



Improved Scheduling Algorithm For Real Time, Time Sharing And Batch Operating Systems.

¹Amit Kumar Tyagi, ²Siddharth Bandam,

¹Assistant Professor, ²Student

¹ Computer Science and Engineering, School of Computer Science and Engineering,

¹ Vellore Institute of Technology Chennai (UGC), Chennai, 600127, Tamil Nadu, India

²Computer Science and Engineering, School of Computer Science and Engineering,

²Vellore Institute of Technology Chennai (UGC), Chennai, 600127, Tamil Nadu, India

Abstract: In today's world, there are multiple kinds of users prevalent. Different users require different systems with varying demands. There are three major kinds of operating systems present namely, Real-time, Time-sharing and Batch Operating Systems. The main intention of writing this paper is to introduce an algorithm to assess the needs and use of appropriate scheduling algorithm for maximum efficiency. The CPU scheduling algorithms present today are not applicable in all kinds of systems. Different algorithms are required by different systems. For example, real time operating systems can't use normal round-robin scheduling due to their large waiting time, response time and turnaround time, high context switch rates and less throughput. The primary aim is to come up with a method that is adaptive to all kinds of systems and provide them with CPU scheduling algorithms which enhances the CPU performance in the system. A comparative analysis of our algorithm for real-time systems with the algorithms for other systems like priority and round robin will be presented on the basis of average turnaround time, average waiting time, varying time quantum and number of context switches.

Index Terms – CPU Scheduling, Operating systems, Real time OS, Time Sharing OS, Batch Operating OS

I. INTRODUCTION

The method by which processes are given access to system resources like processor cycles and communications bandwidth is known as scheduling. To execute multitasking and multiplexing scheduling algorithms are required by the computer systems. Fundamentally, scheduling determines which process run, in the presence of multiple run-able processes. Resource utilization and other performance parameters are affected by the method of CPU scheduling, Some assumptions are considered in CPU scheduling such as:

1. Job pool has run able processes that wait for the CPU.
2. Each and every process is independent. They compete for resources.
3. Distributing the limited resources of the CPU among different processes optimize certain performance criteria is the primary aim of CPU scheduling.

The component of the kernel which selects processes to run next is the scheduler. There are three types of schedulers as follows:

LONG TERM	MID TERM	SHORT TERM
The long term scheduler or job scheduler selects processes from the job pool and loads them into memory for execution.	The medium term scheduler removes processes from memory and reduces the degree of multiprogramming results in the scheme of swapping.	The short term scheduler, or CPU scheduler selects from among the processes that are ready to execute, and allocates CPU to one of them.

II. ALGORITHM AND WORKFLOW

1. Read the Kind of System for which scheduling is required.
2. For batch systems:
 - a. Sort processes according to priority.
 - b. Execute the processes in the order of priority.

This will then follow FCFS algorithm.

3. For Time- sharing systems:
 - a. While queue not empty
 - b. For time from 0 to allotted time quantum, execute process.
 - c. Move to next process
4. For Real-time systems, Sort the processes according to priority as well as shortness and assign a new priority to each process which is the sum of the original priority and shortness rank.
 - a. Calculate priority component. If there are n processes, for a process i in n $PC_i=0$ if its new priority is $> 2n/3$ (Not Important) $PC_i=1$ if its new priority is $> n/3$ (Moderately Important) $PC_i=2$ if its new priority is ≥ 1 (Important)
 - b. Calculate shortness component. If there are n processes, for a process i in n $SC_i=0$ if the (Burst Time) $_i >$ (Burst Time) $_{i-1}$ (Longer) $SC_i=1$ if the (Burst Time) $_i \leq$ (Burst Time) $_{i-1}$ (Shorter)
 - c. Calculate the intelligent time slice (ITS) for each process as the sum of the initial time slice, burst time, priority component and shortness component.
 - d. Repeat Step 'E' until all processes are completed.
 - e. If the Round number(j) is 1, calculate time quantum as $TQ_{j,i}=ITS_i$ if $SC_i=1$ $TQ_{j,i}=ITS_i/2$ if $SC_i=0$.

If the Round number(j) is not 1, calculate time quantum as $TQ_{j,i}=TQ_{j-1,i} * 2$ if $SC_i=1$ $TQ_{j,i}=TQ_{j-1,i} * 1.5$ if $SC_i=0$. Calculate average waiting time and average turnaround time.

