AUTOMATED FOOD ORDERING SYSTEM BY PROCESS SCHEDULING SJF, SRTF, PRIORITY

SEE WEN XIANG
School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia.
wxsee@graduate.utm.my

AIMAN ARRIADY BIN AZMAN
School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia.
arriady2018@graduate.utm.my

ABU SAYED KHUDRI ROMAN
School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia.
sayedkhudri@graduate.utm.my

Abstract

In this globalization era, our lives are getting busier and more hectic each day, we tend to organise our daily asks in such inefficient manner. This led us to often become tired and non-energetic for the following day. An operating system operates in a very similar way as us, human. They require process scheduling to make sure that the system efficient, fast and fair. For example, when the CPU is in the idle state, the operating system will select one of the processes available in the ready queue for execution. The selection of process is carried by the CPU scheduler.

The main objective of this project is to develop a program for shortest job first scheduling Shortest Remaining Time First and Priority scheduling which helps to improve the Food Ordering process. There are many algorithms available for CPU scheduling. But we cannot implement in a real-time operating system because of high context switch rates, large waiting time, large response time, large turnaround time, and less throughput. The proposed program improves all the drawbacks of simple shortest job first scheduling Shortest Remaining Time First and Priority scheduling architecture.

The project has also given a comparative analysis of the proposed shortest job first scheduling Shortest Remaining Time First and Priority scheduling algorithm. Therefore, we strongly feel that the proposed program solves all the problems encountered while doing manually calculate the waiting time, turnaround time, or burst Time (cooking time).

1 Introduction

In this modern era, people are more and more particular about the concept of eating. Hence, restaurants need to pursue orders for delicious food in a shorter time. However, most of the restaurant faces the problem of scheduling the food order. In order to solve the problem of restaurant, our team write this proposal to propose a program system which can help the restaurant to manage food order.

Operating system is the most important software that runs on the computer. It acts as a bridge between the software and hardware of the computer (Li & Ierapetritou, 2008). Furthermore, it manages the computer’s resources include the central processing unit (CPU), memory, storage, I/O devices and network connections. In addition to that, operating system also includes process manager that handles task such as scheduling. A process scheduler schedules different processes to be assigned to the CPU based on a particular scheduling algorithm (Çalhan, 2016).

Shortest Job First (SJF) is a non-preemptive scheduling algorithm that is commonly found in a batch system (javatpoint, 2011) (Akhtar, et al., 2015). SJF scheduling primarily works based on the CPU Burst Time (Cooking Time) or known as process duration. In SJF scheduling, the process with the lowest Burst Time (Cooking Time) is given priority to be executed first among the list of available processes in the ready queue. With this, the algorithm can achieve the maximum throughput as well as minimum average waiting time and turnaround time (Rungta, 2021). On the other hand, SJF algorithm may also cause starvation problems where the lower priority process will never be executed as the high priority process keeps arriving in the queue. In addition to that, the exact Burst Time (Cooking Time) for a process cannot really be known in advance, hence it is very hard to implement this algorithm.

Shortest Time Remaining First (SRTF) is a method of scheduling that applies the pre-emptive version of SJF algorithm (javatpoint, 2011). In SRTF, a process can be stopped or paused from execution after a certain amount of time. The reason is that the short-term scheduler has scheduled the arrival of every process based on the least remaining Burst Time (Cooking Time). For example, the execution of process A will be stopped when a new process arrives with a shorter Burst Time (Cooking Time). This will allocate space in the CPU for the shorter process to take place. Once all the processes are available in the ready queue, there is no pre-emption to be done and the algorithm will work as per normal SJF scheduling algorithm. A Process Control Block (PCB) is also used to save the context of the process when the process is removed from the CPU execution. With that said, the PCB is accessed again to continue the previous process when the pre-empted process ends.

Priority scheduling is a technique of scheduling processes based on the priority (techopedia, 2021). The process with high importance which is high priority will be executed first. In this scheduling algorithm, all process involve with the priority assignment and processes with higher priority will be carried out first, if the tasks process with equal importance priority, then first-come first served (FCFS) or round robin will be implemented.

2 Problem Statement

In this project, we are going to propose a system called the Automated Food Ordering System. We will be using the **C++ language** for our program. The system is implemented using scheduling algorithms for preparing the customer’s food. The algorithms are the Shortest Job First (SJF), Shortest Remaining Time First (SRTF), and lastly priority scheduling. Furthermore, the type of customers are dine-in customer, take-away customer, and home delivery customer. In the real world, the most important factors that ensure the success of the restaurant is the customer’s satisfaction. Therefore, the number of completed orders and the customer’s serving time has to be tracked. Once seated, the customer should be served no more than 20 minutes from ordering.

