



International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

An Efficient Dynamic Round Robin CPU Scheduling Algorithm

Raman

Dr.Pardeep Kumar Mittal

Asst. Professor

M.Tech Scholar Department of Computer Science and Applications Kurukshetra University, Kurukshetra, India

Department of Computer Science and Applications Kurukshetra University, Kurukshetra, India

Abstract- CPU Scheduling is one of the fundamental concepts of Operating System. Round Robin (RR) CPU scheduling algorithm is optimal CPU scheduling algorithm in timeshared systems. The performance of the CPU depends on the selection of time quantum in timeshared systems. The time quantum taken in RR algorithm is static that decreases the performance of CPU. In this paper selection of time quantum is discussed and a new CPU scheduling algorithm for timeshared systems is proposed and is called as EDRR (Efficient Dynamic Round Robin) algorithm. The objective of this paper is to make a change in round robin CPU scheduling algorithm so that the performance of CPU can be improved. EDRR also includes advantages of round robin CPU scheduling algorithm of less chance of starvation. Round robin CPU scheduling algorithm has high context switch rates, large response time, large waiting time, large turnaround time and less throughput, these disadvantages can be improved with new proposed CPU scheduling algorithm. In this paper analysis of number of context switches, the average waiting time and the average turnaround time of processes in round robin CPU scheduling algorithm, SRBRR (Shortest Remaining Burst Round Robin), ISRBRR (Improved Shortest Remaining Burst Round Robin) and new proposed EDRR CPU scheduling algorithm has been done.

Keywords – Context switching, CPU scheduling, Gantt chart, Response time, Round Robin CPU scheduling algorithm, Turnaround time, Waiting time.

I. INTRODUCTION

The Central Processing Unit (CPU) is the heart of the computer system so it should be utilized efficiently. For this purpose CPU scheduling is very necessary. CPU Scheduling is one of the fundamental concepts of Operating System. Sharing of computer resources between multiple processes is called scheduling [1]. The various CPU scheduling algorithms are: -

- FCFS (First Come, First Serve) CPU Scheduling: In this scheduling the process that request the CPU first is allocated to CPU first.
- SJF (Shortest Job First) CPU Scheduling: In this scheduling the process with the shortest CPU burst time is allocated to CPU first.
- Priority Scheduling: In this scheduling the process with high priority is allocated to CPU first.
- Round Robin Scheduling: RR scheduling is used in timesharing systems. It is same as FCFS scheduling with pre-emption is added to switch between processes. A static Time Quantum (TQ) is used in this CPU Scheduling The various scheduling parameters [2][3] for the selection of the scheduling algorithm are:
 - 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. Context switching is wastage of time and memory that leads to the increase in the overhead of scheduler, so the goal of CPU scheduling algorithms is to optimize only these switches.
 - **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.
 - **CPU Utilization:** It is defined as the fraction of time cpu is in use. Usually, the maximize the CPU utilization is the goal of the CPU scheduling
 - **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.
 - Waiting Time: Waiting time is defined as the total time a process has been waiting in ready queue.
 - **Response Time:** Respond Time is better measure than turnaround 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.

The various characteristics of good scheduling algorithm are [3]:

• Minimum context switches.

- Maximum CPU utilization.
- Maximum throughput.
- Minimum turnaround time.
- Minimum waiting time.

In this paper related work is discussed in Section 2. Section 3 explains Round Robin Scheduling Algorithm. Section 4 introduces the Proposed Algorithm with example. Section 5 will provide conclusion of the work.

II. Previous Work Done

The RR scheduling algorithm has disadvantage that it uses static time quantum (TQ). Many research works has been done to improve the performance of the RR scheduling algorithm. A new approach SAAR [4] algorithm uses dynamic time quantum which is repeatedly adjust by the burst time of running processes. SMDRR [5] algorithm uses sub contrary mean or harmonic mean for dynamic time quantum. Similarly SRBRR [6] algorithm uses median for dynamic time quantum and a new approach for time quantum is equal to ceil (sqrt(median * highest_burst_time)) is used in ISRBRR [7].

