IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 4, April 2021

DOI 10.17148/IJARCCE.2021.10413

Comparison Between Simple Round Robin and Intelligent Round Robin Algorithms in CPU Scheduling

Tri Dharma Putra[1], **Andry Fadjriya**[2]

Department of Informatics, Faculty of Computer Science,

University of Bhayangkara Jakarta Raya, Jalan Perjuangan Bekasi Utara, West Java, Indonesia[1,2]

Abstract: The objective of this journal is to compare the efficiency between these two algorithms, the simple round robin and intelligent round robin algorithms. In real time algorithm, round robin plays a significant role to be use in embedded systems. Here we compare the number of context switching, average turn around time, average waiting time between the two. Intelligent round robin algorithm, is an algorithm with dynamic time quantum. The quantum is calculated and the quantums can be different between tasks. While in simple round robin algorithm, the quantum is the same for every tasks. This comparison proves that intelligent round robin algorithm is more efficient, with lower context switching and lower average turn around time and lower average waiting time. There by this increases the system throughput.

Keywords: Round Robin Algorithm, Intelligent Round Robin Algorithm, Average Turn Around Time, Average Waiting Time, Quantum

I. INTRODUCTION

OS, operating systems provides an interface between a user and computer hardware which helps the user to handle the system in a convenient manner [9]. Operating systems nowadays becomes more complex when they switch from a single task to a multitasking environment. It provides a means for the user to work with computer. Operating system performs major functions like memory management, process scheduling and resource management [4].
Round robin algorithm is a real-time scheduling algorithm in operating system. Dynamic time slice is one solution to find the effective algorithm running in the system. The intelligent time slice (quantum) for round robin architecture for real time operating systems is a modified version of simple round robin scheduling. Simple round robin architecture cannot be implemente in real time operating systems because of high context switch rate, larger waiting time and larger respons time [10]. The standard round robin scheduling algorithm is designed espescially for time sharing system. Each task is assigned a time slice or quantum [8], which is a time interval.

II. LITERATURE SURVEY

Dhruv, proposed in the new modified algorithm, processes are organized in rising order of burst time and dynamic time quantum is used based on shortest remaining burst time. Time quantum plays a significant role in performance of Round Robin Algorithm. If time quantum of algorithm is substansial, Round Robin is similar to FCFS, and if time quantum of given algorithm it will result in extravagant context switches [2].
Lipika Datta proposed, to modify Round Robin algorithm by adjusting time slices (quantum) of different rounds depending on the remaining CPU burst of currently running processes and considering their waiting times until that round in respect of the other processes waiting times. Experimental analysis reveals that the proposed algorithm produces better average turnaround time, average waiting time and fewer number of context switches than existing algorithm [1].
Govind Prasad Arya, Kumar Nilay, Devendra Prasad, proposed an algorithm that categorizes available processes into process with high priority and low priority. This algorithm reduces the average waiting time of process with high priority in all cases and of process with low priority in not all but some cases. The overall waiting time changes on the basis of set of processes considered [3].
Manish Kumar Mishra and Faizur Rashid presented an improved Round Robin CPU scheduling algorithm which enhancing CPU performance using the features of Shortest Job First and Round Robin scheduling with varying time quantum. The proposed algorithm is experimentally proven better than conventional Round Robin [6].
M. LaxmiJeevani, T.S.P. Madhuri, and Y. Sarada Devi, proposed algorithm with a few step. 1)Allocate process to CPU which has arrived first, giving an initial time quantum (k units). 2)After the completion of first process initial execution the scheduler will check the queue for processes which have arrived by the time of completion of first process

DOI 10.17148/IJARCCE.2021.10413

execution with initial time quantum. Next among the multiple processes that have arrived in the queue by the k units of time, the process with the minimum burst time is allocated to the CPU. If the first process has completed its execution, it is removed from the queue or again a chance is provided to it. Lastly, for the complete execution of all processes we have to repeat the same [5].
Neha and Ankita Jiyani proposed an improved round robin algorithm. Take the first process in the ready quue and execute it till the allocated time quantum, then we check the remaining burst time of the running process. If the remaining burst time is less than the allocated time quantum then the running process is executed again till it finishes. But if the remaining burst time of the running process is more than the allocated time quantum, then the next process in the ready queue is executed [7].

