A Survey Paper on CPU Process Scheduling

Kuldeep Vayadande[1], Sangam Patil[2], Sayee Chauhan[3], Rohit Thakur[4], Tanuj Baware[5], Samir Naik[6]

[1]VIT, Upper Indiranagar, Bibwewadi, Pune Maharashtra 411037, India

[1]kuldeep.vayadande1@vit.edu,

Abstract— This paper provides an extensive study related to CPU Process Scheduling by using Shortest Job First algorithm and Shortest Job Next algorithm. An operating system must employ process scheduling to ensure that processes run effectively and with fewer wait periods. The goals of process scheduling strategies are to make efficient use of CPU resources, boost throughput, shorten wait times, and quicken reaction and turnaround times. Processes and tasks are scheduled in order to complete the task on time. Scheduling the CPU is a procedure. One process may utilize the CPU, thanks to CPU scheduling. As a result of the lack of any resources, such as I/O, etc., another process is delayed (put on standby), fully utilizing the CPU. The primary goals of CPU scheduling is to increase the system's effectiveness, speed, and fairness. To get around the shortfalls of the Shortest Job Next (SJN) algorithm, SRTF (Shortest Remaining Time First) was created. The work that is closest to completion is given the processor in the SRTF (Shortest Remaining Time First) algorithm. In this paper we have discussed about CPU process Scheduling using different Scheduling Algorithms.

Keywords— CPU, scheduling, Shortest Job Next (SJN), and Shortest Remaining Time First (SRTF) (SJN)

1. INTRODUCTION

The activity of method, the method manager, which deals with the withdrawal of the active process from the mainframe as well as the selection of another process based on a particular strategy is an essential component of a concurrent execution. Such operational systems allow for the loading of just one method at a time into the working memory, and the loaded method also shares the mainframe by using temporal multiplexing.

Different Types of Scheduling

Scheduling falls into one of two categories:

Non-preemptive: In this case, a resource cannot be removed from a method until the method has finished running. Once the running method completes and transitions to a waiting state, the swap of resources takes place. Preemptive: In this case, the OS gives a method a specific amount of time to use the resources. The technique shifts between running state and prepared state during resource allocation, or from watching for the state to prepared state.

1. FCFS (First Come First Serve)

This is the only software system planning algorithm currently in use. The mainframe initial is assigned to the method that demands it. The Convoy effect and the much longer average waiting time as compared to the other scheduling algorithms are FCFS's key drawbacks.

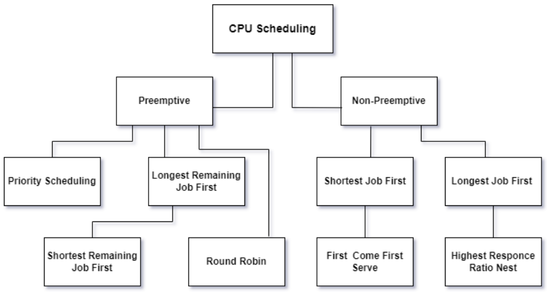

Figure 1 Types of CPU Scheduling

2. Shortest Job First (SJF):

This programming technique may choose the waiting procedure with the shortest execution time to run next. This programming method significantly shortens the usual waiting time for various processes that are waiting to die, whether or not it is preventative. The Shortest Job First is the SJF's overall kind. This strategy's drawback is that, SJF suffers from hunger. Additionally, it is frequently difficult to forecast the length of an impending CPU request in SJF.

3.Longest Job First (LJF):

As the name implies, this algorithmic rule is dependent on the real fact that the method with the longest burst time is processed initially. Longest Job First (LJF) is just the reverse of shortest job initial (SJF). Initial Longest Job is not preventive in nature. The LJF algorithm's high average waiting and turn-around times for a given collection of processes, which might cause a convoy effect, is a drawback of this technique.

4. Round Robin:

It is a CPU scheduling mechanism that cycles around assigning each task a specific time slot. It is the First come, First served CPU Scheduling technique with preemptive mode. The Round Robin CPU algorithm frequently emphasizes the Time-Sharing method.

