ISSN 2277 – 5528
Impact Factor- 4.015

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES &MANAGEMENT

A REVIEW ON CPU SCHEDULING TECHNIQUES

Atal Anand*, Priya Vats, Deepanshu, Yogesh Kumar
B.Tech Scholar, Ganga Technical Campus, Bahadurgarh - HR, India
Department of Computer Science & Engineering, Ganga Technical Campus, Bahadurgarh - HR, India

ABSTRACT

CPU Scheduling is a key concept in designing of multitasking, multiprocessing and real-time operating system. Scheduling refers to the way of processes are assigned to run on the available CPUs, since there are typically many processes running than there are available CPUs. CPU scheduling deals with the problem of deciding which of the processes in the ready queue is to be allocated the CPU. In this paper, we have analyzed existing CPU Scheduling Techniques. In the first section, we introduced about CPU Scheduling. In the Second section, we discussed about different existing CPU Scheduling algorithm. In last, we compared the waiting time and turnaround time of all existing algorithm and provide a comparative result.

Keywords- *CPU Scheduling, Throughput, Turnaround Time, Waiting Time*.

INTRODUCTION

Operating system is an interface between user and machine. Operating system is also called a Resource Manager because it manages all machine resources and allocate them to a specific program and use as it require to complete its task. One of the important functions of operating system is Process Management. Process Management function allocates the processor to execute a chosen process. A process is a program at the time of its execution. A process is required to complete its task – files, memory, I/O devices and CPU Time. A CPU with Single processor executes only one process at a time [1]. CPU scheduler selects a process from the ready queue submitted to the system for execution and decides when a process is to be executed in the case of multiprogramming. CPU scheduling allows one process to use the CPU while the execution of another process is on hold due to unavailability of any resource, thereby making full use of CPU. CPU Scheduling is to make the system efficient, fast and fair [2].

There are two types of CPU Scheduling Techniques- Nonpreemptive and Preemptive scheduling. When a process is waiting for input/output or the process terminated then scheduling scheme is Nonpreemptive. When a process switches into ready state from running or waiting state by occurrence of interrupt or by completion of input/output then scheduling scheme is Preemptive.

There exist a number of CPU Scheduling algorithms like FCFS, SJF, Round Robin, Priority, EDRR and CAT Next. Many criteria have been suggested for comparison of CPU scheduling algorithms like CPU utilization, Waiting Time, Turnaround Time, Throughput and Response Time [6]. Fairness can be reflected by treating all the processes same and no process should be suffer infinite postponement. A given Process should run in about the same amount of time and at the same cost irrespective of the load on the system. A certain portion of system resources invested as overhead can greatly improve overall performance of the system. Scheduling mechanism should keep the resources of the system busy. Scheduling mechanism should favour the high priority processes. For an effective use of CPU, Utilization of CPU should be high and CPU should not remain idle frequently to make the system efficient, fast and fair. Waiting time is the sum of period spent by a process in ready queue and the turnaround time of a process is the total amount of time taken from its submission to termination, both of these should be low.

This paper is further divided as, in section II we discussed about different existing Algorithms. We compare the waiting time and turnaround time of existing CPU Scheduling Algorithms in Section III. In section IV, we provide a comparative result to show which technique is best for effective utilization of CPU.

EXISTING CPU SCHEDULING ALGORITHM

As we discussed in section I that there is various CPU Scheduling Algorithms and we also discussed that an efficient CPU scheduling algorithm focuses on high CPU utilization and throughput. Fig. 1 shows several of the many CPU scheduling algorithms that exist:

