Grenze International Journal of Engineering and Technology, Special Issue

Improved Priority Scheduling Algorithm

Mr. P V Ramanaiah[1] and Mrs. Chada Sravanthi[2]
[1]Assistant Professor, Methodist College of Engineering & Technology/CSE Department, Hyderabad, India
Email: pvrcseit@gmail.com
[2]Assistant Professor, Methodist College of Engineering & Technology/CSE Department, Hyderabad, India
Email: sravanthireddychada@gmail.com

***Abstract*—The main objective of this paper is to introduce a new CPU scheduling algorithm called PSJF (PRIORITY WITH SHORTEST JOB FIRST) Scheduling Algorithm which is preemptive based on priority with less burst time. This scheduling algorithm tries to improve the average waiting time of priority algorithm in multi programming operating system. CPU Scheduling is the one of main task in operating system. The scheduler is responsible for to select one process among all by scheduling. There are several scheduling algorithms available for scheduling jobs like FCFS, SJF, Priority, Round Robin etc. The proposed algorithm is based on priority with less burst time in scheduling. In this paper, the aftereffects of the current need calculation and shortest job are contrasted and the proposed calculation.**

***Index Terms*— PSJF algorithm, improved priority algorithm, New scheduling algorithm.**

I. INTRODUCTION

Operating system performs various tasks [1-2]. One of the basic tasks is scheduling. Scheduling means select one process among several [1, 3, 4, 5] .whenever we are working with multi core and multi programming systems, we perform multiple tasks at a time, at that moment internally for every task a process will be created. So in multiprogramming system, we can perform multiple tasks at a time, here multiple processes will be created there system has to select one process among all. To select one process we have to use scheduling algorithms.

We have several scheduling algorithms. These algorithms may be preemptive and non-preemptive in nature. Preemptive means the process will not be come out from its execution in the middle. Non-preemptive means process may be suspended in the middle of the execution due to any interrupt, after completion of interrupt again the suspended process enter into the execution state.

The algorithm proposed in this article is preemptive in nature and try to improve the CPU utilization.

Scheduling Parameters:

The basic parameter for a good scheduling algorithm depends on the following measures:-

CPU Utilization: It is the time the processor is busy with executing processes. In multi core system CPU utilization is more.

Throughput: The number of processes the system can execute in a unit of time. The higher the number, the more work is done by the system.

Waiting Time: It is the time; the process is waiting in a ready queue.

Grenze ID: 01.GIJET.4.3.50

Turnaround Time: The time interval of submission of a process to the time of completion.
Response Time: Response time is the time from submission of a request until the first response is produced.
Priority: give preferences to processes with higher priority as high or low priority as high.

II. EXISTING CPU SCHEDULING ALGORITHMS OVERVIEW

Here we are taking two sample process tables to conduct the tests against existing scheduling algorithms with our proposed one. We have two samples with four different processes with following properties:

SAMPLE 1:

TABLE 1: PROCESS TABLE – 1

Process Name	Burst Time	Priority
P1	6	1
P2	4	1
P3	5	3
P4	7	2

SAMPLE 2:

TABLE 2 : PROCESS TABLE - 2

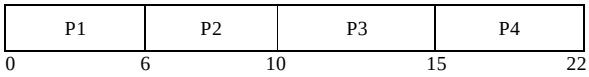
Process Name	Burst Time	Priority
Task1	8	2
Task2	3	1
Task3	7	1
Task4	5	2

Let us calculate and compare the average waiting time for above sample process tables 1 & 2.

A. First Come First Serve (FCFS) Scheduling

The FCFS is simple CPU Scheduling algorithm [1, 5, 6, 7, 9, and 10]. The criteria of this algorithm is, "the process that requests first, holds the CPU first" or the process which enters the ready queue first is execute by processor first. The selection of a process is based on f arrival time. Once a process has been given to the CPU, it runs into completion without being interrupted by any other process. The disadvantage of this algorithm is for long jobs or more burst time tasks other process has to wait long time for their turn. It makes the system as low CPU utilization.
The Gantt chart for process table -1 (ref. table 1) using FCFS Scheduling is as follows:

TABLE III: GANTT CHART FOR FCFS FOR SAMPLE – 1



So the waiting time of each process from table-3 is:-
P1 = 0; P2 = 6; P3 = 10; P4=22;

The average waiting time is = (0 + 6 + 10 + 15) / 4 = 7.75

The Gantt chart for process table -2 using FCFS Scheduling is as follows:

TABLE IV: GANTT CHART FOR FCFS FOR SAMPLE – 2

So the waiting time of each process from above table is:-
P1 = 0; P2 = 8; P3 = 11; P4=18;

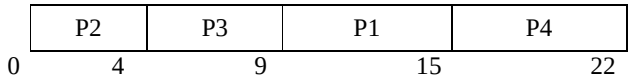
The average waiting time is = (0 + 8 + 11 + 18) / 4 = 9.25

