Q What is Operating System.

⇒ It is a system software which is used to connect a user with the Hardware. It is also manage the computer's memory and process

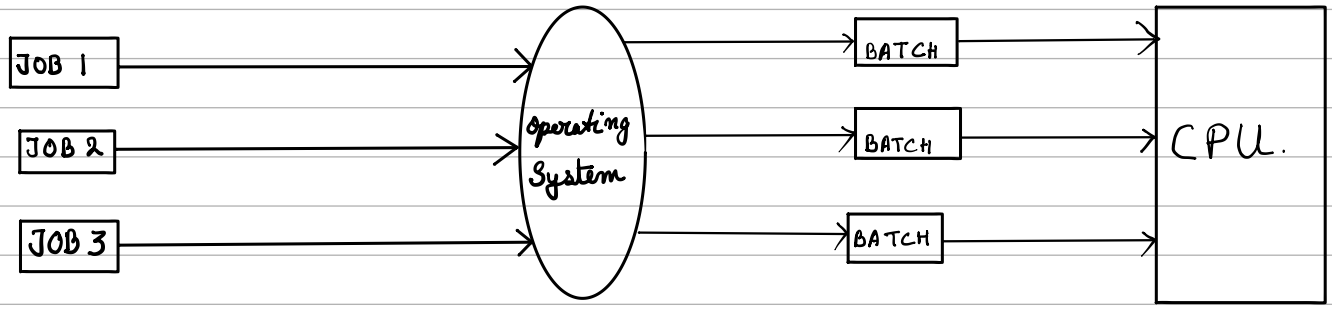
Q What are the different type of OS?

⇒ There are 8 types of operating System.

Batch Operating System
Multi-Programming System
Multi-Processing System
Multi-Tasking Operating System
Time-Sharing Operating System
Distributed Operating System
Network Operating System
Real-Time Operating System

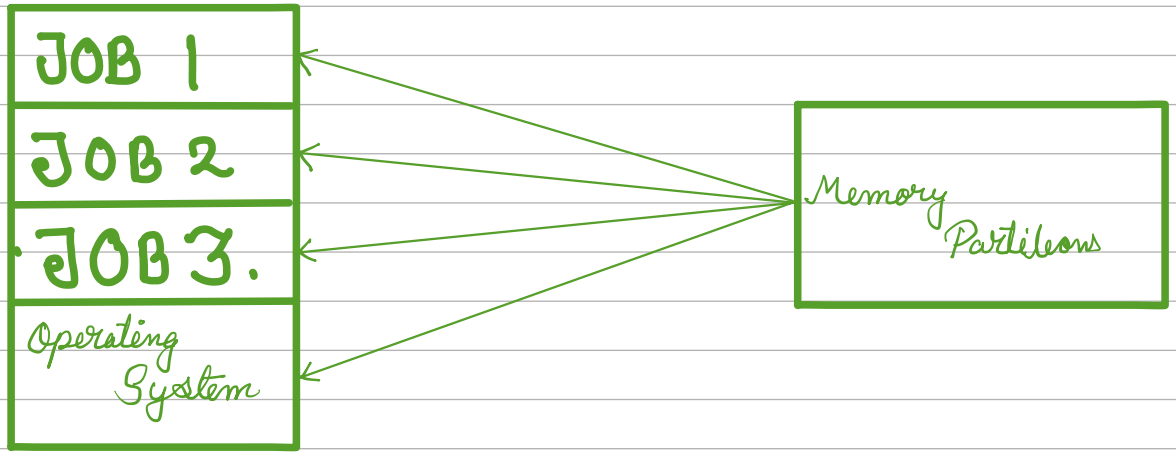
Batch Operating System :-

This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having
the same requirement and groups them into batches. It is the responsibility of the operator to sort jobs with similar needs



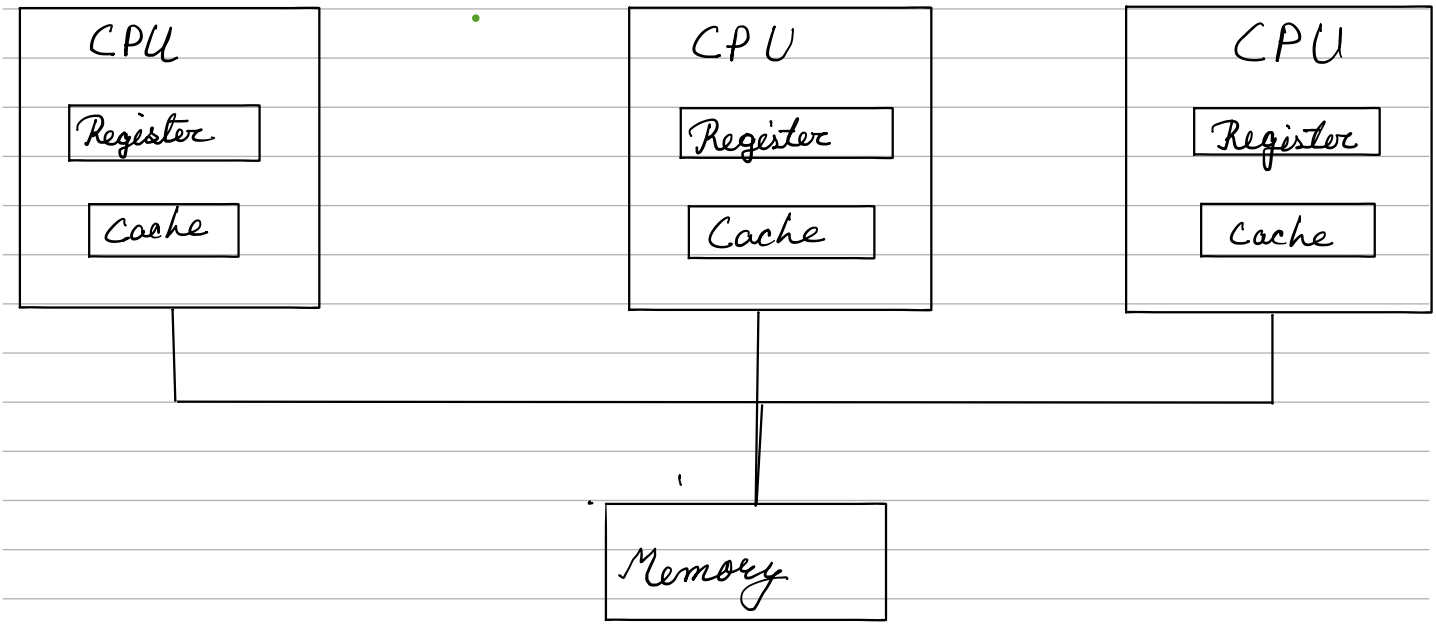
Multi-Programming Operating System :-

Multiprogramming Operating Systems can be simply illustrated as more than one program is present in the main memory and any one of them can be kept in execution. This is basically used for better execution of resources.