5. Shortest Remaining Time First:

The preventive SJF programming algorithm is called SRTF (Shortest Remaining Time First). When using SRTF, a method's execution is halted after a specific amount of time. The brief computer hardware schedules the operating process with the least amount of remaining burst time from the list of available processes upon the arrival of each method. No preemption is performed once all processes have been unit tested in the prepared queue, allowing the algorithmic rule to function as SJF programming. When a method is stopped from being executed and a regular process follows, the method's context is kept in the method management block. On a subsequent execution of this procedure, the PCB is accessible. Given that its overhead costs are not taken into account, the SRTF algorithm processes tasks more quickly than the SJF method. In SRTF, the context transition occurs far more frequently than in SJF, taking up significant CPU processing time. This increases the processing time and lessens the benefit of quick processing. One of SRTF's benefits is that brief procedures are performed quickly.

1. Also, the system uses minimal overhead because decisions are only made when a process is finished or a new one is added.

2. SRTF has a few drawbacks, including the potential for process hunger like shortest job first.

If short processes are consistently introduced, large processes may be postponed forever.

2. RELATED WORK

A Shortest Job First Scheduling Algorithm for CPU Scheduling[1] The authors of this study contrasted Round Robin, Shortest Job Remaining First, and the suggested method. In comparison to Shortest Job First search, the suggested approach made a little adjustment to the average waiting time, which resulted in a respectable reduction in context switching.Preemptive Shortest Job First (SJF) Analysis in CPU Scheduling[2] The Preemptive Shortest Job First method is the subject of the author's case study in this essay, which is described with the aid of statistics, Gantt charts, and average waiting times."Shortest-Job-First (SJF) Scheduling Algorithm for Multiprocessor and Real-Time Scheduling"[3] The execution of the processors from less execution time to high execution time and determining the duration for each processor. In addition to this work, the author used SJF to 4 alternative CPU values and implemented them in a specific amount of time.A preemptive scheduling algorith for soft real-time systems called "Equitable Shortest Job First"[4] In this research, the authors suggested an algorithm that protects a process by prioritizing, addressing the issues associated with the lengthy process starvation in SJF. They also came to the conclusion that the suggested approach might be used to guarantee quality of service in soft real-time systems by integrating it into more complex scheduling algorithms.Round Robin, SJF, and FCFS CPU Scheduling Comparison and Analysis[5] In order to determine which algorithm is best for a certain process, the author of this research analysed three algorithms: Round Robin, SJF, and FCFS. Following all the comparisons, it was determined that FCFS performs best when used in short bursts of time. SJF performs better, however, when the process is run simultaneously on many processors. Last but not least, Round Robin can be used to change the standard waiting time."A Case of Backpack Production Scheduling Systems: Comparison of First In First Out with Shortest Job First in a Production Schedule Development"[6] The study found that, with an AWT difference of 68% and an ATT difference of 62%, SJF scheduling is preferred to FIFO scheduling. SJF is appropriate for situations when there is a build-up of processes concurrently or nearby. Whereas FIFO is used when the process arrival period is lengthy enough.

An innovative Round Robin and Dynamic Variable Quantum Time Task Scheduling method combination.[7] In this paper, the authors focused on separating the ready queue into sub-queues for lengthy and short jobs while also developing a hybrid algorithm based on dynamic quantum RR and SJF. The findings showed that the suggested algorithm was effective in minimizing turnaround and waiting time relative to reaction time, hence lowering the starvation of lengthy jobs.

