Improved Round Robin Algorithm: Proposed Method to Apply SJF using Geometric Mean

Omar Hani Mohammad Dorgham
Computer Science and Informatics. Amman Arab University
Amman, Jordan
omar.dorgham1992@outlook.com

Dr. Mohammad Othman Nassar
Computer Science and Informatics. Amman Arab University
Amman, Jordan
moanassar@aau.edu.jo

***Abstract*— One of the most important components of the computer resources is the CPU, which is the basis of the operating system, and the major concept of the CPU is the CPU scheduling which control all jobs in the operating system using algorithms like First-Come-First-Serve (FCFS), Shortest Job First (SJF), Priority Scheduling, and Round Robin (RR) algorithms. It is wellknown that all CPU schedule algorithms concentrate on maximizing the CPU utilization and minimizing turnaround time, waiting time, response time, and the context switching. RR algorithm is an algorithm designed for Time-Sharing Systems, on the other hand, RR algorithm allocate processes to the CPU for only one time quantum, and this paper discussed a new algorithm that attempts to set an alternative method for the existing RR algorithm by using Geometric mean as a time quantum and applying the SJF algorithm in choosing the proper process to be allocated for execution.**

Keywords- CPU Scheduling; Round Robin Algorithm; Shortest Job First (SJF); Geometric Mean;

I. INTRODUCTION

Computers were a very complicated machine in terms of usage, accordingly, scientists invent a new way to makes the computers compatible with the human language and easy to use, accordingly, scientists invent the Operating System which can be defined as "is a program that manages a computer's hardware. It also provides a basis for application programs and acts as an intermediary between the computer user and the computer hardware". [1]
Operating System (OS) can be said as an interface between the computer hardware and the user and that's done by the OS software's a neither program nor application software's but until now there is no universal definition for the OS.

Operating System (OS) has a lot of procedures, but in this paper we will focus on the Round Robin (RR) algorithm. Scheduling can be defined as algorithms that coordinate the task in the CPU according to priority, size of the process and its behavior. CPU scheduling can be defined also as a technique by which processes are allocating to the CPU for a specific time quantum.

The CPU Scheduling have an algorithms to coordinate its work like First-Come-First-Served Scheduling, Shortest-Job-First Scheduling, Priority Scheduling, Round-Robin Scheduling, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling algorithms.

II. ROUND ROBIN SCHEDULING ALGORITHM (RR)

Round Robin algorithm is designed especially for time-sharing System, which can be defined as "Time-Sharin System allows many people to simultaneously use a centrally located computer from terminal devices for problem definition, problem solution, or program editing." [2].

Round Robin Algorithm has the similar policy of the FCFS "First Come First Serve" algorithm, which it depends on a time quantum usually between 10 to 100 milliseconds in length. And, it use a queue to enhance the algorithm procedure, and that's done by allocating the CPU to each process for only one time quantum, so its apply the FCFS policy and that's by adding the new processes to the tail of the queue, then the CPU Scheduler take the first process and sets a timer for the interruption after 1 time quantum, and then dispatches the process.

Therefore, two cases may happen here, first one occur if the process needs less than 1 time quantum, the process will

leave the CPU in a voluntarily manner then the scheduler will proceed to the next process according to the ready queue, but if the process needs more than 1 time quantum, then the timer will cause an interruption for the operating system, and the scheduler will proceed to the next process in the ready queue. On the other hand, any scheduling algorithm has some characteristics like Average Waiting Time and Turnaround Time, in the following example we will considered these characteristics and understand how the RR works.

Consider the following set of process that arrives at time 0, with time quantum 5 seconds:

EXAMPLE OF RR ALGORITHM TABLE I.

Burst Time	*Process*
30	P1
5	P2
7	P3

When we use a time quantum 5 seconds, then process P1 come in the CPU first and takes the first 5 seconds, and according to its burst time, it needs more than 1 time quantum, therefore, the timer will cause an interrupt as we mentioned above, and then the next process in the ready queue will allocated to the CPU which it is P2.

