

# Round Robin Scheduling Algorithm based on Dynamic time quantum



Rashmi Dhruv

**Abstract:** After studying various CPU scheduling algorithms in Operating System, Round Robin scheduling algorithm is found to be most optimal algorithm in timeshared systems because of the static time quantum that is designated for every process. The efficacy of Round Robin algorithm entirely depends on the static time quantum that is being selected. After studying and analyzing Round Robin algorithm, I have proposed a new modified Round Robin algorithm that is based on shortest remaining burst time which has resulted in dynamic time quantum in place of static time quantum. This improves the performance of existing algorithm by reducing average waiting time and turn-around time and minimizing the number of context switches.

**Keywords :** Waiting time, Round Robin algorithm, Turn-around time, CPU Scheduling, Context Switching.

## I. INTRODUCTION

To communicate with computer hardware, user requires an interface which is the operating system. One of the important role of operating system is to coordinate the use of various resources among various application programs[3]. CPU scheduling is how different processes are assigned to execute on available CPUs in multitasking and multiprocessing environment. The main objective of CPU scheduling is to improve the performance of the system by minimizing turn-around time, response time and waiting time and maximizing CPU utilization. Various scheduling algorithms are available to perform this task. Among them, Round Robin Scheduling algorithm is the one ,which uses ‘Time quantum’ as a deciding factor to decide for how much time a process will execute before getting pre-empted by another process in the queue. Although it is found to be most optimal scheduling algorithm for time shared systems, its disadvantage is, it takes larger time to execute a process and process with variable burst time has to wait for a longer period of time in waiting queue because of the static time quantum that is used. Because of this limitation, I Proposed a modified algorithm which uses dynamic time quantum concept with sorted remaining burst time.

Revised Manuscript Received on October 30, 2019.

\* Correspondence Author

**Rashmi Dhruv\***, Vivekananda School of Information Technology (VSIT), Vivekananda Institute of Professional Studies affiliated from Guru Gobind Singh Indraprastha University, Delhi India. Email: dhruv.rashmi@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

## A. PRELIMINARIES

Various terms that are used in CPU scheduling are:

- **Process:**  
A program that is currently executing in a system is known as a process.
- **Ready queue:**  
Ready queue contains those processes that are ready for execution but the processor is not allocated to them because processor is currently busy in doing some other task.
- **Burst time:**  
Time span, that a process needs control of CPU so that execution can be completed.
- **Arrival time:**  
Instance when a process arrives or enters the ready queue.
- **Turn-around time:**  
Entire time taken by a process to complete its execution, which is inclusive of time consumed in waiting queue as well as time utilized in execution.
- **Waiting time:**  
Time consumed by a process waiting in a queue (ready queue).
- **Response time:**  
It is the time from submission of process till it gets its first response.

Scheduling can be either Preemptive or non-preemptive. Under non-preemptive scheduling, a CPU is consigned to a process till the time of its completion while in Preemptive scheduling, a process that is being executing may be forced to release the CPU by some other processes based on the scheduling criteria. In multiprogramming and time sharing systems, the CPU switches among various processes fastly. The term “context switches”, is used to describe the number for which CPU switches among various processes.

## B. EXISTING SCHEDULING ALGORITHMS

A CPU scheduling algorithm decides the order of execution of processes or order of assignment of processes to the CPU based on various scheduling criteria. The various CPU scheduling algorithms that are used are FCFS(First Come First serve), SJF(Shortest Job First), SRTF(Shortest Remaining Time First), Priority scheduling and RR(Round Robin Scheduling). Among the above mentioned algorithms, FCFS and SJF are non-preemptive algorithms, Priority scheduling can be either non-preemptive or preemptive, and SRTF and Round Robin algorithm are preemptive scheduling algorithms.



**II. PROPOSED ALGORITHM**

In the new modified algorithm, processes are organized in rising order of burst time, & dynamic time\_quantum is used based on shortest remaining burst time. Time quantum plays a significant role in the performance of Round Robin algorithm. If time quantum of algorithm is substantial, RR is similar to FCFS, and if time quantum of given algo is too little it will result in extravagant context switches. The suggested algorithm resolves the problem by using time quantum, which is dynamic in nature and adjusted repeatedly in consonance with burst time of processes which are currently running. In this algorithm, processes are first arranged in ascending order in accordance with their burst time and median of burst time is considered as time\_quantum to get optimal time quantum.

