International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.4, No.4, August 2014

AN IMPROVED ROUND ROBIN CPU SCHEDULING ALGORITHM WITH VARYING TIME QUANTUM

Manish Kumar Mishra[1] and Dr. Faizur Rashid[2]

[1]Department of Information Technology, Haramaya University, Dire Dawa, Ethiopia
[2]Department of Computer Science, Haramaya University, Dire Dawa, Ethiopia

ABSTRACT

Process management is one of the important tasks performed by the 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. In time shared systems, Round Robin CPU scheduling is the preferred choice. In Round Robin CPU scheduling, performance of the system depends on the choice of the optimal time quantum. This paper presents an improved Round Robin CPU scheduling algorithm coined enhancing CPU performance using the features of Shortest Job First and Round Robin scheduling with varying time quantum. The proposed algorithm is experimentally proven better than conventional RR. The simulation results show that the waiting time and turnaround time have been reduced in the proposed algorithm compared to traditional RR.

KEYWORDS

CPU Scheduling, Round Robin Scheduling, Improved Round Robin Scheduling, Burst Time, Waiting Time, Turnaround Time

1. INTRODUCTION

Improper use of CPU can reduce the efficiency of the system in multiprogramming computing systems. In multiprogramming systems, multiple processes are being kept in memory for maximum utilization of CPU [1]. CPU utilization can be maximized by switching CPU among waiting processes in the memory and running some process all the time [2]. The main aim of the CPU scheduling algorithms is to minimize waiting time, turnaround time, response time and context switching and maximizing CPU utilization. This study focuses on improving the effectiveness of Round Robin CPU scheduling algorithm.

1.1. Performance parameters

The processes that have been submitted to the system and waiting for the processor time are put in a queue called ready queue. The CPU should be busy as much as possible to use it effectively. Whenever CPU becomes idle, a waiting process from the ready queue is selected and CPU allocated to that process [2]. The time for which a process uses the CPU is known as burst time. Arrival Time is the time at which a process joins the ready queue. The total time taken by a process from the time of submission to the time of completion of the process is the turnaround time. The total time spent by the process waiting in the ready queue is termed as waiting time of the process. The CPU scheduling algorithms focus on reducing the waiting time, turnaround time and context switches by scheduling the processes from the ready in an effective manner.

1.2. CPU Scheduling Algorithms

CPU scheduling algorithms are used to allocate the CPU to the processes waiting in the ready queue. First-Come-First-Served (FCFS) Round Robin (RR), Shortest Job First (SJF) and, Priority Scheduling are some popular CPU scheduling algorithms. In FCFS CPU scheduling algorithm, the process that arrives first in the ready queue is served first. The average waiting time in this scheduling is quite long [2]. In SJF CPU scheduling algorithm, the process with shortest CPU burst time executes first from the ready queue. In SJF average waiting time decreases. CPU is allocated to the processes based on their priority in Priority scheduling algorithm. The process with highest priority gets executed first and then the second highest and so on. Each process from the ready queue is given a fixed time quantum in RR CPU scheduling algorithm. In this paper we have proposed a new algorithm that uses the features of SJF and RR with varying time quantum to reduce the waiting time and turnaround time.

