International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 8, Issue 2 (August 2013), PP. 57-61

Modified Round Robin Scheduling Algorithm Using Variable Time Slice

[1]Suman Bhatia, [2]Supriya Raheja
[1,2]Department of CSE & ITITM University Gurgaon, India

Abstract:- CPU scheduling is the basis of multi-programmed operating systems. The main objective of scheduling algorithms is to reduce the response time, waiting time and to increase the throughput and CPU utilization. Round Robin assumes that all processes are equally important; each receives an equal portion (time slice) of the CPU. This will sometimes affect the performance of CPU. This paper considers the problem of fixed time slice. We propose a modified version of the Round Robin Algorithm that is based upon the variation in time slice to improve the performance of the system. We finally compare our new algorithm with the existing one in terms of waiting time.

Keywords:- CPU Scheduling Algorithm, Round Robin (RR), Time Slice, CPU utilization, waiting time, turnaround time, response time.

I. INTRODUCTION

Now days, the task of modern operating system is more complex than ever before. They have evolved from a single user to a multi tasking environment. When a computer is multi-tasked, it frequently has multiple processes competing for the CPU at the same time. When more than one process is in ready state and there is a single CPU available, the operating system must decide which process to run first. The part of the operating system that makes the choice is called the short term scheduler or CPU scheduler. The functioning of the CPU scheduler is to take some process from the ready queue and start its execution. Now the process that is given to the CPU totally depends upon the scheduling algorithm [1][2][5]. There are many scheduling algorithms available in the literature viz. FIFO, SJF, SRTF, Priority, Round Robin Scheduling etc. Our work is on Round Robin Scheduling algorithm.

To measure the efficiency of a scheduling algorithm there are some metrics such as Throughput (No. of processes completed per unit time), Turnaround time (The interval from the time of submission of a process to the time of completion), waiting time (The time spend by the process in the ready queue), Response time (The time interval between the submission of job until the first response is produced)[2].

In this paper authors propose a modified Round Robin Scheduling Algorithm for the uniprocessor systems using variable time slice that will effectively reduce the average response time, average waiting time and consequently average turnaround time. Proposed algorithm also reduces the number of context switches, which again improves the performance of the system.

In section 1 we deal with the basic introduction about the round robin algorithm. In section 2, authors present the detailed view of the proposed algorithm. In section 3 we compare the original round robin algorithm and proposed round robin algorithm for uniprocessor systems using variable time in terms of waiting time, turnaround time and the total number of context switches. Section 4 presents the improvements that can be obtained after implementing the modified Round Robin Scheduling algorithm. Lastly we include the future scope of our work.

II. ROUND ROBIN SCHEDULING ALGORITHM

The round-robin (RR) scheduling algorithm is designed especially for time-sharing systems. It is also the simplest and widely used scheduling algorithm. In Round Robin Algorithm CPU scheduler goes around the ready queue allocating the CPU to each process (on the head of ready queue) for a time interval of one time slice. If the process does not complete its execution within the allocated time slice, the process is put at the tail of the ready queue [1][5] and process switch occurs where state of the running process is stored onto the stack and the state of the next process is loaded. Sometimes the remaining burst time is more than time slice by a fraction, even then process is preempted and context switch occurs. This results into more waiting time for that process and more overheads due to unnecessary context switch. In this paper our main concern is to reduce the waiting time consequently the turn around time by taking the variable time slice for the processes based on the average burst time and the number of processes.

III. PROPOSED ALGORITHM

In the proposed algorithm, we attempt to effectively reduce the average waiting time and average response time for the processes by putting some variation in the time slice rather assigning the fixed time slice. According to our algorithm the processes with short CPU burst time will be dispatched first. Firstly sort the ready queue in the increasing order of burst time and then calculate the varying time slice. The variable time slice depends on the average of the burst time of the processes available in the ready queue. CPU will be dispatched to the processes in the sorted ready queue until the completion of one cycle. The time slice will be recalculated for each succeeding cycles by taking the average of the burst time of processes in the ready queue. This sequence of steps would be repeated until the ready queue is empty.
The complete algorithm is described in this section.

