IJARCCE
ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⋇ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.11501

Simulation of Preemptive Shortest Job First Algorithm

Rakhmat Purnomo, Tri Dharma Putra*

***Correspondence Author**

Department of Informatics, Faculty of Computer Science, University of Bhayangkara Jakarta Raya,

Jalan Perjuangan Bekasi Utara, Indonesia

Abstract: With simulation, we imitate the operation of a real world process or system over time. It requires the use of models; model that represent the behavior or characteristics of the selected process or system. Computer are used to execute the simulation. In operating systems, OS-SIM is one of simulation application to represent the system characteristic or behavior of any process scheduling algorithm. Several scheduling algorithms in operating systems are available. There are preemptive shortest job first scheduling algorithm and non-preemptive shortest job first scheduling algorithm in operating system. Preemptive shortest job first scheduling algorithm is where when the shortest process arrives and is positioned at the head of the queue and interrupted the longer process. In this journal, we simulate preemptive shortest job first (SJF) algorithm with OS-SIM. Case study is discussed to understand the simulation thoroughly.

Keywords: Preemptive Shortest Job First, Scheduling Algorithm, OS-SIM

I. INTRODUCTION

Operating System (OS) is software which its role is as an interface between a user and the computer hardware. OS is known as a resource manager because its main duty is to manage the resources of computer system. Scheduling is one fundamental and most important design (Putra, 2020). Scheduling refers to set of rules, policies and mechanism that govern the order in which resources is allocated to various processes and the work is to be done. The scheduling is a methodology of managing many queues of processes in order to make delay minimum and to make performance optimal of the system. A scheduler is a module in operating system that implements the scheduling policies. Its main objective is to make systems performance's optimal that match with the criteria set by the system designer (Pawan Singh, Amit Pandey, 2015).

Scheduling is a prime concept in multiprocessing and multitasking of OS design and in real-time operating system design by arranging switching in the CPU among process. Premptive Shortest Job First (SJF) Algorithm is a well known algorithm in CPU processing. A scheduler is an OS module that implements the scheduling policies. Its primary objective is to optimize the systems performance according to criteria set by the system designer (Shweta Jain, 2016)

A computer simulation is an attempt to model a real-life or hypothetical situation on a computer so that it can be studied to see how the system works. By changing variables in the simulation, predictions may be made about the behavior of the system. It is a tool to virtually investigate the behavior of the system under study. Computer simulation in operating systems has become a useful part of modelling many scheduling algorithms. But also in other area of science like in physics, chemistry and biology and human systems in economics and social science (e.g., computational sociology) as well as in engineering computer simulation. Simulation is used to gain insight into the operation of those systems. A good example of the usefulness of using computers to simulate can be found in the field of operating systems with OS-SIM. In such simulations, the model behavior of file, disk, memory management, process scheduling simulation can be implemented (Wikipedia, 2022).

With simulation, we imitate the operation of a real world process or system over time. It requires the use of models; model that represent the behavior or characteristics of the selected process or system. Computer are used to execute the simulation. OS-SIM is one of simulation application to represent the system characteristics or behaviors of Preemptive Shortest Job First (SJF) algorithm (OS Concep Simulator, n.d.).

The Operating System Simulator (OS-SIM) is designed to support two main aspects of a computer system's resource management: process management and memory management. Once a compiled code is loaded in CPU Simulator's memory, its image is also available to the OS Simulator. It is then possible to create multiple instances of the program images as separate processes. The OS Simulator displays the running processes, the ready processes and the waiting processes. Each process is assigned a separate Process Control Block (PCB) that contains information on the process state. This information is displayed in a separate window. The main memory display demonstrates the dynamic nature of page allocations according to the currently selected placement policy. The OS maintains a separate page table for each process which can also be displayed. The simulator demonstrates how data memory is relocated and the page tables are maintained as the pages are moved in and out of the main memory illustrating virtual memory activity . In terms of process scheduling, OS-SIM can simulate all scheduling algorithms like FCFS, priority, round robin, and shortest job first. Modelling these algorithms is the main objective of OS-SIM.

Preemptive SJF scheduling algorithm is usually used among various other algorithms for scheduling CPU, however it can make the problem of starvation which happens when processes with longer burst time are not given any chance of CPU utilization due to prolong CPU usage by processes with shorter burst time.

OS Sim (Operating System Concepts Simulator) is an educational purpose application to graphically simulate Operating System concepts and support the computer science students learning process (OS Concep Simulator, n.d.) .

II. OVERVIEW OF EXISTING ALGORITHMS

There are several scheduling algorithms proposed by experts, namely:

