COMP 2432 Group Project

(2023/2024 Semester 2)

Group 43

| Student ID | Student Name |
| --- | --- |
| 23028215D | Huang Baoxian |
| 23035686D | WONG Yuk Yu |
| 23028054D | WONG CHUN KIT |

1. Abstract

This project aims to create a customized Production Line Scheduler (PLS) application and the need to optimize the steelmakers' scheduling process. Currently, the manufacturer has three production lines but the production efficiency of the production lines is insufficient. The primary goal now is to address the underutilization of factory capacity caused by suboptimal scheduling practices. This PLS application utilizes common process scheduling algorithms. For example, First-Come, First-Served (FCFS), Shortest Job First (SJF), and a proprietary Priority-based algorithm developed and solved the problem.

The PLS has four main modules. The first one is an input module for obtaining order details. The second one uses different algorithms to internally optimize the utilization rate in the factory to ensure that there is no waste of performance. The third output module of the current factory scheduling will be displayed, and the last one will obtain different scheduling strategies based on the score.

Early test results based on PLS indicate that the new system will significantly increase factory utilization and reduce waste.

This report will introduce in detail the development methods and performance of pls and explain the efficiency aspects of different algorithms.

2. Introduction

Currently, this supplier has three different production lines, and they hope to produce different steel products through these productions. Moreover, the daily output is different, and they need to produce 300, 400, and 500 products per day. However, there is currently a problem in the factory because Due to poor allocation efficiency, orders are often unable to be completed, which directly affects the company's financial reporting problems.

According to the survey results, the factory lacks an effective production scheduling system, which will lead to insufficient utilization of the current production efficiency and cause the factory to waste resources. Therefore, the factory's order processing efficiency is low and some orders need to be canceled before other orders can be completed.

So this project hopes to use the Production Line Scheduler (PLS) application to solve these critical problems. The system's capabilities use advanced scheduling algorithms to automatically optimize and increase productivity. Through different algorithms, it can be adjusted according to the needs of the factory to ensure that the manufacturing efficiency of each factory is fully used and no manufacturing efficiency space is wasted.

Finally, this system can also simplify the production process, use different algorithms for different factories, and regularly generate reports to analyze the current factory utilization rate, which can greatly improve the efficiency of factory manufacturing.

3. Scope/Related work

1. Process Scheduling:

Because this system uses a process scheduling algorithm to manage the production processes and plans of different factories, each process will be optimized and executed using multiple processes to ensure that the CPU usage is not too high. This effectively manages the production rate of the factory and ensures Factory resources are maximized.

2. Concurrency and Synchronization:

This system can handle multiple production lines and ensure that operations are synchronized without conflicting concurrent processes in the system.For example, orders can be processed and production plans analyzed at the same time.

3. Resource Management:

The system will ensure that resources are released each time different analysis and allocation processes are completed to ensure that excessive system resources will cause system failure. Because the system needs to process the productivity of multiple factories for a long time to optimize the resource allocation of the factories, the system needs to constantly calculate. If the resources are not released properly, the burden on the system will increase.

4. Priority Scheduling:

The system will also use priority production permissions. For example, products starting with "Product A" will be completed first. This way, different products can be prioritized and ensure that related production work can be completed on time.

5. Fork

The system uses the fork() system call to create child processes, enabling parallel execution of tasks. The software is divided into several different modules, each dealing with a specific aspect of the production process.

6. Pipe

PLS utilizes the inter-process communication (IPC) mechanism for message passing and shared memory, allowing different modules to send data to other modules or receive new data at runtime

4. Concept

1. First-Come First-Served (FCFS)

In the system, an algorithm will prioritize the sequential orders received. Orders are arranged from top to bottom, so the factory's production orders will start with the orders received from top to bottom until reaching the last one of the entire order. But there will be a problem, that is, larger orders will affect the waiting time of smaller orders because larger orders require longer production time.

2. Shortest Job First (SJF):

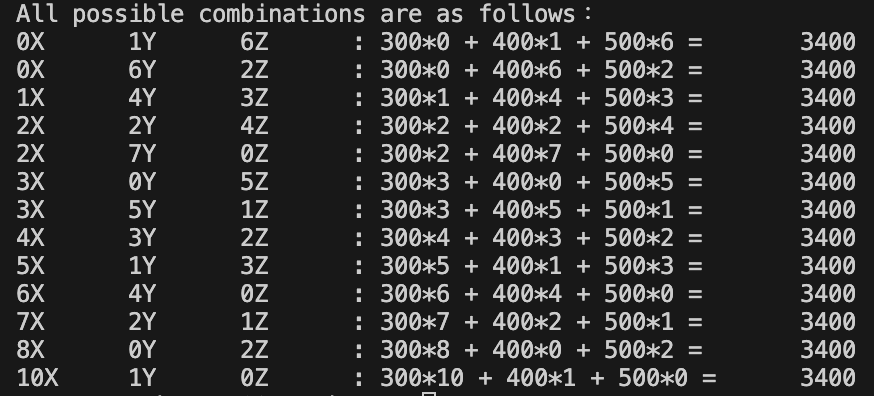
This algorithm prioritizes processing of the smallest order, which can reduce the need to wait longer when the order is large. Also, it will reduce the average production time and therefore increase the steel output.

