An Improved Optimum Multilevel Dynamic Round Robin Scheduling Algorithm

S. R. Chavan, P. C. Tikekar

Abstract — The main objective of this paper is to improve the Round Robin scheduling algorithm using the dynamic time slice concept. One of the fundamental functions of an operating system is scheduling. Sharing of computer resources between multiple processes is called as scheduling. The main use of round robin scheduling algorithms is in the real time systems but still it have several difficulties. The intention should be allowed as many as possible running processes at all time in order to make best use of CPU. CPU scheduling has strong effect on resource utilization as well as overall performance of the system. Round Robin algorithm performs optimally in timeshared systems, but it is not suitable for soft real time systems, because it gives more number of context switches, larger waiting time and larger response time. In this paper, a new CPU scheduling algorithm called An Optimum Multilevel Dynamic Round Robin Scheduling Algorithm is proposed, which calculates intelligent time slice and changes after every round of execution. The suggested algorithm was evaluated on some CPU scheduling objectives and it was observed that this algorithm gave good performance as compared to the other existing CPU scheduling algorithms.

Index Terms — Operating System, CPU scheduling, Round Robin, utilization, waiting time, response time, throughput, fairness, proposed algorithm.

1 INTRODUCTION

Operating system is a system software which makes an interface between end user and computer hardware, so that the user can handle the system in a convenient manner [9][7]. The CPU is one of the primary computer resources, so its scheduling is essential to an operating system’s design [5]. In a single-processor system, only one process can run at a time; any others must wait until the CPU is free and can be rescheduled. The objective of multiprogramming is to have some process running at all times, to maximize CPU utilization [17][4]. Almost all computer resources are scheduled before use. The CPU is, of course, one of the primary computer resources. CPU scheduling determines which processes runs when there are multiple run-able processes [18].

OS provides basically three types schedulers: a long term scheduler (also known as an admission scheduler or high level scheduler), a mid-term or medium-term scheduler and a short-term scheduler (also known as a dispatcher or CPU scheduler)[16]. In Round Robin (RR) every process has equal priority and is given a time quantum or time slice after which the process is preempted. Although RR gives improved response time and uses shared resources efficiently [6][17]. Its limitations are larger waiting time, undesirable overhead and larger turnaround time for processes with variable CPU bursts due to use of static time quantum. This motivates us to implement RR algorithm with sorted remaining burst time with dynamic time quantum concept. Another concept employed in this algorithm is the use of more than one cycle instead of a single Round Robin.

In OS, there are different algorithms like First Come First Serve (FCFS), shortest Job First (FJF), Priority totally based on priority of the arriving processes, Round Robin(RR). But all algorithms have its own deficiency and limitations.[14] In this paper, I proposed a new approach for round robin scheduling algorithm which helps to improve the efficiency of CPU.

A. Scheduling Criteria:-

Different CPU scheduling algorithms have different properties, and the choice of a particular algorithm may favor one class of processes over another [17]. In choosing which algorithm to use in a particular situation, we must consider the properties of the various algorithms. Many criteria have been suggested for comparing CPU scheduling algorithms. Which characteristics are used for comparison can make a substantial difference in which algorithm is judged to be best [17][18][16]. The criteria include the following [4]

a. Utilization/Efficiency: keep the CPU busy 100% of the time with useful work [2]
b. Throughput: maximize the number of jobs processed per hour. [3]
c. Turnaround time: from the time of submission to the time of completion - minimize the time batch users must wait for output [3]
d. Waiting time: Sum of times spent in ready queue - Minimize this [3]
e. Response Time: time from submission till the first response is produced, minimize response time for interactive users [4]
f. Fairness: make sure each process gets a fair share of the CPU [7]
The paper is divided into four parts. Part I gives a brief introduction on the various aspects of the scheduling algorithms, the approach to the current paper and the motivational factors leading to this improvement. Part II presents the proposed algorithm and illustration of our proposed new algorithm (OMDRR). In Part III, an experimental analysis and Result of our algorithm (OMDRR) and its comparison with the static RR algorithm. Conclusion is presented in Part IV followed up by the references used.