2. RELATED WORK

In the past years, several CPU scheduling algorithms have been introduced for improving the system performance. A fixed time quantum is allocated to the process waiting in the ready queue, only in first cycle and then SJF is used to select next process in An Improved Round Robin Scheduling Algorithm for CPU Scheduling [1]. Time quantum is continuously adjusted according to the burst time of the processes in Self-Adjustment Time Quantum in Round Robin Algorithm [3]. The job mix order for the algorithm in [3] is used in Dynamic Quantum with Readjusted Round Robin Scheduling Algorithm [4]. Robust quantum time value has been proposed in [5] after arranging the process in the ascending order and taking the average of minimum and maximum burst time of the processes in the ready queue. A new weighting technique is introduced for CPU Schedulers in Burst Round Robin (BRR) [6]. In this approach, the processes having smaller CPU time are given more time, so that these processes can be cleared from the ready queue in a short time span. Debashree Nayak et. al. [7] did the similar work as [3] [4]. The optimal time quantum is assigned to each process after every cycle of execution. Optimal time quantum is the average of highest CPU burst time and the median. In [8] a new CPU scheduling algorithm is presented. In this approach, the running processes are scheduled based on three parameters of CPU. These parameters are burst time, I/O service time, and priority of processes. A new fare-share scheduling with weighted time slice [9] assigns a weight to each process and the process having the least burst time is assigned the largest weight. The time quantum is calculated dynamically, using weighted time slice method and then the processes are executed. Algorithm in [10] calculates the original time slice suited to the burst time of each processes and then dynamic ITS (Intelligent Time Slice) is found out in conjunction with the SRTN algorithm [2]. The scheduling algorithm proposed in [11] uses two processors, one processor is dedicated to execute CPU-intensive processes only and the other processor is dedicated to executed I/O-intensive process. This gives better result in a two processor environment than [4]. A New Round Robin based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average [12] introduce a new concept of calculating the average burst time of the processes from the ready queue after every cycle and allocated as dynamic time quantum. The processes are scheduled for execution by giving importance to both the user priority and shortest burst time priority in Fair Priority Round Robin with Dynamic Time Quantum [13]. In this approach, the individual time quantum for each process is decided by a factor based on both user priority and the burst time priority. The CPU is allocated to the first process from the ready queue for a time interval of up to one time quantum in An Improved Round Robin (IRR) CPU Scheduling

Algorithm [14]. After completion of process's time quantum, the remaining CPU burst time of the currently running process is compared with the time quantum. If the burst time of the currently running process is less than one time quantum, the CPU is again allocated to the same process so that it can finish execution and removed from the queue. It reduces the waiting time of the process in the ready queue and hence improve the performance. An Additional Improvement in Round Robin (AAIRR) CPU Scheduling Algorithm [15] proposed an improvement in the conventional RR and IRR [14]. This approach is similar to the IRR but it picks the process from the ready queue whose remaining burst time is shortest. The algorithm proposed in [16] is again an improvement of IRR [14] which uses two queues, ARRIVE queue and REQUEST queue. It is giving significance performance improvement compared to IRR.

3. IRRVQ CPU Scheduling Algorithm

The improved Round Robin CPU scheduling algorithm with varying time quantum (IRRVQ) combines the features of SJF and RR scheduling algorithms with varying time quantum. Initially the processes in the ready queue are arranged in the ascending order of their remaining burst time. CPU is allocated to the processes using RR scheduling with time quantum value equal to the burst time of first process in the ready queue. After each cycle processes in the ready queue are arranged in the ascending order of their remaining burst time and CPU is allocated to the processes using RR scheduling with time quantum value equal to the burst time of first process in the ready queue.

Following is the proposed IRRVQ CPU scheduling algorithm

1. Make a ready queue RQUEUE of the Processes submitted for execution.
2. DO steps 3 to 9 WHILE queue RQUEUE becomes empty.
3. Arrange the processes in the ready queue REQUEST in the ascending order of their remaining burst time.
4. Set the time quantum value equal to the burst time of first process in the ready queue RQUEUE.
5. Pick the first process from the ready queue RQUEUE and allocate CPU to this process for a time interval of up to 1 time quantum.
6. Remove the currently running process from the ready queue RQUEUE, since it has finished execution and the remaining burst time is zero.
7. REPEAT steps 8 and 9 UNTIL all processes in the ready queue gets the CPU time interval up to 1 time quantum.
8. Pick the next process from the ready queue RQUEUE, and allocate CPU for a time interval of up to 1 time quantum.
9. IF the currently running process has finished execution and the remaining CPU burst time of the currently running process is zero, remove it from the ready queue ELSE remove the currently running process from the ready queue RQUEUE and put it at the tail of the ready queue.