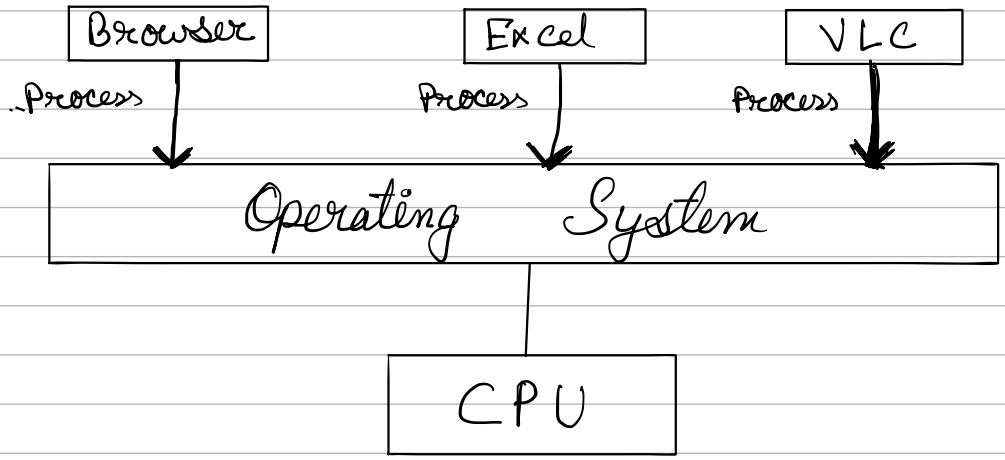
Multi-Processing Operating System :-

Multi-Processing Operating System is a type of Operating System in which more than one CPU is used for the execution of resources. It betters the throughput of the System.



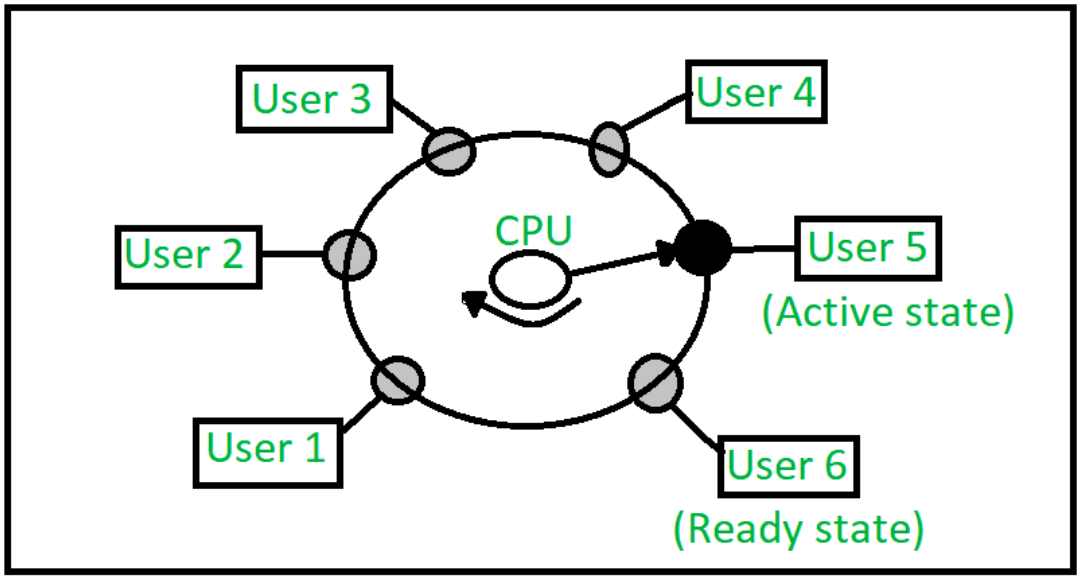
Multi-Tasking Operating System :-

Multitasking Operating System is simply a multiprogramming Operating System with having facility of a Round-Robin Scheduling Algorithm. It can run multiple programs simultaneously.



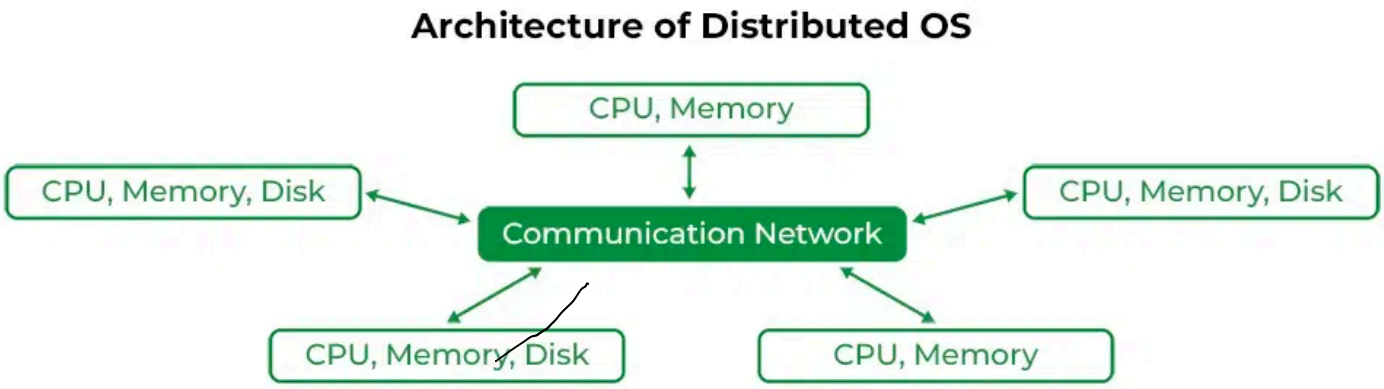
Time-Sharing Operating System :-

Each task is given some time to execute so that all the tasks work smoothly. Each user gets the time of the CPU as they use a single system. These systems are also known as Multitasking Systems. The task can be from a single user or different users also. The time that each task gets to execute is called quantum. After this time interval is over OS switches over to the next task.