2 PROPOSED ALGORITHM

The early the shorter processes are removed from the ready queue, the better the turnaround time and the waiting time. So in our algorithm, the shorter processes are given more time quantum so that they can finish their execution earlier. Here shorter processes are defined as the processes having less assumed CPU burst time than the previous process. Performance of RR algorithm solely depends upon the size of time quantum. If it is very small, it causes too many context switches [12]. If it is very large, the algorithm degenerates to FCFS. So our algorithm solves this problem by taking dynamic intelligent time quantum where the time quantum is repeatedly adjusted according to the shortness component.

A. Our Proposed Algorithm

In our algorithm, combines the working principle of fundamental scheduling algorithms. Dynamically Time Slice (DTS) is calculated which allocates different time quantum to each process based on priority, shortest CPU burst time and context switch avoidance time.

B. Experimental Analysis

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 turnaround 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.

C. Comparison of Proposed Algorithm and Previous Algorithm with Arrival Time and Without Arrival Time

Experiment 1:

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:

<table>
<thead>
<tr>
<th>Process Id</th>
<th>CPU Burst Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>22</td>
</tr>
<tr>
<td>P2</td>
<td>18</td>
</tr>
<tr>
<td>P3</td>
<td>9</td>
</tr>
<tr>
<td>P4</td>
<td>10</td>
</tr>
<tr>
<td>P5</td>
<td>5</td>
</tr>
</tbody>
</table>

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

According to proposed algorithm:

<table>
<thead>
<tr>
<th>Process Id</th>
<th>CPU Burst Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P3</td>
<td>1</td>
</tr>
<tr>
<td>P4</td>
<td>2</td>
</tr>
<tr>
<td>P1</td>
<td>3</td>
</tr>
<tr>
<td>P2</td>
<td>4</td>
</tr>
<tr>
<td>P5</td>
<td>5</td>
</tr>
</tbody>
</table>

No of context switches = 9
Average waiting time =28.6 ms
Average turnaround time = 41.4 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 =3

According to simple RR scheduling:

<table>
<thead>
<tr>
<th>Process Id</th>
<th>Arrival Time</th>
<th>CPU Burst Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>P2</td>
<td>2.4</td>
<td>7</td>
</tr>
<tr>
<td>P3</td>
<td>5.1</td>
<td>5</td>
</tr>
<tr>
<td>P4</td>
<td>6.2</td>
<td>8</td>
</tr>
<tr>
<td>P5</td>
<td>8.019</td>
<td>9</td>
</tr>
</tbody>
</table>
According to the simple RR algorithm:

```
Figure 2 Gantt chart

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | 2 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |
```

No of context switches = 8
Average waiting time = 14.2 ms
Average turnaround time = 20.8 ms

We can see from the above experiment context switch, average waiting time and average turnaround time both are reduced by using our proposed algorithm. The reduction of context switch, 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.

3. 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.

4. References:


4) An Optimised Multilevel Dynamic Round Robin Scheduling Algorithm by Neetu Goel in Published in National Conference on Information Communication & Networks, Dated: April 6, 2013 organized by Tecnia Institute of Advanced Studies


6) Optimized Round Robin CPU Scheduling Algorithm by Sukumar Babu B., Neelima Priyanka N. & Dr. P. Suresh Varma in Global Journal of Computer Science and Technology Interdisciplinary Volume 12 Issue 11 Version 1.0 Year 2012


12) CPU Scheduling and Memory Management for Interactive Real-Time Applications by Shinpei Kato, Yutaka Ishikawa, Ragunathan Rajkumar


14) “Adaptive Round Robin Scheduling using Shortest Burst Approach Based on Smart Time Slice”. by Saroj Hiranwal, Dr. K.C. Roy

16) Operating system concepts and techniques by M.Naghibzadeh, ISBN: 978-0-5953-7597-4

17) A. Silberschatz, and P. Galvin, Operating System Concept, Ed.5, John Wiley and Sons (Asia), Inc,(1999).