A Comparative Study of CPU Scheduling Algorithms

Submitted By

Newton
ID:1804090
Sadia Islam Nova
ID:1804091
Rajarshi Sen
ID:1804092
Mirza Asif Razoan
ID:1804093
Mohammad Fayez Ullah
ID:1804094
Fahmida Ahmed
ID:1804095

Submitted To

Md. Shafiul Alam Forhad
Assistant Professor, Dept. of CSE
Chittagong University Of Engineering And Technology

Hasan Murad
Lecturer, Dept. of CSE
Chittagong University Of Engineering And Technology

A Comparative Study of CPU Scheduling Algorithms

Abstract

An operating system's primary function is to manage a group of processes using a technique called CPU scheduling. For multitasking operating systems and time-sharing operating systems Round-robin CPU scheduling algorithm is used. For real-time systems, however, it is not the best option because it results in more context switches, longer wait times, and slower turnaround times. A priority algorithm can resolve the priority levels of processes. This indicates that each process is given a priority, and the processes with the highest priority are carried out first. But in this CPU scheduling algorithm starvation problem arises. That means the processes of lower priority have to wait long. In this project we will work on both preemptive priority and round robin and try to build a scheduling algorithm. This algorithm helps the CPU in overcoming some of the drawbacks of the previous two techniques.

Keywords: CPU Scheduling, Round Robin Scheduling Algorithm, Preemptive Priority Scheduling Algorithm, Turnaround Time, Waiting Time, Gantt Chart

1. Introduction

CPU making a decision on which process will use the CPU exclusively while another is waiting is called scheduling. The basic goal of CPU scheduling is to make sure that the operating system at least chooses one of the processes in the ready queue to be executed whenever the CPU is idle. The CPU scheduler will carry out the process of choosing. It chooses one of the processes in memory that is prepared to run. Preemptive and non-preemptive scheduling are the two different scheduling methods. In preemptive, processes release the CPU before even ending when necessary, however non-preemptive processes don't, which delays the execution of crucial processes. For CPU scheduling, a variety of algorithms are available. The following scheduling criteria are used by these algorithms: CPU utilization, throughput, turnaround time, waiting time, and response time. Each algorithm has a unique way of operating and strives for the optimal outcome given these standards. The main goal of our proposed algorithm is to outperform the current algorithms in terms of scheduling criteria.

2. Related Works

i) First-Come First-Serve Algorithm :

It's a non-preemptive algorithm. In this algorithm, the process which arrives first gets the chance to be executed first. It's the simplest of all algorithms and implementation easier than other algorithms. But it has poor resource utilization as resource utilization in parallel is not possible which leads to convoy effect.

ii) Non-Preemptive SJF Algorithm :

The burst time of processes is used to develop this algorithm. However, it is not realistically possible for the CPU to predict the burst moment. If every process is active at the same moment, the algorithm is optimal. But the shorter process must wait a long time until the present process is completed if the arrival timings are different, which causes starvation.

iii) Preemptive SJF Algorithm :

This algorithm is known as Shortest Remaining Time First(SRTF) algorithm because at any given point of time, the process with the shortest remaining time is executed first. Processes are put into the ready queue as they arrive. When a process with a short burst time comes, the preexisting process is preempted or withdrawn from execution, and the shorter task is completed first. Average waiting time for preemptive SJF scheduling is less than both non preemptive SJF and FCFS scheduling algorithms.

iv) Non-preemptive priority scheduling algorithm:

This algorithm works on the basis of priority of the process where the process with highest priority executed first followed by the ones with lesser priority in order. It also leads to starvation as lower priority processes have to wait longer to get CPU for execution.

v) Preemptive priority scheduling algorithm :

Working of this algorithm is the same as non-preemptive. But if the new process arriving at the ready queue has a higher priority than the currently running process, the CPU is preempted, which means the processing of the current process is stopped and the incoming new process with higher priority gets the CPU for its execution.

