Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, Juli 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13352 p-ISSN : 2541-044X

Simulation and Modelling of Pre-emptive Priority CPU Scheduling Algorithm

Tri Dharma Putra[1], Rakhmat Purnomo[2*]
[1,2]Department of Informatics, Faculty of Computer Science, Universitas Bhayangkara Jakarta Raya
tri.dharma.putra@dsn.ubharajaya.ac.id, rakhmat.purnomo@dsn.ubharajaya.ac.id*

Submitted: Jan 5, 2024 | **Accepted**: Feb 1, 2024 | **Publish**: Jul 8, 2024

Abstract: A model is a representation of an idea, thing, or object in a simplified form. The model contains information about a system that is created with the aim of studying the actual system. Simulation is an imitation of the system using a computer model. In this journal, simulation is done by using OS-SIM, an operating system simulator. Process scheduling is an important part of operating systems. Several scheduling algorithms exist in the field. Shortest job first, round robin, first come first serve, priority, and all of their variants. In this journal, discussion about the pre-emptive priority scheduling algorithm is presented thoroughly. A pre-emptive priority scheduling algorithm is an algorithm based on priority. The higher the number of the priority, the higher the priority. Five processes are available and given. Each with burst time, priority, and different arrival times. Simulation and modeling with OS-SIM are discussed to understand this algorithm more easily. Some statistics numbers in the system are calculated automatically by the OS-SIM. Some screenshot pictures of the simulator are given to describe the model. It is concluded that for these processes the average turnaround time is 42/5 = 8.4 ms and for average waiting time is 28/5=5.6 ms and the total burst time is 14 ms.

Keywords: Modelling, OS-SIM, Pre-emptive Priority Scheduling, Simulation, Turnaround Time; Waiting Time

INTRODUCTION

In today computer systems, operating systems plays an important role. Android, Windows and several other operating systems are available in the field. The main function of operating systems is acted as an interface between hardware below the operating systems and the application above the operating systems (Alexmazinho, 2022) (Purnomo & Putra, 2022a). Process scheduling acts to allocate resources between processes. Scheduling has a scheduler in operating systems and acts to manage the process in a proper way (Chandra Shekar N, 2017) (Omar et al., 2021) (Kunal Chandiramani, Rishabh Verma, 2019) (Putra & Purnomo, 2021) (Sakshi et al., 2022) (Putra & Purnomo, 2023). There are several scheduling algorithms exist in operating systems. In multiprocessing and multitasking systems nowadays, scheduling functions as the prime concept. Some algorithm can be implemented, however some are just theoretical. The main focus of scheduling algorithm is to make the performance system optimum that match the criteria set by the designer (Tri Dharma Putra, 2021) (Putra & Purnomo, 2022a) (Ledina Hoxha Karteri, 2015).

Computer simulator and modelling act to model and simulate the concept inside operating systems. Characteristic of variables inside the system can be changed and simulation can be done again and again. So, it makes us easier to understand the systems (Putra & Purnomo, 2022b) (Purnomo & Putra, 2022b) (Putra, 2022). Changing the variables in operating systems, so that prediction can be made to systems. Scheduling will make the performance of the systems improved (Putra & Purnomo, 2022a). By using OS-SIM simulator, we can simulate almost all aspects of operating systems, ranging from disk, memory, process, and file. Implementation of the systems' characteristic can be done easily, and changing variables can be done repeatedly with ease. Thus it makes understanding of the systems better.

In this journal, simulation and modelling of pre-emptive priority scheduling will be presented. There are pre-emptive and non pre-emptive priority scheduling algorithms. In this journal the focus will be analysed in pre-emptive priority scheduling algorithm only. Examples and screen shots of the feedback from systems will be given.

METHOD

A model is a representation of an idea, thing or object in a simplified form. The model contains information about a system that is created with the aim of studying the actual system . Simulation is an imitation of a dynamic system using a computer model. The function of it is to improve and evaluate system performance. Simulation is

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, Juli 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13352 p-ISSN : 2541-044X

a representation of a real world process or system which involves generating processes and observing the process to draw conclusions about the system being represented (Wikipedia, 2023).

