RESEARCH ARTICLE OPEN ACCESS

A Comparative Study of Fixed Length Time-Sliced Round Robin Scheduling and Priority Based Variable Length Time Sliced Round Robin Scheduling Algorithms

Suchishree Sengupta
Post Graduate
Department of Computer Science and Technology
University Of Calcutta
West Bengal - India

ABSTRACT
In multiprogramming systems, CPU scheduling is an important criteria. Various scheduling algorithms like FCFS, SJF, SRTF and RR are available. In this paper, a comparison between the fixed length time-sliced Round Robin Scheduling and Priority based variable length time sliced Round Robin Scheduling Algorithm is made in respect to average waiting time, turnaround time and response time of the processes entering the system.
Keywords:- RR, FCFS, SJF, SRTF, waiting time, response time, turnaround time, gantt chart, time-slice, priority.

I. INTRODUCTION

In a single processor system, only one process can run at a time; any other must wait until the CPU is free and can be rescheduled. The objective of multiprogramming is to have some process running at all times, to maximize CPU utilization. The idea is relatively simple. Several processes are kept in memory at one time. When one process has to wait the operating system takes away the CPU from that process and gives the CPU to another process. This pattern continues. Every time one process has to wait, the other process can take over use of CPU. Scheduling of this kind is a fundamental operating system function. Almost all resources are scheduled before use.[1]

II. CHEDULING CRITERIA

Different scheduling algorithms have different properties, and the choice of a particular algorithm may favour one class of processes over another. Many criteria have been suggested for comparing CPU scheduling algorithms. The criteria include the following [1]:

A. CPU utilization*:* The idea is to utilize the CPU as much as possible i.e maximize the utilization factor. Hence, the algorithm is selected keeping that factor in mind.

B. Throughput: Throughput is the measure of work done per unit time. The algorithm chosen must increase the throughput of the system

C. Turnaround time*: The* interval from the time of submission to the time of completion of a process is the turnaround time. The approach of the algorithm is to minimize the average turnaround time of the system.

D. Waiting time*:* Waiting time is the sum of the amount of time periods a process waits in the queue before it is being assigned the CPU. Lower the waiting time, better the algorithm.

E. Response time*:* The response time is the time from the submission of the process until the first response from the process is received. Algorithms with faster response time is better as an initial response from the process is essential which helps realize that the system has started giving responses.

III. SCHEDULING ALGORITHMS

In this paper two scheduling algorithms are discussed and their performances are measured on the basis of the above mentioned criteria.

A. Fixed Length Time-Slice Round-Robin (RR) scheduling**:** The Round Robin scheduling algorithm is designed specifically for time-sharing systems. It is similar to FCFS scheduling with pre-emption to switch between processes. A time quantum or time slice is defined which is generally from 10 to 100 milliseconds. The ready queue is treated as a circular queue where each process is assigned 1 time unit or time quantum [1]. The CPU scheduler picks the first process from the ready queue, sets a timer to interrupt after 1 time quantum, and dispatches the scheduler. Windows operating system uses RR scheduling.

B. Priority based variable length time slice Round-Robin Scheduling: This scheduling algorithm assigns different time quantum to different processes based on the priority of the processes. The process with highest priority is assigned the highest time quantum and gradually the time slice is decreased with decreasing level of process priority.

The comparison between the above mentioned algorithms is shown using the following set of processes as examples.

Example 1:

Consider the set of following processes with the length of the CPU bursts given in milliseconds.

Process	Burst Time	Priority
P1	10	1
P2	29	2
P3	3	3
P4	7	4
P5	12	5

The time quantum or times slice being 10 milliseconds. Using RR scheduling, we would schedule the processes using the following Gantt chart[1]:

P1	P2	P3	P4	P5	P2	P5	P2

0 10 20 23 30 40 50 52 61

The waiting time is 0 milliseconds for process P1, 32 milliseconds for process P2, 20 milliseconds for process P3, 23 milliseconds for process P4 and 40 milliseconds for process P5. Thus, the average waiting time is (0+32+20+23+40)/5=23 milliseconds.

The turnaround time for P1 is 10 milliseconds, P2 is 61 milliseconds, P3 is 23 milliseconds, P4 is 30 milliseconds and P5 is 52 milliseconds.

The response time for P1 is 0 milliseconds, P2 is 10 milliseconds, P3 is 20 milliseconds, P4 is 23 milliseconds and P5 is 30 milliseconds.

In the above scheduling process, we can see that priority is not taken into consideration.

Now in the priority based approach the process with highest priority is given more time slice as compared to other processes. For example, let's assign process P1 with time slice/time quantum=10 milliseconds, P2 with 8 milliseconds,P3 with 6 milliseconds,P4 with 4 milliseconds and P5 with 2 milliseconds gradually decreasing the time quantum from higher to lower priority of the processes. Using this algorithm, we would schedule the processes using the following Gantt chart:

P1	P2	P3	P4	P5	P2	P4	P5	P2	P5	P2	P5

10 18 21 25 27 35 38 40 48 50 55 61

The waiting time is 0 milliseconds for process P1, 26 milliseconds for process P2, 18 milliseconds for process P3, 31 milliseconds for process P4 and 49 milliseconds for process P5. Thus, the average waiting time is (0+26+18+31+49)/5=24.8 milliseconds.

The turnaround time for P1 is 10 milliseconds, P2 is 55 milliseconds, P3 is 21 milliseconds,P4 is 38 milliseconds and P5 is 61 milliseconds.