In [1] , Hussain Mohammad Abu-Dalbough developed an CPU scheduling algorithm which will work on Priority Preemptive and Round Robin scheduling algorithm's disadvantages.

vi) Round-Robin Algorithm :

It's a preemptive algorithm where a fixed time known as time quantum is allotted to each process for execution. Existing CPU processes have to exit the CPU and move to the ready queue after each time quantum.The most essential factor of this method is that it is starvation-free as every process receives a fair portion of CPU utilization.

An enhanced Round-Robin CPU Scheduling Algorithm was proposed by the researchers. In [2], the (ERR) algorithm aims to improve CPU efficiency by lowering the average waiting time and turnaround time.

In the paper [3] , authors proposed a way of selecting time quantum dynamically which was giving a better CPU performance than typical round robin scheduling algorithm.

3. Proposed Algorithm

Each process is given a priority in the priority algorithm. The highest priority processes are carried out first. However, because the order of the processes and the amount of time they have been waiting for the CPU to execute are ignored, hence starvation might occur. On the other hand, in a round-robin algorithm because of using time quantum starvation can be overcome but performance of the entire system may be impacted by the delayed execution of important processes. Additionally, context switches become larger if the time quantum is too little, and if the time quantum is too large, it behaves like FCFS. Our main target is to overcome these disadvantages and propose an optimal solution for CPU scheduling.That's why we proposed such an algorithm which possesses some of the optimal advantages of both priority and round-robin scheduling algorithms.

Proposed Algorithm's Flowchart:

