IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Comparison of Various CPU Scheduling Algorithms

#1Ubita, #2 Mohit Sharma

M.tech Scholar CSE, HOD CSE

I.G.U Rewari,YCET,Narnaul

Haryana, India

Abstract: CPU scheduling, also known as process scheduling, is one of the most important features within an operating system that guarantees multiprogramming where a number of programs would be competing for computer resources at the same time in the system. In this paper we are trying to study and compare various types of known CPU scheduling algorithms namely First Come First Serve (FCFS), Round Robin (RR), Shortest Job First (SJF), Priority and Shortest Remaining Time First (SRTF) scheduling. Which one to use out of these algorithms totally depends on types of process running on the system, CPU utilization, throughput, waiting time, response time needed etc. Sometimes, more than one of these algorithms will be used within same system in different situations. In this paper we will try to test these scheduling algorithms and summarize which one performs better in what situation.

Keyword:- FCFS, SJF, SRTF, PS, RR, FIFO, Ms, AWT, TAT,CT,OS,PCB,I/O

1. INTRODUCTION

Scheduling is done to complete the work on time. Only one process can run at a time in single processor system. When one process is running, others have to wait until the process completes its execution and after that, other processes that are in waiting can be given CPU for its execution, that is what happens in single processor system. In multiple processor system, more processes can run simultaneously at same time. When more numbers of processes running all times, to maximize CPU utilization is the purpose of multiprogramming. The work of multiprogramming depends on CPU scheduling. The main aim of CPU scheduling is to make system efficient, fair and fast.The main work of CPU scheduling is to make sure that whenever the CPU remains idle, the operating system at least selects one of the processes which are available in the ready queue for execution. CPU scheduling is a process that permits one process to use CPU while the execution of another process is on waiting state due to lack of any resources like I/O etc., thereby making full use of CPU. Process execution consists of a cycle of CPU execution and I/O wait. When the process is executed in CPU is known as CPU burst and when the CPU is waiting for I/O for further execution is known as I/O burst.

Scheduling main objectives

- Maximize throughput
- Implementation of priorities
- Be predictable
- Minimize overhead
- Balanced available resources
- Balance between response and CPU utilization
- Fairness to all processes

A. Long-term Scheduler

The long-term or admission scheduler decides which jobs or processes are to be admitted to the ready queue; that is, when an attempt is made to execute a process its admission to the set of currently executing processes is either authorized or delayed by the long-term scheduler. Thus, this scheduler dictates what processes are to run on a system, and the degree of concurrency to be supported at any one time.

B. Mid-term Scheduler

The mid-term scheduler temporarily removes process from main memory and place them on secondary memory (such as a disk drive) or vice versa. This is only referred to as “swapping of processes out" or "swapping in" (also incorrectly as "paging out" or "paging in").

C. Short-term Scheduler

The short-term scheduler (also known as the CPU scheduler) decides which of processes in the ready queue, in memory are to be executed (allocated a CPU) next following a clock interrupt, an Input-Output (IO) interrupt and an OS call or another form of signal.

2.SCHEDULING CRITERIA

Different CPU scheduling algorithms have different properties, and the choice of a particular algorithm may favor one class of processes over another. In choosing which algorithm to use in a particular situation, we must consider the properties of the various algorithms. Many criteria have been suggested for comparing CPU scheduling algorithms. Which characteristics are used for comparison can make a substantial difference in which algorithm is judged to be best. The criteria include the following:

1. Utilization/Efficiency: keep the CPU busy 100% of the time with useful work

2. Throughput: maximize the number of jobs processed per hour.

3. Turnaround Time: from the time of submission to the time of completion, minimize the time batch users must wait for output

4. Waiting time: Sum of times spent in ready queue - Minimize this

5. Response Time: time from submission till the first response is produced, minimize response time for interactive users

6. Fairness: make sure each process gets a fair share of the CPU

CPU Scheduler whenever the CPU becomes idle; the operating system must select one of the processes in the ready queue to be executed. The selection process is carried out by the short term scheduler (or CPU scheduler). The scheduler selects from among the processes in the memory that are ready to execute and allocates the CPU to one of them

3. SCHEDULING ALGORITHMS:

1. FIRST COME FIRST SERVE (FCFS) SCHEDULING

It is simplest CPU scheduling algorithm. The job that requests the CPU first allots the CPU first. It means whoever comes first will be given the CPU first. The execution of the FCFS policy is easily managed with a FIFO queue. When a job comes in the ready queue, its PCB is linked onto tail of the queue. When the CPU is free, the job allots to the CPU at the head of queue. If job of higher burst time arrived before smaller burst time, afterwards the job queued in the ready queue, and wait for a long time, namely 'convoy effect'.

Characteristics

- The lack of prioritization does permit every process to eventually complete, hence no starvation.
- Turnaround time, waiting time and response time is high.
- One, Process with longest burst time can monopolize CPU, even if other process burst time is too short.

- Non preempted Shortest Job First.

2. SHORTEST JOB FIRST (SJF) SCHEDULING:

SJF means the shortest job will obtain CPU first. In FCFS, the one who comes first gets the CPU regardless the length of the job however, in this instance the length of the job matters.SJF According to the arrival time of the processes Based on the lowest CPU burst time (BT),Complexity More complex than FCFS, average waiting time Smaller than FCFS, performance Minimum Average Waiting Time.

3. PRIORITY SCHEDULING:

Each jobs have a specific priority and in accord with priority the scheduling takes place. The job whichever has high priority will be given CPU first. Equal priority jobs are scheduled in FCFS order. In such a circumstance, the short job has to wait for a long time if long job queued and that problem is starvation. The solution is aging; it increases the priority of jobs that are waiting for a long time. It can be both preemptive or non-preemptive. In preemptive case, when newly high priority job arrived, in comparison to currently executing in CPU, then CPU will carry away from currently executing job and will be given to newly arrived job with high priority. In non-preemptive case, newly arrived high priority job inserted at the head of the ready queue, after the execution of current job, that job will execute next.