Several computer system simulators are available in the field of operating systems. One of it is OS-SIM. In this journal, we use OS-SIM as the simulator. OS-SIM can simulate several behaviour of operating system's design. It consists of four domains which are process, Memory, File System, and Disk (Purnomo & Putra, 2022b). In this journal analysis, we use process simulator. Some calculation of average waiting time and average turnaround time can be done automatically by the system. Please take a look at Fig 1 below

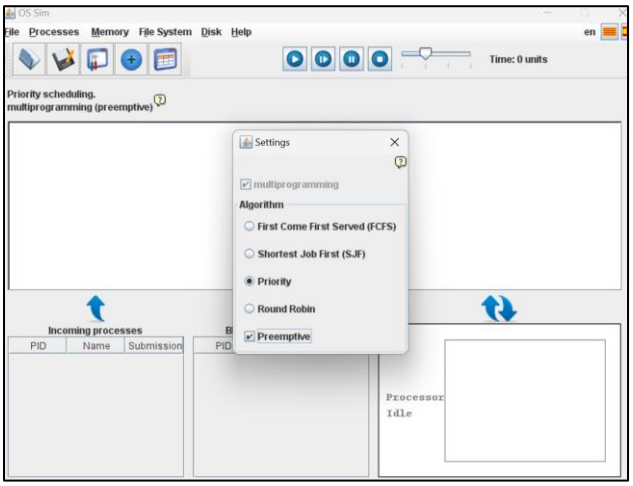

Fig 1. OS-SIM Configuration of Process

OS-SIM can simulate several scheduling algorithms. These are major algorithms in operating systems. First Come First Serve. Round Robin, Priority, Shortest Job First algorithms. All of them can be pre-emptived or non pre-emptived. Our main focus in this journal is priority pre-emptive algorithm with this OS-SIM. OS-SIM is proposed to support two main aspects of computer system. Namely, memory management system and process management system. Each process contains PCB (Process Control Block) which contain information about the state of the processes.

RESULT AND DISCUSSION

Please take a look at table 1, below. There are five processes. Each with arrival time (submission), priority, and burst time. The higher the priority number, the higher the actual priority.

Table 1. Process with arrival time, priority, and burst time

Process	Arrival Time	Priority	Burst Time
C1	0	1	3
C2	1	3	2
C3	1	4	4
C4	3	2	2
C5	4	5	3

So if the priority is 5. It has the highest priority in this scheduling. For arrival time, also called submission, starts from 0 for C1 and starts at 1 for processes C2 and C3. And for C4 starts at 3. For C5 starts at 4. So the idea of submission time is that each process starts with different arrival time or submission. Then because this is the pre-emptive algorithm, then in the middle of running process, it can pre-empt the running process if the newly arrival process has the higher priority.

The total burst times is 14 ms. With C3 has the highest burst time, which is 4 ms. For C5, its burst time is 3 ms and for C4, the burst time is 2 ms. For C2 the burst time is 2 ms and for C1 the burst time is 3 ms.

C1 is executed first as per the Fig 2 below. The arrival time of C1 is 0. So, it is executed the first time in this simulation. C1 has burst time 3 ms. However, at 1 ms, C1 is preemptive, since its priority is only 1. The lowest priority. Please take a look at Fig 2, below. After C1 is executed for 1 ms, then it will be pre-empted by C3, since C3 has higher priority. So it is concluded that C1 is executed only for 1 ms. Then C1 is backed to the ready queue, waiting for its time to be executed.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, Juli 2024
DOI : https://doi.org/10.33395/sinkron.v8i3.13352
e-ISSN : 2541-2019
p-ISSN : 2541-044X

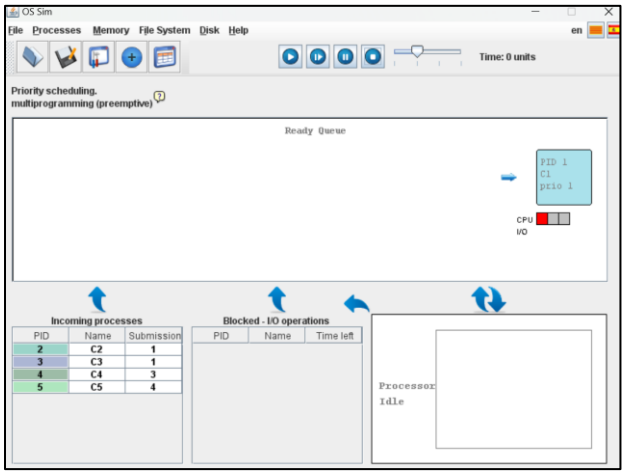

Fig 2. C1 is executed for the first time. Since its arrival time is 0.

