An Optimized Round Robin Scheduling Algorithm for CPU Scheduling

Ajit Singh, Priyanka Goyal, Sahil Batra

Department of Computer Science and Engineering
The Technological Institute of Textile and Science,Bhiwani, Haryana

Abstract:- The main objective of this paper is to develop a new approach for round robin scheduling which help to improve the CPU efficiency in real time and time sharing operating system. There are many algorithms available for CPU scheduling. But we cannot implemented in real time operating system because of high context switch rates, large waiting time, large response time, large trn around time and less throughput. The proposed algorithm improves all the drawback of simple round robin architecture. The author have also given comparative analysis of proposed with simple round robin scheduling algorithm. Therefore, the author strongly feel that the proposed architecture solves all the problem encountered in simple round robin architecture by decreasing the performance parameters to desirable extent and thereby increasing the system throughput.

Keywords: CPU scheduling,RR scheduling algorithm,Turnaround time,Waiting time,Response time,,Context switch rate, Gantt chart

I. Introduction

Modern operating system become more complex, when they switch from a single task to multitasking environment. CPU scheduling is an essential operating system task, therefore its scheduling is central to operating system design. When there is more than one process in the ready queue waiting in turn to be assigned to the CPU, the operating system must decide through the scheduler the order of execution in which they execute. Allocating CPU to a process requires careful attention to assure fairness and avoid process starvation for CPU.

According to Silberchatz, Galvin and Gagne; CPU scheduling play a vital role by switching the CPU among several process. The aim of operating system to allow a number of processes concurrently in order to maximize the CPU utilization. In a multi-programmed Operating system, a process is executed until it must wait for the completition of some input-output request.

According to Sabrina, F.C.D, Nguyen in Processing resources scheduling in programmable networks; Scheduling is a fundamental operating system function. Beside that, CPU scheduling determines which process run when there are multiple runable processes. CPU scheduling is important because it can have a big effect on resource utilization and other performance parameters. There exist a number of CPU scheduling algorithm like First-Come-First-Serve, Shortest-Job-Scheduling, Round Robin scheduling etc, but due to a number of disadvantages these are severly used except Round Robin scheduling in timesharing and real time operating system. The paper is organised in sections. Section 2 and 3 discuss the simple one algorithm and scheduling objectives. In section 4 and 5, scheduling criteria and proposed Round Robin scheduling algorithm are given. Section 6 discusses the case study of comparative analysis of proposed with simple RR algorithm. Finally, in section 7 conclusion are given.

II. Simple RR Scheduling Algorithm:

According to Silberchatz, Galvin, Gagne in operating system design and operating system by D M Dhamdhere, the simple RR scheduling algorithm is given by following steps:-

1. The schedular maintains a queue of ready processes and a list of blocked and swapped out processes.
2. The PCB of newly created process is added to end of ready queue. The PCB of terminating process is removed from the scheduling data structures.
3. The schedular always selects the PCB at head of the ready queue.
4. When a running process finishes its slice, it is moved to end of ready queue.
5. The event handler perform the following action

a) When a process makes an input -output request or swapped out,its PCB is removed from ready queue to blocked/swapped out list

b)When input-output operation awaited by a process finishes or process is swapped in its process control block is removed from blocked/swapped list to end of ready queue.

III. Scheduling Objective

A system designer must consider a variety of factors when developing a scheduling discipline ,such as what type of systems and what are user's needs. Depending on the system, the user and designer might expect the scheduler to:

- Maximize throughput: A scheduling discipline should attempt to service the maximum number of processes per unit of time.
- Avoid indefinite postponement and starvation: A process should not experience an unbounded wait time before or while process service.

- Minimise overhead: Overhead causes wastage of resources. But when we use certain portion of system resources effectively,then overhead can greatly improve overall system performance.
- Enforcement of priorities:if system assign priorities to processes ,the scheduling mechanism should favour the higher-priority processes.
- Achieve balance between response and utilisation:The scheduling mechanism should keep resources of system busy.
- Be predictable: By minimizing statistical variance in process response times ,a system can guarntee that processes will recieve predictable service levels.
- Favour processes exhibiting desirable behaviour.
- Degrade gracefully under heavy load.

A system can accomplish these goals in several ways. The scheduler can prevent indefinite postponement of processes through aging. The scheduler can increase throughput by favouring processes whose requests can be satisfied quickly, or whose completion frees other processes to run.

