ISSN 2278-3083

International Journal of Science and Applied Information Technology, Vol.7. No.1, Pages : 12-16 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJSAIT/static/pdf/Issue/icsic2017sp02.pdf

Hybrid Scheduling Algorithms Using Round Robin And Short Job First

Rakan Nawaf Alhawadi[1] , mohammad othman nassar[2] , Omar Al tarawneh[3]
[1]Amman Arab university , Amman , Jordan, rakannawaf@yahoo.com
[2]Amman Arab university , Amman , Jordan, moanassar@aau.edu.jo
[3]Alzahra college for women , Amman , Jordan, Omar.t@zcw.edu.com

ABSTRACT

Scheduling is considered one of the most important topics in operating system. Generally, it helps to choose certain processes for execution. This paper will present a new integrated scheduling algorithm for multiprocessor system using Round Robin and Shortest Job First algorithms. Based on this integration, the result will be presented and compared with the previous algorithms. The new integrated scheduling algorithm was found to be simple and significant by reducing the waiting time. The result shows an improvement in this algorithm up 3.4% in terms of waiting time .

Key words Scheduling algorithms, hybrid algorithms, Round Robin(RR), Shortest Job First (SJF).

1. INTRODUCTION

Operating system is an interface between end user and system hardware, modern operating system become more complex than legacy operating system, especially when the turn of a single task into multiple tasks. CPU scheduling it is one of the most important works by operating system and the process time which is very important and also help in the selection process by which disease execution of several processes exist within the whole queue waiting for their turn to primitive execution.

The Central Processor Unit (CPU) executes the processes within the system operations one after the other, and dividing the time between these operations and serve each Process as necessary [5]. The scheduling algorithms have different properties, and may be one of these algorithms better than the other, the final goal in scheduling is to obtain optimum scheduling. In Single-processor computer systems, the efficiency of the scheduling algorithms rely on fairness, CPU Utilization, CPU Throughput , Turnaround Time , Waiting Time and Response Time [4]. Scheduling tasks on processors is considered one of the most important issues studied to make processors operate without being idle for long time . This increased the interest in studying scheduling and its algorithms [6] .[2] stated that scheduling policy is considered one of the main factors that affect the performance. It helps to choose which task should be selected first from ready queue to run. Round Robin (RR) scheduling algorithm is widely used furthermore, its execution exceptionally Reliability on a Quantum estimate Qt, which is a predefined measure of time appointed by CPU to each errand to be executed. Notwithstanding, the execution debases regarding a normal holding up time (A WT), a normal Pivot time (ATT) and various Setting Switches (NCS). [2] presented new enhanced dynamic Round Robin booking calculation to lessen the normal holding up time, turnaround time and the quantity of setting changes with a specific end goal to enhance the framework general execution. It also talented a comparative analysis between several existing Round Robin algorithms based on the average time for waiting and turn-around and number of context switches. [1] studied the static scheduling issue for the independent tasks on a homogenous multiprocessor system. They develop an algorithm based on Bees Colony Optimization. They compare their algorithm with a previous one inspired also by the bees behavior for the same purpose. Their algorithm shown that the imposed algorithm gives the best solution for the scheduling problem, in most cases, and improves the traditional BCO algorithm.

2. PROBLEM OF STUDY

Task Scheduler is considered as an important issue addressed because All work carried out by processors seeking to reduce the time of completion. Most scheduling algorithms are having trouble finding the optimal time to perform tasks through Distribution blindness processors, as well as the cost consumed by the algorithm to find the optimal solution.

This paper presents a new hybrid algorithm that mix the Round Robin scheduling algorithm with the Short job first Planning algorithm to decrease the Normal Holding up time, turnaround time and the number from claiming setting switches so as on move forward the framework general execution. We will compare between the existing Round Robin algorithms and our proposed hybrid algorithm based on the average time.

ISSN 2278-3083

International Journal of Science and Applied Information Technology, Vol.7. No.1, Pages : 12-16 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJSAIT/static/pdf/Issue/icsic2017sp02.pdf

3. THE PROPOSED ALGORITHM

This section will introduce the proposed hybrid algorithm; the algorithm suggests the merge of two algorithms (Round-Robin , Short job first); we enter the smallest value (the needed execution times for processes) rather than the distribution of the quantum on all operations without considering the value of time needed by each process.