III.ORGANIZATION STRUCTURE

This journal comprises of six chapters. The first one is introduction, here we discuss about what is round robin algorithm. In the second chapter, we discuss about some literature survey about round robin algorithm proposed by experts. The third chapter, is about organization structure. In the fourth chapter we discuss the theory behind the intelligent round robin algorithm, which is our main subject. The fifth chapter is about case study analysis of simple round robin and intelligent round robin algorithm. The last chapter is conclusion.

IV. INTELLIGENT ROUND ROBIN ALGORITHM

Round robin algorithm is scheduling algorithm with time quantum. Each process is assigned a time interval which is called time slice or quantum. Round Robin is a preemptive algorithm. If the quantum ends, and the tasks are still executing, then the algorithm will force the next queue to pre-empt the tasks and keep it at the end of the ready queue. The choice of the quantum is critical in this algorithm, since this will give great effect on performance of the algorithm.

In intelligent time slice (quantum) calculation we use these formulas [10]:

- Intelligent Time Slice = Original Time Slice (OTS)+Priority Component (PC)+Shortness Component for CPU burst time (SC)+Context Switch Component (CSC)
- Priority Component (PC) is assigned by the user depending upon the priority which is proportional with the priority number. The first priority get a '1' in the calculation table.
- The Shortness Component (SC) is assigned inversely proportional to the length of the next CPU burst for the process. In SC the current CPU burst is compared with the next CPU burst, if the previous CPU burst is greater, then the next task get a '1', but if the previous CPU burst is smaller then the current one, it will get a '0' in the calculation table.
- Context Switch Component (CSC): CPU Burst – OTS – PC – SC. If the CSC smaller then OTS (original time slice), then the calculation result can be input in the CSC component, but if CSC greater or the same with OTS, then it will be a '0' in the table.

V. CASE STUDY ANALYSIS

A. *Simple Round Robin*

TABLE I TASK BURST TIME AND PRIORITY

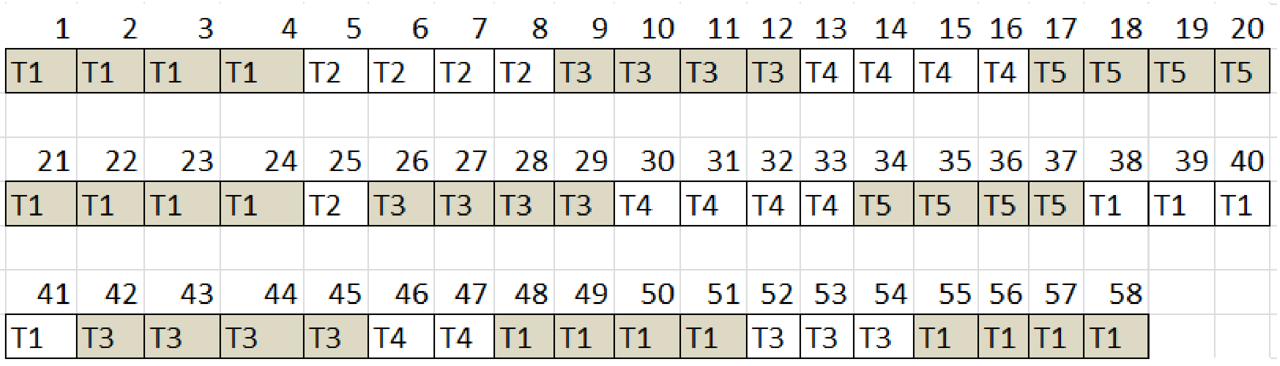
Task ID	CPU Burst Time	Priority
T1	20	2
T2	5	3
T3	15	1
T4	10	2
T5	8	1

The example is with five tasks, T1 until T5. With CPU burst time 20, 5, 15, 10, and 8 ms. Also the priority, with 1 is the highest priority. In simple round robin algorithm there is no priority, so we can skip this column. We use time slice quantum, 4ms.

