International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 10, pp: 7-11 www.ijaem.net ISSN: 2395-5252

A Comprehensive Study of CPU Scheduling Algorithms

Keerth ana V, Gopika Meena S, Shri Haritha S, Dr.M. Sujithra M.C.A, M.Phil., PhD, Dr.A.D. Chitra M.C.A, M.Phil., PhD,
2nd Year, M.Sc. Software Systems (Integrated), Coimbatore Institute of Technology, Coimbatore
Assistant Professor, Department of Data Science, Coimbatore Institute of Technology, Coimbatore
Assistant Professor, Department of Software Systems, Coimbatore Institute of Technology, Coimbatore

Date of Submission: 20-11-2020 Date of Acceptance: 03-12-2020

ABSTRACT: In the operating system, Scheduling of CPU plays a vital role in designing. The main objective of our project is to compare different types of algorithms like First Come First Served Scheduling, Shortest Job First Scheduling, Priority Scheduling, Round Robin Scheduling, which helps us to improve the CPU efficiency and Timesharing in the operating system. This paper depicts the study of various scheduling algorithms for a CPU with the help of the Gantt chart.
Why CPU scheduling?
In the uni-programming systems, when a process waits for an I/O operation, the CPU remains idle. This is an overhead and it wastes the time and causes the problem of starvation. However, In Multiprogramming systems, the CPU doesn't remain idle during the waiting time of the Process and it executes other processes.
In Multiprogramming systems, the Operating system schedules the processes of the CPU to make maximum utilization of it and this is called CPU scheduling. The Operating System uses various scheduling algorithm to schedule the processes of the CPU.
The short-term scheduler schedules the task of CPU for the number of processes present in the Job Pool.
Whenever the running process requests some I/O operation then the short-term scheduler saves the present context of the method called PCB and changes its state from running to waiting. Now the process is in waiting state; the Short-term scheduler picks another process from the ready queue and assigns the CPU to the present process. This procedure is named context switching. In Multiprogramming, if the long-term scheduler has more I/O bound processes then most of the time, the CPU remains idle. The task of OS is to optimize the use of resources.
If most of the running processes change their state from running to waiting then there could also be an opportunity of deadlock within the system. Hence to scale back this overhead, the OS must schedule the roles to urge the optimal utilization of CPU and to avoid the likelihood to deadlock.

I. INTRODUCTION

Nowadays the operating system is more complex, they are working on multitasking, multiprocessing environment in which process is executed concurrently. 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 multiprocessing (transmit multiple flows simultaneously). There are several scheduling algorithms available as FCFS (First Come First Serve), SJF (Shortest Job First), Priority Scheduling, and RR (Round Robin) scheduling algorithm. There are some scheduling criteria, like turnaround time, response time, waiting time, and several contexts switching based on these criteria we analyze and determine which scheduling algorithm is best. In this paper, we introduce basic CPU-scheduling concepts and present several CPU-scheduling algorithms.

II. ORGANIZATION OF THE PAPER

(Section iii) describes the dispatcher (section iv) describes the types of scheduling (section v)
Describes the scheduling criteria (section vi) describes various CPU scheduling algorithms with their Gantt charts (section vii) contains the conclusion (section viii) provides the references

III. CPU SCHEDULING: DISPATCHER

The dispatcher is one of the components involved in the CPU scheduling function. The dispatcher is a module that gives control to the CPU process selected by the short-term scheduler. The functions are:

- Switching context
- Switching to user mode

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 10, pp: 7-11 www.ijaem.net ISSN: 2395-5252

- Jumping to the proper location in the user program to restart that program from where it left last time.

The dispatcher should be as fast as possible, as long as it's invoked during every process switch. The time taken by the dispatcher to prevent one process and start another process is Dispatch Latency. Dispatch Latency is often explained using the below figure:

IV. TYPES OF SCHEDULING.

4.2 Medium-Term Scheduling

Long-term scheduling performs like a gatekeeping function. It decides whether there is enough memory, to allow jobs into the system. It limits the process of multi-tasking to prevent slow performance on current-running programs. When a job gets into the long-term scheduler, it is sent on to the medium-term scheduler.