In this paper, a comparison of RR, SRBRR, ISRBRR and new proposed algorithm has been done.

III. ROUND ROBIN SCHEDULING ALGORITHM

The RR scheduling algorithm [8] is given by following steps:-

Step 1:

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

Step 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.

Step 3:

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

Step 4:

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

Step 5:

The event handler performs the following actions:

- a) 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.
- b) 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.

There are some disadvantages [9] of round robin CPU scheduling algorithm for operating system which are as follows:

- Static time quantum
- Larger waiting time and Response time
- Large number of context switches
- Low throughput

So it can be concluded that the round robin algorithm is not suitable for real time systems. The proposed algorithm can be used for real time system which is described in next section.

IV. PROPOSED ALGORITHM

The proposed CPU Scheduling algorithm is based on the small change in round-robin scheduling algorithm. It executes the shortest job first instead of FCFS during round robin algorithm. The proposed eliminates on the drawbacks of round robin algorithm in which processes are scheduled in first come first serve manner. This round robin algorithm is not efficient for processes with smaller CPU burst. So it increases the waiting time and response time of processes which decreases in the system throughput.

The new proposed algorithm uses the dynamic time quantum instead of static time quantum. The proposed architecture eliminates the defects of implementing simple round robin architecture. The proposed algorithm will be executed in two steps which will helps to minimize a number of performance parameters such as context switches, the average turnaround time and the average waiting time.

This algorithm has following steps:

Step 1:

Processes are arranged in increasing order of their CPU burst time.

Step 2:

Set the time quantum is equal to the CPU burst time of the first process (The shortest process).

Step 3:

Calculate the median and mean of CPU burst time of all the processes

Step 4:

Set the time quantum (TQ) according to following method-

If (mean>median)

TQ = ceil (sqrt((mean * highest burst time) +(median * lowest burst time)))

Else If (median>mean)

TQ = ceil (sqrt((median * highest burst time) + (mean * lowest burst time)))

Else

TQ = mean

Step 5:

Allocate CPU to every process according to the round robin algorithm.

Step 6:

Go to step 1

For example:

CASE 1 (Increasing Order)

Consider seven processes named A, B, C, D, E, F and G with their CPU burst time

Table 1: INPUT TABLE

Process Name	CPU Burst Time (ms)
A	20
В	25
С	35
D	50
E	80
F	90
G	120

Round Robin Scheduling:

Let the time quantum is 40 ms.

Gantt chart	Α	В	С	D	Е	F	G	D	Е	F	G	F	G
0	20	45	80	120	160	200	240 250	290	330	370 3	380 420		

Context switches : 12 Average waiting Time : 152.14 ms

Average Turnaround Time: 212.14 ms

According to proposed algorithm:

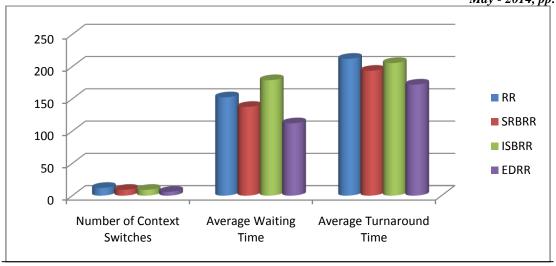
Gantt chart	D	Е	С	В	A	С	В
0	20	45	80	130	210	300	420

Context switches : 6

Average waiting Time : 112.14ms Average Turnaround Time: 172.14 ms

Table 2: Comparison between RR, SRBRR and Proposed algorithm (Case 1)

Algorithm	Time Quantum	Avg. WT	Avg. TAT	CS
RR	40	152.14	212.14	12
SRBRR	50,40,30	137.57	193.57	9
ISRBRR	78	179.00	205.57	9
EDRR	91,29	112.14	172.14	6

