Types Of Scheduling

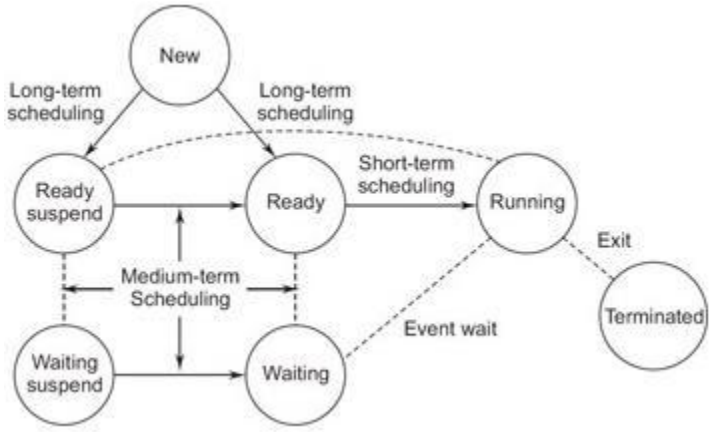
This is an article on *Types Of Scheduling* in *Operating System*.

The objective of multiprogramming is to have some process running at all times, to maximize CPU utilization. The objective of time sharing is to switch the CPU among processes so frequently. In uniprocessor only one process is running.A process migrates between various scheduling queues throughout its lifetime.The process of selecting processes from among these queues is carried out by a **scheduler**.The aim of processor scheduling is to assign processes to be executed by the processor. Scheduling affects the performance of the system, because it determines which process will wait and which will progress.

Types of Scheduling

Long-term Scheduling

Long term scheduling is performed when a new process is created. It is shown in the figure below. If the number of ready processes in the ready queue becomes very high, then there is a overhead on the operating system (i.e., processor) for maintaining long lists, context switching and dispatching increases. Therefore, allow only limited number of processes in to the ready queue. The "long-term scheduler" managers this. Long-term scheduler determines which programs are admitted into the system for processing. Once when admit a process or job, it becomes process and is added to the queue for the short-term scheduler. In some systems, a newly created process begins in a swapped-out condition, in which case it is added to a queue for the medium-term scheduler scheduling manage queues to minimize queueing delay and to optimize performance.



The long-term scheduler limits the number of processes to allow for processing by taking

the decision to add one or more new jobs, based on FCFS (First-Come, first-serve) basis or priority or execution time or Input/Output requirements. Long-term scheduler executes relatively infrequently.

Medium-term Scheduling

Medium-term scheduling is a part of the swapping function. When part of the main memory gets freed, the operating system looks at the list of suspend ready processes, decides which one is to be swapped in (depending on priority, memory and other resources required, etc). This scheduler works in close conjunction with the long-term scheduler. It will perform the swapping-in function among the swapped-out processes. Medium-term scheduler executes some what more frequently.

Short-term Scheduling

Short-term scheduler is also called as dispatcher. Short-term scheduler is invoked whenever an event occurs, that may lead to the interruption of the current running process. For example clock interrupts, I/O interrupts, operating system calls, signals, etc. Short-term scheduler executes most frequently. It selects from among the processes that are ready to execute and allocates the CPU to one of them. It must select a new process for the CPU frequently. It must be very fast.

Scheduling Criteria

Scheduling criteria is also called as scheduling methodology. Key to multiprogramming is scheduling.Different CPU scheduling algorithm have different properties .The criteria used for comapring these algorithms include the following:

- **CPU Utilization:**

  Keep the CPU as busy as possible. It range from 0 to 100%. In practice, it range from 40 to 90%.

- **Throughput:**

  Throughput is the rate at which processes are completed per unit of time.

- **Turnaround time:**

  This is the how long a process takes to execute a process. It is calculated as the time gap between the submission of a process and its completion.

- **Waiting time:**

  Waiting time is the sum of the time periods spent in waiting in the ready queue.

- **Response time:**

  Response time is the time it takes to start responding from submission time.It is calculated as the amount of time it takes from when a request was submitted until the first response is produced.

- **Fairness:**

  Each process should have a fair share of CPU.
  Non-preemptive Scheduling :

  In non-preemptive mode, once if a process enters into running state, it continues to execute until it terminates or blocks itself to wait for Input/Output or by requesting some operating system service.

  Preemptive Scheduling :

  In preemptive mode, currently running process may be interrupted and moved to the ready State by the operating system.
  When a new process arrives or when an interrupt occurs, preemptive policies may incur greater overhead than non-preemptive version but preemptive version may provide better service.

  It is desirable to maximize CPU utilization and throughput, and to minimize turnaround time, waiting time and response time.

Scheduling Algorithms

Scheduling algorithms or scheduling policies are mainly used for short-term scheduling. The main objective of short-term scheduling is to allocate processor time in such a way as to optimize one or more aspects of system behavior.
For these scheduling algorithms assume only a single processor is present. Scheduling algorithms decide which of the processes in the ready queue is to be allocated to the CPU is basis on the type of scheduling policy and whether that policy is either preemptive or non-preemptive. For scheduling arrival time and service time are also will play a role. List of scheduling algorithms are as follows:

- **First-come, first-served scheduling (FCFS) algorithm**
- **Shortest Job First Scheduling (SJF) algorithm**
- **Shortest Remaining time (SRT) algorithm**
- **Non-preemptive priority Scheduling algorithm**
- **Preemptive priority Scheduling algorithm**
- **Round-Robin Scheduling algorithm**
- **Highest Response Ratio Next (HRRN) algorithm**
- **Multilevel Feedback Queue Scheduling algorithm**
- **Multilevel Queue Scheduling algorithm**

For describing various scheduling policies, we would use the following information, present below:

Process|Arrival time|Service time (Burst time)|Priority

P1 |0 |4 |2
P2 |3 |6 |1
P3 |5 |3 |3
P4 |8 |2 |1

First-come First-served Scheduling (FCFS)

First-come First-served Scheduling follow first in first out method. As each process becomes ready, it joins the ready queue. When the current running process ceases to execute, the oldest process in the Ready queue is selected for running. That is first entered process among the available processes in the ready queue.The average waiting time for FCFS is often quite long. It is non-preemptive.

