Barishal University Journal Part 1, 5(1&2): 257-271 (2018) ISSN 2411-247X

AN IMPROVED ROUND ROBIN SCHEDULING ALGORITHM BASED ON MAXIMUM DIFFERENCE OF TWO ADJACENT PROCESSES

Dipto Biswas, Sagor Saha, Rahat Hossain Faisal and Md. Samsuddoha*

Department of Computer Science and Engineering, University of Barishal, Barishal-8200, Bangladesh.

Abstract

Scheduling algorithm of CPU is fundamentally and inextricably involved with the proficiency of system as well as manipulation of resources. Round Robin is the widely used scheduling algorithm in resource allocation in real time environment. But, its performance degrades with respect to context switching and waiting time. To enhancing the performance, the context switching, waiting time and turnaround time need to be reduced which depends on the choice of an optimal time quantum. This paper presents an optimized round robin algorithm where time quantum is selected based on the maximum difference between two adjacent processes among all processes of the sequence. An experimental evaluation has been conducted and the obtained result of this proposed algorithm has been compared with some existing algorithms to evaluate the performance. The experimental result shows that the performance of the proposed algorithm is better than some other existing algorithms by reducing average waiting time, average turnaround time and number of context switches.

Keywords: Resource Allocation, Scheduling Algorithm, Optimized Algorithm, Time Quantum, Turnaround Time, Waiting Time, Context Switching.

Introduction

Multitasking as well as Multiprocessing is one of the most noticeable phenomena in terms of computer resource allocation. In this regard, a perfect scheduling is required by which the reliability and efficiency as well as allocation of resources will be implemented. The effective scheduling process and the allocation of resources is one of

*Corresponding author's *e-mail: msamsuddoha@bu.ac.bd*

the desired concepts in this modern computing technology in the field of cloud computing and CPU.

Alsheikhy et al., explicitly told that the policy of scheduling is such a system that increases or decreases the performance of a system (Rami, 2009). Round Robin (RR) is one of the most efficient scheduling algorithms for optimizing the performance of processes which one is based on Time Quantum (TQ). To enhance the performance of RR algorithm, the TQ must be optimum (Fataniya and Patel, 2018).

RR process will be excellent during concurrent execution situation if the number of context switching is low and the TQ is high and the entire process time will be optimum (Nayak and Sanjay, 2012). Resource management along with noticeable number of management such as file, input-output, memory, protection, security and networking is highly needy demand from computing system (Ajit and Sahil, 2010). Operating system builds a bridge between the user and expected application in terms of successful installation and deployment. Enormous scheduling algorithms are developed in terms of increasing the efficiency of performance of CPU which is undoubtedly fundamental resource of data processor called computer.

Scheduling refers the sequence of jobs and it creates effect at the proficiency in different activities. By which the number of throughput is increased besides the amount of waiting time is decreased where context switches are available and it ensures that no starvation is present. Process management means managing processes successively among plenty number of processes. Operating System (OS) is nothing but an allocator of resources that means an accumulation of hardware and software in order to develop the characteristics of being robust of a system. Scheduling is the concept that is consecutively utilized in terms of managing the proper distribution of resources (CPU) by the assistant named OS. In terms of developing the performance of a system as well as enhancing the strategy of the allocation of resources scheduling is very user-friendly occurrence in CPU. When a successful sharing is happened among the hardware and software in a body into a computer system, it is known as scheduling. Such scheduling of CPU affects the on the utilization of resources.

The designing of scheduling of CPU or other resources are required and convenient to make the best utilization of resources of a computer system besides file management as well as all management sectors of computing. It does several optimizations such as increment of throughput, make the CPU busy, reducing the amount of waiting time as well as response time.