1. **Shortest Job First (SJF)**. This one is a well known scheduling algorithm, the system will choose the shortest job of process to be executed first. That's why it is named Shortest Job First (SJF). One main weakness of this algorithm is that starvation could happen, the long process will never be executed because there are still shorter jobs exist in the system. If shorter job still keeps coming in then starvation will occur (Putra, 2020). SJF algorithm is one optimal algorithm. It executes the short process before the long process and thus reduces the waiting time for short process more than increases waiting time for long process. Which finally ends up with minimum average waiting time compared to the other scheduling algorithm (Asma Joshita Trisha, 2019).
2. **Round Robin (RR)**. Round robin algorithm is a real-time scheduling algorithm in operating system. The Round Robin scheduling algorithm is cited as standard Round Robin and it is a preemptive type that allocates a slice of context switching. Whenever time slice or context switching completes the current process is preempted and put in the rear of ready queue. Round robin algorithm is usually applied in real-time and time-sharing operating system because it provides every process an average share of time to utilize the CPU and gives a small responds time. However, the standard round robin algorithm has many weaknesses such as small throughput and big turnaround time as well as the big waiting time and also huge context switches number (Hoger K. Omar, Kamal H. Jihad, n.d.). In advanced round robin, a dynamic time slice is used to find the effective algorithm that is running in the system. The intelligent time slice (quantum) for round robin architecture for real time operating systems is a modified version of simple round robin scheduling (Putra, 2020).
3. **Priority Scheduling**. In priority scheduling, the idea behind this algorithm is straightforward, each process is allocated a priority. The equal process priority are scheduled in first come first serve basis. SJF is one example. SJF has the same idea as Priority Scheduling. Namely, the longer the burst of CPU, then it makes the lower the priority and the smaller the burst time, the higher the priority (Kunal Chandiramani, Rishabh Verma, 2019). Priority can be defined internally or externally. Internally defined priorities use some measurable quantities or qualities to computer priority of a process (Chandra Shekar N, 2017). Priority scheduling algorithm manages processes in its queue based on its priority. Something else that gives priority on running state is preemption (Ledina Hoxha Karteri, 2015).
4. **First Come First Serve (FCFS).** This is a standard algorithm. The process which arrives first will be executed first. That is why it is called first come first serve. The first process that exist will be executed first. This is the same as the concept of First In First Out (FIFO) (Kunal Chandiramani, Rishabh Verma, 2019). This is the simplest scheduling algorithm. While the processes in the ready queue will occupy the CPU in the order of their arrival to the ready queue. The process which enters first in the ready queue will occupy the CPU first and the process which enters afterwards will occupy CPU sequentially in the arrival order. This is a non-preemption scheduling algorithm. Once the process is allocated in the ready queue, it will be release by the CPU until it is terminated. The weaknesses of this algorithm is a high average waiting time (Pawan Singh, Amit Pandey, 2015) (Siahaan, 2016) (Hoger K. Omar, Kamal H. Jihad, n.d.).

III. ORGANIZATION STRUCTURE

This journal is divided to be six chapters. The first chapter is introduction. In this chapter, we discuss the idea behind the the CPU scheduling algorithm and about CPU simulation. In the second chapter we discuss the overview of existing algorithms. Discussion about algorithm like shortest job first, round robin, priority algorithm, first come first serve. The

DOI: 10.17148/IJARCCE.2022.11501

third chapter is explanation about organization structure of this journal. The fourth chapter is about OS-SIM Simulator, Chapter five is about case study analysis of simulation of preemptive shortest job first algorithm. Here we discuss one case study simulation. The last chapter is conclusion and future work.

IV. OS-SIM

Operating Systems concepts are best learned through implementation. However, this can be difficult and time consuming. To support this learning process an OS simulator has been developed that allows students to learn about OS concepts using this simulator, OS-SIM.

The OS-SIM can be used to explore the behavior of the operating system. For example, the user can step through each process to see when it has access to the processor, how long it has had access, and where in memory it is located. The use of memory is visually depicted in the simulator based upon which memory allocation method the user chooses.This modular approach permits the user to experiment with different implementations of different algorithms. Through this experimentation the user is able to gain a better understanding of the behavior and performance characteristics of the OS element under consideration.

Figure 1. below is the dashboard display of OS-SIM Simulator. There are five pull-down menus, namely: File, processes, Memory, File System, Disk, and Help. Here we can see four major divisions of the operating system to be simulated, namely, process scheduling, disk management, memory management, and file management. File Menu is for arranging file. Menu Processes is about process scheduling manipulation, we will use this a lot in this simulation. This is our main focus. Menu Memory is for memory management simulation. Disk is for disk simulation and the last menu, Help, is for information and help about this simulator.

FIGURE 1. OS-SIM SIMULATOR

In figure 2 below, we can see here the simulation of first come first serve algorithm. We will get into deep on process scheduling. Here we can manipulate the processes scheduling of the systems.

There are four algorithms that can be simulated: First Come First Serve (FCFS), Shortest Job First (SFJ), Priority, Round Robin. All can be made preemptive or non preemptive algorithm.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.11501

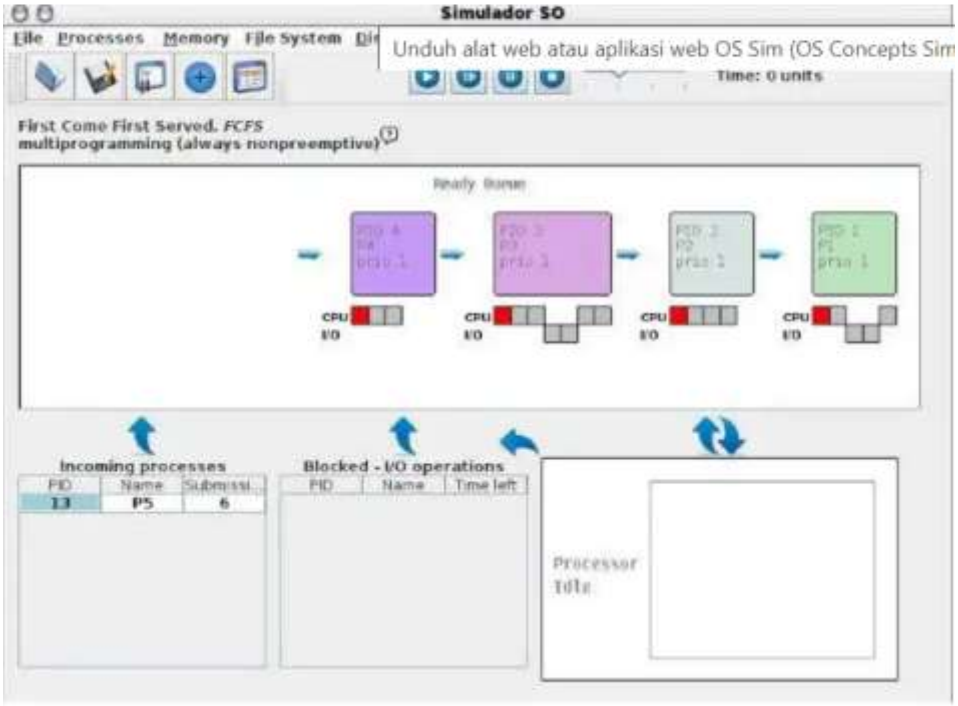


FIGURE 2. PROCESS SCHEDULING ALGORITHM SIMULATION

In figure 3, we can see the simulation of Non Contiguous Memory Management, with Pagination Page: 2 unit