IV. Scheduling criteria:

There are lot of CPU scheduling algorithms having different properties,and the choice of a particular algorithm may favour one class of processes over anothers.For selection of a algorithm for a particular situation ,we must consider properties of various algorithms.The criteria include the following:

- **Context Switch:**A context switch is computing process of storing and restoring state of a CPU so that execution can be resumed from same point at a later time.Context switch are usually computationally intensive,lead to wastage of time,memory,schedular overhead so much of the design of operating system is to optimize these switches.
- **Throughput:**Throughput is defined as number of process completed per unit time.Throughput will be slow in round robin scheduling implementation.Context switch and throughput are proportional to each other.
- **CPU Utilization:**We want to keep the CPU as busy as possible.
- **Turnaround Time**:Turnaround time is sum of periods spent waiting to get into memory,waiting in ready queue,executing on CPU and doing input-output. It should be less.
- **Waiting Time**:Waiting time is the amount of time a process has been waiting in ready queue. The CPU scheduling algorithm does not affect the amount of time during which a process executes or does input-output; it affects only the amount of time that a process spends waiting in ready queue.
- **Response Time:**Response time is the time it takes to start responding,not the time it takes to output the response.Large response time is a drawback in round robin architecture as it leads to degradation of system performance.

So we can conclude that a good scheduling algorithm for real time and time sharing system must possess following characteristics:

- Minimum context switches.
- Maximum CPU utilization.
- Maximum throughput.
- Minimum turnaround time.
- Minimum waiting time.
- Minimum response time .

V. Proposed Round Robin Scheduling Algorithm

The proposed algorithm will be executed in three phase which help to minimize a number of performance parameters such as context switches, waiting time and average turnaround time. The algorithm performs following steps as:

Phase 1: Allocate every process to CPU, a single time by applying RR scheduling with a initial time quantum(say k units).

Phase 2: After completing first cycle perform the following steps:
a) Double the initial time quantum (2k units).
b) Select the shortest process from the waiting queue and assign to CPU.
c) After that we have to select the next shortest process for execution by excluding the already executed one in this phase.

Phase3: For the complete execution of all the processes we have to repeat phase 1 and 2 cycle.

VI. Case Studies

Five processes have been defined with CPU burst time, these five processes are scheduled in round robin and also in the proposed algorithm. The context switch, average waiting time, average turn around time has been calculated and the results were compared. For doing this we have carry a number of experiments but here I will discuss only two experiments because we assured results analysis is remain unchanged. This paper presents two types of problem analysis where first one is without arrival time and second one is with arrival time.

Experiment1:

In this we have to consider the processes only with CPU burst time and also let round robin quantum =5

According to simple RR scheduling:

Process Id	CPU Burst Time (ms)
P1	22
P2	18
P3	9
P4	10
P5	5

RR quantum=5
According to the simple RR algorithm:

Gantt chart:

P1	P2	P3	P4	P5	P1	P2	P3	P4	P1	P2

P1	P2	P1

No. of context switches =13
Average waiting time=34 ms
Average turnaround time= 46.8 ms

According to proposed algorithm:-

Gantt chart:

P1	P2	P3	P4	P5	P3	P4	P2	P1	P1	P2

No of context switches = 9
Average waiting time =29.8 ms
Average turn around time = 42.6 ms

Experiment 2:

In this we have to consider the processes along with CPU burst time and process arrival time and also let round robin quantum =5
According to simple RR scheduling:

Process Id	Arrival Time	CPU Burst Time (ms)
P1	0.000	4
P2	2.004	7
P3	5.010	5
P4	6.020	8
P5	8.019	9

According to the simple RR algorithm:

Gantt chart:

P1	P2	P3	P4	P5	P1	P2	P3	P4	P5	P2

P4	P5

No of context switches = 12
Average waiting time =19 ms
Average turn around time = 25.6 ms

According to proposed algorithm:-

Gantt chart:

P1	P2	P3	P4	P5	P1	P3	P2	P4	P5

No of context switches = 9
Average waiting time =17 ms
Average turn around time = 23.2 ms

We can see from the above experiment average waiting time and average turnaround time both are reduced by using our proposed algorithm. The reduction of average waiting time and average turnaround time shows maximum CPU utilization and minimum response time. We observed that proposed algorithm much more efficient as compared to simple RR algorithm.

Conclusion

A comparative study of simple RR algorithm and proposed one is made. It is concluded that the proposed algorithm is superior as it has less waiting response time, usually less pre-emption and context switching thereby reducing the overhead and saving of memory space. Future work can be based on this algorithm modified and implemented for hard real time system where hard deadline systems require partial outputs to prevent catastrophic events.

References:

[1] http://en.wikipedia.org./wiki/scheduling
[2] Operating Systems Sibsankar Haldar 2009,Pearson Education, India
[3] D.M. Dhamdhere operating Systems A Concept Based Approach, Second edition, Tata McGraw-Hill, 2006.
[4] Sabrina, F.C.D, Nguyen, S.jha, D. Platt and F. Safaei, 2005.Processing scheduling in programmable networks. Computer commun, 28:676-687.
[5] Silberchatz, Galvin and Gagne, 2003. Operating systems concepts.