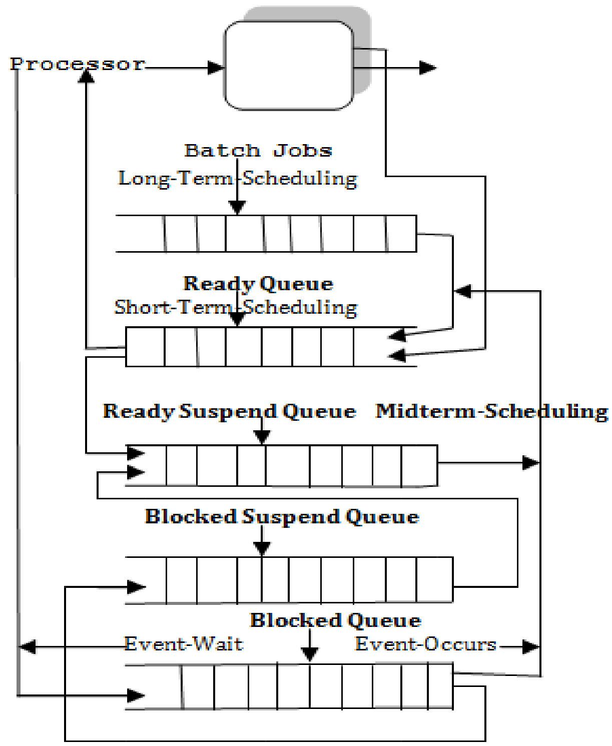

Fig. 1. Scheduling Queues and Schedulers in Operating System.

Fig. 1 demonstrates the perfect queuing model for scheduling. Different number of queues are available in the model named blocked queue, blocked suspend queue, ready suspend queue and ready queue. In general, 3 separated and different types of scheduling are available in an operating system namely short-term-scheduling, long-term-scheduling and medium-term-scheduling.

In this paper, an Optimized Round Robin Algorithm based on Maximum Difference between Two Adjacent Processes has been proposed. Our intention is to build a method that is more efficient than former Round Robin. Our proposal is to do successful resource allocation by sorting all the processes according to Burst Time in increasing order. After that the maximum difference between two adjacent processes had been calculated and measured.

Finally, Time Quantum had been calculated appropriately. Time Quantum is referred as to the summation of maximum difference between two adjacent processes and the first process Burst Time after sorting. The efficiency as well as effectiveness of the allocation and distribution of resources into computing devices will be increased and robust by applying our proposed algorithmic model. Main objectives of this proposed algorithm can be identified as following.

- The number of outcomes will be increased.
- The amount of waiting time will be decreased.
- The amount of response time will be minimized.
- The context switches amount will be reduced.
- The utilization of CPU in resource allocation will be enhanced and enriched.

Our proposed algorithm is sophisticated and highly reliable for making comparatively better allocation of resources by CPU which is a fundamental and primary resource of computing device.

Related Work

Managing the precise allocation of resources is highly required subject in this modern era. It enhances the accessibility of resource whenever one wants or wherever one wants. The Round Robin efficient algorithm for resource distribution is extensively utilized in terms of increasing the proficiency and effectiveness, accessibility, portability of resources (Pradhan, Prafulla and Ray, 2016).

The utilization of a group of hardware and a group of software in a body from wherever one is present through Internet is known as cloud computing to modern technology. CPU is known as a popular and fundamental resource of a computer that has a schedule or routine of processes. Approved Round Robin method based on time quantum as well as burst time named Self Adjustment Round Robin is familiar approach to enrich the efficiency of scheduling of processes of CPU where the time quantum is dynamic (Rami, 2009).

Different number of queues are available in the model named blocked queue, blocked suspend queue, ready suspend queue and ready queue. In general, 3 separated and different types of scheduling are available in an operating system namely short-term-scheduling, long-term-scheduling and medium-term-scheduling. There is a blooming display of event occurrence, waiting as well as time-out (Noon, 2011). The strategy of scheduling is the best subject in terms of creating better effectiveness on the increment of efficiency in the allocation of resources like CPU. Quantum size is required term in Round Robin algorithm that is enormously utilized in scheduling here context switching number, amount of waiting time in average and turn-around time in average are inextricably involved. The involvement of these terms can be the issue of decrement of the performance or the increment of performance (Alsheikhy, 2015). Multitasking as well as multiprogramming are the key concepts in our modern technology. Scheduling algorithms are generally used with a view to managing these tasks. To schedule each process a quantum time related with turn-around-time besides burst-time as well as the quantum time is approximately optimal. Improved Round Robin algorithm is commonly used to do such scheduling proficiently and precisely (Nayak and Malla, 2012).

