International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 1, January 2014)

Analysis and Comparison of CPU Scheduling Algorithms

Pushpraj Singh[1], Vinod Singh[2], Anjani Pandey[3]
[1,2,3]Assistant Professor, VITS Engineering College Satna (MP), India

***Abstract*— Scheduling is a fundamental operating system function, since almost all computer resources are scheduled before use. The CPU is one of the primary computer resources. Central Processing Unit *(*CPU) scheduling plays an important role by switching the CPU among various processes. A processor is the important resource in computer; the operating system can make the computer more productive. The purpose of the operating system is that to allow the process as many as possible running at all the time in order to make best use of CPU. The high efficient CPU scheduler depends on design of the high quality scheduling algorithms which suits the scheduling goals. In this paper, we reviewed various fundamental CPU scheduling algorithms for a single CPU and shows which algorithm is best for the particular situation.**

***Keywords*— burst time, CPU scheduling, operating system, Scheduling Algorithm, turnaround time, waiting time.**

I. INTRODUCTION

Scheduling is a fundamental operating-system function. Almost all computer resources are scheduled before use. The CPU is one of the primary computer resources. Thus, its scheduling is central of operating system design. Thus its scheduling algorithm is heart of the OS design. When more than one process is runable, the OS must decide which one is to run first. That part of the OS concerned with this decision is called scheduler and the algorithm it uses is called scheduling algorithm. A CPU scheduler is the part of an operating system and responsible for arbitrating access to the CPU [6]. A scheduler is an OS module that selects the next job to be admitted into the system and the next process to run.

An operating system has three types of schedulers- long term scheduler (also known as Job scheduler), medium-term scheduler and short-term scheduler (also known as dispatcher and CPU scheduler).

A) Long-term scheduler

The long-term scheduler (also known as admission scheduler) decides which jobs or processes are to be admitted to the ready queue (in the Main Memory); that is, when an attempt is made to execute a program, its admission to the set of currently executing processes is either authorized or delayed by the long-term scheduler.

Thus, this scheduler dictates what processes are to run on a system, and the degree of concurrency to be supported at any one time - i.e.: whether a high or low amount of processes are to be executed concurrently, and how the split between input output intensive and CPU intensive processes is to be handled. So long term scheduler is responsible for controlling the degree of multiprogramming. In modern operating systems, this is used to make sure that real time processes get enough CPU time to finish their tasks. Without proper real time scheduling, modern GUIs would seem sluggish. The long term queue exists in the Hard Disk or the "Virtual Memory"

B) Medium-term scheduler

The medium-term scheduler (also known as mid-term scheduler) may decide to swap out a process which has not been active for some time, or a process which has a low priority, or a process which is taking up a large amount of memory in order to free up main memory for other processes, swapping the process back in later when more memory is available, or when the process has been unblocked and is no longer waiting for a resource. [Stallings, 396] [Stallings, 370]

C) Short-term scheduler

The short-term scheduler (also known as CPU scheduler) selects from among the pool of process resident in memory that are ready to execute, and allocates the CPU to one of them. Thus the short-term scheduler makes scheduling decisions much more frequent than the long-term or mid-term schedulers. This scheduler can be preemptive, implying that it is capable of forcibly removing processes from a CPU when it decides to allocate that CPU to another process, or non pre-emptive (also known as "voluntary" or "co-operative"), in that case the scheduler is unable to force processes off the CPU [6].

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 1, January 2014)

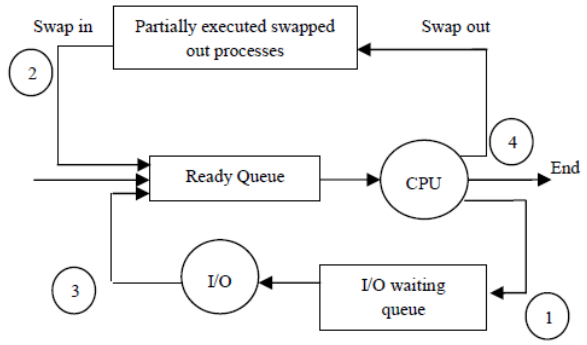

Figure I: Process of Schedulers

Figure I, 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.

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. Thus, the main impetus of this work is to develop a generalized optimum high quality scheduling algorithm suited for all types of job.

II. Scheduling Algorithms

A. Algorithms and Its Characteristics

The fundamental scheduling algorithms and its characteristics are described in this section.

a. 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 [2]. The next process is taken from the head of the queue when each finishes running.