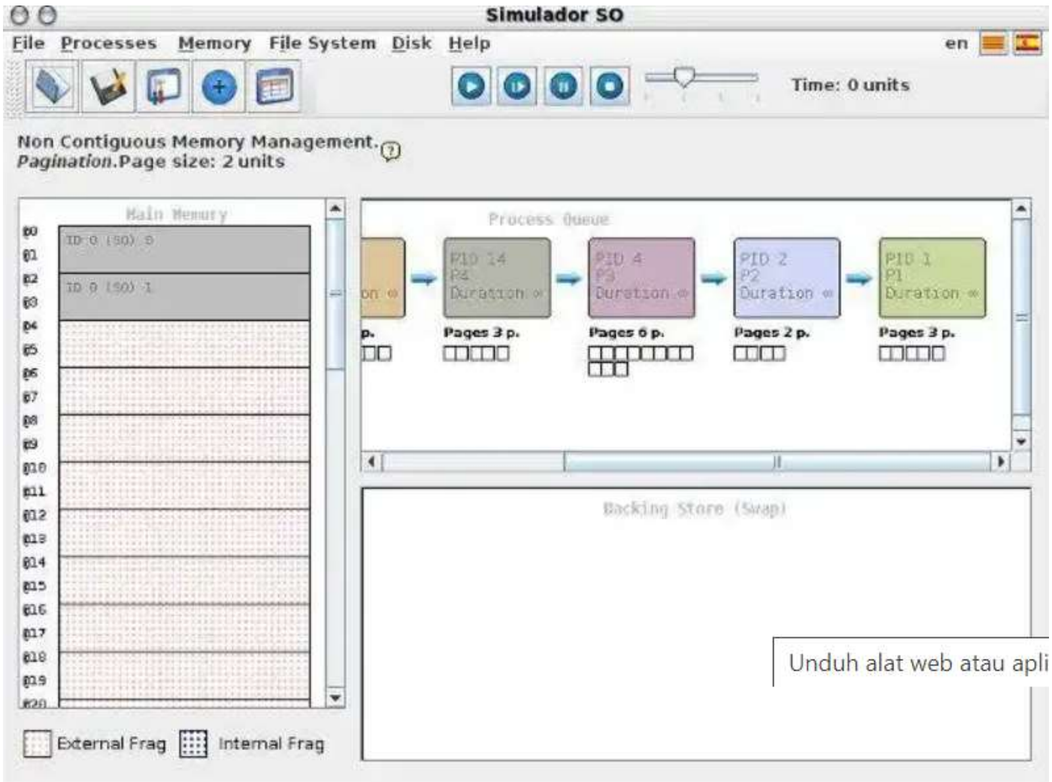


FIGURE 3. MEMORY MANAGEMENT

In figure 4, we can see the display of file allocation table, the cluster, the sectors, tracks, clusters. For saving the data in the disk.

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⋇ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.11501

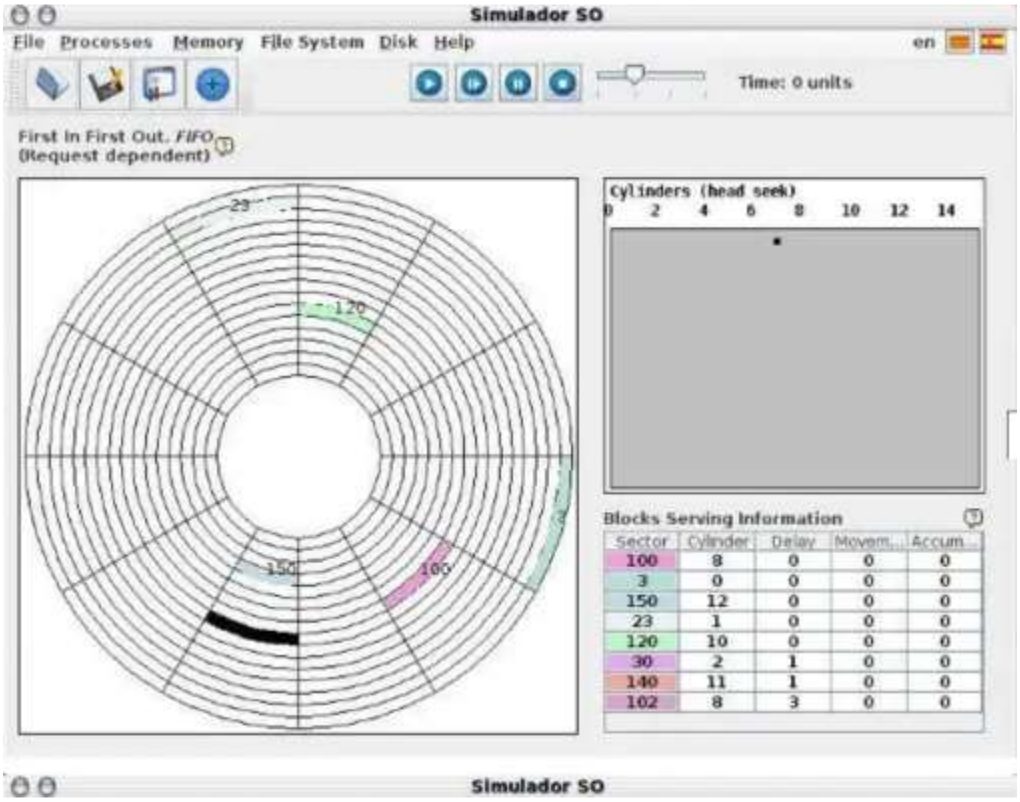


FIGURE 4. DISK ALLOCATION IN SECTORS, TRACKS, AND CLUSTER.