3. Priority Scheduling (PR):

This algorithm will classify orders according to product names and will prioritize orders according to alphabetical order(A-Z). For example, if the product name is Product A, it will be the priority product for this production line.

#### Handling different combinations

After assigning the order to the order list. PLS will determine the number of days each order will take to be processed at the plant. For example, depending on the number of orders (e.g 3400) and the production capacity of each plant, PLS will find all possible combinations and assign the minimum number of the days, and allocate 1 days to plant X, 4 days to plant Y, and 3 days to plant Z to fulfill the order. Depending on the number of orders and the capacity of the plant, there can be different combinations of plant utilization. This is the first time to find all possible combinations. And allocate the best combinations to all 3 plants, which means use the fewest days to produce products.

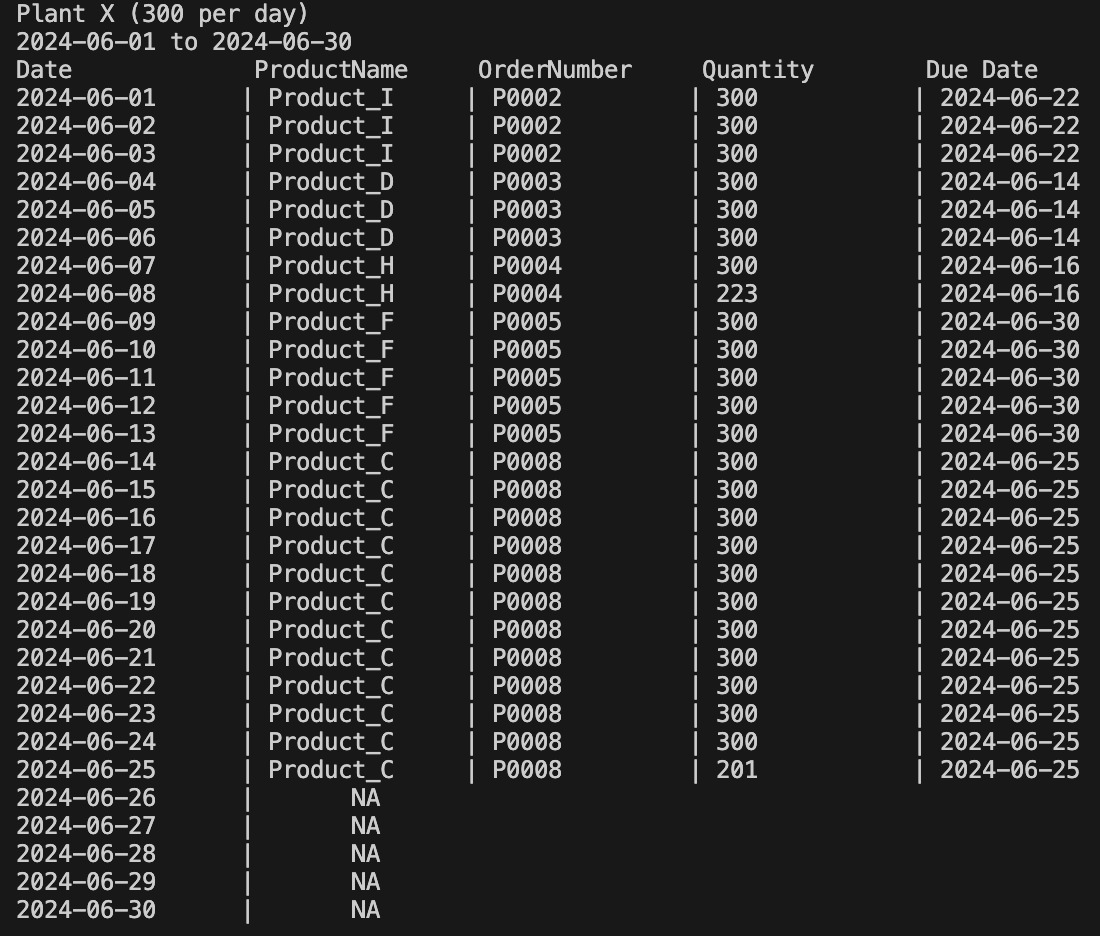


When scheduling production, PLS takes into account the requirements of the due date of the orders. It will try different combinations to determine how best to use the resources of the three plants to meet the order.

If plant Y available days before product due date are less than 4 days, PLS will find other combinations such as allocate 1 day to plant X, 4 days to plant Y, and 3 days to plant Z to fulfill the order. This is the second time to find all possible combinations. And allocate the fit case to all 3 plants.

When all orders in the list are allocated. PLS will add the days used in the different plant schedules. If the days used plus current days are over the due date, the system will find another combinations, until all combinations could not be allocated. Then the order will be rejected.

Once all the orders are allocated to the respective plants based on their availability and due dates, the PLS will calculate the total number of days used by summing up the individual durations for each plant schedule. This cumulative duration reflects the overall production timeline needed to complete all the orders successfully.

