Enlist types of operating system? Explain Real time Operating System.

- An Operating System is a system Software that acts as an intermediary/interface between a user of a computer and the computer hardware.
- Types of Operating Systems
- Batch Operating System
- Multiprogramming Operating System
- Time-Sharing OS
- Multiprocessing OS
- Distributed OS
- Network OS
- Embedded OS

- Real Time OS –

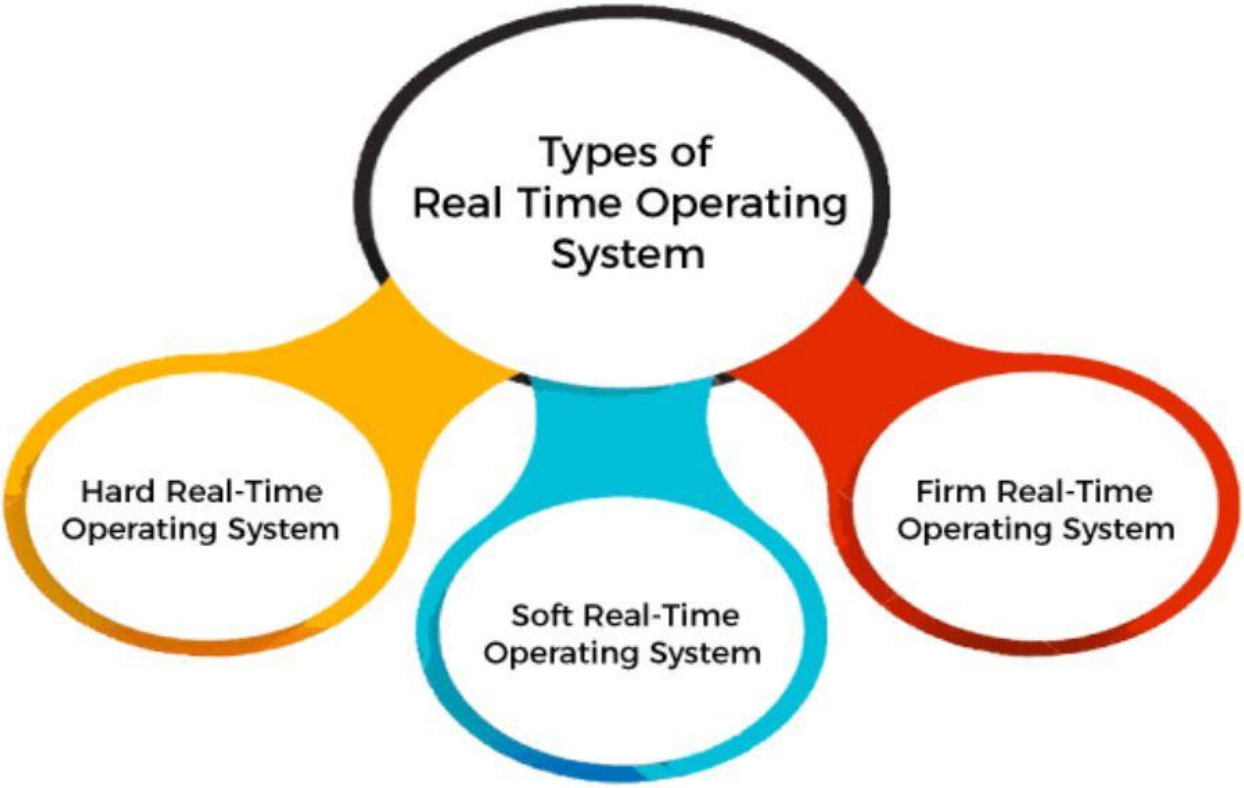
A real-time operating system (RTOS) is a special-purpose operating system used in computers that has strict time constraints for any job to be performed.

Whenever an event external to the computer occurs, it communicates to the computer with the help of some sensor used to monitor the event.

RTOSes provide mechanisms for real-time scheduling, synchronization, inter-process communication, and memory management that are optimized for low latency.

Real-time systems are used when there are time requirements that are very strict like missile systems, air traffic control systems, robots, etc.

Following are the three types of RTOS systems are:



What are major activities of an OS in regard to memory management and file management?

- **memory management** –
  - Memory is a large array of words or bytes, each with its own address.
  - Memory management is an essential aspect of an OS, as it ensures that each process has access to the memory it needs to execute without interfering with other processes.

  The operating system is responsible for the following activities in connections with memory management:
  - *Memory Usage Tracking* – the os keeps track of which parts of memory are currently being used and by whom.
  - *Process Loading* – the os decides which processes to load when memory space becomes available.
  - *Allocation of memory*: The OS manages the allocation of memory to processes, ensuring that each process has sufficient memory to execute without interfering with other processes.
  - *Deallocation of memory:* The OS is responsible for deallocating memory that is no longer needed by a process, freeing it up for other processes to use.
- **File management** :
  - A file is a collection of related information defined by its creator.
  - File systems provide the conventions for the encoding, storage and management of data on a storage device such as a hard disk.

The operating system is responsible for the following activities in connections with file management:

- *File creation and deletion*. – the os manages to create or delete a file and manages the allocation of disk space to files, ensuring that each file has sufficient space to store its data.
- *Directory creation and deletion* : Directory creation and deletion are important functions of file management that allow users to organize and manage their files efficiently. Directories are used to organize files in a hierarchical structure, making it easier for users to locate and access files on a file system.
- *File and Directory Primitives* : The support of primitives for manipulating files and directories refers to the basic operations that an operating system provides to allow users and applications to interact with the file system
- *File Mapping :* In file management, mapping files onto secondary storage refers to the process of allocating space for files on secondary storage devices such as hard disks, solid-state drives, or network storage
- *File backup and recovery*: The OS provides backup and recovery mechanisms to protect against data loss due to system failures, user errors, or other issues.

