ISSN 2278-3083
Volume 8, No.6, November –December 2019
International Journal of Science and Applied Information Technology
Available Online at http://www.warse.org/ijsait/static/pdf/file/ijsait10862019.pdf
https://doi.org/10.30534/ijsait/2019/108620198

Improved performance for Round Robin Scheduling algorithm depends on burst time

Nawal Hamdan[1], ***Jumana Al Shweiki***[2] ***Mohammad Nassar***[3]
[1]Departent of computer science, Amman Arab University, Amman, Jordan

ABSTRACT

Operating system acts as an interface between user and hardware, there are many processes included into the operating system, these processes can be scheduled by different algorithms, scheduling is a very important process of the computer resources in the central processing unit (CPU) , it controls all jobs and select which process can be start the first, each algorithm uses different ways to calculate it's criteria like first come first served (FCFS), shortest job first (SJF), Priority Scheduling, and Round Robin (RR). Round robin algorithm depends on a minor factor called quantum time, this quantum maybe determined at the beginning or it can be calculated by different ways. This paper proposes a new way to improve performance for round robin by arranging processes in an increasing order according to the burst time, then calculating the mean of the burst time as quantum time finally calculating turnaround time and waiting time, thus to obtain better results comparing with the standard RR.

Key words: CPU Scheduling, Round Robin, Mean (avg), Waiting Time, Turn Around Time.

1. INTRODUCTION

Round Robin is one of scheduling algorithms uses for arranging time-sharing environments; it is similar to first come first served algorithm but with preemptive capability to switch between processes.

Each process obtains a small unit of CPU time (time quantum), there are many ways used to determine this quantum time, this is done from the top to the bottom of a process list, then starting again from the top of the list and so until all processes are completed.

Since round robin algorithm relying on quantum time , it's concept is to determine this time by the user, when the first process burst's time is long than quantum time, so the process will be still in the memory till it is completed , in this paper we suggested a new scheme to improve the performance of the round robin algorithm, this can be done by calculating the mean according burst time as a quantum time after arranging the burst time in an increasing order, then we addressed different examples using the standard RR and using our proposed scheme, and compared the results between them such like average waiting time and average turnaround time [1] [2].

2. THEORETICAL

There are many CPU scheduling algorithms available, each have its own properties, and to choose an algorithm among the available algorithms, we have to take into consideration the below criteria:

1. CPU utilization: The optimum usage for the computer resources [1].
2. Throughput: "number of processes that complete their execution per time unit" [1].
3. Turnaround time: sum of waiting time and burst time for a single process [3].
4. Waiting time: "amount of time a process has been waiting in the ready queue" [3].
5. Response time: "amount of time it takes from a request submitted until the first response is produced" [3].

Good scheduling algorithm must achieve the following characteristics:
1. Maximum CPU utilization.
2. Maximum Throughput.
3. Minimum turnaround time.
4. Minimum waiting time.
5. Minimum response time.

2.1 Literature Review

The previous studies talk about round robin and how it can be improved to get better performance in many ways, however there are many researches illustrate how to improve RR performance. Below is a List for some of research papers which adopted Round Robin improvements:

Rajput, I. S., & Gupta, D. Concentrate on disadvantage of simple round robin, which gives the same priority to all process [4].

Singh, A., Goyal, P., & Batra, S their suggested algorithm is implemented in three steps in which to help minimized the number of performance parameter like context switch. [5].

Yaashuwanth, and R. Ramesh A Modern Round Robin algorithm calculate the quantum time for every task for Round Robin algorithm. The throughput of the system is increased because the total number of switch is decreased [6,12].
Rami J. Matarneh their proposed algorithms called Self-Adjustment-Round-Robin rely on a new technique called dynamic-time-quantum; the main idea of this technique is to make the time quantum repeatedly adjusted based on burst time of the now-running processes [7].
Tarek Helmy, Abdelkader Dekdoukwe suggests new approach for round robin to attempt to mix low scheduling and overhead of round robin. The target for this approach to obtain better throughput and waiting time.

Round Robin

The following table (1) is the first example for CPU scheduling processes using the standard round robin, which contains 5 processes with the same arrival time (which is zero) by using a static quantum time of (5):

Table 1: An example with quantum time of 5

Process	Burst time
A	28
B	20
C	12
D	18
E	6

Gant Chart for the example one as figure (1) below:

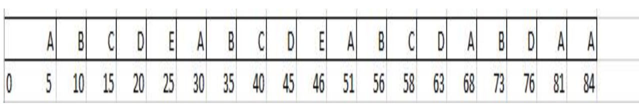

Figure 1: Gant chart

The waiting time as shown in Table (2)

Table 2: Waiting time for example No. 1

Process	Waiting time
A	61
B	53
C	46
D	58
E	40

Average waiting time: (61+53+46+58+40) /5= 51.6
Average turnaround time: (61+28) + (53+20) + (46+12) + (58+18) + (40+6) = (102+81+58+76+46)/5=342/5=68.4

3. METHODOLOGY

Proposed Scheme

Many researches illustrate different approaches to calculate quantum time to obtain best performance either to minimize mean (Avg) turnaround time or to minimize waiting time.
Our approach, we suggest a new way to obtain better performance of Round robin scheduling algorithm. The processes arrange in an increasing order based on burst time and then by calculating mean (Avg) for the burst time as quantum time [9] [10],
The following chart shows how the algorithm works as figure (2).

