Research Journal of Applied Sciences, Engineering and Technology 10(8): 938-941, 2015
DOI:10.19026/rjaset.10.2450
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: February 22, 2015 Accepted: March 7, 2015 Published: July 20, 2015

Research Article

Enhanced Precedence Scheduling Algorithm with Dynamic Time Quantum (EPSADTQ)

[1]G. Siva Nageswara Rao, [2]S.V.N. Srinivasu, [3]N. Srinivasu and [4]G. Ramakoteswara Rao
[1]Department of Computer Science and Engineering, K L University,
[2]Department of Computer Science, HOD, PNC & KR PG College Narasaraopet,
[3]Department of Computer Science and Engineering, K L University,
[4]Department of Computer Science and Engineering, VR Siddhartha Engineering College, Vijayawada, India

Abstract: This study proposes a new algorithm which is a logical extension of the popular Round Robin CPU scheduling algorithm. The Round Robin algorithm can be effective only if the time quantum is chosen accurately. Even by taking mean average of burst times as time quantum, the performance of the RR cannot be improved beyond a certain point. However, the novel method proposed here, suggests that a priority be assigned to each process based on balanced precedence factor. The novel method also uses mean average as a time quantum. Experiments are conducted in order to measure the effectiveness of this novel method. The results clearly showed that EPSADTQ is superior to RR and PSMTQ and its variants. EPSADTQ resulted in a significant reduction of the no. of context switches, average waiting time and average turnaround time.

Keywords: Avg. waiting time, avg. context switch, process, ready queue, turnaround time

INTRODUCTION

CPU scheduling is an essential operating system task, which is the process of allocating the CPU to a specific process.

Scheduling requires careful attention to ensure fairness and avoid process starvation in the CPU. Main aim of the scheduling is to maximize CPU utilization, Throughput and to minimize response time, context switches, waiting time and turnaround time. Existing scheduling algorithms are based on priority of the process are not fair and suffer from problem of starvation. If we use round robin algorithm then we can achieve fairness but at the cost of neglecting the effect of priority totally and large number of context switches. The Precedence scheduling algorithm with Intelligent Service time is still producing large number of context switches and Factor of Precedence is not balanced among the priority and burst times. This motivated me to design and develop a novel algorithm which uses Balanced Factor of Precedence (BFP) to determine the order of execution of various processes. It uses Mean average of burst times of processes as its time quantum. This will reduce the number of context switches, average waiting time and average turnaround time. It also attains better response time.

PRELIMINARIES

A process is Program in execution. The processes waiting to be assigned to a processor are put in a queue called ready queue. The time for which a process holds the CPU is known as burst time. Arrival Time is the time at which a process arrives at the ready queue. The interval from the time of submission of a process to the time of completion is the turnaround time. Waiting time is the amount of time a process has been waiting in the ready queue. The number of times CPU switches from one process to another is known as context switch. The optimal scheduling algorithm will have minimum waiting time, minimum turnaround time and minimum number of context switches.

Basic scheduling algorithms:

First Come First Serve (FCFS): First-Come-First-Serve (FCFS) algorithm, the CPU is assigned immediately to that process which arrives first at the ready queue.

Shortest Job First (SJF): In SJF the process with low burst time is scheduled first.

If two processes having same burst time and arrival time, then FCFS procedure is used.

Shortest Remaining Time First (SRTF): Similar as the SJF with pre emption, which small modification. For scheduling the jobs system need to consider the remaining burst time of the job which is presently executed by the CPU also along with the burst time of the jobs present in the ready queue.

Corresponding Author: G. Siva Nageswara Rao, Department of Computer Science and Engineering, K L University, India

Priority scheduling algorithm: The priority to each process and selects the highest priority process from the ready queue for the execution.

Round robin scheduling algorithm: Round Robin (RR) is one of the oldest, simplest and fairest and most widely used scheduling algorithms, designed especially for time-sharing systems. In this every process has equal priority and is given a time quantum after which the process is preempted.

LITERATURE REVIEW

Efforts have been made to modify round robin in order to give better results of turnaround time, average waiting time and minimize context switches (Varma, 2012; Siva Nageswara Rao, 2014). The concept of average mean time quantum has been suggested in Noon *et al*. (2011) and Siva Nageswara Rao (2014, 2015). Employing the concept of priority along with round robin has been suggested in Himanshi and Prashant (2012) and Siva Nageswara Rao (2014, 2015).

Proposed algorithm: Newly proposed algorithm overcomes the problem of starvation since it considers both priority as well as burst time equally called balanced factor of precedence to compute the order of execution:

Using, $BFP_i = 0.5 * P_i + 0.5 * BT_i$

where, BT is assigned in such a way that shorter process gets Lower number P denotes Priority of the Process.

To produce good response time, the algorithm shares the time quantum among the processes. It is best suited for time shared systems. The time quantum is taken as the mean average of burst times of the processes with varying dynamic time quantum (Siva Nageswara Rao, 2014). Due to this, it does not produce poor response time and also it will have less number of context switches. That is why the proposed algorithm produces better results.

