. It is responsible for the execution of all the processes, Resource Allocation, CPU management, File Management and many other tasks.
- The purpose of an operating system is to provide an environment in which a user can execute programs in convenient and efficient manner.
- Structure of a Computer System
- A Computer System consists of:Users (people who are using the computer)
- Application Programs (Compilers, Databases, Games, Video player, Browsers, etc.)
- System Programs (Shells, Editors, Compilers, etc.)
- Operating System (A special program which acts as an interface between user and hardware)
- Hardware (CPU, Disks, Memory, etc)



- Operating System can be defined as an interface between user and the hardware. It provides an environment to the user so that, the user can perform its task in convenient and efficient way. In the Computer System (comprises of Hardware and software), Hardware can only understand machine code (in the form of 0 and 1) which doesn't make any sense to a naive user.
- We need a system which can act as an intermediary and manage all the processes and resources present in the system.

- Why to Learn Operating System?
- An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
- Some popular Operating Systems include Linux Operating System, Windows Operating System, unix , linux,MAC os.

Four Components of a Computer System

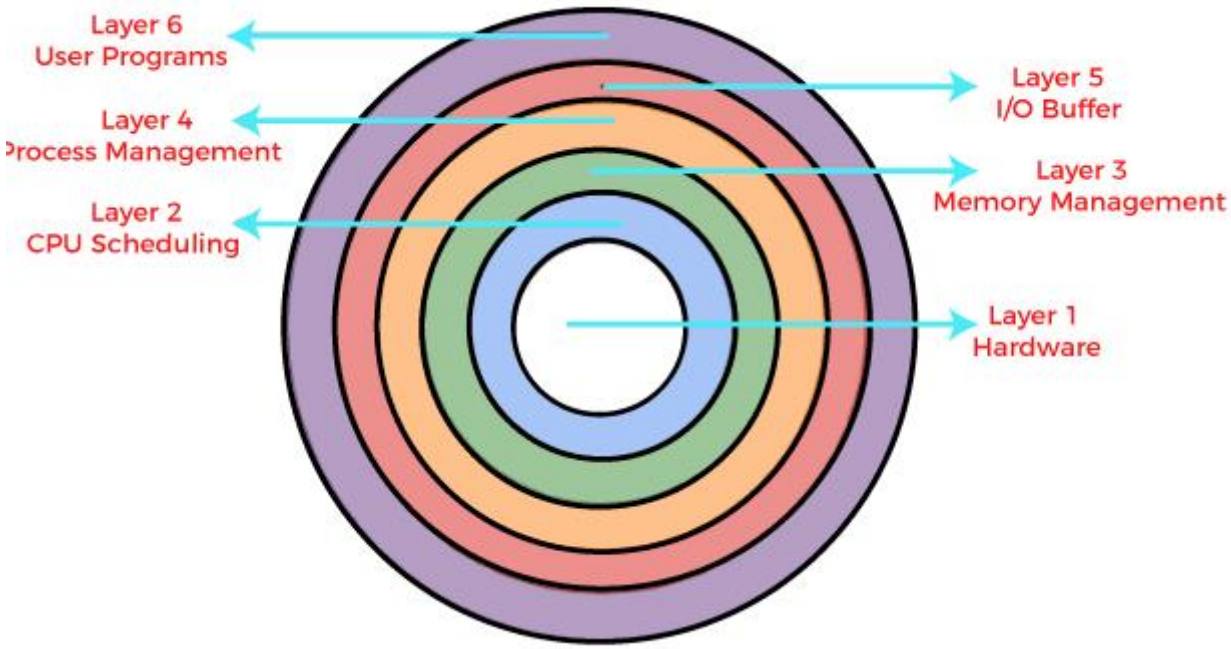


Disadvantages of an Operating system

- **Expensive**
- **Virus Threat**
- **System Failure**

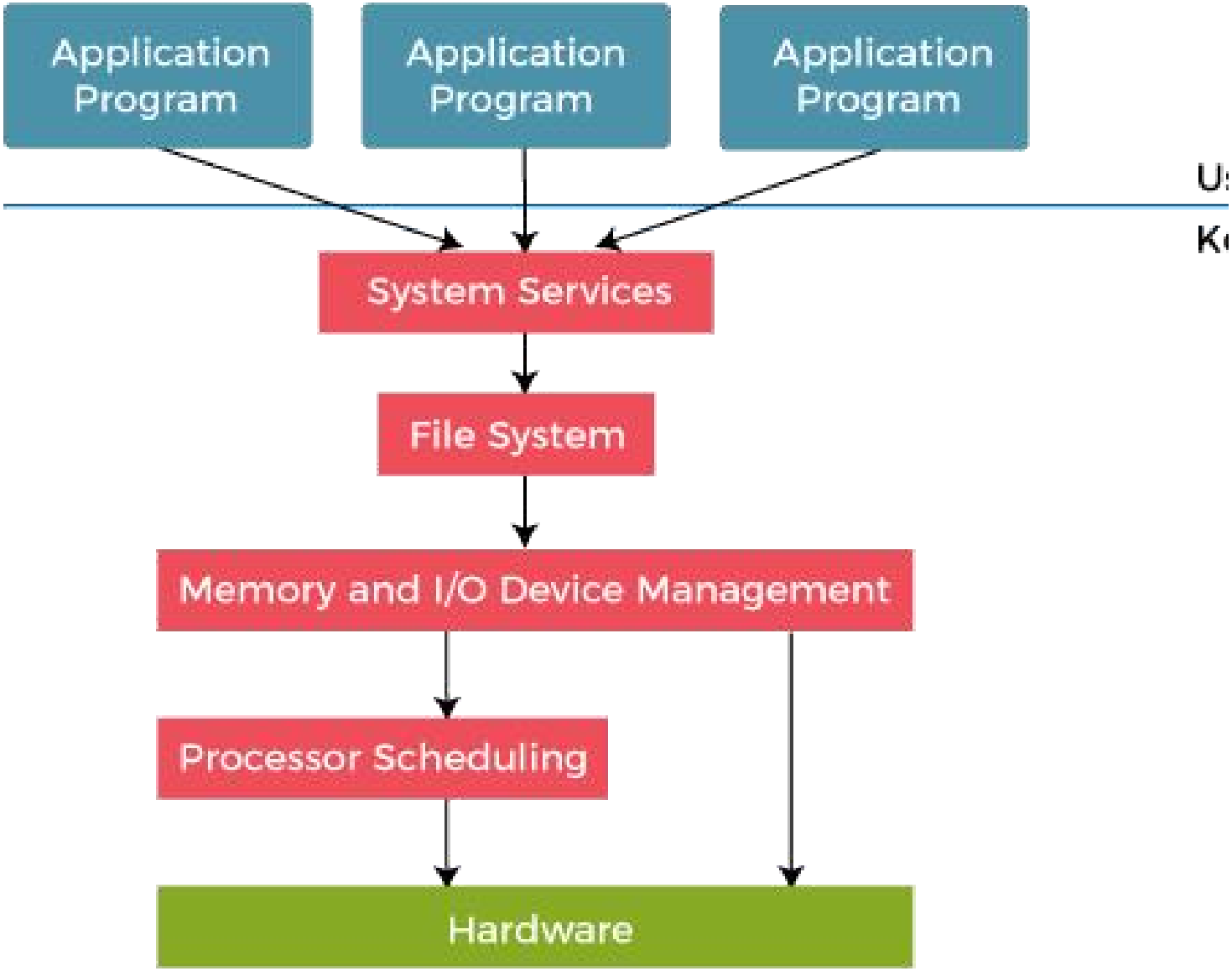
Layered Structure of Operating System

- The layered structure approach breaks up the operating system into different layers and retains much more control on the system. The bottom layer (layer 0) is the hardware, and the topmost layer (layer N) is the user interface. These layers are so designed that each layer uses the functions of the lower-level layers only. It simplifies the debugging process as if lower-level layers are debugged, and an error occurs during debugging. The error must be on that layer only as the lower-level layers have already been debugged.