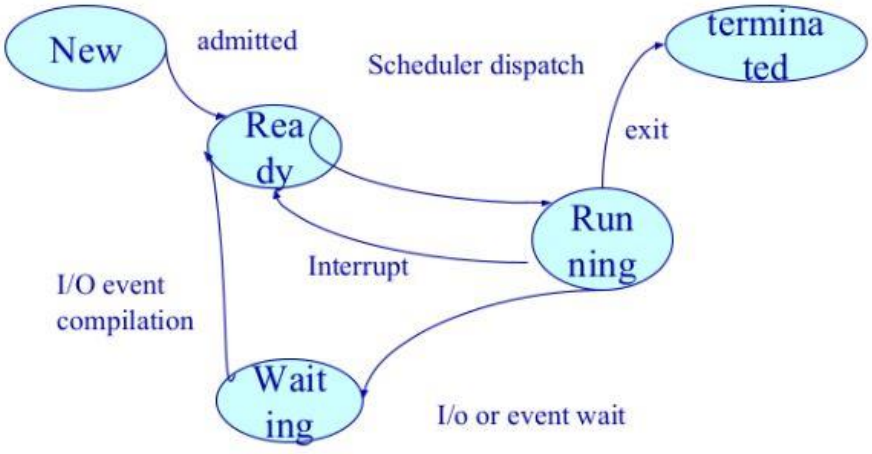
Explain process, process states and PCB in details?

- A **process** is a program in execution that requires resources such as CPU time, memory, files, and I/O devices to accomplish its task.
- These resources are allocated to a process either when it is created or while it is executing.
- Processes can be categorized into two types: operating system processes that execute system code and user processes that execute user code.
- The operating system is responsible for managing processes, which includes creating and deleting processes, scheduling them for execution, providing mechanisms for synchronization and communication between processes, and handling deadlocks that may arise.
- **Process states :**

  The state of a process can change as it executes, depending on its current activities and the resources it requires.

  The different states that a process can be in are:

1. **New** - A process in the "new" state represents a newly created process that has not yet been initialized or started.
2. **Running** - In the "running" state, the process is currently executing on the CPU and performing its assigned tasks.
3. **Waiting** - A process may enter the "waiting" state when it needs to wait for some event to occur, such as input/output (I/O) completion or a signal from another process.
4. **Ready** - A process in the "ready" state is waiting for the CPU to be available so that it can start or resume its execution.
5. **Terminated** - When a process has completed its execution, it enters the "terminated" state, and its resources are released by the operating system.



N.B : Only one process running at a time

Process control Block :

- Each process In os is represented by process control block (PCB).
- The process control block (PCB) is a data structure used by the operating system to manage and control each process's execution.
- It contains information such as the process ID, process state, program counter, CPU registers, memory management information, I/O status information, accounting information, and priority.

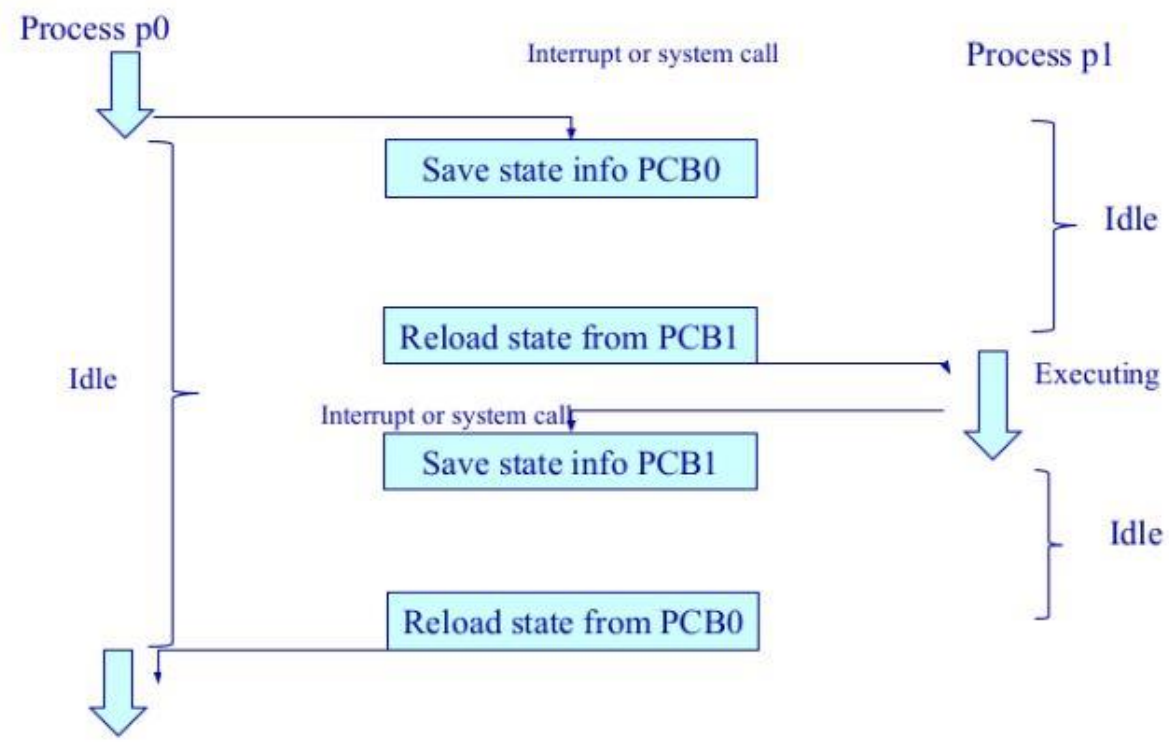


Fig diagram showing CPU switch from process to process

Describe the difference among short-term, medium term and long term scheduling.

