IJITKM Special Issue (ICFTEM-2014) May 2014 pp. 136-139 (ISSN 0973-4414)

CPU Scheduling –A Review

Rajesh Garg[1], Lucky[2], Anil Raman[3]
[1]Lect.(ECE Deptt.),Damla Polytechnic for Engineering,
[2]Lect(ECE Deptt.),Ganpati polytechnic for Engineering,
[3]Lect(CSE Deptt.),Ganpati polytechnic for Engineering,
[1]rajesh_damla@yahoo.co.in,[2]lucky.bakshi27@gmail.com,[3]raman.anil1984@gmail.com

Abstract:-Operating system performs variety of tasks in which scheduling is one of the basic task. Scheduling is heart of any computer systems as it contains the decision of giving resources between possible processes. Sharing of computer resources among multiple processes is also called scheduling. CPU scheduling is the basis of multi programmed operating systems. Switching of CPU among different processes make the computer more productive. The objective of multiprogramming is to have some process running at all the time to maximize the CPU utilization. Different scheduling techniques are employed that enable faster and efficient process execution thereby reducing the waiting time and increases the CPU utilization. In a uniprocessor system only one process run at a time; any other process must wait until the CPU is free and can be rescheduled.The problem of determining when processor should be assigned and to which process is CPU scheduling.

1. Introduction

In the past, most computers ran standalone, and most operating system was designed to run on a single processor. This situation is rapidly changing into one in which computers are networked together, making distributed operating system more important, Computer software can be roughly divided into two kinds, the system program, which manage the operation of the computer itself, and the application program, which solve problems for their user. The most fundamental of the system program is the Operating System, which controls all the computer's resources and provides the base upon which the application program can be written. The 1960's definition of an operating system is "the software that controls the hardware". But due to a better definition, an operating system is an important part of almost every computer system. A major goal of an operating system is to hide all the complexities and give the programmer a more convenient set of instructions to work with.

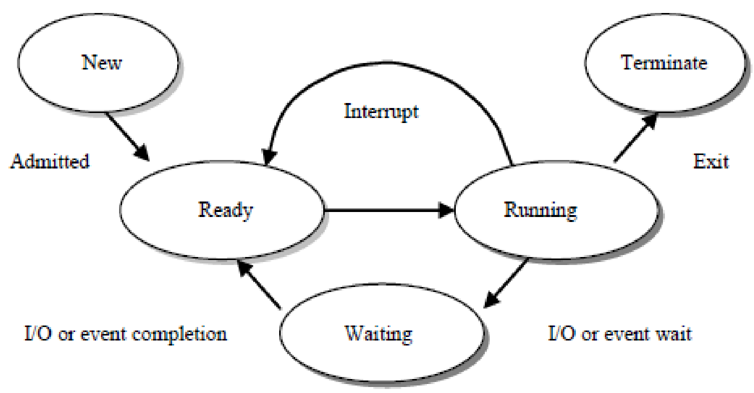

Figure1: Diagram of process state

To improve both the utilisation of CPU and the speed of its response to users, the computer must keep several processes in memory. Many memory schemes exist and each algorithm depends on situation. CPU scheduling deals with the problem of deciding which of the process in the ready queue is to be allocated the CPU.
The success of a CPU scheduler depends on the design of high quality scheduling algorithm. High-quality CPU scheduling algorithms rely mainly on criteria such as CPU utilization rate, throughput, turnaround time, waiting time and response time.

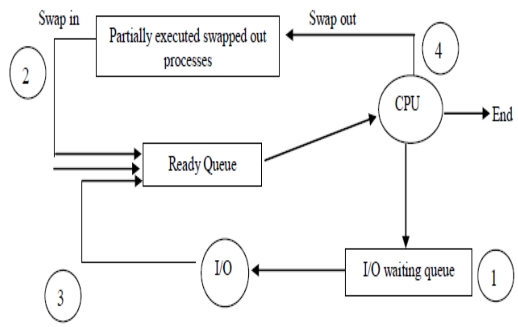

Figure 2: Shows the following states have been executed in the CPU Scheduler
1. When a process switches from the running state to the waiting state.
2. When a process switches from the running state to the ready state.
3. When a process switches from the waiting state to the ready state.
4. When a process terminates.

