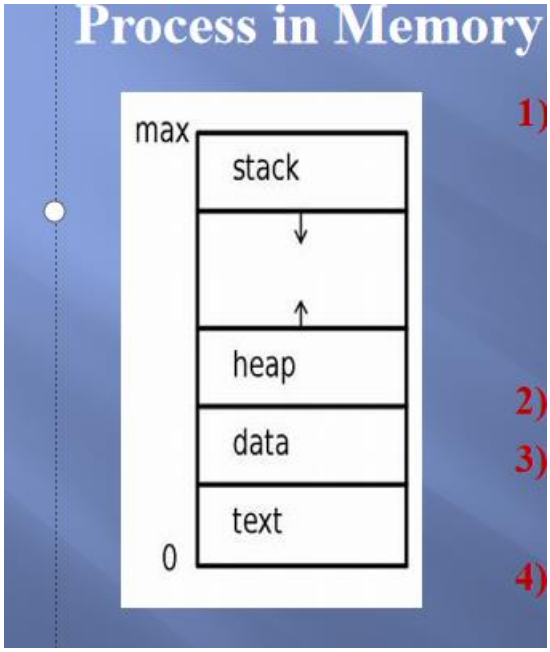
Q1)whta is process?

SOL)

- A Process is a program in execution
- Process is not as same as program code but a lot more than it.
- the Attributes held by process include hardware, memory, CPU etc.

Process Memory is divided into 4 sections for efficient working:

1. Text Section
2. Data Section
3. Heap
4. Stack



Q2)Process State?

SOL)

- The process, from its creation to completion, passes through various states.

The minimum number of states is five:

1. New
2. Ready
3. Running

4.wait

5. Completion or termination

1.NEW:

✓ The Process is being created

2.Ready:

✓ The Process is waiting to be assigned to a processor.

3.Running:

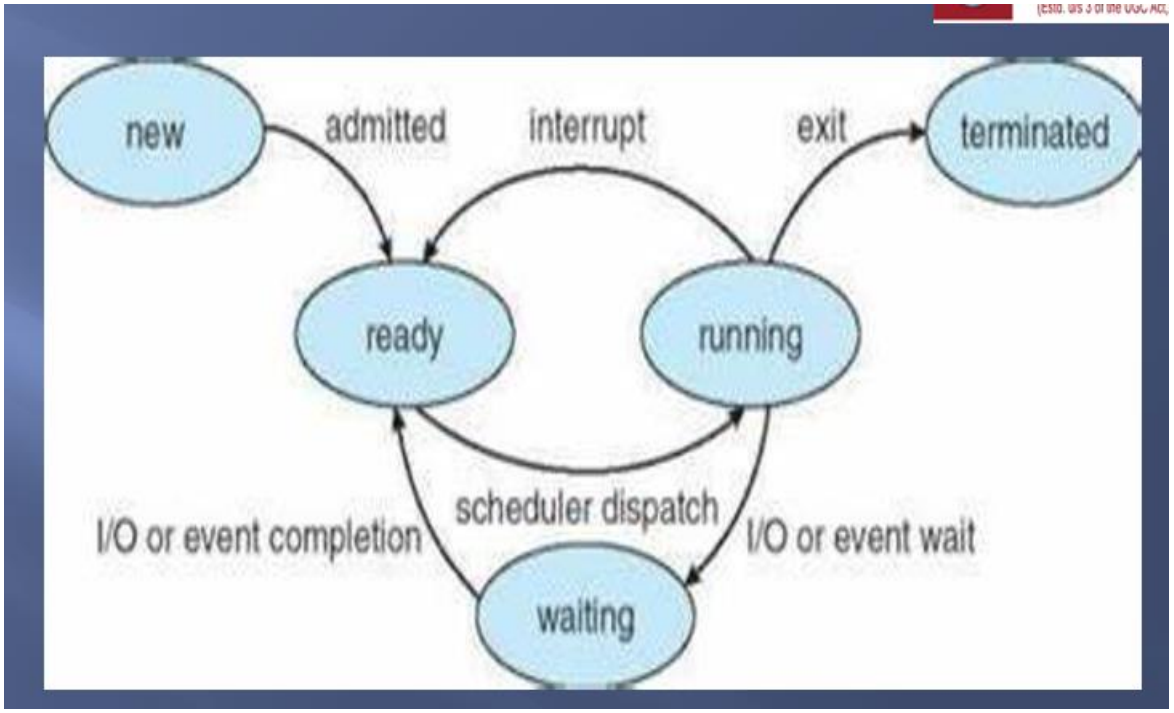
✓ Instructions are being executed.

