1. **FCFS (First Come First Serve):**

Home Work-1 Dtd:- 22/03/2023

Subject :- OS

* **Selection criteria :**

The process that request first is served first. It means that processes are served in the exact order of their arrival.

* **Decision Mode :**

Non preemptive: Once a process is selected, it runs until it is blocked for an I/O or some event, or it is terminated.

* **Implementation:**

This strategy can be easily implemented by using FIFO queue, FIFO means First In First Out. When CPU becomes free, a process from the first position in a queue is selected to run.

* **Advantages:**
  + Simple, fair, no starvation.
  + Easy to understand, easy to implement.
* **Disadvantages :**
  + Not efficient. Average waiting time is too high.
  + Convoy effect is possible. All small I/O bound processes wait for one big CPU bound process to acquire CPU.
  + CPU utilization may be less efficient especially when a CPU bound process is running with many I/O bound processes.

==================================================================================

1. **Shortest Job First (SJF):**

* **Selection Criteria :**

The process, that requires shortest time to complete execution, is served first.

* **Decision Mode :**

Non preemptive: Once a process is selected, it runs until either it is blocked for an I/O or some event, or it is terminated.

* **Implementation :**

This strategy can be implemented by using sorted FIFO queue. All processes in a queue are sorted in ascending order based on their required CPU bursts. When CPU becomes free, a process from the first position in a queue is selected to run.

* **Advantages:**
  + Less waiting time.
  + Good response for short processes.
* **Disadvantages :**
  + It is difficult to estimate time required to complete execution.
  + Starvation is possible for long process. Long process may wait forever.

==================================================================================

1. **Non Preemptive Priority Scheduling:**

* **Selection criteria :**

The process, that has highest priority, is served first.

* **Decision Mode:**

Non Preemptive: Once a process is selected, it runs until it blocks for an I/O or some event, or it terminates.

* **Implementation :**

This strategy can be implemented by using sorted FIFO queue. All processes in a queue are sorted based on their priority with highest priority process at front end. When CPU becomes free, a process from the first position in a queue is selected to run.

* **Advantages:**
  + Priority is considered. Critical processes can get even better response time.
* **Disadvantages:**
  + Starvation is possible for low priority processes. It can be overcome by using technique called ‘Aging’.
  + Aging***:*** gradually increases the priority of processes that wait in the system for a long time.

==========================================================================

1. **Preemptive Priority Scheduling:**

* **Selection criteria :**

The process, that has highest priority, is served first.

* **Decision Mode:**

Preemptive: When a new process arrives, its priority is compared with current process priority. If the new job has higher priority than the current, the current process is suspended and new job is started.

* **Implementation :**

This strategy can be implemented by using sorted FIFO queue. All processes in a queue are sorted based on priority with highest priority process at front end. When CPU becomes free, a process from the first position in a queue is selected to run.

* **Advantages:**
  + Priority is considered. Critical processes can get even better response time.
* **Disadvantages:**
  + Starvation is possible for low priority processes. It can be overcome by using technique called ‘Aging’.
  + Aging: gradually increases the priority of processes that wait in the system for a long time.
  + Context switch overhead is there.

==========================================================================

1. **Round Robin:**

* **Selection Criteria:**

Each selected process is assigned a time interval, called time quantum or time slice. Process is allowed to run only for this time interval. Here, two things are possible: First, Process is either blocked or terminated before the quantum has elapsed. In this case the CPU switching is done and another process is scheduled to run. Second, Process needs CPU burst longer than time quantum. In this case, process is running at the end of the time quantum. Now, it will be preempted and moved to the end of the queue. CPU will be allocated to another process. Here, length of time quantum is critical to determine.

* **Decision Mode:**

Preemptive: When a new process arrives, its priority is compared with current process priority. If the new job has higher priority than the current, the current process is suspended and new job is started.

* **Implementation :**

This strategy can be implemented by using circular FIFO queue. If any process comes, or process releases CPU, or process is preempted. It is moved to the end of the queue. When CPU becomes free, a process from the first position in a queue is selected to run.

* **Advantages:**
* One of the oldest, simplest, fairest and most widely used algorithms.
* **Disadvantages:**
* Context switch overhead is there.
* Determination of time quantum is too critical. If it is too short, it causes frequent context switches and lowers CPU efficiency. If it is too long, it causes poor response for short interactive process.

===========================================================================