4. THE PROPOSED ALGORITHM STEPS

1. we assume the arrival time of a set of processes to be at the same time.
2. We identify the value of Q (quantum), which means amount of execution of processes evenly.
3. the algorithm rank the process based on their CPU Burst Time (Ascending); thin it takes the smaller value between CPU Burst Times for the processes, then grant it the quantum value, if it is finished then it will move to the next process, if not finished within the quantum granted to it; then it will wait for the next loop. We move to the next process.

Experiment 1 :

Process name	**BT**
Pr1	22ms
Pr2	18ms
Pr3	9ms
Pr4	10ms
Pr5	5ms

According to simple Round Robin scheduling:
Round Robin quantum =5
Gantt chart:

Pr 1	Pr 2	Pr 3	Pr 4	Pr 5	Pr 1	Pr 2	Pr 3	Pr 4	Pr 1	Pr 2	Pr 1	Pr 2	Pr 1

Average waiting time=34 ms

According to The proposed algorithm scheduling:
Round Robin quantum =5
Gantt chart:

Pr 5	Pr 3	Pr 4	Pr 2	Pr 1	Pr 3	Pr 4	Pr 2	Pr 1	Pr 2	Pr 1	Pr 2	Pr 1	Pr 1

Average waiting time=25 ms

The method used	**Average waiting time**
Round Robin	34 ms
The proposed algorithm	25 ms

Experiment 2 :

Process Id	AT	BT
Pr1	0ms	80ms
Pr2	0ms	45ms
Pr3	0ms	62ms
Pr4	0ms	34ms
Pr5	0ms	78ms

According to simple Round Robin scheduling:
Round Robin quantum =25
Gantt chart:

Pr 1	Pr 2	Pr 3	Pr 4	Pr 5	Pr 1	Pr 2	Pr 3	Pr 4	Pr 5	Pr 1	Pr 3

Average waiting time=187.2 ms
According to The proposed algorithm scheduling:
Round Robin quantum =25
Gantt chart:

Pr 4	Pr 2	Pr 3	Pr 5	Pr 1	Pr 4	Pr 2	Pr 3	Pr 5	Pr 1	Pr 3	Pr 5

Average waiting time = 164.6 ms

The method used	**Average waiting time**
Round Robin	187.2 ms
The proposed algorithm	164.6 ms

Experiment 3 :

Process name	**AT**	**BT**
Pr1	0ms	77ms
Pr2	0ms	54ms
Pr3	0ms	45ms
Pr4	0ms	19ms
Pr5	0ms	14ms

According to simple Round Robin scheduling:
Round Robin quantum =25
Gantt chart:

P1	P2	P3	P4	P5	P1	P2	P3	P1	P2	P1

Average waiting time = 117.4 ms
According to The proposed algorithm scheduling:
Round Robin quantum =25

ISSN 2278-3083
International Journal of Science and Applied Information Technology, Vol.7. No.1, Pages : 12-16 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJSAIT/static/pdf/Issue/icsic2017sp02.pdf

Gantt chart:

Pr 5	Pr 4	Pr 3	Pr 2	Pr 1	Pr 3	Pr 2	Pr 1	Pr 2	Pr 1	Pr 1

Average waiting time = 71.4ms

The method used	**Average waiting time**
Round Robin	117.4 ms
The proposed algorithm	71.4 ms

Experiment 4 :

Process name	**AT**	**BT**
Pr1	0ms	105ms
Pr2	0ms	90ms
Pr3	0ms	60ms
Pr4	0ms	45ms
Pr5	0ms	35ms

According to simple Round Robin scheduling:
Round Robin quantum =25
Gantt chart:

Pr 1	Pr 2	Pr 3	Pr 4	Pr 5	Pr 1	Pr 2	Pr 3	Pr 4	Pr 5	Pr 1	Pr 2	Pr 3	Pr 1	Pr 2	Pr 1

Average waiting time = 214 ms
According to The proposed algorithm scheduling:
Round Robin quantum =25
Gantt chart:

Pr 5	Pr 4	Pr 3	Pr 2	Pr 1	Pr 5	Pr 4	Pr 3	Pr 2	Pr 1	Pr 3	Pr 2	Pr 1	Pr 2	Pr 1	Pr 1

Average waiting time = 167 ms .