3.1. Illustration

A ready queue with four processes P1, P2, P3 and P4 has been considered for illustration purpose. The processes are arriving at time 0 with burst time 12, 8, 21 and 15 respectively. The processes P1, P2, P3 and P4 are arranged in the ascending order of their burst time in the ready queue which gives the sequence P2, P1, P4 and P3. The time quantum value is set equal to the burst time of first process in the ready queue i.e. 8. CPU is allocated to the processes P2, P1, P4 and P3 from the ready queue for a time quantum of 8 milliseconds (ms). After first cycle, the remaining burst time for P2, P1, P4 and P3 are 0, 4, 7 and 13 respectively. The process P2 has finished execution, so it is removed from the ready queue. The processes P1, P3 and P4 are arranged in the ascending order of their remaining burst time in the ready queue which gives the sequence P1, P4 and P3. The time quantum value is set equal to the burst time of first process in the ready queue i.e. 4. CPU is allocated to the processes P1, P4 and P3 from the ready queue for a time quantum of 4 ms. After second cycle, the remaining burst time for P1, P4 and P3 are 0, 3 and 9 respectively. The process P1 has finished execution, so it is removed from the ready queue. The processes P3 and P4 are arranged in the ascending order of their remaining burst time in the ready queue which gives the sequence P4 and P3. The time quantum value is set equal to the burst time of first process in the ready queue i.e. 3. CPU is allocated to the processes P4 and P3 from the ready queue for a time quantum of 3 ms. After third cycle, the remaining burst time for P4 and P3 are 0 and 6 respectively. The process P4 has finished execution, so it is removed from the ready queue. Now only process P3 is in the ready queue, so CPU is allocated to P3 for a time quantum of 6 ms which is the burst time of first process in the ready queue. The waiting time is 24 ms for P1, 0 ms for P2, 35 ms for P3 and 32 ms for P4. The average waiting time is 22.75 ms. With the same set of process with same arrival and CPU burst times, the average waiting time is 25.75 ms for time quantum 8, 27.75 ms for time quantum 4, 29.25 ms for time quantum 3 and 28.75 ms for time quantum 6 in RR. The average turnaround time is 41.75 in IRRVQ while average turnaround time is 39.75 for time quantum 8, 41.75 for time quantum 4, 43.25 for time quantum 3 and 42.75 for time quantum 6 in RR.

4. EXPERIMENTAL ANALYSIS

4.1. Assumptions

For performance evaluation, it has been assumed that all the processes are having equal priority in a single processor environment. The number of processes and their burst time are known before submitting the processes for the execution. The context switching overhead incurred in switching from one process to another has been considered zero. The overhead of arranging the ready queue processes in ascending order has also been considered zero. All processes are CPU bound. No processes are I/O bound. The time quantum is taken in milliseconds (ms).

4.2. Experiments Performed

Two different cases have been taken for performance evaluation of our proposed IRRVQ algorithm. In the case 1, CPU burst time is in random orders and processes arrival time is assumed zero. In the case 2, CPU burst time is in random orders and processes arrival time is assumed non zero. The CPU burst time in ascending or descending orders have not been considered since it gives the same result as the CPU burst time in random orders.

4.2.1. CASE 1 – Zero Arrival Time

In this case, CPU burst time is in random orders and processes arrival time is assumed zero. A ready queue with five processes P1, P2, P3, P4, and P5 has been considered as shown in table 1.

Table 1. Processes with their arrival and burst time (Case 1).

Process	Arrival Time	Burst Time
P1	0	15
P2	0	32
P3	0	10
P4	0	26
P5	0	20

The comparison result of RR and proposed IRRVQ is shown in Table 2. Figure 1, Figure 2 and Figure 3 show the Gantt chart representation of RR with time quantum 10, 5 and 6 respectively. Figure 4 shows the Gantt chart representation of IRRVQ.

Table 2. Comparison of RR and IRRVQ.