In figure 5 below, we can see the simulation of file management in blocks.

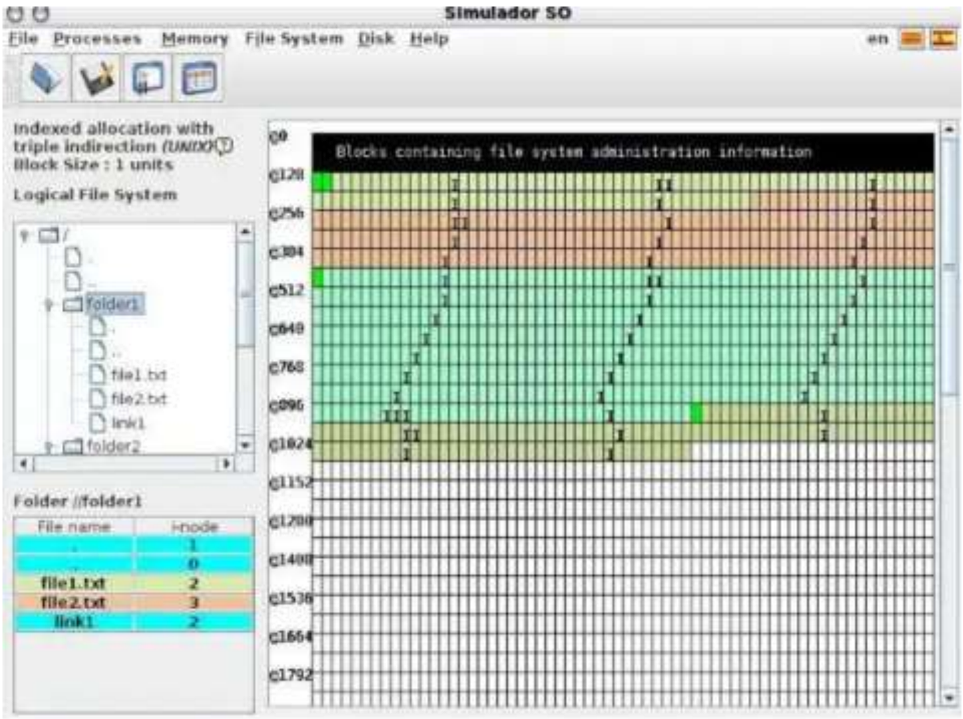


FIGURE 5. FILE MANAGEMENT

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.11501

In this simulator we can see the virtual representation of the systems process scheduling, that is behind the scene. With it we can describe and understand the process work clearly. Another advantage of this OS-SIM is that we can explore more deeply into the scheduling process of the systems. And manipulate it, making it easier to be understood.

V. CASE STUDY ANALYSIS

In this chapter we will discuss one case study about preemptive shortest job first using OS-SIM. There will be four processes. Each with burst time and arrival time. The discussion is as follows:

Here we have four processes, Please take a look on table 1, we have four processes namely A, B, C, and D. Also given here different arrival times, with different burst times each.

TABLE 1. CASE STUDY

Process	Arrival Time	Burst Time
A	0	4
B	1	2
C	2	2
D	4	2

Please take a look on figure 6. The table of table 1. is implemented as shown in figure 6. on OS-SIM.

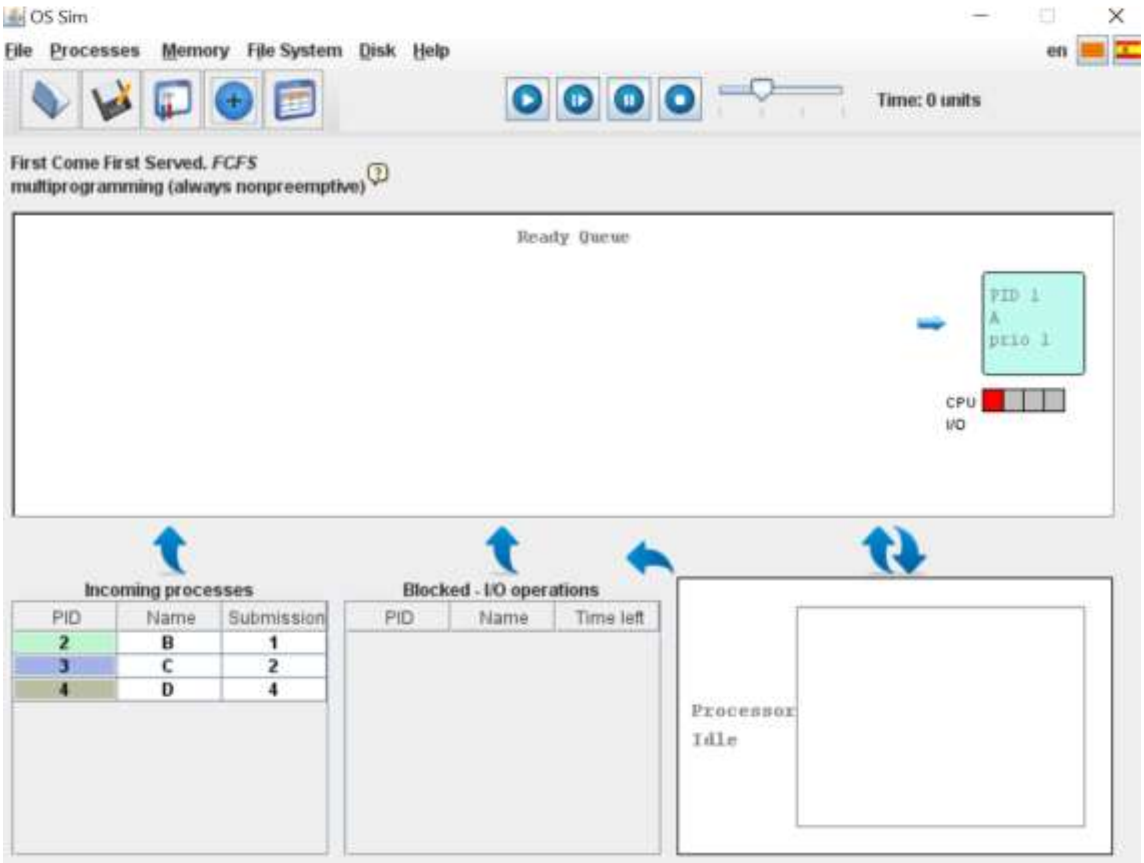


FIGURE 6. SIMULATION IMPLEMENTATION OF TABLE 1.

