ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 4, April 2013

Comparative Analysis of Various Scheduling Algorithms

Lalit Kishor, Dinesh Goyal

Abstract— **Scheduling is the technique used for controlling the order of the job which is to be performed by a CPU of a computer. The motive of scheduling is to engage the CPU at its maximum capacity and no process shall wait for longer time and to finish the entire task in minimum possible time. In this paper, we discuss various types of Scheduling algorithms and Compare their performance on terms of throughput and waiting time. First of them is First Come First Served (FCFS) which is a non-preemptive and the simplest scheduling. FCFS is good for long job. Second is Shortest Job First (SJF) scheduling which selects that job first which has least processing time i.e. the processing demanding the less CPU time is executed first. Next is Round Robin (RR) scheduling, it removes the drawbacks of FCFS by preempting running jobs periodically. But if the length of time quantum is too short then more time will waste in context switching. Last one is Priority Based scheduling where each process is assigned a priority (preference) to create the order of execution.**

***Index Terms*-Scheduling, FCFS Scheduling, Shortest Job First Scheduling, Round Robin Scheduling, Priority Based Scheduling.**

I. SCHEDULING

Scheduling can be defined as a mechanism or a tool to control the execution of number of processes performed by a computer. CPU is the most important of all the resources available in a computer system that are scheduled before use; Multiprogramming is attained by efficient scheduling of the CPU. The basic idea is to keep the CPU busy as much as possible by executing a process, and then switch to another process.

The key to Multi- Programming is scheduling. The Multi-Layer Queue (MLQ) scheduling partitions the queue into several Separate queues. Each queue has its own scheduling Algorithm. Each process can be easily classified into Groups based on several properties of the process and permanently assigned to one queue. In the MLFQ scheduling, the processes can be dynamically moved in different queues. So processes that need a large amount of CPU time are sent to the low priority queues and process requiring less amount of CPU and more other bounds are sent to high priority queues. [01]

The Scheduling performance can be analyzed on following criteria:

1. **CPU utilization**:-The maximum use of CPU when it is busy.
2. **Throughput**: - It is the number of processes that complete there execution per unit time.
3. **Turnaround Time**:-It is the amount needed for execution of a single process.
4. **Waiting Time**: - It is the amount of time a process waits in the ready queue.
5. **Response Time**:-This is the amount of time takes from when a request was submitted until the first response is produced not output.
6.

Scheduling can be divided into two categories.

1. **Non preemptive: -** A non preemptive scheduling algorithm picks a process to run and then just lets it run until it blocks or until it voluntarily released by CPU, in other words it engages itself with the first task or job until unless finished, for e.g. FCFS, SJF.
2. **Preemptive:-** in this type of scheduling execution of process may be preempted before the completion of the burst time of process and some other process may starts its execution whose priority is higher than the first arrived process in the CPU, for e.g. Round Robin, Priority Driven.

Let's take 10 processes that arrive at same time in the below given order and analysis their performance in various scheduling (namely FCFS, SJF, RR & Priority) algorithms, with given priorities and time quantum of 9ms.

We have used MATLAB for above analysis and tried to attain maximum possible accurate results.

Process	CPU burst time(ms)	Priority
P1	34	9
P2	23	10
P3	11	8
P4	66	6
P5	21	7
P6	56	5
P7	16	4
P8	9	1
P9	17	3
P10	29	2

Table 1.1 List of Process with Burst time and Priority

II. FIRST COME FIRST SERVE (FCFS)

Manuscript received April, 2013.

Lalit Kishor,*Department of Computer Engineering,Suresh Gyan Vihat University Jaipur, India, +919214012880*

Dinesh Goyal, *Department of Computer Engineering,Suresh Gyan Vihat University Jaipur , India.*

FCFS is a non preemptive scheduling algorithm. It uses First in- First out) FIFO strategy to assign the priority to processes in the order, that is same as the request made by process for the processor. The process or job that requests the CPU first is allocated the CPU first and other if in the queue has to wait until the CPU is free. FCFS is also known as First-In-First-Out (FIFO), it is the simplest scheduling methodology. All the later arriving jobs are inserted into the tail (rear) of the ready queue and the process to be executed next is removed from the head (front) of the queue and the control of current process is transferred to the CPU. FCFS gives better performance for longer jobs and less multiple processes in ready queue. Relative importance of jobs measured only by arrival time (poor choice).