Short-term Scheduling	Medium-term Scheduling	Long-term Scheduling
Selects the next process to be executed from those that are ready to run	Determines which processes should be swapped in and out of main memory	Determines which processes should be admitted to the system
Occurs on a timescale of milliseconds to seconds	Occurs on a timescale of seconds to minutes	Occurs on a timescale of minutes to hours
Decides which process should be allocated CPU time next	Decides which processes should be swapped out of main memory to free up space	Decides which new processes should be allowed into the system
Based on the current state of the system and the priority of each process	Based on the amount of available memory and the I/O activity of each process	Based on the overall system workload and the expected demand for resources
Uses a variety of algorithms, including round-robin, priority-based, and shortest job first	Uses algorithms such as page replacement and swapping to manage memory	Uses algorithms such as admission control and load balancing to manage system resources
Affects the performance of the system on a short timescale	Affects the memory utilization of the system on a medium timescale	Affects the overall throughput and responsiveness of the system on a long timescale
Can be preemptive or non-preemptive, depending on the scheduling algorithm used	Can be preemptive or non-preemptive, depending on the page replacement algorithm used	Can be proactive or reactive, depending on the admission control and load balancing algorithms used

Short-term Scheduling	Medium-term Scheduling	Long-term Scheduling
Has a direct impact on the CPU utilization and response time of the system	Has a direct impact on the amount of available memory and the frequency of page faults	Has a direct impact on the number of processes that can be accommodated by the system and the overall system performance

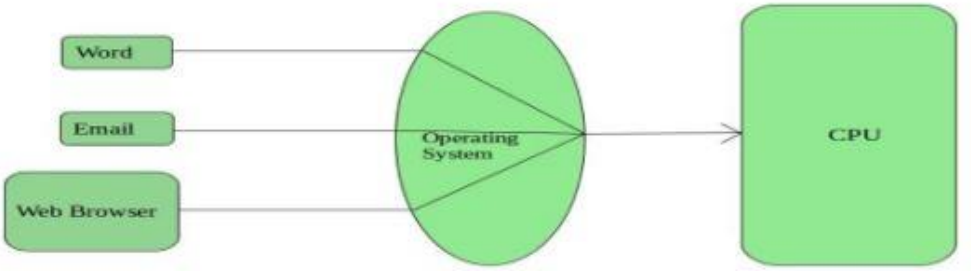
Compare between the following two types of OS's i) Time sharing system ii) Multiprogrammed system

Feature	Time-Sharing Operating Systems	Multiprogramming Operating Systems
Execution of Jobs	Executes jobs in a shared environment by giving each process a small amount of time to execute.	Executes multiple programs simultaneously by organizing jobs so that the CPU always has one job to execute.
CPU Utilization	Increases CPU utilization by giving each process a small amount of time to execute.	Increases CPU utilization by running multiple programs simultaneously.
Job and CPU Scheduling	Uses CPU scheduling algorithms to allocate CPU time to different processes.	Uses job scheduling algorithms to decide which jobs to run and in which order, and uses CPU scheduling algorithms to allocate CPU time to different jobs.
Memory Management	Uses virtual memory to allocate memory to different processes.	Uses dynamic memory allocation to allocate memory to different programs.

Feature	Time-Sharing Operating Systems	Multiprogramming Operating Systems
User Interaction	Designed for interactive use by multiple users, providing immediate responses to user inputs.	Designed for batch processing of jobs without user interaction.
Goal	To provide a fast response time and to maximize the number of users that can use the system.	To maximize the throughput and utilization of the system resources.

Time-Sharing Operating Systems .3

- Each task is given some time to execute so that all the tasks work smoothly.
- These systems are also known as **Multi-tasking 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.



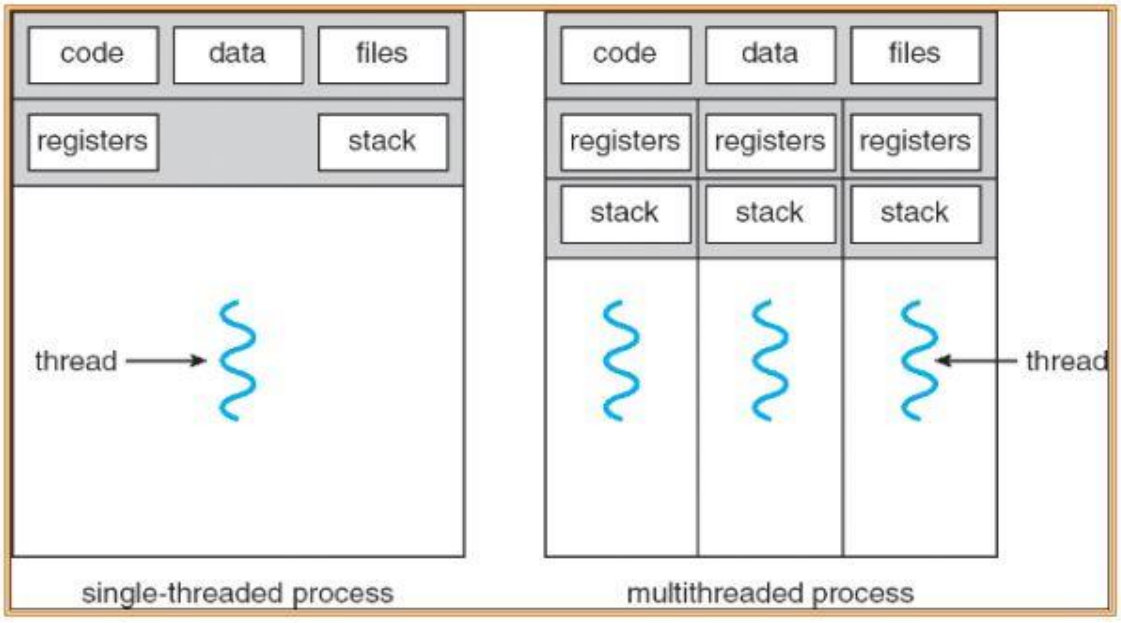
:Multiprogramming Operating System .2

- This type of OS is used to execute more than one jobs simultaneously by a single processor.
- It increases CPU utilization by organizing jobs so that the CPU always has one job to execute.
- Multiprogramming operating systems use the mechanism of job scheduling and CPU scheduling.

Explain Thread in detail?

- A thread is **a lightweight process** that can run in parallel with other threads within the same process.
- Unlike a process, a thread does not have its own address space and does not represent a separate program.
- Instead, a thread shares the same address space as other threads within the same process.
- Threads can execute on different cores on a multi core CPU (parallelism for performance) and can communicate with other threads by updating memory.
- A threads has no data segment or heaps and cannot live on its own, it must live within a process
- There can be more than one thread in a process, the first thread calls main & has the process's stack
- It is a basic unit of cpu execution consisting of a program counter, thread id , stack , set of registers.
- Each thread can run on a different physical process and have Inexpensive creation and context switch.

