Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 3, July 2022
DOI : https://doi.org/10.33395/sinkron.v7i3.11530
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Case Study: Improved Round Robin Algorithm

Tri Dharma Putra[1], Rakhmat Purnomo[2*]
[1,2]Universitas Bhayangkara Jakarta Raya
rakhmat.purnomo@dsn.ubharajaya.ac.id*

Submitted : July 7, 2022 | **Accepted** : July 23, 2022 | **Published** : July 26, 2022

Abstract: In this journal, discussion is given to analyse the improved round robin algorithm more thoroughly. Round robin algorithm plays a significant role to be used in embedded systems. Round robin algorithm usually applied in real-time systems. Here, three case studies are given, and also the analysis of each case study. Comparisons are given about the average turn around time and average waiting time, also number of context switching between the three case studies. Improved round robin algorithm, is a modification from the generic round robin algorithm. In improved round robin algorithm if the remaining burst time is less than the time slice that is allocated, then the currently running process is continue to be executed. Then finish the currently running process from ready queue and execute the next ready queue. Three case studies are given with three different time quantum, which are 3, 4, and 5 ms. The result of this case study analysis is that, the efficiency of the quantum 5 ms is the most effective one. There is an increase of 50% context switching from quantum 3 to quantum 5. And for average turn around time we get 13.13% reduction in efficiency. While in average waiting time we get reduction 12.08% efficiency.

Keywords: Average Turn Around Time, Average Waiting Time, Context Switching, Improved Round Robin Algorithm, Quantum

INTRODUCTION

Process scheduling is crucial in the operations of OS (operating system). OS functions as an interface between computer hardware and its user. OS help user to use the computer more easily, so that, the user does not need to know the hardware behind the systems (Purnomo & Putra, 2022) (Putra & Purnomo, 2021). All application is installed to the operating system's API. OS handles the system in a convenient manner (Paul et al., 2019). Today, in modern operating systems, the systems become more complex. System can handle multitasking program by switching between processes. In this way, the user can interact with computer. OS performs major functions like resource management, process scheduling, and memory management (Indusree & Prabadevi, 2017).

The main duty of operating systems is to manage resources of computer systems. Operating systems is known as a resource manager. Process scheduling is the most important and one fundamental idea. Scheduling defines a set of policies, rules, and mechanism that govern the order in which resources are allocated to various processes and the work is to be done (Putra, 2022) (Putra, 2020). There is methodology of process scheduling to manage many queues of processes in order to make performance optimal and the delay to be minimum in the system. There are a scheduler in operating system that implements the scheduling policies.

Round robin algorithm is used as a real-time scheduling algorithm in operating system. The solution is to use static and dynamic time slice or quantum, so that operating systems can be operated more effectively. The improved round robin algorithm is a modification of simple round robin algorithm with the idea to make the scheduling more effective. With lower average turn-around time and lower average waiting time. Also reduce the context switching. Simple round robin architecture is difficult to be implemented in real-time operating systems because of high context switch rate, longer respons time and longer waiting time (Yaashuwanth & R. Ramesh, 2010). The generic round robin scheduling algorithm is used especially for time sharing system. Each task is given a time slice or quantum (Sharma, 2015) (Putra & Fadjriya, 2021), which is a time interval.

Neha and Ankita Jiyani proposed an improved round robin algorithm. What proposed by Neha And Ankita Jiyani, is an improvement which was more simple from the researches before. The basic idea to continue executing the currently running process if the remaining burst time is below the quantum time, was proved in their experiments with this algorithm improved the efficiency. Instead of using shortest job first algorithm

*Corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 3, July 2022 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v7i3.11530 p-ISSN : 2541-044X

proposed by Manish Kumar and Rashid, and other algorithms discussed below. Neha and Ankita Jiyani's proposal proved to be more simple. Simplicity in this algorithm is a great idea from the previous researches.

LITERATURE REVIEW

In his journal, Dhruv, defined a new modification of this generic algorithm. Burst time of processes are organized in rising order. This algorithm uses dynamic time quantum which is based on the remaining shortest burst time. And time quantum of this algorithm is important. Round robin is pretty much similar with FCFS (First Come First Serve). Time quantum is important and plays an significant role in the performance of round robin algorithm. Too many context switchings will occur if time quantum of given algorithm is short (Dhruv, 2019).