Gantt chart for above process as per FCFS is

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10

0 34 57 68 134 155 211 227 236 253 282

Average waiting time= 1375/10=137.5ms
Turnaround time =burst time +waiting time
Average turnaround time = 1657/10=165.7 ms

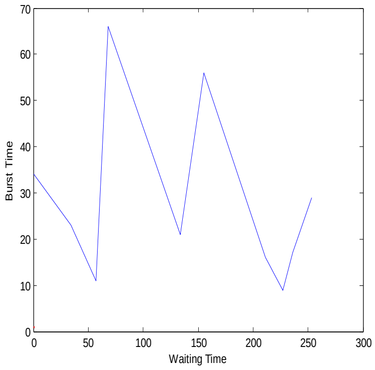

Fig 1:-Graph representation of The Performance of the Process between Burst Time and Waiting Time for FCFS.

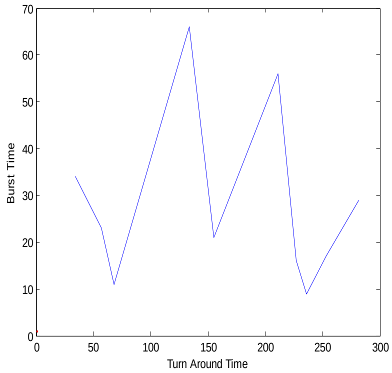

Fig 2:- Graph representation of The Performance of the Process between Burst Time and Turnaround Time for FCFS.

Drawback of FIFO can be observed as, the average time for waiting for purely FIFO system is very poor as seen in above diagrams also, ***"a process with 9ms as burst time has waiting time of 230ms"*** especially for low burst time.

Let's take a case of implementing FIFO in Multitasking system. As we know in a multi-task system, several processes are kept in the main memory and the CPU is kept active to run a process while the others are waiting. In case of FIFO the multiple tasks will be waiting in the ready queue till the first job is over.

III. SHORTEST JOB FIRST (SJF)

In SJF technique the shortest amongst the entire ready queue job is executed first rest all are preempted. The benefit if this is that waiting time is minimal for the shorter jobs.

The SJF is especially appropriate for the batch jobs for which the run time are known in advance.

SJF can classified in two schemes non preemptive and preemptive.
Non preemptive SJF:-Once the CPU is given to the process it cannot be preempted until completes its CPU burst, even though if arriving process has shorter burst time.
Preemptive SJF (SRTF):- If a new process arrives to the CPU with shorter burst time than the current executing process then the CPU preempts the currently executing time of current executing process. This scheme is also known as the shortest remaining time first (SRTF).

Gantt chart for above process as per Non preemptive SJF is

P8	P3	P7	P9	P5	P2	P10	P1	P6	P4

0 9 20 36 53 74 97 126 160 216
282

Average waiting time= 791/10=79.1ms

Average turn around time = 1082/10=108.2 ms

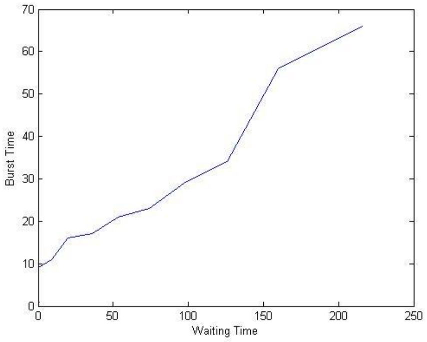

Fig 3:-Graph representation of The Performance of the Process between Burst Time and Waiting Time for SJF.

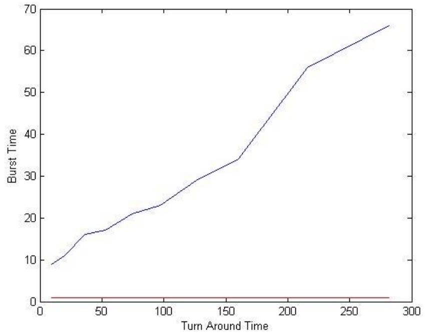

Fig 4:-Graph representation of The Performance of the Process between Burst Time and Turnaround Time for SJF.

