A NEW TECHNIQUE FOR CPU SCHEDULING: STANDARD DEVIATION BASED

[1]Dr. Kuldeep Singh Kaswan, [2] Amandeep

[1]Asstt. Prof., Department of Computer Science & Engineering
PDM College of Engg., Bahadurgarh (Haryana)

[2]Department of Computer Science & Engineering
PDM College of Engg., Bahadurgarh (Haryana)

Abstract- Scheduling is the main function of an operating system. A number of techniques are there for the scheduling and each technique has its own significance, but no one is best in all circumstances. In this paper, we discussed about different exisitng CPU scheduling algorithms. We also proposed a new CPU scheduling algorithm, based on standard deviation approach. In the last we compared our proposed algorithm results with the existing algorithms.

***Index Terms*: Scheduling, Round Robin, Mixed Mode, SJF.**

I INTRODUCTION

The fundamental and core function of an operating system is Scheduling [1]. Scheduling is the method by which work specified by some means is assigned to resources that complete the work. It allows computer resources to be shared in between multiple processes as it contains decisions of giving resources. It mainly targets to achieve fairness, low overhead, efficient use of processor time and avoids processes not to face the problem of starvation.

One of the primary computer resource is CPU and its Scheduling is very important to an operating system design. CPU Scheduling is the main task of an operating system as it determines which process to use the CPU and which process to wait for the CPU [3]. The aim of CPU Scheduling is not only to have the proper utilization of resource but enhancing the overall performance of the system by making the system more efficient, fast and fair.

There are various CPU Scheduling algorithms used for performing scheduling and the performance of these CPU Scheduling algorithms are measured on the behalf of some parameters like CPU Utilization, Throughput, Waiting Time, Turn Around Time, etc.

II CPU SCHEDULING ALGORITHM

i. First Come First Serve Scheduling

First Come First Serve Scheduling is also termed as First-In-First-Out, that is, allocates the CPU in order in which process arrive to the ready queue and ready queue is maintained as a FIFO queue. The entry of a new process takes place through the tail of the queue and exit takes place through the head of the queue [4].

First Come First Serve is easy to understand and implement. It is suitable for the batch system but not suitable for time sharing system. This scheduling is fair for the smaller burst time but unfair for the larger burst time processes [6]. Below is the Table 1 which has the process ID and the corresponding burst time.

Table 1: PID and Burst Time of Processes

Process ID	**Burst Time (ms)**
P_0	4
P_1	15
P_2	31
P_3	6

P_0	P_1	P_2	P_3

0 4 19 50 56

Figure 1: Gantt chart for FCFS

As shown in Gantt chart for FCFS, processes are excuted according to their entry in the ready queue. Thus sequence of the execution is P0, P1, P2 and P3.

Table 2: Calculation of Waiting and Turnaround time for FCFS

Job Sequence	Process ID	Burst Time (ms)	Waiting Time	Turnaround Time
1	P_0	4	0	4
2	P_1	15	4	19
3	P_2	31	19	50
4	P_3	6	50	56

Average Waiting Time = (0+4+19+50) / 4 = 18.25

Average Turnaround Time = (4+19+50+56)/4 =32.25

ii. Shortest Job First Scheduling

In the Shortest Job First Scheduling, ready queue is treated as a priority queue based on the smallest CPU burst requirement. Process with the smallest CPU burst is executed first than those processes whose CPU burst is larger. There may be a situation when the CPU burst is same of two processes then First Come First Serve is used to resolve the situation [5].

When a process with a shorter CPU burst is ready for the execution then the running process will be interrupted in between its execution and it is called preemptive SJF and when the running process is not interrupted in its execution then it is called Non-Preemptive SJF [10].

Shortest Job First plays a significant role in minimizing the average waiting time but it can lead to unfairness and starvation.

Figure 3 shows the Gantt chart for SJF.

P_0	P_3	P_1	P_2

0 4 10 25 56

Figure 2: Gantt chart for SJF

As shown in Gantt chart for SJF process has lowest CPU burst will run first and so on. Thus sequence of the execution is $P_{0,}$ P_3, P_1 and $P_{2.}$

Table 3: Calculation of waiting and turnaround time for SJF

Job Sequence	Process ID	Burst Time (ms)	Waiting Time	Turnaround Time
1	P_0	4	0	4
2	P_3	6	4	10
3	P_1	15	10	25
4	P_2	31	25	56

Average Waiting Time = (0+4+10+25) / 4 = 9.75

Average Turnaround Time = (4+10+25+56) / 4 = 23.75

iii. Round Robin Scheduling

Round Robin Scheduling assigns time slices to each process in a circular order for the execution of the processes. It is similar to the first Come First Serve, except for the time slices which is assigned for each process [2]. The first process in the queue is allowed to execute until the time quantum is expired and the same criteria for all the processes in the queue. But if the process requires more than a time slice then the process will run for the full length of the time slice and then the process is preempted. In this, context switching is used to save states of preempted processes [8].