Algorithm	Time Quantum (TQ)	Average Waiting Time (ms)	Average Turnaround Time (ms)
RR	10, 5, 6	54.2, 56.2, 60.2	74.8, 76.8, 80.8
IRRVQ	10, 5, 5, 6, 6	46.2	66.8

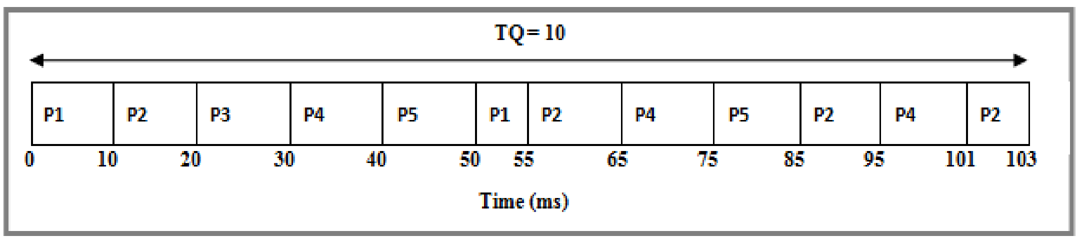

Figure 1. Gantt chart representation of RR with TQ = 10

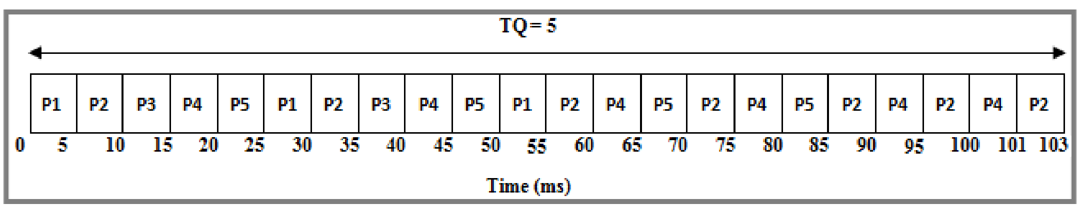

Figure 2. Gantt chart representation of RR with TQ = 5

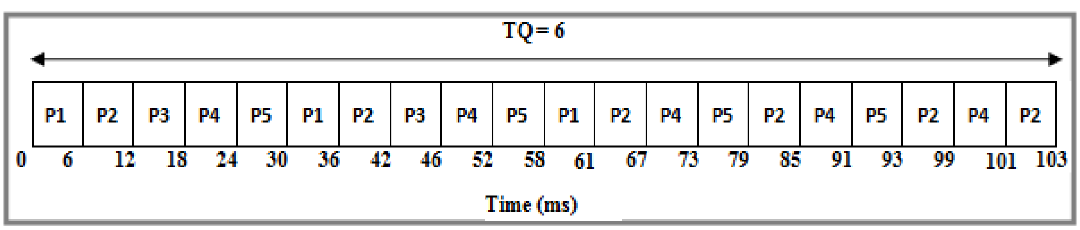

Figure 3. Gantt chart representation of RR with TQ = 6

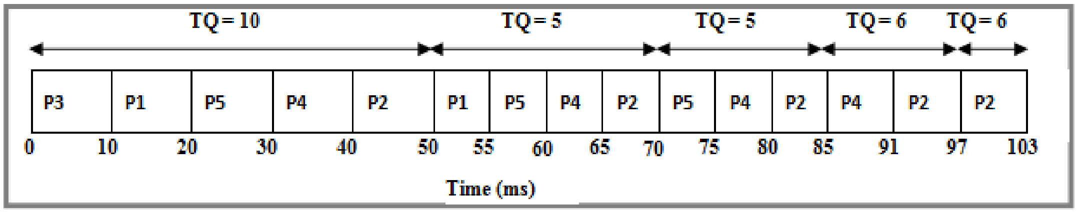

Figure 4. Gantt chart representation of IRRVQ

4.2.2. CASE 2 – Non - Zero Arrival Time

In this case, CPU burst time is in random orders and processes arrival time is assumed non zero. A ready queue with five processes P1, P2, P3, P4, and P5 has been considered as shown in table 3.