P2 according to its burst time needs 1 time quantum, therefore, it will ends here and the release the CPU voluntarily as we mentioned.

Now, its P3 turn, P3 need more than 1 time quantum so the scheduler will cause an interrupt and allocate the next process in the ready queue, but in this case, there is no new process but there is an old one which is still waiting to finished which it is P1, so the next process to be allocated is P1, and so on.

P1	P1	P1	P1	P1	P3	P2	P1
40	35	30	25	20	15	10	5 0

After we considered the CPU scheduling let us calculate the AWT "Average Waiting Time" for these processes.

P1, wait for 10 seconds to resume its execution, P2 wait for 5 seconds, P3 wait for 10 seconds, and then P1 resume its execution because it still not finished.

On the other hand, turnaround time can be calculated here but first we need to identify the turnaround time which is can be defined as the time in which the process take from the beginning of the execution to the end of it.

And, in our example the turnaround time for P1 is (40-15-5 = 20), P2 (10-5=5), and P3 (15-10 = 5).

III. RELATED WORK

After we have discussed the round robin algorithm in brief, let us considered the development of this algorithm starting with the Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Processes which considered by Rami J. Matarneh in 2009 [3].

Rami J. Matarneh proposed a new approach based on the status of the processes in the ready queue, by assigning a 0 or 1 to the process. Furthermore, if the status equal 0 that's means the process has entered the CPU for the first time or it has a new operations (updating, deleting….. etc.). In this Case, the running process will be assigned to a counter by the operating system to find its burst time, and remain the execution for this process and the new processes in the same round of time quantum. On the other hand, when the status equal 1, the operating system recalculate the time quantum by the median approach, according to the remaining burst time of all processes in the queue and the new processes.

Accordingly, Rami said "I have found through experience that the optimal time quantum can be presented by the median for the set of the processes in the ready queue". Therefore, he present a formula that describes the calculation operation which can be discussed that if the time quantum equal the median if the median is equal or reater than 25, and equal 25 if the median is less than 25.

However, the algorithm provides that 50% of the processes will be finished its execution in the first round, on the other hand, the remaining 50% will be terminated in the second round according to time quantum calculations which it's done in a repeating manner, etc. until it reached the round 6 which it's the maximum number of rounds without taken into consideration the number of processes or their burst time.

After we considered Rami's approach, let us take another approach which is made by Rakesh Kumar Yadav, Abhishek K. Mishra, Navin Prakash, and Himanshu Sharma in 2010 [4] which they proposed the Improved Round Robin Scheduling Algorithm.

In the proposed algorithm, they combined the shortest job first technique which it is mean that the lowest burst time between the processes in the ready queue, enter the

CPU first. Furthermore, the following steps represents the proposed algorithm:

- ∑ Allocate all processes in the queue.
- ∑ After first quantum time, the algorithm select the shortest job, according to SJF (shortest Job First) algorithm, from the waiting queue and assign the process to CPU.
- ∑ After that the algorithm select the next shortest job and repeat step 2.
- ∑ Repeat step 2,3 until all process finish.

When they apply the proposed algorithm on a set of processes, the results were satisfied, and that's based on the reduction of the total waiting time and the turnaround time. On the other hand, the proposed algorithm is more efficient than the contemporary algorithm and that s by enhancing the CPU utilization which maximize it, and minimize the response time.

In 2012, Debashre Nayak, Sanjeev Kumar Malla, and Debashree Debadarshini proposed a new approach in RR algorithm which they called it, Improved Round Robin Scheduling using Dynamic Time Quantum [5], the proposed algorithm depends on the time quantum which they calculate it by two steps, first, arrange

With Arrival Time:

The main idea here is to choose the proper process according to SJF algorithm. Furthermore, in time

quantum 1, the proposed algorithm choose the lowest burst time, by applying the SJF algorithm,

from the ready queue to CPU to execute it, after the second process arrived, the proposed algorithm

applies the SJF again to choose the lowest burst, and the lowest one will be enter the CPU to the

execution , and so on. But, if there is two or more processes have the same burst time the proposed

processes in an increasing order in terms of their burst time, and then calculate the median, and it depends on the number of processes, if it is odd the median will be calculated as follows:

Y*((n+2)\2), and if the number of processes is even then median= 0.5(Y(n\2)+Y((1+n)\2)

while Y= the number located in the middle of the arranged processes.