The main goal is to come up with a program that allows the user to pick the kind of system for which scheduling is required. Based on the choice the appropriate algorithm is picked. Improved algorithm for real time systems has been implemented which works on a dynamic time quantum, that is it changes after every round of execution. This will reduce the average waiting time, average turnaround time and reduce the number of context switches that take place which in turn enhances the working of the computer system.



Figure 1. The flowchart for the Workflow Diagram.

III. BATCH OPERATING SYSTEMS

In a batch OS, prepares his work on an offline device instead of interacting with the computer directly. This is then submitted to the computer operator. Jobs with around the same requirements are batched together and executed as a group in order to speed up processing.

Once the programmers are done they give their programs to the operator who sorts the programs into batches according to similar requirements. We propose implementation of PRIORITY SCHEDULING for optimal use of processor time and enhance efficiency of the system.

One of the most common algorithms for scheduling in batch systems is priority scheduling. A priority is assigned to each process and the one with the highest priority is executed first and so on. If more than one process has the same priority then they are executed on first come first served basis. The priority is decided based on a variety of factors such as memory, time or any other resource requirement. Response and waiting time in this algorithm depend on the priority assigned to each process. This means that higher priority processes will have smaller waiting and response times. Higher priority can be given to the earlier deadline processes to eradicate the problem of deadlines not being met.

Disadvantage of this method is that of starvation of lower priority processes. This is possible when large number of higher priority processes continuously arrive.

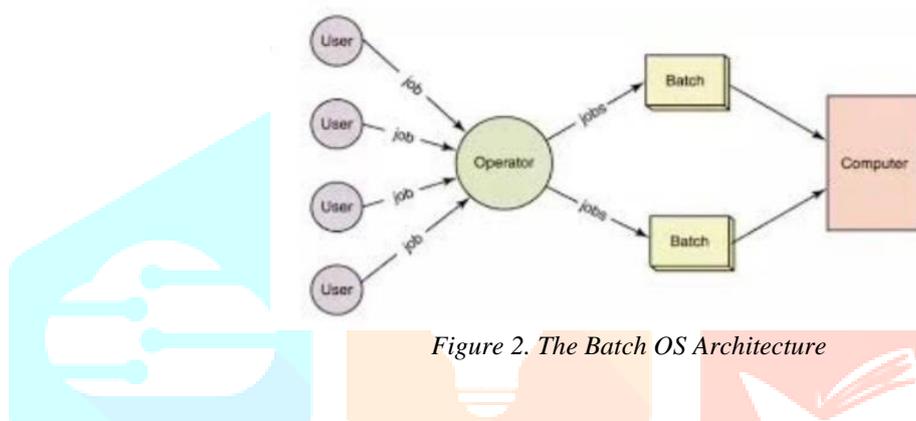


Figure 2. The Batch OS Architecture

IV. TIME SHARING OPERATING SYSTEMS

A technique which helps many people who are present at various terminals use a particular computer simultaneously is known as time-sharing. Hence timesharing is termed as the processor's time which is shared among multiple users. Multiple jobs are completed by the CPU when it switches between them.

These switches occur frequently enabling the user to receive an immediate response. Earlier computer systems that were designed as batch systems have now been modified into time-sharing systems. We propose implementation of ROUND-ROBIN SCHEDULING for optimal use of processor time and enhance efficiency of the system.

The preemptive scheduling where every process is executed in a cyclic way that is, a time slice known as time quantum is given to every process is known as round robin scheduling. Its designed mainly for time- shared systems. A circular queue which has a head and a tail contain the process that are ready to be run. All the processes are arranged in the order of first come first serve. The CPU scheduler allocates the CPU one by one for the time interval equal to one quantum. Newly arrived processes are added to the tail of the queue.

1. Disadvantages of this type of scheduling is that there will be larger waiting and response time with high rates of context switching. This will intern delay the results.

2. Another disadvantage is that of low throughput. This will lead to severe degradation of system performance in soft real time systems. Context switch and throughput are inversely proportional to each other.