3. COMPARISON TABLE

Table 1 Comparison Table

Sr no	Authors	Year	Title	Algorithm used	Remarks / Conclusions
1	Inayat ur-Rehman, Mamoona Humayun, Maryam, Hira Khurshid, Muhammad Akhtar, Bushra Hamid	2015	An Optimized CPU Scheduling Shortest Job First Scheduling Algorithm[1]	FIFO, Shortest Job Remaining First, Round Robin	The authors of this study contrasted , Shortest Job Remaining First, Round Robin and the suggested method. In comparison to Shortest Job First search, the suggested approach made a little adjustment to the average waiting time, which resulted in a respectable reduction in context switching.
2	Tri Dharma Putra	2020	CPU Scheduling Analysis of Preemptive Shortest Job First (SJF)[2]	Pre-emptive Shortest Job First	The Pre-emptive Shortest Job First, this method is the subject of the author's case study in this paper, which is described with the aid of statistics, Gantt charts, and average waiting times.
3	Adel Abdullah Abbas	2015	Real-time, Multiprocessor, and Shortest-Job-First (SJF) Scheduling Algorithm[3]	Non-Preemptive Shortest Job First and Preemptive Shortest Job First	The execution of the processors from less execution time to high execution time and determining the duration for each processor.

4	Mario Jean Rene, Dimitri Kagaris	2014	A preemptive scheduling algorithm for soft real-time systems called equitable shortest job first[4]	Enhanced Shortest Job First, Shortest Remaining Time, Highest Response Ratio Next,	In this research, the authors suggested an algorithm that protects a process by prioritizing, addressing the issues associated with the lengthy process starvation in SJF.
5	Andysah Putera Utama Siahaan	2016	FCFS, SJF, and Round Robin CPU Scheduling Comparison Analysis[5]	Round Robin,SJF,FCFS	In order to determine which algorithm is best for a certain process, the author of this research analysed three algorithms: Round Robin, SJF, and FCFS.

4. GRAPHS AND TABLES OF COMPARISON

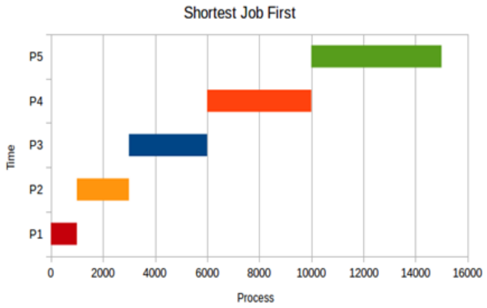

Figure 2 Shortest Job First

Table 2 Processes and their own burst and arrival time

Process	Burst Time	Arrival Time
J1	7	0
J2	3	1
J3	4	3

Gantt Chart for SRJF:

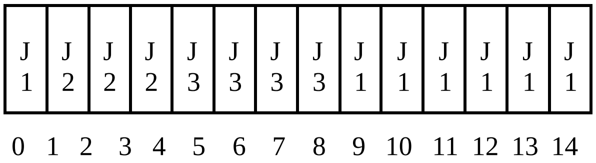

Figure 3 Gantt Chart

As the three processes P1, P2, and P3 arrive and burst at their respective times. Let's compute the waiting time, turnaround time, and completion time. Here, we may use: to determine Turn Around and Waiting Time:

1. Turnaround time equals arrival time minus completion time
2. Waiting Time = Turnaround Time-Burst Time

Table 3 Shortest Remaining Time First (Preemptive)

Process	**Arrival Time**	**Burst Time**	**Completion time**	**Turn Around Time**	**Waiting Time**
J1	0	7	14	14-0=14	14-7=7
J2	1	3	4	4-1=3	3-3=0
J3	3	4	8	8-3=5	5-4=1

Table 4 Average Waiting and Turnaround Time in MIPS

Amount Of tasks & VM					
Existing System/Algorithm	5	10	15	20	25
Policy (Default) Scheduling	4.1	8.2	10.6	13.8	17
SRJF	4.3	6.2	8.2	9.6	8.7

5. SCOPE OF IMPLEMENTATION

This review contributes to discuss various CPU-scheduling algorithms considering the various aspects like Waiting time, Turnaround time, Burst time and other field of scheduling. The analysis shows that no one technique is perfect. Each scheduling algorithm has different use cases and is useful in different domains based on its advantages. Thus, reviewing the limitations of each of the scheduling algorithm and also the main focus being the best and efficient usages of CPU we should use that scheduling algorithm, which best suits a particular operating system.

In this work, we explored several strategies for improving the accuracy and efficiency of the Shortest Job First Algorithm. Its two varieties are Non-Preemptive Shortest Job First and Preemptive Shortest Job First. The algorithm's major goal is to cut down on waiting times on average.