Layer 6
User Programs
Layer 4
Process Management
Layer 2
CPU Scheduling
Layer 5
I/O Buffer
Layer 3
Memory Management
Layer 1
Hardware

- This allows implementers to change the inner workings and increases modularity.
- As long as the external interface of the routines doesn't change, developers have more freedom to change the inner workings of the routines.
- The main advantage is the simplicity of construction and debugging. The main difficulty is defining the various layers.
- The main disadvantage of this structure is that the data needs to be modified and passed on at each layer, which adds overhead to the system. Moreover, careful planning of the layers is necessary as a layer can use only lower-level layers. UNIX is an example of this structure.

There are six layers in the layered operating system. A diagram demonstrating these layers is as follows:



- **Hardware:** This layer interacts with the system hardware and coordinates with all the peripheral devices used, such as a printer, mouse, keyboard, scanner, etc. These types of hardware devices are managed in the hardware layer. The hardware layer is the lowest and most authoritative layer in the layered operating system architecture. It is attached directly to the core of the system.
- **CPU Scheduling:** This layer deals with scheduling the processes for the CPU. Many scheduling queues are used to handle processes. When the processes enter the system, they are put into the job queue. The processes that are ready to execute in the main memory are kept in the ready queue. This layer is responsible for managing how many processes will be allocated to the CPU and how many will stay out of the CPU.

- **Memory Management:** Memory management deals with memory and moving processes from disk to primary memory for execution and back again. This is handled by the third layer of the operating system. All memory management is associated with this layer. There are various types of memories in the computer like RAM, ROM. If you consider RAM, then it is concerned with swapping in and swapping out of memory. When our computer runs some processes move to the main memory (RAM) for execution, and when programs such as calculator, exit, it is removed from the main memory.
- **Process Management:** This layer is responsible for managing the processes i.e. assigning the processor to a process and deciding how many processes will stay in the waiting schedule. The priority of the processes is also managed in this layer. The different algorithms used for process scheduling are FCFS (first come, first served), SJF (shortest job first), priority scheduling, round-robin scheduling, etc.

- I/O Buffer: I/O devices are very important in computer systems. They provide users with the means of interacting with the system. This layer handles the buffers for the I/O devices and makes sure that they [illegible]. Suppose you are typing from the keyboard. There is a keyboard buffer attached with the keyboard which stores data for a temporary time. Similarly, all input/output devices have some buffer attached to them. This is because the input/output devices have slow processing or storing speed. The computer uses buffers to maintain the good timing speed of the processor and input/output devices.
- User Programs: This is the highest layer in the layered operating system. This layer deals with the many user programs and applications that run in an operating system, such as word processors, games, browsers, etc. You can also call this an application layer because it is concerned with application

- Following are some of important functions of an operating System.
- Memory Management
- Processor Management
- Device Management
- File Management
- Security
- Control over system performance
- Job accounting
- Error detecting aids
- Coordination between other software and users

What does an Operating system do?

- Process Management
- Process Synchronization
- Memory Management
- CPU Scheduling
- File Management
- Security

Memory Management

- Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.
- Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must in the main memory. An Operating System does the following activities for memory management −
- Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.
- In multiprogramming, the OS decides which process will get memory when and how much.
- Allocates the memory when a process requests it to do so.
- De-allocates the memory when a process no longer needs it or has been terminated.

Processor Management

- In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called **process scheduling**. An Operating System does the following activities for processor management –
- Keeps tracks of processor and status of process. The program responsible for this task is known as **traffic controller**.
- Allocates the processor (CPU) to a process.
- De-allocates processor when a process is no longer required.

Device Management

- An Operating System manages device communication via their respective drivers. It does the following activities for device management –
- Keeps tracks of all devices. Program responsible for this task is known as the **I/O controller**.
- Decides which process gets the device when and for how much time.
- Allocates the device in the efficient way.
- De-allocates devices.

File Management

- A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions.
- An Operating System does the following activities for file management –
- Keeps track of information, location, uses, status etc. The collective facilities are often known as **file system**.
- Decides who gets the resources.
- Allocates the resources.
- De-allocates the resources.

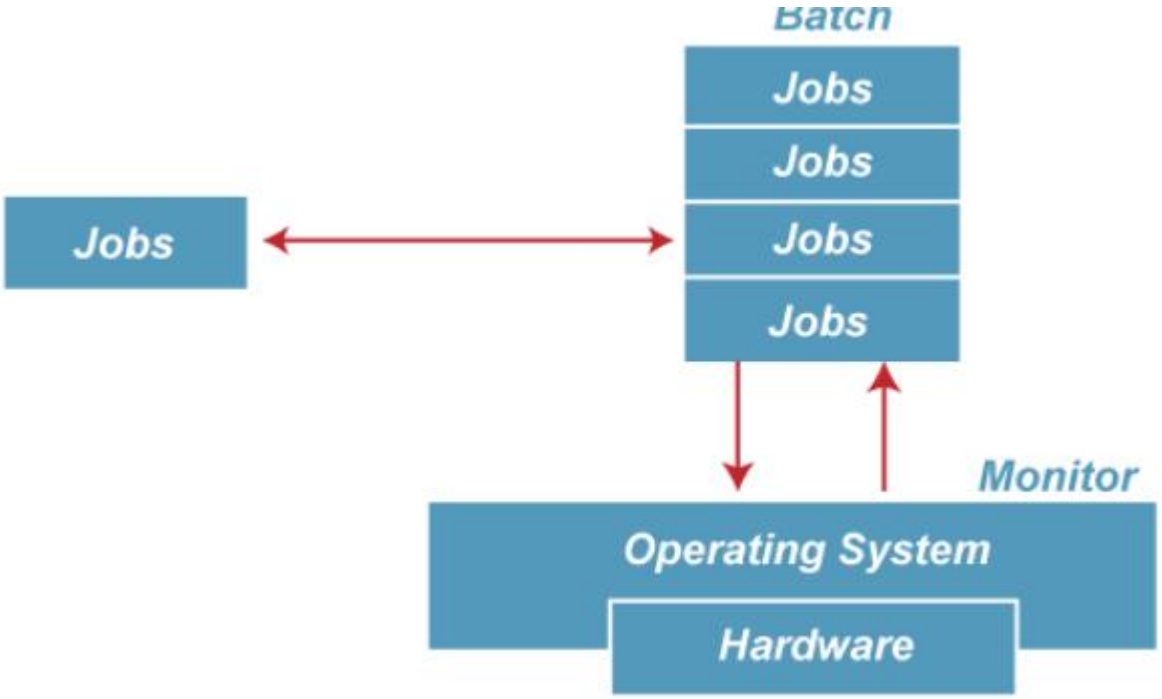
- Following are some of the important activities that an Operating System performs −
- **Security** − By means of password and similar other techniques, it prevents unauthorized access to programs and data.
- **Control over system performance** − Recording delays between request for a service and response from the system.
- **Job accounting** − Keeping track of time and resources used by various jobs and users.
- **Error detecting aids** − Production of dumps, traces, error messages, and other debugging and error detecting aids.
- **Coordination between other softwares and users** − Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