Single and Multithreaded Processes



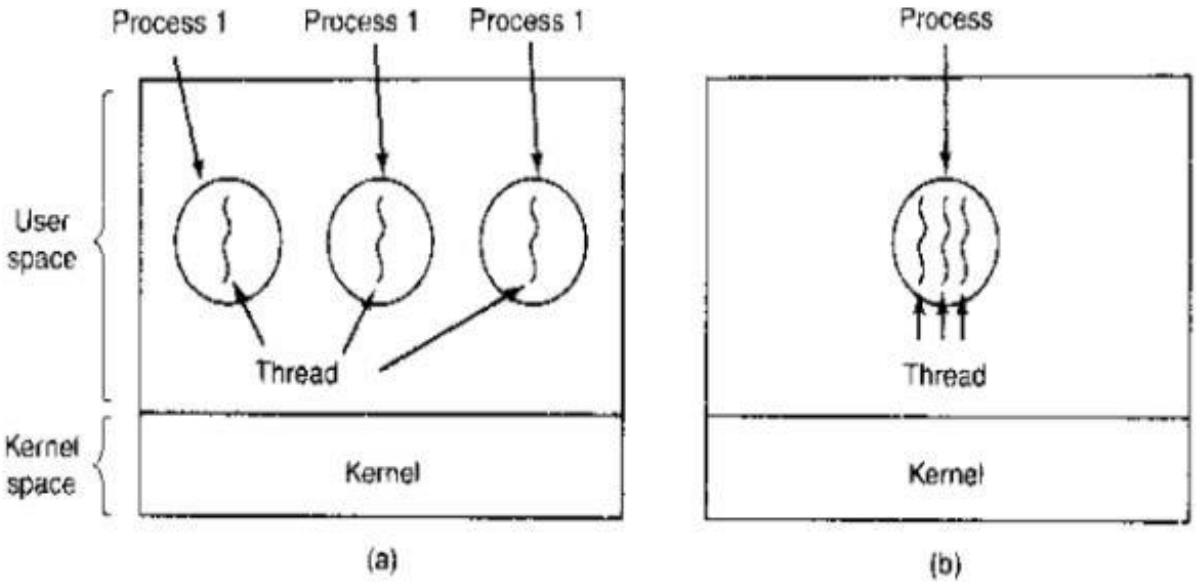
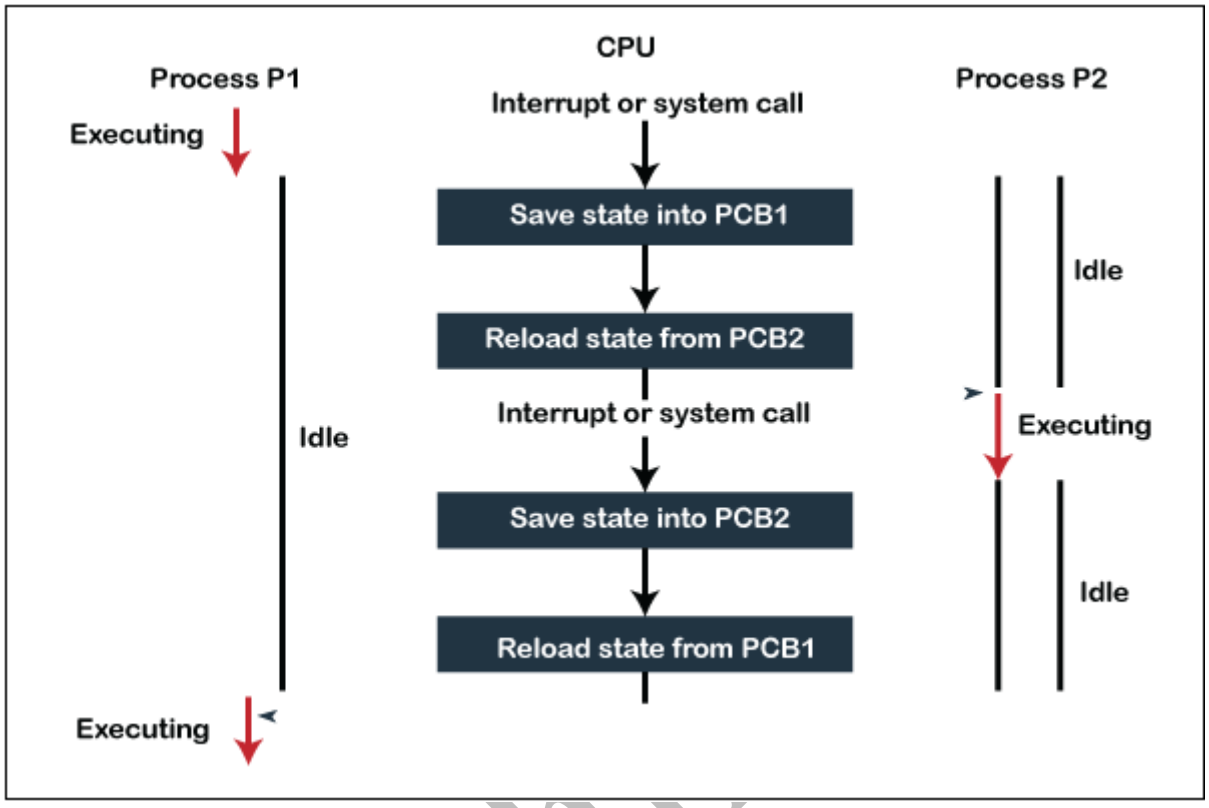


Figure 2-6. (a) Three processes each with one thread. (b) One process with three threads.

Explain context switching with the help of a diagram.