Table 3. Processes with their arrival and burst time (Case 2).

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

The comparison result of RR and proposed IRRVQ is shown in Table 4. Figure 5 and Figure 6 show the Gantt chart representation of RR with time quantum 7 and 11 respectively. Figure 7 shows the Gantt chart representation of IRRVQ.

Table 4. Comparison of RR and IRRVQ

Algorithm	Time Quantum (TQ)	Average Waiting Time (ms)	Average Turnaround Time (ms)
RR	7, 11	25.6, 27.0	43.8, 54.4
IRRVQ	7, 11, 7, 11	19.4	37.6

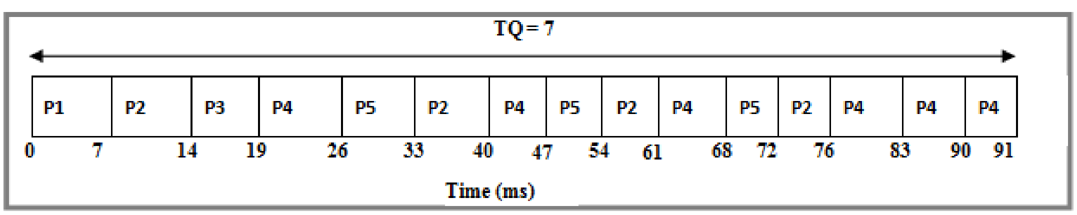

Figure 5. Gantt chart representation of RR with TQ = 7

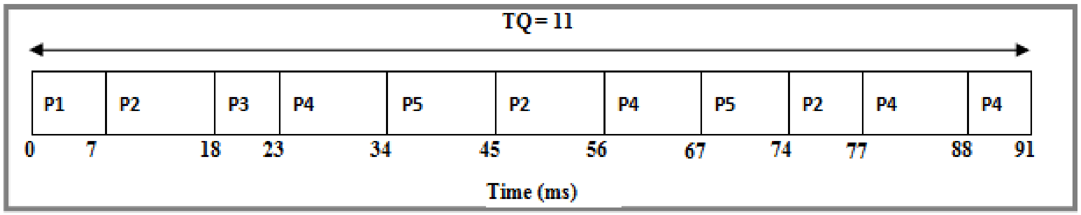

Figure 6. Gantt chart representation of RR with TQ = 11

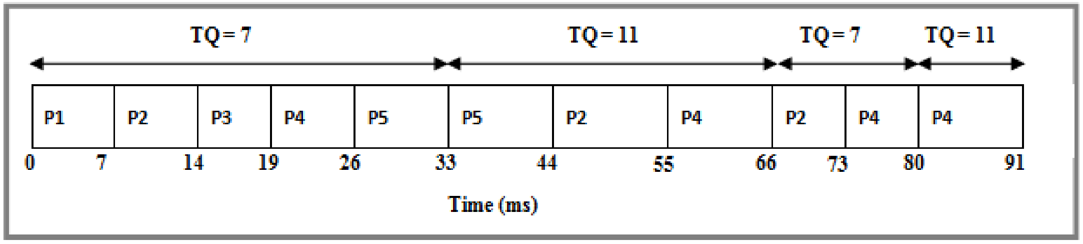

Figure 7. Gantt chart representation of IRRVQ

5. CONCLUSIONS

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 improved round robin CPU scheduling algorithm with varying time quantum proposed in this paper giving better performance than conventional RR algorithm. The waiting time and turnaround time have been reduced in the proposed IRRVQ scheduling algorithm and hence the system performance has been improved. Simulation results also prove the correctness of the theoretical results. The proposed algorithm can be integrated to improve the performance of the systems.

REFERENCES

[1] Rakesh Kumar Yadav, Abhishek K Mishra, Navin Prakash & Himanshu Sharma, (2010) “An Improved Round Robin Schedduling Algorithm for CPU Scheduling”, International Journal on Computer Science and Engineering, Vol. 2, No. 4, pp 1064-1066.