INPUT: N processes and the total burst time of all the process in the ready queue.
OUTPUT: Performance improvement with the reduction in waiting time and consequently turnaround time.

Algorithm
1. Sort all the processes in the ready queue on the basis of increasing burst time.
2. Calculate the average of the burst time of all the processes present in the ready queue.
3. Calculate the time slice
TS=Average burst time/N
4. Dispatch the CPU to the processes for the calculated time slice, only for one CPU cycle.
5. Repeat Steps 2 and 3 until ready queue becomes empty.
6. Calculate the Waiting time of each process.

IV. PERFORMANCE EVALUATION

In this section authors evaluated their work by considering the example. We presented the example with the original Round Robin Scheduling Algorithm and with the modified Round Robin Scheduling Algorithm using variable time slice. Finally we presented the comparison of both the algorithm in terms of waiting time.

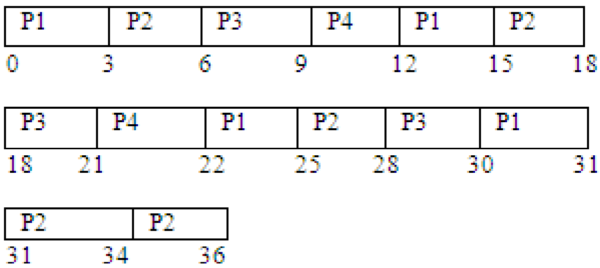
Example: Assume four processes (P1, P2, P3 and P4) arrived at time unit 0.
Terminology used:
TS: Time Slice
N: No. of processes in the ready queue
W: Total waiting time
The burst time of the process is given in Table 1.

TABLE I

PROCESS	BURST TIME
P1	10
P2	14
P3	8
P4	4

A. Using the original Round Robin Algorithm

Let the Time Slice TS = 3ms
Gantt chart for Original Round Robin Scheduling Algorithm



Waiting Time for each process
P1 = 21
P2 = 22
P3 = 22
P4 = 18
W = 83

Average Waiting Time= W/N = 83/4= 20.75
Turnaround time for each process
P1 = 21+31=52
P2 = 22+14=36
P3 = 22+8=30
P4 = 18+4=22
Average Turnaround time=35
Total Number of context switch: 11

B. Using the Modified Round Robin Algorithm with Variable Time Slice

Sorted ready queue based on the modified RR algorithm is as shown in Table 2.

TABLE II. SORTED READY QUEUE

PROCESS	**BURST TIME**
P4	4
P3	8
P1	10
P2	14

Calculate the time slice that is equivalent to the average of burst time of all the processes and dispatch the CPU to the processes in the ready queue for one time slice and repeat this step until the ready queue is empty**.** Here we present the example in different passes.

Pass 1

N = 4.
Now, the value of time slice is calculated as
Time Slice= (4+8+10+14)/4 = 36/4 = 9
Now, Gantt chart for Pass1 is:
Now, the value of TS is calculated as
Time Slice = 36/4 = 9

Gantt chart for Pass1 is:

P4	P3	P1	P2

0 4 12 21 30

Pass 2

P4 with burst time 4 time units and P3 with burst time 8 time units are completely executed. The current status of ready queue after pass1 is as mentioned in Table 3.

TABLEIII. READY QUEUE AFTER PASS 1

PROCESS	**BURST TIME**
P1	1
P2	5

N= 2
Time Slice= 6/2 = 3
Gantt chart for Pass 2 is:

P1	P2

30 31 34

Pass 3

After pass 2 Process P1 has completed its execution. The updated ready queue is as shown in Table 4.