According to the priority, the bigger priority task executes first. This type is less complex smaller than FCFS. This type has well performance but contain a starvation problem

Characteristics

- Starvation can happen to the low priority process.
- The waiting time gradually increases for the equal priority processes.
- Higher priority processes have smaller waiting time and response time.

4. SHORTEST REMAINING TIME FIRST (SRTF) SCHEDULING

The preemptive version of SJF, is SRTF. The jobs with the shortest remaining process time, is always selected by scheduler. It helps to improve the turnaround time. According to the arrival time of the processes the allocation of the CPU is based on the lowest CPU burst time (BT). But it is preemptive, more complex than FCFS and waiting time depending on some measures e.g., arrival time, process size, etc. The preference is given to the short jobs

Characteristics

- The real difficulty with the SJF algorithm is, to know the length of the next CPU request.
- SJF minimizes the average waiting time because it services small processes before it services large ones. While it minimizes average wait time, it may penalize processes with high service time requests. If the ready list is saturated, then processes with large service times tend to be left in the ready list while small processes receive service. In extreme case, when the system has little idle time, processes with large service time will never be served.

5. ROUND ROBIN (RR) SCHEDULING:

Round Robin scheduling work on the principle where each job gets an equal time to run. A small unit of time, termed as time quantum (usually from 10 to 100 milliseconds). Preemption added in FCFS scheduling is round robin. According to the order of the process arrives with fixed time quantum (TQ).The complexity depends on TQ size complexity is large as compared to SJF and Priority scheduling. Each process has given a fairly fixed time. According to the priority, the bigger priority task executes first. This type is less complex Smaller than FCFS and Well performance but contain a starvation problem.

Characteristics

Setting the quantum too short causes too many context switches and lower the CPU efficiency.

- Setting the quantum too long may cause poor response time and approximate FCFS.
- Because of high waiting times, deadlines are rarely met in a pure RR system.
- Setting the quantum too long may cause poor
- Setting the quantum too short causes too many

Figures of Gantt chart, Waiting Time, Average waiting Time, Average Turnaround time, Average of Response Time

Consider four set of processes with the CPU burst time in milliseconds is shown in table:

	Burst Time (ms)
P1	15
P2	6
P3	9
P4	7

Table 1: processes with Burst Time

1. First come First Serve

	P2	P3	P4

0 15 21 30 37

Fig.- Gantt chart of FCFS

2. Shortest Job First

P2	P4	P3	P1

0 6 13 22 37

Fig.- Gantt chart of SJF

3. Priority Scheduling:

P1	P2	P1	P3	P4	
0	4	11	13	22	37

Fig.- Gantt chart of PS

3. Round Robin

Time quantum is 6ms for each process:

P1	P2	P3	P4	P1	P3	P4	P1

0 6 12 18 24 30 33 34 37

Fig.- Gantt chart of Round Robin

4. Shortest Remaining Time First Scheduling

Process	Arrival Time	Burst Time
P1	0	15
P2	1	6
P3	2	9
P4	3	7

P1	P2	P3	P4	P3	P1

0 1 7 8 15 23 37

Fig.- Gantt chart of SRTF Scheduling

Conclusion:

This paper presented studies related to CPU scheduling algorithms. This study has shown that programming algorithms designed for different CPUs can reduce response time, waiting time, and overheads in CPU, disk, and memory usage and increase productivity. In general, we believe that a customized CPU algorithm improves CPU performance, just as results have shown that setting up a CPU algorithm helps reduce waiting times and response processes. Also, CPU algorithms allow the user to get better results without wasting valuable time. In the future, some researchers may do additional research and testing to determine which algorithms will be most appropriate and thus provide the complete satisfaction of the user. Each scheduling algorithm is designed to optimize result based on one metrics or the other. Which one out of these to actually implement in Operating system is a design question. This comparative study tends to find out which one performs better based on given metrics like throughput, CPU utilization, response time etc. Further study on optimizing the results based on types of processes and system type is needed.

References:

1. Galvin, P. B., Gagne, G., & Silberschatz, A. (2003). Operating system concepts. John Wiley & Sons.
2. Abbas, A. (2015). Multiprocessor and Real-Time Scheduling Shortest-Job-First (SJF) Scheduling Algorithm. European Journal of Computer Science and Information Technology, 3(5), 8-20.
3. Siahaan, A. P. U. (2016). Comparison analysis of CPU scheduling: FCFS, SJF and Round Robin. International Journal of Engineering Development and Research, 4(3), 124-132.
4. Zouaoui, S., Boussaid, L., & Mtibaa, A. (2019). Priority based round robin (PBRR) CPU scheduling algorithm. International Journal of Electrical & Computer Engineering (2088-8708), 9(1).Technologies and Optimization (Trends and Future Directions)(ICRITO) (pp. 397-400). IEEE.
5. Zouaoui, S., Boussaid, L., & Mtibaa, A. (2019). Priority based round robin (PBRR) CPU scheduling algorithm. International Journal
6. Electrical & Computer Engineering (2088-8708), 9(1).
7. Sabrian, F., C.D. Nguyen, S. Jha, D. Platt and F. Safaei, (2005). Processing resource scheduling in programmable networks. Computer communication, 28:676-68.
8. Umar Saleem and Muhammad Younus Javed, “Simulation of CPU Scheduling Alogrithm”,0-7803-6355-8/00/$10.00@2000 IEEE