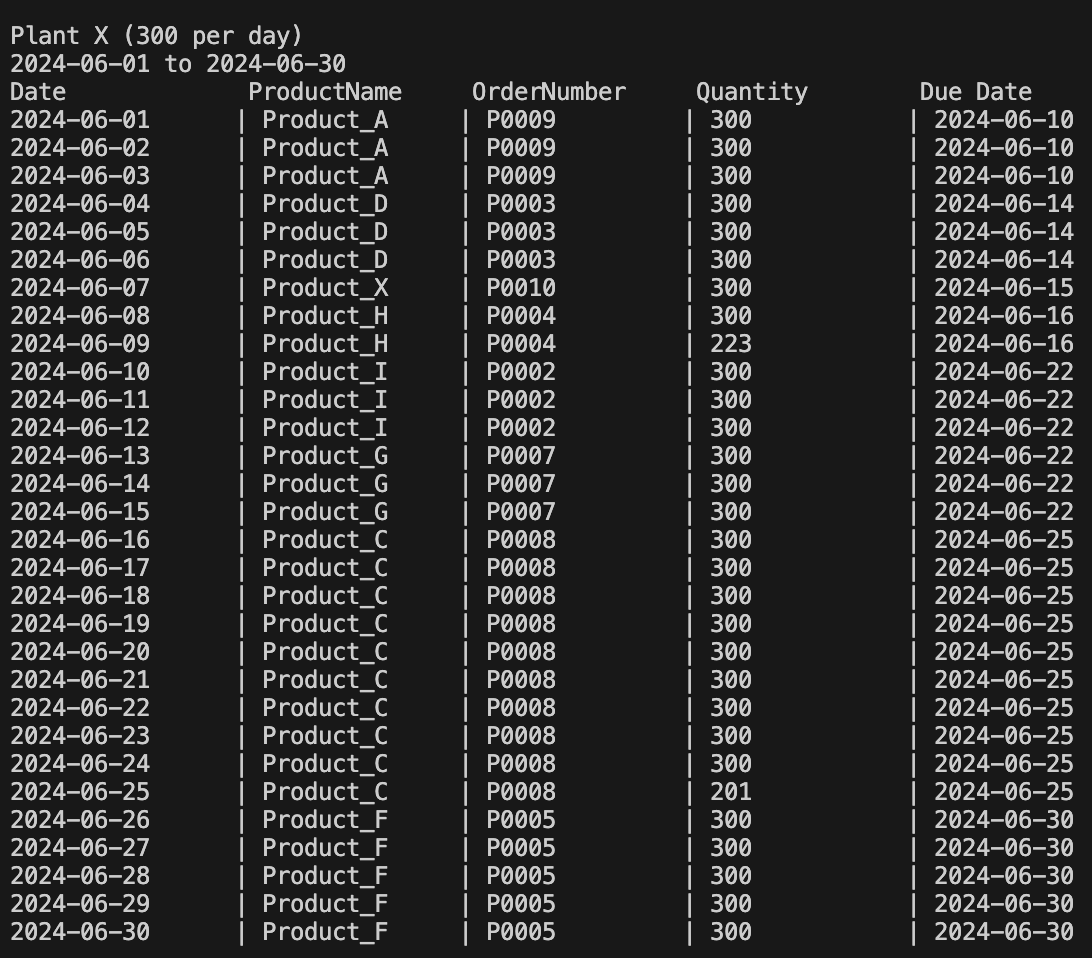


5.Own scheduling algorithm

DDF (Due Date First) is a scheduling algorithm that prioritizes tasks or orders based on their respective deadlines (which is due date). It aims to complete tasks with the earliest deadline first, ensuring timely delivery or execution of time-sensitive work.

In DDF scheduling, each task or order is associated with a specific due date, which represents the time by which it needs to be completed. The algorithm examines the deadlines of all pending tasks and selects the one with the earliest deadline for execution. This ensures that critical or urgent tasks are given the highest priority.

In PLS, the order list will be sorted by due date and allocate the order list in best combinations to all 3 plants. In order to finish the plant's schedule.



6.Software structure of system

The Steel-making Production Line Scheduler (PLS) is designed using a modular architecture that helps enhance maintainability and scalability. The software is divided into several different modules, each dealing with a specific aspect of the production process.

Different modules can be implemented as separate programs in the form of subroutines created by the parent process through the fork() system call. To send messages to different modules, pipe() and associated write() / read() system calls are used, and the ring communication is used..

Below is a detailed breakdown of each module within the PLS:

## Input Module:

The input module is the parent process and is responsible for collecting user input data, processing user input, mapping commands to corresponding functions, and sending and receiving messages between child processes through pipes.

Information that needs to be collected includes start dates and end dates, order details: order number, expiration date, quantity and product name, as well as the algorithm and report name.

## Scheduling module:

As a child process of the parent process (input module), the scheduling module receives data from the parent process. The core of PLS, implements various scheduling algorithms (FCFS, SJF, Priority and DDF) to optimize the use of production resources.

The scheduler module schedules different orders accordingly based on the algorithm selected in the input module. It calculates the optimal production schedule by allocating tasks among available factories to maximize utilization and meet delivery dates.

After completing the scheduling of different orders, the data will be sent to the output module through the pipe.

## Output Module:

The output module receives data from the Scheduling module and prints out schedules for the factory manager. Formats and displays the generated schedules on the command line.

When completed, the data will be sent to the analyzer module through the pipe.

## Analyzer Module:

The Analyzer module receives data from the Output module and analyzes the performance of the scheduling algorithm by generating reports on various indicators such as factory utilization, number of orders processed, and incidence of rejected orders. Users can compare the utilization rates of different algorithms.