ISSN 2277 – 5528
Impact Factor- 4.015

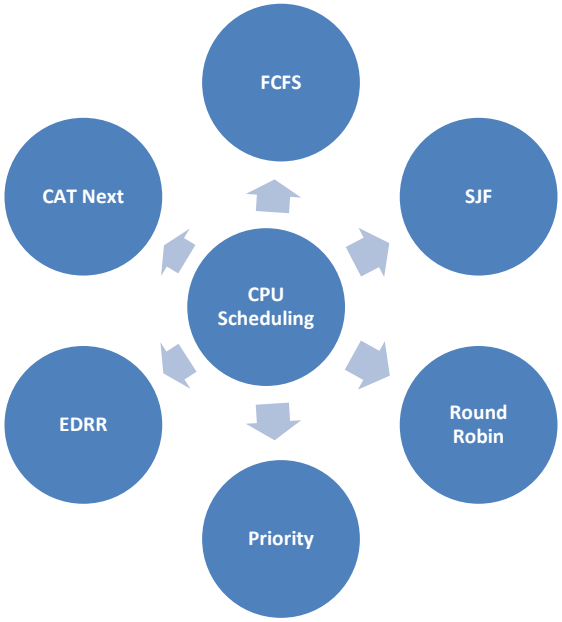

Fig. 1: various CPU scheduling algorithms

First come first serve scheduling

FCFS Scheduling is managed on the phenomena of Queue i.e. First in First out (FIFO) [5]. In FCFS allocation of CPU is in the order in which the processes arrive. CPU will allocate to the process which entered first in the ready queue. First entered process will have higher priority of CPU allocation for execution and so on. It is simplest of all technique of CPU Scheduling; once the CPU has been given to a process, that process keeps the CPU until it releases the CPU either by terminating or by requesting I/O devices. Table 1 has the Process ID, Burst Time, Arrival Time and Priority.

Table 1: PID, Burst Time, Arrival Time and priority of Processes

Process ID	Burst Time (ms)	Arrival Time (ms)	Priority
P_0	15	0	4
P_1	80	1	2
P_2	5	2	1
P_3	160	3	3

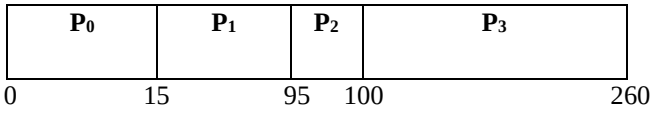

Fig. 2: Gantt chart for FCFS

Fig 2 shows Gantt chart for FCFS, process executes as per arrival. Thus execution sequence is P_0, P_1, P_2 and P_3.

Table 2: Calculation of Waiting and Turnaround Time for FCFS

Job Sequence	Process ID	Burst Time (ms)	Waiting Time	Turnaround Time
1	P_0	15	0	15
2	P_1	80	15	95

ISSN 2277 – 5528
Impact Factor- 4.015

3	P_2	5	95	100
4	P_3	160	100	260

Average Waiting Time = (0+15+95+100) / 4 = 52.5 ms
Average Turnaround Time = (15+95+100+260)/4 =117.5 ms

Shortest job first scheduling
CPU is allocated to the process with least CPU burst time [4]. The SJF associates with each process the length of the letter's next CPU burst. When the CPU is available, it is assigned to the process that has the smallest next CPU burst time. If the two processes have the same length of CPU burst time then FCFS scheduling algorithm is followed. This algorithm is considered to be an optimal algorithm, as it gives the minimum average waiting time as a result.

The SJF algorithm may be either Preemptive or non-Preemptive.

Preemptive SJF Scheduling
A Preemptive SJF will preempt this currently executing process and starts the execution of newly entered process if the newly process have smaller burst time as compared to currently executing process.
Fig. 3 shows the Gantt chart for Preemptive SJF as below:

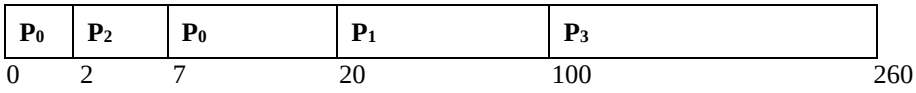

Fig. 3: Gantt chart for Preemptive SJF

As shown in Fig. 3 for Preemptive SJF process has lowest CPU burst will run first and so on.