4.Waiting:

✓ The Process is waiting for some event to occur(such as an I/O completion)

5.Terminated:

✓ The Process has finished execution

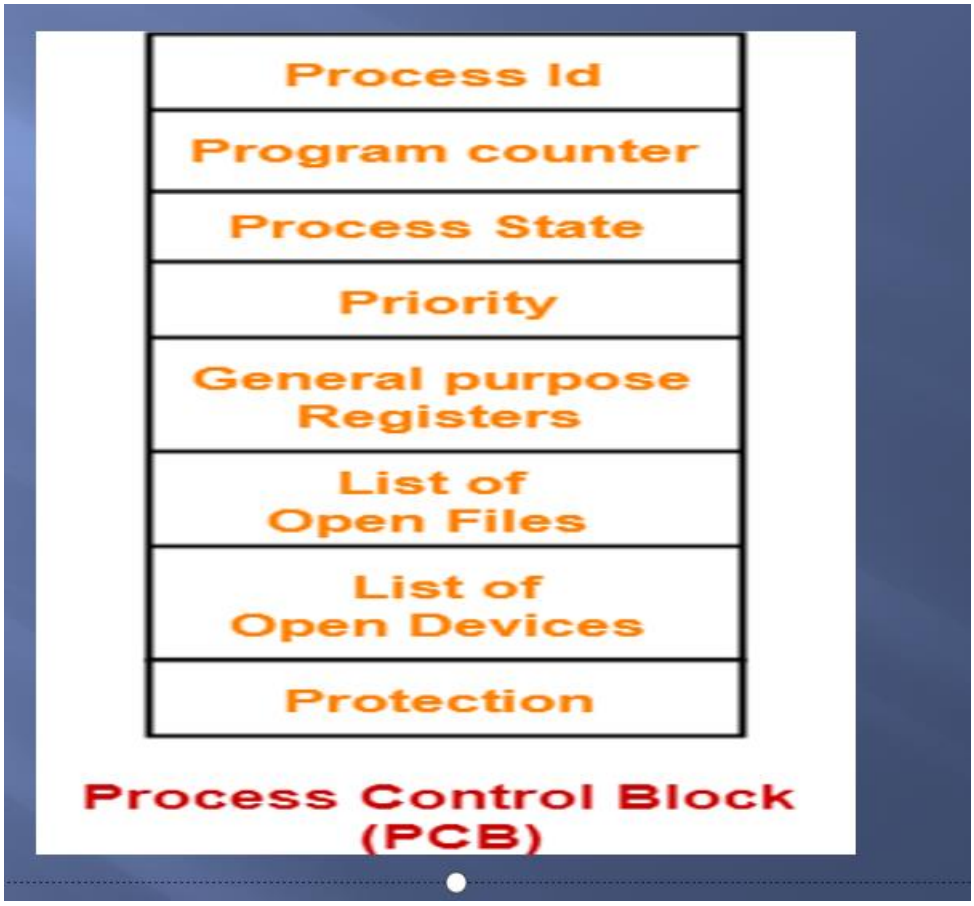


.....

********Q3)Process Control Block(PCB) or Task Control Block(TCB):**

SOL)

- The process control stores many data items that are needed for efficient process management.
- Some of these data items are explained with the help of the given diagram
-



Process State:

- ✓ This specifies the process state i.e. new, ready, running, waiting or terminated.

Process Number:

- ✓ This shows the number of the particular process.

Program Counter:

- ✓ This contains the address of the next instruction that needs to be executed in the process.

List of Open Files:

- ✓ These are the different files that are associated with the process

I/O Status Information:

- ✓ This information includes the list of I/O devices used by the process, the list of files etc.