2. Scheduling Criteria

Different CPU scheduling algorithms have different properties, and the choice of a particular algorithm may favor one class of processes over another. In choosing which algorithm to use in a particular situation, we must consider the properties of the various algorithms. Many criteria have been suggested for comparing CPU scheduling algorithms. Which characteristics are used for comparison can make a substantial difference in which algorithm is judged to be best. The criteria include the following:

Utilization/Efficiency: keep the CPU busy 100% of the time with useful work.

Throughput: maximize the number of jobs processed per hour.

Turnaround time: from the time of submission to the time of completion, minimize the time batch users must wait for output.

Waiting time: Sum of times spent in ready queue - Minimize this.

Response Time: time from submission till the first response is produced, minimize response time for interactive users.

Fairness: make sure each process gets a fair share of the CPU.

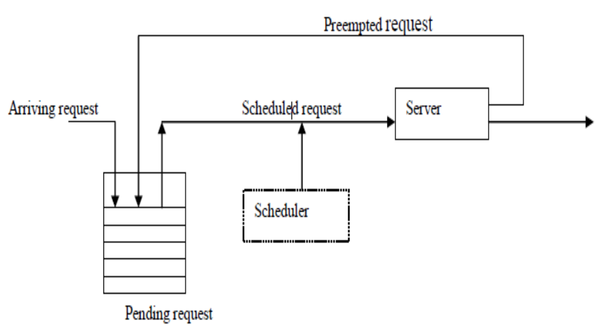

Figure 3: Schematic of Scheduling

It may be implemented as a FIFO queue, priority queue, a tree or simply an unordered linked list. Conceptually, however, all the processes in the ready queue are lined up waiting for a chance to run on the CPU. An operating system must allocate computer resources among the potentially competing requirements of multiple processes. In the case of the processor, the resource to be allocated is execution time on the processor and the means of allocation is scheduling. This way, the scheduler is the component of the operating system responsible to grant the right to CPU access to a list of several processes ready to execute. This idea is illustrated in the five state diagram of figure 4

2.1 Scheduling Algorithms

2.1.1 First Come First Serve

The most intuitive and simplest technique is to allow the first process submitted to run first. This approach is called as first-come, first-served(FCFS) scheduling. In effect, processes are inserted into the tail of a queue when they are submitted[1]. The next process is taken from the head of the queue when each finishes running. This idea is illustrated in the four state diagram of figure 4.

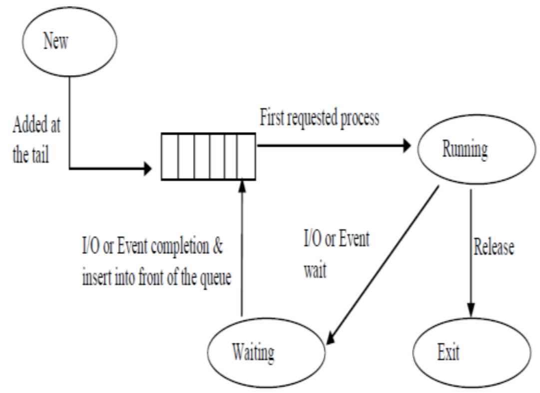

Figure 4: First Come First Serve Scheduling

The lack of prioritization does permit every process to eventually complete, hence no starvation.
Turnaround time, waiting time and response time is high.
One, Process with longest burst time can monopolize CPU, even if other process burst time is too short. Hence throughput is low [3].

2.1.2 Non preempted Shortest Job First

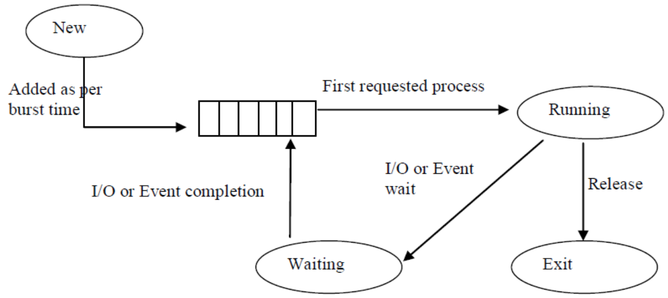

Figure 5: Shortest Job First Scheduling

The real difficulty with the SJF algorithm is, to know the length of the next CPU request. SJF minimizes the average waiting time[3] because it services small processes before it services large ones.