For the first scenario of problems, which is the Shortest Job First (SJF) algorithm. It will be applied to the home delivery customers. Orders will be prepared based on the delivery address of the customer. For example, the customer that has the shortest distance to the delivery address will have their order prepared first. On the other hand, customers with a farther delivery address will be put in the waiting list. Since the SJF is not pre-emptive, the current orders will have to be finished first before proceeding with the other orders.

Moving to the second scenario of problems, we will be implementing the Shortest Remaining Time First (SRTF) algorithm in increasing the efficiency of food preparation. The scenario begins when there are orders arriving at one time from two different customers. For example, when the restaurant has finished preparing one dish from customer A, the restaurant will then stop preparing the next dishes in the order list and start preparing a dish from customer B that takes less time to complete. This is directly related with the principle of SRTF which means that current processes were pre-empted in order for the next process that has less Burst Time (Cooking Time) to take place.

Finally, the third algorithm that we are going to implement in the system is the priority scheduling. This method is fairly simple because food preparation will be based on the type of customers. For example, order preparation for multiple customers with the same orders will be based on the customers' type. A take-away customer should get the priority the highest, followed by dine in and lastly the home delivery customer.

3 Research Methodology

A better extensive exploration research improves the nature of any research project. It assists specialists with overseeing research steps organized appropriately. Each progression of the interaction has its own expectations and aides the specialists to certainly move to the following stages of the research.

This research consisted of four phases:

Phase 1: Research & Literature Review

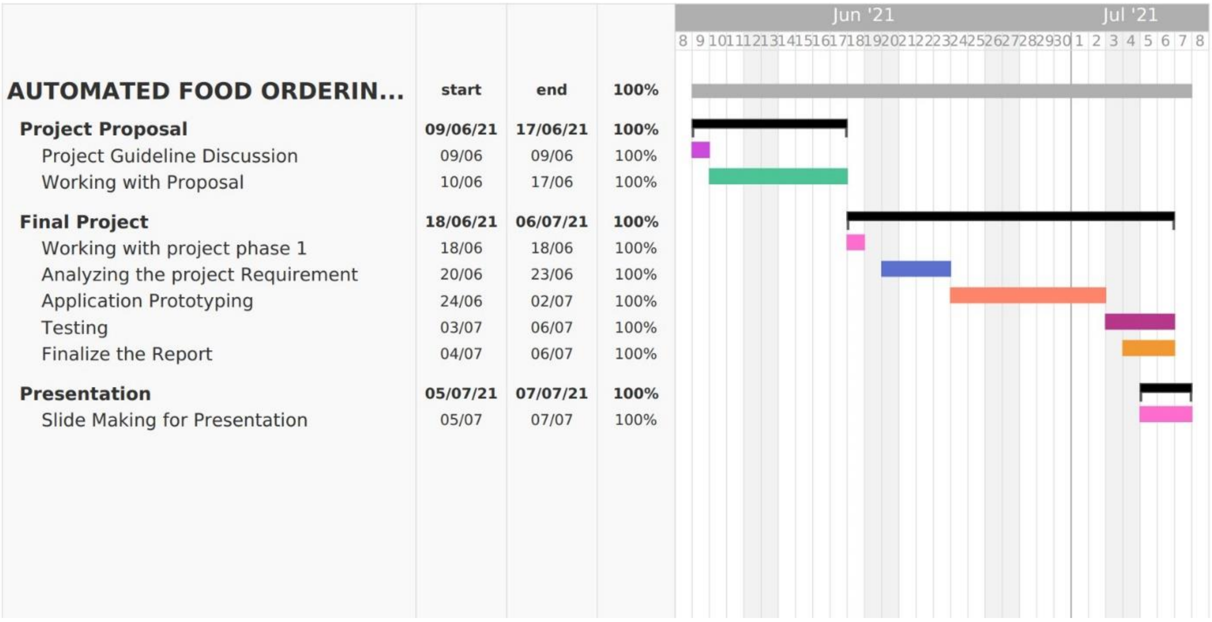
This research is related to one of the most important issues in principles of the operating system design. In This section, the more frequently used terms in this literature are defined. Some of these terms, are used as the efficiency criteria by some researchers.

- Process: The program which is loaded into memory to be executed.
- Scheduling: Making decision about allocating policies of the recourses to the processes
- Waiting time: A time period which a process waits to allocate a resource in the system.
- Burst Time (Cooking Time): A time period which a process needs for completion.
- Turnaround Time: The time interval between dispatching and completing of a job. Response Rime: The time taken until the first response of a typical request is emerged.
- Throughput: The number of completed(finished) jobs in a specific time period.

Phase 2: Requirement Analysis

The most crucial aspect of every project is the requirement analysis. It will help in identifying what stakeholders' actual needs are. Simultaneously, it enables stakeholders to be communicated with in a language they understand (charts, models, flowcharts, etc.) rather than complicated text.

In our project we use Gantt chart which is a graphical representation of a schedule that helps to coordinate, plan and track specific tasks in a project. It represents the total time span of the object, broken down into increments.