6. CONCLUSION

Scheduling algorithms' primary goals are to decrease resource depletion and ensure fairness among the parties using the resources. Determining which of the open requests will receive resources is the subject of scheduling. The system's behaviour shouldn't be impacted by the scheduling methods. The effectiveness and reaction time of the system are, however, affected by the scheduling algorithms. Adaptive systems are the finest. To perform at the highest level, we need to be able to foresee what would be the most appropriate method for next execution. The Greedy Algorithm is The Shortest Remaining Time First. The main problem with this approach is that if shorter processes keep appearing, it can lead to starvation. The idea of ageing can be used to overcome this issue. It is basically impossible since the operating system might not be aware of the burst time and would not be able to sort them.

REFERENCES

[1] Muhammad Akhtar, Bushra Hamid, Inayat ur-Rehman, Mamoona Humayun, Maryam, Hira Khurshid,An Optimized Shortest Job First Scheduling Algorithm for CPU Scheduling

[2] Tri Dharma Putra, Analysis of Preemptive Shortest Job First(SJF) in CPU Scheduling

[3]Adel Abdullah Abbas, Multiprocessor and Real-time Scheduling Shortest-Job-First(SJF) Scheduling Algorithm

[4] Mario Jean Rene,Dimitri Kagaris, Equitable Shortest Job First: A Preemptive Scheduling Algorithm for Soft Real-Time Systems

[5] Andysah Putera Utama Siahaan, Comparison Analysis of CPU Scheduling : FCFS,SJF,Round Robin

[6] Iqbal Abdul Jabbar,Ade Kania Ningsih,Faiza Renaldi, Comparison of First In First Out with Shortest Job First in a Production Schedule Development: A Case of Backpack Production Scheduling Systems

[7] Samir Elmougy,Shahenda Sarhan,Manar Joundy, A novel hybrid of Shortest job first and Round Robin with dynamic variable quantum time task scheduling technique

[8] https://m.youtube.com/watch?v=EWkQl0n0w5M

[9] https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/amp/

[10]https://www.researchgate.net/publication/360783034_Simulation_of_Preemptive_Shortest_Job_First_Algoritm

[11]https://www.researchgate.net/publication/351492569_Priority-based_Shortest_Job_First_Broker_Policy_for_Cloud_Computing_Environments

[12] https://www.researchgate.net/publication/234556241_OPTIMIZED_SOLUTION_TO_SHORTEST_JOB_FIRST_BY_ELIMINATING_THESTARVATION

[13] https://ieeexplore.ieee.org/document/6601295

[14] https://ieeexplore.ieee.org/document/7113202

Biographies

Kuldeep Vayadande is working as Assistant Professor in Dept. Of AI and DS. He has completed PhD in Computers Science and Engineering and having 14 Years of Teaching Experience. Published various papers in International Journals. His area of Specialization includes Operating System, System Programming, Information Security, Automata Theory and Cloud Computing. He is also working as Reviewer for various International Journals.

Sayee Chauhan studying the bachelor's degree in Artificial Intelligence & Data Science from Vishwakarma Institute of Technology, Pune. Her research areas include machine learning, deep learning, Operating System.

Sangam Patil studying the bachelor's degree in Artificial Intelligence & Data Science from Vishwakarma Institute of Technology, Pune. He is a Machine Learning Engineer and Data Scientist. His research areas include machine learning, deep learning, blockchain and cyber Security.

Rohit Thakur studying the bachelor's degree in Artificial Intelligence & Data Science from Vishwakarma Institute of Technology, Pune. His research areas include machine learning, deep learning.

Tanuj Baware studying the bachelor's degree in Artificial Intelligence & Data Science from Vishwakarma Institute of Technology, Pune. He is a Bug-Bunty Hunter. His research areas include blockchain and cyber Security.

Sameer Naik studying the bachelor's degree in Artificial Intelligence & Data Science from Vishwakarma Institute of Technology, Pune. His research areas include machine learning, cyber Security.