- A context switch is the process of switching the CPU from one process to another. This involves saving the state of the current process and loading the saved state for the new process.
- The purpose of a context switch is to enable multiple processes to run on a single CPU, and it is an essential feature of multitasking operating systems.
- During the process, the system does not perform any useful work, which can result in a decrease in system performance.
- The speed of a context switch can vary from machine to machine, depending on factors such as memory speed and the number of registers.
- Context switching speed can range from 1 microsecond to 1000 microseconds. Faster context switches generally lead to more efficient system performance because the overhead is minimized
- To optimize system performance, operating systems implement various techniques to reduce the overhead of context switching. One such technique is prioritizing processes,
- In conclusion, context switching is an essential feature of multitasking operating systems that enables multiple processes to run on a single CPU.



Explain the mechanism of process creation and termination.

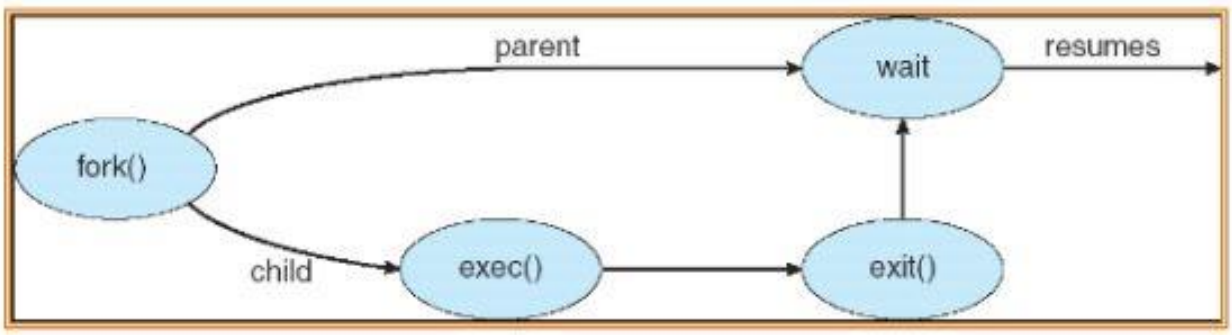
- **Process creation** involves the creation of new processes during the course of execution, typically done via a createprocess system call.
- A parent process can create child processes, forming a tree of processes
- When a process creates a new process, the two possibilities in terms of execution are:

1. The parent continues to execute concurrently with its children.
2. The parent waits until some or all of its children have terminated.

- When a process creates a new process, the two possibilities in terms of execution are:

1. The parent continues to execute concurrently with its children.
2. The parent waits until some or all of its children have terminated.

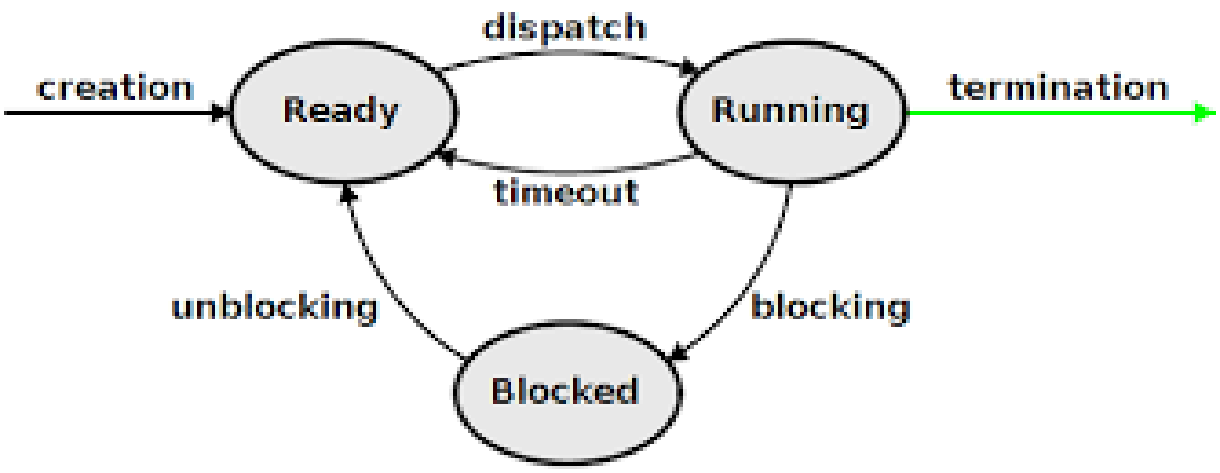
- When a process creates a new process, the two possibilities in terms of execution are:
- The parent continues to execute concurrently with its children.
- The parent waits until some or all of its children have terminated.



Process Termination :

- Process termination occurs when a process executes its last statement and asks the operating system to delete it.

- The output data from a child to a parent is typically done via wait.
- The process's resources are deallocated by the operating system, and the parent may terminate the execution of children processes if they have exceeded allocated resources
- Parent may terminate execution of children processes (abort) if,
  - Child has exceeded allocated resources
  - Task assigned to child is no longer required
  - If parent is exiting
- If a parent process terminates, some operating systems do not allow its child processes to continue executing. In this case, all children processes are terminated, which is known as cascading termination.
- In summary, process termination is a crucial mechanism in the management of processes in an operating system.



Explain multiprocessor system? Give advantages and disadvantages.

- A multiprocessor operating system is an OS that has more than one processor working together to execute multiple jobs simultaneously.
- Multiprocessor operating systems are also known as parallel OS or tightly coupled OS.
- In this system, multiple processors are in close communication and share the computer bus, clock, memory, and peripheral devices.
- It executes multiple jobs at the same time and makes the processing faster.
- If one processor fails then other processor should retrieve the interrupted process state so execution of process can continue.
- Inter-processes communication mechanism is provided and implemented in hardware.