AUTOMATED FOOD ORDERIN...	start	end	100%
Project Proposal	**09/06/21**	**17/06/21**	**100%**
Project Guideline Discussion	09/06	09/06	100%
Working with Proposal	10/06	17/06	100%
Final Project	**18/06/21**	**06/07/21**	**100%**
Working with project phase 1	18/06	18/06	100%
Analyzing the project Requirement	20/06	23/06	100%
Application Prototyping	24/06	02/07	100%
Testing	03/07	06/07	100%
Finalize the Report	04/07	06/07	100%
Presentation	**05/07/21**	**07/07/21**	**100%**
Slide Making for Presentation	05/07	07/07	100%

Figure 1 Gantt Chart of Proposed Automated Food Ordering System

Phase 3: Application Prototyping

The C++ programming language is used to create an application for this problem. Our work has been separated into three primary options and one additional option that is utilized to leave the program in this application. The user of this program will be presented with the primary menu interface, which will display these three options.

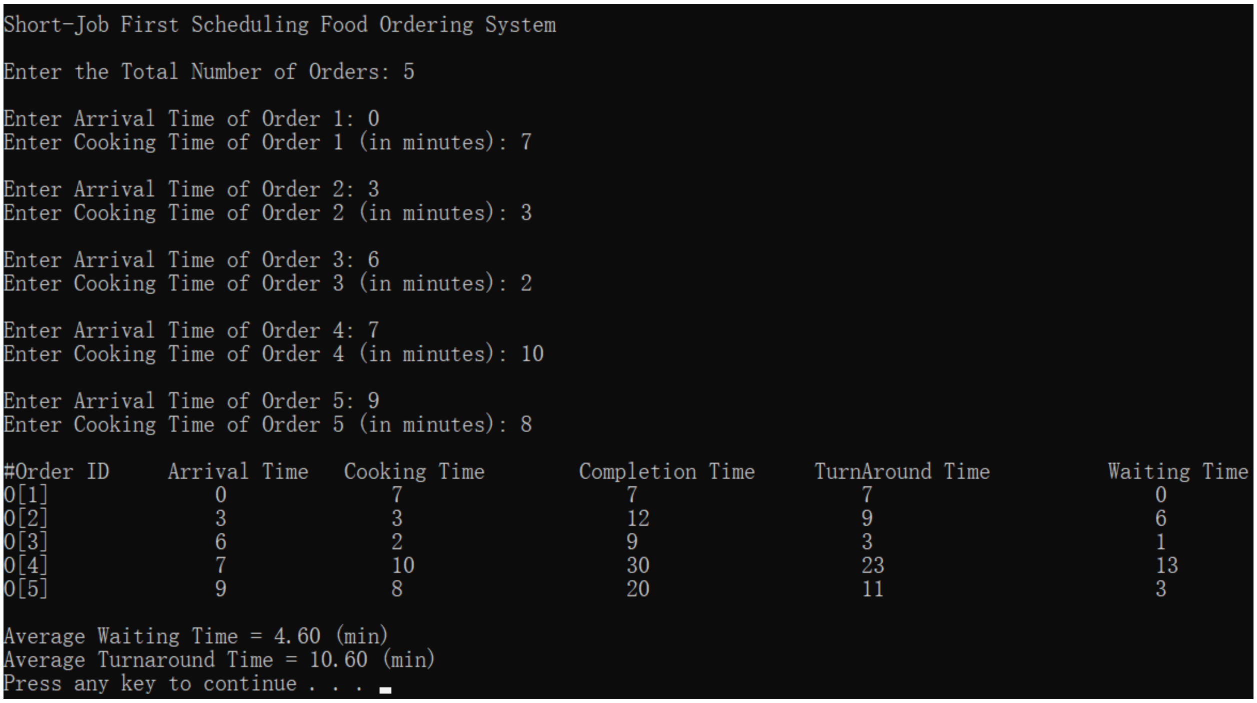
[1] Short-Job First Scheduling
[2] Short-Remaining Time First
[3] Priority Scheduling
[4] Exit

If the user chooses option 1, the system will begin calculating Short-Job First Scheduling by asking for a few details from the user, such as the total number of orders, arrival time, and order Burst Time (Cooking Time). Using the Short-Job First Scheduling method, the program will determine waiting time, turnaround time, and average time. According to their algorithm, the application works in the same way as the other three options. To quit this program, the user must hit 4 to pick the exit option.