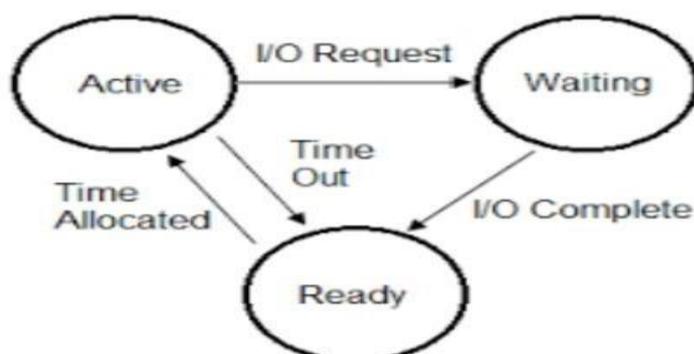


Figure 3. Time-Sharing OS Architecture

A. Real-Time Operating Systems

There are two main kinds of real-time systems:

-- Hard Real-Time System: They require fixed deadlines that need to be met otherwise errors may occur.

-- Soft Real-Time System: Here missing an occasional deadline is tolerable but undesirable. The system's performance is degraded but not destroyed when a failure takes place on meeting response time constraints. CPU scheduling plays an important role in real time operating systems since they have time constraint on computations. These systems are mission-critical, that is real-time tasks should be scheduled to be completed on or before their deadlines. It is required that each task be invoked after the ready time and be completed before the assigned deadline. Simple round robin algorithm is not applicable for Soft real time systems due to longer waiting, increased number of context switches and response times which in turn leads to lowered throughput. Priority scheduling is not applicable either due to the problem of starvation that is, low priority processes will not get processor time and will starve. We propose implementation of round-robin and priority scheduling with a dynamic time quantum to overcome this problem.

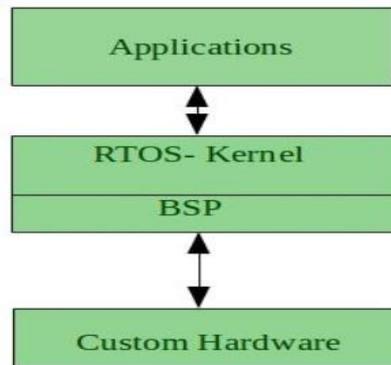


Figure 4. Real Time OS Architecture

V. COMPARISONS FOR HIGH BURST AND LOW BURST TIMES

Process No	Algorithm			Priority		Round Robin	
	Burst Time	Waiting Time	Turn Around Time	Waiting Time	Turn Around Time	Waiting Time	Turn Around Time
1	5	8	13	0	5	6	11
2	2	6	8	5	7	2	4
3	6	0	6	7	13	7	13
Average:		4.6667	9	4	8.333	5	9.333

Table 1. Low Burst Times

Process No:	Algorithm			Priority		Round Robin	
	Burst Time	Waiting Time	Turn Around Time	Waiting Time	Turn Around Time	Waiting Time	Turn Around Time
1	500	0	500	1540	2040	996	1496
2	690	927	1617	850	1540	1188	1878
3	850	1190	2040	0	850	1190	2040
Average:		705.667	1385.67	796.67	1476.67	1124.667	1804.667

Table 1. High Burst Times

VI. LITERATURE SURVEY

AUTHOR	TITLE	YEAR	JOURNAL	METHOD	PURPOSE	ADVANTAGES	DISADVANTAGES
Neetu Goeland R.B. Garg	A Comparative Study of CPU Scheduling Algorithms	2013	International Journal of Graphics & Image Processing Vol 2 issue 4 November 2012	CPU Scheduler	Presents a state diagram that depicts the comparative study of various scheduling algorithms for a single CPU and shows which algorithm is best for the particular situation	Makes much easier to understand what is going on inside the system and why a different set of processes is a candidate for the allocation of the CPU at different time	The treatment of shortest process in SJF scheduling tends to result in increased waiting time for long processes. And the long process will never get served, though it produces minimum average waiting time and average turnaround time.
Shweta Jain and Saurabh Jain	A Research Survey and Analysis for CPU Scheduling Algorithms using Probability-Based Study	2015	International Journal of Engineering and Management Research	A probability based Markov chain analysis	CPU Scheduling deals with the problem of deciding which of the processes in the ready queue is to be allocated the CPU. Hence, the performance of system depends on which scheduling algorithm used. In this paper, are discussed, the various approaches that can be used for this purpose and elaborate the research analysis in this field.	A probability based Markov chain analysis done in order to determine the performance of these algorithms.	Research is restricted to c++
Dr. Pardeep Kumar, Dr. Sanjay Tyagi and Monika	Survey on Various Scheduling Algos in Cloud Computing	2016	International Journal of Scientific & Engineering Research, Volume 7, Issue 12	Batch mode heuristic scheduling	In this paper, various scheduling algorithms have been discussed with respect to their various constraints such as time, cost, energy, SLA and so on.	It was identified that the issues of reliability and availability was not given due weightage. These are very important aspects of scheduling.	Performance Of resources was not taken in account
Mingyue Jiang and Tsong Yueh Chen	Testing Central Processing Unit scheduling algorithms using Metamorphic Testing	2013	Conference: Software Engineering and Service Science (ICSESS), 2013 4th IEEE International Conference	Metamorphic Testing Technique	In this paper, they used, MT to test the Highest Response Ratio Next (HRRN) scheduling algorithm.	experiments are performed based on mutants, and the experimental results show that MT is an ineffective strategy to test CPU scheduler.	Only some real life faults in one open source simulator are detected by MT.