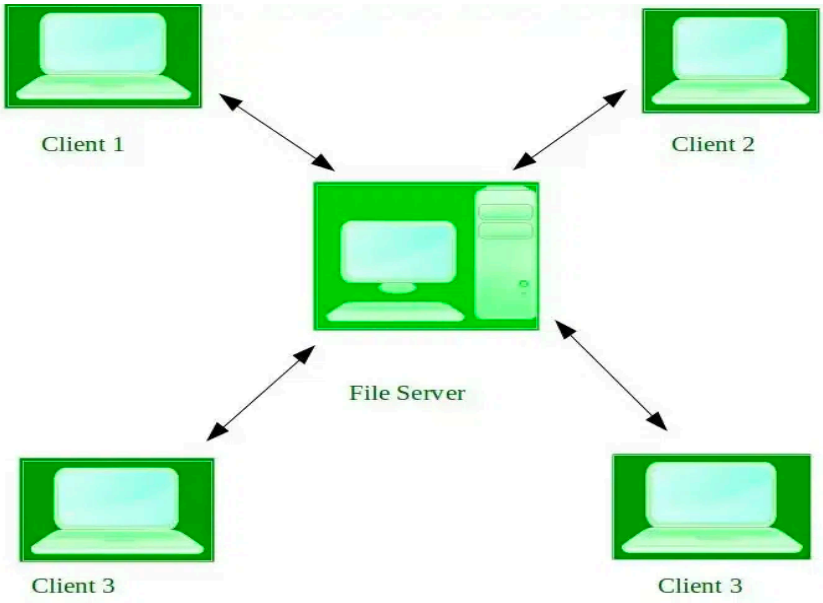
Distributed operating System :-

These systems' processors differ in size and function. The major benefit of working with these types of the operating system is that it is always possible that one user can access the files or software which are not actually present on his system but some other system connected within this network i.e., remote access is enabled within the devices connected in that network.



Network Operating System :-

These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions. These types of operating systems allow shared access to files, printers, security, applications, and other networking functions over a small private network

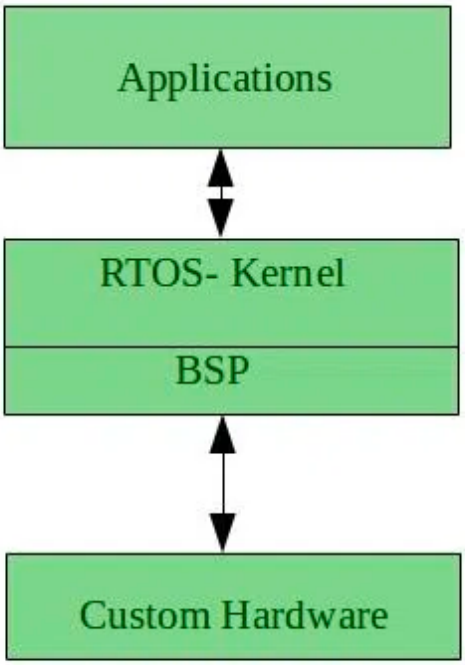


Real-Time Operating System

These types of OSs serve real-time systems. The time interval required to process and respond to inputs is very small. This time interval is called response time.
Real-time systems are used when there are time requirements that are very strict like missile systems, air traffic control systems, robots, etc.

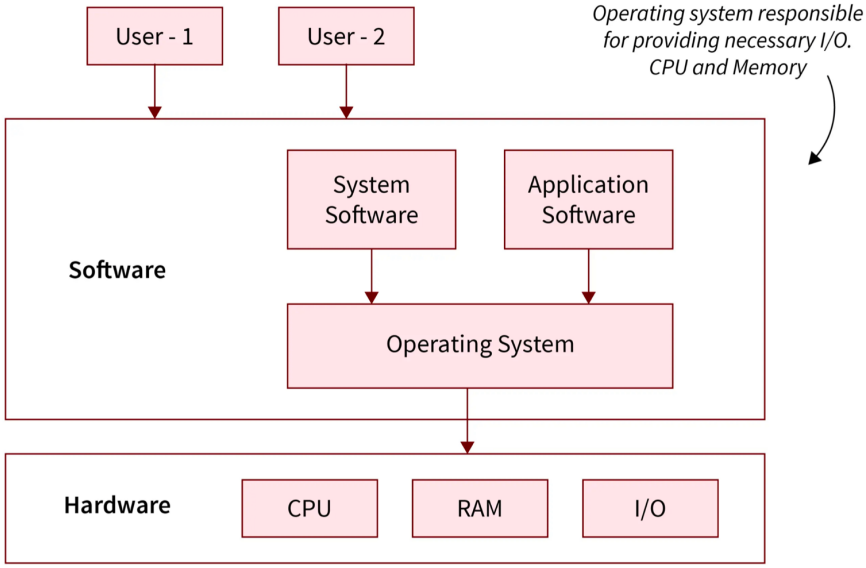
Types of Real-Time Operating Systems

Hard Real-Time Systems:
Hard Real-Time OSs are meant for applications where time constraints are very strict and even the shortest possible delay is not acceptable. These systems are built for saving life like automatic parachutes or airbags which are required to be readily available in case of an accident. Virtual memory is rarely found in these systems.
Soft Real-Time Systems:
These OSs are for applications where time-constraint is less strict.



3) Explain the OS Architecture Diagram.

An Operating System is characterized as a software package that functions as a bridge connecting the user and the hardware components. Its role is to simplify user interactions with intricate hardware systems. The primary objective of an Operating System is to establish a conducive platform for the streamlined execution of programs. It undertakes significant responsibilities, including Resource Management, Process Management, Memory Management, Security, and File Management, to enhance the efficiency of operations.



5) What is process? Different between process and thread.

1) What is Process?

Ans) A process is basically a program in execution. The execution of a process must progress in sequential fashion.

A process is define as an entity which represent the basic unit of work to be implemet in the system.