The algorithm is as follows:

1. Begin
2. Burst times, Priority of each process are taken as input
3. Give value of Burst Time in such a way that shorter process gets lower number.
4. Evaluate Balanced Factor of Precedence (BFP) as follows:

   $BFP_i = 0.5 * P_i + 0.5 * BT_i$

   For each process in the Queue. Where P denotes priority and Burst Time is the value denoted in step 3.
5. The Order of Execution (OE) is assigned to processes such that the process with lowest BFP will be executed first. If there are any ties in BFP values then it will be resolved by considering process with lowest burst time.
6. Now, the execution sequence is found and compute the time quantum using:

   $TQ = \sum \text{Burst time}_i / N$

   where, N represents Number of Processes i.e., Time quantum is the Mean Average of the Burst times of processes in the Ready Queue.
7. Assign a Time Quantum to process P_i for each i if the remaining burst time of current execution process is less than or equal to one time quantum then execute the same process otherwise go for next process in the ready queue.
8. Repeat step 7 until the ready queue becomes empty.
9. END

EXPERIMENTS AND ILLUSTRATION

Assumptions: All experiments are assumed to be performed in single processor environment and all the processes are not dependent from each other. Parameters like burst time and priority are known prior to submission of process. All processes are CPU bound. No process is input and output bound.

Illustration and results:

Example: Consider there are 5 processes. Burst time and arrival time have been assumed to be in milliseconds. We calculate BFP to find the order of execution and assign value of Burst Time in such a way that shorter process gets lower number. Mean average is calculated and considered as time quantum.

Let us consider the following processes with their priorities and burst times.

Now Lower number of Burst Time is given to shorter process.

Balanced Factor of Precedence (using $BFP_i = 0.5 * P_i + 0.5 * BT_i$) is evaluated to find the order of execution. Process with Lowest BFP value will be executed first and Process with high BFP value will be executed last. Hence Table 1 to 3 having the all the information regarding all the process of BT, BFP and Execution Order. With the help of Execution Order, process the jobs from the ready queue.

Now Time Quantum is computed as Mean average of Burst times:

Time Quantum (TQ) = $\sum$ Burst time$_i$/n
TQ = (11+53+8+41+20)/5 = 27
AWT = 28+80+20+39+0/5 = 33.
ATT = 39+133+28+80+20/5 = 60.

Table 1: Various jobs with their burst time

Process	Burst time	Priority
P1	11	3
P2	53	5
P3	8	4
P4	41	2
P5	20	1

Table 2: Jobs with execution order

Process	Burst time	Priority	BT	BFP	Execution order
P1	11	3	2	2.5	3
P2	53	5	5	5.0	5
P3	8	4	1	2.5	2
P4	41	2	4	3.0	4
P5	20	1	3	2.0	1

Table 3: Jobs with execution grant chart

P5	P3	P1	P4	P4	P2	P2	
0	20	28	39	66	80	107	133

NCS = 4
PSMTQ: AWT = 45.8
ATT = 70.4
NCS = 6
Ex.2: PSMTQ: AWT = 38.8
ATT = 58.2
NCS = 7
EPSADTQ: AWT = 28.4
ATT = 47.8
NCS = 4
Ex.3: PSMTQ: AWT = 40.2
ATT = 59.6
NCS = 7
EPSADTQ: AWT = 29
ATT = 47
NCS = 4
Ex.3: PSMTQ: AWT = 90.2
ATT = 137.6
NCS = 6
EPSADTQ: AWT = 66.8
ATT = 114.2
NCS = 4

Compare the above results with other variants of RR and PSMTQ scheduling algorithm are as follows:

From the above graphs we can observe that the proposed algorithm is producing optimal values in terms of context switches, average turnaround time and average response time. Figure 1 to 4 shows comparison among various scheduling algorithms.