Table 3: Calculation of Waiting and Turnaround Time for Preemptive SJF

Job Sequence	Process ID	Burst Time (ms)	Waiting Time	Turnaround Time
1	P_0	15	5	20
2	P_1	80	19	99
3	P_2	5	0	5
4	P_3	160	97	257

Average Waiting Time = (5+19+0+97) / 4 = 30.25 ms
Average Turnaround Time = (20+99+5+257) / 4 = 95.25 ms

Non-Preemptive SJF Scheduling
A Non-Preemptive SJF will allow the currently executing process to complete its burst time without any interruption in its execution.
Fig. 4 shows the Gantt chart for SJF as below:

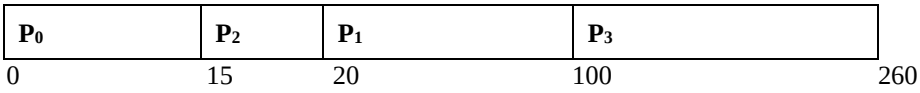

Fig. 4: Gantt chart for Non-Preemptive SJF

As shown in Fig. 4 for Non-Preemptive SJF process has lowest CPU burst will run first and so on.

Table 4: Calculation of Waiting and Turnaround Time for Non-Preemptive SJF

ISSN 2277 – 5528
Impact Factor- 4.015

Job Sequence	Process ID	Burst Time (ms)	Waiting Time	Turnaround Time
1	P_0	15	0	15
2	P_1	80	19	99
3	P_2	5	13	18
4	P_3	160	99	257

Average Waiting Time = (0+19+13+99) / 4 = 32.75 ms
Average Turnaround Time = (15+99+18+257) / 4 = 97.25 ms

Round Robin scheduling

Round Robin scheduling algorithm is designed especially for time-sharing systems [1]. It is similar to FCFS scheduling algorithm, but preemption is added to switch between processes. A small unit of time called a quantum or time-slice (t_q) is defined where range is kept between 10ms to 100ms. CPU is allotted to the each process in the ready queue as per time quantum (t_q) [3]. Round Robin algorithm decrease the response time of CPU. It follows the FCFS technique. Selection of time quantum is more important to implement this Algorithm.
Fig. 5 shows the Gantt chart for RR where $t_q = 15$.

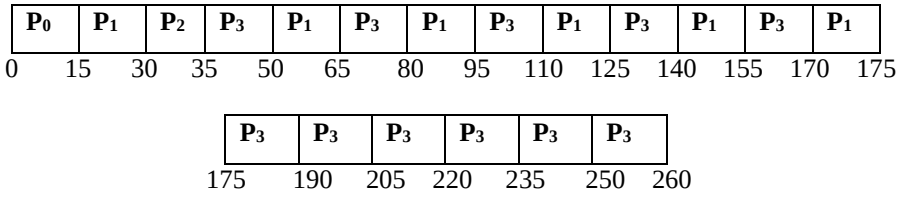

Fig. 5: Gantt chart for Round Robin

Table 5: Calculation of Waiting and Turnaround Time for RR

Job Sequence	Process ID	Burst Time (ms)	Waiting Time	Turnaround Time
1	P_0	15	0	15
2	P_1	80	94	175
3	P_2	5	28	33
4	P_3	160	97	257

Average Waiting Time = (0+94+28+97) / 4 = 54.75 ms
Average Turnaround Time = (15+175+33+257) / 4 = 120 ms

Priority scheduling

An SJF is simply a priority algorithm where the priority is the inverse of the predicted next CPU burst. The larger the CPU burst, the lower the priority and vice-versa [6]. Priorities are generally some fixed range of numbers such as 0 to 7. However, there is no general agreement on whether 0 is the highest or lowest priority.

Priority Scheduling is either Preemptive or Non-Preemptive.

Preemptive Priority Scheduling

A Preemptive priority scheduling algorithm will preempt the CPU if the priority of the newly arrived process is higher than the priority of the process currently running in the system.
Gantt chart in Fig. 6 shows preemptive priority scheduling, process with lowest numeric value will run first and so on.