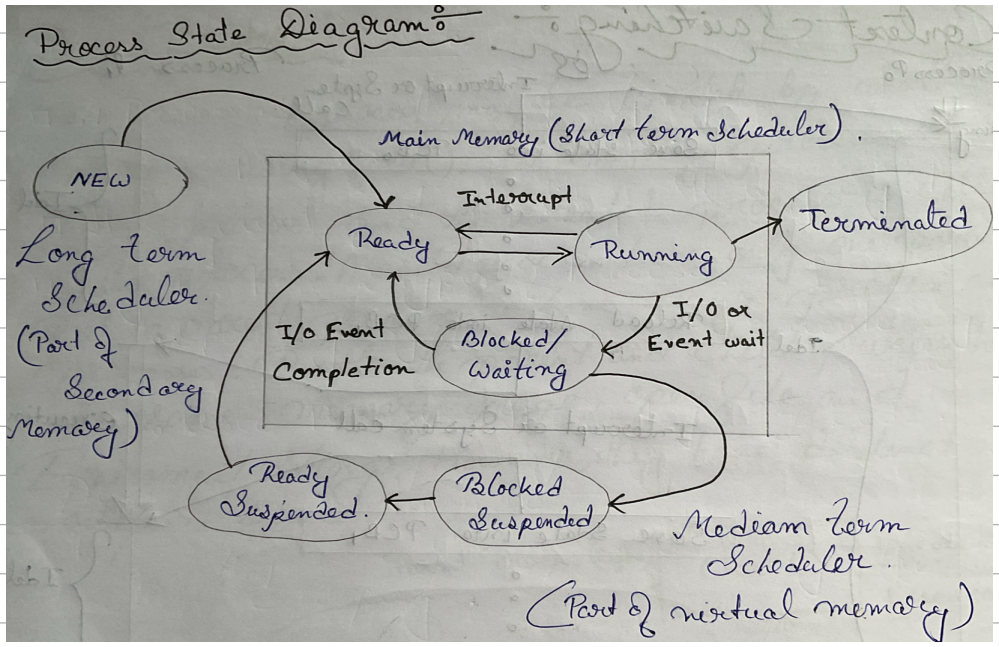
To put it in a simple terms, we write our Computer programs in a text file and when we execute this program, it become a process which performs all the task mention in the program.

5) Difference between process and thread.

S.N	PROCESS	THREAD
1	Process is a heavy weight or resource intensive	Thread is light weight, taking lesser resources than a process.
2.	Process switching needs interaction with operation System.	Thread switching does not need to interact with operation System.
3.	In multiple processing environment, each process executes the same code but has its own memory and file resources.	All threads can share same set of open files, child processes.
4.	If one process is blocked, then no other process can execute until the first process is unblocked.	While one thread is blocked and waiting, a second thread in the same task can run.
5	Multiple processes without using threads use more resources.	Multiple threaded processes use fewer resource.

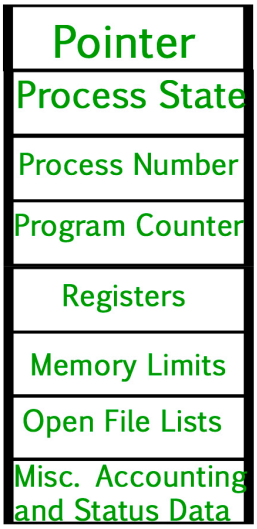
6) Explain process state Diagram.

A process state diagram shows the transitions between the different states of a process. It's a key concept in process management for operating systems.



7) What is PCB? State it's Content. (Diagram)

Process Control Block (PCB) is an important data structure used by the Operating System (OS) to manage and control the execution of processes. It is also known as Task Control Block (TCB) in some operating systems. A PCB contains all the necessary information about a process, including its process state, program counter, memory allocation, open files, and CPU scheduling information.



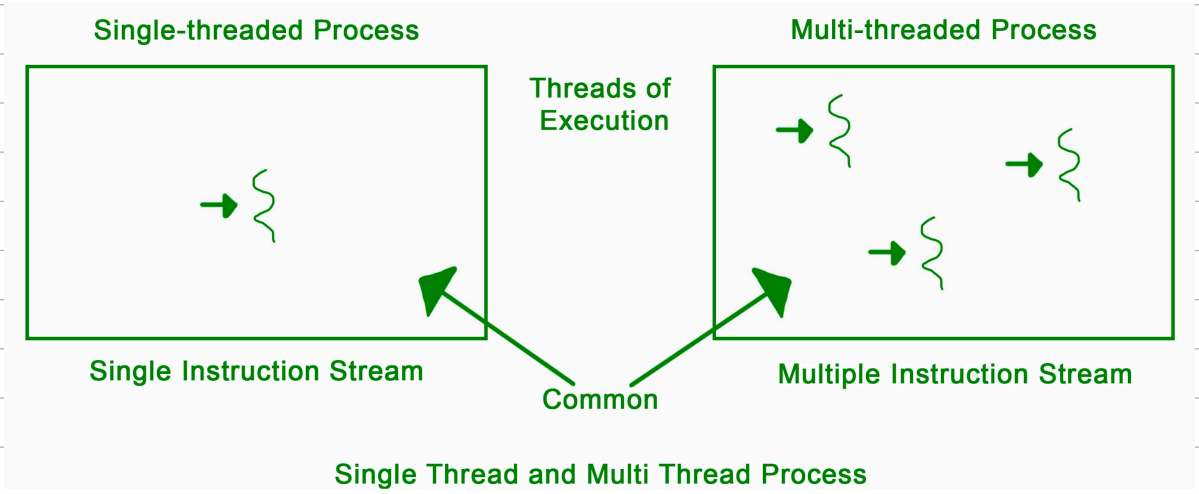
Process Control Block

9) what is the different between User level thread and kernel level thread.