Characteristics

- Since context switches only occur upon process termination, and no reorganization of the process queue is required, scheduling overhead is minimal.
- 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].

b. Shortest Job First

The process is allocated to the CPU which has least burst time. A scheduler arranges the processes with the least burst time in head of the queue and longest burst time in tail of the queue. This requires advanced knowledge or estimations about the time required for a process to complete [2]. This algorithm is designed for maximum throughput in most scenarios.

Characteristics

- The real difficulty with the SJF algorithm is, to know the length of the next CPU request.
- Starvation is possible, especially in a busy system with many small processes being run.
- SJF minimizes the average waiting time [3] because it services small processes before it services large ones. While it minimizes average wait time, it may penalize processes with high service time requests. If the ready list is saturated, then processes with large service times tend to be left in the ready list while small processes receive service. In extreme case, when the system has little idle time, processes with large service time will never be served. This total starvation of large processes is a serious liability of this algorithm.

c. 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].

Characteristics

- 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.
- RR scheduling involves extensive overhead, especially with a small time unit.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 1, January 2014)

d. Priority Scheduling

The OS assigns a fixed priority rank to every process, and the scheduler arranges the processes in the ready queue in order of their priority. Lower priority processes get interrupted by incoming higher priority processes.

Characteristics

- 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.
- Overhead is not minimal, nor is it significant.

B. Computation of Gantt chart, Waiting Time and Turnaround Time

Consider the following set of processes, with the length of the CPU-burst time in milliseconds is shown in Table I.

TABLE I
PROCESS WITH ITS ID AND BURST TIME

Process ID	Burst Time (ms)
P_1	10
P_2	2
P_3	8
P_4	6

a. First Come First Serve

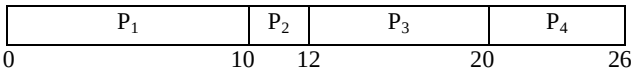

Figure II: Gantt chart for FCFS

b. Shortest Job First

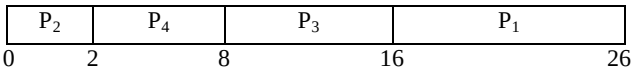

Figure III: Gantt chart for SJF

c. Round Robin

Assign time quantum as 5 ms for each process

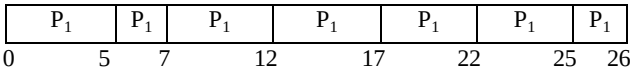

Figure IV: Gantt chart for RR

d. Priority Scheduling

Priority is assigned for each process as follows:

TABLE II
PROCESS WITH ITS ID, BURST TIME AND PRIORITY

Process ID	Burst Time (ms)	Priority
P_1	10	3
P_2	2	1
P_3	8	4
P_4	6	2

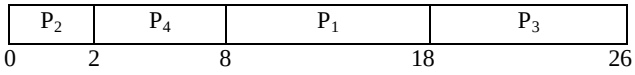

Figure V: Gantt chart for PS

For example, turnaround time for the process is calculated as time of submission of a process to the time of completion of the process is obtained through Gantt chart for SJF scheduling. Turnaround time for process P_1, P_2, P_3 and P_4 is observed as 26, 2, 16 & 8 and average turnaround time is (26+2+16+8)/4=13 ms.

The waiting time for the process is calculated as time taken by the process to wait in the ready queue is observed from Gantt chart for SJF scheduling. Waiting time for process P_1, P_2, P_3 and P_4 is obtained as 16, 0, 8 & 2 respectively and average waiting time is (16+0+8+2)/4 = 6.5 ms.

Similarly the turnaround time and waiting time is calculated for all other algorithms and summarized in Table III and Table IV.

From the above discussion it is clear that First Come First Serve (FCFS) & Shortest Job First (SJF) is generally suitable for batch operating systems and Round Robin (RR) & Priority Scheduling (PS) is suitable for time sharing systems. SJF algorithm is optimum for all type of scheduling algorithm. Hence, it is an algorithm with an optimum criteria and suitable for all scenarios.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 1, January 2014)

TABLE III
WAITING TIME FOR INDIVIDUAL PROCESS AND AVERAGE WAITING TIME FOR EACH SCHEDULING

Process ID	WAITING TIME (ms)			
	FCFS	SJF	Round Robin	Priority Scheduling
P_1	0	16	12	8
P_2	10	0	5	0
P_3	12	8	17	18
P_4	20	2	20	2
Avg Waiting Time	10.5	6.5	13.5	7

TABLE IV
TURNAROUND TIME FOR INDIVIDUAL PROCESS AND AVERAGE TURNAROUND TIME FOR EACH SCHEDULING

Process ID	TURNAROUND TIME (ms)			
	FCFS	SJF	Round Robin	Priority Scheduling
P_1	10	26	22	18
P_2	12	2	7	2
P_3	20	16	25	26
P_4	26	8	26	8
Avg Turnaround Time	17	13	20	13.5

