ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

Analysis of Priority Scheduling Algorithm on the Basis of FCFS & SJF for Similar Priority Jobs

[1] Chandra Shekar N, [2] Karthik V,
[1] Assistant Professor, [2] Engineering 3rd Year,

***Abstract*— Scheduling is one of the most important activities of the process manager which take decision to choose which of the process in the ready queue will be assigned to the CPU. There are different types of scheduling algorithms available for taking decision. One of them is Priority Scheduling Algorithm, which is based on the priority assigned to each process. In priority scheduling the Processes are executed on the basis of priority, the process having highest priority is executed first. In case of similar priority FCFS is used. In this paper, the priority scheduling algorithm is used in such a way that, in case of similar priority SJF algorithm is used instead of FCFS and average waiting time and average turnaround time is calculated. The comparative analysis is performed on the SJF based priority scheduling and FCFS based priority scheduling to compare the average waiting time and average turnaround time**

***Index Terms*—CPU, FCFS, SJF, priority.**

I. INTRODUCTION

Nowadays multicore systems have larger number of cores in real-time applications such as robots [1, 2, 3, and 4]. In order to detect and avoid objects, robots usually require imprecise tasks [5] such as image processing tasks, which consume much CPU time. In addition, robots have many real-time tasks such as motor control, motion control and sensor processing. In order to complete these real-time tasks by their deadlines, both multiprocessor real-time scheduling policies and real-time scheduling algorithms are important. There are mainly two multiprocessor real-time scheduling policies: partitioned scheduling and global scheduling. Partitioned scheduling assigns all tasks to specific processors. Each processor has its own run queue and scheduler. The task assignment algorithm is performed off-line and all tasks are statically assigned to processors. In contrast, global scheduling permits to migrate tasks between processors. Ready tasks are enqueued in a logical global queue and the *M* highest priority tasks, where *M* is the number of processors, are assigned to processors.

Resource scheduling in modern IT systems serves a number of different goals. One of the key goals is to schedule system resources, e.g. CPU and disk, such as to provide each user the illusion of "owning the system" or a fair share of it. Another goal is to schedule requests such as to make efficient use of the system resources (e.g. minimize the movements of a disk head).

Scheduling as a means to provide *shorter mean response times* across *all* requests in a system. The response time of a request is defined as the time from when a user submits a request until the user receives the complete response. We use two common real-world applications to illustrate how scheduling can greatly improve user experienced system performance, without requiring major changes to system hardware or software.

II. FIRST COME FIRST SERVE (FCFS)

In FcFs Jobs are executed on first come, first serve basis, It is a non-preemptive, pre-emptive scheduling algorithm, Easy to understand and implement, Its implementation is based on FIFO queue, Poor in performance as average wait time is high.

III. SHORTEST JOB FIRST (SJF)

SJF is a non-preemptive, pre-emptive scheduling algorithm. It is Best approach to minimize waiting time, Easy to implement in Batch systems where required CPU time is known in advance .It is impossible to implement in interactive systems where required CPU time is not known, Here the processer should know in advance how much time process will take.

IV. PRIORITY BASED SCHEDULING

Priority scheduling is a non-preemptive algorithm and one of the most common scheduling algorithms in batch systems. Each process is assigned a priority. Process with highest priority is to be executed first and so on. Processes with same priority are executed on first come first served basis. Priority can be decided based on memory requirements, time requirements or any other resource requirement.

V. EXISTING FCFS BASED PRIORITY SCHEDULING ALGORITHM

In the existing algorithm the processes are executed according to priority, such that the process having highest priority will execute first. On the basis of execution of each process, the waiting time and turnaround time is calculated but in the case of similar priority FCFS i.e. first Come First Serve is used. In which, if the two or more process have similar priority then the process which comes first is executed first, algorithm for priority Scheduling [6] are as follows:

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

Existing Algorithm:
Step 1: Assign the process to ready queue.
Step 2: Assign the process to the CPU according to the priority, higher priority process will get the CPU first than lower priority process.
Step 3: If two processes have similar priority then FCFS is used to break the tie.
Step 4: Repeat the step 1 to 3 until ready queue is empty. **Step 5:** Calculate Waiting time and Turnaround time of individual Process.
Step 6: Calculate Average waiting time and Average Turnaround time.
Step 7: Calculate Average waiting time and Average Turnaroun
d time.