Phase 4: Testing

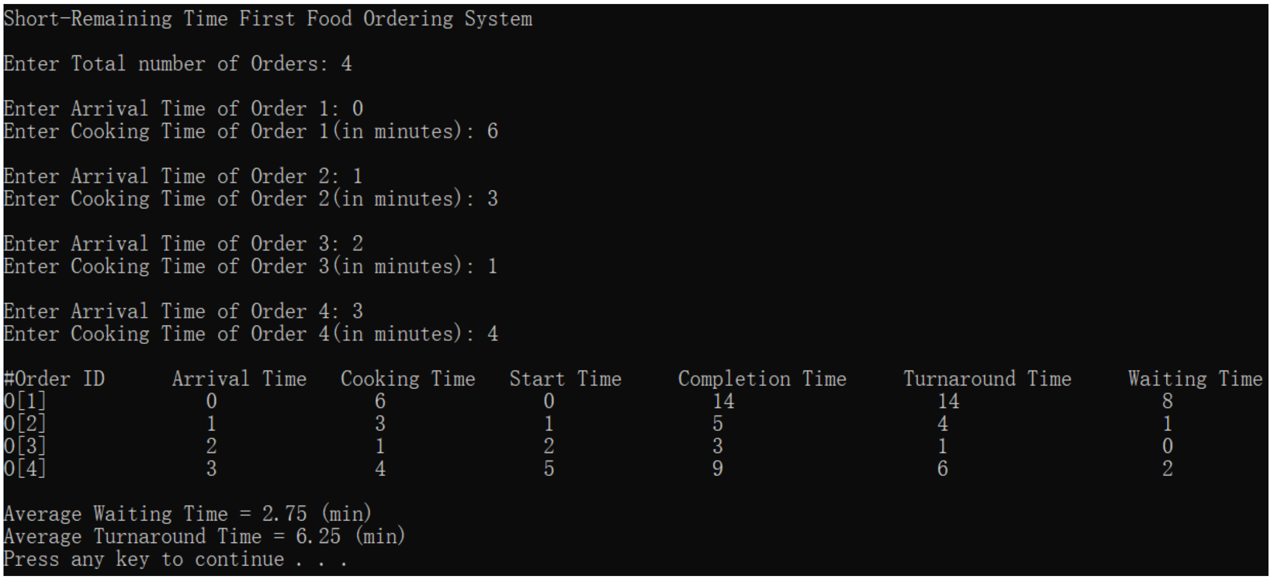
For the testing phase, we conducted the testing by monitoring the output from the screen. We made a lot of adjustments especially in terms of the execution flow and the tidiness of the code. The output screen is very organised with proper spacing between each element so that the user is able to understand the calculation clearly.

From the screenshots below, we can see that the structure for each scheduling algorithm output is the same. First, the user will be asked to enter the total number of orders. After that, the user will be able to enter arrival time and burst time (cooking time) based on the order number. Next, the table output will be shown by the order ID, arrival time, cooking time, completion time, turnaround time, and waiting time. Finally, the output will display the calculation result for the average waiting time and the average turnaround time in minutes based on value entered by the user in the beginning.



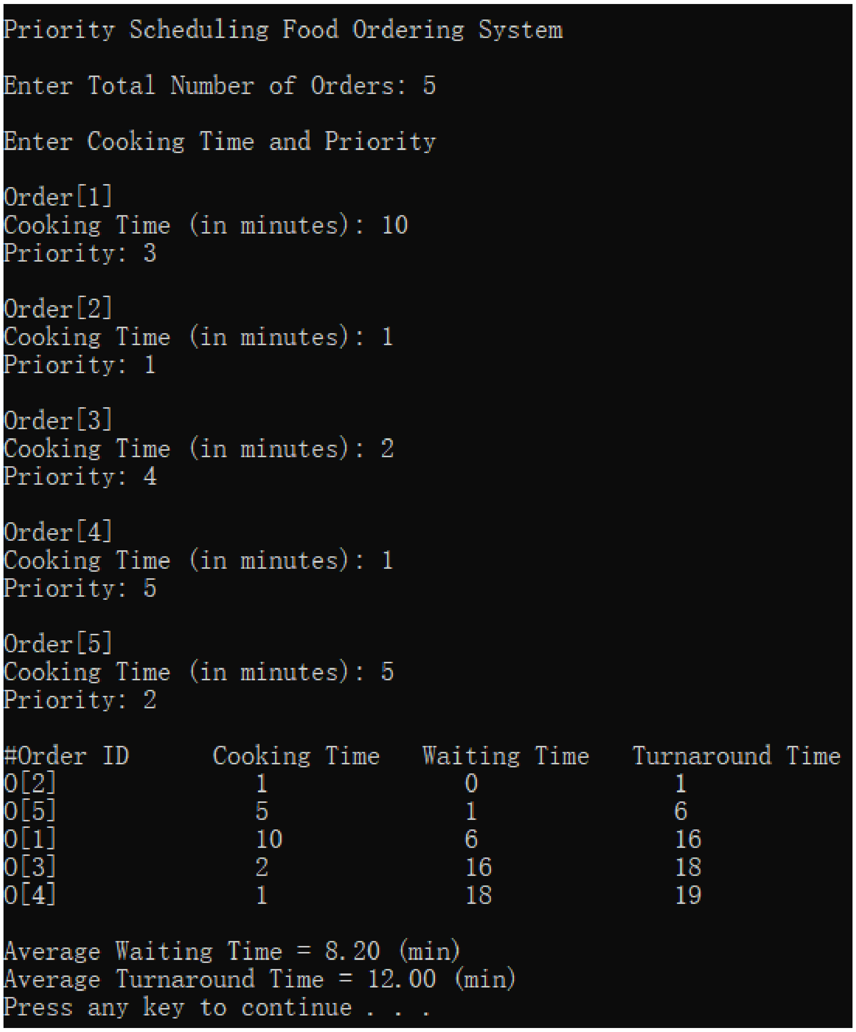
```
Short-Job First Scheduling Food Ordering System

Enter the Total Number of Orders: 5

Enter Arrival Time of Order 1: 0
Enter Cooking Time of Order 1 (in minutes): 7

Enter Arrival Time of Order 2: 3
Enter Cooking Time of Order 2 (in minutes): 3

Enter Arrival Time of Order 3: 6
Enter Cooking Time of Order 3 (in minutes): 2

Enter Arrival Time of Order 4: 7
Enter Cooking Time of Order 4 (in minutes): 10

Enter Arrival Time of Order 5: 9
Enter Cooking Time of Order 5 (in minutes): 8

#Order ID     Arrival Time   Cooking Time       Completion Time     TurnAround Time          Waiting Time
O[1]              0              7                  7                   7                        0
O[2]              3              3                  12                  9                        6
O[3]              6              2                  9                   3                        1
O[4]              7              10                 30                  23                       13
O[5]              9              8                  20                  11                       3

Average Waiting Time = 4.60 (min)
Average Turnaround Time = 10.60 (min)
Press any key to continue . . .
```