*****q4)process scheduling?**

sol)

- ➢ The act of determining which process in the ready state should be moved to the running state is known as Process Scheduling
- ➢ The aim of the process scheduling system is to keep the CPU busy all the time and to deliver minimum response time for all programs.
- ➢

- For achieving this, the scheduler must apply appropriate rules for swapping processes IN and OUT of CPU.
-

*****q5)Scheduling Queues?**

sol)

-
- All Processes when enters into the system are stored in the Job Queue/Job Pool.
- Process in the Ready state are placed in the Ready Queue.
- It is represented in Linked List and PCB contain pointer field.
- Processes waiting for a device to become available are placed in Device Queues.

- The objective of multiprogramming is to have some process running at all time, to maximize the CPU utilization.
- Time sharing is used to switch CPU among the processes

For this scheduling we need Scheduling queues.

I)Job queue:

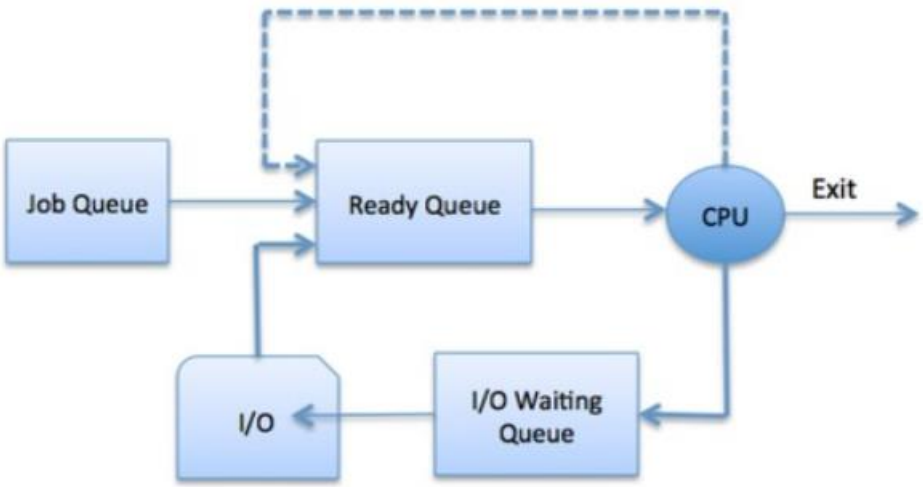
This queue keeps all the processes in the system.

II)Ready queue:

This queue keeps a set of all processes in main memory,and ready and waiting to execute.

III)Device queues:

The processes which are blocked due to unavailability of an I/O device constitute this queue.



.....

****Q6)Schedulers?**

SOL)

- Schedulers are special system software which handle process scheduling in various ways
- Their main task is to select the jobs to be submitted into the system and to decide which process to run.

Schedulers are of three types

- ✓ 1.Long-Term Scheduler
- ✓ 2.Short-Term Scheduler
- ✓ 3.Medium-Term Scheduler

1.Long Term Scheduler:

- ✓ It is also called a job scheduler.
- ✓ Long Term Scheduler run less frequently.
- ✓ It selects processes from job pool and loads them into the memory for execution.
- ✓

2. Short-Term Scheduler:

- ✓ It is also called as CPU scheduler.
- ✓ Short-Term Scheduler run frequently.
- ✓ it increase the CPU Performance and Execution Rate.
- ✓
- ✓ It used for selects from among the processes that are ready to execute and allocates the CPU to one of them.

3. Medium-Term Scheduler:

- ✓ Medium-term scheduling is a part of swapping
- ✓ It removes the processes from the memory.
- ✓ It reduces the degree of multiprogramming

****Q7)COMPARISION BETWEEN Long-Term Scheduler VS medium term schedular vs long term schedular?

sol)

SR. NO.	SHORT TERM SCHEDULER	MEDIUM TERM SCHEDULER	LONG TERM SCHEDULER
1.	It is a CPU scheduler.	It is a process swapping scheduler.	It is a job scheduler.
2.	Speed is fastest than other two.	Speed is in between both short and long term scheduler.	Speed is lesser than short term scheduler.
3.	It is minimal in time sharing system	It is a part of Time sharing systems..	It is almost absent or minimal in time sharing system.
4.	It provides lesser control over degree of multi-programming.	It reduces the degree of multi-programming.	It controls the degree of multi-programming.
5.	It selects those processes which are ready to execute.	It can re-introduce the process into memory and execution can be continued.	It selects processes from pool and loads them into memory for execution.