IX. IMPLEMENTATION

The proposed algorithm has been implemented on Windows XP Operating System by using Turbo C++ editor. Different cases are used to implement the algorithm. The cases have number of processes along with the burst time and priority. To calculate the waiting time and turnaround time we use

Formulas as follows:
Turnaround Time = Process completion time - Arrival time.
Waiting time =Turnaround Time - Burst Time

Table1

VI. PROBLEM STATEMENT

One of the most important problems in operating systems designing is CPU Scheduling and challenge in this field is to build a program to achieve proper scheduling. In case of priority scheduling algorithm when similar priority jobs arrive then FCFS is used and as a consequence the average waiting and turnaround time becomes relatively higher. The process that arrives first is executed first, no matter how long it takes the CPU. So in this case if long burst time processes are execute earlier, then other process will remain in waiting queue for a long time. This type of arrangement of processes in the ready queue results in the higher average waiting and turnaround time.

VII. OBJECTIVE

The objective of this paper is:
- To Analysis SJF priority based and FCFS priority based scheduling algorithm.
- To reduce the Average waiting time and Average Turnaround time of CPU.

VIII. PROPOSED SJF BASED PRIORITY SCHEDULING ALGORITHM

In proposed algorithm, SJF based Priority Scheduling Algorithm is used. In which each process that have similar priority is executed on the basis of burst time, i.e. the process which have least burst time will execute first. The SJF based priority algorithm results in reduced average waiting time and turnaround time.

Proposed Algorithm:
Step 1: Assign the process to ready queue.
Step 2: Assign the process to the CPU according to the priority, higher priority process will get the CPU first than lower priority process.
Step 3: If two processes have similar priority then use FCFS.
Step 4: If in FCFS both arrival times are same
then, SJF is used to break the tie.
Step 5: Repeat the step 1 to 4 until ready queue is empty.
Step 6: Calculate Waiting time and Turnaround time of individual Process.

INPUT VALUES:

Process	Burst Time	Priorities
P1	183	4
P2	112	2
P3	190	5
P4	84	1
P5	245	6
P6	26	1
P7	123	3
P8	132	4
P9	197	6
P10	256	7

A: Expected outcome on the basis of existing FCFS based priority scheduling algorithm.

P4	P6	P2	P7	P1	P8	P3	P5	P9	P10
0 84	110	222	345	528	660	850	1095	1292	548

Figure 1: Expected Gantt chart for the above processes (For Similar Priorities)

B: Expected outcome on the basis of proposed SJF based priority scheduling algorithm.

P4	P6	P2	P7	P1	P8	P3	P5	P9	P10
0 26	110	222	345	477	660	850	1047	1292	1548

Figure 2: Expected Gantt chart for the above processes (For Similar Priorities)

X. RESULT AND ANALYSIS

In this paper an algorithm is proposed in which SJF is used to schedule the similar priority jobs. For this different test cases were used through which we can easily compare the result of FCFS based Priority Scheduling Algorithm and SJF based Priority Scheduling Algorithm.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

A: Result:
The results are shown with the help of snapshot according to the case.

process	pid	Burst Time	Priority	Waiting Time	Turnaround Time
p	4	84	1	0	84
p	6	26	1	84	110
p	2	112	2	110	222
p	7	123	3	222	345
p	1	183	4	345	528
p	8	132	4	528	660
p	3	190	5	660	850
p	5	245	6	850	1095
p	9	197	6	1095	1292
p	10	256	7	1292	1548

Average waiting Time is 518.600000
Average Turnaround Time is 673.400000

Figure 3 Case 1 FCFS based priority scheduling for 10 processes.

process	pid	Burst Time	Priority	Waiting Time	Turnaround Time
p	6	26	1	0	26
p	4	84	1	26	110
p	2	112	2	110	222
p	7	123	3	222	345
p	8	132	4	345	477
p	1	183	4	477	660
p	3	190	5	660	850
p	9	197	6	850	1047
p	5	245	6	1047	1292
p	10	256	7	1292	1548

Average Waiting Time is 502.900000
Average Turnaround Time is 657.700000

Figure 4 Case 1 SJF based priority scheduling for 10 processes.

COMPARISION OF WAITING TIME AND TURNAROUND TIME FOR 10 PROCESSES:

Process	Waiting Time (ms)		Turnaround Time (ms)	
	FCFS based Priority Scheduling Algorithm	SJF based Priority Scheduling Algorithm	FCFS based Priority Scheduling Algorithm	SJF based Priority Scheduling Algorithm
P1	345	477	528	660
P2	110	110	222	222
P3	660	660	850	850
P4	0	26	84	110
P5	850	1047	1095	1292
P6	84	0	110	26
P7	222	222	345	345
P8	528	345	660	477
P9	1095	850	1292	1047
P10	1292	1292	1548	1548
Average	518.60	502.90	673.40	657.70

ANALYSIS:
On the basis of the above outcome for FCFS and SJF based priority scheduling algorithm, the average waiting and average turnaround time is calculated which is shown in graph.
 Comparison of average waiting time and turnaround time for 10 processes.

