Assignment-3

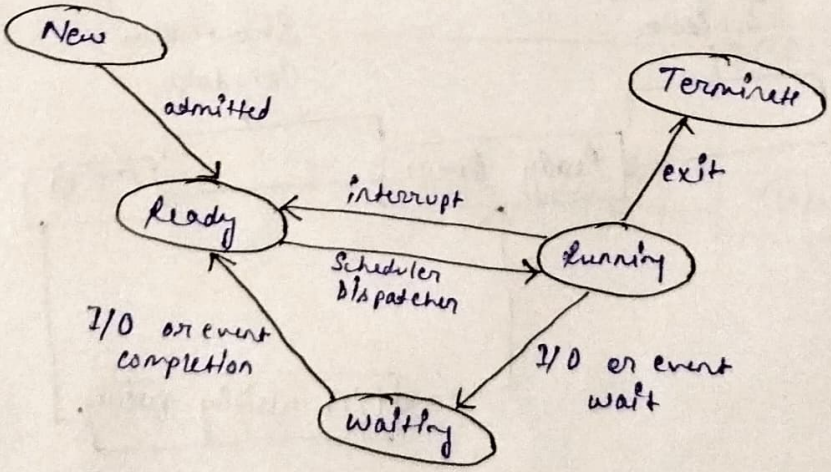
Ques1 Explain Process states with transition diagram in detail.

Ans A process is a program in execution and it is more than a program code called as text section and this concept works under all the operating system because all the task perform by the operating system needs a process to perform the task.

The process executes when it changes the state. The state of a process is defined by the current activity of the process. Each process may be in any one of the following states -

- New: The process is being created.
- Running: In this state the instructions are being executed.
- Waiting: The process is in waiting state until an event occurs like I/O operation completion or receiving a signal.
- Ready: The process is waiting to be assigned to a processor.
- Terminated: The process has finished execution.

Diagram →



Ques 2. Explain various Schedulers in detail.

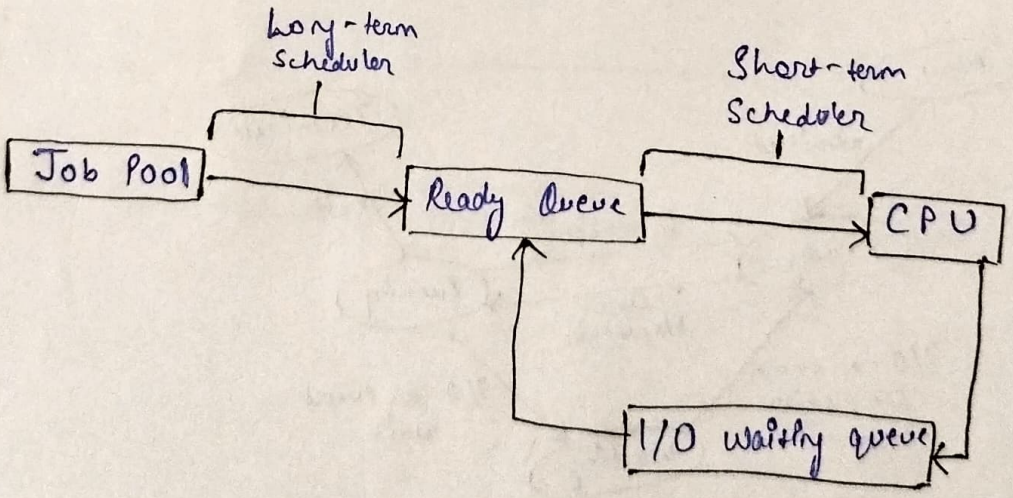
Ans There are three types of process schedulers:

i) Long Term or Job Scheduler

It brings the new process to the 'Ready State'. It controls the degree of multi-programming, i.e. the number of process present in a ready state at any point in time. It is important that the long-term scheduler make a careful selection of both I/O and CPU bound processes. The job scheduler increases efficiency by maintaining a balance between the two.