Figure 2 Screenshot of Short-Job First Scheduling Food Ordering System



```
Short-Remaining Time First Food Ordering System

Enter Total number of Orders: 4

Enter Arrival Time of Order 1: 0
Enter Cooking Time of Order 1(in minutes): 6

Enter Arrival Time of Order 2: 1
Enter Cooking Time of Order 2(in minutes): 3

Enter Arrival Time of Order 3: 2
Enter Cooking Time of Order 3(in minutes): 1

Enter Arrival Time of Order 4: 3
Enter Cooking Time of Order 4(in minutes): 4

#Order ID      Arrival Time   Cooking Time   Start Time     Completion Time     Turnaround Time     Waiting Time
O[1]              0              6              0              14                  14                  8
O[2]              1              3              1              5                   4                   1
O[3]              2              1              2              3                   1                   0
O[4]              3              4              5              9                   6                   2

Average Waiting Time = 2.75 (min)
Average Turnaround Time = 6.25 (min)
Press any key to continue . . .
```

Figure 3 Screenshot of Short-Remaining Time First Scheduling Food Ordering System



```
Priority Scheduling Food Ordering System

Enter Total Number of Orders: 5

Enter Cooking Time and Priority

Order[1]
Cooking Time (in minutes): 10
Priority: 3

Order[2]
Cooking Time (in minutes): 1
Priority: 1

Order[3]
Cooking Time (in minutes): 2
Priority: 4

Order[4]
Cooking Time (in minutes): 1
Priority: 5

Order[5]
Cooking Time (in minutes): 5
Priority: 2

#Order ID      Cooking Time   Waiting Time   Turnaround Time
O[2]              1              0              1
O[5]              5              1              6
O[1]              10             6              16
O[3]              2              16             18
O[4]              1              18             19

Average Waiting Time = 8.20 (min)
Average Turnaround Time = 12.00 (min)
Press any key to continue . . .
```