Features	User Level Threads	Kernel Level Threads
Implemented by	It is implemented by the users. It is implemented by the OS.	
Context switch time	Its time is less.	Its time is more.
Multithreading	Multithread applications are unable to employ multiprocessing in user-level threads.	It may be multithreaded.
Implementation	It is easy to implement.	It is complicated to implement.
Blocking Operation	If a thread in the kernel is blocked, it blocks all other threads in the same process.	If a thread in the kernel is blocked, it does not block all other threads in the same process.
Recognize	OS doesn't recognize it.	It is recognized by OS.
Thread Management	Its library includes the source code for thread creation, data transfer, thread destruction, message passing, and thread scheduling.	The application code on kernel-level threads does not include thread management code, and it is simply an API to the kernel mode.
Hardware Support	It doesn't need hardware support.	It requires hardware support.
Creation and Management	It may be created and managed much faster.	It takes much time to create and handle.
Examples	Some instances of user-level threads are Java threads and POSIX threads.	Some instances of Kernel-level threads are Windows and Solaris.
Operating System	Any OS may support it.	The specific OS may support it.

10) Explain the concept of multi threading model.

A thread is a path which is followed during a program's execution. Majority of programs written now a days run as a single thread.Lets say, for example a program is not capable of reading keystrokes while making drawings. These tasks cannot be executed by the program at the same time. This problem can be solved through multitasking so that two or more tasks can be executed simultaneously. Multitasking is of two types: Processor based and thread based. Processor based multitasking is totally managed by the OS, however multitasking through multithreading can be controlled by the programmer to some extent. The concept of multi-threading needs proper understanding of these two terms – a process and a thread. A process is a program being executed. A process can be further divided into independent units known as threads. A thread is like a small light-weight process within a process. Or we can say a collection of threads is what is known as a process.



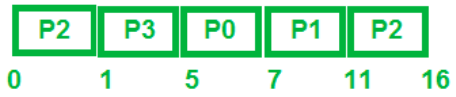
Single Thread and Multi Thread Process

11) Explain primitive and non primitive CPU scheduling.

Preemitive Scheduling :-

Preemptive scheduling is used when a process switches from the running state to the ready state or from the waiting state to the ready state. The resources (mainly CPU cycles) are allocated to the process for a limited amount of time and then taken away, and the process is again placed back in the ready queue if that process still has CPU burst time remaining. That process stays in the ready queue till it gets its next chance to execute.

Process	Arrival Time	CPU Burst Time (in millisec.)
P0	3	2
P1	2	4
P2	0	6
P3	1	4

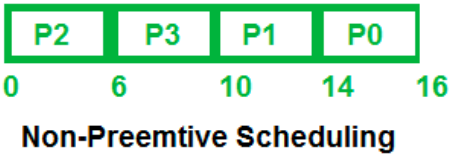


Preemptive Scheduling

Non-Preemitive Scheduling :-

Non-preemptive Scheduling is used when a process terminates, or a process switches from running to the waiting state. In this scheduling, once the resources (CPU cycles) are allocated to a process, the process holds the CPU till it gets terminated or reaches a waiting state. In the case of non-preemptive scheduling does not interrupt a process running CPU in the middle of the execution. Instead, it waits till the process completes its CPU burst time, and then it can allocate the CPU to another process.

Process	Arrival Time	CPU Burst Time (in millisec.)
P0	3	2
P1	2	4
P2	0	6
P3	1	4



Non-Preemtive Scheduling

12) What is CPU utilisation, through put, turn around time, respone time, waiting time.

CPU utilization

The main objective of any CPU scheduling algorithm is to keep the CPU as busy as possible. Theoretically, CPU utilization can range from 0 to 100 but in a real-time system, it varies from 40 to 90 percent depending on the load upon the system.

Throughput

A measure of the work done by the CPU is the number of processes being executed and completed per unit of time. This is called throughput. The throughput may vary depending on the length or duration of the processes.

Turnaround Time

For a particular process, an important criterion is how long it takes to execute that process. The time elapsed from the time of submission of a process to the time of completion is known as the turnaround time. Turn-around time is the sum of times spent waiting to get into memory, waiting in the ready queue, executing in CPU, and waiting for I/O.

Turn Around Time = Completion Time - Arrival Time.

Waiting Time

A scheduling algorithm does not affect the time required to complete the process once it starts execution. It only affects the waiting time of a process i.e. time spent by a process waiting in the ready queue.

Waiting Time = Turnaround Time - Burst Time.

Response Time

In an interactive system, turn-around time is not the best criterion. A process may produce some output fairly early and continue computing new results while previous results are being output to the user. Thus another criterion is the time taken from submission of the process of the request until the first response is produced. This measure is called response time.

Response Time = CPU Allocation Time(when the CPU was allocated for the first) - Arrival Time

13) What is Starvation?

Starvation occurs when a process is unable to acquire the necessary resources required to complete its task. Starvation is also known as indefinite blocking. It is a problem often associated with Priority Scheduling Algorithms. It occurs when a process that is ready for CPU resources is unable to run because of its low priority, resulting in an indefinite wait time. This can lead to poor performance and prolonged wait times for processes and reduce the system throughput.

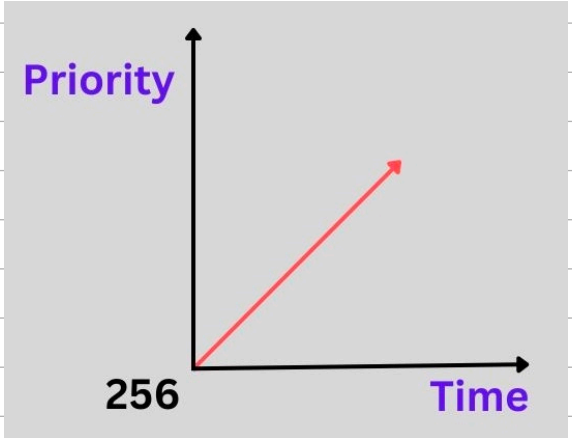
14) What is Aging?

Aging in OS is a scheduling technique used to prevent starvation in operating systems. It involves gradually increasing the priority of processes that have been waiting for a long time. It increases the chance of them getting the necessary resources to execute. Thus it reduces the risk of starvation.

Example

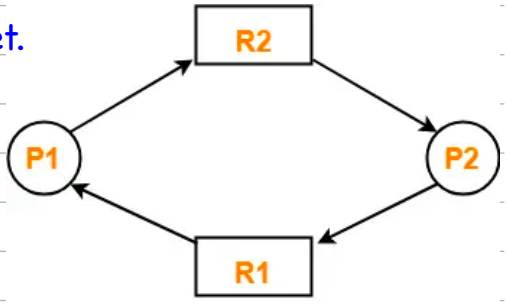
Aging is a technique used in operating systems to increase the priority of waiting processes. When a process with low priority waits in the system for an extended period, its priority is increased over time. It ensures that the process does not indefinitely starve of resources.