After the calculation is completed, a report will be generated, and the name will be based on the report name provided by the user in the input module. After generating the report, the system will return to the input module for the user to enter new data.

This section provides a comprehensive overview of the software's structure, clearly explaining the role of each component. This detailed description should help to effectively illustrate the complex design and functionality of PLS ​​applications.

7.Testing cases/Assumptions

#### Assumptions

Same arrival time (AT) :In this project, we assume that all products have no arrival time or have the same arrival time because products have no start date. And the FCFS algorithm depends on order number.

Order Independence: The scheduling algorithm assumes that the orders are independent of each other, meaning the completion of one order does not depend on the execution or completion of any other order.

Fixed Resource Capacity: The algorithm assumes that the capacity of each plant (Plant X, Plant Y, Plant Z) remains fixed and does not change during the scheduling period. There are no constraints or limitations on the availability or capacity of resources.

No Preemption: The algorithm assumes that once a task or order is assigned to a plant, it continues to be processed without interruption or preemption. There are no interruptions or pauses in the production process.

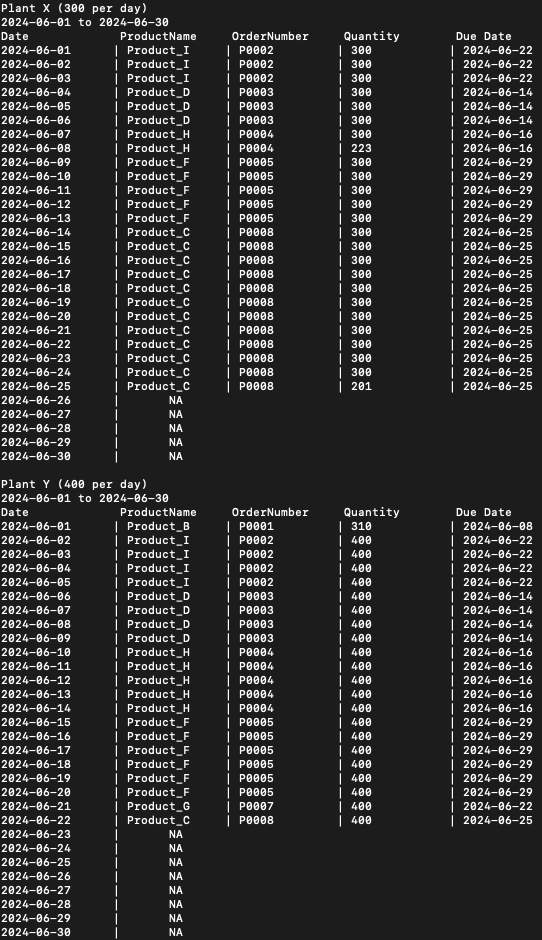
Deterministic Execution Time: The algorithm assumes that the execution time for each order is known and deterministic, meaning the time required to complete an order is fixed and does not vary.

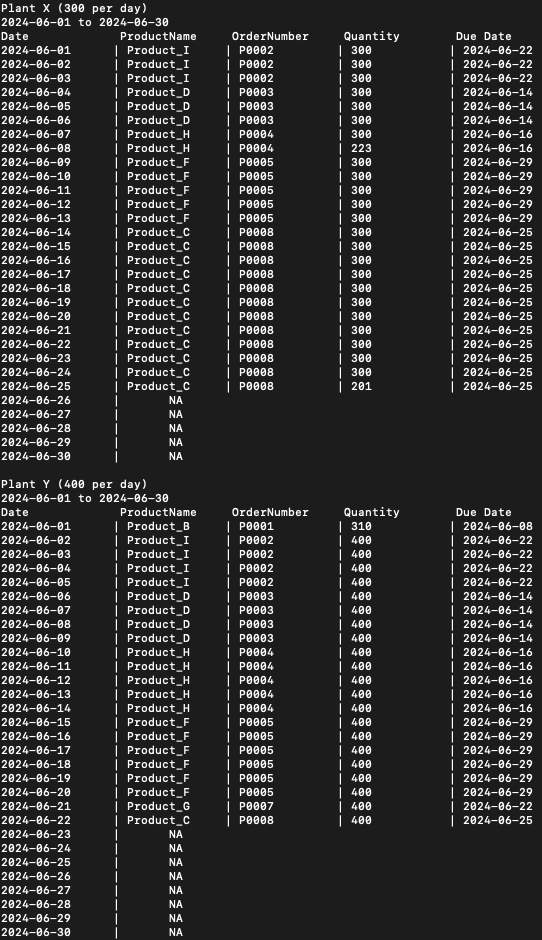
Deadline Consistency: There are no changes or modifications to the deadlines once the scheduling process begins.

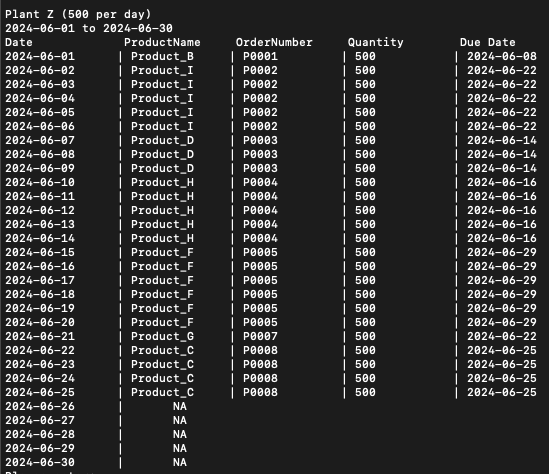
No External Factors: The algorithm assumes that there are no external factors or disruptions that could affect the scheduling process, such as machine breakdowns, material shortages, or unexpected events.