Please take a look at gantt chart below:

Vol. 10, Issue 4, April 2021

DOI 10.17148/IJARCCE.2021.10413



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
T1	T1	T1	T1	T2	T2	T2	T2	T3	T3	T3	T3	T4	T4	T4	T4	T5	T5	T5	T5
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
T1	T1	T1	T1	T2	T3	T3	T3	T3	T4	T4	T4	T4	T5	T5	T5	T5	T1	T1	T1
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58		
T1	T3	T3	T3	T3	T4	T4	T1	T1	T1	T1	T3	T3	T3	T1	T1	T1	T1		

Fig. 1 Gantt Chart Simple Round Robin

First, T1 is executed until 4, since the quantum is 4. Then T2 is executed until 8. Then T3 followed with 4 quantum until 12, then T4 is executed with 4 quantum, at 16, T5 executed until 20, Then system goes circular back to T1 again. Again T1 is executed with 4 quantum until 24. At 24, T2 is executed, but only 1 quantum since it is just 5 ms, and 4 ms has been executed before. Then T3 is executed until 29. Still with 4 quantum. Then T4 is executed with quantum 4, until 33. After that T5 is executed untill 37, T5 has only 8 ms, so T5 is finished. After T5, then system gets back in circular to T1 again. T1 is executed until 41. Then T3 is executed until 45. The next is T4. Since T4 has only 10 ms and two times has been executed, it only has 2 quantum. Then systems goes on to T5. T5 is finish, no more burst time. Then system gets back circulary to T1. T1 is executed with 4 quantum. Then T2 is skipped, because it already finished. The system then goes on to T3. T3 has only 3 ms left. T3 is executed, until 54. And since T4 and T5 already finished, system goes back to T1, the last one is with quantum 4, Then we comes to an end at 58 ms.

Now we calculate the waiting time, from the gantt chart we concluded:
T1 = 0+(20-4)+(37-24)+(47-41)+(54-52)=38
T2 = 4+(24-8)=20
T3 = 8+(25-12)+(41-29)+(51-45)=39
T4 = 12+(29-16)+(45-33)=37
T5 = 16+(33-20)=29

So, the average waiting time is (38+20+39+37+29)/5=32.6

Next we calculate the turn around time. As per the table below:

TABLE 2 TASK BURST TIME SIMPLE ROUND ROBIN

Process	Arrival Time	Burst Time	Start Time	Finish Time	Turn Around Time
T1	0	20	0	58	58
T2	0	5	4	25	25
T3	0	15	9	54	54
T4	0	10	14	47	47
T5	0	8	19	37	37

So the average turn around time is 221/5=44.2

B. Intelligent Round Robin

From table 3 below, T3 and T5 get a 1 on Priority Component (PC). Please take a look at T1 and T2. For SC (Shortness Component), since CPU burst time of T2 is lower than CPU burst time of T1, which is 20, then T2 gets a 1 in SC. Please take a look at T2 And T3. Since CPU burst time of T3 is higher than CPU burst time of T2, then T3 gets a 0 in SC. Take a look at T3 and T4. CPU burst time for T3 is 15, while for T4 is 10. Since 10 is lower then 15, then T4 get a 1. Lastly, take a look at T4 and T5. The CPU burst time of T4 is 10, and for T5 is 8. Since CPU burst time of T5 is lower then T4, which is 8 compared to 10, then T5 get a 1 in SC.
Next, we calculate the CSC (Context Switch Component). Take a look at T1. Its CPU burst time is 20. 20 minus 4 (OTS) is 16, Since this is higher than OTS (Original Time Slice), then the CSC for T1 is 0. Take a look at T2. It has CPU burst time of 5. CPU burst time minus OTS minus PC and minus SC is : 5 minus 4 minus 1, the result is 0. So, the CSC for T2 is 0. Take a look at T3. The CPU burst time of T3 is 15. Then 15 minus 4 minus 1 (from PC) is 10. Since