The response time for P1 is 0 milliseconds, P2 is 10 milliseconds, P3 is 18 milliseconds, P4 is 21 milliseconds and P5 is 25 milliseconds.

Example 2 :

Consider the next set of following processes with the length of the CPU bursts given in milliseconds.

Process	Burst Time	Priority
P1	10	1
P2	1	2
P3	2	3
P4	1	4
P5	5	5

The time quantum or times slice being 5 milliseconds. Using RR scheduling, we would schedule the processes using the following Gantt chart:

P1	P2	P3	P4	P5	P1

0 5 6 8 9 14 19

The waiting time is 9 milliseconds for process P1, 5 milliseconds for process P2, 6 milliseconds for process P3, 8 milliseconds for process P4 and 9 milliseconds for process P5. Thus, the average waiting time is (9+5+6+8+9)/5=7.4 milliseconds.

The turnaround time for P1 is 19 milliseconds, P2 is 6 milliseconds, P3 is 8 milliseconds, P4 is 9 milliseconds and P5 is 14 milliseconds.

The response time for P1 is 0 milliseconds, P2 is 5 milliseconds, P3 is 6 milliseconds, P4 is 8 milliseconds and P5 is 9 milliseconds.

Now in the priority based approach the process with highest priority is given more time slice as compared to other processes. For example, let's assign process P1 with time slice/time quantum=5 milliseconds, P2 with 4 milliseconds,P3 with 3 milliseconds,P4 with 2 milliseconds and P5 with 1 milliseconds gradually decreasing the time quantum from higher to lower priority of the processes. Using this algorithm, we would schedule the processes using the following Gantt chart:

P1	P2	P3	P 4	P 5	P1	P5

0 5 6 8 9 10 15 19

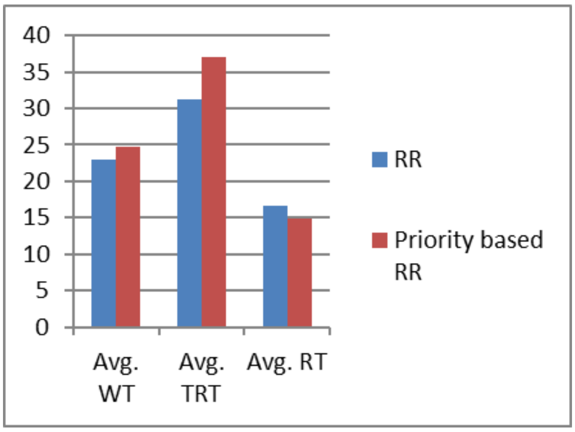
The waiting time is 0 milliseconds for process P1, 5 milliseconds for process P2, 6 milliseconds for process P3, 8 milliseconds for process P4 and 14 milliseconds for process P5. Thus, the average waiting time is (5+5+6+8+14)/5=7.6 milliseconds.

The turnaround time for P1 is 15 milliseconds, P2 is 6 milliseconds, P3 is 8 milliseconds, P4 is 9 milliseconds and P5 is 19 milliseconds.

The response time for P1 is 0 milliseconds, P2 is 5 milliseconds, P3 is 6 milliseconds, P4 is 8 milliseconds and P5 is 9 milliseconds.

IV. ERFORMANCE EVALUATION OF THE ALGORITHMS

Bar-chart showing average Waiting Time (WT), Turaround Time (TRT) and Response Time (RT) using example 1:



Bar-chart showing average Waiting Time (WT), Turaround Time (TRT) and Response Time (RT) using example 2:

Waiting time: The fixed length time slice RR scheduling algorithm performs little better when average waiting time for the processes is considered as compared to variable time slice RR scheduling. But individual waiting time of higher priority jobs is generally less as compared to lower priority jobs but results are highly dependent on the set of processes and the values of time slice being used.

Turnaround time: The individual turnaround time for higher priority jobs is less for the variable length time slice RR scheduling algorithm. The average turnaround time is little high for variable length time slice RR scheduling.

Response time: The average response time is better for variable length time slice RR scheduling for example 1 but no change is seen with example 2.

V. CONCLUSION

From the above comparative study it is seen that the average waiting time slightly varies for the two algorithms but higher priority jobs has to wait less amount of time in some cases. The response time and turnaround time is also in favour of highest priority jobs but it is dependent on the set of processes, their burst times and the value of time slice being used. Hence, it cannot be concluded that one algorithm out performs the other. But priority is an important criteria for real time systems. So the inclusion of priority in scheduling a job is definitely an improved approach because higher priority jobs needs to be completed earlier as compared to lower priority jobs.

REFERENCES

[1] A. Silberschatz, P. B. Galvin, and G. Gagne, "Operating System Concepts", 7th Edn. , John Wiley and Sons Inc, 2005

[2] Y.A. Adekunle, Z.O. Ogunwobi, A. Sarumi Jerry, B.T. Efuwape, Seun Ebiesuwa, and Jean-Paul Ainam , "A Comparative Study of Scheduling Algorithms for Multiprogramming in Real-Time Systems" , International Journal of Innovation and Scientific Research ISSN 2351-8014 Vol. 12 No. 1 Nov. 2014, pp. 180-185

[3] Miss. Jayashree S. Somani,Miss. Pooja K. Chhatwani, "Comparative Study of Different CPU Scheduling Algorithms", IJCSMC, Vol. 2, Issue. 11,November 2013, pg.310 –318

[4] Pushpraj Singh, Vinod Singh, Anjani Pandey ,"Analysis and Comparison of CPU Scheduling Algorithms" , International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 1, January 2014