International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 7 (2014), pp. 765-772

http://www. irphouse.com

# A Comparative Performance Analysis of Various Variants of Round Robin Scheduling Algorithm

**Sandeep Negi[1] and Priyanka Kalra[2]**

*[1]Delhi Institute of Technology & Management,*
*[2]Centre for Development of Advanced Computing*
*[1] Sonepat (Haryana), INDIA, [2]Noida (U.P.), INDIA*
*[1]Department of Computer Science, [2]School of IT*
*[1]s.negi1986@gmail.com, [2]kalra.priyankapk@gmail.com*

## ABSTRACT

As processor is one of the most important computer resource, it becomes obvious to use this resource in a most efficient way such that its utilization is maximum and hence waiting time, turnaround time, response time and context switch of processes submitted to it is reduced. Various scheduling algorithm are described in literature such as First Come First Serve (FCFS), Shortest Job First (SJF), Round Robin (RR) etc. Among these RR is the most popular and widely used scheduling policy in timesharing system. In RR all process in ready queue, seize the processor for a short period of time quantum circularly. This time quantum plays a very important role in scheduling processes in an efficient way. If this time quantum is large, response time of processes would be high. On the other hand if it is less, context switching between processes will increase and hence overhead would increase. In this paper we discuss various variants of RR scheduling policy and compare them experimentally in terms of waiting time, turnaround time and number of context switches.

**Keywords -** CPU Scheduling, Turnaround time, Waiting Time, Response Time, Context Switching, RR Scheduling.

## 1. INTRODUCTION

Operating system (OS) may be viewed as an organized collection of software extension of hardware, consisting of control routines for operating a computer and providing an environment for execution of programs. Among such task, scheduling is central to OS design. To increase CPU efficiency, the OS must be multiprogrammed and here OS plays an important role to manage all those programs. In multiprogramming, OS keeps track of various active processes and allocate system resources to them. This is called CPU scheduling. Many scheduling algorithms [10, 11] have been used for CPU scheduling like A) First Come First Serve in which processes present in ready queue are scheduled in the same order in which they come. B) Shortest Job First (SJF) in which a process with the shortest burst time is given the processor first to execute and then the next shortest and so on until the ready queue become empty. This algorithm requires the pre knowledge of the next process burst time. In preemptive SJF the process is preempted if any process with a shorter burst time arrives. C) Fixed Priority Preemptive Scheduling in which the process which has the highest priority is scheduled first, then the process with second largest priority and so on. D) Round Robin Scheduling which assigns a fixed time quantum to each process in equal portions in cyclic order. The processes can execute in their time quantum only and when their time quantum expires the process is preempted from the CPU and placed at the end of ready queue.

Main aim of a scheduling algorithm is to maximize throughput and processor utilization. Several performance parameters [10, 12] have been described in literature such as A) Context Switch: A context switch is a process of storing and restoring context (state) of a preempted process, so that execution can be resumed from same point at a later time. Context switching leads to wastage of time and memory, which in turn increases the overhead of scheduler. B) Processor Utilization: This is a measure of how much busy the processor is. Usually, the goal is to maximize processor utilization. C) Throughput: It is defines as the number of processes completed per unit of time. Throughput is low in round robin scheduling implementation. D) Waiting Time: It is the total time a process has been waiting in ready queue. E) Turnaround Time: The time interval from the time of submission of a process to the time of its completion is called turnaround time. F) Response Time: It is the time from the submission of a request until the first response is produced, that means time when the task is submitted until the first response is received. So the response time should be low for best scheduling.

## 2. RELATED WORK

In recent years many CPU scheduling algorithms have been developed. Rami J. Matarneh [1] proposed a method in which the time quantum is chosen based on the median of all the processes present in the ready queue. Time quantum will be equal to median if median is greater than 25 otherwise it will be 25. By using this approach 50% of the remaining processes will be finished in each round. Negi [2] proposed an approach that extends the time quantum of processes which require only a fractional more amount of time to complete than the allocated fixed time quantum. Hiranwal et al. [3] proposed a method in which the processes are arranged in non-decreasing order of their burst time. Now a smart time quantum is used to service the processes. If the processes are odd in number, the time quantum will be equal to the burst time of mid process otherwise average of the burst time is taken. Dawood [4] proposed a method in which time quantum is chosen by summing the max and min burst and then multiplying it by (80) percentage because 80% process of CPU burst should be smaller than time quantum. Noon et al. [5] proposed to take the time quantum equal to burst time of the process if it was the only process in ready queue otherwise the average of the burst time of all the processes in ready queue. Banerjee et al. [6] algorithm adjusts the time quantum dynamically so as to yield better performance. In this algorithm all the processes in ready queue are first sorted according to the burst time and then mid process is found out. The time quantum is now found out to be average of all the processes burst from mid to last. This procedure is then recursively applied by deleting all the process whose remaining burst is 0. Nayak et al. [7] calculated the optimal time quantum by taking the average of highest burst and lowest burst in ready queue. Yaashuwanth et al [8] introduced a term intelligent time slice which was calculated using the formula (range of burst * total number of processes)/ (priority range * Total number of priority). A number of other new variants of improved RR have been developed some of which also consider priority as a parameter [9].