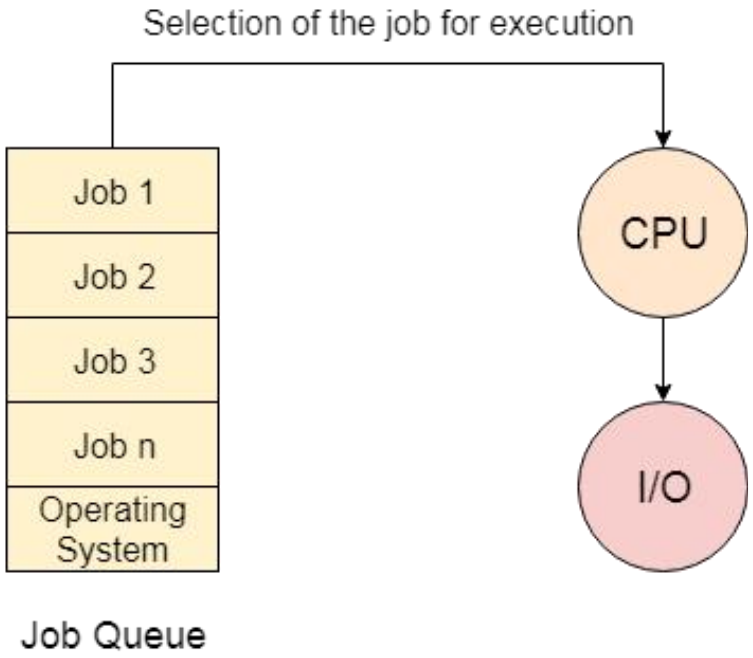
Types of Operating Systems

- Simple Batch System
- Multiprogramming Batch System
- Multiprocessor System
- *Time-Sharing Operating System*
- Distributed Operating System
- Clustered System
- Realtime Operating System

Batch operating system

- The users of a batch operating system do not interact with the computer directly. Each user prepares his job on an off-line device like punch cards and submits it to the computer operator. To speed up processing, jobs with similar needs are batched together and run as a group. The programmers leave their programs with the operator and the operator then sorts the programs with similar requirements into batches.
- The problems with Batch Systems are as follows –
- Lack of interaction between the user and the job.
- CPU is often idle, because the speed of the mechanical I/O devices is slower than the CPU.
- Difficult to provide the desired priority.


Batch
Jobs
Jobs
Jobs
Jobs
Jobs
Monitor
Operating System
Hardware



Disadvantages of Batch OS

1. Starvation

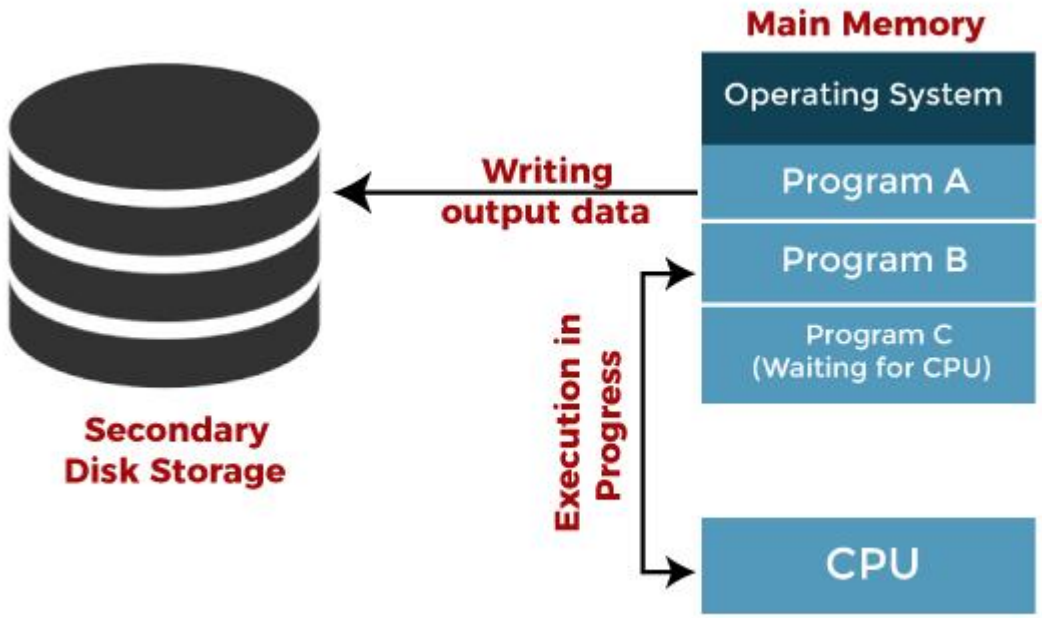
Batch processing suffers from starvation.

For Example:

- There are five jobs J1, J2, J3, J4 and J5 present in the batch. If the execution time of J1 is very high, then the other four jobs will never be executed, or they will have to wait for a very long time. Hence the other processes get starved.
- **2. Not Interactive**
- Batch Processing is not suitable for jobs that are dependent on the user's input. If a job requires the input of two numbers from the console, then it will never get it in the batch processing scenario since the user is not present at the time of execution.

Multiprogramming Operating System

- Multiprogramming is an extension to batch processing where the CPU is always kept busy. Each process needs two types of system time: CPU time and IO time.
- In a multiprogramming environment, when a process does its I/O, The CPU can start the execution of other processes. Therefore multiprogramming improves the efficiency of the system.



Jobs in multiprogramming system

- Advantages of Multiprogramming OS
- Throughout the system, it increased as the CPU always had one program to execute.
- Response time can also be reduced.
- Disadvantages of Multiprogramming OS
- Multiprogramming systems provide an environment in which various systems resources are used efficiently, but they do not provide any user interaction with the computer system.

multitasking operating system

The multitasking operating system is a logical extension of a multiprogramming system that enables **multiple** programs simultaneously. It allows a user to perform more than one computer task at the same time..

Advantages of Multitasking operating system

This operating system is more suited to supporting multiple users simultaneously.