After the median calculation, the proposed algorithm is consider to calculate the Optimal Time quantum (OTQ) =)Highest BT + Median)\2. Therefore, the optimal time quantum is assigned to each process in terms to calculate the remaining burst time for the processes in the ready queue.

Also in 2012, another approach designed and implemented by Ishwari Singh Rajput and Deepa Gupta, A Priority Based Round Robin CPU Scheduling Algorithm for Real Time Systems [6].The Proposed algorithm based on the priority scheduling algorithm, which allocate all processes in the CPU in RR manner, based on the given priorities and time quantum. Then, all processes arranged in the ready queue in ascending order, and give these processes new priorities, the shortest remaining CPU burst time is assigned with the highest priority, and then the execution will be start according to the new Priorities.

On the other hand, a new approach performed by P.Surendra Varma in 2012, Improved Shortest T remaining

Burst Round Robin (ISRBRR) Using RMS as its time quantum [7], which use the root-mean-square (RMS) to calculate the time quantum used in the scheduling, and after calculating it, it applies the Shortest-Job-First (SJF) to improve the performance of RR algorithm in terms of turnaround time and waiting time.

In 2014 Abdulrazaq Abdulrahim, Saleh Abdulhadi, and Junaida Sahalu, A New Improved Round Robin (NIRR) CPU Scheduling Algorithm [8].

The Proposed algorithm added a new queue to the procedure and they called it Arrive Queue which contains the arrival processes according to their arrival times, when the first process allocate the CPU, it will use the CPU until finish, then all the processes in the arrive queue will move to the existing queue, which the called it Request Queue, in an ascending order according to their burst times, using time quantum as the ceiling of the average of burst times of the processes in the request queue.

Also in 2014, Aashna Bisht, Mohd Abdul Ahad, and Sielvie Sharma, published a paper which contains calculating time quantum, Calculating Dynamic Time Quantum for Round Robin Process Scheduling Algorithm [9].

They proposed a new approach in calculating the time quantum used in the RR algorithm by determine the minimum and maximum burst times and then determine two variables C, and Z, where C equal the ceiling of maximum – minimum over 2, and Z equal the floor of the summation of the burst times over the number of processes. Furthermore, time quantum equal C when C < Z and equal Z when C > Z.

In 2014 a new approach performed by Radhe Shyam and Sunil Kumar Nandal, the Improved Mean Round Robin with Shortest Job First Scheduling [10]. The algorithm proposed by them is based on the calculation of the time quantum to improve the efficiency of RR algorithm with respect to context switching, average turnaround time, and average waiting time.

The time quantum in this proposed algorithm depends on the square root of the mean and the highest burst multiplied values. The processes will arranged in an ascending order in the ready queue and then the time quantum will assign to these ascending processes in the CPU.

In 2015, Radhe Shyam and Parmod Kumar present the Improved Round Robin with Shortest Job First Scheduling [11] which they proposed a new algorithm using SJF on the scheduling criteria by arrange all the processes in ready queue in an ascending order according to their burst times, the algorithm allocate the first ordered process from the ready queue to the CPU, if the process need less than 1 time quantum, then the process will still running until finish, then remove the process from the CPU and allocate it in the tail of the ready queue.

