A Dynamic Time Quantum SJRR CPU Scheduling Algorithm

Shailendra Shukla, Lalit Kishore

Abstract—CPU Scheduling is a vital discipline which helps us gain deep insight into the complex set of policies and mechanisms used to govern the order in which tasks are executed by the processor. This article proposes an Dynamic time quantum Shortest Job Round Robin CPU Scheduling algorithm having better average waiting time (AWT) and average turnaround time (ATT) as compared to other CPU Scheduling techniques. The primary objective of this algorithm is to optimize system performance according to the criteria deemed most important by system designers.

Index Terms—Shortest Job First Algorithm, Round Robin Algorithm, Priority Algorithm, Average Waiting Time, Turnaround Time, Response Time, Throughput.

I. INTRODUCTION

Scheduling is a technique which involves complex set of policies and mechanisms working at the back of a processor, instructing it the order in which it should execute a given set of processes. Process is a smallest work unit of a program which requires a set of resources for its execution that are allotted to it by the CPU. These processes are many in number and keep coming in the queue one after the other. In order to execute them in a particular fashion, different scheduling techniques are employed that enable faster and efficient process execution thereby reducing the waiting time faced by each process and increasing CPU utilization. A process has five basic states namely New, Ready, Running, Waiting and Terminate [1] [5].

Throughout its lifetime a process migrates between various scheduling queues by different schedulers until it gets terminated. These queues mainly contain the ready queue which contains set of processes ready for CPU response. The second queue is the device or the I/O queue which contains all the processes that are waiting for I/O response [1]. The operating system must select processes for scheduling from these queues in a specific manner. This selection process using a particular scheduling technique is carried out by schedulers.

Schedulers in general try to maximize the average performance of a system according to the given criterion [2]. Scheduling Algorithms can be broadly classified into preemptive and non-preemptive scheduling disciplines. The algorithm proposed in this article is preemptive in nature and attempts to give fair CPU execution time by focusing on average waiting time and turnaround time of a process. This article comprises of the following sections: Section 2 presents scheduling parameters which will decide against which parameters the new CPU scheduling algorithm will be tested. Section 3 introduces existing scheduling algorithm. Section 4 explains the SJRR scheduling algorithm. Section 5 contains pseudo code of the algorithm. Section 6 explains the two basic elements that make up the simulation and provide an interactive user interface. Section 7 presents a graphical comparison of the new algorithm with existing techniques. Last but not the least Section 8 will provide conclusion of the work.

II. SCHEDULING PARAMETERS

Different scheduling algorithms have different Characteristics which decide selection of processes using different criteria for execution by CPU. The criteria which decide how one algorithm differs from the other have been listed below:

A. Processor utilization
It is the average fraction of time during which the processor is busy [2]. Being busy means the processor is not idle.

B. Throughput
It refers to the amount of work completed in a unit of time [2]. That is, the number of user jobs executed in a unit of time. The more the number of jobs, the more work is done by the system.

C. Turnaround Time
It is defined as the time taken to execute a given process [1]. That is, it is the time spends by the process in the system from the time of its submission until its completion by the system.

D. Waiting Time
Scheduling algorithms do not affect the amount of time during which a process executes or does I/O, it affects only the amount of time spend by the process in the ready queue [1]. That is, the amount of time spent in the ready queue by the process a waiting CPU execution.

E. Response Time:
While turnaround time includes total time taken by the process from the time of its submission until the time of its completion, response time is the measure of time from the submission of requests until the first response is produced [1]. This response time does not include the time taken to output that response.

Manuscript received August 19, 2014

Shailendra Shukla, Lecturer, Department of Computer Science and Engineering, Govt. R.C. Khaitan Polytechnic College, Jaipur, Rajasthan India
Lalit Kishore, Asst. Prof., Department of Computer Science, Apex Group of Institution, Jaipur, Rajasthan India
III. OVERVIEW OF EXISTING CPU SCHEDULING ALGORITHMS

CPU scheduling algorithms aim at deciding which Processes in the ready queue are to be allotted to the CPU. Discussed in this section are some common CPU scheduling algorithms.