Let's look at the example:



17) What is Deadlock? State it's necessary condition.

A deadlock in OS is a situation in which more than one process is blocked because it is holding a resource and also requires some resource that is acquired by some other process. The four necessary conditions for a deadlock situation are mutual exclusion, no preemption, hold and wait and circular set.



Example of a deadlock

Conditions For Deadlock-

There are following 4 necessary conditions for the occurrence of deadlock-

Mutual Exclusion

Hold and Wait

No preemption

Circular wait

A. Mutual Exclusion- There must exist at least one resource in the system which can be used by only one process at a time.

If there exists no such resource, then deadlock will never occur.

Printer is an example of a resource that can be used by only one process at a time.

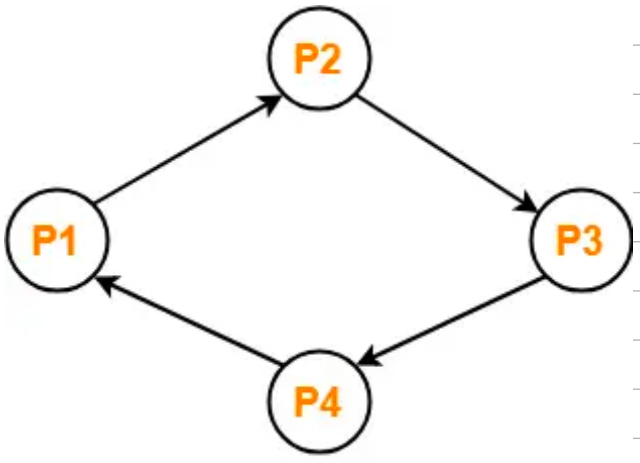
B. Hold and Wait- There must exist a process which holds some resource and waits for another resource held by some other process.

C. No Preemption- Once the resource has been allocated to the process, it can not be preempted.

It means resource can not be snatched forcefully from one process and given to the other process.

The process must release the resource voluntarily by itself.

D. Circular Wait- All the processes must wait for the resource in a cyclic manner where the last process waits for the resource held by the first process.

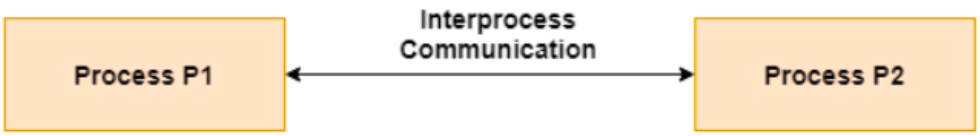


Circular Wait

i. Process P1 waits for a resource held by process P2.
ii. Process P2 waits for a resource held by process P3.
iii. Process P3 waits for a resource held by process P4.
iv. Process P4 waits for a resource held by process P1.

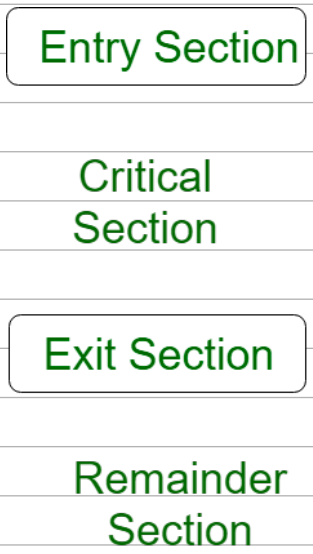
18> What is IPC ?

Interprocess communication is the mechanism provided by the operating system that allows processes to communicate with each other. This communication could involve a process letting another process know that some event has occurred or the transferring of data from one process to another.



19> What is critical section?

The critical section refers to the segment of code where processes access shared resources, such as common variables and files, and perform write operations on them. Since processes execute concurrently, any process can be interrupted mid-execution.



20> What is race Condition?

A race condition is a situation that may occur inside a critical section. This happens when the result of multiple thread execution in critical section differs according to the order in which the threads execute.

Race conditions in critical sections can be avoided if the critical section is treated as an atomic instruction. Also, proper thread synchronization using locks or atomic variables can prevent race conditions

2) What are the necessary condition of process Synchronization?

When two or more process cooperates with each other, their order of execution must be preserved otherwise there can be conflicts in their execution and inappropriate outputs can be produced.

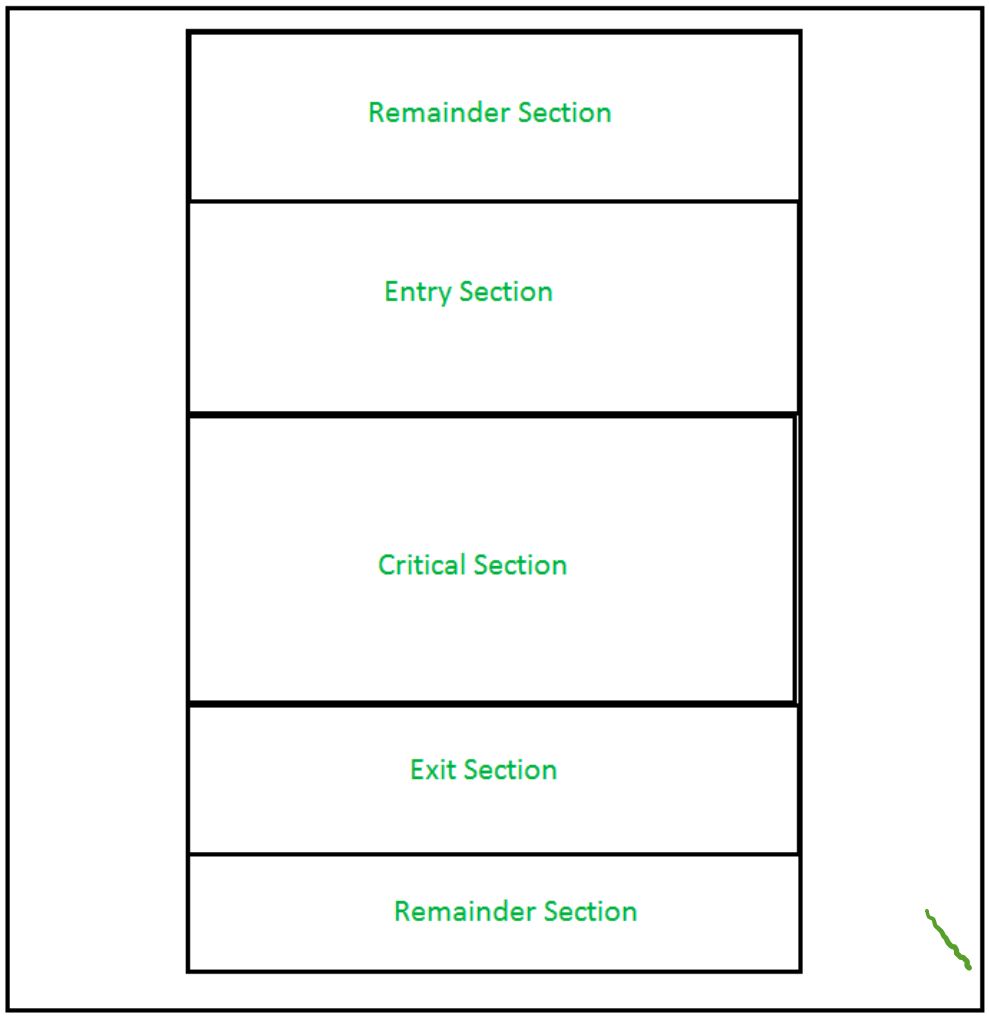
A cooperative process is the one which can affect the execution of other process or can be affected by the execution of other process. Such processes need to be synchronized so that their order of execution can be guaranteed.