In total computing system, operating system is an assistant and manager that manages and provides assist to prepare an intimate relationship between computer resources. To increase the performance of the assistant, scheduling is must and such scheduling is possible by Round Robin based on time slice method where the time is dynamic (Chavan and Tikekar, 2013). Panda et al., defines that one of the considerable concepts during scheduling is Time Quantum (Panda and Sanjaya, 2014).

Among all the processes there must be present a process that has maximum burst time and minimum burst time and the Time Quantum had been determined based on their difference. To do various optimizations in terms of decrement of waiting time, response time besides the increment of desired output, proper utilizing capacity of CPU, scheduling can be done by making more improvement on Round Robin according to the shortest quantum time of a job among the running jobs in CPU (Yadav and Rakesh, 2010). CPU is a resource allocator and makes the best use of resources based on their efficient use of time. The preferable method for this is scheduling. In terms of multitasking and multi-programming occurrences, such efficient allocation is necessary for resources (Ajit and Batra, 2010).

Proposed Round Robin Algorithm

The proposed method is precise and close to the desired requirements and it is eligible. The consecutive steps have been published through the proposed algorithm addition to a brief description concerning the algorithm. Firstly, all the processes of the ready queue are sorted according to their burst time in an increasing order. Secondly, the differences between two adjacent processes such as P_1-$P_{2,}$ P_2-$P_{3,}$ ……………, P_{n-1}-P_n are calculated and measured the maximum differences among these differences. After that the time quantum is calculated. The term time quantum refers to the absolute sum of the maximum difference and the first process burst time among the sorted processes. The proposed methodology contains the following steps:

- Step 1: According to the Burst Time, all the processes of the job queue need to be sorted in an increasing order.
- Step 2: The differences between two adjacent processes are required to calculate and find the maximum difference among these.
- Step 3: IF process Burst Time - Time Quantum = 0, the process will be terminated.
- Step 4: IF process Burst Time – Time Quantum != 0, the process will be executed based on the time quantum and the process will be pushed at the tail of the queue as well as step 3 to step 4 need to continue without any interruption until all the processes of the series remain completed.
- Step 5: Turn-Around-Time and Waiting-Time of the processes of the sequence of processes will be calculated.

The proposed algorithm clarifies all the required and desired processes related to implementation ofthe programming section. In the entire program, there are two inputs: one is number of processes (N) and Burst Time (BT) for each process. According to algorithm the desired outputs are Turn Around Time (TAT) and Waiting Time (WT). Context Switching (CS) has been calculated from the illustrated Gantt Chart. A sorting among all processes are required based on BT.

Among the sorted array of BT, differences have been calculated and the maximum difference value among the processes have been determined. The whole calculation has been done in three major steps.

Using Equation 1 the differences among all processes will be calculated and the maximum difference will be determined using the Equation 2. Finally, Equation three will be used to calculte the time quantum.

$$diff := BT[i+1] - BT[i] \quad \ldots\ldots\ldots\ldots\ldots \quad (1)$$

$$MAX := \max(MAX, diff) \quad \ldots\ldots\ldots\ldots\ldots \quad (2)$$

$$TQ := MAX + BT[0] \quad \ldots\ldots\ldots\ldots\ldots \quad (3)$$

Then the most noticeable point TQ has been selected by determining the ultimate maximum differences of all processes based on the maximum difference between two adjacent processes. After that all the processes had been completed according to the defined TQ.