A. First Come First Served (FCFS) Scheduling
FCFS employs the simplest scheduling technique on the basis of first come first served. The work load is processed in the order of arrival, with no preemption [2]. Once a process has been submitted to the CPU, it runs into completion without being interrupted. Such a technique is fair in the case of smaller processes but is quite unfair for long unimportant jobs [3]. Since FCFS does not involve context switching therefore it has minimal overhead. It has low throughput since long processes can keep processor occupied for a long time making small processes suffer. As a result waiting time, turnaround time and response time can be low [4].

B. Shortest Job First (SJF) Scheduling
Shortest Job first is non-preemptive in nature in which process with smallest estimated run time to completion is executed next. SJF reduces average waiting time of processes as compared to FCFS. SJF favors shorter processes over longer ones which is an overhead as compared to FCFS [6]. It selects the job with the smallest burst time ensuing CPU availability for other processes as soon as the current process reaches its completion. This prevents smaller processes from suffering behind larger processes in the ready queue for a long time [3] [7].

C. Round Robin (RR) Scheduling
Round Robin is preemptive in nature. It employs FCFS for process execution by assigning a quantum or time slice to each process [7]. As soon as the quantum expires control is forcefully taken from the current process under execution and is transferred to the next in the queue for the same period of time slice [3]. The outcome of RR algorithm in term of performance depends entirely on the size of time quantum. If the quantum is very large, RR algorithm works the same as the FCFS algorithm. If the quantum is very small, RR algorithm makes the user feels processor sharing between multiple processes very fast. Average waiting time is high because of FCFS policy and context switching [1].

D. Priority Based Scheduling
Priority scheduling executes processes based on their priority which may be assigned by the system or by the user himself [3]. Processes with the high priority are executed first and those with low priorities are executed next [6]. Processes with equal priority values are executed using FCFS approach [1].

IV. PROPOSED SCHEDULING ALGORITHM
For this the steps taken are
In starting only one job is arrive then time quantum is assigned as mid. Of the burst time of this job then the method to calculate the time quantum is a

1. The jobs are different Arrival time. If we put the jobs in Ascending order by this formula Factor
F=Arrival time + CPU Burst time.
Then arrange the jobs by this factor F in ascending order.

2. Then the time Quantum is calculated as Mean of jobs burst times according to this
   1. If the jobs are odd then
      Mean is middle job burst time.
   2. If the jobs are even then
      Mean is middle job position burst time + (middle job+1 jobs position burst time)

The time quantum is assigned to each process and is recalculated taking the remaining burst time in account after each cycle. This procedure goes on until the ready queue is empty.

V. RESULT AND ANALYSIS
The round robin scheduling is best for those processes that want high response time. In general round robin algorithm the time quantum is fix and all processes Run according to this time quantum value. In this dynamic time quantum approach we get better result on the basis of Cpu scheduling criteria. The following result and analysis show the performance of this approach
There are some processes P0, P1, P2, P3, P4, P5, P6
And there burst time 15, 13, 8, 10, 12,26,7
And there arrival time is 0,2,4,5,1,6,3
Then by the fix time quantum round robin scheduling algorithm.

<table>
<thead>
<tr>
<th>Avg. waiting time of processes</th>
<th>Avg. turnaround time of processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>51.85</td>
<td>64.85</td>
</tr>
</tbody>
</table>

By the dynamic time quantum SJRR new proposed method the

<table>
<thead>
<tr>
<th>Avg. waiting time of processes</th>
<th>Avg. turnaround time of processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>32.74</td>
<td>45.71</td>
</tr>
</tbody>
</table>

Now this is clear that the proposed method gives better performance.
VI. CONCLUSION

From the comparison of the obtained results, it is observed that proposed algorithm has successfully beaten the existing CPU scheduling algorithms. It provides good performance in the scheduling policy. As SJF is optimal scheduling algorithm but for large processes, it gives increased waiting time and sometimes long processes will never execute and remain starved. This problem can be overcome by the proposed algorithm. In future, the working of proposed algorithm will be tested on any open source operating system.

REFERENCES