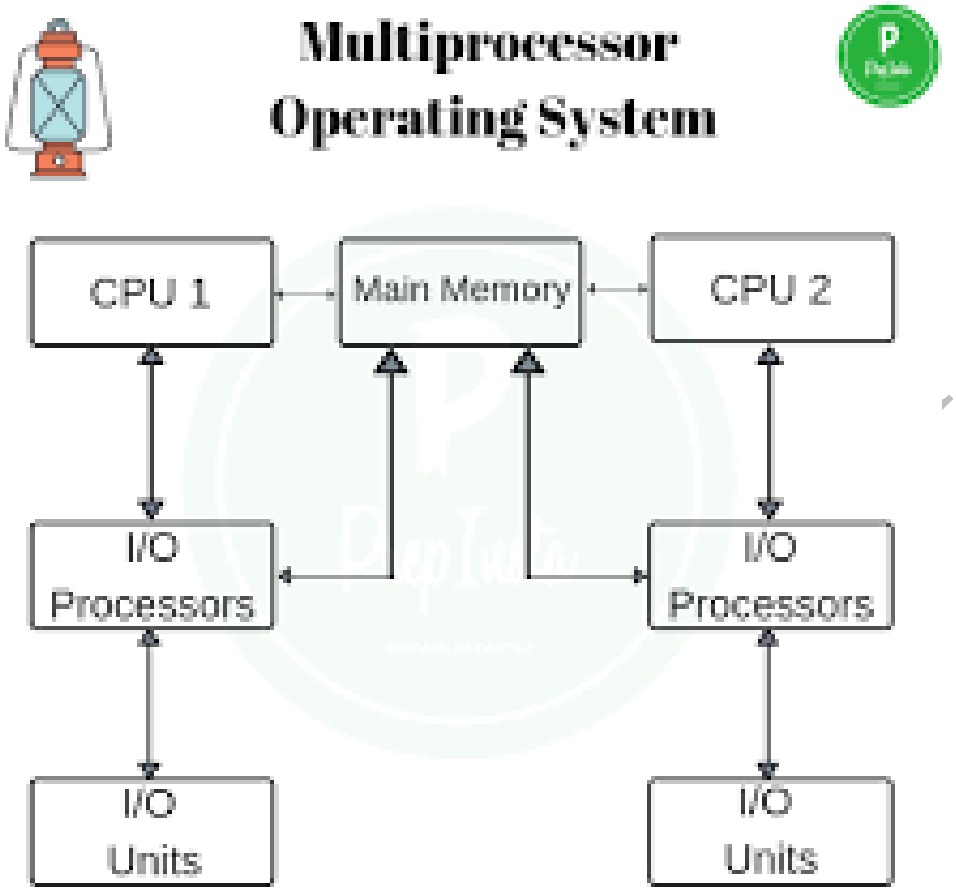
Advantages :

- Multiprocessor systems have the potential to process large amounts of data faster than a single processor system.
- It supports large physical and virtual address space.
- If one processor fails, the other processors can continue to function, increasing system reliability.
- Multiprocessor systems can be more cost-effective than single-processor systems, as multiple processors can be integrated into a single system.

Disadvantages :

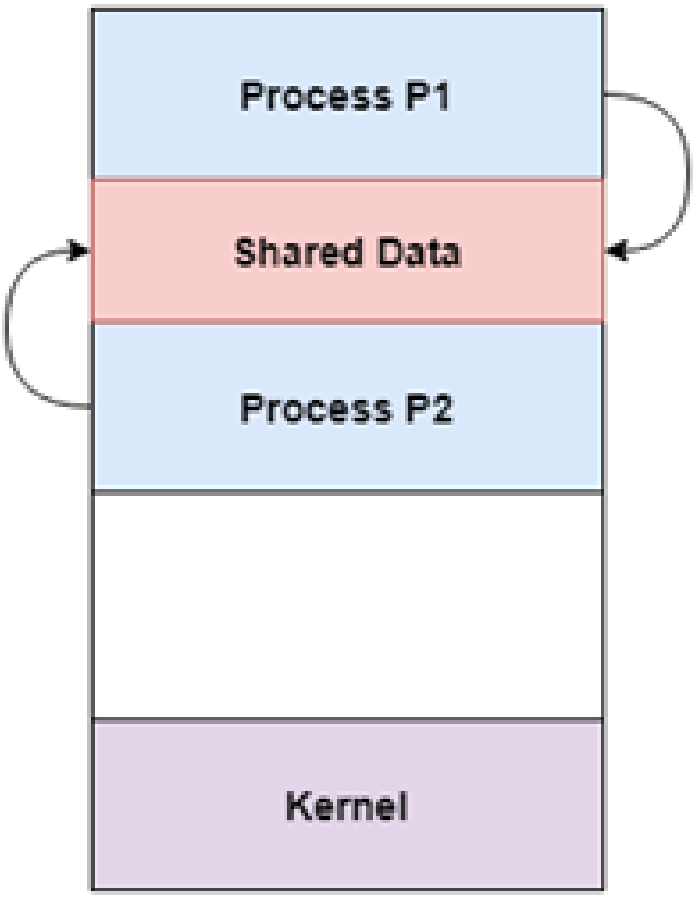
- Implementing and managing a multiprocessor system can be more complex than a single-processor system.
- Parallel processing can create additional overhead and can be less efficient than serial processing for certain tasks.

- Inter-processor communication can be difficult and time-consuming to manage.

