International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-3, Issue-5, May 2017
ISSN: 2395-3470
www.ijseas.com

BEST TIME QUANTUM ROUND ROBIN CPU SCHEDULING ALGORITHM

Dolly Khokhar , Mr Ankur Kaushik

Meerut Institute Of Engineering And Technology, Meerut, India

Abstract- CPU Scheduling is one of the fundamental concepts of Operating System. The performance of the system depends on the CPU scheduling algorithms. The main aim of the CPU scheduling algorithms is to minimize waiting time, turnaround time, response time and context switching and maximizing CPU utilization. First-Come-First-Served (FCFS) Round Robin (RR), Shortest Job First (SJF) and, Priority Scheduling are some popular CPU scheduling algorithms. Round Robin (RR) CPU scheduling algorithm is optimal CPU scheduling algorithm in timeshared systems. The performance of the CPU depends on the selection of time quantum in timeshared systems. The objective of this paper is to make a change in round robin CPU scheduling algorithm so that the performance of CPU can be improved. The proposed algorithm is experimentally proven better than conventional RR. Round robin CPU scheduling algorithm has high context switch rates, large response time, large waiting time, large turnaround time and less throughput, these disadvantages can be improved with new proposed CPU scheduling algorithm.

Keywords*-Context switching, CPU scheduling, Gantt chart, Response time, Round Robin CPU scheduling algorithm, Turnaround time, Waiting time*

1. INTRODUCTION

Scheduling is the essential to any computer system since it contain decision of giving resources between possible processes. Sharing of computer resources between multiple processes is also called as scheduling. The CPU is one of the primary computer resources, so its scheduling is essential to an operating system's design. Efficient resource utilization is achieved by sharing system resources amongst multiple users and system process [1]. The operating system must decide through the scheduler the order of execution. Allocating CPU to a process requires careful consciousness to assure justice and avoid process starvation for CPU. Scheduling decision try to reduce the following: turnaround time, response time and average waiting time for processes and the number of context switches. Scheduling algorithms are the method by which a resource is allocated to a process or task.

CPU scheduling is the method by which a resource is allocated to a process or task and executes in different ways. For scheduling many scheduling algorithms are used like FCFS, SJF, RR, and Priority scheduling algorithm. The processes are scheduled according to the given burst time, arrival time and priority. The execution of processes used number of resources such as Memory, CPU etc. [2] .A scheduling decision refers to the theory of selecting the next process for execution. During each scheduling decision, a context switch occurs, meaning that the current process will stop its execution and put back to the ready queue and another process will be dispatched. Scheduling algorithms are widely used in communications networks and in operating systems to allocate resources to competing tasks.

The various scheduling parameters [3] for the selection of scheduling algorithm are:

- Utilization/Efficiency: keep the CPU busy 100% of the time with useful work
- Throughput: maximize the number of jobs processed per hour.

• Turnaround time: time required for a particular process to complete, from submission to completion- minimize this

• 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

2 Scheduling Algorithm

2.1 FIRST COME FIRST SERVED (FCFS) Scheduling- It is the simplest CPU Scheduling algorithm. The criteria of this algorithm is „the process that requests first, holds the CPU first" or which process enter the ready queue first is served first. The workload is processed in the order of arrival time, with no preemption . Once a process has been submitted to the CPU, it runs into completion without being interrupted. Such a technique is fair in the case of smaller processes but is quite unfair for long an unimportant job. Since FCFS does not involve context switching therefore it has minimal overhead. It has low throughput since long processes can keep processor occupied for a long time making small processes suffer. As a result waiting time, turnaround time and response time can be low .

2.2 Shortest Job First (SJF) Scheduling- The criteria of this algorithm are which process having the smallest CPU burst, CPU is assigned to that process next. If two process having the same CPU burst time FCFS is used to break up the tie [4]. SJF can be worked as preemptive and non–preemptive in nature based on the arrival time and burst time of the processes. SJF reduces average waiting time of the processes as compared to FCFS. SJF favors shorter processes over longer ones which is an overhead as compared to FCFS. It selects the job with the smallest burst time ensuing CPU availability for other processes as soon as the current process reaches its completion. This prevents smaller processes from suffering behind larger processes in the ready queue for a longer time .

2.3 Priority Based Scheduling- In this algorithm, priority is associated with each process and on the basis of that priority CPU is allocated to the processes. Higher priority processes are executed first and lower priority processes are executed at the end. If multiple processes having the same priorities are ready to execute, control of CPU is assigned to these processes on the basis of FCFS [5]. Priority Scheduling can be preemptive and non-preemptive in nature.

2.4 Round Robin (RR) Scheduling-It is a preemptive scheduling algorithm. It is designed especially for time sharing systems. In this algorithm, a small unit of time called time quantum or time slice is assigned to each process [6]. When the time quantum expired, the CPU is switched to another process. Performance of Round Robin totally depends on the size of the time quantum.

3 Previous Work Done

The performance of Round Robin Scheduling is sensitive to time quantum selection, because if time quantum is very large then Round Robin scheduling is same as the FCFS scheduling. If the time quantum is extremely too small then Round Robin scheduling is same as Processor Sharing algorithm and number of context switches is very high. Each value will lead to a specific performance and will affect the algorithm's efficiency by affecting the processes waiting time, turnaround time, response time, throughput, CPU utilization and number of context switch. A new algorithm is developed that describes improvement in Revamped Mean Round Robin scheduling algorithm. The algorithm is good because; it minimizes waiting time and turnaround time with respect to the Revamped Round Robin scheduling algorithm. The goal of this algorithm is to further modify the Revamped Mean Round Robin CPU scheduling algorithm by Sachin Kathuria, Piyush Pratap Singh, Pradyumn Tiwari, Prashant in order to increase the performance of the Round Robin scheduling

algorithm by affecting the average waiting time, average turnaround time.
In this paper, a comparison of RR, RMRR and new proposed algorithm has been done.

