Volume 4, Issue 6, June 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Comparative Performance Study of CPU Scheduling Algorithms

Himani Aggarwal
M.Tech CSE
Geeta Engineering College, Naulatha, India

Er. Shakti Nagpal
HOD, CSE Department
Geeta Engineering College, Panipat, India

ABSTRACT- Some CPU scheduling algorithms such as First-Come-First-Serve (FCFS), Shortest Job First (SJF), Priority, Round Robin (R-R), Multilevel Queue(MQ) and Multilevel Feedback Queue (MFQ has been elaborated and assessed on the basic CPU scheduling objectives i.e.; average waiting(AWT), average turnaround (ATT)and average response time(ART), average CPU utilization (AU)and average throughput(AT). These will form the base parameters in making a decision for the suitability of the given algorithm for a given objective. Comparative performance study of various CPU scheduling algorithm is done in this review paper.

Keywords— Scheduling algorithm; First-Come-First-Serve scheduling; Shortest Job First scheduling; Priority scheduling; Round Robin scheduling; Multilevel Queue scheduling; Multilevel Feedback Queue scheduling

I. Introduction:

Scheduling is the method by which threads, processes or data flows are given access to system resources. The need for a scheduling algorithm arises from the requirement for most modern systems to perform multitasking (execute more than one process at a time) and multiplexing (transmit multiple flows simultaneously). Scheduling is a fundamental operating-system function. Almost all computer resources are scheduled before use. The CPU is, of course, one of the primary computer resources. Thus, its scheduling is central to operating-system design.

Scheduling Criteria:

a) CPU Utilization: (max) To keep the CPU as busy as possible.
b) Turnaround time: (min) Amount of time to execute a particular process. The interval from the time of submission of a process to the time of completion is the turnaround time
c)Response time: (min) Time taken from first request until first response produced. It is the time it takes to start responding, not the time it takes to output the response.
d)Waiting time: (min)Amount of time a process waiting in the ready queue. The CPU scheduling algorithm affects only the amount of time that a process spends waiting in the ready queue.
e)Throughput: (max) No. of processes that complete their execution per unit of time.

Scheduling objectives:

a) Achieve fairness, lack of starvation of any particular process, efficient use of processor time, and low overhead.
b) To describe various CPU-scheduling algorithms.
c) To discuss evaluation criteria for selecting a CPU-scheduling algorithm for a particular system.
d) Avoid indefinite blocking or starvation: A process should not wait for unbounded time before or while process service.
e) Achieve balance between response and utilization: The scheduling mechanism should keep resources of system busy.

Scheduling Algorithms:-

1) First-Come- First- Serve (FCFS) Scheduling: -In this, the process that requests the CPU first is allocated the CPU first. Its implementation is easily managed with FIFO queue. When the CPU is free, it is allocated to the process which is at the head of the queue. It is a non-pre-emptive scheduling algorithm. Processes are assigned the CPU in the order they request it. The most intuitive and simplest technique is to allow the first process submitted to run first.

2) Shortest-Job-First (SJF) Scheduling:-In this scheduling algorithm, the CPU is allotted to the process which has the smallest next CPU burst. The SJF uses the FCFS to break tie (a situation where two processes have the same length next CPU burst). The SJF algorithm can be pre-emptive or non-pre-emptive. In pre-emptive SJF scheduling, the execution of a process that is currently running is interrupted in order to give the CPU to a newly arrived process with a shorter next CPU burst. On the other hand, the non-pre-emptive SJF will allow the currently running process to finish its CPU burst before a new process is allocated to the CPU.

3) Priority Scheduling:-In priority scheduling algorithm, a priority is associated with each process and the CPU is allocated to the process with the highest priority. The processes which have equal priority; they are scheduled with FCFS policy. Priority scheduling can be either pre-emptive or non-pre-emptive.

4) Round-Robin (RR) Scheduling:-This scheduling is a pre-emptive version of FCFS scheduling. Processes are dispatched in a first-in-first-out sequence but each process is allowed to run for only a limited amount of time. This time interval is known as a time-slice or time quantum. In this, the ready queue is treated as the circular queue.

5) Multilevel Queue Scheduling:-The processes can be classified into different groups depending upon their situation. For example, the common division between processes is foreground processes(interactive) and background (batch) processes.

6) Multilevel queue scheduling:- The processes are permanently assigned to a queue when they enter in the system. But, multilevel feedback queue allows the processes to move between the queues. In this, the processes are categorized according to their CPU bursts.