Drawback of SJF algorithm is to know which incoming process is indeed shorter than another. This requires a separate algorithm running for monitoring and sorting the jobs in real time. Also, long running jobs may starve, because the CPU may have a good and steady supply of short jobs.

SJF has two variants, one as always known SJF and the other is LJF (longest Job First) [02].

IV. ROUND ROBIN SCHEDULING

In this approach a fixed time slot is defined before the execution of processes starts, which is a normally small unit of time. In each time slice (quantum) the CPU executes the current process only up to the end of time slice. If that process is having less burst time than the time slice then it is completed and is discarded from the queue and the next process in queue is handled by CPU. However, if the process is not completed then it is halted (preempted) and is put at the end of the queue and then the next process as per arrival time in line is addressed during the next time slice. Round Robin reduces the penalty that short jobs suffer with FCFS by preempting running jobs periodically, and also saves starving of longer jobs and scheduling effort in case of SJF. The main advantage of Round Robin Scheduling is that every process gets the CPU and thus there is no starvation.

For this analysis we have assumed a time quantum of 9ms

Gantt chart:-

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10

0 9 18 27 36 45 54 63 72 81 90

P1	P2	P3	P4	P5	P6	P7	P9	P10	P1

99 108 110 119 128 137 143 151 160 169

P2	P4	P5	P6	P10	P1	P4	P6	P10	P4

174 183 186 195 204 211 220 229 231 240

P6	P4	P6	P4	P6	P4

249 258 267 276 278 281

Average waiting time= 1555/10=155.5ms

Average turnaround time = 1837/10=183.7ms

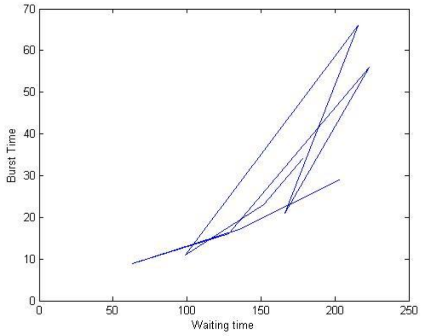

Fig 5:-Graph representation of The Performance of the Process between Burst Time and Waiting Time for Round Robin.

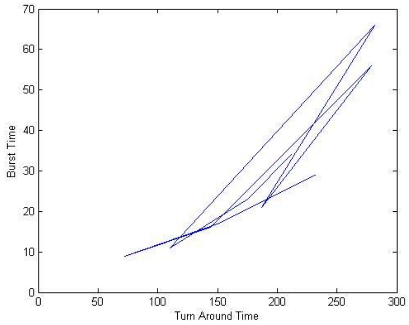

Fig 6:-Graph representation of The Performance of the Process between Burst Time and Turnaround Time for Round Robin.

Drawback of this method is that it slows down the short processes because they have to share the CPU time with other processes instead of just finishing up quickly. Thus the critical issue with the RR policy is the length of the quantum. In case it is too short, then the CPU will be spending more time on context switching or if too long then processes demanding less CPU time will suffer.

V. PRIORITY BASED SCHEDULING

In Priority scheduling algorithm each process is assigned priority by either an outer agency or as per their system requirements and as soon as each process hits the queue it is sorted in based on its priority so that process with higher priority are dealt with first. In case two processes arrive with same priority in different order then they are executed in FCFS order. The main advantage of Priority scheduling is that the important jobs can be finished first.

Gantt chart:-

P8	P10	P9	P7	P6	P4	P5	P3	P1	P2

0 9 38 55 71 127 193 214 225 259

Average waiting time= 1191/10=119.1ms

Average turnaround time = 1473/10=147.3ms

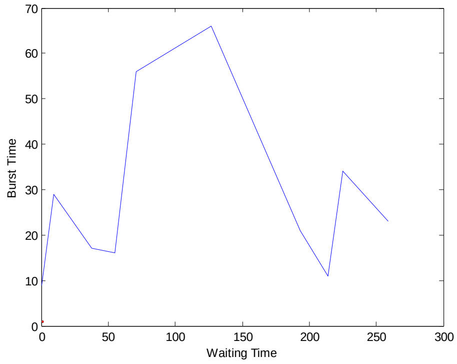

Fig.7:- Graph representation of The Performance of the Process between Burst Time and Waiting Time for Priority Scheduling

