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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

Comparison of Round Robin Based CPU Scheduling Algorithms

Charu Rani[1], Mrs. Manju Bala[2]

Student, Cse, Jan nayak Choudhary Devi lal Memorial College Of Engineering, Sirsa, India [1]

Asst. Professor, Cse, Jan nayak Choudhary Devi lal Memorial College Of Engineering, Sirsa, India[2]

Abstract: The main objective of this paper is to implement the real time scheduling algorithms and to minimize the average waiting time so that given set of tasks may be completed in a minimal time with an efficient output. We will also discuss the advantages and disadvantages of the same. Task within the real time system are designed to accomplish certain service(s) upon execution, and thus, each task has a particular significance to overall functionality of the system. Scheduling algorithms in non real time system not considering any type of dead line but in real time system deadline is main criteria for scheduling the task.

Keywords: Real Time System, CPU Scheduling, First Come First Serve (FCFS), Shortest Job First (SJF), Shortest Remaining Time First (SRTF), Round Robin (RR), Round Robin with SRTF (IRR), Round Robin with Priority (RP).

I. INTRODUCTION

A. Real time system

A real-time System (RTS) is an operating system that supports [1] and guarantees timely responses to external and internal events of real-time systems. According to Silberchatz, Galvin and Gagne[3]; The aim of operating system to allow a number of processes concurrently in order to maximize the CPU utilization. The most important attribute of a task in a real-time system is its timing constraints. Such timing constraints must be expressed precisely. A deadline is the most widely used form of a timing constraint. It offers a dual view of the usefulness of a task's completion with respect to a single point in time. The completion of a task is of no value beyond the deadline(timing constraints). Improper use of CPU can reduce the efficiency of the system in multiprogramming computing systems. As per "Office of Aviation Research and Development Washington, D.C. 20591. Real-Time Scheduling Analysis. November 2005" [2] A real-time scheduling System is composed of the scheduler, clock and the processing hardware elements. In a real-time system, a process or task has schedule ability; tasks are accepted by a real-time system and completed as specified by the task deadline depending on the characteristic of the scheduling algorithm. In multiprogramming systems, multiple processes are being kept in memory for maximum utilization of CPU [4]. The main aim of the CPU scheduling algorithms is to minimize waiting time, turnaround time, response time and context switching and maximizing CPU utilization. This study focuses on improving the effectiveness of Round Robin CPU scheduling algorithm. There are two distinct types of systems in this field: hard real-time systems and soft real-time systems.

1) Hard real time system:

Hard real-time systems are those in which it is imperative that all computations are strictly performed within the specified time, regardless of the operating conditions. Failure to meet the timing constraints of even one task may invalidate the correctness of the entire system.

2) Soft real time systems:

Soft real-time systems, in contrast, are those in which strict adherence to the timing a constraint of tasks is not always guaranteed. These are activities that are not subject to strict deadlines. Usually, they are aperiodic tasks.

The objective of this research paper is to study the available task schedulers in practice. To make an extensive literature survey on real-time operating system with its own mechanism of scheduling concepts. After comparing some scheduling policies, like SRTF, RR, IRR and RRP etc., an idea of new round robin scheduler is proposed.

II. FAULT TOLERANCE

Fault tolerance is a major concern to guarantee availability and reliability of critical services as well as application execution.The ability to deliver service is called dependability.

Fault removal aims at detecting and eliminating existing faults. Fault removal are older that those on fault prevention. Fault removal techniques are often considered at the end of the model definition, particularly when an operational model of the system is complete.

Scheduling Algorithms are used for fault tolerance as well as fault avoidance. Some of the popular CPU scheduling algorithms are:

A. FCFS (First Come, First Serve) CPU Scheduling

In this scheduling the process that request the CPU first is allocated to CPU first. The average waiting time in FCFS is quite long [3].

B. SJF (Shortest Job First) CPU Scheduling

In this scheduling the process with the shortest CPU burst time is allocated to CPU first.

C. Priority Scheduling

In this scheduling the process with high priority is allocated to CPU first.