It is simple and easy to implement and best suitable for the timesharing system. It provides every process an equal share for holding CPU and overcomes the problem of starvation. But giving equal share of the CPU is always not beneficial as CPU-bound processes schedule more frequently than highly interactive processes and also average waiting time may be not good. Take time qantum = 4

P_0	P_1	P_2	P_3	P_1	P_2	P_3	P_1	P_2	P_1	P_2	P_2	P_2	P_2	P_2

0 4 8 12 16 20 24 26 30 34 37 41 45 49 53 56

Figure 3: Gantt chart for Round Robin

The sequence of execution is $P_0, P_1, P_2, P_3, P_1, P_2, P_3, P_1, P_2, P_1, P_2, P_2, P_2, P_2, P_2$

Table 4: Calculation of waiting and turnaround time for RR

Job Sequence	Process ID	Burst Time (ms)	Waiting Time	Turnaround Time
1	P_0	4	0	4
2	P_1	15	22	37
3	P_2	31	25	56
4	P_3	6	20	26

Average Waiting Time = (0+22+25+20) / 4 = 16.75

Average Turnaround Time = (4+37+56+26) / 4=30.75

iv. Job Mix Scheduling

In Job Mix scheduling, a separate queue is maintained from the queue in which processes are kept in the order of lesser burst time first and then the higher burst time [7]. All the processes are ordered in the same way and then the execution takes place.

This method helps to eliminate the problem of starvation for the longer jobs and it provides better average waiting time which results in better performance of CPU [9]. It is simple and easy to implement but there is increase in overhead in maintaining separate queue.

As shown in Gantt chart for Job Mix scheduling. Thus sequence of execution is P_0, P_2, P_3 and P_1.

P_0	P_2	P_3	P_1

0 4 35 41 56

Figure 4: Gantt chart for Job Mixing

The average waiting Time and Average Turnaround Time for Job Mix scheduling .

Table 5: Calculation of waiting and turnaround time for Job Mix

Job Sequence	Process ID	Burst Time (ms)	Waiting Time	Turnaround Time
1	P_0	4	0	4
2	P_2	31	4	35
3	P_3	6	35	41
4	P_1	15	41	56

Average Waiting Time = (0+4+35+41) / 4 = 20

Average Turnaround Time = (4+35+41+56) / 4 = 34

III PROPOSED ALGORITHM

SD Scheduling

In proposed SD scheduling, a queue is maintained in which processes are ordered on the basis of how close is the value of their burst time with the value of the standard deviation. Process whose burst time is close to the standard deviation value will be executed first and so on.

Standard Deviation is given by:

$$SD = \sqrt{\frac{\sum(X - \bar{X})^2}{N}}$$

Where: $\bar{X}$ is the mean value.

X is the burst time of each process.

N is the number of the processes.

After evaluation, Standard Deviation value is 10.653 and thus sequence of the execution is P_1, P_3, P_0, and P_2

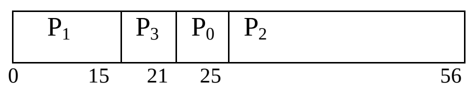

Figure 5: Gantt chart for SD

The average waiting Time and Average Turnaround Time for SD is calculated according to the Table.

Table 6: Calculation of waiting and turnaround time for SD

Job Sequence	Process ID	Burst Time (ms)	Waiting Time	Turnaround Time
1	P_1	15	0	15
2	P_3	6	15	21
3	P_0	4	21	25
4	P_2	31	25	56

Average Waiting Time = (0+15+21+25 / 4 = 15.25

Average Turnaround Time = (15+21+25+56) / 4 = 29.25

IV COMPARISON

In this table we compared the different existing CPU scheduling techniques with the proposed SD based technique. In this comparison, we select the FCFS, SJF, RR, and Job mixing techniques and then find the average waiting time and the average turnaround time.

Table 7: Comparison of waiting time and turnaround time for different CPU scheduling algorithms

	Waiting Time					Turnaround Time				
	FCFS	SJF	RR	Job Mixing	SD	FCFS	SJF	RR	Job Mixing	SD
P_0	0	0	0	0	21	4	4	4	4	25
P_1	4	10	22	41	0	19	25	37	56	15
P_2	19	25	25	4	25	50	56	56	35	56
P_3	50	4	20	35	15	56	10	26	41	21
Average	18.25	9.75	16.75	20	15.25	32.25	23.75	30.75	34	29.25

This table reveals that our purposed algorithm gave better performance than FCCS, RR and Job mixing in terms of waiting time and turnaround. But the proposed algorithm could not compete the SJF algorithm.