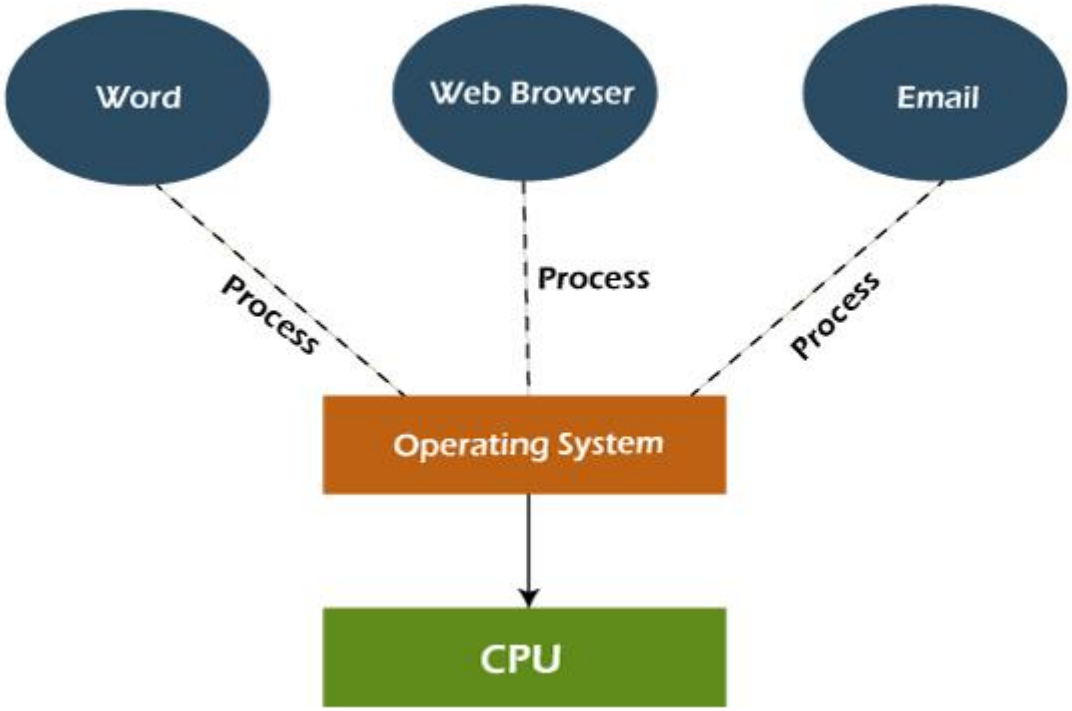
The multitasking operating systems have well-defined memory management.

Disadvantages of Multitasking operating system

The multiple processors are busier at the same time to complete any task in a multitasking environment, so the CPU generates more heat.

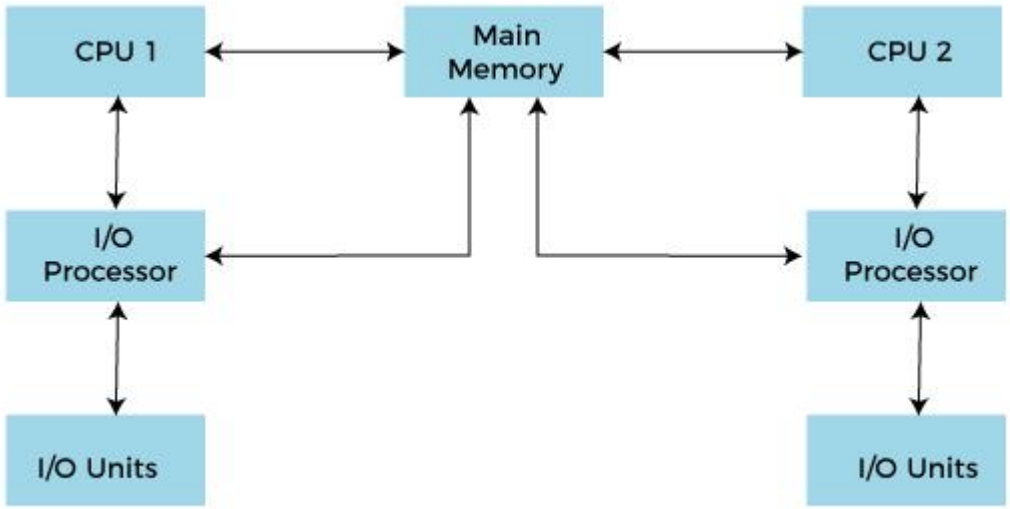
multitasking operating system

- The multitasking operating system is a logical extension of a multiprogramming system that enables **multiple** programs simultaneously. It allows a user to perform more than one computer task at the same time.


Word
Web Browser
Email
Process
Process
Process
Operating System
CPU

Multiprocessor Systems

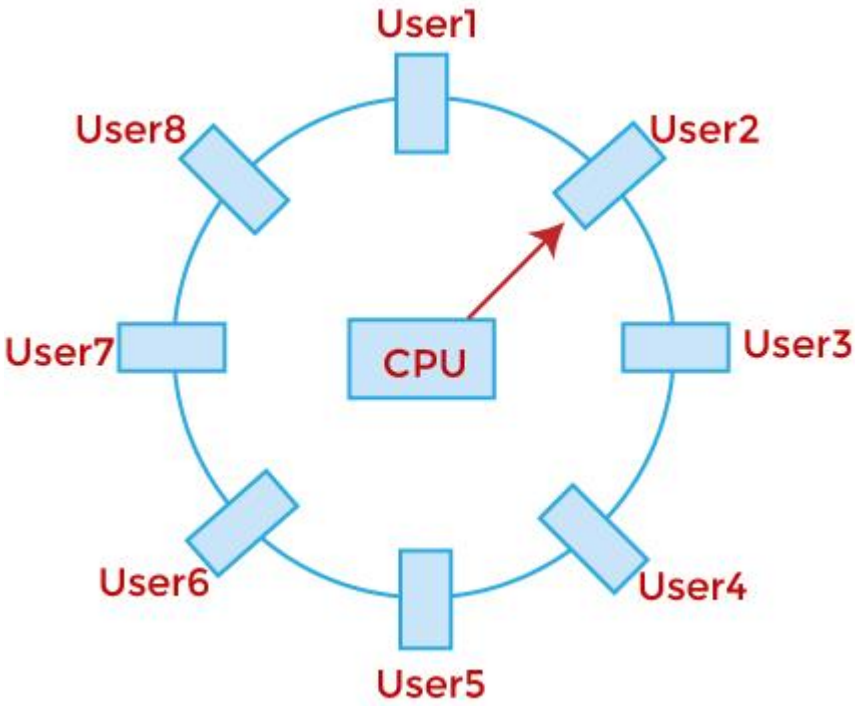
- In Multiprocessing, Parallel computing is achieved. There are more than one processors present in the system which can execute more than one process at the same time. This will increase the throughput of the system.