Algorithm 1: Optimized Round Robin

```
Input: BT, N // BT – Burst Time
Output: TAT, WT // TAT – Turn Around Time, WT – Waiting Time
Initialize : MAX ← 0, WT ← 0
Begin
    Sort(BT, BT + N)
    for i ← 0 to N - 1 do
            diff ← BT[i + 1] − BT[i]
            MAX ← max(MAX, diff)
    end for
TQ ← MAX + BT[0]
while Queue != 0 do
    front ← Queue: front
    if TQ  front then
            sum ← sum + front
            TAT: pushback(sum)
            TAT_total ← TAT_total + sum
            Queue: pop
    else
            front ← front − TQ
            sum ← sum + TQ
            Queue. pop
            Queue. push (front)
    end if
end while
for i ← 0 to N − 1 do
    TAT ← finish[i] − BT[i];
end for
for i ← 0 to N − 1 do
    WT ← (WT + TAT[i]) − BT[i]
end for
End
```

The optimized RR algorithm is shown in the Algorithm 1. The algorithm takes burst time as input and provides turn-around-time, waiting-time and context-switching as output. Inputs and outputs have been mentioned in (Algorithm 1 the lines 1-3). The required sorting of all inputs is presented in (Algorithm 1 line 5) and the maximum difference between two adjacent processes has been calculated (Algorithm 1 lines 6-9). Then, the considerable term time quantum is calculated (Algorithm 1 line 10). In the lines 11-24, the entire program has been implemented in a preemptive manner. Finally, our desired outputs turn-around-time and waiting-time have been calculated in (Algorithm 1 lines 25-29). The context-switching has been measured with the help of Gantt Chart. Based on this proposed methodology an experiment has been conducted to evaluate the result which is presented in the result section.

Results and Discussion

The proposed algorithm has been developed in order to increase the performance of scheduling by accomplishing some particular criteria such as Maximum Throughput, Minimum Turn-Around-Time (TAT), Minimum Waiting-Time (WT), decrement of Context-Switching (CS) etc. The proposed algorithm had been implemented with C++ along with 64bit windows10 operating system, with Intel core i7, 4GB RAM.

To evaluate the proposed method, experiment was conducted with three different case studies and three sets of data by comparing some other existing proposed methods. These data sets had been utilized by different method previously (Pradhan and Ray, 2016), (Matarneh and Rami, 2009), (Arif and Javed, 2016) and selected from existing literature to avoid biasness of the experiment. Each data set consists of five processes but if the number of processes is increased, the result will not be changed largely acquired from the experiment.

To make a proficient comparison following existing algorithms had been selected.

- Modified Round Robin Algorithm (MRRA) (Pradhan and Ray, 2016)
- Self-Adjustment Round Robin (SARR) (Matarneh and Rami, 2009)
- Median Based Round Robin (MBRR) (Arif and Javed, 2016)
- Modulus Based Round Robin (MoBRR) (Arif and Salman, 2015)
- Improved Round Robin Scheduling (IRRS) (Nayak, 2012)

Case Study 1

Table-1 presents the first data set which is consisted of 5 different processes. All the processes have different burst times but same arrival time that is zero (0). In this section, an explicit Gantt Chart and a comparison table with existing algorithms had been demonstrated against TAT, WT, CS.

Table 1. Data Set 1.

Process ID	Arrival Time (AT)	Burst Time (BT)
P1	0	105
P2	0	60
P3	0	120
P4	0	48
P5	0	75

According to the mentioned data set, after sorting and based on the equation (1) and (2), the differences between two adjacent processes are 12, 15, 30, 15. The maximum difference among these is 30. After that according to the equation (3), the appropriate time quantum will be (30+48) = 78. Then a Gantt Chart Table-2 that represents clearly the finishing time of all processes based on selected TQ.

Table 2. Gantt-Chart on data set 1.

P4	P2	P5	P1	P3	P1	P3
48	108	183	261	339	366	408

Here, a comparison Table-3 has been demonstrated with proposed algorithm against mentioned algorithms on the result of turn-around-time, waiting-time and context-switching. By a close concentration, our proposed algorithm is comparatively better and efficient against all other RR algorithms.