4.2 Medium-Term Scheduling

The medium-term scheduler decides to send a job to the sideline until a more important process is finalized. Later, when the computer has less important jobs, the medium-term scheduler will allow the suspended job to pass.

4.3 Short-Term Scheduling

The short-term scheduler takes jobs from the "ready" line and it will execute to run with a green light. The short-term scheduler runs the highest-priority jobs first and it makes on-the-spot decisions. For example, when a running process is interrupted it may be changed, the short-term scheduler must recalibrate and gives the green light for the highest-priority job.

V. 5.SCHEDULING CRITERIA

5.1 CPU Utilization

To make out the simplest use of CPU and to not waste any CPU cycle, CPU would be working most of the time(Ideally 100% of the time). Considering a true system, CPU usage should range from 40% (lightly loaded) to 90%(heavily loaded.)

5.2 Throughput

It is the entire number of processes completed per unit time or rather says the total amount of labor done in a unit of your time. this might range from 10/second to 1/hour counting on the precise processes.

5.3 Turnaround Time

It is the quantity of your time taken to execute a particular process, i.e. The interval from the time of submission of the method to the time of completion of the process (Wall clock time).

5.4 Waiting Time

The sum of the periods spent waiting within the ready queue amount of your time a process has been waiting within the ready queue to accumulate get control on the CPU.

5.5 Load Average

It is the typical number of processes residing in the ready queue expecting their address get into the CPU.

5.6 Response Time

The amount of your time it takes from when an invitation was submitted until the primary response is produced. Remember, it's the time till the primary response and not the completion of process execution (final response). In common CPU utilization and Throughput are maximized and other factors are reduced for proper optimization.

VI. SCHEDULING ALGORITHMS

To decide which process to execute first and which process to execute last to achieve maximum CPU utilization, computer scientists have defined some algorithms, they are:

- First Come First Serve (FCFS) Scheduling
- Shortest-Job-First (SJF) Scheduling
- Priority Scheduling
- Shortest Remaining Time First (SRTF) Scheduling
- Round Robin (RR) Scheduling
- Highest Response Ratio Next (HRRN) Scheduling
- Multilevel Feedback Queue Scheduling

6.1 First Come First Serve (FCFS) Scheduling

In the "First come first serve" scheduling algorithm, because the name suggests, the method which arrives first, gets executed first, or we will say that the method which requests the CPU first, gets the CPU allocated first.

- First Come First Serve, is simply like FIFO(First in First out) Queue arrangement,

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 10, pp: 7-11 www.ijaem.net ISSN: 2395-5252

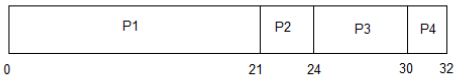
where the info element which is added to the queue first, is that the one who leaves the queue first.

- This is employed in Batch Systems.
- It's easy to know and implement programmatically, employing a Queue arrangement, where a replacement process enters through the tail of the queue, and therefore the scheduler selects a process from the top of the queue.
- Poor in performance because the average wait time is high.
- A perfect real-life example of FCFS scheduling is buying tickets at the ticket counter.

Consider the processes P1, P2, P3, P4 given in the below table, arrives for execution in the same order, with Arrival Time 0, and given Burst Time, let's find the average waiting time using the FCFS scheduling algorithm.

PROCESS	BURST TIME
P1	21
P2	3
P3	6
P4	2

The average waiting time will be = (0 + 21 + 24 + 30)/4 = 18.75 ms



This is the GANTT chart for the above processes

The average waiting time will be 18.75 ms
For the above-given processes, first P1 will be provided with the CPU resources,

- Hence, the waiting time for P1 will be 0
- P1 requires 21 ms for completion, hence waiting time for P2 will be 21 ms
- Similarly, the waiting time for process P3 will be the execution time of P1 + execution time for P2, which will be (21 + 3) ms = 24 ms.
- For process P4 it will be the sum of execution times of P1, P2, and P3.

The GANTT chart above perfectly represents the waiting time for each process.