2.1.2 Round Robin

The Round Robin (RR) scheduling algorithm assigns a small unit of time, called time slice or quantum. The ready processes are kept in a queue. The scheduler goes around this queue, allocating the CPU to each process for a time interval of assigned quantum. New processes are added to the tail of the queue [4]. This idea is illustrated in the four state diagram of figure 6.

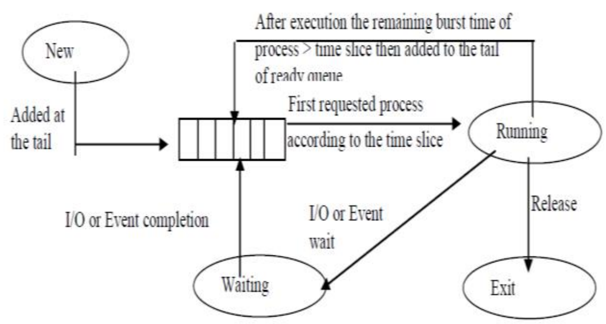

Figure 6: Round Robin Scheduling

Setting the quantum too short causes too many context switches and lower the CPU efficiency.
Setting the quantum too long may cause poor response time and approximates FCFS. Because of high waiting times, deadlines are rarely met in a pure RR system.

2.1.3 Priority Scheduling

The O/S assigns a fixed priority rank to each process. Lower priority processes get interrupted by incoming higher priority processes. This idea is illustrated in the four state diagram of figure 7

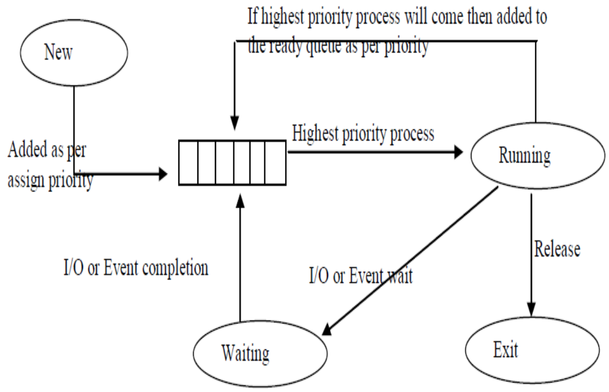

Figure 7: Priority Scheduling

Starvation can happen to the low priority process.
The waiting time gradually increases for the equal priority processes [5].
Higher priority processes have smaller waiting time and response time.

3. Conclusion

The treatment of shortest process in SJF scheduling tends to result in increased waiting time for long processes. And the long process will never get served, though it produces minimum average waiting time and average turnaround time. It is recommended that any kind of simulation for any CPU scheduling algorithm has limited accuracy. The only way to evaluate a scheduling algorithm to code it and has to put it in the operating system, only then a proper working capability of the algorithm can be measured in real time systems.

References

[1] SukumarBabuBandraupalli, A Novel CPU Scheduling Algorithm –Preemptive & Non-Preemptive- International Journal of Modern Engineering Research (IJMER) www.ijmer.com Vol.2, Issue.6, Nov-Dec. 2012 pp-4484-4490 ISSN: 2249-6645

[2] M.V. PanduranagaRao,K.C. Shet,A Research In Real Time Scheduling Policy For Embedded System Domain Clei Electronic Journal, Volume 12, Number 2, Paper 4, August 2009

[3] Maria Abur, A Critical Simulation of CPU Scheduling Algorithm using Exponential Distribution, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 6, No 2, November 2011 ISSN (Online): 1694-0814 www.IJCSI.org

[4] SukanyaSuranauwarat, A CPU Scheduling Algorithm Simulator

[5] Ashok Kumar ,Analysis of CPU Scheduling Policies through Simulation- International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 5, May 2013 ISSN: 2277 128X www.ijarcsse.com