Since at 1 ms, C3 has the highest priority which is 4 compared to C2 which have priority 3, which is lower than the priority of C3. Then C3 arrived at the ready queue. C3 is executed until 4 ms. At 1 ms, there are two arrival times which are C2 and C3. But since C3 has higher priority which is 4, then C3 is executed first. And for C2 with priority 3. It is arrived in the ready queue, waiting to be executed. As per Fig 3, below.

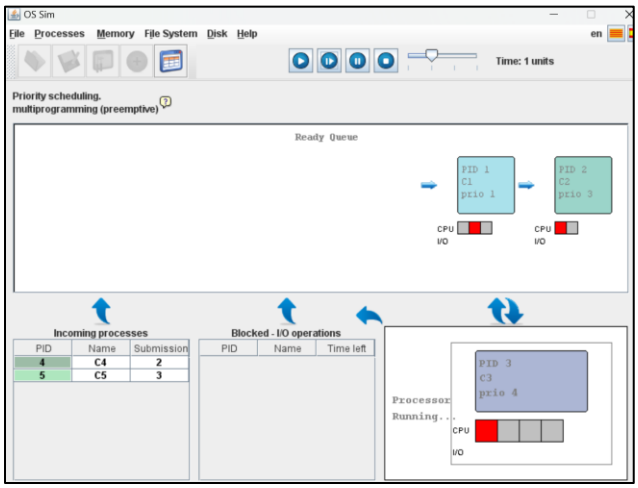

Fig 3. the modelling at 1 ms.

Please take a look at Fig 4, below. C3 which has burst time 4 ms, is executed until 4 ms. C3 is executed until 4 ms. Since at 4 ms, C5 arrives. C5 has priority 5, which is higher than then priority of C3. C3 is left with 1 burst time. Then C3 is assigned to the ready queue. At 3 ms, C4 arrived. But its priority is only 2., which is lower than the priority of C3. Again in this case, the C3 is executed continuously until 4 ms. So it is concluded that C3 is left with 1 ms to be executed. Not all of its burst time of C3. C3 then at 4 ms is assigned to the ready queue.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, Juli 2024
DOI : https://doi.org/10.33395/sinkron.v8i3.13352
e-ISSN : 2541-2019
p-ISSN : 2541-044X

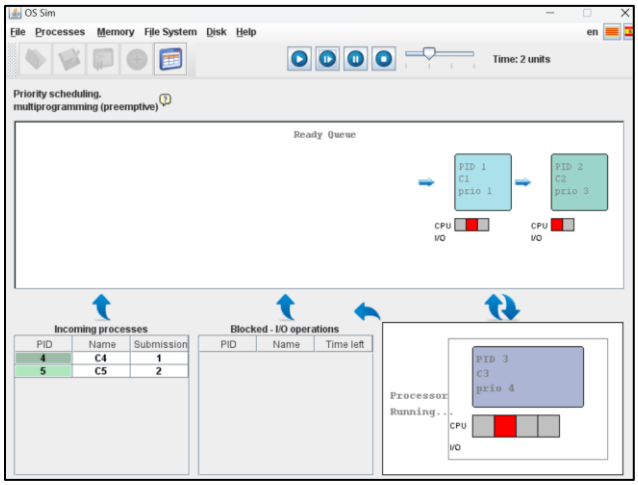

Fig 4. C3 is executed.

At Fig 4 above, C3 is still executed. At 3 ms. C4 is arrives in the ready queue. However, since C4 has only priority 2, it has to wait, because its priority is only 2, meanwhile C3 has higher priority which is 4. Since C4 arrives at the ready queue with priority 4, then it will assigned to the ready queue, since its priority is lower than C3. Please take a look at Fig 5, below.