6.2 Shortest-Job-First(SJF) Scheduling

Shortest Job First scheduling works on the method with the shortest burst time or duration first.This is the best approach to attenuate waiting time.This is employed in Batch Systems.It is of two types:

- Non-Preemptive
- Pre-emptive

To successfully implement it, the burst time/duration time of the processes should be known to the processor beforehand, which is practically not feasible all the time.This scheduling algorithm is perfect if all the jobs/processes are available at an equivalent time. (either time of arrival is 0 for all, or time of arrival is the same for all)

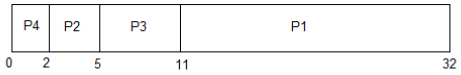
6.2.1 Non-Preemptive Shortest Job First

Consider the below processes available in the ready queue for execution, with arrival time as 0 for all and given burst times.

PROCESS	BURST TIME
P1	21
P2	3
P3	6
P4	2

In Shortest Job First Scheduling, the shortest Process is executed first.

Hence the GANTT chart will be following :



Now, the average waiting time will be = (0 + 2 + 5 + 11)/4 = 4.5 ms

As you can see in the GANTT chart above, the process P4 will be picked up first as it has the shortest burst time, then P2, followed by P3 and at last P1.
We scheduled the same set of processes using the First Come First Serve algorithm in the previous tutorial and got the average waiting time to be 18.75 ms, whereas, with SJF, the average waiting time comes out 4.5 ms.

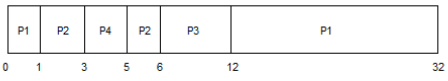
6.1.2 Pre-emptive Shortest Job First

In Preemptive Shortest Job First Scheduling, jobs are put into the ready queue as they arrive, but as a process with short burst time arrives, the existing process is preempted or removed from execution, and the shorter job is executed first.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 10, pp: 7-11 www.ijaem.net ISSN: 2395-5252

PROCESS	BURST TIME	ARRIVAL TIME
P1	21	0
P2	3	1
P3	6	2
P4	2	3

The GANTT chart for Preemptive Shortest Job First Scheduling will be,



The average waiting time will be, ((5-3) + (6-2) + (12-1))/4 = 4.25 ms

The average waiting time for preemptive shortest job first scheduling is less than both, non-preemptive SJF scheduling and FCFS scheduling.

As you can see in the GANTT chart above, as P1 arrives first, hence its execution starts immediately, but just after 1 ms, process P2 arrives with a burst time of 3 mswhich is less than the burst time of P1, hence the process P1(1 ms done, 20 ms left) is preempted and process P2 is executed.

As P2 is getting executed, after 1 ms, P3 arrives, but it has a burst time greater than that of P2, hence the execution of P2 continues. But after another millisecond, P4 arrives with a burst time of 2 ms, as a result, P2(2 ms done, 1 ms left) is preempted and P4 is executed.

After the completion of P4, process P2 is picked up and finishes, then P2 will get executed and at last P1.

The Pre-emptive SJF is also known as Shortest Remaining Time First because, at any given point of time, the job with the shortest remaining time is executed first.

6.3 Priority Scheduling

- Priority scheduling is a non-preemptive algorithm and one among the foremost common scheduling algorithms in batch systems.
- Each process is assigned a priority. The process with the highest priority is to be executed first then on.
- Processes with the same priority are executed on a first-come-first-served basis.
- Priority is often decided to support memory requirements, time requirements, or the other resource requirement.

Consider the below table for processes with their respective CPU burst times and the priorities. Let the arrival time be 0 for all the processes.

PROCESS	BURST TIME	PRIORITY
P1	21	2
P2	3	1
P3	6	4
P4	2	3

The GANTT chart for following processes based on Priority scheduling will be,

The average waiting time will be, (0 + 3 + 24 + 26)/4 = 13.25 ms

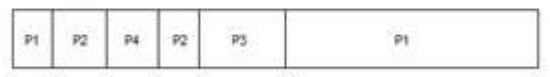
As you can see in the GANTT chart that the processes are given CPU time just based on the priorities.

6.4 Shortest Remaining Time

- Shortest remaining time (SRT) is that the preemptive version of the SJN algorithm.
- The processor is allocated to the work closest to completion but it is often preempted by a more ready job with a shorter time to completion.
- Impossible to implement in interactive systems where required CPU time isn't known.
- It is usually utilized in batch environments where short jobs got to give preference.