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

D. Round Robin Scheduling

In Round Robin (RR), a small unit of time quantum is given to each process present in the ready queue which maintains the fairness factor. Round Robin is considered the most widely used scheduling algorithm in CPU scheduling [6]-[7], also used for flow passing scheduling through a network device [8]-[9].

III. CPU SCHEDULING

In multiprogrammed computing systems, inefficiency is often caused by improper use of CPU. In multiprogramming systems, multiple processes are being kept in memory for maximum utilization of CPU [4]. CPU utilization can be maximized by switching CPU among waiting processes in the memory and running some process all the time [5]. Which process should be selected next for service, is an important question, because it affects the effectiveness of the service.

A. Scheduling Criteria

1) Context Switch: A context switch is process of storing and restoring context (state) of a pre-empted process, so that execution can be resumed from same point at a later time.

2) Throughput: Throughput is defined as number of processes completed in a period of time. Throughput is less in round robin scheduling. Throughput and context switching are inversely proportional to each other.

3) CPU Utilization: It is defined as the fraction of time cpu is in use.

4) Turnaround Time: Turnaround time is defined as the total time which is spend to complete the process and is how long it takes the time to execute that process.

5) Waiting Time: Waiting time [10] is defined as the total time a process has been waiting in ready queue.

6) Response Time: Response time is defined as the time used by the system to respond to the any particular process. Thus the response time should be as low as possible for the best scheduling.

B. States for a task.

1) Running: Task is the task under execution by the CPU. Only one task can be in the running condition.

2) Ready: Task is ready to run. Only RTOS is holding it. Any number of tasks can be in ready state.

3) Blocked: Task cannot run. It is waiting for something to happen. Any numbers of tasks can be in blocked state.

There can be other states like sleeping suspended, pended, waiting, delayed as shown in Fig.1

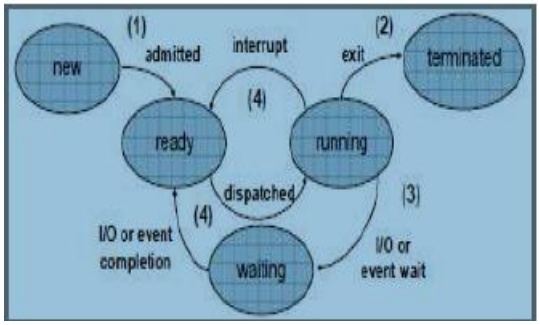

Fig.1 States Of A Process

IV. ROUND ROBIN SCHEDULING ALGORITHM

A. Round Robin: The RR scheduling algorithm [4] is given by following steps:

1) The scheduler maintains a queue of ready processes and a list of blocked and swapped out processes.

2) The Process Control Block of newly created process is added to end of ready queue. The Process Control Block of terminating process is removed from the scheduling data structures.

3) The scheduler always selects the Process Control Block from the head of the ready queue.

4) When a running process finishes its time slice, then it is moved to end of ready queue

5) The event handler performs the following actions:

When a process makes an input -output request or swapped out, its Process Control Block is removed from ready queue to blocked/swapped out list.

When I/O operation awaited by process is swapped in its Process Control Block or a process finishes is removed from blocked/swapped list to end of ready queue.

Fig.2 Example Of RR

V. PROBLEM FORMULATION

Round Robin algorithm is better. There are some disadvantages [11] of round robin CPU scheduling algorithm for operating system which are as follows:

1) Larger waiting time and Response time
2) Large number of context switches
3) Low throughput

Average waiting time of RR from all scheduling is better. However the average waiting time of SRTF is lesser than Round Robin but SRTF may lead to starvation leaving the CPU in a deadlock position. So in order to reduce the average waiting time and starvation problem we have to developed new algorithms.

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

VI. PROPOSED RESEARCH WORK

In the recent years, a number of CPU scheduling mechanisms have been developed for predictable allocation of processor.