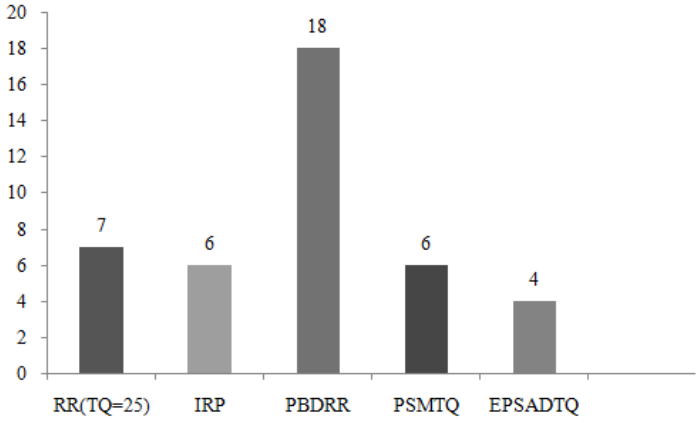

Fig. 1: Comparison of context switches

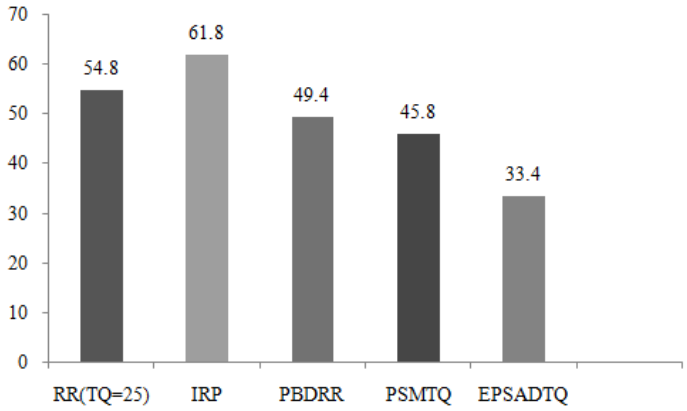

Fig. 2: Comparison of average waiting time

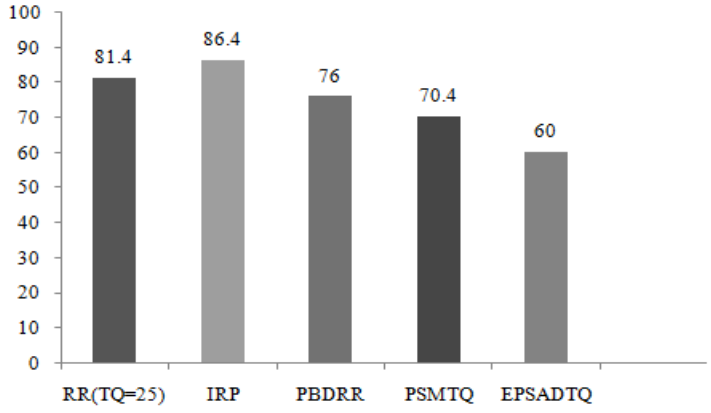

Fig. 3: Comparison of average turnaround time

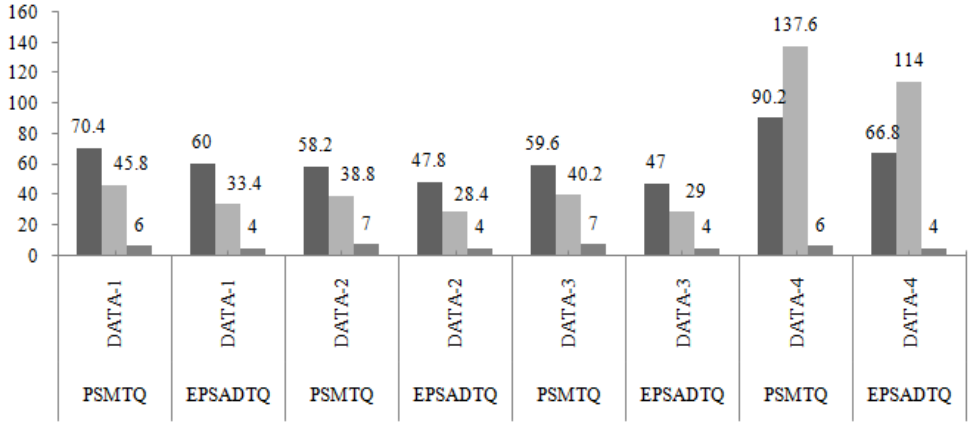

Fig. 4: Comparison of various scheduling algorithms

CONCLUSION AND RECOMMENDATIONS

From the above experimental results, we can say that the proposed algorithm EPSADTQ works better than any other variants of RR and PSMTQ scheduling algorithms in terms of decreasing number of Context switches, average waiting time and average turnaround time. Future work can be based on this algorithm that includes processes Preemptive process mechanism and arrival time for each process.

REFERENCES

Himanshi, S. and A. Prashant, 2012. Design and performance evaluation of Precedence Scheduling Algorithm with Intelligent Service Time (PSIST). Proceeding of 4th International Conference on Computer Modeling and Simulation (ICCMS, 2012).

Noon, A., A. Kalakech and S. Kadry, 2011. A new round robin based scheduling algorithm for operating systems: Dynamic quantum using the mean average. Int. J. Comput. Sci. Issues, 8(3): 224-229.

Siva Nageswara Rao, G., 2014. Comparison of Round Robin CPU scheduling algorithm with various dynamic time quantum. Int. J. Appl. Eng. Res., 9(18): 2749-2760.

Siva Nageswara Rao, G., 2015. A new proposed dynamic dual processor based CPU scheduling algorithm. Int. J. Theor. Appl. Inform. Technol., 73(2): 1991-8645.

Varma, P.S., 2012. Design and performance evaluation of Precedence Scheduling algorithm with Mean Average as Time Quantum (PSMTQ). Int. J. Adv. Res. Comput. Sci. Electron. Eng., 1(7): 248-251.