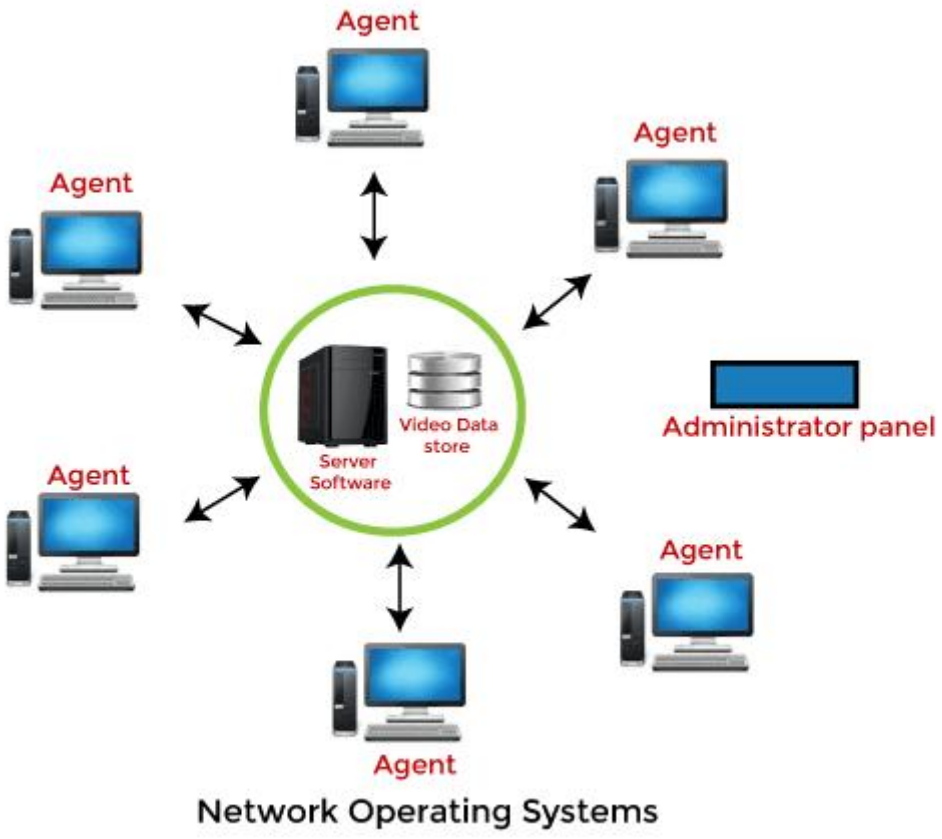
Working of Multiprocessor System

- Advantages of Multiprocessing operating system:
- **Increased reliability:** Due to the multiprocessing system, processing tasks can be distributed among several processors. This increases reliability as if one processor fails, the task can be given to another processor for completion.
- **Increased throughput:** As several processors increase, more work can be done in less.
- Disadvantages of Multiprocessing operating System
- Multiprocessing operating system is more complex and sophisticated as it takes care of multiple CPUs simultaneously.

- Time-Sharing Operating System
- In the Time Sharing operating system, computer resources are allocated in a time-dependent fashion to several programs simultaneously. Thus it helps to provide a large number of user's direct access to the main computer. It is a logical extension of multiprogramming. In time-sharing, the CPU is switched among multiple programs given by different users on a


User1
User2
User3
User4
User5
User6
User7
User8
CPU

Network Operating System



- An Operating system which includes software and associated protocols to communicate with other computers via a network conveniently and cost-effectively is called Network Operating System.
- Advantages of Network Operating System
- In this type of operating system, network traffic reduces due to the division between clients and the server.
- This type of system is less expensive to set up and maintain.
- Disadvantages of Network Operating System
- In this type of operating system, the failure of any node in a system affects the whole system.
- Security and performance are important issues. So trained network administrators are required for network administration

Real Time Operating System

- In Real-Time Systems, each job carries a certain deadline within which the job is supposed to be completed, otherwise, the huge loss will be there, or even if the result is produced, it will be completely useless.
- *The Application of a Real-Time system exists in the case of military applications, if you want to drop a missile, then the missile is supposed to be dropped with a certain precision.*
- Examples of the real-time operating systems: **Airline traffic control systems, Command Control Systems**, Airlines reservation system, Robot etc

Advantages of Real-time operating system:

Easy to layout, develop and execute real-time applications under the real-time operating system.

In a Real-time operating system, the maximum utilization of devices and systems.

Disadvantages of Real-time operating system:

Real-time operating systems are very costly to develop.

Real-time operating systems are very

Distributed Operating System

- The Distributed Operating system is not installed on a single machine, it is divided into parts, and these parts are loaded on different machines. A part of the distributed Operating system is installed on each machine to make their communication possible. Distributed Operating systems are much more complex, large, and sophisticated than Network operating systems because they also have to take care of varying networking protocols.
-

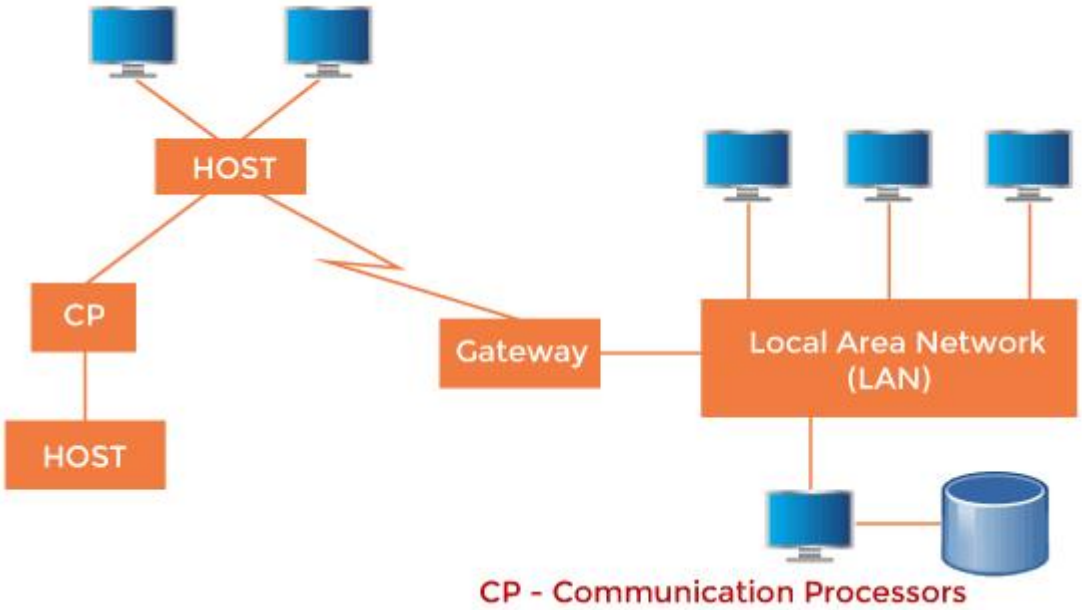
Advantages of Distributed Operating System

The distributed operating system provides sharing of resources.

This type of system is fault-tolerant.

Disadvantages of Distributed Operating System

Protocol overhead can dominate computation cost.