ii) Short-Term or CPU Scheduler

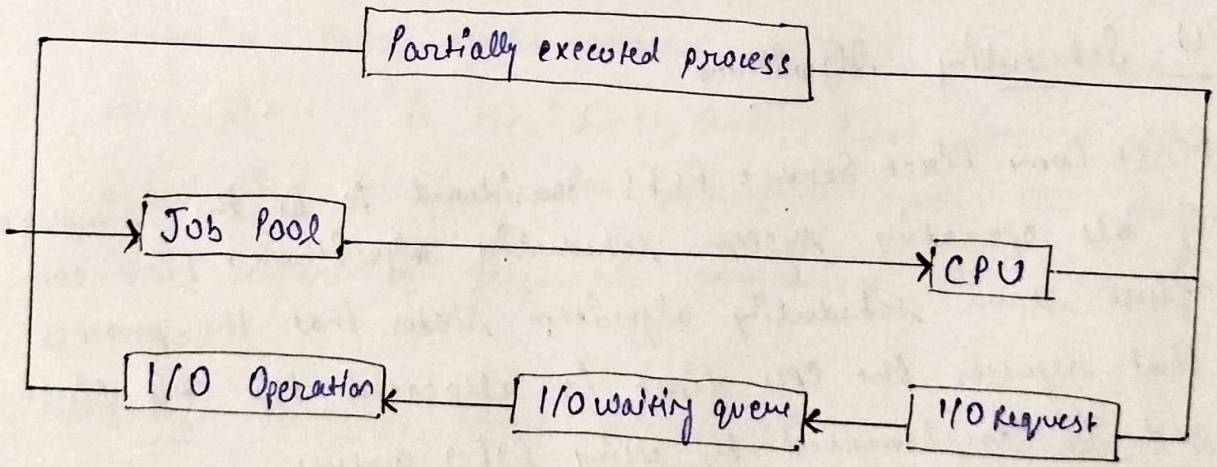
It is responsible for selecting one process from the ready state for scheduling it on the running state. Short term scheduler only selects the process to schedule it doesn't load the process on running. The CPU scheduler is responsible for ensuring no starvation due to high burst time processes.



The dispatcher is responsible for loading the process selected by the Short-term scheduler on the CPU (Ready to Running state) Context switching is done by the dispatcher only.

iii) Medium-Term Scheduler

It is responsible for suspending and resuming the process. It mainly does swapping (moving processes from main memory to disk and vice-versa). Swapping may be necessary to improve the process mix or because a change in memory requirements has overcommitted available memory, requiring memory to be freed up. It is helpful in maintaining a perfect balance between the I/O bound and the CPU bound. It reduces the degree of multiprogramming.



Ques 3. what is a CPU Scheduling? Explain various Scheduling algorithms in detail.

Ans CPU Scheduling is the process of deciding which process will own the CPU to use while another process is suspended. The main function of the CPU scheduling is to ensure that whenever the CPU remains idle, the OS has at least selected one of the processes available in the ready-to-use line.

There are mainly two types of scheduling methods:

- Preemptive Scheduling: Preemptive scheduling is used when a process switches from running state to ready state or from the waiting state to the ready state.
- Non-Preemptive Scheduling: Non-preemptive scheduling is used when a process terminates, or when a process switches from running state to waiting state.

CPU Scheduling Algorithms

i) First Come First Serve: FCFS considered to be the simplest of all operating system scheduling algorithms. First come first serve scheduling algorithm states that the process that requests the CPU first is allocated the CPU first and is implemented by using FIFO queue.

Characteristics:

- FCFS supports non-preemptive and preemptive CPU Scheduling algo.
- FCFS is easy to implement and use.
- This algorithm is not much efficient in performance and the wait time is quite high.

ii) Shortest Job First: SJF is a scheduling process that selects the waiting process with the smallest execution time to execute next. This scheduling method may or may not be preemptive. Significantly reduce the average waiting time for other processes waiting to be executed.

iii) Priority Scheduling: It is a preemptive method of CPU Scheduling Algorithm that works based on the priority of a process. In this algorithm, the editor sets the functions to be as important, meaning that the most important process must be done first. In the case of any conflict, that is, where there is more than one process with equal value, then the most important CPU planning algo. works on the basis of the FCFS algorithm.

iv) Round Robin CPU Scheduling: Round Robin is a CPU scheduling mechanism those cycles around assigning each task a specific time slot. It is the First Come, First Served CPU Scheduling technique with preemptive mode. The Round Robin CPU algorithm frequently emphasizes the Time-Sharing method.

v) Shortest Remaining Time: This is similar to Shortest Job First, except that if a new program is started, the OS compares the time it needs with the time the currently running program has left. If the new program would finish sooner, then the currently running program is switched out and the CPU starts processing the new program.