Figure 4 Screenshot of Priority Scheduling Food Ordering System

4 Design & Implementation

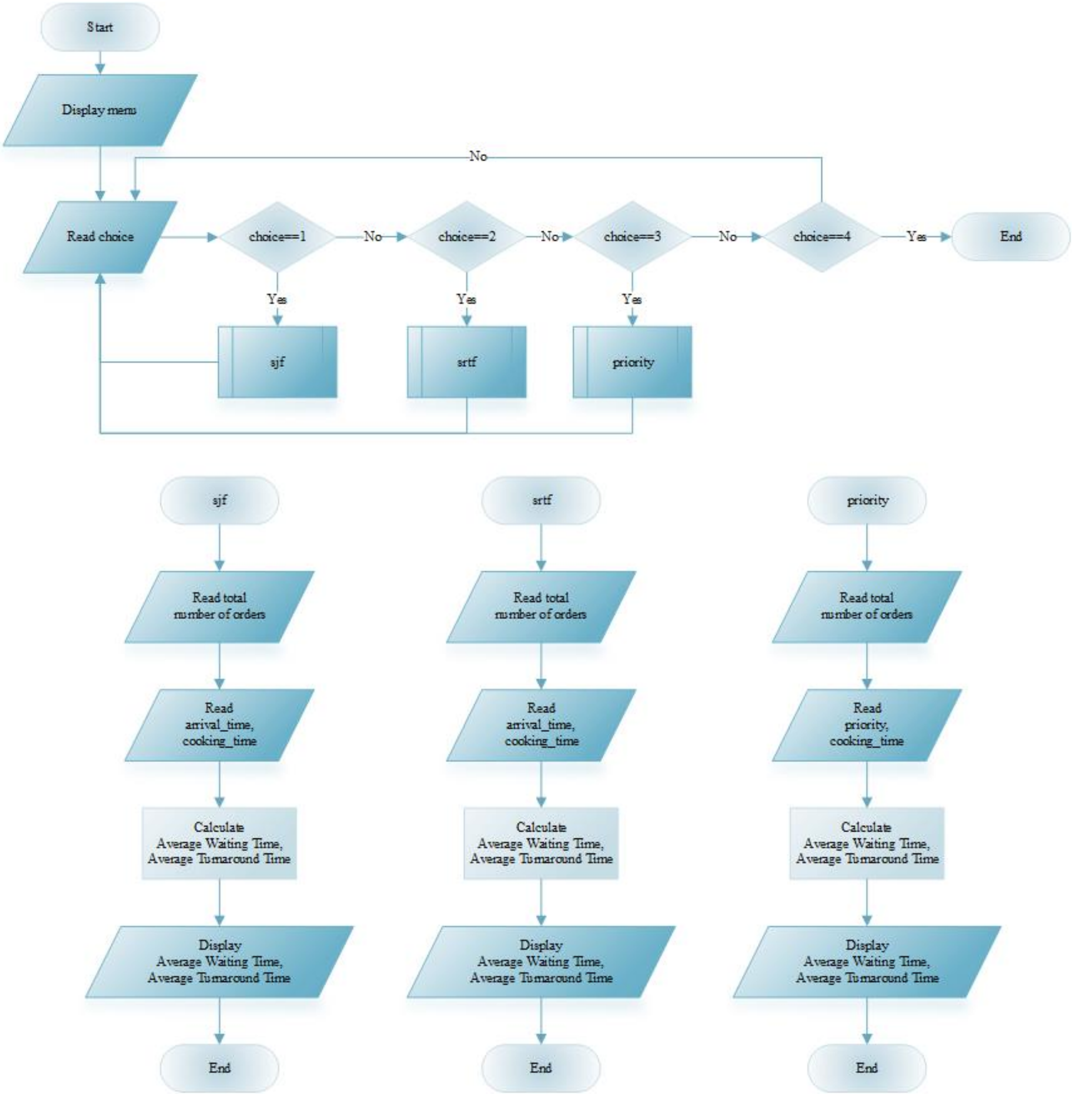

Start
Display menu
Read choice
choice==1
choice==2
choice==3
choice==4
No
Yes
End
sjf
srtf
priority
Read total number of orders
Read arrival_time, cooking_time
Read priority, cooking_time
Calculate Average Waiting Time, Average Turnaround Time
Display Average Waiting Time, Average Turnaround Time
End

Figure 5 Flowchart of Proposed Automated Food Ordering System

Process scheduling algorithm is depicted in a flowchart in Figure 4. It depicts that it will display a menu from which user must choose one option. Option 1 will take the user to the shortest job first scheduling, option 2 to the shortest remaining time, option 3 to Priority scheduling, and option 4 to quit the application if needed. In the shortest job, the user is first asked to enter the total number of orders, followed by the arrival time and cooking time. After receiving all of the data, the application calculates the average waiting time and displays all of the information. By tapping any key, the user can now leave the program. To utilize the Shortest Remaining Time First Algorithm, choose option 2 from the menu. It will then operate in the same way as the shortest job first algorithm, requiring information such as the total number of orders, arrival time, and cooking time. If the user enters all of the information, the average waiting time and average turnaround time will be calculated and shown. Priority scheduling is the last process-scheduling algorithm in our program. This algorithm begins when the user enters 3, and it will ask the user for the total number of orders, as well as the same inquiries as the previous two scheduling algorithms.

5 Results & Discussion

Process scheduling is a frequently overlooked determinant of real-time performance. We have shown some of the consequences of changing the scheduling algorithms on several value function features, especially in the face of an overload condition. We believe that, although being the fundamental attribute of a real-time system, the time value of process completion has been overlooked when evaluating real-time performance. The project used shortest job first scheduling Shortest Remaining Time first and Priority scheduling. As mentioned before, we are trying in this work to increasing the efficiency of food preparation and food delivery. Five processes have been defined with CPU Burst Time (Cooking Time), these five processes are scheduled in out three proposed algorithms. The order number, average wait time, and average turnaround time were all calculated and compared. We conducted many experiments to achieve this, but we will just explain two of them here because the data analysis will stay same.