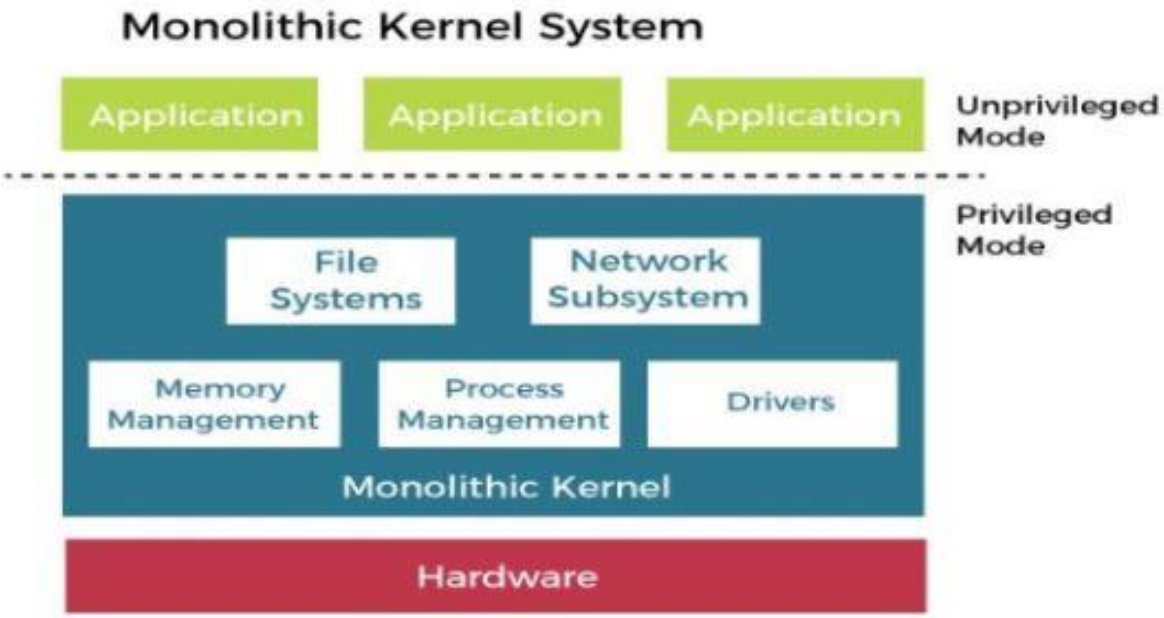


What is the need of cooperation between processes? Explain process cooperation with an example.

- The concurrent processes executing in the operating system may be either
  - independent processes or
  - cooperating processes.
- The need for cooperation between processes arises in the concurrent execution of processes in the operating system.
- When processes need to share data or synchronize their actions, they become cooperating processes.
- A process is cooperating if it can affect or be affected by the other processes executing in the system- any process that shares data with other processes is a cooperating process.
- Inter-process communication is required for processes to communicate with each other.
- An example of process cooperation is the producer-consumer problem. In this paradigm, a producer process generates data or information that is consumed by a consumer process.
- a print program produces characters that are consumed by the printer driver.
- A compiler produce assembly code, which is consumed by an assembler.
- To allow producer and consumer processes to run concurrently, we must have available a buffer of items that can be filled by the producer and emptied by the consumer.
- A producer can produce one item while the consumer is consuming another item.
- The producer and consumer must be synchronized, so that the consumer does not try to consume an item that has not yet been produced.



Explain operating system with monolithic structure.



- In a monolithic operating system structure, all operating system services, such as process management, memory management, file systems, device drivers, and network protocols, are implemented as a single, large program called the kernel.
- The kernel has complete access to all system resources, and user-level applications interact with the kernel through a set of system calls.
- This means that all operating system services run in the same address space as the kernel, sharing the same memory and resources.
- A monolithic kernel is an operating system architecture where the entire operating system is working in kernel space.
- A microkernel is divided into two parts, kernel space, and user space. Both parts communicate with each other through IPC (Inter-process communication).
- Microkernel's advantage is that if one server fails, then the other server takes control of it.

- the monolithic kernel architecture is relatively simple to design and implement, and it can offer good performance in certain applications.

What are the different scheduling queues? Represent process scheduling with the help of a queuing diagram.

- The process scheduling is the activity of the process manager that handles the removal of the running process from the CPU and the selection of another process on the basis of a particular strategy.
- Ready queue :
  - This is a list of processes that are in main memory and ready to execute.
  - The ready queue is typically implemented as a linked list, where each node of the list represents a Process Control Block (PCB) that contains information about a particular process.
  - The ready queue header contains pointers to the first and last PCBs in the list, and each PCB has a pointer field that points to the next process in the ready queue.
- Device queue :
  - This is a list of processes waiting for a particular I/O device.
  - Each I/O device has its own device queue, and processes waiting for that device are placed on the queue.
  - Like the ready queue, the device queue is typically implemented as a linked list of PCBs.
- **Process scheduling :**

  A process can be scheduled on any CPU in a multiprocessor system.

  Process scheduling is the mechanism by which an operating system selects which process should run on the CPU at a given time.

  The scheduler selects processes from a queue of processes waiting to be executed, based on various scheduling algorithms.

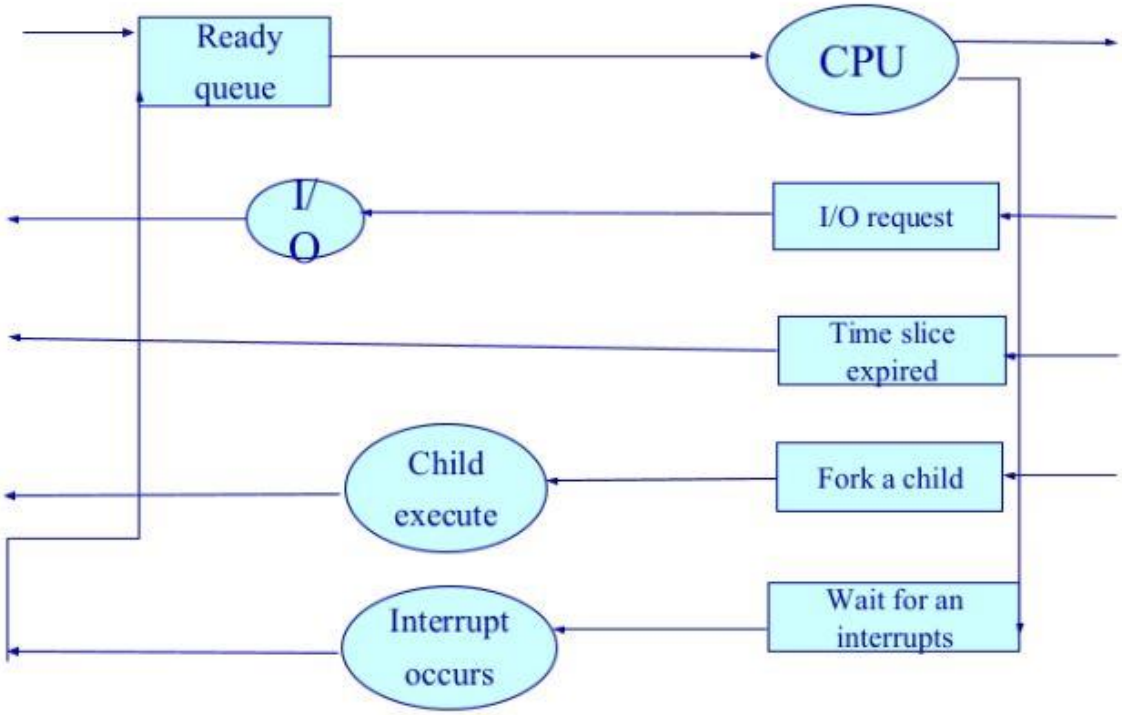


Fig Queuing diagram representation of process scheduling

Short note on: i) Inter process Communication(IPC) ii) System Call iii) Bounded buffer problem iv) Batch OS

i) Inter process Communication(IPC) :