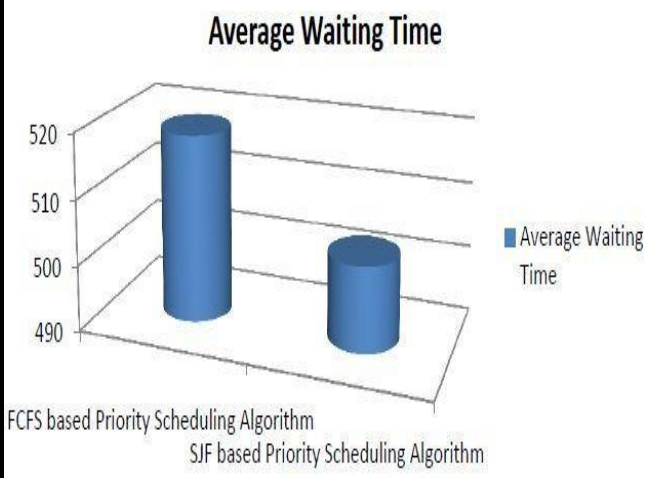

Figure 5: Average Waiting Time of FCFS based Vs SJF based Priority Scheduling Algorithm.

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

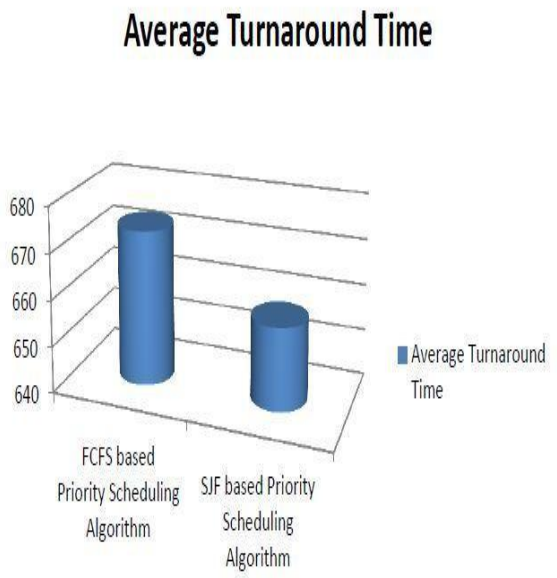

Figure 6: Average Turnaround Time of FCFS based Vs SJF based Priority Scheduling Algorithm

CONCLUSION

There are many scheduling algorithms having their own benefits and drawbacks. Scheduling can also be done on the basis of priority. Each process assigned a priority and the process which has highest priority will execute first. In case of similar priority generally FCFS is used to select the next process. If SJF based priority scheduling algorithm is used when two or more process having similar priority, instead of FCFS, then the average waiting time and average turnaround time is reduced. We proposed SJF based priority scheduling algorithm in which, the process that having lowest burst time will execute first. The existing FCFS based priority scheduling algorithm and proposed SJF based priority scheduling algorithm is analyzed and the result shows that the average waiting time and average turnaround time is reduced.

REFERENCES

[1]] F. Kanehiro, H. Hirukawa, and S. Kajita. OpenHRP: Open Architecture Humanoid Robotics Platform. The International Journal of Robotics Research, 23(2):155–165, 2004.

[2] T. Taira, N. Kamata, and N. Yamasaki. Design and Implementation of Reconfigurable Modular Robot Architecture. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3566–3571, 2005.

[3] H. S. Ahn, Y. M. Beak, I.-K. Sa, W. S. Kang, J. H. Na, and J. Y. Choi. Design of Reconfigurable Heterogeneous Modular Architecture for Service Robots. In IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems, pages 1313–1318, Sept. 2008.

[4] F. R. T. Center. http://www.furo.org/

[5]] K. Lin, S. Natarajan, and J. W. S. Liu. Imprecise Results: Utilizing Partial Computations in Real-Time Systems. In

[6] Proceedings of the 8th IEEE Real-Time Systems Symposium, pages 210–217, Dec. 1987.

[7] Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, “Operating System Concepts”, 7th Edition, John Wiley & Sons, ISBN: 978-81-265-2051-0, 2005.