The method used	**Average waiting time**
Round Robin	214 ms
The proposed algorithm	167 ms

Experiment 5 :

Process name	**AT**	**BT**
Pr1	0ms	92ms
Pr2	0ms	70ms
Pr3	0ms	35ms
Pr4	0ms	40ms
Pr5	0ms	80ms

According to simple Round Robin scheduling:
Round Robin quantum =25
Gantt chart:

Pr 1	Pr 2	Pr 3	Pr 4	Pr 5	Pr 1	Pr 2	Pr 3	Pr 4	Pr 5	Pr 1	Pr 2

Average waiting time = 173.4 ms
According to The proposed algorithm scheduling:
Round Robin quantum =25
Gantt chart:

Pr 3	Pr 4	Pr 2	Pr 5	Pr 1	Pr 3	Pr 4	Pr 2	Pr 5	Pr 1	Pr 2	Pr 5

Average waiting time =166 ms .

The method used	**Average waiting time**
Round Robin	173.4 ms
The proposed algorithm	166 ms

Experiment 6 :

Process name	**AT**	**BT**
Pr1	0ms	24ms
Pr2	0ms	20ms
Pr3	0ms	8ms
Pr4	0ms	10ms
Pr5	0ms	3ms

According to simple Round Robin scheduling:
Round Robin quantum =5
Gantt chart:

Pr 1	Pr 2	Pr 3	Pr 4	Pr 5	Pr 1	Pr 2	Pr 3	Pr 4	Pr 1	Pr 2	Pr 1

Average waiting time = 29.2 ms .
According The proposed algorithm scheduling:
Round Robin quantum =5
Gantt chart:

Pr 5	Pr 3	Pr 4	Pr 2	Pr 1	Pr 3	Pr 4	Pr 2	Pr 1	Pr 2	Pr 1	Pr 2

Average waiting time = 23.2 ms .

The method used	**Average waiting time**
Round Robin	29.2 ms
The proposed algorithm	23.2 ms

We can see the difference in result between the previously proposed algorithm in and the results the proposed algorithm is better than Round Robin.

ISSN 2278-3083

International Journal of Science and Applied Information Technology, Vol.7. No.1, Pages : 12-16 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJSAIT/static/pdf/Issue/icsic2017sp02.pdf

Experimental results

A table showing the difference between the results :

No. of experiment	Round Robin	The proposed algorithm
Experiment 1	34 ms	25 ms
Experiment 2	187.2 ms	164.6 ms
Experiment 3	117.4 ms	71.4 ms
Experiment 4	214 ms	167 ms
Experiment 5	173.4 ms	166 ms
Experiment 6	29.2 ms	23.2 ms