At t=0, A arrives with burst time 4. But at t=1, B interrupted A. The system compares the burst time of A and B. A is 3 ms left, but B is still 2 ms. Then at t=1, B interrupted A to be executed. As shown in figure 7.

 ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⋇ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.11501

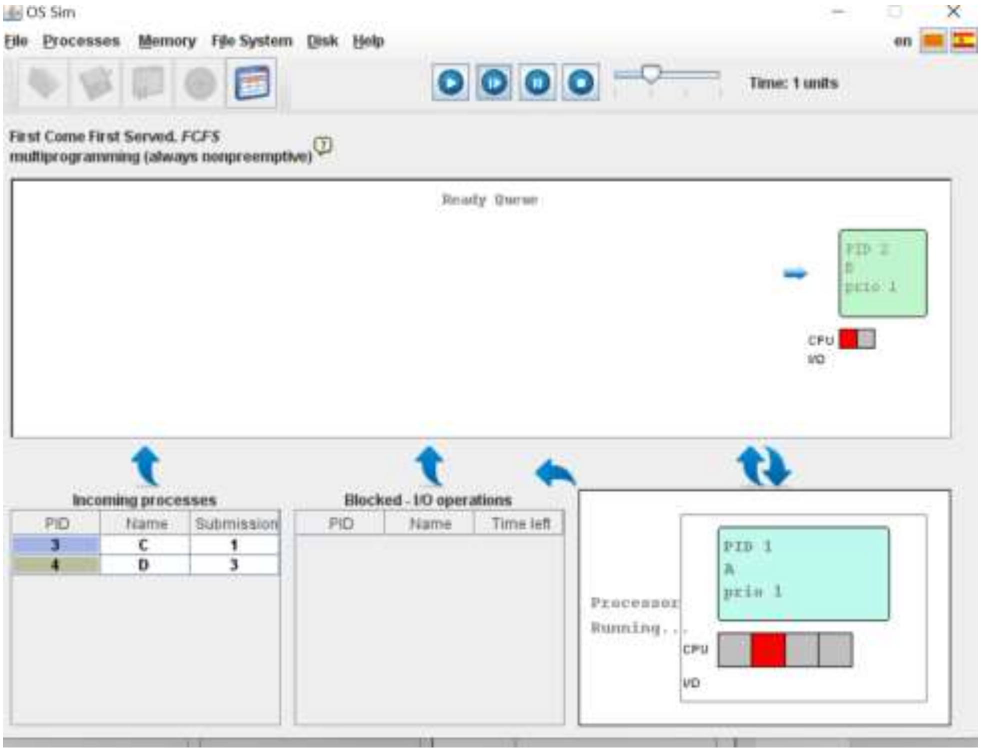


FIGURE 7. STEP BY STEP SIMULATION IMPLEMENTATION AT T=2

At t=2 C gets in. As shown in figure 8. B is compared with C. B is one ms left. But C is still 2 ms left. So that the systems continues execute B until B is finished.

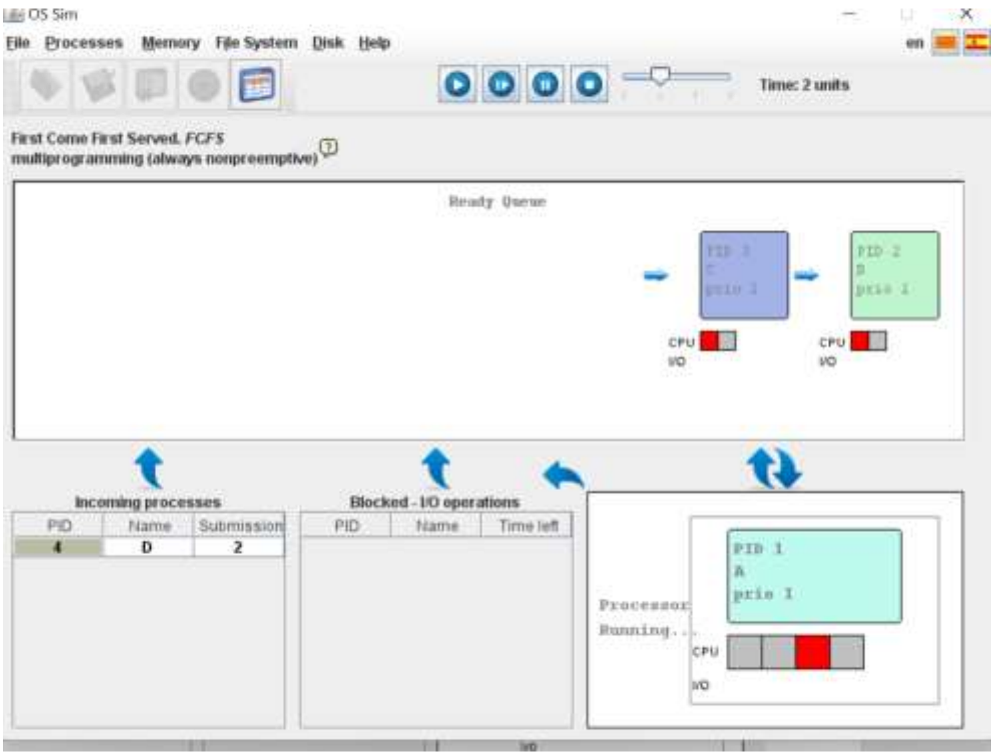


FIGURE 8. STEP BY STEP SIMULATION IMPLEMENTATION AT T=3

Please take a look on figure 9. At t=3, C gets in. C will be compared with D at t=4. Since process C is shorter than D, then C id executed first.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 7.39 ⁜ Vol. 11, Issue 5, May 2022
DOI: 10.17148/IJARCCE.2022.11501