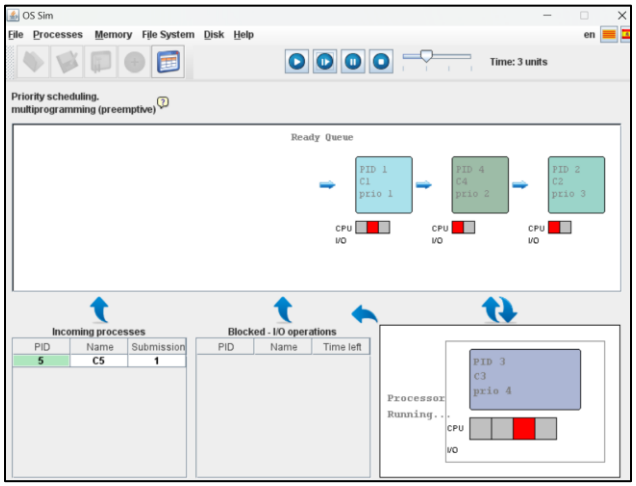

Fig 5. C3 is executed until 4 ms. It is left with 1 ms burst time.

Please take a look at Fig 6 below. At 4 ms. C5 is executed. Its priority is compared with C3, since C3 has only priority 4, but C5 has priority 5, which is higher. Then C3 is left with 1 ms and put in the ready queue to wait. C5, then is executed until finish. It started at 4 ms, then finished at 7 ms, since the burst time of C5 is 3 ms. Therefore, for C5 it is not assigned to the ready queue, since its burst times are already executed until finish.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, Juli 2024
DOI : https://doi.org/10.33395/sinkron.v8i3.13352
e-ISSN : 2541-2019
p-ISSN : 2541-044X

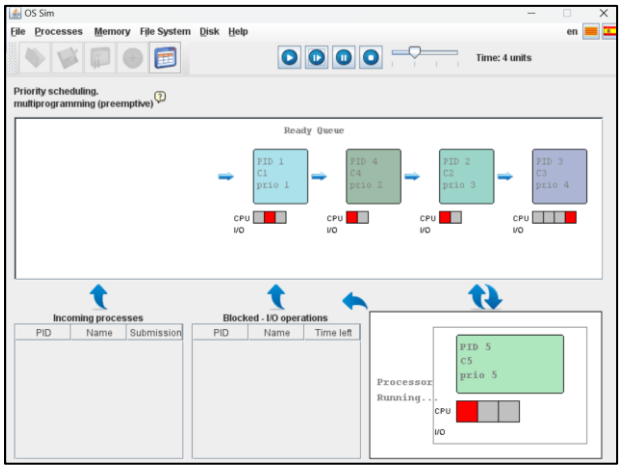

Fig 6. C5 is executed at 4 ms.

At 3 ms, all of the processes is in the ready queue. Since the arrival times of C2 is 1 ms and the priority of C2 is only 3 and the burst time of C2 is only 2 ms. The arrival times of C4 is 3 ms, and the arrival time of C5 is 4 ms. At 3 ms, C4 is in the ready queue. C5 is executed until finish. The burst time of C5 is 3 ms. Then C5 is executed all of its burst time. Please take a look at Fig 7, below.

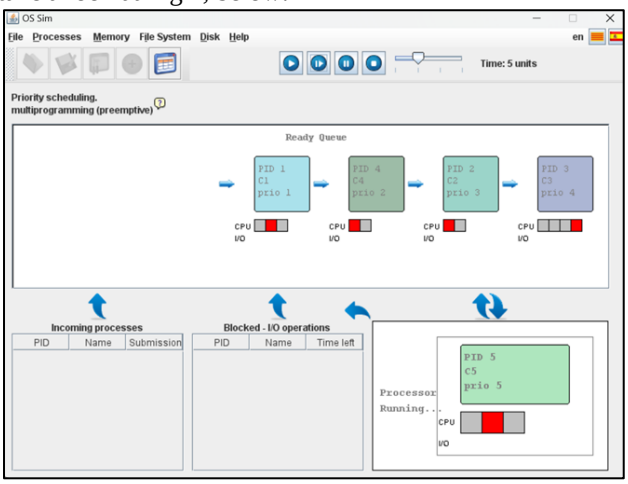

Fig 7. C5 is executed in the running process.