No rest day in the period: The three plants work 24x7, no need to consider holiday.

#### FCFS

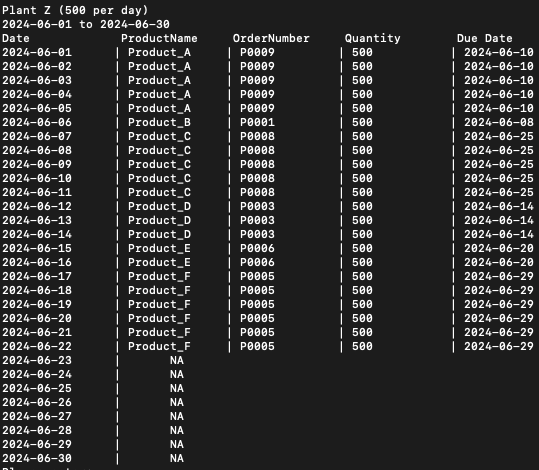




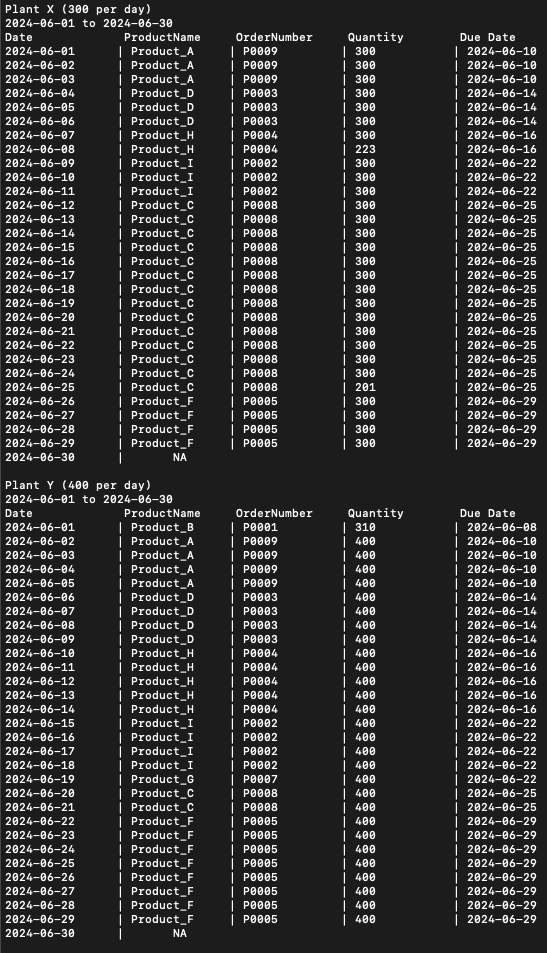


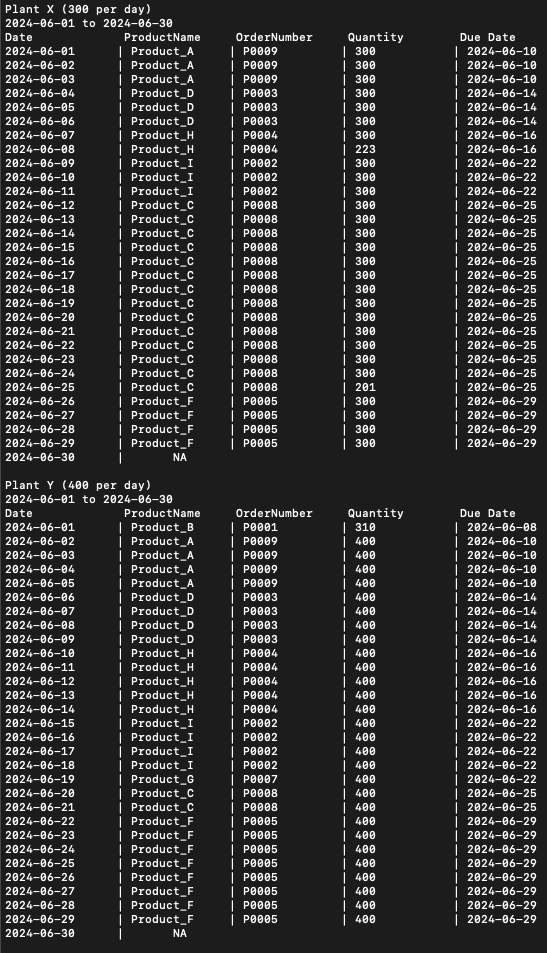
#### SJF

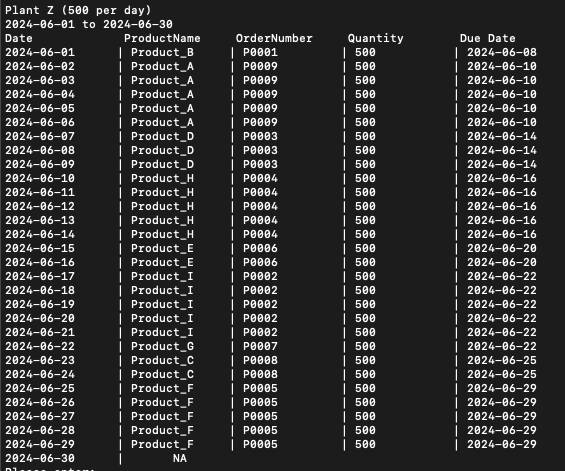
#### PR



#### DDF





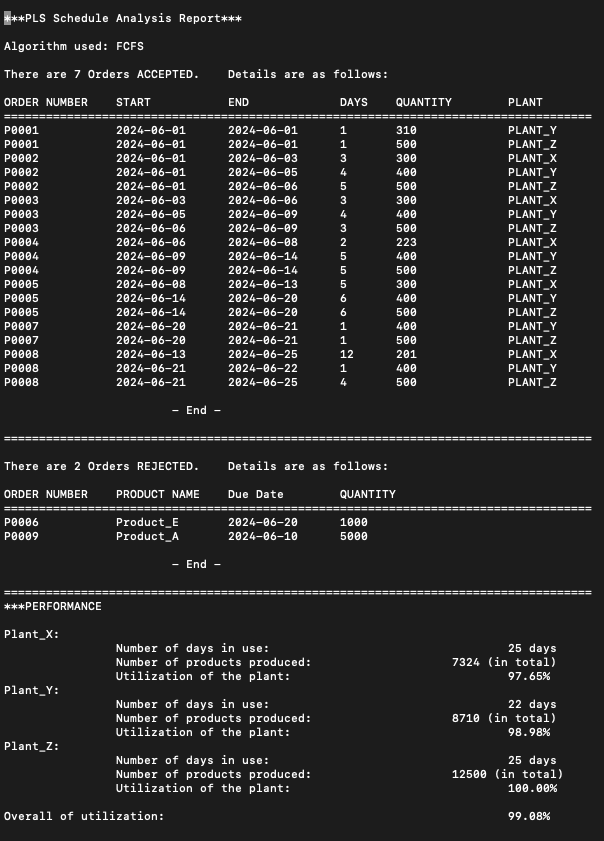


8.Performance analysis

FCFS (First-Come, First-Served) Algorithm:

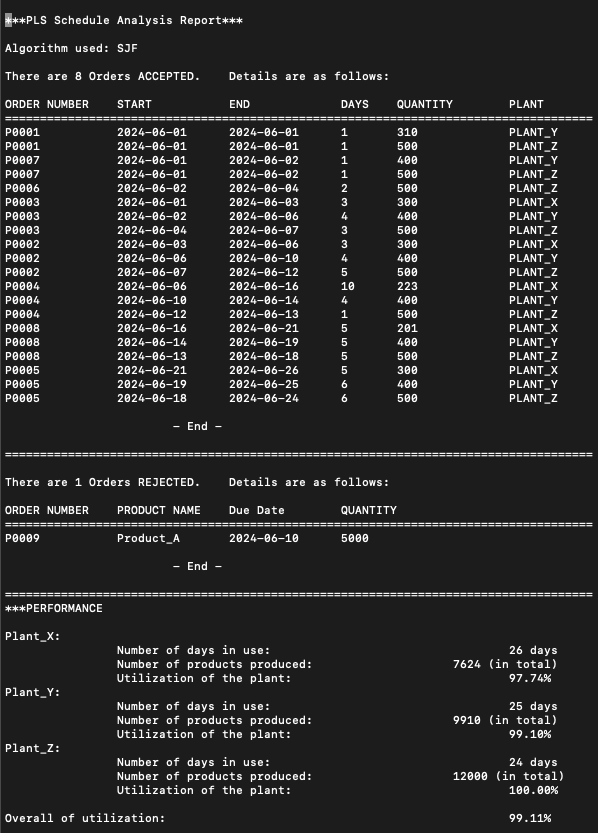
Pros: Processes orders in the order of arrival, ensuring fairness.

Cons: Cannot optimize based on urgency or execution time of orders.



SJF (Shortest Job First) Algorithm:

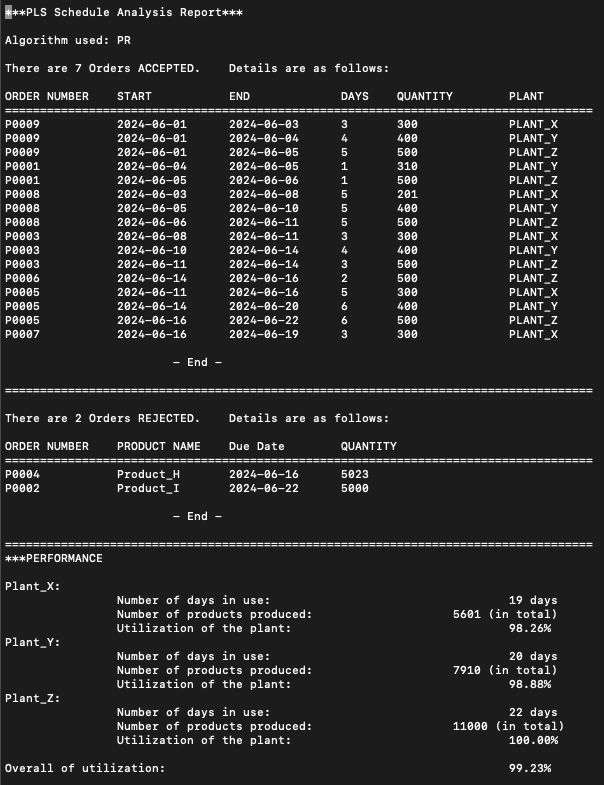
Pros: Prioritizes orders with shorter execution times, reducing waiting times and improving system responsiveness.