II. COMPARISON OF SCHEDULING ALGORITHMS-

When we perform the comparison we came to know that the SJF algorithm is performing with minimum average waiting time, average turnaround time and average response time whereas FCFS with minimum CPU utilization and throughput.

Our analysis revealed that a single policy is not sufficient for resource management in parallel computational environments. Such environments need to implement dynamic and adaptive scheduling policy.

Basically the scheduling algorithms are divided into two categories as Pre-emptive and Non-pre-emptive scheduling algorithms.

In non-pre-emptive, once the CPU has been allocated to a process, the process keeps the CPU until it releases the CPU either by terminating or switching to the waiting state. In pre-emptive, CPU allocated to a process is switched if another process of higher priority is scheduled. In this case, the currently running process is interrupted and moved to the ready state by the operating system.

Table 1: First Observation:-

Scheduling	AWT (ms)	ATT (ms)	ART (ms)	AT %	AU %
FCFS	25.5	32.0	25.5	71	69
SJF	14.3	20.8	14.3	94	78
PR	41.2	47.7	41.2	77	85
RR	33.8	40.3	33.8	82	98
MQS	31.5	38.0	16.8	76	74
MFQS	26.8	33.3	15.7	82	77

Table 2: Second Observation:-

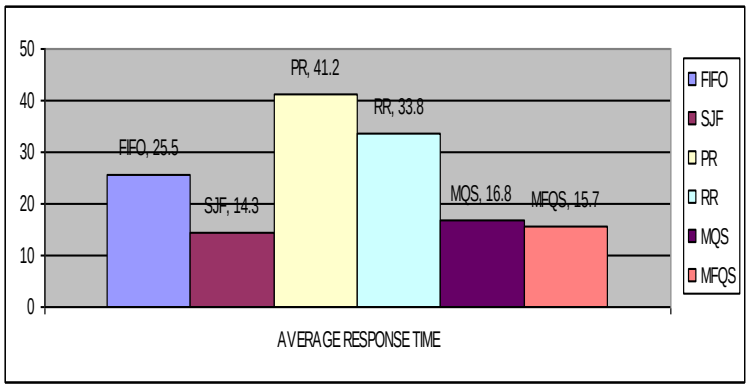
Scheduling	AWT	ATT	ART	AT	AU
FCFS	25.3	32.4	25.3	72.8	69.2
SJF	14.39	21.8	14.9	94.1	78.5
PR	41.6	46.8	41.6	77.8	85.1
RR	33.0	41.5	33.0	82.2	98.3
MQS	31.58	38.8	16.2	76.3	74.5
MFQS	27.0	33.6	15.1	82.8	77.8