B. Shortest Job First (SJF) Scheduling

The SJF algorithm selects a process which is having the smallest CPU burst time among all without considering arrival time [1, 5, 8].. If two processes have the same CPU burst time FCFS is used to break up

the tie. SJF can be preemptive and non– preemptive in nature based on the arrival time and burst time of the processes. SJF reduces average waiting time of the processes as compared to FCFS.
The Gantt chart for process table-1 (ref. table 1) using SJF scheduling is:

TABLE V: GANTT CHART FOR SJF FOR SAMPLE -1

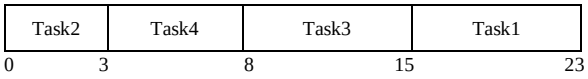


So the waiting time of each process is:-
P1 = 9; P2 = 0; P3 = 4; P4=15;

The average waiting time is = (9 + 0 + 4 + 15) / 3 = 7

The Gantt chart for process table -2 (ref. table 2) using SJF scheduling is:

TABLE VI: GANTT CHART FOR SJF FOR SAMPLE – 2



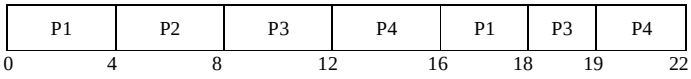
So the waiting time of each process is:-
Task1 = 23; Task2 = 0; Task3 = 15; Task4=8;

The average waiting time is = (23 + 0 + 15 + 8) / 3 = 11.5

C. Round Robin (RR) Scheduling

It is a preemptive scheduling algorithm. In this algorithm, a small unit of time called time quantum or time slice is assigned to each process. When the time quantum expired, the process is preempted from execution, sent to ready queue and another process comes for execution and so on. Like this CPU is switched from one process to another process based on time slice called as context switching. Performance of Round Robin relies on the size of the time quantum.
The Gantt chart for process table -1 (ref. table 1) using R.R scheduling with a time quantum – 4 is:

TABLE VII: GANTT CHART FOR ROUND ROBIN FOR SAMPLE – 1

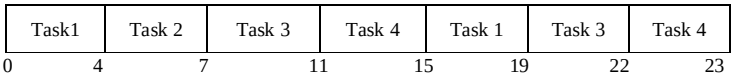


The waiting time of each process is:-
P1 = 0+ (16-4) = 12; P2 = 4 = 4; P3 = 8+ (18-12) = 14; P4=12+ (19-16) = 15;

The average waiting time is = (12 + 4+ 14+15) / 4 = 11.25

The Gantt chart for process table -2 (ref. table 2) using R.R scheduling with a time quantum – 4 is:

TABLE VIII: GANTT CHART FOR ROUND ROBIN FOR SAMPLE – 2



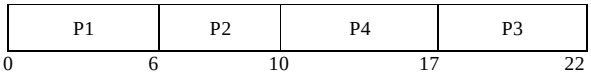
The waiting time of each process is:-
Task1 = 0+ (15-4) = 11; Task2 = 4 = 4; Task3 = 7+ (19-11) = 15; Task4=11+ (22-15) = 18;
The average waiting time is = (11 + 4+ 15+18) / 4 = 12

D. Priority Based Scheduling

In this algorithm, priority is assigned to each process and based on that priority CPU is allocated to the processes. Higher priority processes are executed first and lower priority processes are executed at the end. If

multiple processes having the same priorities, then selection is based on FCFS scheduling. Priority Scheduling can be preemptive and non-preemptive in nature.
The Gantt chart for process table -1 (ref. table 1) using Priority scheduling is:

TABLE IX: GANTT CHART FOR PRIORITY I.E., PFCFS SCHEDULING FOR SAMPLE – 1

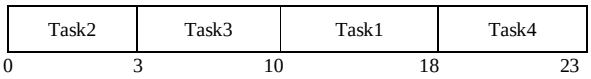


The waiting time of each process is:-
P1 = 0; P2 = 6; P3 = 17; P4 = 10;

The average waiting time is = (0 + 6+ 17+10) / 4 = 8.25

The Gantt chart for process table -2 (ref. table 2) using Regular Priority scheduling (PFCFS) is:

TABLE X: GANTT CHART FOR PRIORITY (PCFS) SCHEDULING FOR SAMPLE – 2



So the waiting time of each process from the above table is:-
Task1 = 10; Task 2 = 0; Task 3 = 3; Task 4 = 18;

The average waiting time is = (10 + 0+ 3+18) / 4 = 7.75

III. PROPOSED WORK: PSJF CPU SCHEDULING ALGORITHM

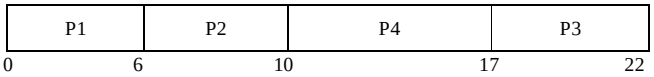
The proposed algorithm PSJF CPU Scheduling algorithm is preemptive in nature and based on Priority and shortest job first scheduling mechanisms. In this algorithm we will try to improve CPU utilization. In this research we showed the comparison between traditional scheduling techniques to proposed algorithm.