Please take a look at Fig 8, below. C5 is executed until finish from 4 ms until 7 ms.. It has only burst time 3. It is executed until 7 ms. In the Fig 8, can be seen that C3 is in the ready queue. C3 in ready queue is ready to be executed until finish. Before C3 is executed, the priority of C2 and C4 and C3 are compared. It is concluded that C3 is has the highest priority than C2 and C4. So C3 still has the highest priority. It has only 1 ms left. Afterwards, C3 is executed.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, Juli 2024
DOI : https://doi.org/10.33395/sinkron.v8i3.13352
e-ISSN : 2541-2019
p-ISSN : 2541-044X

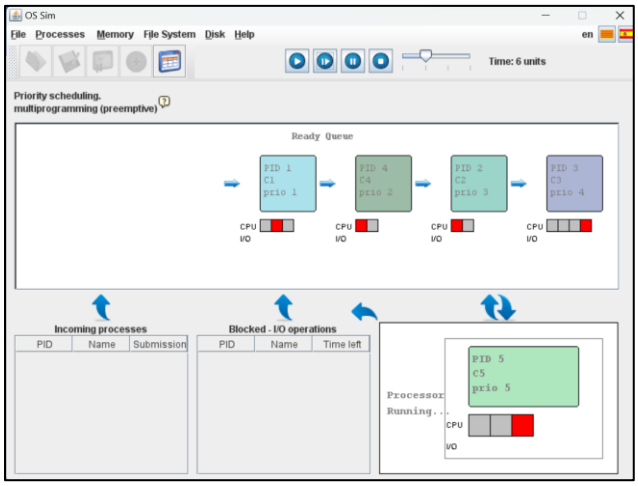

Fig 8. The simulation is finishing C5 until finish.

Please take a look at Fig 9, below. C5 is executed until finish at 7 ms from 4 ms. Afterwards, since C5 is finish, then, directly since C3 is in the front of ready, than C3 is executed until finish at 8 ms. In this case, C3 is only executed for 1 ms, C3 is then finishes. C3 is finished because it has the second higher priority. The highest priority is C5 which is 5, compared to C3 which is 4. So that after C5 is executed and finish, than C3 is directly executed, since it is already waiting in the ready queue.

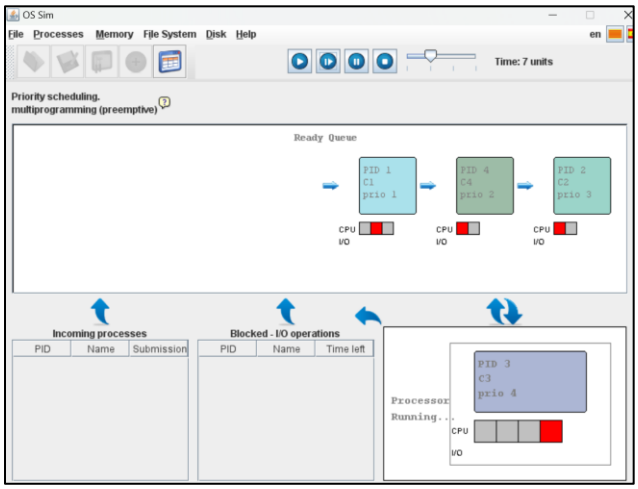

Fig 9. C3 is executed the 1 ms left.

Please take a look at Fig 10, below. After C3 is finished, then comparison between priorities should be done again between processes, below. The priority of C2 is compared with C4 and C1. Since C2 has priority 3 and C4 has priority 2, and C1 has only priority 1, then C2 is executed first. C4 and C1 are assigned to the ready queue, waiting their time to be executed. Then, C2 is executed until finish at 10 ms, with 2 burst times.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, Juli 2024
DOI : https://doi.org/10.33395/sinkron.v8i3.13352
e-ISSN : 2541-2019
p-ISSN : 2541-044X

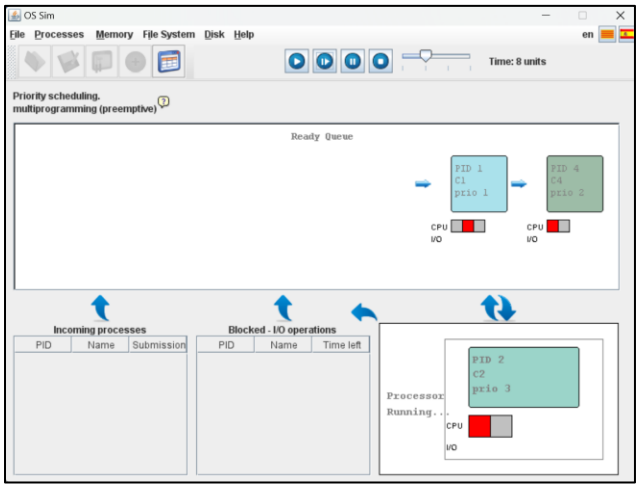

Fig 10. C2 is executed in the running process.