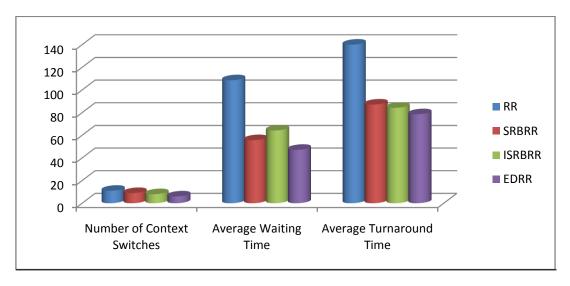


CASE 2 (Decreasing Order)

Consider seven processes with 80, 50, 40, 20, 15, 10 and 5 ms CPU burst time.

Table 3: Comparison between RR, SRBRR and Proposed algorithm (Case 2)

Algorithm	Time Quantum	Avg. WT	Avg. TAT	CS
RR	20	108.57	140.00	11
SRBRR	20,30,30	55.71	87.14	9
ISRBRR	40	64.28	84.28	8
EDRR	51,29	47.14	78.57	6



CASE 3 (Random Order)

Consider five processes with 80, 60, 20, 10 and 30 ms CPU burst time.

Table 4: Comparison between RR, SRBRR and Proposed algorithm (Case 3)

Algorithm	Time Quantum	Avg. WT	Avg. TAT	CS
RR	30	94.00	134.00	7
SRBRR	30,50	50.00	90.00	6
ISRBRR	49	73.40	93.80	6
EDRR	60,20	44.00	84.00	4

V. CONCLUSION

The paper presents a new CPU scheduling algorithm. Comparison of various algorithms i.e. round robin, SRBRR, ISRBRR and the proposed algorithm EDRR has been done. It is concluded that the proposed algorithm is more efficient than round robin algorithm because of it has less average waiting time, less average turnaround time and less number of context switches as compared to round robin, so it reduces the operating system overhead. The proposed algorithm is the combination of the shortest job first CPU scheduling algorithm and the round robin CPU scheduling algorithm with efficient & dynamic time quantum.

REFERENCES

- [1] Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Concepts", Sixth Edition.
- [2] E.O. Oyetunji, A. E. Oluleye," Performance Assessment of Some CPU Scheduling Algorithms", Research Journal of Information Technology, *1*(1): pp 22-26, 2009
- [3] Ajit Singh, Priyanka Goyal, Sahil Batra," An Optimized Round Robin Scheduling Algorithm for CPU Scheduling", (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 07, 2383-2385, 2010
- [4] Rami J. Matarneh. "Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of Now Running Processes", American J. of Applied Sciences 6(10):1831-1837, 2009.
- [5] Sourav Kumar Bhoi, Sanjaya Kumar Panda, Debashee Tarai, "Enhancing cpu performance using subcontrary mean dynamic round robin (smdrr) scheduling algorithm", JGRCS, Volume 2, No. 12, December 2011, pp.17-21
- [6] Rakesh Mohanty, H. S. Behera, Khusbu Patwari, Monisha Dash, "Design and Performance Evaluation of a New Proposed Shortest Remaining Burst Round Robin (SRBRR) Scheduling Algorithm", International Symposium on Computer Engineering & Technology (ISCET), Vol 17, 2010
- [7] P.Surendra Varma, "A FINEST TIME QUANTUM FOR IMPROVING SHORTEST REMAINING BURST ROUND ROBIN (SRBRR) ALGORITHM" Journal of Global Research in Computer Science, 4 (3), March 2013, 10-15
- [8] Rakesh Kumar Yadav, Abhishek K Mishra, Navin Prakash, Himanshu Sharma," An Improved Round Robin Scheduling Algorithm for CPU Scheduling", (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 1064-1066, 2010
- [9] 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