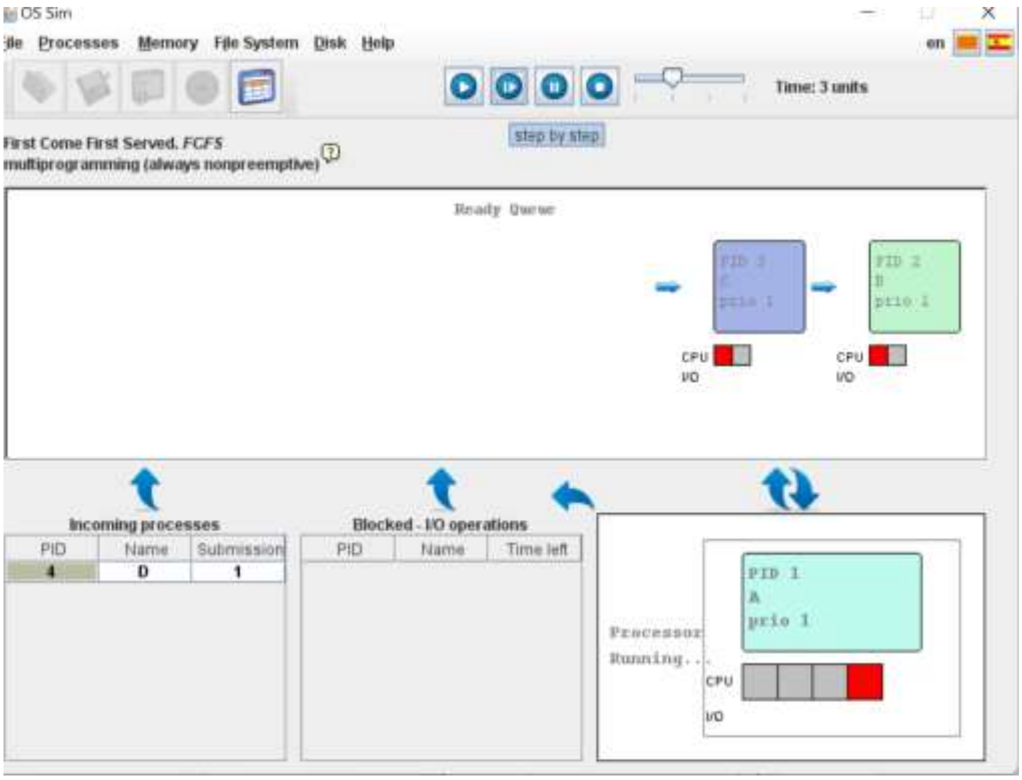


FIGURE 9. STEP BY STEP SIMULATION IMPLEMENTATION AT T=4

At t=4, C is compared with D. C is 1 ms left. But D is still 4 ms. Then the shortest process is still C. Then C continues until finish at t=5. Please take a look on figure 10.

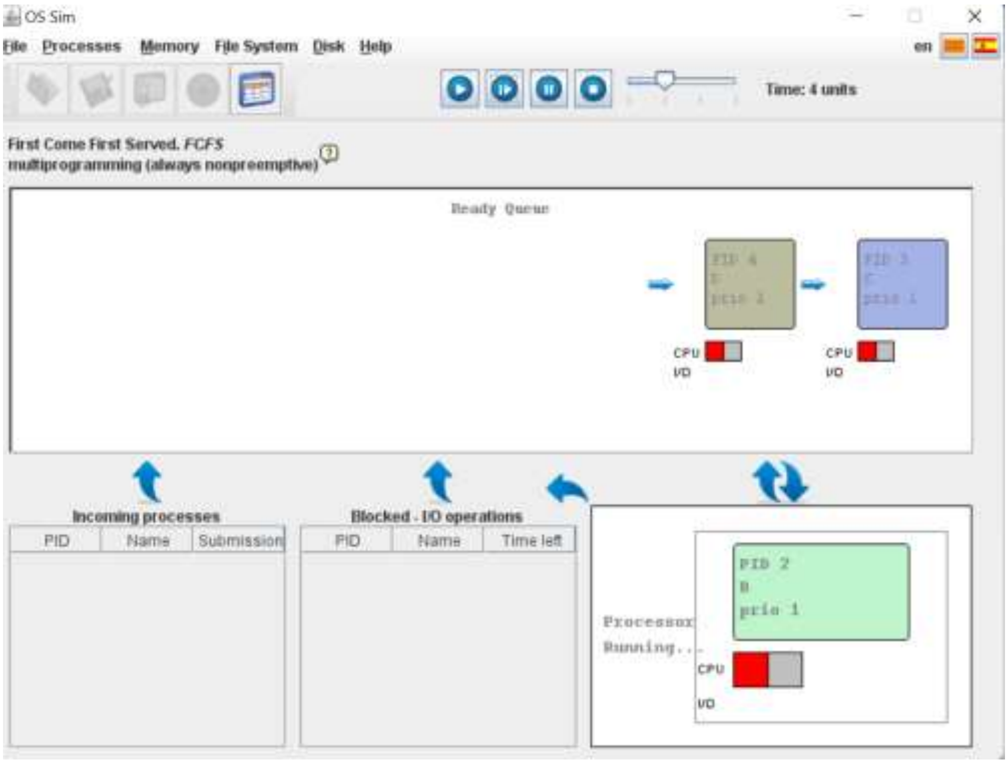


FIGURE 10. STEP BY STEP SIMULATION IMPLEMENTATION ATR T =5

At t=5, C is finishes. Then A is compared with D. A is 3 ms, but D is 2 ms. So the systems executes D until D is finish for 2 ms. Please take a look on figure 11.