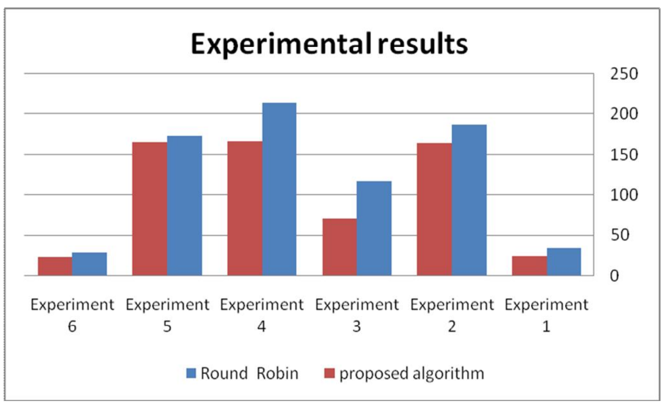

Figure 1: Experimental all results

5. 2.1 PSEUDO CODE THE PROPOSED ALGORITHM

```
burst time[n] ← 0
 arrival time ← 0
numprocess[n] ← 0
turn[n] ← 0
wait[n] ← 0
temp ← 0
current time ← 0
tslice ← 0
waittime = 0
turntime = 0
avgwaititme=0.0
avgturntime=0.0
Read bursttime[n]
For i← 0 to n-1
 For j ← I to n
temp ← burst time[i]
 burst time[i] ← burst time[j]
 burst time[j] ← temp
end for
For i← 0 to n-1
 For j ← I to n
Order burst time[i]
 temp ← burst time[i]
 burst time[i] ← burst time[j]
 burst time[j] ← temp
 end for j
if bursttime[i] != 0
if tslice < bursttime[i]
temp← bursttime[i]< bursttime[j])
bursttime[i← bursttime[i]-tslice[i]
 currenttime← currenttime+tslice
 else
arrivaltime← arrivaltime+bursttime[i];
currenttime← currenttime+ bursttime[i]
bursttime[i← bursttime[i]- bursttime[i]
end else
if bursttime[i] == 0
 turn[i] ← currenttime
end if
end if
else
currenttime++
end else
for i ← 0 to n
wait[i]← turn[i]-burst[i]-arrivaltime
turn[i]← turn[i]-arrivaltime
waittime← waittime+wait[i]
turnrime← turntime+turn[i]
end of for
avgwaittime← waittime/n
avgturntime ← turntime/n
```

5. CONCLUSION AND FUTURE WORK

The study found an efficient algorithm to schedule a series of operations on Optional number of independent processors, and can be applied in the case of this algorithm Processes independent, as well as whether the execution times equal Operations.

The proposed variant of RR algorithm drastically decreases context switching . The proposed algorithm performs better than RR scheduling algorithm with respect to average waiting time . our proposed algorithm can be further in investigated to be useful in providing more and more task-oriented results in future along with developing adaptive algorithms to fit the varying situations in today multifaceted complex working of operating system.

ISSN 2278-3083

International Journal of Science and Applied Information Technology, Vol.7. No.1, Pages : 12-16 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJSAIT/static/pdf/Issue/icsic2017sp02.pdf

REFERENCES

1. Alam, Bashir and Biswas,R (2008) , **Finding Time Quantum of Round Robin CPU Scheduling Algorithm Using Fuzzy Logic** , International Conference on Computer and Electrical Engineering , pp 795-798
2. Alsheikhy, Ahmed & Ammar, Reda ,(2015) , **An Improved Dynamic Round Robin Scheduling Algorithm Based on a Variant Quantum Time** ,IEEE 978-1-5090-0275-7/15.
3. Iraji , M. S,(2015), **"Time Sharing Algorithm with Dynamic Weighted Harmonic Round Robin"**, *Journal of Asian Scientific Research, Vol. 5, No. 3, pp. 131-142.*
4. keqin, l. and yi,pan. (2000), “probabilistic analysis of Scheduling precedence constrained parallel tasks a Multiprocessor with contiguous processor allocation”, *IEEE Trans. Comput. Vol. (49), No. (10), PP. 137-146.*
5. Kwok, Y-K and Ahmed, I. (1999), **“Static Scheduling Algorithm for Allocating Directed Task Graphs to Multiprocessor”**, *J. ACM. Comput. Surv. Vol. (31), No. (4), PP. 406-471.*
6. Oumran , Bassim & Rastanawi , Muhammad (2014) , **Scheduling independent Tasks on homogenous multiprocessors using Bees Algorithm**, *Published research, Al-Baath University, Iraq*
7. H.S.Behera & R.Mohanty & Debashree Nayak (2010) ,**A New Proposed Dynamic Quantum with Re-Adjusted Round Robin Scheduling Algorithm and Its Performance Analysis**, *International Journal of Computer Applications (00975-8887), Vol. (5), No. (5), pp.10-15*
8. Sanjaya Kumar Panda & Sourav Kumar Bhoi (2012),**An Effective Round Robin Algorithm using Min-Max Dispersion Measure**, *International Journal of Computer Science and Engineering (IJCSE), Vol. (4), No. (1), pp.45-53.*
9. Rakesh Kumar Yadav & Abhishek K Mishra & Navin Prakash and Himanshu Sharma (2010),**An Improved Round Robin Scheduling Algorithm for CPU scheduling**, *International Journal of Computer Science and Engineering (IJCSE), Vol. (2), No. (4), pp.1064-1066*