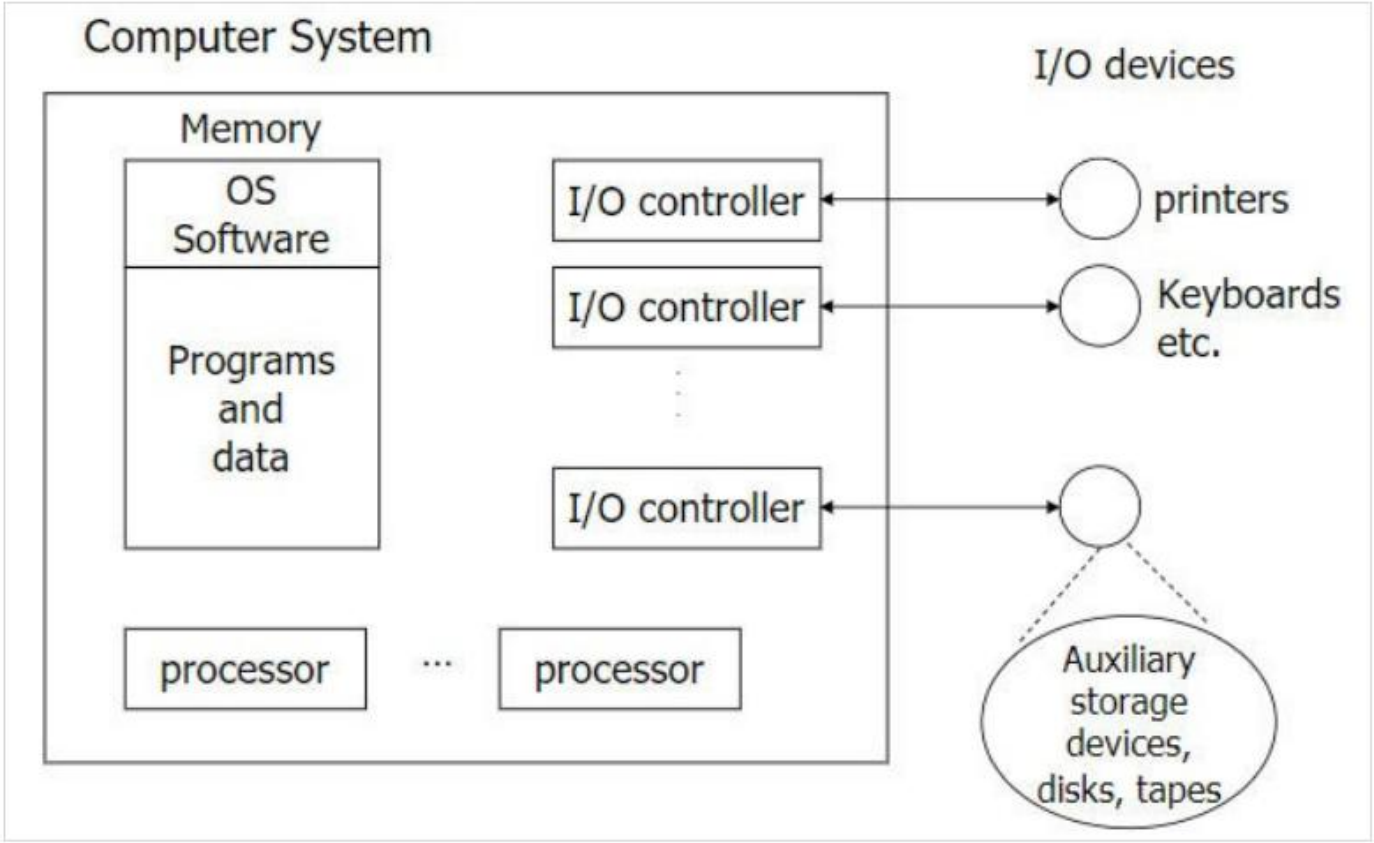


CP - Communication Processors

Operating System as Resource Manager

- Now-a-days all modern computers consist of processors, memories, timers, network interfaces, printers, and so many other devices.
- The operating system provides for an orderly and controlled allocation of the processors, memories, and I/O devices among the various programs in the bottom-up view.
- Operating system allows multiple programs to be in memory and run at the same time.
- Resource management includes multiplexing or sharing resources in two different ways: in time and in space.
- In time multiplexed, different programs take a chance of using CPU. First one tries to use the resource, then the next one that is ready in the queue and so on. For example: Sharing the printer one after another.
- In space multiplexing, Instead of the customers taking a chance, each one gets part of the resource. For example − Main memory is divided into several running programs, so each one can be resident at the same time.

The diagram given below shows the functioning of OS as a resource manager −

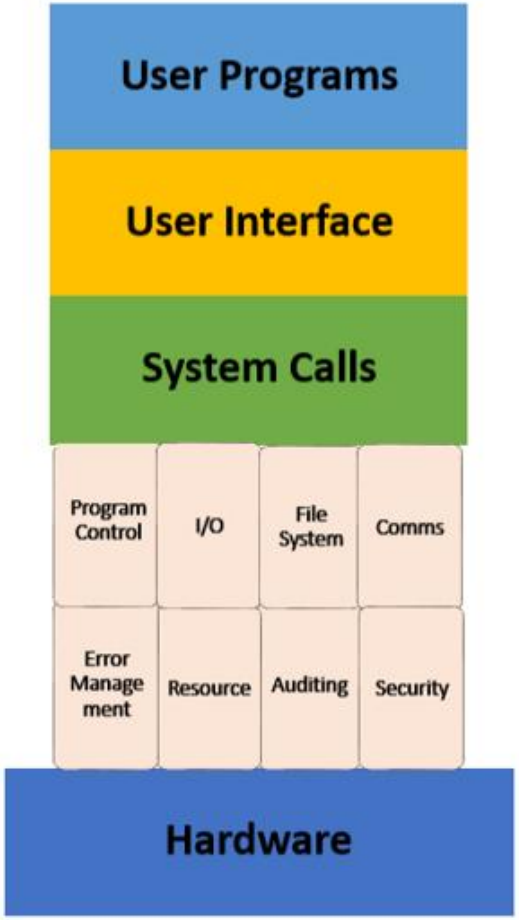


- With the help of system call, we can access the services of os.

system call

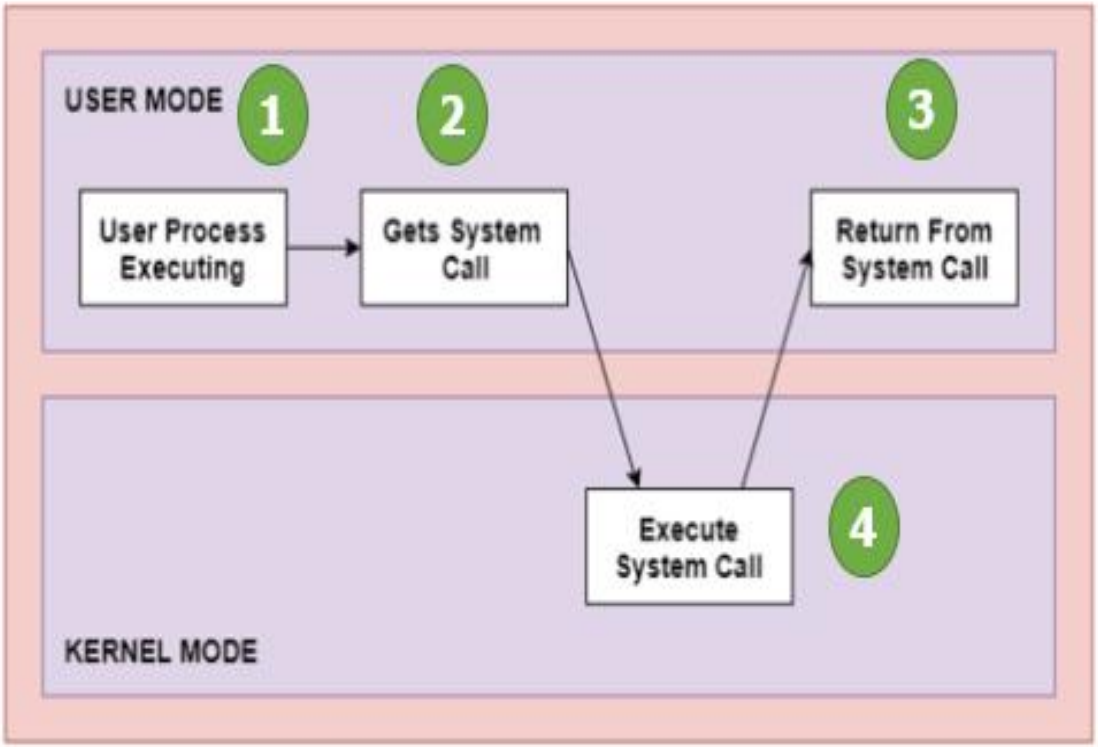
- **What is System Call in Operating System?**
- A **system call** is a mechanism that provides the interface between a process and the operating system. It is a programmatic method in which a computer program requests a service from the kernel of the OS.
- System call offers the services of the operating system to the user programs via API (Application Programming Interface). System calls are the only entry points for the kernel system.