The term turn-around-time is explicitly noticeable that by applying the proposed method it had been reduced along with waiting-time. One of the considerable term context-switching which had been proportionally minimized with the proposed algorithm.

Table 3. Comparison results over data set 1.

Algorithms	TAT	WAT	CS
MRRA	223.2	141.6	8
SARR	259.2	177.6	5
MBRR	222.6	141	7
MoBRR	222.6	141	8
IRRS	226.4	144.8	8
Proposed Algorithm	**222.6**	**141**	**7**

Case Study 2

The second data set has been arranged in Table - 4 in order to test and measure the efficiency and effectiveness of the proposed algorithm than various former RR algorithms. The data set is consisted of 5 different processes. All the processes have different burst times but same arrival time that is zero (0). An explicit Gantt-Chart and a comparison table with existing algorithms had been demonstrated against TAT, WT and CS.

Table 4. Data Set 2.

Process ID	Arrival Time (AT)	Burst Time (BT)
P1	0	24
P2	0	20
P3	0	08
P4	0	10
P5	0	03

According to the mentioned data set, after sorting and based on the equation (1) and (2), the differences between two adjacent processes are 5, 2, 10, 4. The maximum difference among these is 10. After that according to the equation (3), the appropriate time quantum will be (10+03) = 13. Then a Gantt Chart Table-5 that represents clearly the finishing time of all processes based on selected TQ.

Table 5. Gantt-Chart on data set 2.

P5	P3	P4	P2	P1	P2	P1
03	11	21	34	47	54	65

A noticeable comparison on Table-6 like first data set has been displayed with proposed algorithm against several algorithms on the result of turn-around-time, waiting-time and context-switching. With a meticulous eye, our proposed algorithm is comparatively better and efficient against all other RR algorithms.

The term turn-around-time is explicitly noticeable that by applying the proposed method it had been reduced along with waiting-time. One of the considerable term context-switching which had been proportionally minimized with the proposed algorithm.

Table 6. Comparison result over data set 2.

Algorithms	TAT	WAT	CS
MRRA	30.8	17.8	8
SARR	30.2	17.2	8
MBRR	31	18	7
MoBRR	30.8	17.8	8
IRRS	31.6	18.6	7
Proposed Algorithm	**30.8**	**17.8**	**7**

Case Study 3

Table – 7 represents the third data set which is consisted of 5 different processes. All the processes have different burst times but same arrival time that is zero (0). An explicit Gantt Chart and a comparison table with existing algorithms had been demonstrated against TAT, WT and CS.

Table 7. Data Set 3.

Process ID	Arrival Time (AT)	Burst Time (BT)
P1	0	26
P2	0	03
P3	0	74
P4	0	13
P5	0	47

According to the mentioned data set, after sorting and based on the equation (1) and (2), the differences between two adjacent processes are 10, 13, 21, 27. The maximum difference among these is 27. After that according to the equation (3), the appropriate time quantum will be (27+03) = 30. Then a Gantt Chart Table-8 that represents clearly the finishing time of all processes based on selected TQ.

Table 8. Gantt-Chart on data set 3.

P2	P4	P1	P5	P3	P5	P3	P3
03	16	42	72	102	119	149	163

A considerable comparison on Table-9 like first data set and second data set has been displayed with proposed algorithm against several algorithms on the result of turn-around-time, waiting-time and context-switching. With a conscious eye, our proposed algorithm is comparatively better and efficient against all other RR algorithms. The term Turn-Around-Time is explicitly noticeable by applying the proposed method. It had been reduced along with waiting-time.

Table 9. Comparison result over data set 3.