The procedure involved in preserving the appropriate order of execution of cooperative processes is known as Process Synchronization. There are various synchronization mechanisms that are used to synchronize the processes.

- Process synchronization is the coordination of process execution so that no two processes can access the same shared data and resources.

It's necessary to implement process synchronization to avoid:

i. Data inconsistency between processes
ii. Process deadlocks
iii. Race conditions

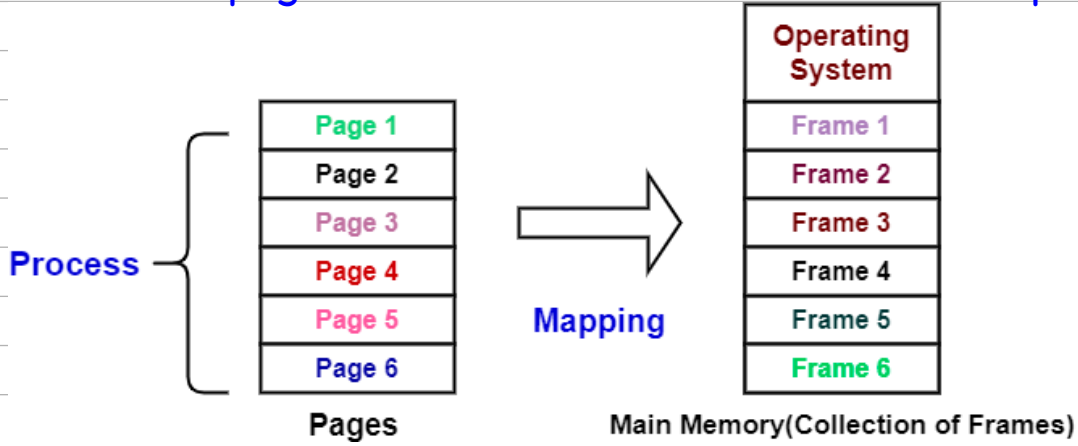


22) Difference Between internal and external fragmentation.

Internal fragmentation	External fragmentation
In internal fragmentation fixed-sized memory, blocks square measure appointed to process.	In external fragmentation, variable-sized memory blocks square measure appointed to the method.
Internal fragmentation happens when the method or process is smaller than the memory.	External fragmentation happens when the method or process is removed.
The solution of internal fragmentation is the best-fit block.	The solution to external fragmentation is compaction and paging.
Internal fragmentation occurs when memory is divided into fixed-sized partitions.	External fragmentation occurs when memory is divided into variable size partitions based on the size of processes.
The difference between memory allocated and required space or memory is called Internal fragmentation.	The unused spaces formed between non-contiguous memory fragments are too small to serve a new process, which is called External fragmentation.
Internal fragmentation occurs with paging and fixed partitioning.	External fragmentation occurs with segmentation and dynamic partitioning.
It occurs on the allocation of a process to a partition greater than the process's requirement. The leftover space causes degradation system performance.	It occurs on the allocation of a process to a partition greater which is exactly the same memory space as it is required.
It occurs in worst fit memory allocation method.	It occurs in best fit and first fit memory allocation method.

23) Explain the logical Diagram of Paging.

In paging, the physical memory is divided into fixed-size blocks called page frames, which are the same size as the pages used by the process. The process's logical address space is also divided into fixed-size blocks called pages, which are the same size as the page frames.



24> Difference between paging and segmentation

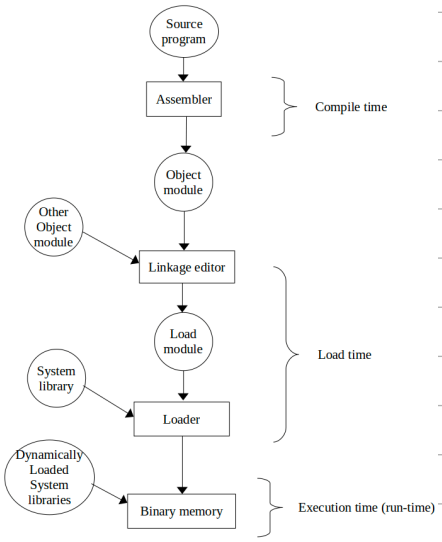
Paging	Segmentation
In paging, the program is divided into fixed or mounted size pages.	In segmentation, the program is divided into variable size sections.
For the paging operating system is accountable.	For segmentation compiler is accountable.
Page size is determined by hardware.	Here, the section size is given by the user.
It is faster in comparison to segmentation.	Segmentation is slow.
Paging could result in internal fragmentation.	Segmentation could result in external fragmentation.
In paging, the logical address is split into a page number and page offset.	Here, the logical address is split into section number and section offset.
Paging comprises a page table that encloses the base address of every page.	While segmentation also comprises the segment table which encloses segment number and segment offset.
The page table is employed to keep up the page data.	Section Table maintains the section data.
In paging, the operating system must maintain a free frame list.	In segmentation, the operating system maintains a list of holes in the main memory.
Paging is invisible to the user.	Segmentation is visible to the user.
In paging, the processor needs the page number, and offset to calculate the absolute address.	In segmentation, the processor uses segment number, and offset to calculate the full address.
It is hard to allow sharing of procedures between processes.	Facilitates sharing of procedures between the processes.
In paging, a programmer cannot efficiently handle data structure.	It can efficiently handle data structures.
This protection is hard to apply.	Easy to apply for protection in segmentation.
The size of the page needs always be equal to the size of frames.	There is no constraint on the size of segments.
A page is referred to as a physical unit of information.	A segment is referred to as a logical unit of information.
Paging results in a less efficient system.	Segmentation results in a more efficient system.