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 4, April 2021

DOI 10.17148/IJARCCE.2021.10413

This is greater then OTS, which is 4, then we put a 0 in CSC for T3.Please take a look at T4. Its CPU burst time is 10. Then 10 minus 4 minus 1 (from SC), is 5. So we put a 0 on its CSC since it is still greater then OTS, which is 4. At last take a look at T5. Its CPU burst time is 8. Then the calculation will be 8 minus 4(OTS) minus 1(PC) minus 1(SC), the resut is 2. Since 2 is below 4 (OTS), then we put a 2 in CSC for T5.

TABLE 3 CALCULATED COMPONEN IN INTELLIGENT ROUND ROBIN

Task ID	CPU burst time	Priority	Calculated Time Slice				
			OTS	PC	SC	CSC	New Time Slice
T1	20	2	4	0	0	0	4
T2	5	3	4	0	1	0	5
T3	15	1	4	1	0	0	5
T4	10	2	4	0	1	0	5
T5	8	1	4	1	1	2	8

*OTS: Original Time Slice, PC: Priority Component, SC: Shortness Component, CSC:Context Switch Component

Based on analysis above, we have a different time slice (quantum) for each tasks. Quantum for T1 is 4, while for T2, T3, T4 is 5. And the last one, T5 has 8 time slice (quantum). The calculation is based on formula New Time Slice = OTS+PC+SC+CSC, then we get a new time slice (quantum).

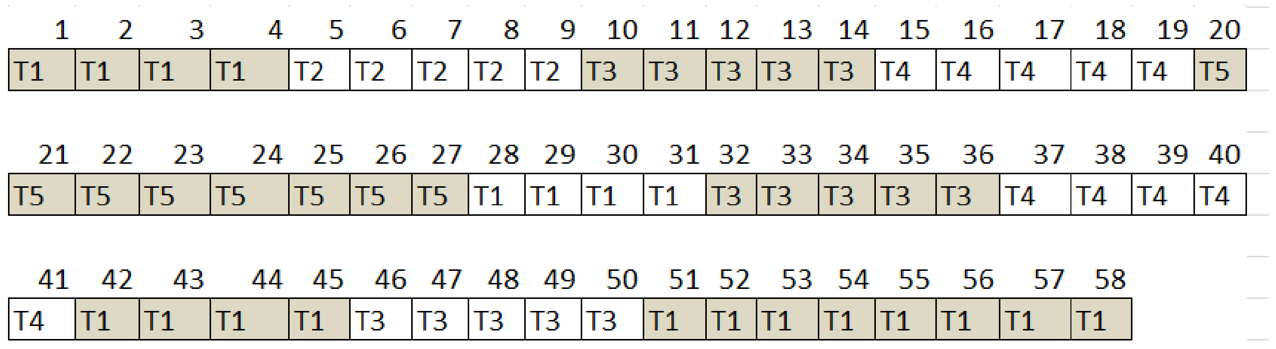


Fig. 2 Gantt Chart Intelligent Round Robin

Take a look at gantt chart above. T1 is executed with 4 quantum until 4. Then T2 is executed with 5 quantum until 9. T3 is executed with 5 quantum to 14. T4 is executed with 5 quantum to 19. Then T5 is executed with quantum 8, making all of it finish. Then system gets back to T1. T1 is executed with quantum 4 until 31. Then T2. However T2 is finish, Then system executes T3 with 5 quantum to 36. Then T4 also is executed with 5 quantum until 41. Since T5 is finishes, the system gets back to T1. T1 is executed with 4 quantum until 45. Since T2 is finished, system goes on to T3. T3 is executed with 5 quantum until 50, then it is finish. Finally, since T2, T3, T4, and T5 are finish, the rest of T1 is executed continuously.