PROCESS	BURST TIME	ARRIVAL TIME
P1	21	0
P2	3	1
P3	6	2
P4	2	3

The GANTT chart for Preemptive Shortest



The average waiting time will be, ((5-3) + (6-2) + (12-1))/4 = 4.25 ms

6.5 Round Robin(RR) Scheduling

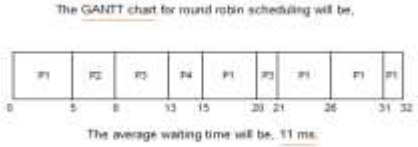
- Round Robin is the preemptive process scheduling algorithm.
- Each process is provided a fixed time to execute, it's called a quantum.
- Once a process is executed for a given time, it's preempted and another process executes for a given period.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 10, pp: 7-11 www.ijaem.net ISSN: 2395-5252

- Context switching is employed to save lots of states of preempted processes.

Consider all the process's arrival time is 0.

PROCESS	BURST TIME
P1	21
P2	3
P3	6
P4	2



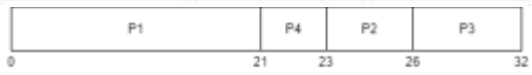
Here, the time quantum=5

6.6 Highest Response Ratio Next (HRRN)Scheduling

- Highest Response Ratio Next (HRNN) is one of the foremost optimal scheduling algorithms.
- This is a non-preemptive algorithm during which, the scheduling is completed on the idea of an additional parameter called Response Ratio.
- A Response Ratio is calculated for every one of the available jobs and therefore the Job with the very best response ratio is given priority over the others.
- Response Ratio is calculated by the given formula.Response Ratio = (W+S)/S

Where,
W → Waiting Time
S → Service Time or Burst Time

PROCESS	BURST TIME	ARRIVAL TIME
P1	21	0
P2	3	1
P3	6	2
P4	2	3



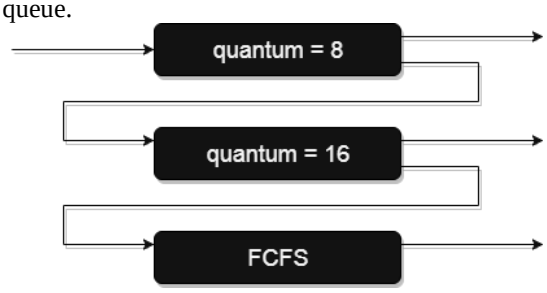
when P1 enters, it gets executed. P2, P3, and P4 will be waiting. After the execution of P1, the process with the highest response ratio must be executed.
RR(P2) = (21-1)+3/3=7.666
RR(P3) = (21-2)+6/6=4.166
RR(P4) = (21-3)+2/2=10
Hence, P4 will be executed next since it has the highest response ratio.
RR(P2) = (23-1)+3/3=8.333
RR(P3) = (23-2)+6/6=4.5
P2 will be executed and then P3 will be executed.

6.7 Multilevel Feedback Queue Scheduling

Multiple-level queues aren't an independent scheduling algorithm. they create the use of other existing algorithms to the group and schedule jobs with common characteristics.

- Multiple queues are maintained for processes with common characteristics.
- Each queue can have its scheduling algorithms.
- Priorities are assigned to each queue.

For example, CPU-bound jobs are often scheduled in one queue, and each one I/O-bound jobs in another queue. the tactic Scheduler then alternately selects jobs from each queue and assigns them to the CPU supported the algorithm assigned to the queue.



VII. CONCLUSION

The above mentioned are the different CPU scheduling utilized in this day. These algorithms are inherited supported by the need for the processor. Scheduling algorithms shouldn't affect the behavior of the system (same results no matter schedule). However, the algorithms do impact the system's efficiency and reaction time.

REFERENCE

[1]. William Stallings, "Operating systems Internals and Design Principles", 4th edition, PHI, 2001.
[2]. https://www.wikipedia.org/
[3]. Silberschatz A., Peterson J.L and Galvin P., "Operating System Concepts", John Wiley Publishing Company, 2002.
[4]. H.M.Deital, " An introduction to Operating System", Pearson Education, 2001
[5]. Charles Crowley, "Operating System a Design Oriented Approach", Tata McGraw Hill, 2000.
[6]. Milankovic M, "Operating System Concepts & Design", McGraw Hill,1999.
[7]. ArmassDanesl, "Mastering Linux", Premium Edition, BPB Publications, 1999.