A. In proposed algorithm the following steps are going to take:

Step -1: Select the sequence of process according to the priority first.
Step -2: On the off chance that numerous processes have same need, select the process dependent on burst time instead FCFS as in practice. Means the process having minimum burst time should be select first then second one.
When we arrange the processes into increasing order to their burst time, the average waiting time is going to improve which is shown in bellow.

In the first sample processes: P1 and P2 have same priority, this tie will be cleared by using FCFS along with priority. By using existing priority algorithm the process will execute process in following order i.e., P1, P2, P4, P3.
The Gantt chart for Existing Priority Scheduling (PFCFS) for Sample -1 (ref. table 1) is:

TABLE XI: GANTT CHART FOR EXISTING PRIORITY ALGORITHM FOR SAMPLE -1



So the waiting time of each process is:-
P1 = 4; P2= 0; P3 =17; P4=10;
The average waiting time is = (0 + 6 + 10 + 17) / 4 = 8.25 (1.1)

By using proposed algorithm the same priority process will be selected based on less burst time. That means low burst time process will execute first. By using proposed scheduling algorithm, the execution order of the processes is: P2, P1, P4 and P3.
The Gantt chart for Proposed Priority Scheduling (PSJF) for Sample -1 (ref. table 1) is:

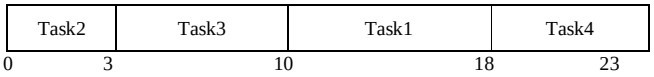
TABLE XII: GANTT CHART FOR PRIORITY SHORTEST JOB FIRST ALGORITHM FOR SAMPLE -1

From the table-12, the waiting time of each process is:-
P1 = 4; P2= 0; P3 =17; P4=10;

The average waiting time is = (4+ 0 + 10+17) / 4 = 7.75 (1.2)

The same test taken for Sample Process Table -2 (ref. table 2) and the results are as follows:

TABLE XIII: GANTT CHART FOR PFCFS FOR SAMPLE -2

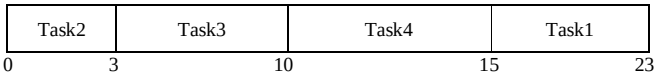


The waiting time for each process is:-
Task1 = 10; Task2 = 0; Task3 = 3; Task4 = 18;

Then the average waiting time = (10 + 0 + 3 + 18) / 4 = 7.75 (2.1)

Gantt chart for proposed algorithm is shown in table -14.

TABLE XIV: GANTT CHART FOR PSJF ALGORITHM FOR SAMPLE -2



So the waiting time for each process for the sample Process Table -2 is:-
Task1 = 15; Task2 = 0; Task3 = 3; Task4 = 10;
Then the average waiting time = (15 + 0 + 3 + 10) / 4 = 7 (2.2)

IV. CONCLUSION

This paper presents a new CPU scheduling algorithm called PSJF CPU Scheduling Algorithm which is based on shortest job first and priority algorithms. It has comparison of proposed algorithm with existing algorithms which can be observed through (1.1),(1.2) for Sample Process Table -1 and (2.1),(2.2) for Sample Process Table - 2 . The comparison was done by multiple process using different scheduling algorithms .The Paper also explains how proposed algorithm improves the performance of traditional one.

REFERENCES

[1] Suranauwarat, S.: A CPU Scheduling Algorithm Simulator. In: WI 37th ASEE/IEEE Frontiers in Education Conference, Milwaukee, October 10-13 (2007)Google Scholar
[2] Sindhu, M., Rajkamal, R., Vigneshwaran, P.: An Optimum Multilevel CPU Scheduling Algorithm. In: 2010 International Conference on Advances in Computer Engineering, pp. 90–94 (2010)Google Scholar
[3] Silberschatz, A., Galvin, P., Gagne, G.: Operating System Concepts, 7th edn. John Wiley & Sons, Chichester (2005)MATHGoogle Scholar
[4] Nutt, G.: Operating System, 3rd edn. Addison Wesley, Reading (2004)Google Scholar
[5] Tang, X., Lian, H., Zhe, F., Tang, Z.: Computer Operating System, 3rd edn. Xian Electronic Science &Technology University Press, Xian (2007) (in Chinese)Google Scholar
[6] Hassin, R.: Equilibrium customers' choice between FCFS and random servers Queuing Syst. 62, 243–254 (2009)MathSciNetCrossRefMATHGoogle Scholar
[7] Cheng, S., Zhang, L.: The Computation of response time of first Come First Serve Scheduling Algorithm. Joumal of Henan Institute of Education (Natural Science) 15(3), 20–22 (2006) (in Chinese) Google Scholar

[8] Chan, W.-T., Lam, T.-W., Liu, K.-S., Wong, P.W.H.: New resource augmentation analysis of the total stretch of SRPT and SJF in multiprocessor scheduling. Theoretical Computer Science 359,430-439 (2006) MathSciNetCrossRefMATHGoogle Scholar.
[9] http://en.wikipedia.org/wiki/Scheduling.
[10] Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, “Operating System Concepts”, SixthEdition.