Basis	Short-Term Scheduler	Medium-term Scheduler	Long-Term Scheduler
1. Alternate Name	It is also called a CPU scheduler.	It is also called a process swapping scheduler.	It is also called a job scheduler.
2. Degree in programming	It provides lesser control over the degree of multiprogramming.	It reduces the control over the degree of multiprogramming.	It controls the degree of multiprogramming.
3. Speed	The speed of the short-term scheduler is very fast.	Speed of medium scheduler between the short-term and long-term scheduler	The speed of a long-term term scheduler is more than medium-term scheduler.
4. Usage in time-sharing system sharing system	It is minimal in the time-sharing system.	It is a part of the time-sharing system.	It is almost absent or minimal in a sharing system.
5. Purpose	It selects the processes from among the process that is ready to execute.	It can reintroduce the from among the process into memory that executes and its execution can be continued.	It selects processes from the pool and loads them into memory for execution.

-***q8)explain Context Switch ?**

sol)

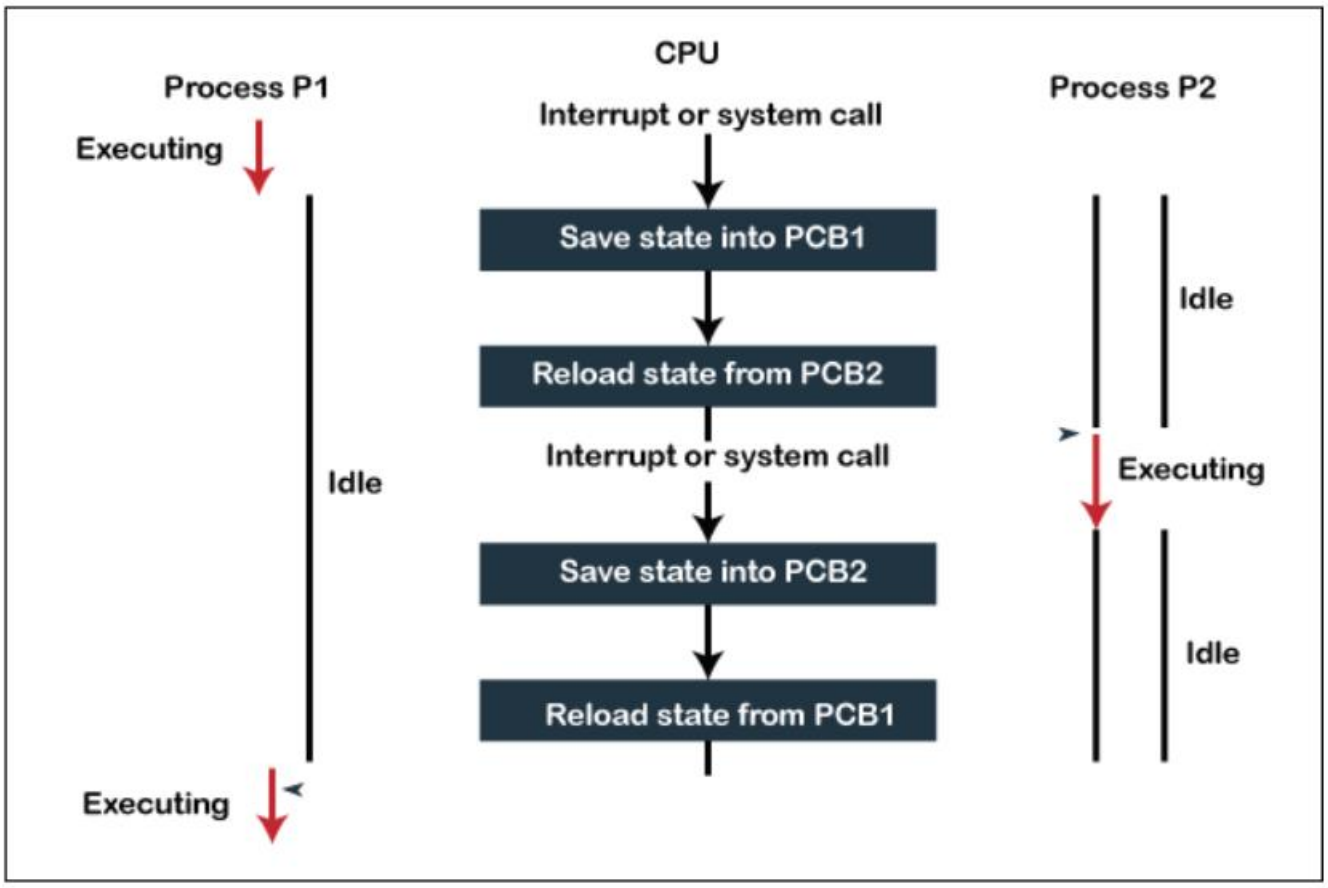
- The Context switching is a technique or method used by the operating system to