TABLE IV. READY QUEUE AFTER PASS 2

PROCESS	**BURST TIME**
P2	2

N = 1
Time Slice = 2/1 = 2

Gantt chart

P2

34 36

The Final Gantt chart

P4	P3	P1	P2	P1	P2	P2

0 4 12 21 30 31 34 36

Waiting time for the each process

P1 = 21
P2 = 22
P3 = 4
P4 = 0
W = 47
Average Waiting Time= W/N =47/4 = 11.75

Through the above mentioned example, we can evaluate that there is an effective reduction in the average waiting time with using the variable time slice. This difference (20.75-11.75) = 9time units. It shows that the waiting time using proposed approach is reduced to approximately half of the waiting time using original Round Robin Scheduling algorithm.

Reduction in waiting time consequently reduces the turnaround time of the processes.

Turnaround time for each process
P1 = 21+31=52
P2 = 22+14=36
P3 = 4+8=12
P4 = 0+4=4
Average turnaround time=26
Total Number of context switch=6
We collect the improvements found after the comparison of two algorithms here as follows:

1. Reduction in average waiting time(approximately reduced to half)
2. Reduction in turnaround time
3. Reduction in number of context switches
4. Increased Throughput

We have shown the comparison of two algorithms in terms of waiting time with the help of the graph in Fig. 1. There is decreased waiting time when we are using the proposed algorithm for scheduling of processes among CPU.

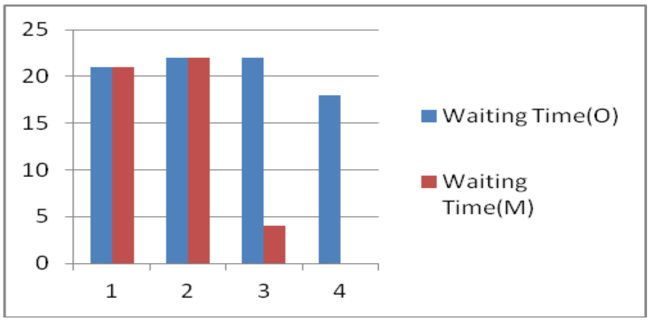


Figure 1. Comparison of two algorithms

V. CONCLUSION

Authors have proposed the concept of variable time slice using the average burst time of all the processes in the ready queue and the total number of processes in ready queue. We also proposed a modified Round Robin Scheduling algorithm that uses the generated variable time slice. The proposed algorithm improves the performance of the original round robin scheduling algorithm by reducing the average waiting time and turnaround time. This algorithm also improves the total number of context switch. We can further improve the processor efficiency by introducing the fuzzy time quantum for the same algorithm. These concepts can be further modified and applied to multiprocessor systems.

REFERENCES

[1] Silbersschatz,A.,Peterson,J.L.,and Galvin,P.B. “,Operating System Concepts”, Addison Wesley,7th Edition, 2006

[2] Andrew S. Tanenbaum, and Albert S. Woodfhull, “Operating Systems Design and Implementaion”, Second Edition,2005

[3] L Kleinrock “Analysis of a time-shared processor Naval Research Logistics Quarterly” Vol. 11, No 1, pp. 59-73, March 1964

[4] Aburas,A.A.;Miho,V., “Fuzzy Logic based algorithm for Uniprocessor Scheduling”, Computer and Communication Engineering, 2008. ICCCE 2008., pp. 499-504, May 2008.

[5] Supriya Raheja, Reens Dhadich, Smita Rajpal, “An Optimum Time Quantum Using Linguistic Synthesis for Round Robin Scheduling Algorithm”, International Journal on Soft Computing, Vol. 3, No. 1, pp. 57-66, Feb 2012.

[6] H.S. Behera, Jajnaseni Panda, “A New Proposed Two Processor Based CPU Scheduling Algorithm with Varying Time Quantum for Real Time Systems”, Journal of Global Research in Computer Science, pp. 81-87, Vol 2, Issue 4,2011.