ISSN 2277 – 5528
Impact Factor- 4.015

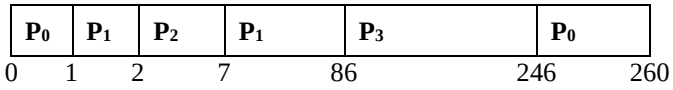

Fig. 6: Gantt chart for Preemptive Priority Scheduling

Table 6: Calculation of Waiting and Turnaround Time for Preemptive PS

Job Sequence	Process ID	Burst Time (ms)	Waiting Time	Turnaround Time
1	P_0	15	245	260
2	P_1	80	5	85
3	P_2	5	0	5
4	P_3	160	83	243

Average Waiting Time = (245+5+0+83) / 4 = 83.25 ms
Average Turnaround Time = (260+85+5+243) / 4 = 148.25 ms

Non-Preemptive Priority Scheduling

A Non-Preemptive priority scheduling algorithm will simply put the new process at the head of ready queue i.e., it does not preempt the execution of current process that is being in execution even when the priority of newly arrived process is higher than the priority of the process currently running in the system.

Gantt chart in Fig. 7 shows Non-preemptive priority scheduling, process with lowest numeric value will run first and so on.

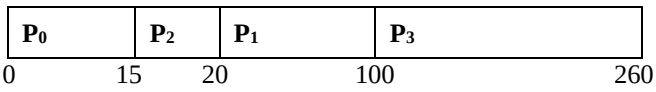

Fig. 7: Gantt chart for Non-Preemptive Priority Scheduling

Table 7: Calculation of Waiting and Turnaround Time for Non-Preemptive PS

Job Sequence	Process ID	Burst Time (ms)	Waiting Time	Turnaround Time
1	P_0	15	0	15
2	P_1	80	19	19
3	P_2	5	13	98
4	P_3	160	97	257

Average Waiting Time = (0+19+13+97) / 4 = 32.25 ms
Average Turnaround Time = (15+19+98+257) / 4 = 97.25 ms

EDRR Scheduling

EDRR (Efficient Dynamic Round Robin) executes the shortest job first instead of FCFS during round robin algorithm [8]. This algorithm eliminates the drawbacks of round robin algorithm in which processes are scheduled in first come first serve manner. This algorithm is not efficient for processors with smaller CPU burst. It decreases the waiting time and response time of processes.

EDRR Scheduling is either Preemptive or Non-Preemptive.

Preemptive EDRR Scheduling

ISSN 2277 – 5528
Impact Factor- 4.015

A Preemptive EDRR will preempt this currently executing process and starts the execution of newly entered process if the newly process have smaller burst time as compared to currently executing process.
Gantt chart in Fig. 8 shows preemptive EDRR scheduling.

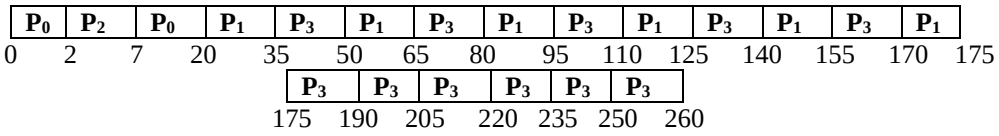

Fig. 8: Gantt chart for Preemptive EDRR Scheduling

Table 8: Calculation of Waiting and Turnaround Time for Preemptive EDRR

Job Sequence	Process ID	Burst Time (ms)	Waiting Time	Turnaround Time
1	P_0	15	5	20
2	P_1	80	94	174
3	P_2	5	0	5
4	P_3	160	97	257

Average Waiting Time = (5+94+0+97) / 4 = 49 ms
Average Turnaround Time = (20+174+5+257) / 4 = 114 ms

Non-Preemptive EDRR Scheduling

A Non-Preemptive EDRR will allow the currently executing process to complete its burst time without any interruption in its execution.
Gantt chart in Fig. 9 shows Non-preemptive EDRR scheduling.