Please take a look at Fig 11. Below. C2 is executed until finish, from 9 ms until 10 ms. Since its burst time is only 2 ms. Now in the ready queue, there are only two processes left, which are C1 and C4.

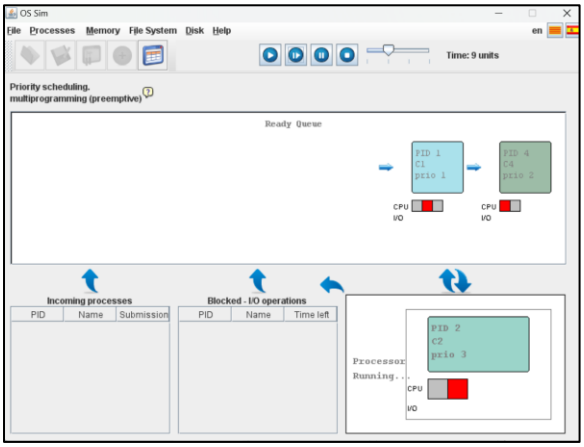

Fig 11. Finishing C2 until all burst time is executed.

Please take a look at Fig 12, Below. The priority of C4 and C1 is compared. Comparison of priorities of C1 and C4 are done. C1 has priority 1. C4 has priority 2. And the burst time of C4 is 2 ms. Then since the priority of C4 is higher than the C1, then C4 is executed for 2 ms as its burst time. Apparently, C4 has higher priority which is 2 compared with 1 in C1. Then C4 is executed until finish. C4 is executed for 2 ms. It is at 11 ms and 12 ms.

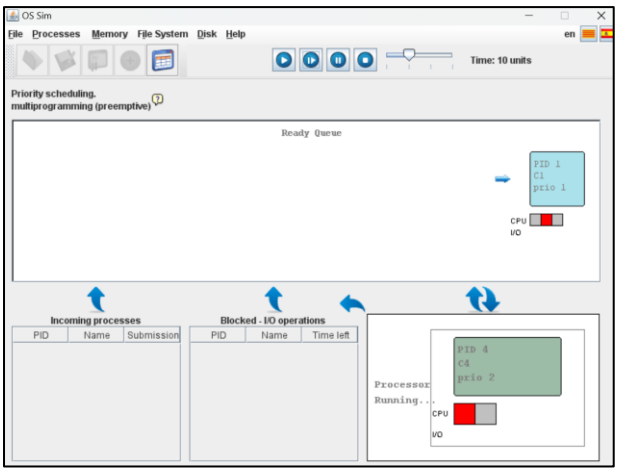

Fig 12. Only left two process C4 and C1. C4 in the running process.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, Juli 2024
DOI : https://doi.org/10.33395/sinkron.v8i3.13352
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Please take a look at Fig 13, below. Only left 1 ms burst time of C4 in the running process. In the ready queue, C1 left with 2 ms. Then, since the priority of C4 still higher than C1, then C4 is continue to be executed until it is finish at 12 ms. However C1 is still in the ready queue.

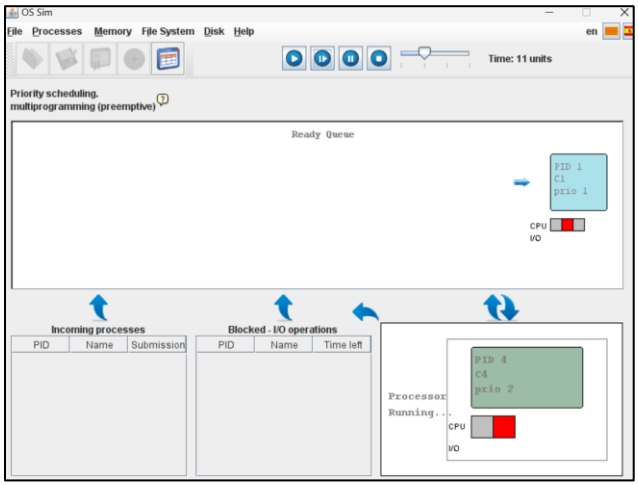

Fig 13. C4 in the last 1 ms process.