In his journal, Lipika Datta, gave proposal, to modify the algorithm by arranging the quantum (time slices) of different rounds based on the remaining CPU burst of the processes that currently running and considering their waiting times until that round in respect of the other processes waiting times. From their experimental analysis, it is concluded that their algorithm produces fewer number of context switching and better average waiting time, better average turnaround time than standard algorithms (Datta, 2015).

In their journal, Govind Prasad Arya, Kumar Nilay, Devendra Prasad, gave proposal to define a new algorithm that categorizes available processes based on process with priorities, which are low priority and high priority. This algorithm proved to reduce the average waiting time of process that have high priority in all cases. And in not all but some cases of processes with low priority. On the basis of set of processes considered, he overall waiting time changes (Arya et al., 2018).

In their journal, Manish Kumar Mishra and Faizur Rashid, gave a proposal about a modification round robin CPU scheduling algorithm which enhancing performance of CPU. Their idea is by using the features of shortest job first (SJF) and round robin algorithms based on varying time quantum. This algorithm, was proven better than simple round robin, based on their experiments (Kumar Mishra & Rashid, 2014).

In their journal, M. Laxmi Jeevani, T.S.P. Madhuri, and Y. Sarada Devi, gave proposal to use an algorithm by using a few steps. a) By allocating process to CPU which has arrived first, giving an initial time quantum (k units). b) The scheduler will check the queue for processes which have arrived by the time of completion of first process execution with initial time quantum, after the completion of first process initial execution. Afterwards, among the multiple processes that have arrived in the queue by using the k units of time. The process will be allocated to the CPU is with the minimum burst time. If the first process has completed its execution, it is removed from the queue. And again, a chance is provided to it. And at last, for the complete execution of all processes, systems have to repeat the same (m. LaxmiJeevani, T.S.P. Madhuri, 2018).

In their journal, Neha and Ankita Jiyani, gave proposal about an improved round robin algorithm (Neha, 2018). The first process in the ready queue is executed until the allocated time slice, afterwards, the algorithm has to check the remaining burst time of the running process. If the remaining burst time is less than the allocated time slice, then the running process is executed again untill it is finished. Afterwards, the next process in the ready queue will be executed. Only if the remaining burst time of the currently running process is bigger than the allocated time slice.

METHODS

Five chapters are presented here. Introduction is given in the first chapter. In this chapter, discussions about the idea about round robin algorithm. In the second chapter, literature surveys did by researcher before are given. Then, in the third chapter. The discussion is about methodology. In this case the improved round robin algorithm which is the modification of generic round robin algorithm, that were given by some researchers. Also in this chapter discussion is about conceptualization behind the improved round robin algorithm. Then, the fourth chapter is about result. Here three case studies are discussed, which uses different quantum, namely 3, 4, and 5 ms. The fifth chapter is chapter is about discussion of the result analysis from case studies in the fourth chapter. And again, in this chapter, comparisons are given with table, about average turn around time, average waiting time, and number of context switching. The last chapter is conclusion and future works.

The basic idea about round robin algorithm is using context switching or time slice, or sometimes it is called time quantum. Each process is assigned a time interval. Round robin is a preemptive algorithm. If the time slice finishes, and the processes are still executing, then, afterwards, the algorithm will force the next queue to preempt the process and keep it at the end of the ready queue. This is the basic concept about improved round robin algorithm. It is cricital in this algorithm to choose of how big the time slice is, because the size of time slice gives great effect on the performance of this algorithm.

Neha and Ankita Jiyani (Neha, 2018), proposed an improved round robin algorithm with basic idea as follows:

*Corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 3, July 2022
DOI : https://doi.org/10.33395/sinkron.v7i3.11530
e-ISSN : 2541-2019
p-ISSN : 2541-044X

- Execute the first process in the ready queue and execute it until the allocated time quantum, afterwards, check the remaining burst time of the running process.
- If the remaining burst time of the running process is below than the allocated time quantum then the execution continues.
- If the remaining burst time of the running process is more than the allocated time quantum, then the next process in ready queue is executed.
- Then remove the running process from the ready queue and afterwards, put it at the tail of the ready queue.