DOI: 10.17148/IJARCCE.2022.11501

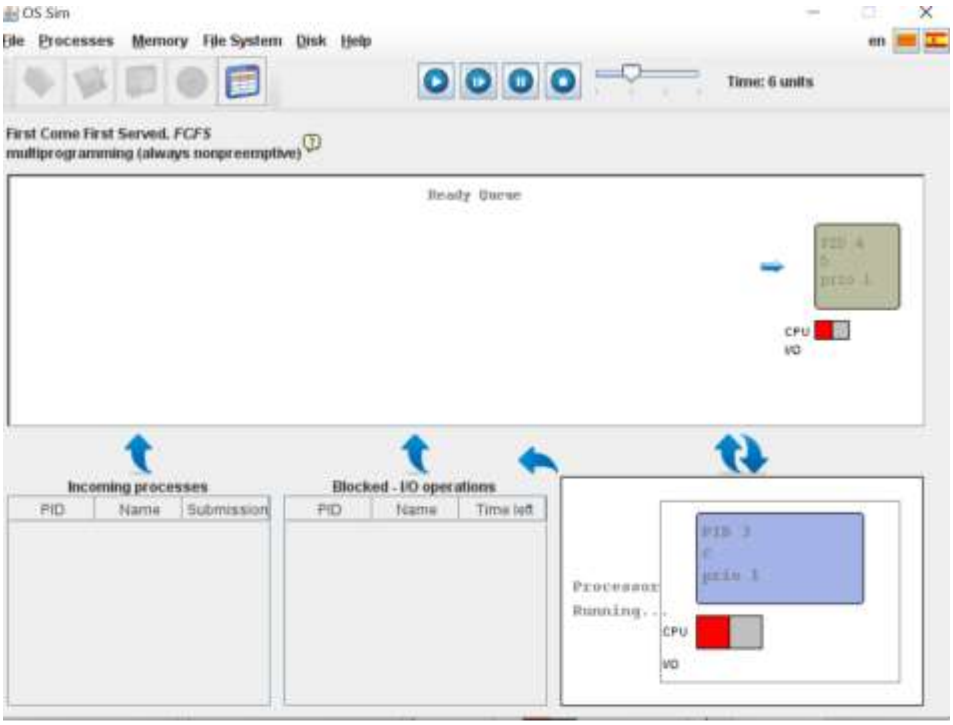


FIGURE 11. STEP BY STEP SIMULATION IMPLEMENTATION AT T = 6

At t=7 A is executed. Please take a look on figure 12. It is left 3 ms for A. Then A is executed until finish at t= 10. A is the last process to be executed.

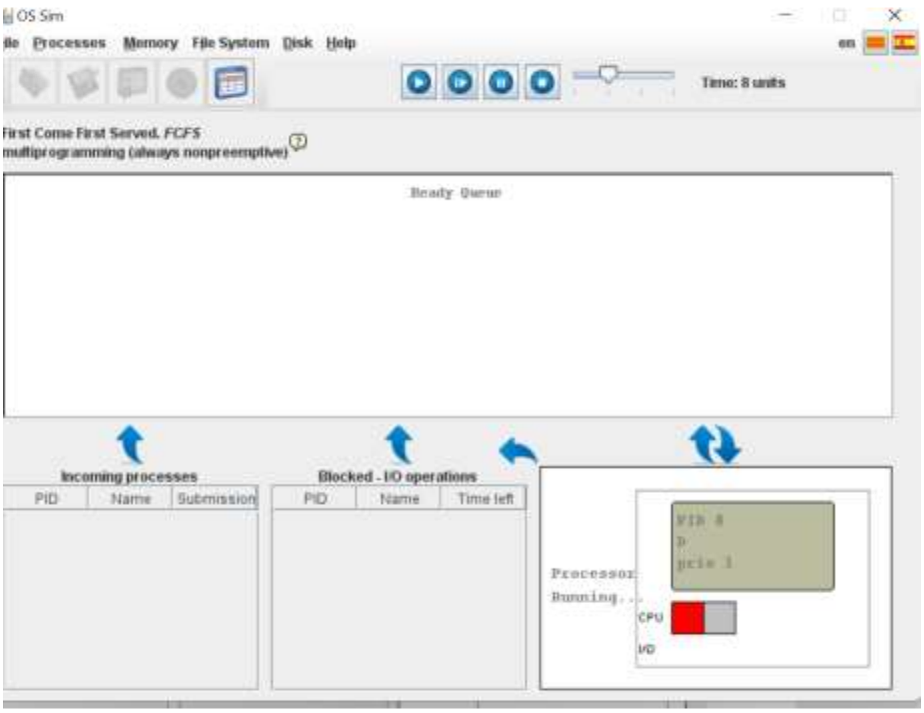
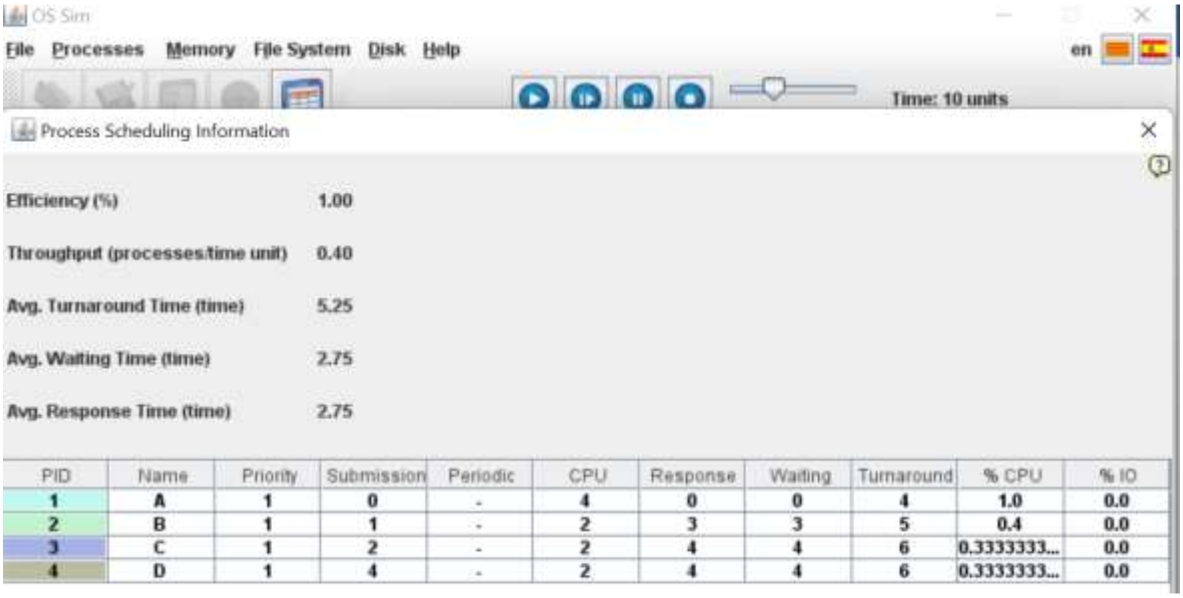