## 3. EXPERIMENTAL ANALYSIS

For evaluating the performance we assume that the environment where all the experiments are performed is a single processor and the burst time for all processes is known prior to submitting of process to the scheduler. Moreover all the processes have equal priority.

### a. Case Study I

To evaluate the performance of different algorithms, we assume that there are 5 processes in ready queue with arrival and burst time as shown in table 1.

**Table 1:** Processes specification for case study I

| Process | Arrival Time | Burst Time |
|---|---|---|
| P1 | 0 | 5 |
| P2 | 0 | 12 |
| P3 | 0 | 30 |
| P4 | 0 | 48 |
| P5 | 0 | 75 |

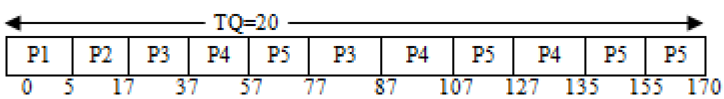



**Fig1:** Gantt chart for RR in Table1

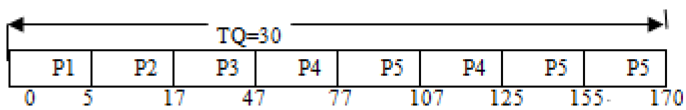



**Fig2:** Gantt chart for Adaptive RR in Table1

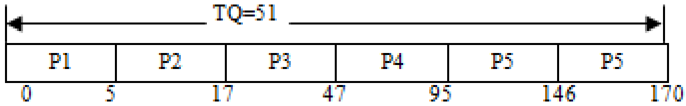



**Fig3:** Gantt chart for MARR in Table1

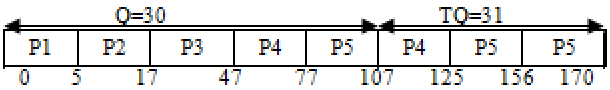



**Fig 4:** Gantt chart for SARR in Table1

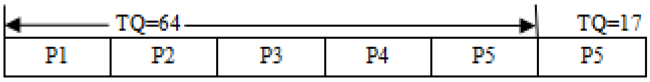



Fig 5: Gantt chart for AQMMRR in Table1

**b. Case Study II**

We assume that there are 5 processes in arrival and ready queue with burst time as shown in table 2.

**Table2:** Processes specification for case study II

| Process | Arrival Time | Burst Time |
|---|---|---|
| P1 | 0 | 8 |
| P2 | 0 | 19 |
| P3 | 0 | 30 |
| P4 | 0 | 54 |
| P5 | 0 | 83 |

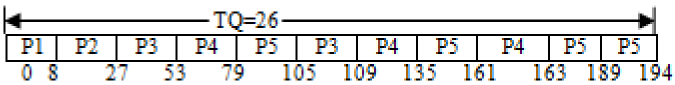



**Fig 6:** Gantt chart for RR in Table 2

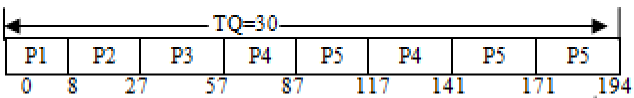



**Fig 7:** Gantt chart for Adaptive RR in Table 2

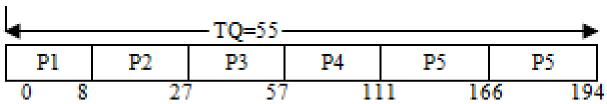
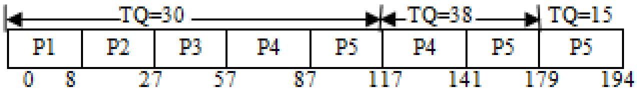



**Fig 8:** Gantt chart for MARR in Table 2




bGantt chart for SARR in Table 2

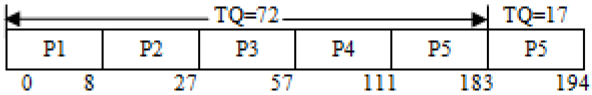



**Fig 10:** Gantt chart for AQMMRR in Table 2

## 4. COMPARISON OF RESULTS

Performance of four selected problems from the literature has been compared by considering average waiting time, average turnaround time, and number of context switches. Table 3 and Table 4 show the result obtained and Figure 11 and Figure 12 show the performance comparison for case study I and case study II respectively.