4 ROUND ROBIN CPU SCHEDULING ALGORITHM

The RR scheduling algorithm [7] is given by following steps:-
Step 1: The scheduler maintains a queue of ready processes and a list of blocked and swapped out processes.

Step 2: The Process Control Block of newly created process is added to end of ready queue. The Process Control Block of terminating process is removed from the scheduling data structures.

Step 3: The scheduler always selects the Process Control Block from the head of the ready queue.

Step 4: When a running process finishes its time slice, then it is moved to end of ready queue .

Step 5: The event handler performs the following actions:
a) When a process makes an input -output request or swapped out, its Process Control Block is removed from ready queue to blocked/swapped out list.
b) When I/O operation awaited by process is swapped in its Process Control Block or a process finishes is removed from blocked/swapped list to end of ready queue.
There are some disadvantages of round robin CPU scheduling algorithm for operating system which are as follows:
- *Static time quantum*
- *Larger waiting time and Response time*
- *Low throughput*

So it can be concluded that the round robin algorithm is not suitable for real time systems.

5 PROPOSED ALGORITHM

In our proposed algorithm, the number of processes is residing in the ready queue, we assume their arrival time is assigned to zero and burst times are allocated to the CPU. The burst time and the number of processes (n) are accepted as input. Now first of all we arrange all processes in increasing order according to their given burst time and choose best time slice with the help of mean and median. The best time slice will be depends on the mean and median of burst time.

In this algorithm some number of processes and their burst time are given. And we have to find out their turnaround time and their waiting time.

STEPS FOR MODIFIED ROUND ROBIN ALGORITHM

Step 1 - Start

Step 2 - Arrange the process in increasing order of the CPU burst time in ready queue.

Step 3 - Calculate the mean of the CPU burst time of all the process.

mean=(P1+P2+P3+.....Pn)/n

Step 4 - Calculate the median .

Step 5 - Calculate the best time quantum

B.T.Q= [mean + median]/2

Step 6 - Allocate the CPU to the first process in the ready queue that have less burst time.

Step 7 - If the remaining CPU burst time of the current process is less than or equal to B.T.Q then (a)- reallocate the CPU to the current process again for the remaining burst time. After the complete execution of the current process remove it from ready queue.

(b)- Otherwise remove the process from the ready queue and put it on the tail of the ready queue for further execution.

Step 8 - Pick the next process from the ready queue and allocate the CPU to it up to the B.T.Q and go to step 7 .

Step 9 - If the ready queue is empty then go to step 11.

Step 10 - If there is any process left in the ready queue allocate the CPU to that process otherwise go to step 3.

Step 11 - Calculate the average waiting time (AWT) , average turnaround time (ATAT), and number of context switches.

For example:

Consider five processes named P1, P2, P3, P4 and P5 with their CPU burst time.

Input Table

Process Name	Arrival time	Burst time
P1	0	7
P2	4	25
P3	10	5
P4	15	36
P5	17	18

Revamped Mean Round Robin Scheduling

Grantt Chart :

P1	P2	P3	P4	P5	P1

0 2 4 6 8 10 15

P2	P2	P3	P4	P4	P5

15 32 38 41 58 75 91

Context switches : 11

Average waiting Time : 33.80 ms

Average Turnaround Time: 42.80 ms

According to proposed algorithm:

Grantt Chart:

P1	P3	P5	P2	P4

0 7 12 30 55 91

Context switches : 4

Average waiting Time : 20.8ms

Average Turnaround Time: 29.8 ms

Comparison Table

Algorithm	Avg Waiting Time	Avg Turnaround Time	No. Of Context Switches
RMRR	33.80	42.80	11
BEST TIMEQUANTUM RR	20.8	29.80	4

6 Conclusion

One of the important tasks of the operating system is the allocation of CPU to the processes waiting for execution. Many CPU scheduling algorithms have been presented with some advantages and disadvantages. An algorithm with best time quantum is proposed in this paper giving better performance than conventional RR algorithm.The waiting time and turnaround time have been reduced in the proposed scheduling algorithm and hence the system performance has been improved.

REFERENCES

[1]Operating System concepts 7th Edition, By Galvin, Silberschatz, Gange John Wiley & Sons, 2005

[2]Seltzer, M P. Chen and J outerhout, 1990.Disk scheduling revisited in USENIX. Winter technical conference. Shamim H M 1998. Operating system, DCSA-2302

[3]E.O. Oyetunji, A. E. Oluleye," Performance Assessment of Some CPU Scheduling Algorithms", Research Journal of Information Technology,*1(1): pp 22-26, 2009*

[4]P. Balakrishna Prasad, "Operating Systems" Second Edition.

[5]M. Dietel, "Operating Systems", Pearson Education, Second Edition.

[6]http://en.wikipedia.org/wiki/Scheduling

[7] Rakesh Kumar Yadav, Abhishek K Mishra, Navin Prakash, Himanshu Sharma," An Improved Round Robin Scheduling Algorithm for CPU Scheduling", (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 1064-1066, 2010