In year 2013, AashnaBisht [16] performed a work Enhanced Round Robin(ERR), in which modify the time quantum of only those processes which require a slightly greater time than the allotted time quantum cycle. The remaining process will be executed in the conventional Round robin manner.

Mixed Scheduling (A New Scheduling Policy) [13], uses the job mix order for non preemptive scheduling FCFS and SJF. According to job mix order, from a list of N processes, the process which needs minimum CPU time is executed first and then the highest from the list and so on till the nth process.

In year 2011, RakashMohanty&Manas Das [12] performed a work in which a new variant of Round Robin scheduling algorithms by executing the processes according to the new calculated Fit factor „f" and using the concept of dynamic time quantum.

Time quantum is continuously adjusted according to the burst time of the processes in Self-Adjustment Time Quantum in Round Robin Algorithm[15].

Robust quantum time value has been proposed in [14] after arranging the process in the ascending order and taking the average of minimum and maximum burst time of the processes in the ready queue.

A. Proposed Improved Round Robin With Priority(RRP) Scheduling Algorithm:

An Improved Round Robin with priority Scheduling Algorithm, in this We have combined the features of Priority Scheduling (PS) and Round Robin (RR). It will never lead to starvation and will have lesser average waiting time.

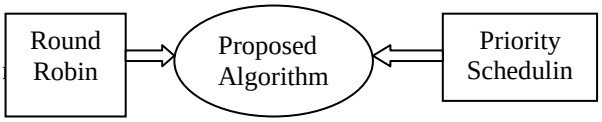

Fig.3 (RRP=RR+PS)

Following is the proposed RRP (Round Robin With Priority) CPU scheduling algorithm:

1) Make a ready queue RQUEUE of the Processes submitted for execution in order of their arrival time.

2) Execute the first process from the queue for the quantum or CPU brust time, whichever is lesser.

3) DO steps 4 to 6 WHILE queue RQUEUE becomes empty.

4) Arrange the processes in the ready queue REQUEST in the ascending order of their priority after every quantum (smaller value higher priority).

5) Execute the first process from the queue for the quantum or CPU brust time, whichever is lesser.

6) Remove the currently running process from the ready queue RQUEUE, if it has finished execution time and Go To step 4.

7) END Algorithm.

Fig.4 Example Of RRP

B. Following is the proposed IRR(Improved Round Robin) CPU scheduling algorithm:

An Improved Round Robin Scheduling Algorithm for CPU Scheduling [3]. We have combined the features of SRTF and Round Robin. The improved Round Robin will never lead to starvation and will have lesser average waiting time.

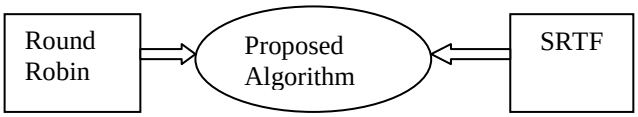

Fig.5 (IRR=RR+SRTF)

1) Make a ready queue RQUEUE of the Processes submitted for execution in order of their arrival time.

2) Execute the first process from the queue for the quantum or CPU brust time, whichever is lesser.

3) DO steps 4 to 6 WHILE queue RQUEUE becomes empty.

4) Arrange the processes in the ready queue REQUEST in the ascending order of their remaining burst time.

5) Execute the first process from the queue for the quantum or CPU brust time, whichever is lesser.

6) Remove the currently running process from the ready queue RQUEUE, if it has finished execution time and Go To step 4.

7) END Algorithm.