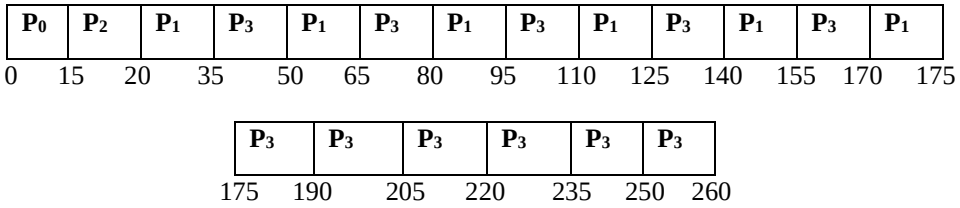

Fig. 9: Gantt chart for Non-Preemptive EDRR Scheduling

Table 9: Calculation of Waiting and Turnaround Time for Non-Preemptive EDRR

Job Sequence	Process ID	Burst Time (ms)	Waiting Time	Turnaround Time
1	P_0	15	0	15
2	P_1	80	89	174
3	P_2	5	13	18
4	P_3	160	97	257

Average Waiting Time = (0+89+13+97) / 4 = 49.75 ms
Average Turnaround Time = (15+174+18+257) / 4 = 116 ms

CAT Next scheduling

ISSN 2277 – 5528
Impact Factor- 4.015

In CAT Next (Closest Average Time Next) scheduling technique, first the average of all burst time is calculated. Execution of the process which has closest burst time to the average time will start first then the process with next closest burst time is executed and vice-versa. Average time is calculated as per burst time given in table 1.
Fig. 10 shows the execution sequence of the processes.

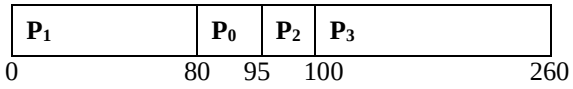

Fig. 10: Gantt chart for CAT Next

The Average Waiting Time and Average Turnaround Time for CAT Next is calculated according to the Table 8.

Table 10: Calculation of Waiting and Turnaround Time for CAT Next

Job Sequence	Process ID	Burst Time (ms)	Waiting Time	Turnaround Time
1	P_1	80	0	80
2	P_0	15	80	95
3	P_2	5	95	100
4	P_3	160	100	260

Average Waiting Time = (0+80+95+100) / 4 = 68.75 ms
Average Turnaround Time = (80+95+100+260) / 4 = 133.75 ms

COMPARISION

In this section we compared the different existing CPU scheduling techniques. In this comparison we select the FCFS, SJF, RR, Priority, EDRR, CAT Next techniques. Then find out waiting time, turnaround time, average waiting time and average turnaround time for each technique. We can find which technique is better by finding average waiting time and average turnaround time and provide a comparative result.

Table 11 show the comparison in non-preemptive CPU Scheduling and table 12 shows Preemptive CPU Scheduling.

Table 11: Comparison of Average Waiting Time and Turnaround Time for different Non-Preemptive CPU Scheduling algorithms

Process ID	Waiting Time						Turnaround Time					
	FCFS	SJF	RR	Priority	EDRR	CAT Next	FCFS	SJF	RR	Priority	EDRR	CAT Next
P_0	0	0	0	0	0	80	15	15	15	15	15	95
P_1	15	19	94	19	89	0	95	99	175	19	174	80

P_2	95	13	28	13	13	95	100	18	33	98	18	100
P_3	100	99	97	97	97	100	260	257	257	257	257	260
Average	**52.5**	**32.25**	**54.75**	**32.25**	**49.75**	**68.75**	**117.5**	**97.25**	**120**	**97.25**	**116**	**133.75**