**Table: 3** Computational Result of case study I

| Algorithm | Time Quantum | Average Waiting Time | Average Turnaround Time | Context Switch |
|---|---|---|---|---|
| RR | 20 | 50.8 | 82.8 | 10 |
| Adaptive RR [3] | 30 | 41.2 | 72.8 | 7 |
| MARR[6] | 51 | 32.8 | 66.8 | 5 |
| SARR[1] | 30, 31 | 38.8 | 72.8 | 7 |
| AQMMRR [4] | 64, 17 | 32.8 | 66.8 | 5 |

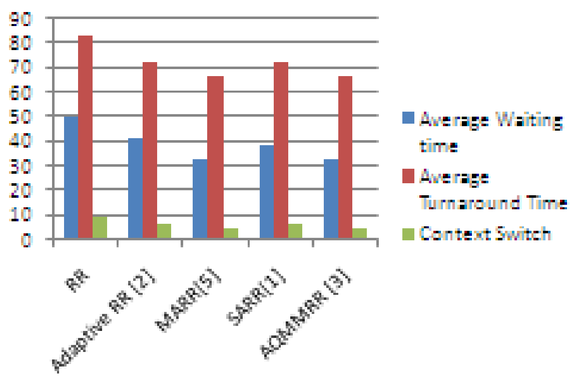



**Fig 11:** Performance Comparison for case study

**Table: 4** Computational Result of case study II

| Algorithm | Time Quantum | Average Waiting Time | Average Turnaround Time | Context Switch |
|---|---|---|---|---|
| RR | 26 | 110.4 | 149.2 | 10 |
| Adaptive RR [3] | 30 | 46.8 | 85.4 | 7 |
| MARR[6] | 55 | 40.6 | 79.6 | 5 |
| SARR[1] | 30, 38, 25 | 46.6 | 85.4 | 7 |
| AQMMRR [4] | 72, 17 | 40.6 | 49.4 | 5 |

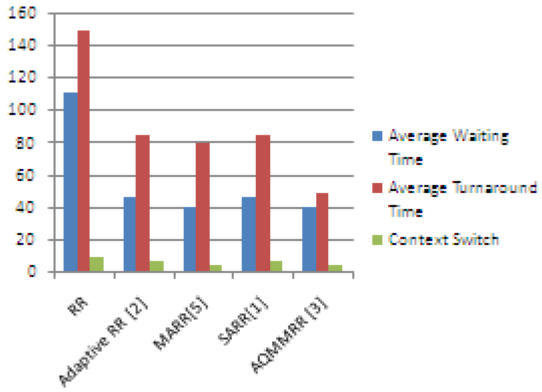



**Fig12:** Performance Comparison for case study II

## 5. CONCLUSION

Time quantum plays a very important role in RR scheduling. Several times the question arise what should be the optimal time quantum of RR scheduling. In this paper a comparative study of RR algorithm with some other proposed algorithm is made. From the experimental results we found that all the proposed algorithms [1, 3, 4, 6] perform better than the simple RR scheduling algorithm in terms of performance metrics such as average waiting time, average turnaround time and number of context switches.

## 6. REFERENCES

[1] Matarneh Rami J., “Self adjustment time quantum in round robin algorithm depending on burst time of the now running process”, Department of Management Information Systems, American Journal of Applied Sciences 6 (10): 1831-1837, 2009, ISSN 1546-9239.

[2] Negi Sandeep, “An Improved Round Robin Approach using dynamic time quantum for improving average waiting time”, International Journal of Computer Applications: Vol. 69 No 14, 2013.

[3] Hiranwal Saroj, Dr. Roy K. C., " Adaptive round robin scheduling using shortest burst approach based on smart time slice", Volume 2, No. 2, July-Dec 2011, pp:319-32.
[4] Dawood Ali Jbaeer, " Improving efficiency of round robin scheduling using ascending quantum and minimum-maximum burst time", Journal of University of ambar for pure science : Vol. 6: No 2, 2012,
[5] Noon Abbas, Kalakech Ali, Kadry Saifedine, " A new round robin based scheduling algorithm for operating systems: dynamic quantum using the mean average" IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1, May 2011.
[6] Banerjee Pallab, Banerjee Prabal Banerjee, Dhal Shweta Sonali, "Comparative performance analysis of mid average round robin scheduling (MARR) using dynamic time quantum with round robin scheduling algorithm having static time quatum", International Journal of Electronics and Computer Science Engineering,
[7] Nayak Debashree, Malla Sanjeev Kumar, Debadarshini Debashree, "Improved round robin scheduling using dynamic time quantum", International Journal of Computer Applications (0975-8887) Volume 38- No 5, January 2012.
[8] Yaashuwanth C. & R. Ramesh, " Intelligent time slice for round robin in real time operating system, IJRRAS 2 (2), February 2010.
[9] Behera H.S, Patel Simpi, Panda Bijayalakshmi, " A new round robin and SRTN algorithm with variable original time slice for soft real time systems" International Journal of Computer Applications volume 16- No.1, February 2011.
[10] Silberschatz, Galvin and Gagne, Operating system concepts, 8th edition, Wiley, 2009
[11] Tenanbaum, A.S., Modern Operating Systems, 3rd edition, PHI.
[12] Harris J Archer" Operating System", Schaum Outline.