Afterwards, please take a look at Fig 14, below. Here there is no comparison at all. Since all other processes are already executed. Then C1 is left in the ready queue with 2 ms. The last two bursts of C1 is executed.

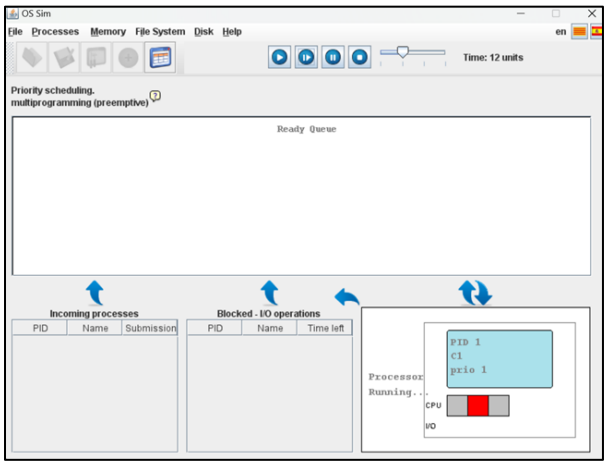

Fig 14. C1 left with 2 ms burst time.

Finally, the Fig 15, below. This last Fig finishes the simulation and modelling. Then, directly C1 is executed for 13 ms and 14 ms. C1 before was executed for the first time, but afterwards has to wait in the ready queue since its priority is only 1. The last of C1. It is concluded C1 started as the first processes but has to wait until all other processed finish to be executed, then, lastly C1 is executed for 2 ms at 13 ms and 14 ms.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, Juli 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13352 p-ISSN : 2541-044X

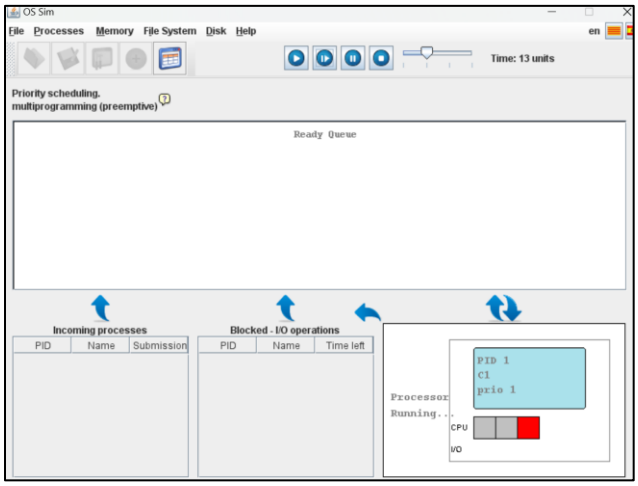

Fig 15. The last process

For the gantt chart after the analysis above, can be seen in the Fig 16, below. C1 is executed first, afterwards C3, which has the higher priority. Then C4 until 4 ms. After that C5 pre-empt the process of C3 until finish, which is C5, until 7 ms. Then C3 which is left only 1 ms, is executed. After that in the system is left C4, C1, and C2. Since C2 has the highest priority, C2 is executed until finish for 2 ms. Then afterwards, C4 for 2 ms and C1 the last for 2 ms also.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
C1	C3	C3	C3	C5	C5	C5	C3	C2	C2	C4	C4	C1	C1

Fig 16. Gantt Chart Analysis of the processes

The Fig 16, below shows the analysis of the system. The priorities are ranging from 1 until 5. With 5 the highest priority. The submission or arrival time is at 0, 1, 3, and 4 ms. The burst times are between 2 and 4 ms. Response time is 7 for C2 and C4 and others are 0. The highest waiting time is C1, which is 11 ms. C5 does not has waiting time, since it pre-empts the systems and directly executed. For C3 waiting time is 3 ms. For C2 waiting time is 7 ms, and for C4 waiting time is also 7 ms. For turnaround time C5 is 3 ms. C3 is 7 ms, C2 is 9 ms, C4 is 9 ms and C1 is 14 ms. It is concluded that for these processes the average turnaround time is 42/5 = 8.4 ms and for average waiting time is 28/5=5.6 ms.