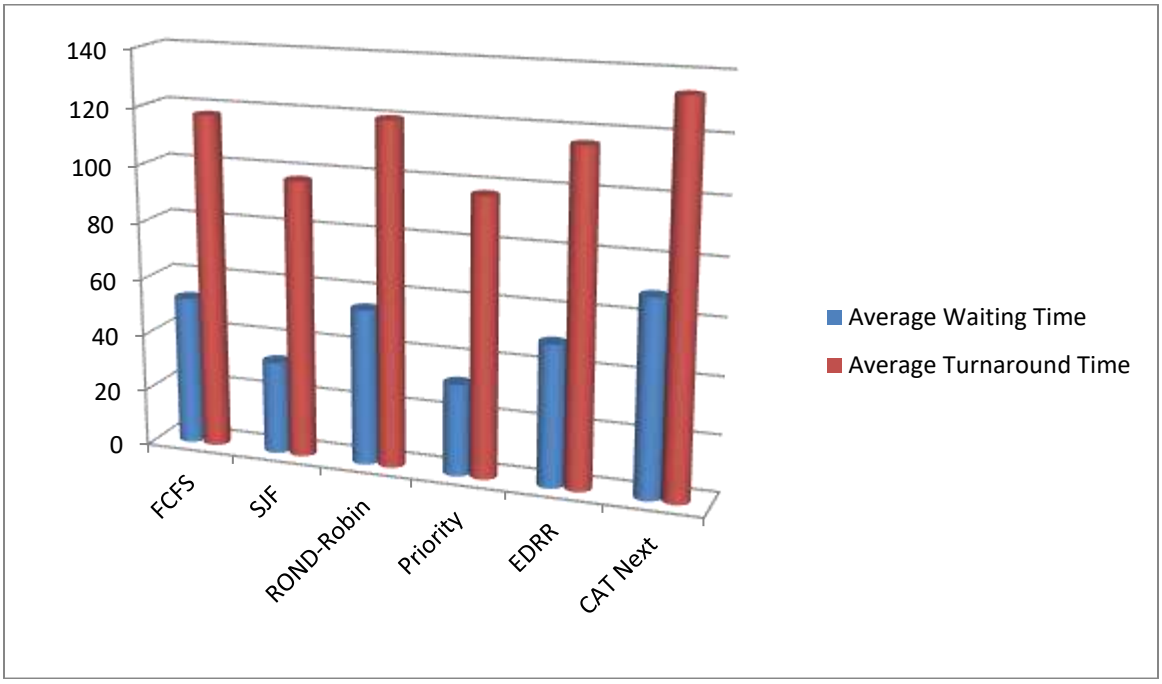

Fig. 11: Plotted chart for comparison in Non-Preemptive CPU Scheduling Algorithms

Table 12: Comparison of Average Waiting Time and Turnaround Time for different Preemptive CPU Scheduling algorithms

Process ID	Waiting Time			Turnaround Time		
	SJF	Priority	EDRR	SJF	Priority	EDRR
P_0	5	245	5	20	260	20
P_1	19	5	94	99	85	174
P_2	0	0	0	5	5	5

P_3	97	83	97	257	243	257
Average	**30.25**	**83.25**	**49**	**95.25**	**148.25**	**114**