System call in OS



How System Call Works?

Here are the steps for System Call in OS:



As you can see in the above-given System Call example diagram.
Step 1) The processes executed in the user mode till the time a system call interrupts it.
Step 2) After that, the system call is executed in the kernel-mode on a priority basis.
Step 3) Once system call execution is over, control returns to the user mode.,
Step 4) The execution of user processes resumed in Kernel mode.

- **Why do you need System Calls in OS?**
- Following are situations which need system calls in OS:
- Reading and writing from files demand system calls.
- If a file system wants to create or delete files, system calls are required.
- System calls are used for the creation and management of new processes.
- Network connections need system calls for sending and receiving packets.
- Access to hardware devices like scanner, printer, need a system call.
-

Types of System calls

Here are the five types of System Calls in OS:

Process Control

File Management

Device Management

Information Maintenance

Communications

- **Process Control**
- This system calls perform the task of process creation, process termination, etc.
- **Functions:**
- End and Abort
- Load and Execute
- Create Process and Terminate Process
- Wait and Signal Event
- Allocate and free memory

- **File Management**
- File management system calls handle file manipulation jobs like creating a file, reading, and writing, etc.
- **Functions:**
- Create a file
- Delete file
- Open and close file
- Read, write, and reposition
- Get and set file attributes

- **Device Management**
- Device management does the job of device manipulation like reading from device buffers, writing into device buffers, etc.
- **Functions:**
- Request and release device
- Logically attach/ detach devices
- Get and Set device attributes

- **Information Maintenance**
- It handles information and its transfer between the OS and the user program.
- **Functions:**
- Get or set time and date
- Get process and device attributes

- **Communication:**
- These types of system calls are specially used for interprocess communications.
- **Functions:**
- Create, delete communications connections
- Send, receive message
- Help OS to transfer status information
- Attach or detach remote devices

Important System Calls Used in OS

- **wait()**
- In some systems, a process needs to wait for another process to complete its execution
- **fork()**
- Processes use this system call to create processes that are a copy of themselves. With the help of this system Call parent process creates a child process,

- **kill():**
- The kill() system call is used by OS to send a termination signal to a process that urges the process to exit.
- **exec()**
- This system call runs when an executable file in the context of an already running process that replaces the older executable file.
- **exit():**
- The exit() system call is used to terminate program execution

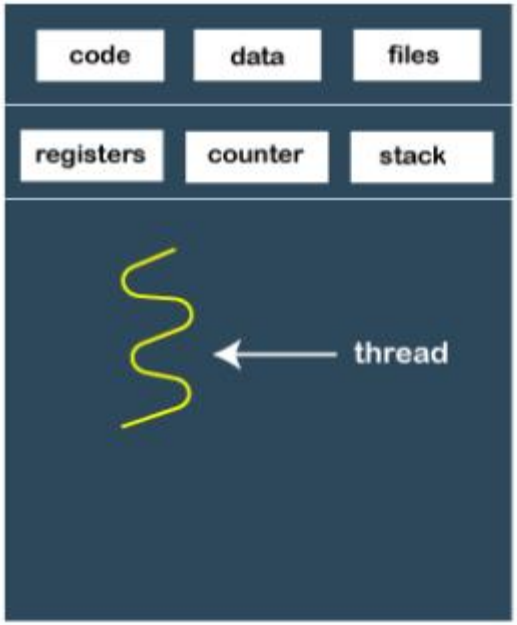
What is Process?

- **A process is an instance of a program that is being executed.** When we run a program, it does not execute directly. It takes some time to follow all the steps required to execute the program, and following these execution steps is known as a process.
- A process can create other processes to perform multiple tasks at a time; the created processes are known as **clone or child process**, and the main process is known as the **parent process**. Each process contains its own memory space and does not share it with the other processes. It is known as the active entity.
-

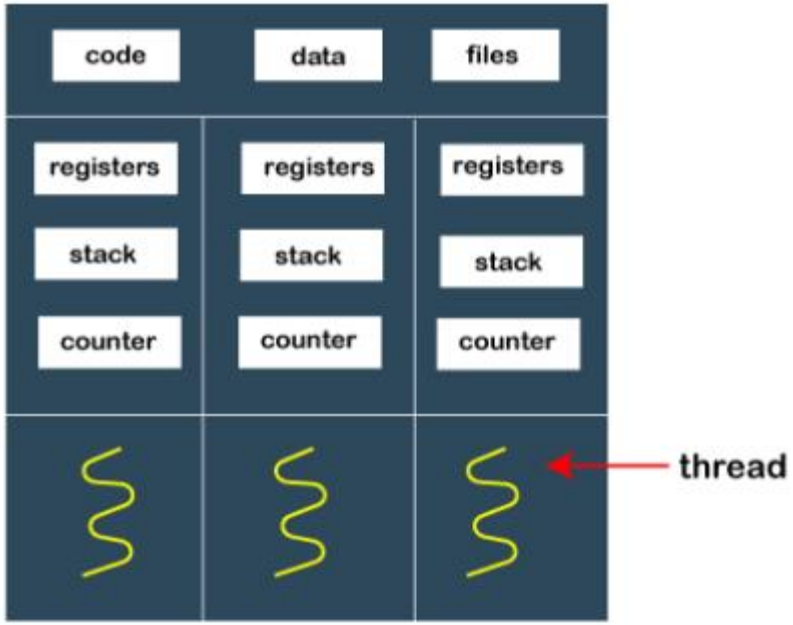
- A process in OS can remain in any of the following states:
- **NEW**: A new process is being created.
- **READY**: A process is ready and waiting to be allocated to a processor.
- **RUNNING**: The program is being executed.
- **WAITING**: Waiting for some event to happen or occur.
- **TERMINATED**: Execution finished.

- Features of Process
- Each time we create a process, we need to make a separate system call for each process to the OS. The **fork**() function creates the process.
- Each process exists within its own address or memory space.
- Each process is independent and treated as an isolated process by the OS.
- Processes need IPC (Inter-process Communication) in order to communicate with each other.