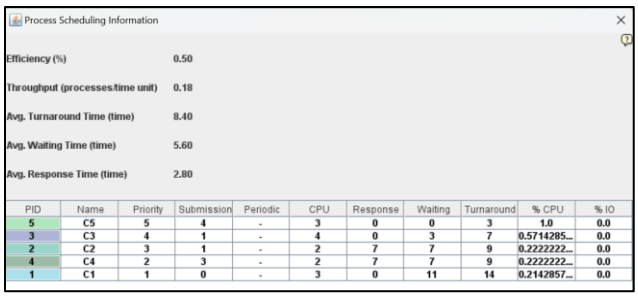

Fig 16. Analysis of Processes

CONCLUSION

Pre-emptive priority scheduling algorithm is based on priority. The higher the priority number, the higher the actual priority. Simulation and modelling this algorithm has been done with OS-SIM. The purpose of this simulation is to analyse and understand this algorithm more clearly. Five processes are modelled and simulated in this analysis. By giving example of five processes with burst times, arrival times, and priorities. It is concluded that the total burst time is 14 ms. The average turnaround time is 8.4 ms. The average waiting time is 5.6 ms. For future works may be the simulation can be done with MATLAB or other simulator, other than OS-SIM.

REFERENCES

Alexmazinho. (2022). *OS-SIM*. OS-Simulator.
Chandra Shekar N, K. V. (2017). Analysis of Priority Scheduling Algorithm on the Basis of FCSF & SJF for

*name of corresponding author

Similar Priority Jobs. *International Journal of Engineering Research in Computer Science and Engineering*, *4*(3), 73–76.

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.

Ledina Hoxha Karteri, A. S. (2015). Preemptive and Non- Preemptive Priority Scheduling. *International Journal of Computer Science & Management Studies*, *19*(01), 1–5.

Omar, H. K., Jihad, K. H., & Hussein, S. F. (2021). Comparative analysis of the essential cpu scheduling algorithms. *Bulletin of Electrical Engineering and Informatics*, *10*(5), 2742–2750. https://doi.org/10.11591/eei.v10i5.2812

Purnomo, R., & Putra, T. D. (2022a). Comparison Between Simple Round Robin and Improved Round Robin Algorithms. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, *9*(3), 2205–2221. https://doi.org/10.35957/jatisi.v9i3.2547

Purnomo, R., & Putra, T. D. (2022b). Simulation of Preemptive Shortest Job First Algorithm. *IJARCCE*, *11*(5), 1–11. https://doi.org/10.17148/IJARCCE.2022.11501

Putra, T. D. (2022). Analysis of Priority Preemptive Scheduling Algorithm: Case Study. *Ijarcce*, *11*(1), 27–30. https://doi.org/10.17148/ijarcce.2022.11105

Putra, T. D., & Purnomo, R. (2021). Analisis Algoritma Round Robin pada Penjadwalan CPU. *Jurnal Ilmiah Teknologi Informasi Asia*, *15*(2), 85. https://doi.org/10.32815/jitika.v15i2.481

Putra, T. D., & Purnomo, R. (2022a). Case Study : Improved Round Robin Algorithm. *Sinkron : Jurnal Dan Penelitian Teknik Informatika*, *7*(3), 950–956.

Putra, T. D., & Purnomo, R. (2022b). Simulation of Priority Round Robin Scheduling Algorithm. *Sinkron*, *7*(4), 2170–2181. https://doi.org/10.33395/sinkron.v7i4.11665

Putra, T. D., & Purnomo, R. (2023). Average Max Round Robin Algorithm: A Case Study. *Sinkron*, *8*(3), 1230–1237. https://doi.org/10.33395/sinkron.v8i3.12051

Sakshi, Sharma, C., Sharma, S., Kautish, S., A. M. Alsallami, S., Khalil, E. M., & Wagdy Mohamed, A. (2022). A new median-average round Robin scheduling algorithm: An optimal approach for reducing turnaround and waiting time. *Alexandria Engineering Journal*, *61*(12), 10527–10538. https://doi.org/10.1016/j.aej.2022.04.006

Tri Dharma Putra, A. F. (2021). Comparison Between Simple Round Robin and Intelligent Round Robin Algorithms in CPU Scheduling. *International Journal of Advanced Research in Computer and Communication Engineering*, *10*(4), 86–90.

Wikipedia. (2023). *Wikipedia*. Wikipedia.

*name of corresponding author