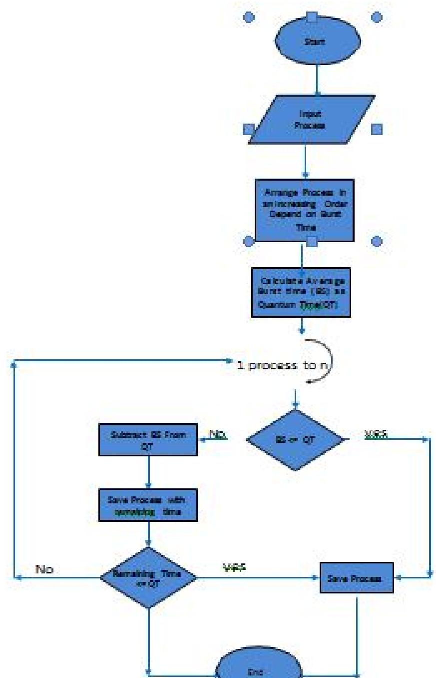

Figure 2: the proposed Scheme

4. RESULTANTS & DISCUSSION

The below steps illustrate RR results for the example one that was mentioned before using our proposed scheme:

Step 1: Arranging processes in an increasing order relying on burst time as shown in table (3).

Table 3: Arranging processes in an increasing order

Process	Burst time
E	6
C	12
D	18
B	20
A	28

Step 2: calculating average burst time then using it as quantum time: Average burst time = (6+12+18+20+28) /5=17

Below figure 3 is the Gant chart with quantum time= 17.

E	C	D	B	A	D	B	A
0 6	18	35	52	69	70	73	84

Figure 3: Gant chart.

Step 3: calculating the waiting time and turnaround time to compare the results with the simple round robin in previous example one shown below in table 4.

Table 4: Waiting time using the proposed scheme

Process	Waiting time
E	0
C	6
D	52
B	53
A	56

Average waiting time= (0+6+52+53+56)/5=33.4 Average turnaround time= (0 + (6+12) + (52+18) + (53+20) + (56+28) = 251/5=50.2

We illustrated below a comparison between simple RR and the proposed scheme shown as table (5).

Table 5: comparison between simple round robin and proposed Scheme

Algorithm	Quantum Time	Avg waiting Time	Avg Turnaround time
Round Robin	5	51.6	68.4
Proposed RR Scheme	17	33.4	50.2

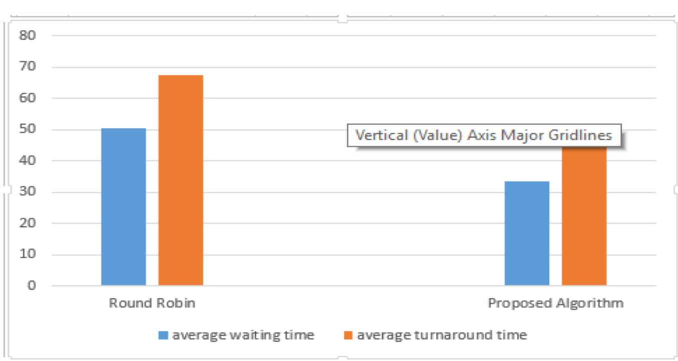

Figure 4: Comparison of AWT, ATT for proposed scheme

Example 2: in this case we used another example, by applying simple round robin with quantum time =10 and arrival time = zero, below table 6 shows the results.

Table 6: Burst time for example no. 2

process	Burst time
A	23
B	15
C	4
D	8
E	13
F	28

Gant Chart

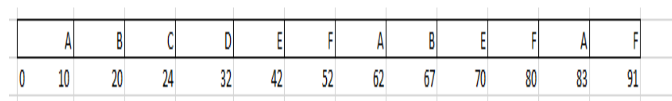

Figure 5: Gant Chart for example no. 2

Then calculating waiting time as shown in Table 7

Table 7: Waiting time for example no. 2

Process	Waiting time
A	60
B	52
C	20
D	24
E	57
F	63

Average waiting time= (60+52+20+24+57+63) /6= 46

Average turnaround time= (60+23) + (52+15) + (20+4) + (24+8) + (57+13) + (63+28) = 61.17

We also applied the proposed RR scheme for the below example 3, burst time calculations shown in the following table 8.