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

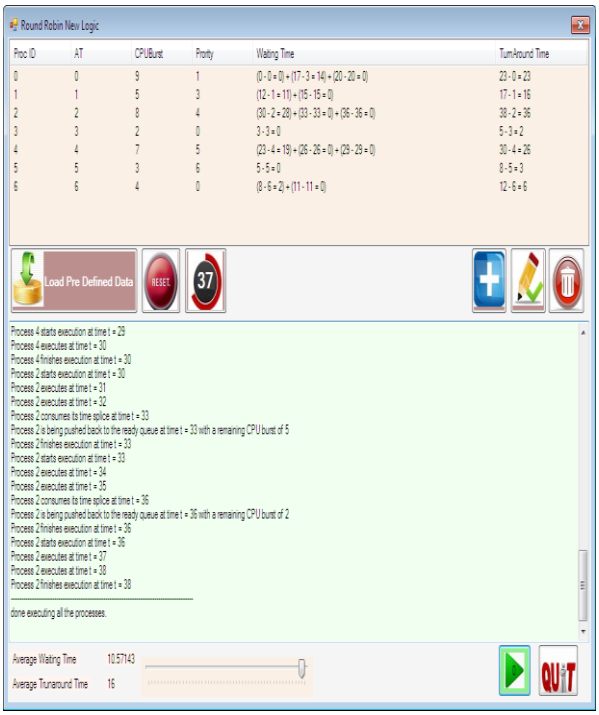

Fig.6 Example Of IRR

VII. ILLUSTRATION

Suppose we have seven processes as per the details given below:

Sr no.	Proc ID	Arrival Time	Brust Time	Priority
1.	0	0	9	1
2.	1	1	5	3
3.	2	2	8	4
4.	3	3	2	0
5.	4	4	7	5
6.	5	5	3	6
7.	6	6	4	0

Fig.7

A. Following is the illustration of proposed RRP(Improved Round Robin With Priority) CPU scheduling algorithm:

1) **In RRP (**Round Robin With Priority) ready queue with seven processes 0,1,2,3,4,5 and 6 has been considered for illustration purpose. The time quantum value is set equal to 3. At first, CPU is allocated to the process 0 at Clock 0 from the ready queue for a time quantum of 3 milliseconds (ms). After first quantum, the remaining burst time for process 0 is 9-3=6.

2) Now, Clock time is 3 there are four processes 0,1,2,3 in the ready queue and having remaining Brust time 6,5,8,2 and priority 1,3,4,0 resp.The processes 0,1,2 and 3 are arranged in the descending order of their priority (smaller value higher priority) in the ready queue which gives the sequence of processes are 3,0,1,2 having new brust time 2,6,5,8 resp. CPU is allocated to the process 3 from the ready queue for a time quantum of 2ms which is less than allocated quantum. After second quantum, the remaining burst time for process 3 is 0. The process 3 has finished execution, so it is removed from the ready queue.

3) Now, Clock time is 5 there are five processes 0,1,2,4,5 having remaining brust time 6,5,8,7,3, and priority 1,3,4,5 resp.The processes 0,1,2,4,5 are arranged in the descending order of their priority(smaller value higher priority) in the ready queue which gives the sequence of processes are 0,1,2,4,5 having new brust time 6,5,8,7,3 resp. CPU is allocated to the process 0 from the ready queue for a time quantum of 3ms. After quantum, the remaining burst time for process 0 is 6-3=3.

4) Now, Clock time is 8 there are six processes 0,1,2,4,5,6 having remaining brust time 3,5,8,7,3,4 and priority 1,3,4,5,6,0 resp.The processes 0,1,2,4,5,6 are arranged in the descending order of their priority(smaller value higher priority) in the ready queue which gives the sequence of processes are 6,0,1,2,4,5 having new brust time 4,3,5,87,3resp. CPU is allocated to the process 6 from the ready queue for a time quantum of 3ms. After quantum, the remaining burst time for process 6 is 4-3=1.

5) Now, Clock time is 11 there are six processes 0,1,2,4,5,6 having remaining brust time 3,5,8,7,3,1 and priority 1,3,4,5,6,0 resp.The processes 0,1,2,4,5,6 are arranged in the descending order of their priority(smaller value higher priority) in the ready queue which gives the sequence of processes are 6,0,1,2,4,5 having new brust time 1,3,5,87,3resp. CPU is allocated to the process 6 from the ready queue for a time quantum of 1ms which is less than allocated quantum. After quantum, the remaining burst time for process 6 is 0. The process 6 has finished execution, so it is removed from the ready queue.

6) Again it repeats the same procedure until Ready queue is not empty.