switch a process from one state to another to execute its function using CPUs in the system.

- When switching perform in the system, it stores the old running process's status in the form of registers and assigns the CPU to a new process to execute its tasks.
- While a new process is running in the system, the previous process must wait in a ready queue.
- The execution of the old process starts at that point where another process stopped it.
- A context switching helps to share a single CPU across all processes to complete its execution and store the system's tasks status.
- theere are three types of context switching triggers as follows

1. Interrupts
2. Multitasking
3.Kernel/User switch

- There are several steps involves in context switching of the processes.
- The following diagram represents the context switching of two processes, P1 to P2, when an interrupt, I/O needs, or priority-based process occurs in the ready queue of PCB.
-



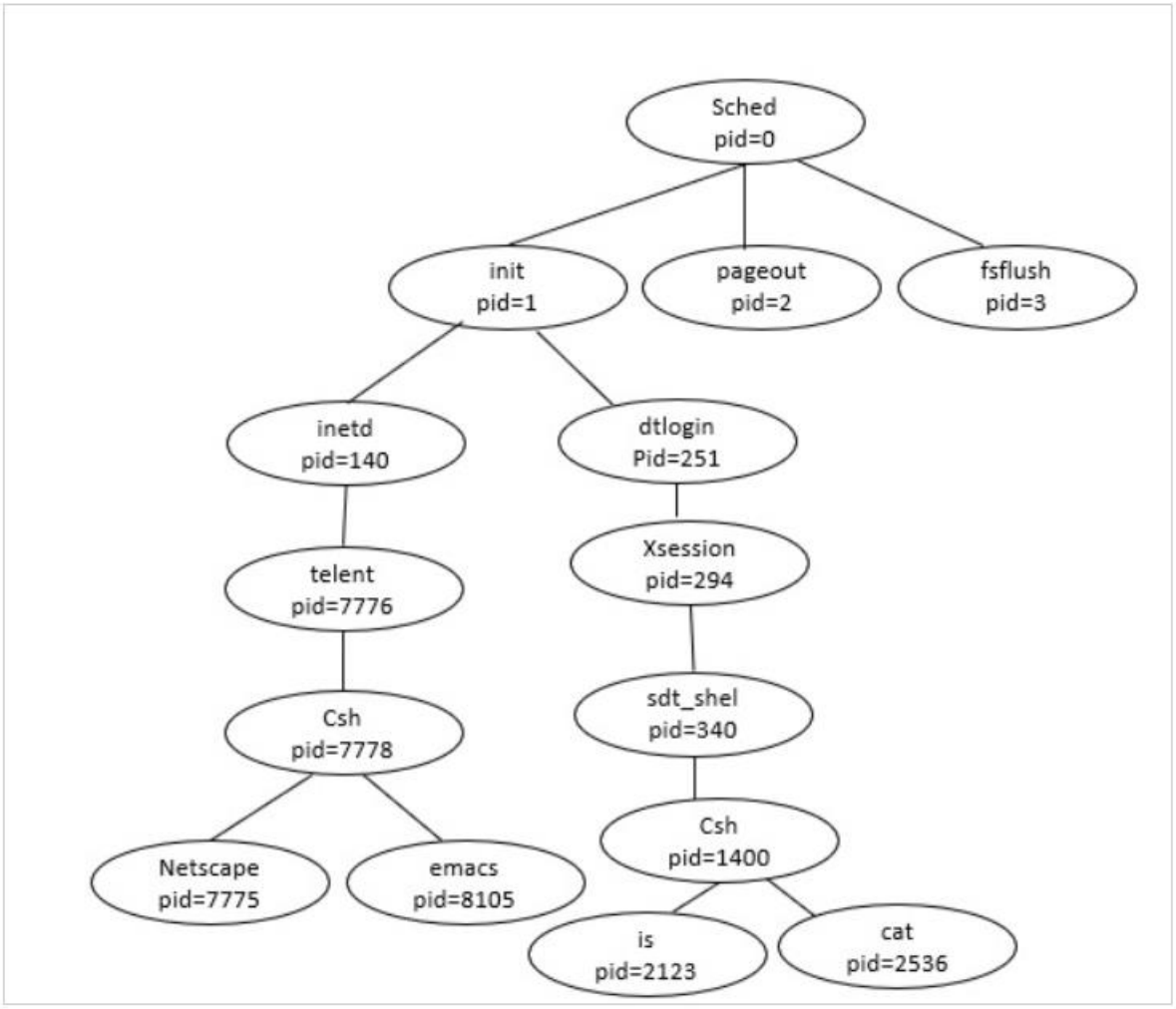
****q9)explain process creation?**

sol)

- A process can create several new processes through creating process system calls during the process execution

- Creating a process we call it the parent process and the new process is a child process.
- Every new process creates another process forming a tree-like structure.
- The creating process is called Parent process.
- The new process created are called Child process of that process
- Each of this new processes may also create other processes ,which are forming a tree.

Let us consider a tree of process:


Sched
pid=0
init
pid=1
pageout
pid=2
fsflush
pid=3
inetd
pid=140
dtlogin
Pid=251
telent
pid=7776
Xsession
pid=294
Csh
pid=7778
sdt_shel
pid=340
Csh
pid=1400
Netscape
pid=7775
emacs
pid=8105
is
pid=2123
cat
pid=2536

example:

****q10)explain process termination?**

sol)

- Whenever the process finishes executing its final statement and asks the operating system to delete it by using exit() system call.