III. Results And Discussion

The fundamental scheduling algorithm has been simulated and justified with C++ code. Comparison of the fundamental algorithm is shown in Figure VI and Figure VII.

i. Waiting Time

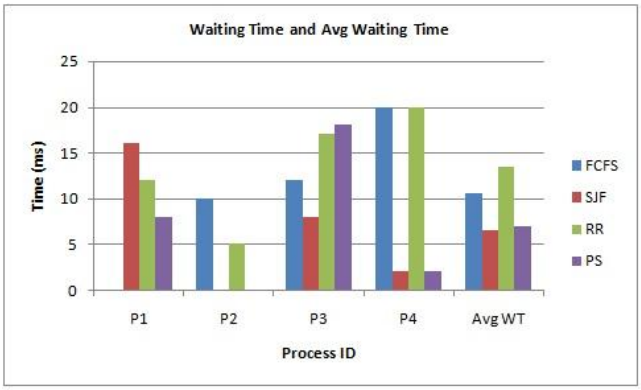

Figure VI: Comparison of Fundamental scheduling Algorithm-Waiting Time

ii. Turnaround Time

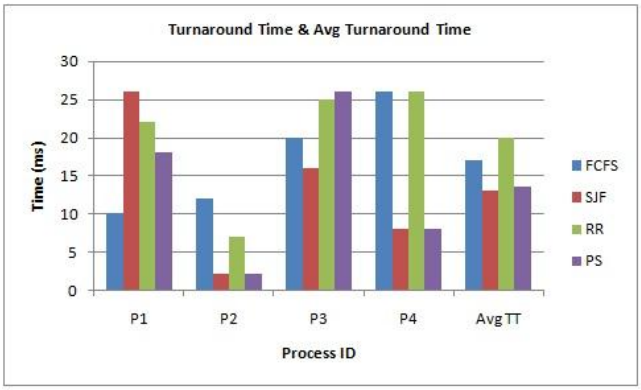

Figure VII: Comparison of Fundamental scheduling Algorithm-Turnaround Time

It is clearly observed that turnaround time, waiting time and response time of the processes are optimum for SJF scheduling algorithm compared to all other fundamental algorithms from Figure II, Figure. III, Fig. IV, Fig. V, (Figure II to Figure V are Gantt Chart) Figure VI and Figure VII. It can also be observed that throughput and CPU utilization rate are optimum.

From above analysis and discussion, we can say that the FCFS is simple to understand and suitable only for batch system where waiting time is large. SJF scheduling algorithm gives minimum average waiting time and average turnaround time. The priority scheduling algorithm is based on the priority in which the highest priority job can run first and the lowest priority job need to wait though it will create a problem of starvation.

The round robin scheduling algorithm is preemptive which is based on round robin policy one of the scheduling algorithm which follows the interactive system and the round robin scheduling algorithm is deal with the time sharing system.

IV. CONCLUSIONS

The SJF scheduling algorithm is to serve all types of job with optimum scheduling criteria. 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.

The shortest job first scheduling algorithm deals with different approach, in this algorithm the major benefit is it gives the minimum average waiting 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] Sukanya Suranauwarat, "A CPU Scheduling Algorithm Simulator", October 10-13, 2007, Milwaukee, WI 37th ASEE/IEEE Frontiers in Education Conference.. .

[2] Abraham Silberschatz, Peter Baer Galvin, Greg Gangne;" Operating System Concepts", edition-6, 2002, John Wiley and Sons, INC.

[3] Mr. Umar Saleem and Dr. Muhammad Younus Javed, "Simulation of CPU Scheduling Algorithm", 0-7803-6355-8/00/ $10.00 @ 2000 IEEE.

[4] Sun Huajin', Gao Deyuan, Zhang Shengbing, Wang Danghui; "Design Fast Round Robin Scheduler in FPGA", 0-7803-7547-5/021/$17.00 @ 2002 IEEE.

[5] Md. Mamunur Rashid and Md. Nasim Adhtar; "A New Multilevel CPU Scheduling Algorithm", Journals of Applied Sciences 6 (9): 2036-2039, 2009.

[6] Mr. Sindhu M, Mr. Rajkamal R and Mr. Vigneshwaran P, "An Optimum Multilevel CPU Scheduling Algorithm". 978-0-7695-4058-0/10 $26.00 © 2010 IEEE

[7] Black, D.L., 1985. Scheduling support for concurrency and parallelism in the mach operating system. Computer, 23:123-126.

[8] Ankur Bhardwas, Rachhpal Singh, Gaurav, "Comparative Study of Scheduling Algorithm in Operating System" International Journal of Computer and Distributed Systems, Vol. No. 3, Issue I, April-May 2013.