FIGURE 12. STEP BY STEP SIMULATION IMPLEMENTATION AT T=7

Based on the analysis above by the OS-SIM, we get the table analysis of systems simulation. Average turnaround time is 5.25 ms. Average waiting time is 2.75 ms. Average respons time is 2.75. In this table we can see also the burst time, the process names, arrival times, and efficiency. As we can take a look on the Table 2. below:

DOI: 10.17148/IJARCCE.2022.11501

TABLE 2. ANALYSIS OF SYSTEM'S SIMULATION



Efficiency (%)	1.00
Throughput (processes/time unit)	0.40
Avg. Turnaround Time (time)	5.25
Avg. Waiting Time (time)	2.75
Avg. Response Time (time)	2.75

PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
1	A	1	0	-	4	0	0	4	1.0	0.0
2	B	1	1	-	2	3	3	5	0.4	0.0
3	C	1	2	-	2	4	4	6	0.3333333...	0.0
4	D	1	4	-	2	4	4	6	0.3333333...	0.0

Below is Gantt Chart for the processes:

A	B	B	C	C	D	D	A	A	A

0 1 2 3 4 5 6 7 8 9 10

FIGURE 13. GANTT CHART OF THE PROCESSES

.

From the gantt chart above, we can conclude that A is the last time to be executed. Since it is the longest process to be executed. A is executed two times, at t=0 until t=1 and t=7 until t=10. A is also executed the last time. B is executed at t=1 until t=3. C is executed at t=3 until t=5. D is executed at t=5 until t=7. This happened because of different arrival times of the processes.

VI. CONCLUSION

The objective of this journal is to simulate Preemptive Shortest Job First (SJF) Scheduling Algorithm in CPU. The calculation of one case study shows that preemptive SJF process scheduling has 2.75 ms average waiting time The average turn around time is 5.25 ms. Average response time is 2.75. This case study gives understanding about the preemptive SJF process scheduling more thoroughly. The preemptive priority scheduling will interrupt the process if the running process has shorter best time than the ready process. The processes in case study finishes after 10 ms. OS-SIM is one well known simulator that represent the behaviors of scheduling processes. By this simulator, we can simulate the processes scheduling that is behind the scene. For future works, I propose to simulate other existing algorithms like round robin or priority with OS-SIM and compared the results.

REFERENCES

1. Asma Joshita Trisha, S. B. (2019). A Combined Preemptive SJF and Preemptive Priority Algorithm to Enhance CPU Utilization. International Journal of Computer Applications, 177(19), 26–30.
2. Chandra Shekar N, K. V. (2017). Analysis of Priority Scheduling Algorithm on the Basis of FCSF & SJF for Similar Priority Jobs. International Journal of Engineering Research in Computer Science and Engineering, 4(3), 73–76.
3. Hoger K. Omar, Kamal H. Jihad, S. F. H. (n.d.). Comparative Analysis of the Essential CPU Scheduling Algorithms. Bulletin of Electrical Engineering and Informatics, 10(5), 2742–2750.
4. Kunal Chandiramani, Rishabh Verma, S. M. (2019). A Modified Priority Preemptive Algorithm for CPU Scheduling. International Confierence on Recent Trends in Advanced Computing 2019, ICRTAC 2019, 363–369.
5. Ledina Hoxha Karteri, A. S. (2015). Preemptive and Non- Preemptive Priority Scheduling. International Journal of Computer Science & Management Studies, 19(01), 1–5. www.ijcsms.com
6. OS Concep Simulator. (n.d.). Retrieved April 12, 2022, from https://sourceforge.net/projects/oscsimulator/
7. Pawan Singh, Amit Pandey, A. M. (2015). Varying Response Ratio Priority: A Preemptive CPU Scheduling

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 7.39 ⸎ Vol. 11, Issue 5, May 2022
DOI: 10.17148/IJARCCE.2022.11501

Algorithm (VRRP). Journal of Computer and Communications, 3(April), 40–51.
8. Putra, T. D. (2020). Analysis of Preemptive Shortest Job First (SJF) Algorithm in CPU Scheduling. IJARCCE, 9(4), 41–45. https://doi.org/10.17148/ijarcce.2020.9408
9. Shweta Jain, S. J. (2016). A Review Study on CPU Scheduling Algorithm. International Journal of Advanced Research in Computer and Communication Engineering, 5(8), 45–50.
10. Siahaan, A. P. U. (2016). Comparison Analysis of CPU Scheduling: FCFS, SJF and Round Robin. International Journal of Engineering Development and Research, 4(3), 124–131.
11. Wikipedia. (2022). No Title. https://en.wikipedia.org/wiki/Simulation#Computer_simulation