- All the resources of the process including physical and virtual memory, open files, I/O

buffers are deallocated by the operating system.

- When the child process terminates it has to send the status to the parent with the wait() system call.
- The child process can also terminated by the parent.

Causes for termination:

1.Time slot expired
2.Memory bound violation
3.I/O failure
4.Process request
5.Invalid instruction

.....

q11)What is Inter Process Communication(IPC)?

sol)

- Inter Process Communication is a type of mechanism usually provided by the operating system (or OS).
- The main aim or goal of this mechanism is to provide communications in between several processes
- "Inter-process communication is used for exchanging useful information between numerous threads in one or more processes (or programs)."
- Process executing concurrently in the OS may be either independent process or cooperating process

INDEPENDENT PROCESS:

- ✓ An independent process is not affected by the execution of other processes

CO-OPERATING PROCESS:

- ✓ co-operating process can be affected by other executing processes
- ➢ Any process that share the data with other process is a cooperating process.
- ➢ IPC is required only in Cooperating process

REASONS FOR PROVIDING AN ENVIRONMENT FOR PROCESS COOPERATION:

1. INFORMATION SHARING
2. COMPUTATTION SPEED
3. MODULARITY
4. CONVINENCE

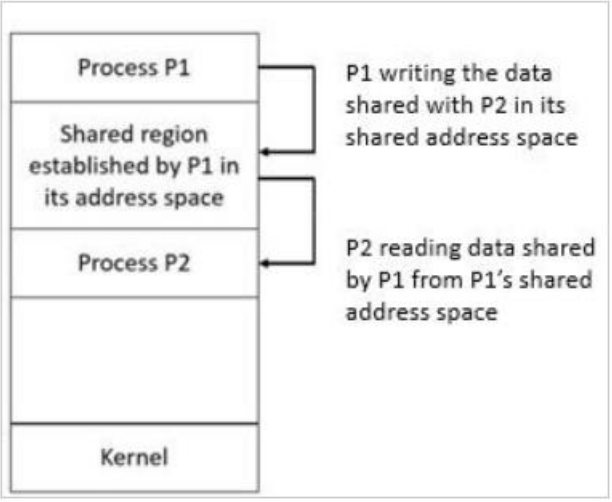
Two fundamental models for inter process communication:

1.Shared Memory
2.Message passing

1. Shared Memory:

- Shared memory system is the fundamental model of inter process communication.
- Shared memory concept works on fastest inter process communication.
-
- In the Shared Memory system, the cooperating processes communicate, to exchange the data with each other
- The processes share data by reading and writing the data in the shared segment of the processes.