Shortest Job First (SJF) scheduling

Experiment 1:

Consider the below processes available in the ready queue for execution, with arrival time as 0 for all and given Burst Time (Cooking Time).

Table 1 Experiment 1 for SJF

Order	Burst Time (Cooking Time)
01	21
02	3
03	6
04	2

In shortest job first scheduling, the shortest process is executed first.

Gantt chart

P4	P2	P3	P1

0 2 5 11 32

Table 2 Result of Experiment 1 for SJF

Order	Burst Time (Cooking Time)	Waiting time	Turnaround time
04	2	0	2
02	3	2	5
03	6	5	11
01	21	11	32

Average Waiting Time = 4.50 min

Average Turnaround Time = 12.50 min

Experiment 2:

In the following experiment, there are five jobs. Their arrival time and Burst Time (Cooking Time) are given in the table below.

Table 3 Experiment 2 for SJF

Order	Burst Time (Cooking Time)	Arrival Time
01	8	0
02	10	2
03	6	3
04	3	7
05	5	0
06	7	5

Gantt chart

P5	P3	P4	P5	P1	P2

0 5 11 14 21 29 39

Table 4 Result of Experiment 2 for SJF

Order	Burst Time (Cooking Time)	Arrival Time	Waiting time	Turnaround time
01	8	0	21	29
02	10	2	27	37
03	6	3	2	8
04	3	7	4	7
05	5	0	0	5
06	7	5	9	16

Average Waiting Time = 10.50 min

Average Turnaround Time = 17.00 min

Shortest Remaining Time First (SRTF)

Experiment 1:

In the following experiment, we use four orders their arrival time and Burst Time (Cooking Time) are given in the table below.

Table 5 Experiment 1 for SRTF

Order	Burst Time (Cooking Time)	Arrival Time
01	8	0
02	4	1
03	9	2
04	5	3

Gantt chart

P1	P2	P4	P1	P3

0 1 5 10 17 26

Table 6 Result of Experiment 1 for SRTF

Order	Burst Time (Cooking Time)	Arrival Time	Waiting time	Turnaround time
01	8	0	9	17
02	4	1	0	4
03	9	2	15	24
04	5	3	2	7

Average Waiting Time = 6.50 min

Average Turnaround Time = 13.00 min

Experiment 2:

In the following experiment, we use six orders their arrival time and Burst Time (Cooking Time) are given in the table below.

Table 7 Experiment 2 for SRTF

Order	Burst Time (Cooking Time)	Arrival Time
01	8	0
02	4	1
03	2	2
04	1	3
05	3	4
06	2	5

Gantt chart

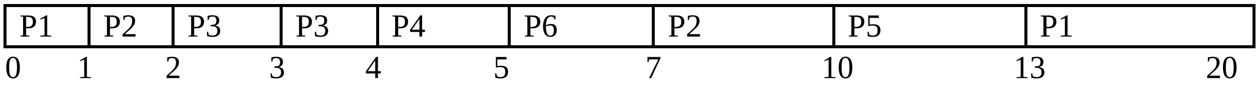


Table 8 Result of Experiment 2 for SJF

Order	Burst Time (Cooking Time)	Arrival Time	Waiting time	Turnaround time
01	8	0	12	20
02	4	1	5	9
03	2	2	0	2
04	1	3	1	2
05	3	4	6	9
06	2	5	0	2

Average Waiting Time = 4.00 min

Average Turnaround Time = 7.33 min

Priority scheduling

Experiment 1:

In the following experiment, we use seven orders their priority and Burst Time (Cooking Time) are given in the table below.

Table 9 Experiment 1 for Priority scheduling

Order	Priority	Burst Time (Cooking Time)
01	2	3
02	6	5
03	3	4
04	5	2
05	7	9
06	4	4
07	10	10

Gantt Chart

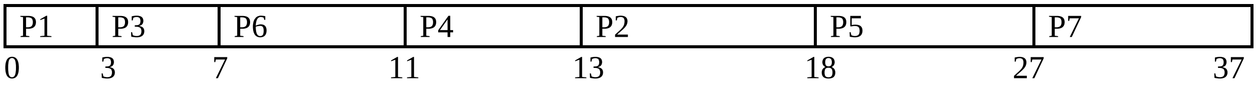


Table 10 Result Experiment 1 for Priority scheduling

Order	Burst Time (Cooking Time)	Waiting time	Turnaround time
01	3	0	3
02	5	13	18
03	4	3	7
04	5	7	13
05	9	18	27
06	4	7	11
07	10	27	37

Average Waiting Time = 11.29 min

Average Turnaround Time = 16.57 min

Experiment 2:

In the following experiment, we use four orders their Priority and Burst Time (Cooking Time) are given in the table below.