Ques4. Explain Threads and their management in detail.

Ans A thread is a single sequence stream within a process. Threads are also called lightweight processes as they possess some of the properties of processes.

Threads run in parallel improving the application performance. Each such thread has its own CPU state and stack, but they share the address space of the process and the environment.

Types of Thread in OS

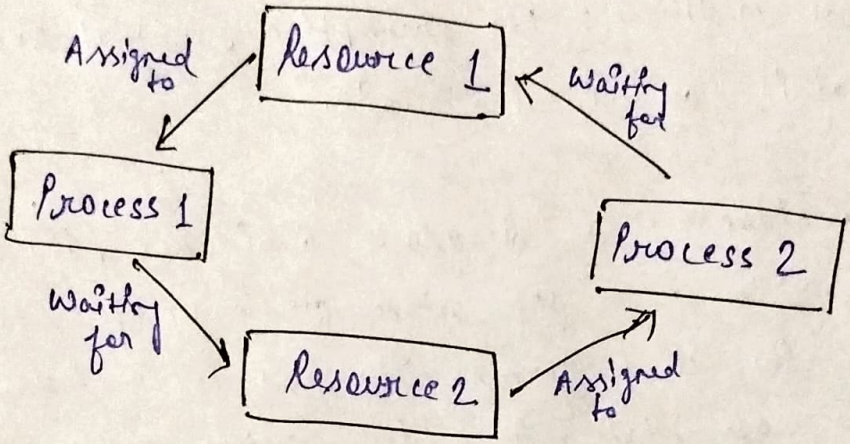
- User Level Thread
- Kernel Level Thread

User Level Threads : It is a type of thread that is not created using system calls. The kernel has no work in the management of user-level threads. User-level threads can be easily implemented by the user. In case when user-level threads are single-handed processes, kernel-level thread manages them.

Kernel Level Threads: A kernel level thread is a type of thread that can recognize the Operating system easily. Kernel Level Threads has its own thread table where it keeps track of the system. The Operating System Kernel helps in managing threads. Kernel Threads have somehow longer context switching time. Kernel helps in the management of threads.

Ques 5. Explain Deadlock and various methods to handle deadlock in detail.

Ans A deadlock is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.



Deadlock can arise if the following four conditions hold simultaneously (necessary conditions)

- Mutual Exclusion: Two or more resources are non-shareable (Only one process can use at a time).
- Hold and wait: A process is holding at least one resource and waiting for resources.
- No preemption: A resource cannot be taken from a process unless the process releases the resources.
- Circular Wait: A set of processes waiting for each other in circular form.

If any one of the above condition does not holds, then deadlock can be prevented.

Methods for handling deadlock

i) Deadlock Ignorance: If a deadlock is very rare, then let it happen and restart the system. This is the approach that both Windows and UNIX take, we use the ostrich algorithm for deadlock ignorance.

ii) Deadlock prevention: The deadlock can be prevented if any one of the necessary conditions of deadlock is violated, i.e. violate one of these conditions: No preemption, mutual exclusion, hold and wait or circular wait.

iii) Deadlock avoidance: In deadlock avoidance, the OS checks whether the system is in safe state or in unsafe state at every step which the OS performs. The process continues until the system is in safe state. Once the system moves to unsafe state, the OS has to backtrack one step.
In simple words, The OS reviews each allocation so that the allocation doesn't cause the deadlock in the system.

iv) Deadlock Detection and Recovery: This approach let the processes fall in deadlock and then periodically check whether deadlock occur in the system or not, if it occurs then it applies some of the recovery methods to the system to get rid of deadlock.