7) The average waiting time is 13.4 ms in RRP. while average waiting time is 19.7 in RR with the same set of processes.

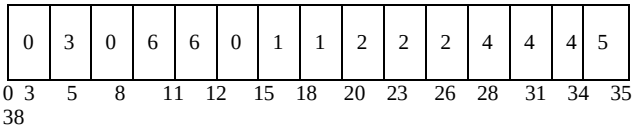

Fig.8 Gantt Chart Of RRP

B. Following is the illustration of proposed IRR(Improved Round Robin) CPU scheduling algorithm:

1) **In IRR** ready queue with the same set of processes in fig.7 seven processes 0,1,2,3,4,5 and 6 has been considered for illustration purpose. The processes are arranged in the ascending order of their arrival time in the ready queue which gives the sequence of processes are 0,1,2,3,4,5 and 6. The time quantum value is set equal to 3. CPU is allocated to the process 0 at Clock 0 from the ready queue for a time quantum of 3 milliseconds (ms). After first quantum, the remaining burst time for process 0 is 9-3=6.

2) Now, Clock time is 3 there are four processes 0,1,2,3 in the ready queue and having remaining Brust

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

time 6,5,8,2 resp.The processes 0,1,2 and 3 are arranged in the ascending order of their remaining burst time in the ready queue which gives the sequence of processes are 3,1,0 and 2 having new brust time 2,5,6,8 resp. CPU is allocated to the process 3 from the ready queue for a time quantum of 2ms which is less than allocated quantum. After second quantum, the remaining burst time for process 3 is 0. The process 3 has finished execution, so it is removed from the ready queue.

3) Now, Clock time is 5 there are five processes 0,1,2,4 and 5 having remaining brust time 6,5,8,7,3 resp.The processes 0,1,2,4 and 5 are arranged in the ascending order of their remaining burst time in the ready queue which gives the sequence of processes are 5,1,0,4 and 2 having new brust time 3,5,6,7,8 resp. CPU is allocated to the process 5 from the ready queue for a time quantum of 3ms. After quantum, the remaining burst time for process 5 is 0. The process 5 has finished execution, so it is removed from the ready queue.

4) Now, Clock time is 8 there are five processes 0,1,2,4, and 6 having remaining brust time 6,5,8,7,4 resp.The processes 0,1,2,4, and 6 are arranged in the ascending order of their remaining burst time in the ready queue which gives the sequence of processes are 6,1,0,4,2 having new brust time 4,5,6,7,8 resp. Again it repeats the same procedure until Ready queue is not empty.

5) The average waiting time is 10.57 ms in IRR. while average waiting time is 19.7 in RR with the same set of processes.

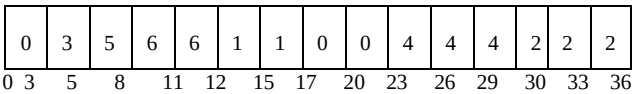

Fig.9 Gantt Chart Of IRR

Average waiting time of all scheduling algorithms with the same set of processes and the values are given below in fig. 10 :

Algorithm	Avg. Waiting Time	Avg. Turnaround Time
FCFS	16.14	21.57
SJF	12	17.4
Priority (Non-Preemptive)	13.8	19.28
Priority (Preemptive)	13.14	18.57
SRTF	10	15.4
RR	19.7	25.14
IRR (RR+SRTF)	10.57	16
RRP (RR+Priority)	13.4	18.85

Fig.10

VIII. CONCLUSION

Many CPU scheduling algorithms have been presented with some advantages and disadvantages. An improved round robin and round robin with priority CPU scheduling algorithms with proposed in this paper giving better performance than conventional RR algorithm & all the algorithms except for SRTF. But SRTF may result in starvation so RRP & IRR is better than SRTF also as it'll never result in deadlock. The waiting time and turnaround time have been reduced in the proposed IRRQ and RRP scheduling algorithm and hence the system performance has been improved. Simulation results also prove the correctness of the theoretical results. The proposed algorithm can be integrated to improve the performance of the systems.