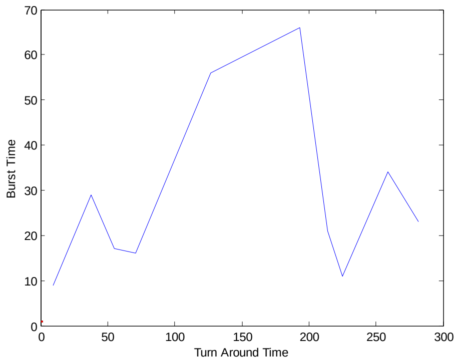

Fig. 8:- Fig 4:-Graph representation of The Performance of the Process between Burst Time and Turnaround Time for Priority Scheduling.

Drawback with Priority scheduling is when the operating system gives a particular task very low priority so it sits in queue for a larger amount of time, not being dealt with by the CPU.

Mr. T. Funkhouser and Mr. P. Shilane in their paper used the priority scheduling for Database in which, they introduce a priority-driven algorithm for searching all objects in a database at once. The algorithm is given a query object and a database of target objects, all represented by sets of local shape features, and its goal is to produce a ranked list of the best target objects sorted by how well any subset of *k* features on the query match features on the target object. To achieve this goal, the system maintains a priority queue of potential sets of feature correspondences (partial matches) sorted by a cost function accounting for both feature dissimilarity and geometric deformation. Initially, all pairwise correspondences between the features of the query and features of target objects are loaded onto the priority queue. [05]

VI.COMPARISON OF SCHEDULING

After doing rigorous analysis of assumed process in Table 1, the given table gives a comparison between various types of scheduling according to given parameters.

Scheduling algorithm	**CPU Engagement**	**Through put**	**Turnaround time (Total TAT)**	**Waiting Time (Total WT)**
FCFS	High	Low	High (1657)	High (1375)
SJF	Medium	High	Medium (1082)	Low (791)
RR	High	Medium	High (1845)	High (1563)
Priority	High	Low	High (1473)	Medium (1191)

Table 2: Comparison of various scheduling Algorithms on the basis of experiments done

VII. CONCLUSION

In this paper we have discussed about scheduling and then various types of scheduling. A comparison of various types of algorithms is also shown with practical implementation using MATLAB. By this experimental setup we have been able to do statistical analysis of the performance of all the four basic scheduling algorithms, as stated above.

VIII FUTURE WORK

By examining the advantages and drawbacks of various algorithms we can suggest a new scheduling algorithm which can solve drawbacks of all discussed algorithms.

We will proposing in our next paper a time quantum based scheduling algorithm which will involve both SJF and round robin and will also be using dynamic time quantum.

We may be able to increase the Performance, Throughput and decrease the Turnaround Time by above solution.

IX. REFERENCES

[01]. Mohammad Reza EffatParvar, Karim Faez, Mehdi EffatParvar, Mehdi Zarei, Saeed Safari. “An Intelligent MLFQ Scheduling Algorithm (IMLFQ) with Fault Tolerant Mechanism”

[02]. Shiann-Tsong Sheu, Yue-Ru Chuang and Hsuen-Wen Tseng. “A Pipelined Scheduling Strategy for Supporting Variable-Length Packets in WDM Networks”.

[03]. *J. M. Arco D. Meziat B. Alarcos. “*Deficit Round Robin Alternated: A New Scheduling Algorithm”

[04]. Roberto Riggio, Daniele Miorandi, and Imrich Chlamtac. Airtime Deficit Round Robin (ADRR) Packet Scheduling Algorithm

[05]. T. Funkhouser and P. Shilane. Partial Matching of 3D Shapes with Priorit

Lalit kishor is a Research scholar at SGVU, pursuing M.Tech, did his B.Tech from University of Rajasthan, good academics and research work/

Dinesh Goyal, Associate Professor at Suresh Gyan Vihar University, has 16 international publications, having 15 years of experience of academics and research, Vice Chairman, CSI, Jaipur, Member IEEE, 15 Conferences, 12 research guidance , Many conference & Workshops and seminars.