TURNAROUND TIME=WAITING TIME + SERVICE TIME

Advantages

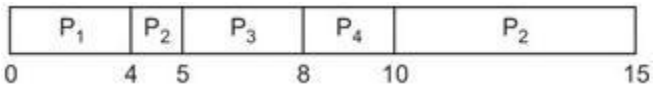
- Better for long processes
- Simple method (i.e., minimum overhead on processor)
- No starvation

Disadvantages

- Convoy effect occurs.Even very small process should wait for its turn to come to utilize the CPU. Short process behind long process reults in lower CPU utilization.
- Throughput is not emphasized.

Shortest Job First Scheduling (SJF)

This algorithm associates ith each process the length of the next CPU burst.Shortest-job-first scheduling is also called as shortest process next (SPN). The process with the shortest expected processing time is selected for execution, among the available processes in the ready queue. Thus, a short process will jump to the head of the queue over long jobs. If the next CPU bursts of two processes are the same then FCFS scheduling is used to break the tie.SJF scheduling algorithm is probably optimal. It givs the minimum average time for a given set of processes.It cannot be implemented at the level of short term CPU scheduling. There is no way of knowing the shortest CPU burst.
SJF can be premptive or non-preemptive.
A premptive SJF algorithm will preempt the currently executing process if the next CPU burst of newly arrived process may be shorter than what is left to the currently executing process.
A Non-premptive SJF algorithm will allow the currently running process to finish.Preemptive SJF Scheduling is sometimes called Shortest Remaining Time First algorithm.



Advantages

- It gives superior turnaround time performance to shortest process next because a short job is given immediate preference to a running longer job.
- Throughput is high.

Disadvantages

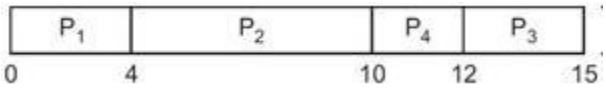
- Elapsed time (i.e., execution-completed-time) must be recorded, it results an additional overhead on the processor.
- Starvation may be possible for the longer processes.

Priority Scheduling

The SJF is a special case of general priority scheduling algorithm.
A Priority (an integer) is associated with each process.The CPU is allocated to the process with the highest priority.Generally smallest integer is considered as the highest priority.Equal priority processes are scheduled in First Come First serve order.It can be preemptive or Non-preemptive.

Non-preemptive Priority Scheduling

In this type of scheduling the CPU is allocated to the process with the highest priority after completing the present running process.



Advantage

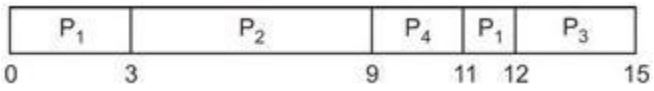
- Good response for the highest priority processes.

Disadvantage

- Starvation may be possible for the lowest priority processes.

Preemptive Priority Scheduling

In this type of scheduling the CPU is allocated to the process with the highest priority immediately upon the arrival of the highest priority process. If the equal priority process is in running state, after the completion of the present running process CPU is allocated to this even though one more equal priority process is to arrive.



Advantage

- Very good response for the highest priority process over non-preemptive version of it.

Disadvantage

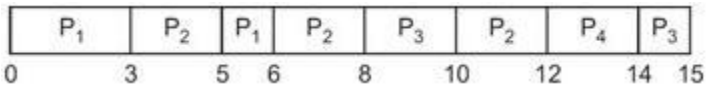
- Starvation may be possible for the lowest priority processes.

Round-Robin Scheduling

This type of scheduling algorithm is basically designed for time sharing system. It is similar to FCFS with preemption added.Round-Robin Scheduling is also called as time-slicing scheduling and it is a preemptive version based on a clock. That is a clock interrupt is generated at periodic intervals usually 10-100ms. When the interrupt occurs, the currently running process is placed in the ready queue and the next ready job is selected on a First-come, First-serve basis. This process is known as time-slicing, because each process is given a slice of time before being preempted.
One of the following happens:

- The process may have a CPU urst of less than the time quantum or
- CPU burst of currently executing process be longer than the time quantum. In this case the a context switch occurs the process is put at the tail of the ready queue.

In round-robin scheduling, the principal design issue is the length of the time quantum or time-slice to be used. If the quantum is very short, then short processes will move quickly.



Advantages

- Round-robin is effective in a general-purpose, times-sharing system or transaction-processing system.
- Fair treatment for all the processes.
- Overhead on processor is low.
- Overhead on processor is low.
- Good response time for short processes.

Disadvantages

- Care must be taken in choosing quantum value.
- Processing overhead is there in handling clock interrupt.
- Throughput is low if time quantum is too small.

Performance of RR Scheduling

- If there are n processes in the ready queue and time quantum is q, then each process gets 1/n of the CPU time in chunks of at most q time units at once.
- No process waits for more than (n-1)*q time units until the next time quantum.

- The performance of RR depends on time slice.If it is large then it is the same as FCFS. If q is small then overhead is too high.

Source: http://www.go4expert.com/articles/types-of-scheduling-t22307/