Algorithms	TAT	WAT	CS
MRRA	69	36.4	8
SARR	68.2	35.8	8
MBRR	70	37	8
MoBRR	69	36.2	8
IRRS	71.6	38.8	8
Proposed Algorithm	**68.6**	**36**	**8**

In a concise statement, it is possible to say that for the first data set on Table-1, the proposed algorithm is excellent and preferable according to Table-3 in terms of Total-Around-Time, Waiting-Time and Context-Switching than other algorithms except self-adjustment-round-robin-algorithm at the point of context-switching. On the other hand, for the second data set on Table-4, the proposed method is comparatively acceptable based on Table-6 in terms of turn-around-time, waiting-time and context-switching than other algorithms except self-adjustment-round-robin-algorithm at the point of turn-around-time and waiting-time.

Moreover, the third data set on Table-7, our proposed algorithm is comparatively better than all other former algorithm according to Table-9 except self-adjustment-round-robin-algorithm at the term of turn-around-time and waiting-time but context-switching is same for all. Even, the proposed methodology had provided better experimental consequences on various data set and, these comparative result was better than mentioned round robin algorithms. Since time-quantum is the most noticeable term in this region, we did a profound analysis on determining an optimal time quantum to make it optimized. Eventually, it is decided that the proposed algorithm is fully optimized and essential in enhancing the performance of a system regarding scheduling of CPU in the operating system.

Table 10. Comparative result analysis over large dataset.

Dataset	No. Of Processes	Contexts	MRRA	SARR	MBRR	MoBRR	IRRS	Proposed Algorithm
Set 1	20	TAT	459.80	458.20	458.60	459.65	459.95	**457.30**
		AWT	408.80	406.80	410.80	411.20	410.20	**406.80**
		CS	28	29	27	28	27	**27**
Set 2	25	TAT	462.75	466.80	465.20	466.55	468.30	**458.80**
		AWT	411.80	416.96	415.25	412.98	415.10	**406.80**
		CS	29	29	30	29	30	**27**
Set 3	30	TAT	476.80	479.25	475.90	476.30	479.20	**459.25**
		AWT	425.20	422.60	425.70	421.28	424.80	**408.90**
		CS	37	37	36	37	38	**31**

In terms of increasing the number of processes into the ready queue, the performance of our proposed algorithm is comparatively better and efficient. To analysis the performance, 20 processes, 25 processes and 30 processes have been accumulated successively. Basically, not more than 5 processes are arrived into the ready queue at a time for execution.

Table 10 specifies the comparison with the parameter of TAT, WAT and CS for large amount of data approximately 20 processes, 25 processes and 30 processes. With a close concentration, it has been found that our proposed algorithm is relatively proficient.

A comparison diagram based on the dataset two from the table is presented on the figure 2 for a clear view of the improvement of our proposed algorithm.

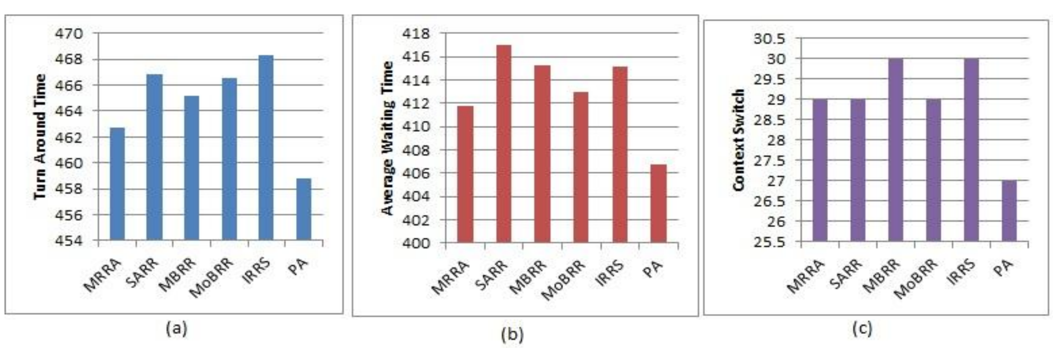

Fig. 2. Comparison with 5 existing algorithms based on data set 2 from the table 10 (a) Average turnaround time (b) Average waiting time and (c) Context switching.