- Inter Process Communication (IPC) is a mechanism by which processes running on an operating system can communicate with each other.
- IPC mechanisms can be categorized into several types, including message passing, shared memory, and synchronization primitives.
- An IPC facility typically provides two basic operations: sending a message and receiving a message.
- When two processes, P and Q, want to communicate with each other, they need to establish a communication link between them and exchange messages via send/receive operations.
- To establish communication between processes, a communication link needs to be established between them. This link can be established through a shared memory area, a network socket or another mechanism.
- Once the communication link is established, the processes can exchange messages through the send/receive operations provided by the IPC facility.

ii) System Call :

- System call is the programmatic way in which a computer program/user application requests a service from the kernel of the operating system on which it is executed.
- System calls are essential for ensuring the security of the operating system, as they prevent user applications from directly accessing privileged resources.

- stead, they must make requests to the operating system, which then carries out the requested operations on their behalf.
- System calls are also known as software interrupts, as they allow user applications to interrupt the normal execution of the CPU and enter a protected mode in which they can access privileged resources.
- When they need to do any I/O or have some more memory or spawn a process or wait for signal/interrupt, it requests operating system to facilitate all these. This request is made through System Call.
- They are an essential component of modern computing systems and enable applications to interact with the underlying operating system in a secure and controlled manner.

iii) Bounded buffer problem :

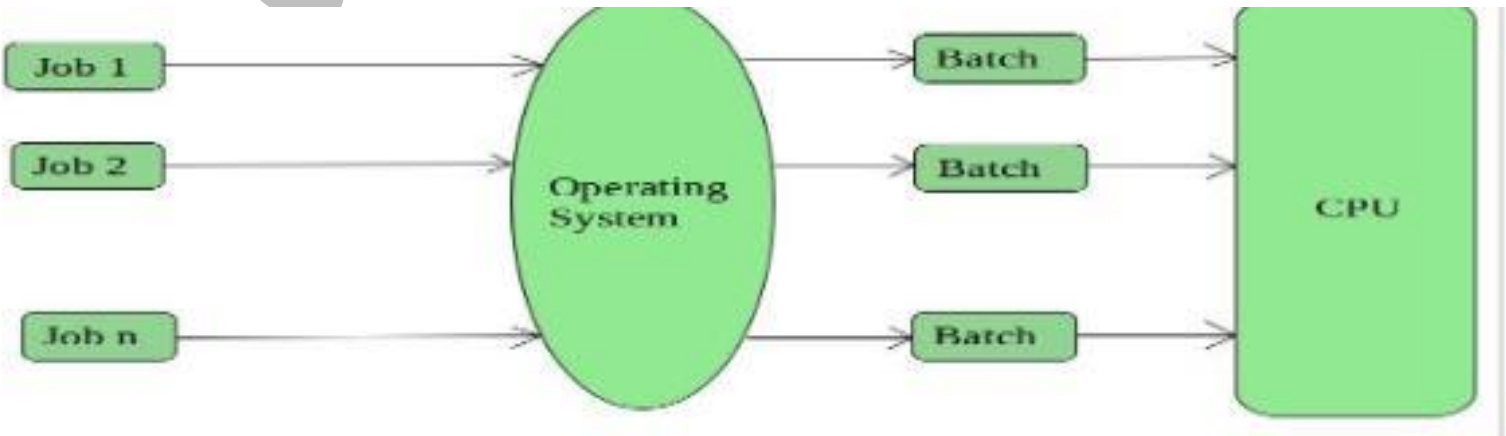
- In the bounded-buffer producer-consumer problem, there is a fixed size buffer which is shared between two processes - the producer and the consumer. The producer's job is to generate data and put it into the buffer, while the consumer's job is to remove data from the buffer and process it.
- The problem arises because the producer must wait if the buffer is full, and the consumer must wait if the buffer is empty.
- To solve this problem , The producer and consumer must coordinate with each other to ensure that the buffer is not overfilled or underfilled.
- When the producer wants to add data to the buffer, it first checks if there is space available in the buffer. If the

buffer is full, the producer must wait until there is space available.

- Similarly, when the consumer wants to remove data from the buffer, it first checks if there is data available. If the buffer is empty, the consumer must wait until there is data available.
- To ensure that the producer and consumer can communicate with each other and coordinate their actions, they must use some form of interprocess communication mechanism, such as message passing or shared memory.

iv) Batch OS :

- Batch operating system is a type of operating system where the users submit their jobs in batches to the computer operator.
- The jobs are collected by the operator and are combined into batches based on their similar requirements. These batches are then executed by the computer without any user interaction.
- In a batch operating system, the users prepare their jobs on an off-line device such as punch cards, magnetic tapes, or disk packs.
- The batch operating system is suitable for processing large volumes of data with a similar type of processing requirements.



Advantages of Batch Operating System:

- Processors of the batch systems know how long the job would be when it is in queue
- Multiple users can share the batch systems
- The idle time for the batch system is very less
- It is easy to manage large work repeatedly in batch systems

Disadvantages of Batch Operating System:

- The computer operators should be well known with batch systems
- Batch systems are hard to debug
- It is sometimes costly
- The other jobs will have to wait for an unknown time if any job fails

Examples of Batch based Operating System:

IBM's MVS