-
- Let us discuss it by considering two processes. The diagram is shown below



-
- Process P1 has some data to share with process P2
- it stores the data or information to be shared in its shared memory region.
- Now, P2 requires the information stored in the shared segment of P1
- Now, P2 can read out the data from there

- The two processes can exchange information by reading and writing data in the shared segment of the process.

The advantages of Shared Memory are as follows:

1.Shared memory is a faster inter process communication system.

2.It allows cooperating processes to access the same pieces of data concurrently.

3.Users can perform multiple tasks at a time.

4. Modularity is achieved in a shared memory system.

2. Message passing:

- In this method, processes communicate with each other without using any kind of shared memory.
- Message passing provides two operations which are as follows :

1.Send message

2.Receive messag

- For fixed size messages the system level implementation is straight forward but the programming task becomes more difficult
- The variable sized messages require a more system level implementation but the programming task becomes simpler.
- If process P1 and P2 want to communicate they need to send a message to and receive a message from each other

that means here a communication link exists between them.

- methods for logically implementing a link and the send() and receive() operations.

Advantages of Message Passing Model :

1.Easier to implement.
2.Easier to build massively parallel hardware.
3.Message passing libraries are faster and give high performance.

.....

******q12) what is thread?**

sol)

- A thread is a single sequential flow of execution of tasks of a process

- it is also known as thread of execution or thread of control.
- There is a way of thread execution inside the process of any operating system.
- there can be more than one thread inside a process.

Types of Threads:

- ✓ there are two types of threads
  1. Kernel level thread.
  2. User-level thread.

1. Kernel level thread:

- ✓ The kernel thread recognizes the operating system.
- ✓ There is a thread control block and process control block in the system for each thread and process in the kernel-level thread.

- ✓ the kernel-level thread is implemented by the operating system.
- ✓ The kernel-level thread offers a system call to create and manage the threads from user-space.

2.User-level thread:

- ✓ The operating system does not recognize the user-level thread
- ✓ User threads can be easily implemented and it is implemented by the user.

- ✓ If a user performs a user-level thread blocking operation, the whole process is blocked.
- ✓ The kernel level thread does not know nothing about the user level thread.

✓ The kernel-level thread manages user-level threads

******q13)thread vs process?**

sol)

S.NO	Process	Thread
1.	Process means any program is in execution.	Thread means a segment of a process.
2.	The process takes more time to terminate.	The thread takes less time to terminate.
3.	It takes more time for creation.	It takes less time for creation.
4.	It also takes more time for context switching.	It takes less time for context switching.
5.	The process is less efficient in terms of communication.	Thread is more efficient in terms of communication.
6.	Multiprogramming holds the concepts of multi-process.	We don't need multi programs in action for multiple threads because a single process consists of multiple threads.
7.	The process is isolated.	Threads share memory.
8.	The process is called the heavyweight process.	A Thread is lightweight as each thread in a process shares code, data, and resources.

8.	The process is called the heavyweight process.	A Thread is lightweight as each thread in a process shares code, data, and resources.
9.	Process switching uses an interface in an operating system.	Thread switching does not require calling an operating system and causes an interrupt to the kernel.
10.	If one process is blocked then it will not affect the execution of other processes	If a user-level thread is blocked, then all other user-level threads are blocked.
11.	The process has its own Process Control Block, Stack, and Address Space.	Thread has Parents' PCB, its own Thread Control Block, and Stack and common Address space.
12.	Changes to the parent process do not affect child processes.	Since all threads of the same process share address space and other resources so any changes to the main thread may affect the behavior of the other threads of the process.
13.	A system call is involved in it.	No system call is involved, it is created using APIs.
14.	The process does not share data with each other.	Threads share data with each other.

****q14)multithreading models?

sol)

- Multithreading allows the application to divide its task into individual threads.
- In multi-threads, the same process or task can be done by the number of threads
- The main drawback of single threading systems is that only one task can be performed at a time
- so to overcome the drawback of this single threading, there is multithreading that allows multiple tasks to be performed.
- we can say that there is more than one thread to perform the task in multithreading.

One Process
Multiple Threads

Multiple Process Multiple
Threads per Process