We can discuss the idea of this algorithm by example. Let say, there are two processes T1 and T2. T1 with 8 ms burst time and T2 with 7 ms burst time. And time quantum is 5 ms. Let say, the first quantum is executed, the remaining of T1 is 3 ms. Since 5 ms is executed. Then because the remaining, the 3 ms is below then the quantum (which is 5 ms), then 3 ms continues to be executed, this will not be preemptived because the remaining of T1 is 3 which is below the quantum 5 ms. Then the process of T1 continues until finish. Afterwards, T2 is executed which makes the remaining of T2 is 2 ms, since 7-5 = 2. This process will not be preemptived because the remaining of the process is 2 ms for T2. Which is below the quantum, which is 5 ms. So T2 is executed until finish, because the remaining time is less than the quantum. As we can see in the gantt chart of Figure 1. below:

T1 = 8 ms, T2 = 7 ms, Quantum = 5 ms

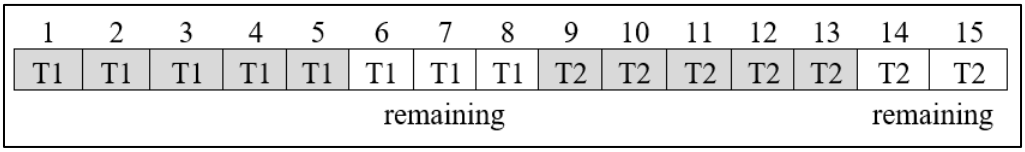

Figure 1. Gantt Chart of basic concept improved round robin with quantum 5 ms.

RESULTS

Below is the analysis of the case studies. We discuss three case studies with different quantum, but with the same processes and burst time. The first with quantum =3, the second with quantum = 4, the last one is with quantum = 5. Then we discuss the comparison analysis of the three case studies for evaluation, which one is more efficient than the others.

Table 1. Processes with Arrival Time and Burst Time

Task ID	Arrival Time	Burst Time
T1	0	10
T2	0	14
T3	0	9
T4	0	12

Please take a look at the Table 1. The case study is with 4 tasks. T1 until T4. And the quantums are 3, 4 and 5 ms. The arrival times are zero for all. Total burst time is 45 in each of these case studies.

5.1 Case Study 1

In this case study 1, the quantum is 3. The total processes are 45 ms. Please take a look on Figure 2. below:

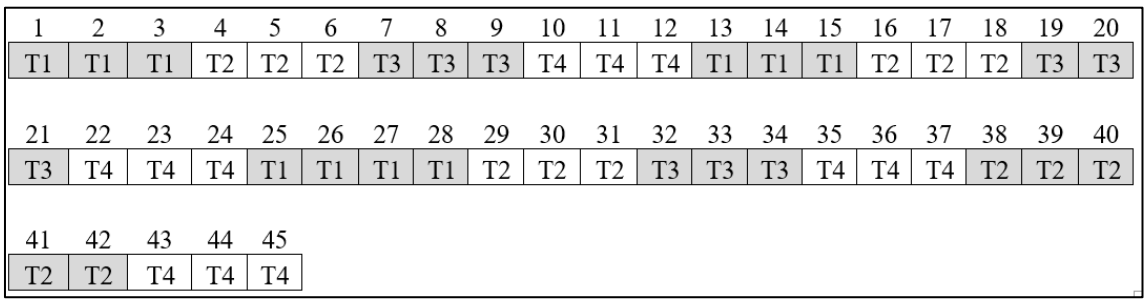

Figure 2. Gantt Chart with quantum 3.

*Corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 3, July 2022
DOI : https://doi.org/10.33395/sinkron.v7i3.11530
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Since the quantum is 3, T1 is executed for 3 ms. The remaining of T1 is 7 ms. Then afterwards, T2 is executed also 3 ms. Then T2 remains 11 ms. Then T3 is executed for 3 ms. This continues to T4, for 3 ms. At 13 ms, T1 is executed for another 3 ms, afterwards T2, T3 and T4. Now, T1 remains 4 ms, since 6 ms has been executed. Then T1 is executed for another 3 ms, since the remains of T1 is 1, which is below the quantum 3, then T1 continues to be executed by 4 ms. T1 is finished here. Then T2, T3 and T4 are executed for another 3 ms each. The remains of T2 is 5 ms. At 37 ms, T2 is executed for another 3 ms, but the remains of T2 is 2 ms. Then T2 continues to be executed from 37 ms until 42 ms. The last one is T4. T4 is left 3 ms, Then 3 ms is executed for T4 until finish at 45.

For calculation of waiting time, we get as follow:

T1 = 0 + (12 – 3) + (24 – 15) = 18
T2 = 3 + (15 – 6) +(28 – 18) + (37 – 31) = 28
T3 = 6 + (18 – 9) + (31 – 21) = 25
T4 = 9 + (21 – 12) + (34 – 24) = 28

It is concluded that for this improved round robin algorithm, the average waiting time is 99/4=24.75 ms.

Then to calculate the turn around time. The first step is, to populate the table below for turn around time analysis, as per Table 2 below:

Table 2. Turn Around Time of Quantum 3

Process	Arrival Time	Burst Time	Start Time	Finish Time	Turn Around Time
T1	0	10	0	28	28
T2	0	14	3	42	42
T3	0	9	6	34	34
T4	0	12	9	45	45

It is concluded that the average turn around time is =(28+42+34+45)/4=37.25 ms

5.2 Case Study 2

In this case study 2, the quantum is 4. The total processes are the same as the case study 1, which is 45 ms. Please take a look on Figure 3. Below:

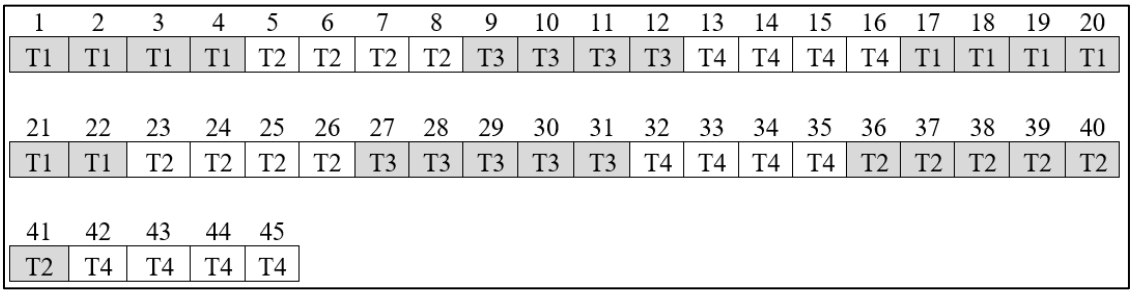

Figure 3. Gantt Chart with quantum 4.

Since the quantum is 4, Then T1, T2, T3, and T4 are executed for 4 ms each. Until 16 ms. T1 which is 10 ms, has been executed for 4 ms, which is left 6 ms. Then at 16 ms, T1 is executed for another 4 ms. But since the remaining of T1 is 2 ms, which is below the quantum 4, Then systems continue to executed T1 for another 2 ms, which is 6 ms. This is from 16 ms until time 22 ms. Afterwards T2 is executed for another 4 ms, until 26 ms. Then at 26 ms, T3 is executed for another 4 ms. Since T3 is left 5 ms, then T3 is executed from 26 until 31 ms, for 5 ms. Then T4 is executed for another 4 ms until 35 ms. Then T2 is executed afterwards. The remaining of T2 is 6 ms. Then this process, T2 continues to be executed for 4 ms and 2 ms afterwards, to be 6 ms, since T2 is left 2 ms. Then the last process is T4. T4 is left 4 ms. Then, T4 is executed for the last 4 ms until 45.