Table 11 Experiment 2 for Priority scheduling

Order	Burst Time (Cooking Time)	Priority
01	5	5
02	8	6
03	2	8
04	12	1

Gantt Chart

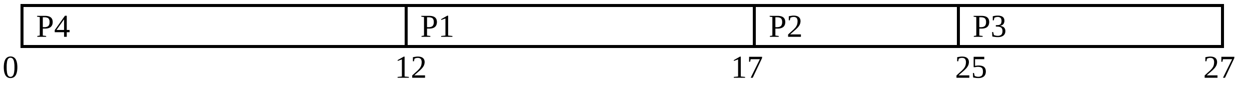


Table 12 Result Experiment 2 for Priority scheduling

Order	Burst Time (Cooking Time)	Priority	Waiting time	Turnaround time
01	5	5	12	17
02	8	6	17	25
03	2	8	25	27
04	12	1	0	12

Average Waiting Time = 13.50 min
Average Turnaround Time = 20.25 min

This usage of value functions to explain the performance of current scheduling algorithms is a first step toward our aim of eventually developing a scheduler algorithm that makes scheduling choices directly using such value functions. In the case that certain processes must be delayed or cancelled due to a (potentially transitory) overload, such a scheduler would seek to improve the overall system value by ensuring that activities delivering the highest value to the system are prioritized. This will give a degree of stability to overload processing in a real-time system that is now unavailable, allowing more effective use of the existing resources while decreasing system costs and benefiting our food ordering system.

6 Future work & Conclusion

Understanding process scheduling is extremely vital in operating system. Given that there could be hundreds of programs that needed to run, process scheduling will allocate resources among all the competing processes. For example, it will determine which process will own the CPU for execution while another process is still on hold. This will allow us to obtain the maximum utilization of the CPU with multi-programming. Hence, an increase in efficiency resulted in the user able to get the minimum response time for the execution of programs.

In addition to that, we also learned that every scheduling algorithm has their own advantage and disadvantages. For example, the SJF algorithm is arguably the best when it comes to getting the minimum average waiting time for the processes. On the other hand, one of the disadvantages of SJF algorithm is that it can cause starvation when it deals with short processes. Besides that, the priority scheduling is another algorithm that is very useful for defining processes based on their importance. A heavy demanding task should be prioritised by the CPU. Priority scheduling may also cause problem. For example, a higher priority process can take a lot of CPU time that end up causing starvation to the lower priority process. Lastly, the SRTF algorithm is the faster version of SJF algorithm. The downside of SRTF is that when context switching occurs, it consumes a lot of CPU time while processing.

In conclusion, this project was extremely beneficial for us to enhance our existing knowledge regarding process scheduling. The implementation of real-world application through programming simulation made us able to picture the scheduling running in the computer. We were able to do the calculation in order to get the waiting time and turnaround time for every scheduling algorithm.

Dedication and Acknowledgement

This final project report has been completed successfully with the dedication to our Operating System Lecturer, Dr. Syed Hamid Hussain Madni. I appreciate that he has been teaching us throughout the semester with passion and kindness. He will answer our doubt without hesitation although he was busy with something else. Without him, we would not have much knowledge of this subject and might not be able to complete this project. Furthermore, Dr. Madni helped us a lot when conducting the project. As we are fairly new to the components and steps to conduct the project, she was willing to assist us group by group without any hesitation.

Next, we would like to thank our friends and course mate who supports us when conducting this project. We helped one another to make sure everyone was not left behind.

Finally, we would like to thank our parents for giving moral support and financial support for us to continue our study in Universiti Teknologi Malaysia.

References

Akhtar, M., Hamid, B., Ur-Rehman, I., Humayun, M., Hamayun, M., & Khurshid, H. (2015). . Appl. Environ. Biol. Sci. *An Optimized Shortest job first Scheduling Algorithm for CPU Scheduling, 5*(12), 42-46.

Çalhan, A. (2016). A Non-Preemptive Priority Scheduling Algorithm for Improving Priority Data Transmission Delay in Wireless Body Area Networks. *Ad Hoc & Sensor Wireless Networks, 34*, 59-75.

javatpoint. (2011). Retrieved from Shortest Remaining Time First (SRTF) Scheduling Algorithm: https://www.javatpoint.com/os-srtf-scheduling-algorithm

javatpoint. (2011). Retrieved from Shortest Job First (SJF) Scheduling: https://www.javatpoint.com/os-sjf-scheduling

Li, Z., & Ierapetritou, M. (2008). Process scheduling under uncertainty: Review and challenges. *Computers & Chemical Engineering*, 715-727. doi:https://doi.org/10.1016/j.compchemeng.2007.03.001

Rungta, K. (2021, June 22). *CPU Scheduling Algorithms in Operating Systems*. Guru99. https://www.guru99.com/cpu-scheduling-algorithms.html

techopedia. (2021). Retrieved from Priority Scheduling: https://www.techopedia.com/definition/21478/priority-scheduling