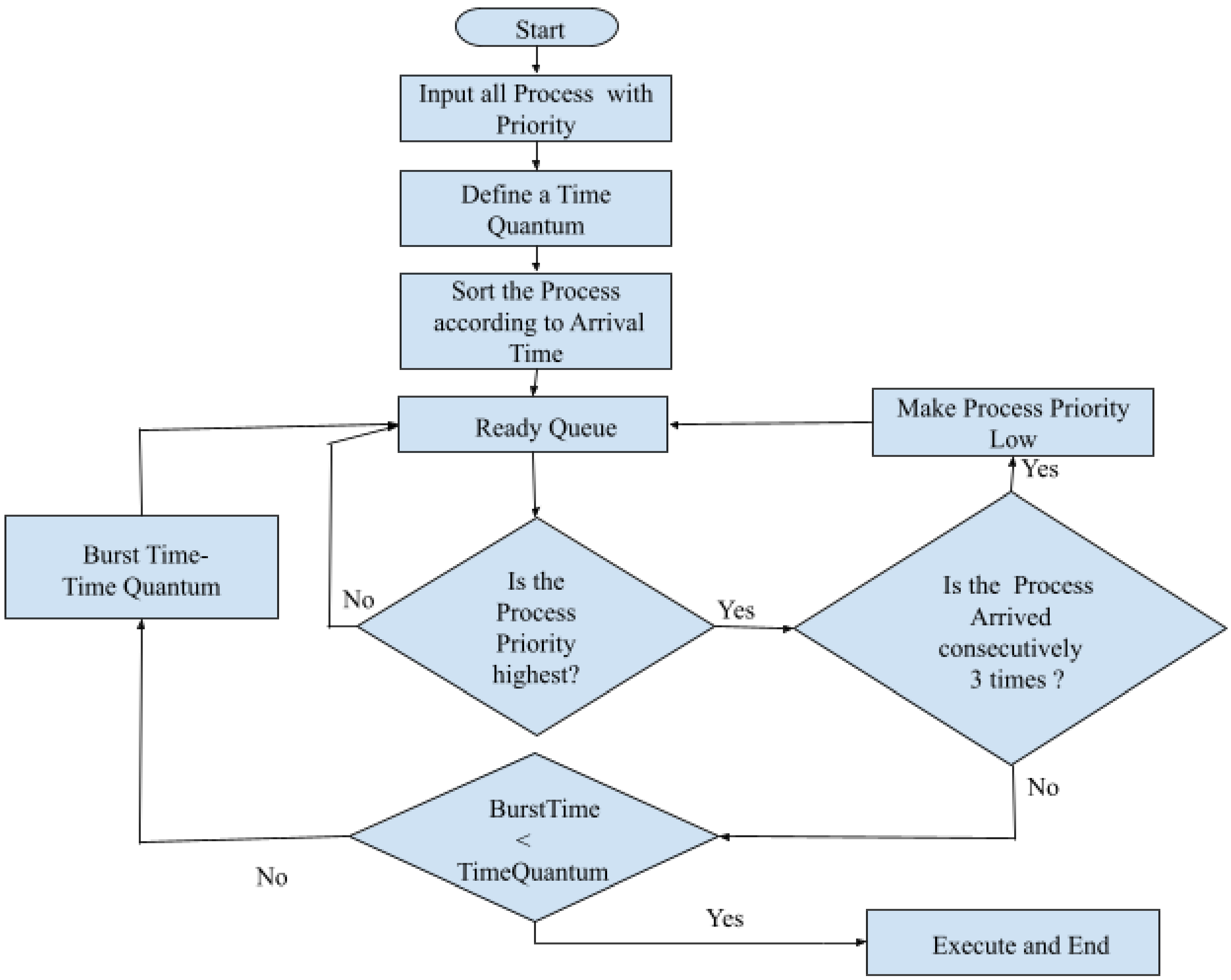


Figure 1: Flow chart of the proposed algorithm

4. Result

In our suggested approach, our main motive was to propose a new algorithm which includes both preemptive priority and round robin scheduling algorithms. For that we have to take a priority column where lower the number means higher priority and like round robin we have taken time quantum value.
Assume that the processes are in the state shown in the table below and time quantum is “ 3 ”.

SAMPLE:

Process	Arrival Time	Burst Time	Priority
P1	0	10	2
P2	1	8	3
P3	4	13	6
P4	5	9	1
P5	6	8	5
P6	7	9	4
P7	7	13	7

Table 1: The arrival, burst time of each process including priority

Now the output of our proposed algorithm is given below.

Gantt Chart:

Created gantt chart using time quantum value ‘3’.

P1	P1	P4	P4	P4	P1	P1	P2	P2	P2	P6	P6
0-3	3-6	6-9	9-12	12-15	15-18	18-19	19-22	22-25	25-27	27-30	30-33

P6	**P5**	**P5**	**P5**	**P3**	**P3**	**P3**	**P7**	**P7**	**P7**	**P3**
33-36	**36-39**	**39-42**	**42-44**	**44-47**	**47-50**	**50-53**	**53-56**	**56-59**	**59-62**	**62-65**

P3	**P7**	**P7**
65-66	**66-69**	**69-70**

Table 2: Gantt Chart

Our Proposed Algorithm Scheduling Algorithm Output Table:

Table 3 basically shows all the required parameters to show the output of the proposed algorithm.

Process	**Finish Time**	**Response Time**	**Waiting Time**	**Turnaround Time**
P1	**19**	**0**	**9**	**19**
P2	**27**	**18**	**18**	**26**
P3	**66**	**40**	**49**	**62**
P4	**15**	**1**	**1**	**10**
P5	**44**	**30**	**30**	**38**
P6	**36**	**20**	**20**	**29**
P7	**70**	**46**	**50**	**63**