For calculation of waiting time, we get as follow:

*Corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 3, July 2022
DOI : https://doi.org/10.33395/sinkron.v7i3.11530
e-ISSN : 2541-2019
p-ISSN : 2541-044X

T1 = 0 + (16 – 4) = 12
T2 = 4 + (22 – 8) + (35 – 26) = 27
T3 = 8 + (26 – 12) = 22
T4 = 12 + (31 – 16) + (41 – 35) = 33

It is concluded that for this improved round robin algorithm, the average waiting time is 94/4=23.50 ms.

Then to calculate the turn around time. The first step is, we populate the table below for turn around time analysis, as per Table 3. below:

Table 3. Turn Around Time with Quantum 4

Process	Arrival Time	Burst Time	Start Time	Finish Time	Turn Around Time
T1	0	10	0	22	22
T2	0	14	4	41	41
T3	0	9	8	31	31
T4	0	12	12	45	45

It is concluded that the average turn around time is = (22 + 41 + 31 + 45) / 4 = 34.75 ms

5.3 Case Study 3

In this case study 3, the quantum is 5. The total processes are the same as the case study 1 and case study 2, which is 45 ms. Please take a look on Figure 4. Below:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
T1	T1	T1	T1	T1	T2	T2	T2	T2	T2	T3	T3	T3	T3	T3	T3	T3	T3	T3	T4
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
T4	T4	T4	T4	T1	T1	T1	T1	T1	T2	T2	T2	T2	T2	T2	T2	T2	T2	T4	T4
41	42	43	44	45															
T4	T4	T4	T4	T4															

Figure 4. Gantt Chart with quantum 5.

T1 and T2 are executed for 5 ms second in a row. Namely, 5 ms for T1 and 5 ms for T2 until 10. Then, afterwards T3 is executed. The burst time of T3 is 9 ms. The first 5 ms is executed, Then T3 is left 4 ms. Then afterwards, systems continue to execute T3 for another 4 ms until 19. So that T3 is executed 9 ms in a row. Then, afterwards, T4 is executed for 5 ms. Then, T1 is executed for another 5 ms. At this point, 29 ms, T1 is finish. Then, T2 is executed. Since T2 is left 9 ms, then the systems continue to execute the 5 ms, then directly the 4 ms, since 4 ms is below the quantum, which is 5 ms. T2 is finish at 38. Then the last one is T4. T4 is left 7. Then the 5 ms of T4 is executed. The remaining of T4 is 2 ms. Then systems continue to execute T4, the last 2 ms.

For calculation of waiting time, we get as follow:

T1 = 0 + (24 – 5) = 19
T2 = 5 + (29 – 10) = 24
T3 = 10
T4 = 19 + (38 – 24) = 33

It is concluded that for this improved round robin algorithm, the average waiting time is 86/4=21.50 ms.

Then to calculate the turn around time. The first step is, to populate the Table 4. below for turn around time analysis:

*Corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 3, July 2022
DOI : https://doi.org/10.33395/sinkron.v7i3.11530
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Table 4. Turn Around Time with Quantum 5

Process	Arrival Time	Burst Time	Start Time	Finish Time	Turn Around Time
T1	0	10	0	29	29
T2	0	14	5	38	38
T3	0	9	10	19	19
T4	0	12	19	45	45

It is concluded that the average turn around time is =(29+38+19+45)/4=32.75 ms

DISCUSSION

Referring to the analysis of case study above, the table below is presented for comparison of number of context switching, average waiting time, and average turn around time:

Table 5. Comparison Table

Case Study	Number of Context Switching	Average Waiting Time	Average Turn Around Time
quantum 3	14	24.75	37.25
quantum 4	10	23.50	34.75
quantum 5	7	21.50	32.75

Please take a look at the Table 5. above. For quantum 3, the number of context switching is 14, where in quantum 4 the number of context switching is 10 and in quantum 5, it is 7. In quantum 5, the context switching reduce up to 50% , which is 7, compared with the quantum 3. The average waiting time for quantum 3 is 24.75 ms, where in quantum 4, it is 23.50 and in quantum 5, it is 21.50 ms. For average turn around time, in quantum 3 is 37.25 and in quantum 4, it is 34.75, and lastly for quantum 5, it is 32.57 ms. The lower the average turn around time, the more efficient the system. We can say, in this context, that the bigger the quantum the more efficient the systems. For average waiting time from quantum 3 and quantum 5, we get reduction 13.13% of average waiting time. For turn around time for quantum 3 and quantum 5, we get reduction 12.08%.