Cons: Long jobs have to wait for short jobs to complete.

PR (Priority Scheduling) Algorithm:

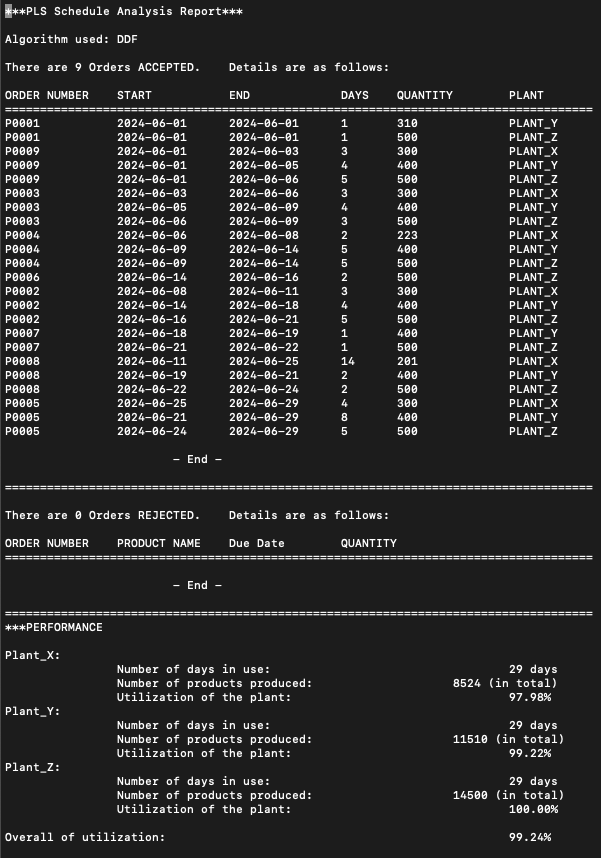
Pros: Allows scheduling based on priority of orders, providing flexibility to set different priority strategies based on business requirements.

Cons: Improper priority assignment may result in longer waiting times for low-priority orders.



DDF (Earliest Deadline First) Algorithm:

Pros: Schedules orders based on their deadlines, ensuring timely delivery of urgent orders.

Cons: May increase waiting times for other orders and fail to fully utilize resources.

In conclusion, the choice of scheduling algorithm depends on specific business requirements and optimization goals. If urgency and timely delivery are crucial, DDF or PR algorithms can be considered. If reducing average waiting times and improving system responsiveness are priorities, SJF algorithm can be chosen. FCFS algorithm, on the other hand, is simple and intuitive, suitable for scenarios where fairness is of high importance. The ultimate decision should consider a comprehensive evaluation of business requirements and performance metrics.

9.Program set up and execution

## Preparation:

* Environment: Linux (apollo)
* File:
* PLS\_G43.c
* record.txt
* orderBATCH01.dat

## Compile the program:

gcc PLS\_G43.c -o PLS\_G43

* There are no additional libraries in the program, and standard compilation statements are used.

## Run the program:

./PLS\_G43

## Enter start and end dates:

Format: addPERIOD [start date] [end date]

E.g. addPERIOD 2024-06-01 2024-06-30

* When entering the program, you must first enter a start and end date.

## Add single order:

Format: addORDER [Order Number] [Due Date] [Quantity] [Product Name]

E.g. addORDER P0001 2024-06-10 2000 Product\_A

## Add multiple orders:

Format: addBATCH [Orders in a batch file]

E.g. addBATCH orderBATCH01.dat

## Generate the schedule and report:

Format: runPLS [Algorithm] | printREPORT > [Report file name]