Table 8: Burst time for example no.3

process	Burst time
C	4
D	8
E	13
B	15
A	23
F	28

Then we calculate average burst time which is =15.1 ≈15 In addition, Quantum time= 15.
Gant Chart

C	D	E	B	A	F	A	F
0 4	12	25	40	55	70	78	91

Figure 6: Gant Chart for example no.3

Calculating waiting time as shown in Table (9)

Table 9: Waiting time for example no. 3

Process	Waiting time
C	0
D	4
E	12
B	25
A	48
F	62

Average waiting time= (0+4+12+25+48+62) /6=25.16
Average turnaround time= (0+4) + (4+8) + (12+13) + (25+15) + (48+23) + (62+28) = 40.3

Comparison result between Round Robin and Proposed Algorithm shown as table (10) and figure (7)

Table 10: comparison between round robin and proposed Algorithm

Algorithm	Quantum Time	Avg waiting Time	Avg Turnaround time
Round Robin	5	46	61.17
Proposed Algorithm	17	25.16	40.3

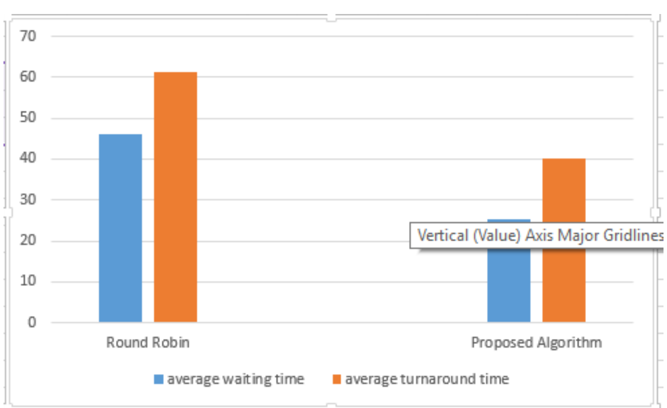

Figure 7: Comparison of AWT, ATT for proposed scheduling algorithm

5. CONCLUSION

From the above comparison, we observe that using our proposed scheme is better than using simple round robin algorithm, since the results showed better performance in terms of reducing in the average waiting time and turnaround time.

REFERENCES

1. Silberschatz, A., Galvin, P. B., Gagne, G., & Silberschatz, A. (1998).Operating system concepts (Vol. 4). Reading: Addison- wesley.
2. Singh, Pawan, and Berhane Wolde-Gabriel. "Executed-time Round Robin: EtRR an online non-clairvoyant scheduling on speed bounded processor with energy management." Journal of King Saud University-Computer and Information Sciences 29.1 (2017): 74-84. https://doi.org/10.1016/j.jksuci.2016.03.002
3. Saeidi, S., & Baktash, H. A. (2012). Determining the Optimum Time Quantum Value in Round Robin Process Scheduling Method. International Journal of Information Technology and Computer Science (IJITCS), 4(10), 67.
4. Rajput, I. S., & Gupta, D. (2012). A priority based round robin CPU scheduling algorithm for real time systems. International Journal of Innovations in Engineering and Technology, 1(3), 1-11.
5. Singh, A., Goyal, P., & Batra, S. (2010). An optimized round robin scheduling algorithm for CPU scheduling. International Journal on Computer Science and Engineering, 2(07), 2383-2385.
6. C. Yaashuwanth, and R. Ramesh, "Intelligent Time Slice for Round Robin in Real Time Operating Systems", IJRRAS, 2(2), 126-131, 2010
7. Matarneh, R. J. (2009). Self-adjustment time quantum in round robin algorithm depending on burst time of the now running processes. American Journal of Applied Sciences, 6(10), 1831.
8. Helmy, T., & Dekdouk, A. (2007). Burst round robin as a proportional-share scheduling algorithm. IEEEGCC 2007.
9. Xu, Y., Lu, R., Shi, P., Li, H., & Xie, S. (2016). Finite-Time Distributed State Estimation Over Sensor Networks With Round- Robin Protocol and Fading Channels. IEEE Transactions on Cybernetics.
10. Rajput, I. S., & Gupta, D. (2012). A priority based round robin
11. CPU scheduling algorithm for real time systems. International Journal of Innovations in Engineering and Technology, 1(3), 1-11.
12. Abualigah, L. M. Q. (2019). *Feature selection and enhanced krill herd algorithm for text document clustering*. Berlin: Springer. https://doi.org/10.1007/978-3-030-10674-4