D. Raghavaraju	Task Scheduling Algorithms in Cloud - A Survey	2016	International Journal of Distributed and Cloud Computing	Kernel Canonical Correlation Analysis (KCCA)	The primary objective is to overcome heterogeneous processing assets, minimize general execution,	The assets are shrewdly picked in light of their accessibility and limit.	Principle difficulties are confronted in scheduling of assignments to the best accessible assets.
Imran Qureshi	CPU Scheduling Algorithms: A Survey	2014	Advanced Networking and Applications Volume: 05, Issue: 04	CPU Scheduler	In this paper the review of different scheduling algorithms are performed with different parameters, such as running time, burst time and waiting times etc.	Different scheduling algorithms are reviewed in this paper according to their CPU overhead, throughput, turnaround time and response time.	The performance of mentioned algorithms is not mentioned or compared properly
Lavanya Dhanesh	Simulation And Comparison of Various Scheduling Algorithm For Improving The Interrupt Latency of Real-Time Kernel	2014	Journal of Computer Science and Applications	Pre-emptive scheduling	The main objective of the research is to improve the performance of the Realtime Interrupt Latency using Pre-emptive task Scheduling Algorithm. Interrupt Latency provides an important metric in increasing the performance of the Real Time Kernel	Low latency	Additional Blocking time
Abdulrazzaq Abdulrahim	A New Improved Round Robin (NIRR) CPU Scheduling Algorithm	2014	International Journal of Computer Applications	Round Robin Scheduling	The choice of the time quantum is critical as it affects the algorithm's performance. This paper proposes a new algorithm that further improved on the Improved Round Robin CPU (IRR) scheduling algorithm by Manish and AbdulKadir	minimal average waiting time (AWT), average turnaround time (ATAT), and number of context switches (NCS)	A bit generic
Dr. Rohit Roshan	LEAST-MEAN DIFFERENCE ROUND ROBIN (LMDRR) CPU SCHEDULING ALGORITHM	2016	Journal of Theoretical and Applied Information Technology	Round Robin Scheduling	This article presents a variant of Round Robin Algorithm called Least Mean-Difference Round Robin (LMDRR) Algorithm. First, it calculates the mean of all processes burst times. Then it obtains the	Dynamic time quantum	A bit complex

					difference of each process burst time with the calculated mean		
--	--	--	--	--	--	--	--

VIII: CONCLUSION

The proposed algorithm has less Avg. waiting time and Avg. turnaround time as compared to Round robin and Priority Algorithm. We see that due to the requirements of batch system, priority scheduling is the most suitable option. For time-sharing operating system, round robin operating system provides optimal efficiency. From the above comparisons, we observe that our new proposed algorithm is performing better than the Round robin, Priority algorithm proposed in terms of average waiting time and average turnaround time thereby reducing the overhead, saving of memory spaces and solving the problem of starvation which was caused by Priority Algorithm. Therefore, the proposed algorithm will meet the demands of a SOFT REAL TIME SYSTEM.