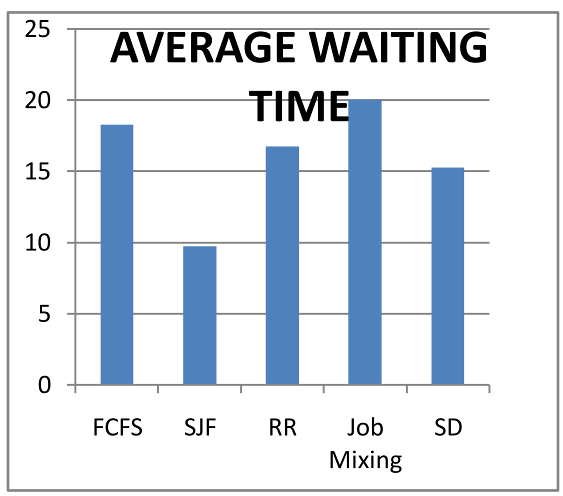

Figure 6: Comparison of Average Waiting Time for different CPU Scheduling Algorithm

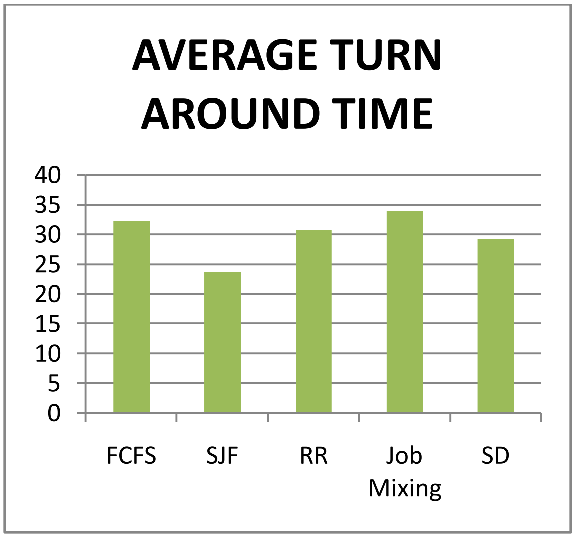

Figure 7: Comparison of Average Turn around Time for different CPU Scheduling Algorithm

V. CONCLUSION

This paper presents a comparative study of various CPU Scheduling Algorithm. From this, it is concluded that the results of the SD based CPU scheduling algorithm is better than FCFS, RR and Job mixing but not than the SJF. This scheduling gives better results than previous researches. This scheduling is better in minimizing the average waiting time and average turnaround time which leads to better performance.

REFERENCES

1. Pushpraj Singh, Vinod Singh, Anjani Pandey," Analysis and Comparison of CPU Scheduling Algorithms", International Journal of Emerging Technology and Advanced Engineering Volume 4, Issue 1, Jnauary 2014).
2. Mona Saini, "Designing Various CPU Scheduling Techniques using SCILAB", International Journal of Computer Science and Information Technologies, Vol. 5(3), 2014, 2918-2923.
3. Maria Ulfah Siregar,'' A New Approach to CPU Scheduling Algorithm: Genetic Round Robin", International Journal of Computer Applications (0975-888) Volume 47-No. 19, June 2012.
4. Neetu Goel and Dr. R.B. Garg," A Comparative Study of CPU Scheduling Algorithms", International Journal of Graphics and Image Processing Volume 2 Issue 4 November 2012.
5. Miss. Jayashree S. Somani, Miss. Pooja K. Chhatwani," Comparative Study of Different CPU Scheduling Algorithms", International Journal of Computer Science and Mobile Computing Volume 2, issue. 11, November 2013, pg.-310-318.
6. Sukumar Babu Bandarupalli, Neelima Priyanka Nutulapati, Prof. Dr. P.Suresh Varma, "A Novel CPU Scheduling Algorithm-Preemptive and Non-Preemptive," International Journal of Modern Engineering Research (IJMER) Vol. 2, Issue 6, Nov-Dec. 2012 pp-4484-4490..
7. Charu Rani, Mrs. Manju Bala, "Comparison of Round Robin Based CPU Scheduling Algorithms", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 9, September 2015.
8. Neeraj Kumar, Nirvikar, "Performance Improvement Using CPU Scheduling Algorithm-SRT", International Journal of Emerging Trends and Technology in Computer Science (IJETICS) Volume 2, Issue 2, March – April 2013.
9. Alka Pant, "A Comparison between FCFS and Mixed Scheduling", International Journal of Computer Science and Technology Volume 2, Issue 2, June 2011.
10. Himani Aggarwal, Er. Shakti Nagpal, "Comparative Performance Study of CPU Scheduling Algorithms", International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, Issue 6, June 2014.