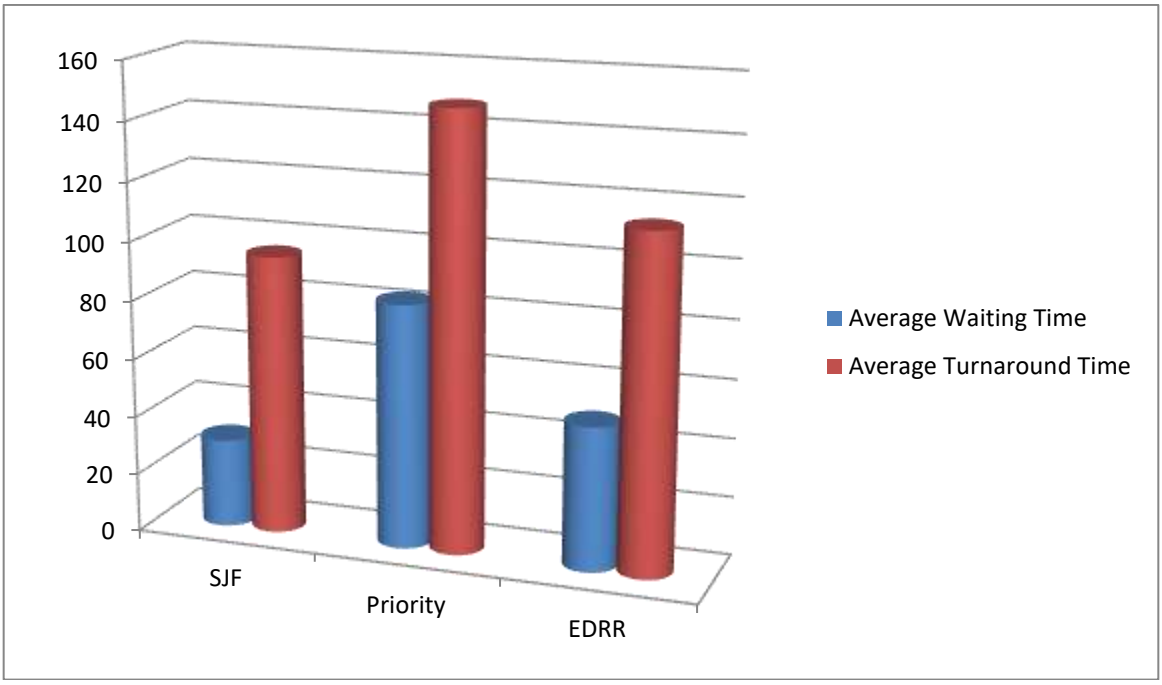

Fig. 12: Plotted chart for comparison in Preemptive CPU Scheduling Algorithms

CONCLUSION

This paper provides a comparative result of various existing CPU Scheduling algorithms. As we know that a better technique requires low waiting time and turnaround time, here in Non-Preemptive CPU Scheduling, the average waiting time and average turnaround time of Shortest Job First and Priority Scheduling is equal and lowest as compare to other techniques. In Preemptive CPU Scheduling the average waiting time and average turnaround time of Shortest Job First Scheduling are lowest as compared to priority scheduling and EDRR scheduling. So, as a result the Shortest Job First technique is best CPU Scheduling technique as compared to FCFS scheduling, Round-Robin scheduling, Priority scheduling, EDRR scheduling and CAT Next scheduling. This paper will assist new researchers of CPU Scheduling algorithm who are keen to contribute their works to the field of CPU scheduling.

REFERENCES

1. Neetu Goel and Dr. R.B. Garg, A Comparative Study of CPU Scheduling Algorithms, International Journal of Graphics & Image Processing Volume 2 issue 4 November 2012.
2. Alka Pant, A Comparison between FCFS and Mixed Scheduling, International Journal of Computer Science and Technology, ISSN: 2229 - 4333, Vol. 2, Issue 2, June 2011.
3. Sukumar Babu Bandarupali,Neelima Priyanka Nutulapati and Prof. Dr. P.Suresh Varma, A Novel CPU Scheduling Algorithm-Preemptive & Non-Preemptive, International Journal of Modern Engineering Research (IJMER), Vol.2, Issue.6, Nov-Dec. 2012 pp-4484-4490.
4. Pushpraj Singh, Vinod Singh and Anjani Pandey, Analysis and Comparison of CPU Scheduling Algorithms, International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Volume 4, Issue 1, January 2014.
5. Nikhil Selvaraj , Yeshwanth Raja , R. Ajay Adithya and Arjun Narendran, Comparison of the Various CPU Scheduling Algorithms, International Journal of Computer Science (IIJCS), ISSN 2321-5992 , Volume 2, Issue 11, November 2014.

6. Mona Saini, Designing Various CPU Scheduling Techniques using SCILAB, International Journal of Computer Science and Information Technologies, Vol. 5 (3), 2014, 2918-2923
7. Neeraj Kumar and Nirvikar, Performance Improvement Using CPU Scheduling Algorithm-SRT, International Journal of Emerging Trends & Technology in Computer Science, Volume 2, Issue 2, March – April 2013.
8. Raman and Pardeep Kumar Mittal, An Efficient Dynamic Round Robin CPU Scheduling Algorithm-EDRR, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 5, May 2014.
9. Yogesh Kumar, Kuldeep Singh Kaswan and Kamal Saluja, A New Approach for CPU Scheduling: CAT Next, International Journal of Research in Computer Applications & Information Technology, Volume 3, Issue 6, November-December 2015.