In this algorithm, if the processes are there in ready queue and waiting for the CPU, the arrival time is set to a value zero prior to allocating CPU to them. The Pseudo code for RR algorithm based on shortest remaining burst time is as follows:

Input : Burst time, Number of processes(N)

- Step 1 : Sort all processes, that are present in ready queue in accordance with their burst time(in increasing order).
- Step 2 : While( ReadyQueue != NULL)  
TIME\_QUANTUM = Median(remaining burst time of processes in ready queue)
- Step 3 : Set time quantum(TIME\_QUANTUM) to process;  
P<sub>i</sub> -> TIME\_QUANTUM
- Step 4 : If(i<N) goto previous step ,i.e., Step number 3  
//compare process<sub>i</sub> with total no. of processes
- Step 5 : If new process has entered,  
Modify number of processes, N, and goto Step number 1  
//Check for any new process  
End of While
- Step 6 : Compute mean waiting time, mean turnaround time and context switches
- Step 7 : End

**A. RESULT AND PERFORMANCE EVALUATION**

To demonstrate the proposed algorithm, several experiments are considered and the result is compared with existing RR algorithm. The following example is comparing the result of proposed algorithm and existing RR algorithm. Five process are taken with different burst time. Firstly, average turn around time and average waiting time is calculated using Round robin algorithm, and then using the new proposed algorithm. In the first case the time quantum is considered as static, with value 25,i.e.,TIME\_QUANTUM=25 and then in second case dynamic time quantum is used.

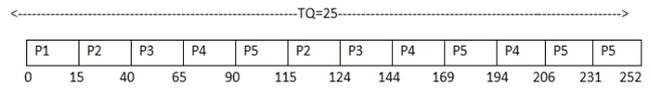
Example using Round Robin Algorithm:

Assuming that there are five processes(P1 to P5), arriving at time 0, sort them in accordance with their burst time(in increasing order). Consider time quantum TIME\_QUANTUM as 25. Assume the following terminologies : Arrival Time=AT, Burst time=BT, Completion Time=CT, Waiting Time=WT, Turn around time=TAT.

Table- I: Turn around & waiting Time for existing Round Robin algorithm

Processes	AT	BT	CT	TAT	WT
P1	0	15	15	15	0
P2	0	34	124	124	90
P3	0	45	144	144	99
P4	0	62	206	206	144
P5	0	96	252	252	156

The Gantt chart for Round Robin algorithm of the above example is as follows :



**Fig. 1 : Gantt chart for Round Robin algorithm in Table I**

The average turn around time for Round Robin algorithm is:  
Average TAT=(15+124+144+206+252)/5 = 148.2

The average Waiting time for the above algorithm is:

Average WT=(0+90+99+144+156)/5 = 97.8

Example using Proposed Algorithm:

Considering same example demonstrated above, sort all processes, that are present in ready queue in accordance with their burst time(in increasing order). (P1 to P5).Then the median of the burst time sequence, i.e., 15, 34, 45, 62 and 96 is calculated which is equal to the time quantum (now, TIME\_QUANTUM=45). This time quantum is now assigned to every process. Then, the next step is to compute left over burst time of every process after assigning the first Time\_Quantum, 'TIME\_QUANTUM'. After execution of first loop, the residual burst time series of the processes will now change to 0, 0, 0, 17, 51 for processes P1 to P5. After process execution, it will automatically be erased from the ready queue. So, process P1, P2 and P3 are deleted from the ready queue. Now again the remaining burst time are sorted in increasing order and median is calculated which is now equal to 34(now, TIME\_QUANTUM=34). After sorting the current remaining burst time of processes that are now available in the ready queue, we will now get the sequence 17, 51. Now taking 34 as time quantum the algorithm is applied again. After second loop of execution, process P4 will be deleted from the ready queue as it has completed its execution, the remaining burst time is calculated which is now equal to 17 for process P5. So 17 is now assigned as the Time\_Quantum, and process P5 now execute and terminate without context switch. All above steps will continue until all processes are terminated and hence removed from the ready queue. In the given example, the algorithm stops here as all the five processes have completed their execution. The table and gantt chart of the proposed algorithm are shown below:

**Table- II: Turn around time and waiting time for Proposed algorithm**

Processes	AT	BT	CT	TAT	WT
P1	0	15	15	15	0
P2	0	34	49	49	15
P3	0	45	94	94	49
P4	0	62	201	201	139
P5	0	96	252	252	156



The Gantt chart for Round Robin algorithm of the above example is as follows:

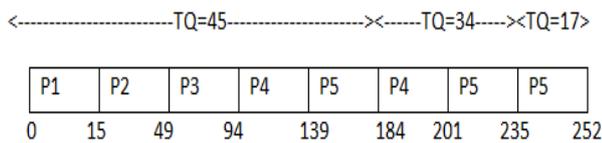


Fig. 2 : Gantt chart for Proposed algorithm in Table II

The average turn around time for proposed algorithm is :  
Average TAT=(15+49+94+201+252)/5 = 122.2

The average Waiting time for the above algorithm is :  
Average WT=(0+15+49+139+156)/5 = 71.8

### III. COMPARISON BETWEEN ROUND ROBIN ALGORITHM AND PROPOSED ALGORITHM

The following chart shows the comparison between Round robin algorithm based on static time quantum and the new proposed algorithm based on dynamic time quantum

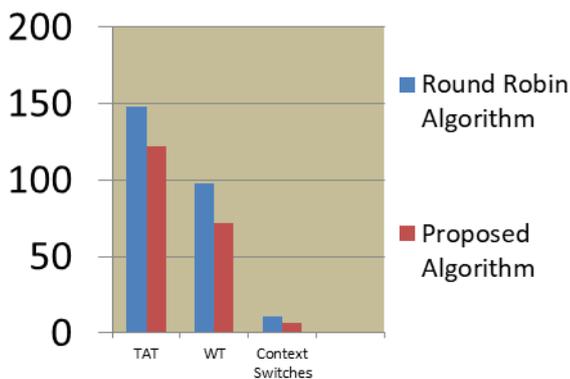


Fig. 3 : Comparison between two algorithm

### IV. COCLUSION

From the above example and other experiments performed on different data sets, it is concluded that by implementing round robin algorithm using dynamic time quantum in place of static time quantum, we will get improved results relating to waiting time, turnaround time and number of context switches.

### REFERENCES

1. "Scheduling Algorithmic Research" (Rami Abielmona), Deptt. of Electrical & Computer Engineering, Ottawa, Carleton Institute, 2000.
2. "Operating Systems Concepts" (Silberschatz, A., P.B. Galvin and G.Gagne), 7<sup>th</sup> Edn., John Wiley and Sons, pp: 944, 2004.
3. "Burst Round Robin: As a Proportional-Share Scheduling Algorithm" (Tarek Helmy, Abdelkader Dekdouk), IEEE Proceedings of the fourth IEEE-GCC Conference on towards Techno-Industrial Innovations, pp. 424-428, 11- 14 November, 2007.
4. "Modern Operating Systems" (A.S. Tanenbaun), 3rd Edn, Prentice Hall, pp: 1104, 2008
5. Improving Average Waiting Time Using Dyanamic Time Quantum" (KN Rout, G.Das,B.M. Sahoo,A.K. Agrawalla), IRAJ,2013.
6. "Time Quantum Based Improved Scheduling Algorithm" (Lalit Kishore, Dinesh Goyal), Volume 3, Issue International Journal of

- Advanced Research in Computer Science and Software Engineering.April, 2013.
7. "Self-Adjustment TQ in RR Algorithm Depending on Burst Time of the Now Running Processes" (Rami J. Matameh,), 2009, American Journal of AS, Vol 6, No. 10.(pp)
8. "Priority Based Dynamic Round Robin with Intelligent Time Slice and Highest Response Ratio Next Algorithm for Soft Real Time System" (Zena Hussain Khalil , Ameer Basim Abdulameer Alaasam), Global Journal of Advanced Engineering Technologies, Volume 2, Issue3, 2013
9. "A Best possible time quantum for Improving Shortest Remaining Burst Round Robin (SRBRR) Algorithm" (P. Surendra Varma), International Journal of Advanced Research in Computer Science and software Engineering, Volume 2, Issue 11, November 2012.

### AUTHORS PROFILE



Ms. Rashmi Dhruv is working as an Assistant Professor with Vivekananda Institute of Professional Studies (VIPS), Pitampura, New Delhi since August 2015. VIPS is affiliated from Guru Gobind Singh Indraprastha University, Delhi India. Before joining VIPS, she worked as Assistant Professor with Guru Premsukh Memorial College of Engineering , Delhi India for more than 4 years. She holds teaching experience of more than 8 years. She holds a degree in B.Tech (Information Technology) from Guru Gobind Singh Indraprastha University and completed her M.Tech in Software Engineering from Maharshi Dayanand University. She is UGC-NET qualified in Computer Science and Applications. Her area of Interest are Operating System, Software Engineering, Software testing and Database Management