[2] A. Silberschatz, P. B. Galvin & G. Gagne, (2005) Operating System Concepts, John Wiley and Sons Inc, pp 157-167.

[3] Rami J. Matarneh, (2009) “Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of Now Running Processes”, American J. of Applied Sciences, Vol. 6, No. 10, pp1831-1837.

[4] H.S. Behera, R. Mohanty & Debashree Nayak, (2010) “A New Proposed Dynamic Quantum with Re-Adjusted Round Robin Scheduling Algorithm and Its Performance Analysis”, International Journal of Computer Applications, Vol. 5, No. 5, pp 10-15.

[5] M Lavanya & S. Saravanan, (2013) “Robust Quantum Based Low-power Switching Technique to improve System Performance”, International Journal of Engineering and Technology, Vol. 5, No. 4, pp 3634-3638.

[6] Helmy, T. & A. Dekdouk, (2007) “Burst Round Robin as a Proportional-share Scheduling Algorithm”, IEEEGCC, http://eprints.kfupm.edu.sa/1462/.

[7] Debashree Nayak, Sanjeev Kumar Malla & Debashree Debadarshini, (2012) “Improved Round Robin Scheduling using Dynamic Time Quantum”, International Journal of Computer Applications, Vol. 38, No. 5, pp 34-38.

[8] Mehdi Neshat, Mehdi Sargolzaei, Adel Najaran & Ali Adeli, (2012) “The New method of Adaptive CPU Scheduling using Fonseca and Fleming’s Genetic Algorithm”, Journal of Theoretical and Applied Information Technology, Vol. 37, No. 1, pp 1-16.

[9] H.S. Behera, Rakesh Mohanty, Jajnaseni Panda, Dipanwita Thakur & Subasini Sahoo, (2011) “Experimental Analysis of a New Fare-Share Scheduling Algorithm with Waited Time Slice for Real Time Systems”, Journal of Global Research in Computer Science, Vol. 2, No. 2, pp 54-60.

[10] H.S. Behera, Simpi Patel & Bijaylaxmi Panda, (2011) “A new dynamic Round-robin and SRTN algorithm using variable original time slice and dynamic intelligent time slice for soft real time system”. International Journal of Computer Applications, Vol. 2, No.1, pp 54-60.

[11] H.S. Behera, Jajnaseni Panda, Dipanwita Thakur & Subasini Sahoo, (2011) “A New Proposed Two Processor Based CPU Scheduling Algorithm with Varying Time quantum for Real Time Systems”, Journal of Global Research in Computer Science, Vol. 2, No. 4, pp 81-87.

[12] Abbas Noon, Ali Kalakech, and Seifedine Kadry, (2011) “A New Round Robin based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average”, International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1, pp 224-229.

[13] M.K. Srivastav, Sanjay Pandey, Indresh Gahoi & Neelesh Kumar Namdev, (2012) “Fair Priority Round Robin with Dynamic Time Quantum”, International Journal of Modern Engineering Research, Vol. 2, Issue 3, pp 876-881.

[14] Manish Kumar Mishra & Abdul Kadir Khan, (2012) “An Improved Round Robin CPU Scheduling Algorithm”, Journal of Global Research in Computer Science, Vol. 3, No. 6, pp 64-69.

[15] Abdulrazak Abdulrahim, Salisu Aliyu, Ahmad M Mustapha & Saleh E. Abdullahi, (2014) “An Additional Improvement in Round Robin (AAIRR) CPU Scheduling Algorithm”, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4, Issue 2, pp 601-610.

[16] Abdulrazak Abdulrahim, Saleh E. Abdullahi & Junaidu B. Sahalu, (2014) “A New Improved Round Robin (NIRR) CPU Scheduling Algorithm”, International Journal of Computer Applications, Vol. 90, No. 4, pp 27-33.