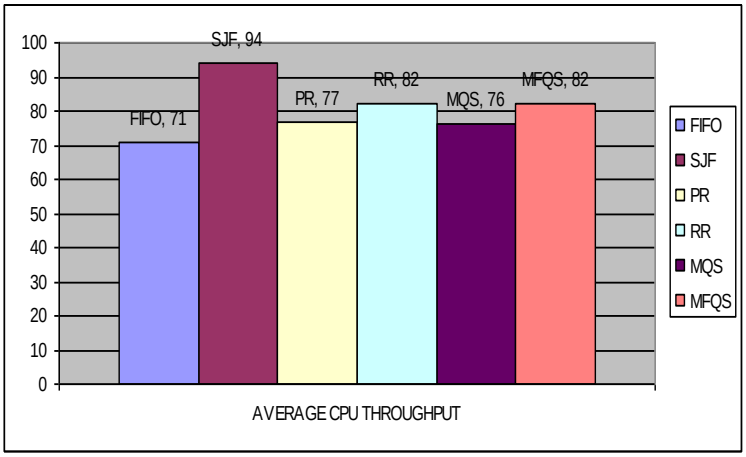
Table 3: Third Observation:-

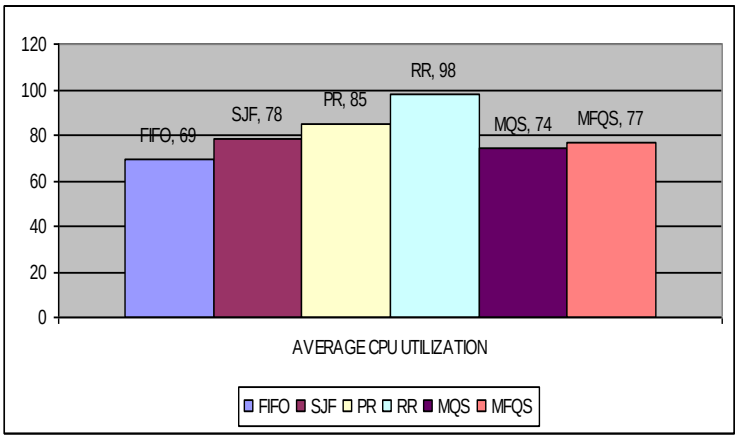
Scheduling	AWT	ATT	ART	AT	AU
FCFS	25.0	32.9	25.1	71.9	69.9
SJF	14.4	21.5	13.9	95.2	78.1
PR	41.2	47.9	40.9	76.9	85.9
RR	33.6	40.5	33.2	82.9	98.5
MQS	31.2	38.1	38.1	76.9	74.7
MFQS	26.0	32.8	32.8	82.5	77.9

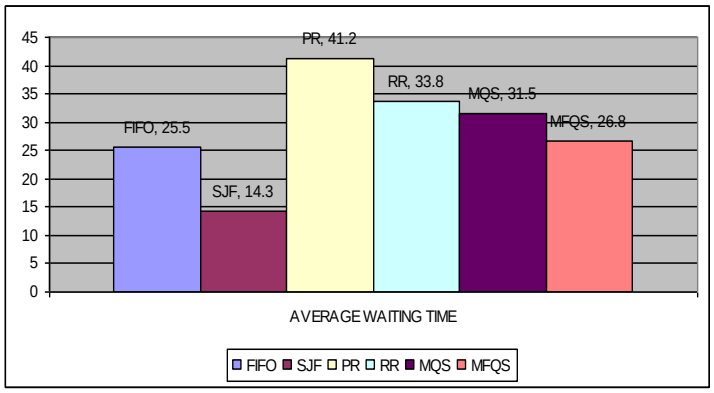
Table 4: Fourth Observation:-

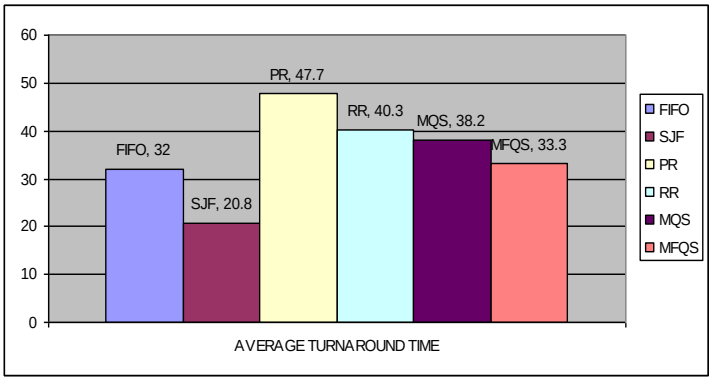
Scheduling	AWT	ATT	ART	AT	AU
FCFS	25.7	32.1	25.7	71.2	69.3
SJF	14.5	20.7	14.2	95	78.9
PR	41.0	47.2	41.0	78.0	85.4
RR	33.5	40.9	33.4	82.5	98.5
MQS	31.9	38.7	16.9	77.0	74.4
MFQS	26.5	33.0	15.6	82.9	77.2











III. Conclusion and Future scope:

We made a comparative study of various CPU scheduling algorithms. It is concluded that the SJF algorithm is superior in terms of increasing fairness, decrease in response time and timely resource allocation to individual process. FCFS algorithm clearly shows maximum CPU utilization and efficient handling of resources. In the future we will try to solve the problem of starvation which is encountered in SJF, Priority and MQS scheduling algorithm.

References:

[1] http://en.wikipedia.org./wiki/scheduling
[2] Operating Systems Sibsankar Haldar 2009,Pearson Education, India
[3] D.M. Dhamdhere operating Systems A Concept Based Approach,Second edition, Tata McGraw-Hill, 2006.
[4] Sabrina, F.C.D, Nguyen, S.jha, D. Platt and F. Safaei,2005.Processing scheduling in programmable networks. Computer commun, 28:676-687.
[5] Silberchatz, Galvin and Gagne, 2003. Operating systems concepts.
[6] S.Sredharran,Operating Systems,2000 ISBN 81-87721-18-9