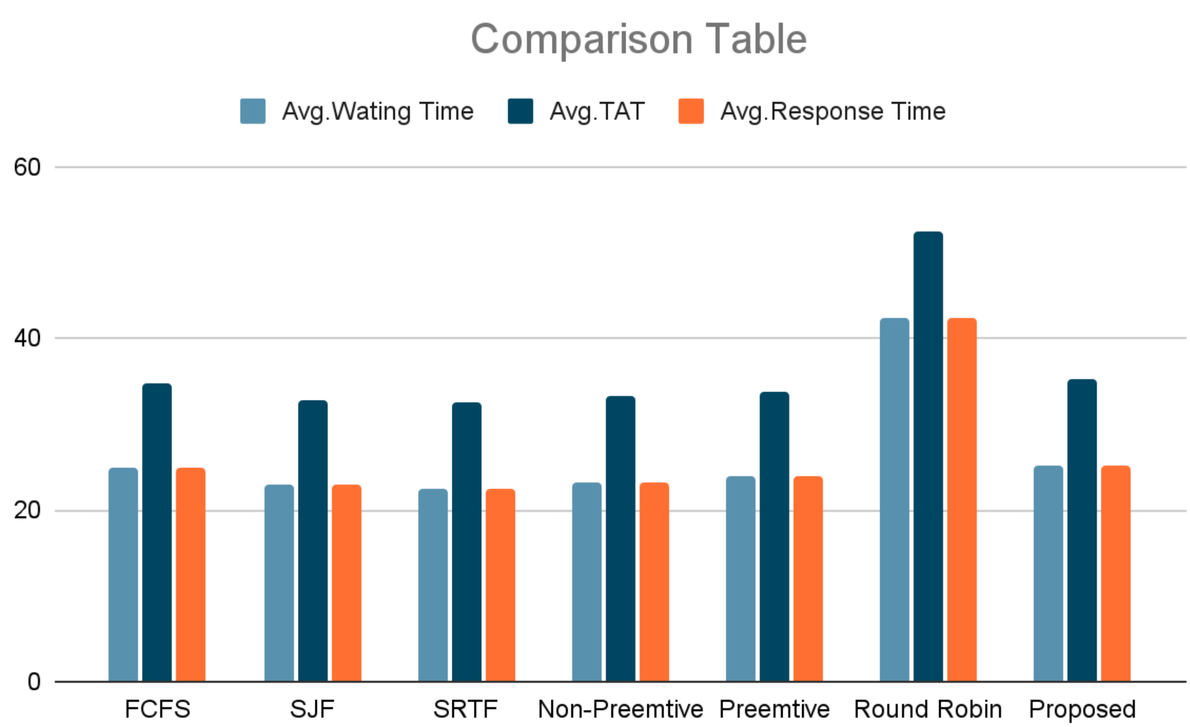
Table 3: Finish time, response time,Waiting time and Turnaround time for our proposed algorithm

Comparison Table of all scheduling Algorithm:

Process	**Avg. Waiting Time**	**Avg. Turnaround Time**	**Avg. Response Time**
FCFS	**24.8571**	**34.8571**	**25.2857**
SJF	**22.8571**	**32.8571**	**23.1429**

SRTF	**22.5714**	**32.5714**	**20.5714**
Non Preemptive Priority	**23.2857**	**33.2857**	**23.2857**
Preemptive Priority	**23.8571**	**33.8571**	**22.5714**
Round Robin	**42.4286**	**52.4286**	**7.7143**
Proposed Algorithm	**25.2857**	**35.2857**	**22.1428**

Using the available dataset, we compared our proposed algorithm to other six scheduling algorithms. The comparison demonstrates that, in terms of turnaround time and average waiting time, our suggested approach performs better than the round robin scheduling algorithm. And response time is less than preemptive priority scheduling.



Conclusion:

We proposed a new algorithm called mix priority and round-robin in this report, after analyzing RR and priority algorithm performances and drawbacks. Our proposed algorithm deals with the drawbacks of simple round-robin and priority algorithms. This helps the CPU avoid permanent blocking or starvation caused by the priority algorithm. It requires more study to be compared to other algorithms. The best scheduling algorithms for various situations should be determined through further research and analysis.

Reference:

[1] H. M. Abu-Dalbouh, “A New Combination Approach to CPU Scheduling based on Priority and Round-Robin Algorithms for Assigning a Priority to a Process and Eliminating Starvation,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 4, pp. 541–546, 2022, doi: 10.14569/IJACSA.2022.0130463.

[2] J. Khatri, “An Enhanced Round Robin CPU Scheduling Algorithm,” *IOSR J. Comput. Eng.*, vol. 18, no. 04, pp. 20–24, Apr. 2016, doi: 10.9790/0661-1804022024.

[3] T. Paul, R. H. Faisal, and M. Samsuddoha, “Improved Round Robin Scheduling Algorithm with Progressive Time Quantum,” *Int. J. Comput. Appl.*, vol. 178, no. 49, pp. 975–8887, 2019.