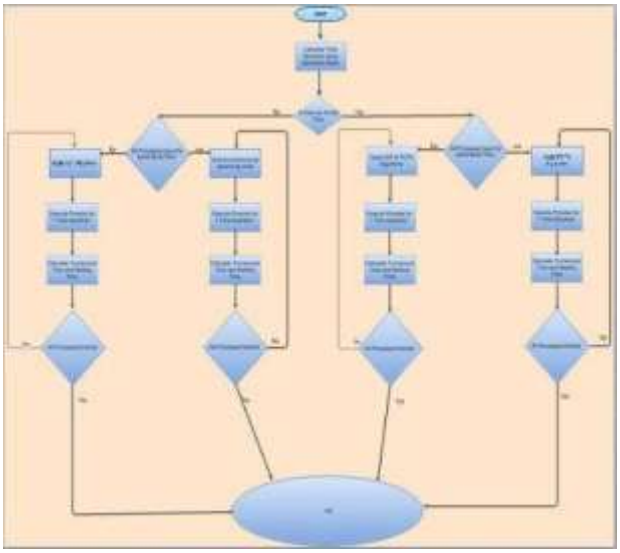
IV. Proposed Algorithm

Based on the algorithm discussed, the proposed algorithm attempts to introduce an alternative method in RR algorithm which combines two algorithms together and calculate a dynamic time quantum using the geometric mean method to enhance the CPU utilization and minimize the waiting and turnaround time in CPU scheduling. Geometric Mean can be calculated by applying the nth root of the product of n numbers, where these numbers is considered as the burst times of the processes, and then take the ceil of the result as a time quantum. On the other hand, the proposed algorithm will take into consideration two cases, first one, when the processes have an arrival times, and the second case, when there is no arrival time. Algorithm will depends on the arrival time, and here the proposed algorithm applies the second algorithm which it is First Come First Serve to take the proper process, where the first process enter the ready queue will be execute first. Accordingly, if there exist a process which need more than 1 time quantum, the remaining burst time will be compared with the other processes burst times by applying the SJF algorithm to choose the proper process to start execution in the CPU.

∑ Without Arrival Time:

In this case the algorithm applies the SJF only to choose the lowest burst time between all processes, where the shortest burst will enter the CPU for the execution and if the lowest burst time needs more than 1 time quantum, the remaining burst time will be compared between other burst times by applying the SJF algorithm, then the proposed algorithm will choose the lowest burst time again.

A. *Proposed Algorithm Flow Chart*



B. *Experiment Discussion*

∑ Example with Arrival Time

Suppose the following set of processes [12].

Table II: Example of RR & Proposed algorithm with Arrival Time

Burst Time	*Arrival Time*	*Process*
23	0	P1
75	20	P2
93	22	P3
48	50	P4
2	55	P5

Using the contemporary algorithm:

Gant Chart

P3	P2	P5	P4	P3	P2	P1

Average turnaround time is =131.8 ms
Average total Waiting Time= 83.6 ms

Using proposed algorithm:

TQ = (23*75*93*48*2)^1/4 = 27.38 ~ 28

Gant Chart

P3	P3	P3	P3	P3	P4	P2	P5	P2	P2	P1

Average Turnaround Time=118.2 ms
Average Total Waiting Time =70 ms

∑ Examples without Arrival Time :

Suppose the following set of processes

TABLE III. EXAMPLE OF RR & PROPOSED ALGORITHM WITHOUT ARRIVAL TIME

Burst Time	*Process*
24	A
20	B
8	C
10	D
3	E

Using the contemporary algorithm:

Gant Chart

A B C D E A B C D A B A

Average turnaround time=146 ms|

Total waiting Time =42.2 ms

Using the proposed algorithm:

TQ = (24*20*8*10*3)^1/5 = 10.28 ~ 11

Gant Chart

E	C	D	B	B	A	A	A		

Average Turnaround Time= 28.2 ms
Average Total Waiting Time= 15.2 ms
Suppose the following set of processes:

TABLE IV EXAMPLE 2 OF RR & PROPOSED WITHOUT ARRIVAL TIME

Burst Time	*Process*
52	P1
22	P2
35	P3
80	P4

Using the contemporary algorithm:

Gant Chart

P4	P4	P3	P4	P3	P2	P1	P4	P3	P2	P1

Average Turnaround Time= 129.25 ms
Average Total Waiting Time= 57 ms

Using the proposed algorithm:

TQ = (52*22*35*80)^1/4 = 42.30 ~ 43

Gant Chart

P4	P4	P1	P1	P3	P2

Average Turnaround Time= 94.25 ms

Average Total Waiting Time= 47 ms

We can see from the above experiments that the total waiting time and average turnaround time reduced, and that's enhance the CPU utilization and minimize the response time. Therefore, we can say that the proposed algorithm can be more efficient than the contemporary one.