- What is Thread?
- A thread is the subset of a process and is also known as the lightweight process. A process can have more than one thread, and these threads are managed independently by the scheduler. All the threads within one process are interrelated to each other. Threads have some common information, such as **data segment, code segment, files, etc.,** that is shared to their peer threads. But contains its own registers, stack, and counter.



Single-threaded process



Multi-threaded process

Types of Threads

- There are two types of threads, which are:
- **1. User Level Thread**
- As the name suggests, the user-level threads are only managed by users, and the kernel does not have its information.
- These are faster, easy to create and manage.

- **2. Kernel-Level Thread**
- The kernel-level threads are handled by the Operating system and managed by its kernel. These threads are slower than user-level threads because context information is managed by the kernel. To create and implement a kernel-level thread, we need to make a system call.

- Features of Thread
- Threads share data, memory, resources, files, etc., with their peer threads within a process.
- One system call is capable of creating more than one thread.
- Each thread has its own stack and register.
- Threads can directly communicate with each other as they share the same address space.

Process Vs Thread

Parameter	Process	Thread
Definition	Process means a program is in execution.	Thread means a segment of a process.
Lightweight	The process is not Lightweight.	Threads are Lightweight.
Termination time	The process takes more time to terminate.	The thread takes less time to terminate.
Creation time	It takes more time for creation.	It takes less time for creation.
Communication	Communication between processes needs more time compared to thread.	Communication between threads requires less time compared to processes.
Context switching time	It takes more time for context switching.	It takes less time for context switching.
Resource	Process consume more resources.	Thread consume fewer resources.
Treatment by OS	Different process are tread separately by OS.	All the level peer threads are treated as a single task by OS.

Memory	The process is mostly isolated.	Threads share memory.
Sharing	It does not share data	Threads share data with each other.

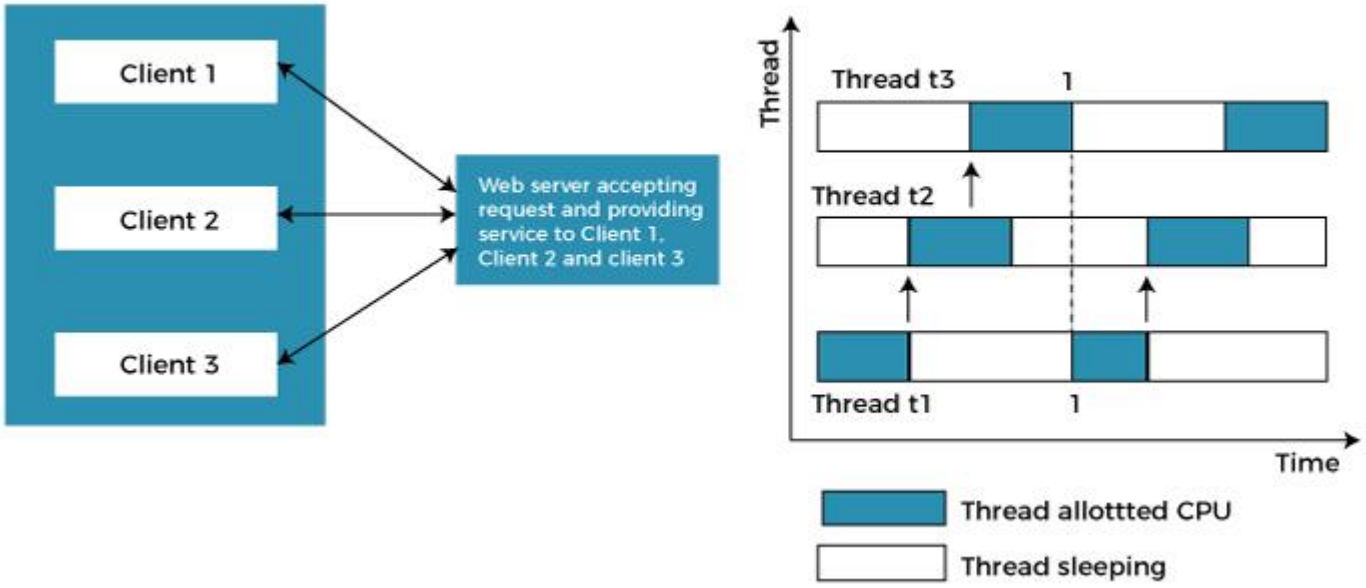
Multithreading Model:

Multithreading allows the application to divide its task into individual threads. In multi-threads, the same process or task can be done bu the number of threads, or we can say that there is more than one thread to perform the task in multithreading. With the use of multithreading, multitasking can be achieved.

One Process
Multiple Threads

Multiple Process Multiple
Threads per Process

- The main drawback of sinale threadina sustems is that onlu one task can be performed at a time. so to overcome the drawback of this sinale threadina. there is multithreadina that allows multiple tasks to be performed.


Client 1
Client 2
Client 3
Web server accepting request and providing service to Client 1, Client 2 and client 3
Thread
Thread t3
1
Thread t2
Thread t1
1
Time
Thread allottted CPU
Thread sleeping

- In the above example, client1, client2, and client3 are accessing the web server without any waiting. In multithreading, several tasks can run at the same time.

BIOS

- The BIOS (Basic Input/Output System) is firmware stored on a computer's motherboard that initializes hardware during the boot process and provides runtime services for operating systems and programs. Here's a breakdown of its key functions and relevance to operating systems (OS):
- **Key Functions of BIOS:**

1. **POST (Power-On Self-Test):**
    1. Checks hardware components like RAM, CPU, and storage devices to ensure they are functioning properly before booting the OS.
2. **Boot Loader:**
    1. Locates and loads the operating system into memory. It typically looks for bootable devices (like HDD, SSD, USB) according to the configured boot order.
3. **Hardware Initialization:**
    1. Configures and initializes hardware components, such as keyboard, mouse, and storage devices, allowing the OS to interact with them.
4. **Runtime Services:**
    1. Provides low-level hardware control and services that operating systems can call during runtime, especially in legacy systems.
5. **CMOS Setup:**
    1. Stores BIOS settings and system configuration data in CMOS memory, allowing users to customize hardware settings (like boot sequence, CPU settings, etc.).

Bootstrap Loader

- A **Bootstrap Loader** (often simply called a "boot loader") is a crucial piece of software that initializes the operating system during the startup process of a computer. Here's a breakdown of its functions and importance:
- **Functions of a Bootstrap Loader:**

1. **Initial Hardware Setup:**
    1. After the BIOS or UEFI firmware performs its Power-On Self-Test (POST) and initializes hardware, the boot loader takes over to start the operating system.
2. **Loading the OS:**
    1. The boot loader locates the operating system kernel (the core component of the OS) on the designated storage device (like a hard drive or SSD) and loads it into memory.
3. **Execution Transfer:**
    1. Once the kernel is loaded into memory, the boot loader transfers control to the operating system, allowing it to begin its own initialization processes.
4. **Multiple OS Support:**
    1. Many boot loaders can handle multiple operating systems installed on the same machine, allowing users to select which OS to boot from at startup.