REFERENCES

1. Rakesh kumar yadav, Abhishek K Mishra, Navin Prakash, Himanshu Sharma," An Improved Round Robin Scheduling Algorithm for CPU Scheduling", (IJCSSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 1064-1066, 2010.
2. Ajit Singh, Priyanka Goyal, Sahil Batra," An Optimised Round Robin Scheduling Algorithm for CPU Scheduling", (IJCSSE) International Journal on Computer Science and Engineering Vol. 02, No. 07, 2383-2385, 2010.
3. William Stallings, Operating Systems Internal and Design Principles, 5th Edition , 2006. *6+ Saroj Hiranwal, K. C. Roy," Adaptive Round Robin Scheduling using shortest burst approach based on smart time slice", International Journal of Computer Science and Communication Vol. 2, No. 2, pp. 319-323, July December 2011.
4. Abbas Noon, Ali Kalakech, Seifedine Kadry, "A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average", IJCSII International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1, May 2011.
5. Barbara Korousic –Seljak (1994) "Task scheduling policies for real-time systems" Journal on MICROPROCESSOR AND MICROSYSTEMS, VOL 18, NO. 9, pg 501-512.
6. Enrico Bini and Giorgio C. Buttazzo (2004) "Schedulability Analysis of Periodic Fixed Priority Systems" journal on IEEE TRANSACTIONS ON COMPUTERS VOL 53 NO.11, pg 1462- 1473.
7. Houssine Chetto and Maryline Chetto (1989) " Some Results of Earliest Deadline Scheduling
8. Blazewicz, J., Ecker, K.H., Pesch, E., Schmidt, G., and Weglarz, J.: Scheduling Computer and Manufacturing Processes, Berlin, Springer, (2001).
9. Swin, B. R., Tayli, M., and Benmaiza, M.: Prospects for Predictable Dynamic Scheduling in RTDOS, Journal King Saud University, Computer & Information Science, Vol. 9, pp. 57-93, (1997).
10. Krithivasan Ramamritham : Scheduling Algorithms and Operating Systems Support for Real-Time System - https://www.researchgate.net/publication/2984750_Scheduling_Algorithms_and_Operating_Systems_Support_for_Real-Time_System
11. Arvindhan MU, Reetu Singh, Kajol Kathuria : "An Efficient Technique for Scheduling Algorithm In Real Time Environment For Optimizing the Operating System" - https://www.researchgate.net/publication/329163839_An_Efficient_Technique_for_Scheduling_Algorithm_In_Real_Time_Environment_For_Optimizing_the_Operating_System
12. E. Douglas Jensen, C. Douglass Locke, Hideyuki Tokuda: "A Time-Driven Scheduling Model for Real-Time Operating Systems" - https://d1wqtxts1xzle7.cloudfront.net/49183273/A_Time-Driven_Scheduling_Model_for_Real-20160928-23538-7vvp7v.pdf?1475076684=&response-content-disposition=inline%3B+filename%3DA_Time_Driven_Scheduling_Model_for_Real.pdf&Expires=1622285637&Signature=U-9-fr21433nimuL15HiVbsUJ9oS6gz30i2p4LK6Ix45ZzyzA3rgnI~1sBNZR6hvIyzqeto6DROZQ8BE0R5sn8gvyEwsU0JJx2oSVsd9okNUQXBszss1uW6vajfISmTkHrxZ5JEOIhfpFd~J1IoQjODVwenXjrx9wBVLAnMQq93rmzhZvwUmT2nDxdlrYdOey8BGGITNhb5liwJkLeGOMFyGq61uVUz~XTj8CCSC2onl0Q1ZfQe2A9Vm5iK36erfY-mhcuc1RISCFdfzNxxgFICYEDXaOphDYWYq15blBfLZayM6E~0lGpgi-nCkNQOa9UsIwh6iaGWNA9FXDVFUg_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA
13. Stoica, H. Abdel-Wahab, K. Jeffay, S. K. Baruah, J. E. Gehrke and C. G. Plaxton, "A proportional share resource allocation

algorithm for real-time, time-shared systems," *17th IEEE Real-Time Systems Symposium*, 1996, pp. 288-299, doi: 10.1109/REAL.1996.563725.

14. Shwetha Jain, Dr. Saurabh Jain: "A Review Study on the CPU Scheduling Algorithms" - <https://www.ijarcce.com/upload/2016/august-16/IJARCCE%205.pdf>
15. Alaa Fiad , Zoulikha Mekkakia Maaza, Hayat Bendoukha: " Improved Version of Round Robin Scheduling Algorithm Based on Analytic Model" - <file:///C:/Users/siddh/Downloads/125943203.pdf>
16. S. S. Dash, N. Bakhara, P. Agrawala, P.P. Behera: " A New Proposed Dynamic Quantum for Priority Based Round Robin Scheduling Algorithm" - <https://www.irjet.net/archives/V5/i2/IRJET-V5I2289.pdf>
17. Himanshi Arora, Deepanshu Arora, Bagish Goel, Parita Jain: "An Improved CPU Scheduling Algorithm" - <https://www.ijais.org/archives/volume6/number6/579-1057>