V. CONCLUSION AND FUTURE WORK

Based on the previous calculations, it can be said that

the proposed method can enhance somehow the turnaround and response time in terms of improving the operation system scheduling especially in time-sharing system while using the geometric mean in calculating time quantum, and applying the SJF algorithm to choose the proper process to be scheduled and execute. As a future work, I want to implement the proposed method on the turnaround and response time, and i hope the results will be satisfied.

REFERENCES

List and number all bibliographical references in 9-point Times, single-spaced, at the end of your paper. When referenced in the text, enclose the citation number in square brackets, for example [1]. Where appropriate, include the name(s) of editors of referenced books. The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first . . ."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

[1] Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne, (2013), Operating Systems Concepts, John Wiley, Inc., 9th Edition.

[2] Time Sharing System, 2015, available online,"http://www.textfiles.com/bitsavers/pdf/ge/GE2xx/CPB-1455_GE-200_Series_Time-Sharing_Operating_System_Ref_Sep67.pdf ", accessed on 10/6/2015.

[3] Rami J. Matarneh, "Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of now Running Processes", American J. Of Applied Science 6(10):1831 – 1837,2009.

[4] Rakish Kumar Yadav, Abhishek K Mishra, Navin Parakash, and Himanshu Sharma, " An Improved Round Robin Scheduling Algorithm for CPU Scheduling", International Journal on Computer Science and Engineering, Vol.02 No.04, 2010, pp. 1046-1066.

[5] Debashree Nayak, Sanjeev Kumar Malla, and Debashree Debadarshini, " Improved Round Robin Scheduling using Dynamic Time Quantum", International Journal of Computer Applications(09758887),Vo.38, No.5, 2012.

[6] Ishwari Singh Rajput and Deepa Gupta, " A Priority based Round Robin CPU Scheduling Algorithm for Real Time Systems", International Journal of Innovations in Engineering and Technology(IJIET) , Vo.1, Issue 3, 2012, ISSN: 2319 – 1058.

[7] P.Surendra Varma, " Improved Shortest Remaining Burst Round Robin (ISRBRR) Using RMS as its time quantum", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 1, Issue 8, October 2012.

[8] Abdulrazaq Abdulrahim, Saleh E Abdullahi, and Junaidu B. Sahalu, "A New Improved Round Robin (NIRR) CPU Scheduling Algorithm", International Journal of Computer Application (0975-

8887), Volume 90, No. 4, March 2014.

[9] Aashna Bisht, Mohd Abdul Ahad, and Sielvie Sharma, " Calculating Dynamic Time Quantum for Round Robin Process Scheduling Algorithm ", International Journal of Computer Applications (09758887), Volume 98, No. 21, July 2014.

[10] Radhe Shyam and Sunil Jumar Nandal, "Improved Mean Round Robin with Shortest Path Job First Scheduling", International Journal of Advanced Research in Computer Science and Software Engineering, Vo.4, Issue 7, ISSN: 2277 128X, 2014.

[11] Radhe Shyam and Parmod Kumar, "Improved Round Robin with Shortest Job First Scheduling" , International Journal of Advance Research in Computer Science and Software Engineering, Volume 5, Issue 3, March 2015.

[12] Samih M. Mostafa, S.Z. Rida, and Safwat H. Hamad, " Finding Time Quantum Of Round Robin CPU Scheduling Algorithm in General Computing Systems using Integer Programming", International Journal Of Research and Reviews in Applied Science, Volume 5, Issue 1, October 2010.