Conclusion

Resource allocation is a great challenge indeed to enrich and make the best and significant utilization of CPU in the operating system. Though Round Robin algorithm is one of the solutions for resource scheduling which one is based on Time Quantum, the question is raised on the bottleneck facing of round robin. Multiple algorithms are proposed in the existing literature to optimize this time quantum. Our proposed algorithm is a trivial effort with a view to increasing the system performance as well as the effective use of resources which one is based on the maximum difference between two adjacent processes in a consequence. This proposed method is concerned not only performance but also minimizing the turn-around-time, decreasing the waiting-time, increasing the number of throughput and so on. By applying and deploying our program it is possible to enhance and improve the resource distribution and allocation through system appropriately. Experimental result shows that the proposed algorithm is reliable and effective in terms of increasing performance of operating system. Scheduling performance will be more efficient by determining an optimal time quantum. Future work can be based on allocating dynamic time quantum and adjusting with other scheduling algorithms which will help to increase the overall performance of a system.

References

Fataniya, Bhavin, and Manoj Patel. 2018. Dynamic Time Quantum Approach to Improve Round Robin Scheduling Algorithm in Cloud Environment. ijsrset. **4:** 963-969

Pradhan, Pandaba, Prafulla Ku Behera, and B. N. B. Ray. 2016. Modified round robin algorithm for resource allocation in cloud computing. Procedia Computer Science. **85**: 878-890.

Matarneh, Rami J. 2009. Self-adjustment time quantum in round robin algorithm depending on burst time of the now running processes. American Journal of Applied Sciences. **1831**: 6-10.

Alsheikhy, Ahmed, Reda Ammar, and Raafat Elfouly. 2015. An improved dynamic Round Robin scheduling algorithm based on a variant quantum time. Computer Engineering Conference (ICENCO), 11th International. IEEE. 98-104

Nayak, Debashree, Sanjeev Kumar Malla, and Debashree Debadarshini. 2012. Improved round robin scheduling using dynamic time quantum. International Journal of Computer Applications. **2012:** 0975-8887.

Chavan, S. R., P. C. Tikekar. 2013. An Improved Optimum Multilevel Dynamic Round Robin Scheduling Algorithm. International Journal of Scientific and Engineering Research. **4.12**: 298-301.

Yadav, Rakesh Kumar, et al. 2010.An improved round robin scheduling algorithm for CPU scheduling. International Journal on Computer Science and Engineering. **2.04**: 1064-1066.

Singh, Ajit, Priyanka Goyal, and Sahil Batra. 2010. An optimized round robin scheduling algorithm for CPU scheduling. International Journal on Computer Science and Engineering. **2.07**: 2383-2385.

Noon, Abbas, Ali Kalakech, and Seifedine Kadry. 2011. A new round robin based scheduling algorithm for operating systems: dynamic quantum using the mean average. arXivpreprint arXiv:1111.5348.

Arif, Salman, Saad Rehman, and Farhan Riaz. 2015. Design of a modulus based Round Robin scheduling algorithm. Software Engineering Conference (MySEC), IEEE. 230-235

Panda, Sanjaya Kumar, and Sourav Kumar Bhoi. 2014. An effective round robin algorithm using min-max dispersion measure. arXiv preprint arXiv:1404.5869.

Arif, Salman, NaveedGhaffar, and Ali Javed. 2016. Implementation of alternating median based round robin scheduling algorithm. Computer and Information Technology (CIT), IEEE. 154-160

Pallab Banerjee, Probal Banerjee, Sonali Dhal. 2012. Comparative performance Analysis of Average Max Round Robin Scheduling Algorithm using Dynamic Time Quantum with Round Robin Scheduling Algorithm using static time Quantum. IJ. of Innovative Technology & Exploring Engineering. **1:** 3.

Manish Kumar Mishra, Faizur Rashid. 2014. An improved Round Robi CPU Scheduling Algorithm with varying time quantum. IJCSEA Journal. **4:** 4.

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. **37:** 1-16.