A. Comparison Table Of RR With RRP & IRR

Comparison Of RR With RRP & IRR			
S.No.	RR	RRP	IRR
1)	Only one RR scheduling algorithm is used.	It is a combination of Round Robin and Priority scheduling.	It is a combination of Round Robin and Shortest Remaining Time scheduling.
2)	Average waiting time is high.	Average waiting time is less than conventional RR and all other scheduling except for SRTF.	Average waiting time is less than conventional RR and all other scheduling such as RRP,FCFS except for SRTF
3)	Turnaround time is very high.	Turnaround time gets optimized than RR.	Turnaround time gets optimized than RR.
4)	CPU utilization is less.	CPU utilization is high.	CPU utilization is high.
5)	Time is not efficient.	Efficient time.	Efficient time.
6)	It selects the processes for execution on the basis of first arrive with time quantum.	It selects the processes for execution on the basis of priority with time quantum.	It selects the processes for execution on the basis of shortest remaining time with time quantum.

Fig.11

REFERENCES

[1] Laplante, 1977; Sha et al., 1990; ISO/IEC, 1996; Ford, 1997; Bollella et al., 2002; Kreuzinger et al., 2002; Ngolah, Wang, & Tan, 2004).
[2] http://www.tc.faa.gov/its/worldpac/techrpt/ar05-27.pdf
[3] A. Silberschatz, P. B. Galvin & G. Gagne, [John Wiley and Sons] *"Operating System Concepts,"* [7th edition 2005] Inc,157-167.
[4] Rakesh Kumar Yadav, Abhishek K Mishra, Navin Prakash & Himanshu Sharma, An Improved Round Robin Schedduling Algorithm for CPU Scheduling *International Journal on Computer Science and Engineering, Vol. 2, No. 4, 2010,1064-1066.*

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

[5] A. Silberschatz, P. B. Galvin & G. Gagne, [John Wiley and Sons] *"Operating System Concepts,"* [7th edition 2005] Inc,157-167.
[6] Silberschatz ,Galvin and Gagne, *Operating systems concepts*, (8th edition, Wiley, 2009).
[7] Lingyun Yang, Jennifer M. Schopf and Ian Foster, *Conservative Scheduling: Using predictive variance to improve scheduling decisions in Dynamic Environments", Super Computing* 2003, USA, Phoenix, AZ, November, 15-21,
[8] Weiming Tong, Jing Zhao, *Quantum Varying Deficit Round Robin Scheduling over Priority Queues, international Conference on Computational Intelligence and Security.*(China, 2007) 252- 256.
[9] Abbas Noon1, Ali Kalakech2, Seifedine Kadry1, A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average, *IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1, May 2011.*
[10] Saroj Hiranwal and 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, July-December 2011,319-323*
[11] Ishwari Singh Rajput," A Priority based Round Robin CPU Scheduling Algorithm for Real Time Systems", (IJIET)International Journal of Innovations in Engineering and Technology Vol. 1 Issue 3 Oct 2012
[12] RakashMohanty&Manas Das, " Design and performance Evaluation of A new proposed fittest Job First Dynamic Round Robin Scheduling Algorithms", International journal of computer information systems, ISSN: 2229-5208, vol. 2, No. 2, Feb 2011.
[13] Sunita Mohan, *Mixed Scheduling, A New Scheduling Policy,Proceedings of Insight'09,* 25-26 November 2009.
[14] M Lavanya & S. Saravanan, Robust Quantum Based Low-power Switching Technique to improve System Performance, *International Journal of Engineering and Technology, Vol. 5, No. 4, 2013,3634-3638.*
[15] Rami J., *Matarneh,Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of Now Running Processes", American J. of Applied Sciences, Vol. 6, No. 10* (2009) 1831-1837.
[16] AashnaBisht, " Enhanced Round Robin Algorithm for process scheduling using varying quantum precision", IRAJ International Conference-proceedings of ICRIEST- AICEEMCS,29 th Dec 2013,pune India. ISBN: 978-93-82702-50-4