Then we calculate the waiting time:
T1 = 0+(27-4)+(41-31)+(50-45)=38
T2 = 4
T3 = 9+(31-14)+(45-36)=35
T4 = 14+(36-19)=31
T5 = 19

So, the average waiting time is (38+4+35+31+19)/5=25.4

Next we calculate the average turn around time, as per the table below:

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 4, April 2021

DOI 10.17148/IJARCCE.2021.10413

TABLE 4 TASK BURST TIME INTELLIGENT ROUND ROBIN

Process	Arrival Time	Burst Time	Start Time	Finish Time	Turn Around Time
T1	0	20	0	58	58
T2	0	5	4	9	9
T3	0	15	9	36	36
T4	0	10	14	41	41
T5	0	8	19	27	27

So the average turn around time = 167/5 = 33.4

C. Comparison Analysis

Based on analysis above, here is the comparison table of number of context switch, average waiting time, and average turn around time:

TABLE 5. COMPARISON TABLE

Algorithms	Number of Contex Switch	Average Waiting Time	Average Turn Around Time
Simple Round Robin	16	32.6	44.2
Intelligent Round Robin	11	25.4	33.4

Please take a look at the table above. For simple round robin, the total context switching is 16, while in intelligent round robin it is 11. Reducing number of context switching will make the system more efficient. For average waiting time, in simple round robin it is 32.6 ms, while in intelligent round robin it is 25.4ms. This is a reduction of 12%. For average turn around time, in simple round robin, it is 44.2 ms, while in intelligent round robin is 33.4ms, this is 24% lower.

VI. CONCLUSION

Based on analysis above, it proves that the intelligent round robin algorithm was more efficient than the simple round robin algorithm. While in intelligent round robin, the number of context switch is minimum. It is 16 context switching compared with 11 in intelligent round robin. Also the average waiting time and average turn around time were minimum compared to the simple round robin algorithm. The average waiting time is reduced to be 12% lower. The average turn round time is lower 24% than in simple round robin. It is concluded that in intelligent round robin turn around time and waiting time are more efficient than the simple round robin algorithm.

REFERENCES

[1]. Datta, L. (2015). Efficient Round Robin Scheduling Algorithm with Dynamic Time Slice. *International Journal on Education and Management Engineering*, *2*, 10–19.
[2]. Dhruv, R. (2019). Round Robin Scheduling Algorithm Based on Dynamic Time Quantum. *International Journal of Engineering and Advanced Technology (IJEAT)*, *X*(X), 593–595.
[3]. Govind Prasad Arya, Kumar Nilay, D. P. (2018). An Improved Round Robin CPU Scheduling Algorithm Based on Priority of Process. *International Journal of Engineering & Technology*, *7*(4.5), 238–241.
[4]. J. R. Indusree, B. P. (2017). Enhanced Round Robin CPU Scheduling with Burst Time Based Time Quantum. *IOP Conf Series: Material Science and Engineering*, 1–8.
[5]. m. LaxmiJeevani, T.S.P. Madhuri, Y. S. D. (2018). Improvised Round Robin Scheduling Algorithm and comparison with Existing Round Robin CPU Scheduling Algorithm. *IOSR Journal of Computer Engineering*, *20*(3), 1–4.
[6]. Manish Kumar Mishra, F. R. (2014). An Improved Round Robin CPU Scheduling Algorithm with Varying Time Quantum. *International Journal of Computer Science, Engineering and Applications (IJCSEA)*, *4*(4), 1–8.
[7]. Neha, A. J. (2018). An Improved Round Robin CPU Scheduling Algorithm. *Iconic Research and Engineering Journal*, *1*(9), 82–86.
[8]. Sharma, A. (2015). Analysis of Adaptive Round Robin Algorithm and Proposed Round Robin Remaining Time Algorithm. *International Journal of Computer Science and Mobile Computing*, *4*(12), 139–147.
[9]. Tithi Paul, Rahat Hossain Faisal, M. S. (2019). Improved Round Robin Scheduling Algorithm with Progressive Time Quantum. *International Journal of Computer Application*, *178*(49), 30–36.
[10]. Yaashuwanth, C., & R. Ramesh. (2010). Intelligent Time Slice for Round Robin in Real Time Operating Systems. *IJRRAS*, *2*(2), 126–131.