CONCLUSION

Based on analysis above, we can conclude that improved round robin algorithm was more efficient if the quantum is bigger. As we can take a look at the comparison analysis above, that there is an increase of 50% context switching from quantum 3 to quantum 5. And for average turn around time we get 13.13% reduction in efficiency. While in average waiting time we get reduction 12.08% efficiency. For future works, the improved round robin algorithm can be modified and combine with priority scheduling like preemptive or non preemptive shortest job first algorithm. Other comparisons also can be done, like comparison between intelligent round robin and improved round robin.

REFERENCES

Arya, G. P., Nilay, K., & Prasad, D. (2018). An improved round robin CPU scheduling algorithm based on priority of process. *International Journal of Engineering and Technology(UAE)*, *7*(4), 238–241. https://doi.org/10.14419/ijet.v7i4.5.20077

Datta, L. (2015). Efficient Round Robin Scheduling Algorithm with Dynamic Time Slice. *International Journal of Education and Management Engineering*, *5*(2), 10–19. https://doi.org/10.5815/ijeme.2015.02.02

Dhruv, R. (2019). Round robin scheduling algorithm based on dynamic time quantum. *International Journal of Engineering and Advanced Technology*, *8*(6), 593–595. https://doi.org/10.35940/ijeat.F8070.088619

Indusree, J. R., & Prabadevi, B. (2017). Enhanced round robin CPU scheduling with burst time based time quantum. *IOP Conference Series: Materials Science and Engineering*, *263*(4). https://doi.org/10.1088/1757-899X/263/4/042038

Kumar Mishra, M., & Rashid, F. (2014). An Improved Round Robin CPU Scheduling Algorithm with Varying Time Quantum. *International Journal of Computer Science, Engineering and Applications*, *4*(4), 1–8. https://doi.org/10.5121/ijcsea.2014.4401

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*

*Corresponding author

Engineering, *20*(3), 1–4. https://doi.org/10.9790/0661-2003010104

Neha, A. J. (2018). An Improved Round Robin CPU Scheduling Algorithm. *Iconic Research and Engineering Journal*, *1*(9), 82–86.

Paul, T., Hossain, R., & Samsuddoha, M. (2019). Improved Round Robin Scheduling Algorithm with Progressive Time Quantum. *International Journal of Computer Applications*, *178*(49), 30–36. https://doi.org/10.5120/ijca2019919419

Purnomo, R., & Putra, T. D. (2022). *Simulation of Preemptive Shortest Job First Algorithm*. *11*(5), 1–11. https://doi.org/10.17148/IJARCCE.2022.11501

Putra, T. D. (2020). Analysis of Preemptive Shortest Job First (SJF) Algorithm in CPU Scheduling. *IJARCCE*, *9*(4), 41–45. https://doi.org/10.17148/ijarcce.2020.9408

Putra, T. D. (2022). Analysis of Priority Preemptive Scheduling Algorithm: Case Study. *Ijarcce*, *11*(1), 27–30. https://doi.org/10.17148/ijarcce.2022.11105

Putra, T. D., & Fadjriya, A. (2021). Comparison Between Simple Round Robin and Intelligent Round Robin Algorithms in CPU Scheduling. *IJARCCE*, *10*(4), 86–90. https://doi.org/10.17148/ijarcce.2021.10413

Putra, T. D., & Purnomo, R. (2021). Analisis Algoritma Round Robin pada Penjadwalan CPU. *Jurnal Ilmiah Teknologi Informasi Asia*, *15*(2), 85. https://doi.org/10.32815/jitika.v15i2.481

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.

Yaashuwanth, C., & R. Ramesh. (2010). Intelligent Time Slice for Round Robin in Real Time Operating Systems. *IJRRAS*, *2*(2), 126–131.

*Corresponding author