25> What is compaction?

Compaction refers to combining of all the empty spaces together and processes. Compaction helps to solve the problem of fragmentation, but it requires a lot of CPU time. It moves all the occupied areas of storage to one end and leaves one large free space for incoming jobs, instead of numerous small ones.

26> What is address Binding?

The Address Binding refers to the mapping of computer instructions and data to physical memory locations. Both logical and physical addresses are used in computer memory. It assigns a physical memory region to a logical pointer by mapping a physical address to a logical address known as a virtual address. It is also a component of computer memory management that the OS performs on behalf of applications that require memory access.



← Address Binding.

28) Explain the concept of segmentation with paging.

Pure segmentation is not very popular and not being used in many of the operating systems. However, Segmentation can be combined with Paging to get the best features out of both the techniques.

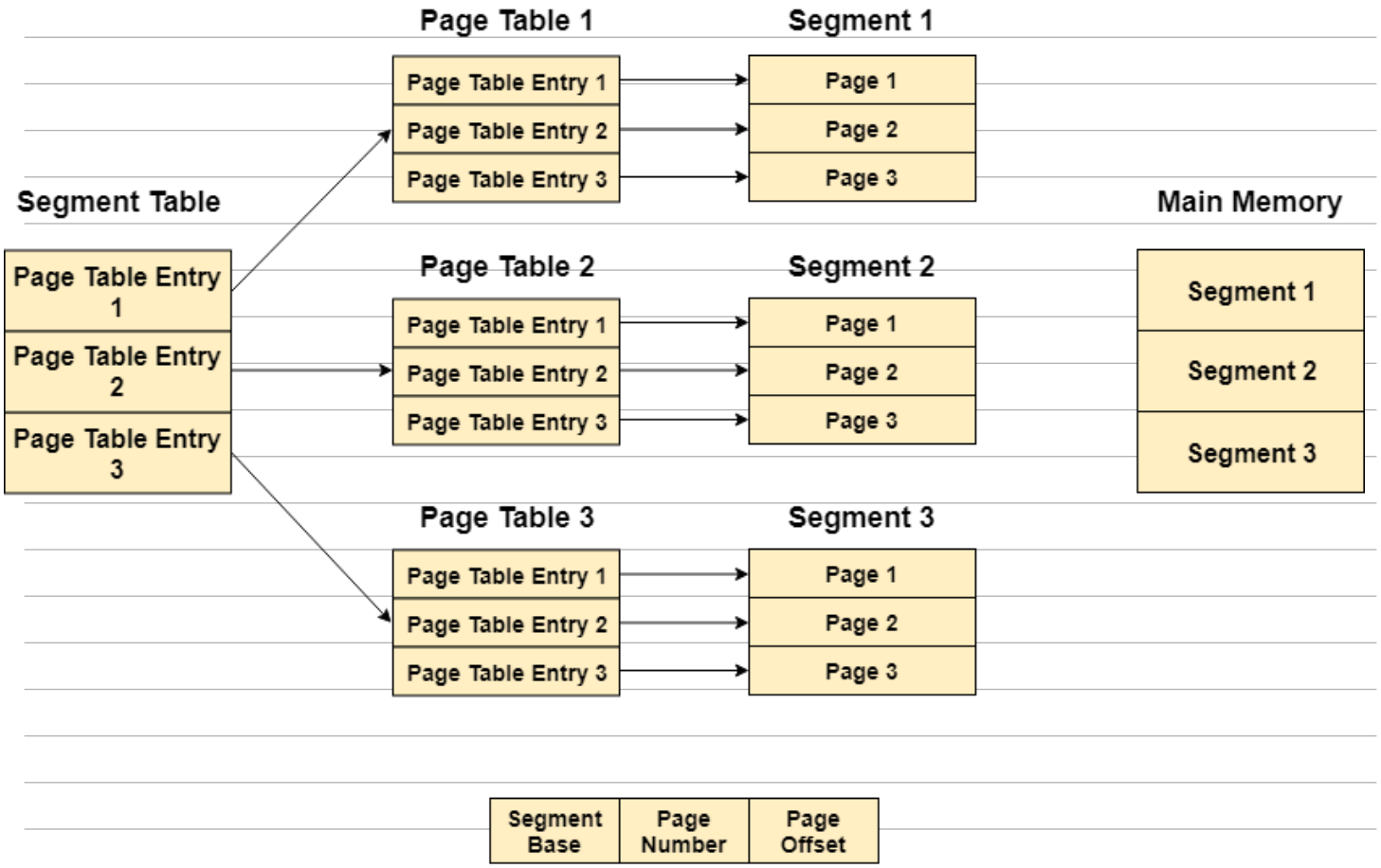
In Segmented Paging, the main memory is divided into variable size segments which are further divided into fixed size pages.

Pages are smaller than segments.
Each Segment has a page table which means every program has multiple page tables.
The logical address is represented as Segment Number (base address), Page number and page offset.
Segment Number → It points to the appropriate Segment Number.

Page Number → It Points to the exact page within the segment

Page Offset → Used as an offset within the page frame

Each Page table contains the various information about every page of the segment. The Segment Table contains the information about every segment. Each segment table entry points to a page table entry and every page table entry is mapped to one of the page within a segment.



Logical Address