E.g. runPLS FCFS | printREPORT > report\_01\_FCFS.txt

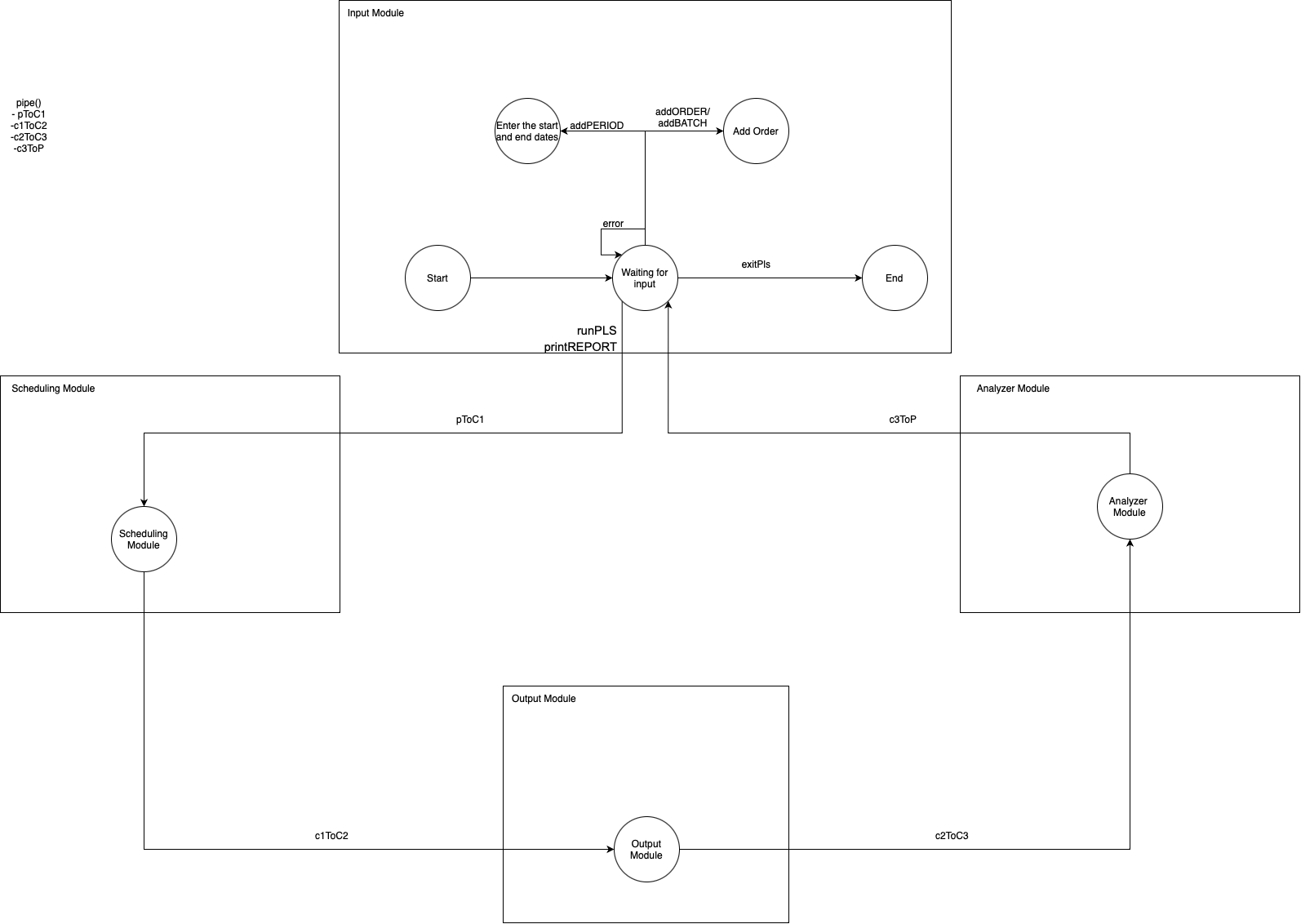
## 

## Exit the program:

Format: exitPLS

* This will completely terminate the program after collecting all child processes and closing all pipes and files.

10.Results/graphs/figures



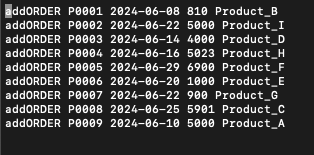
11. Conclusion

This project successfully implemented the Production Line Scheduler (PLS) to address and enhance the scheduling processes of a steel manufacturer with multiple production lines. Through the deployment of several scheduling algorithms—First-Come First-Served (FCFS), Shortest Job First (SJF), Priority Scheduling (PR), and our innovative Due Date First (DDF)—the PLS has significantly improved production efficiency and resource utilization across the factory's operations.

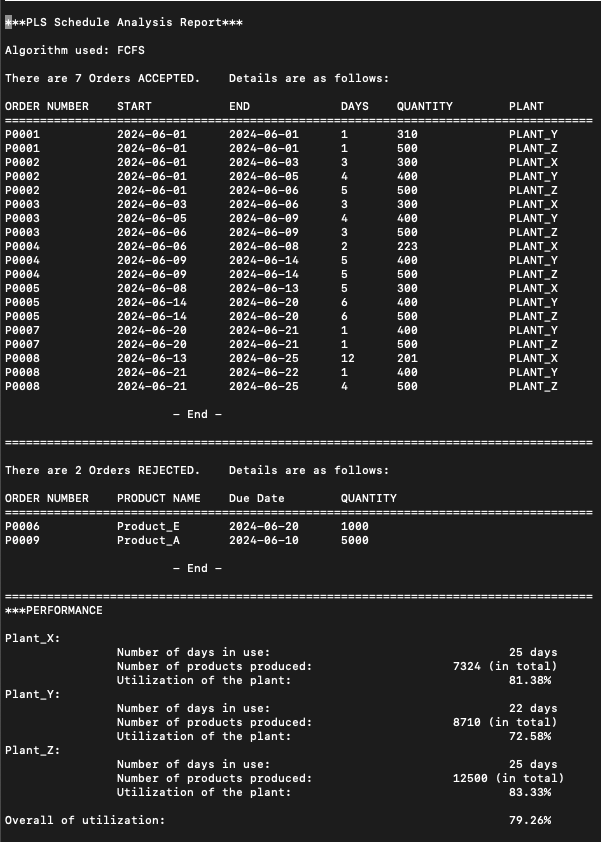
The comparative analysis of these scheduling methods revealed that each algorithm serves specific operational needs effectively. The DDF algorithm, in particular, demonstrated the highest utilization rate, making it the most efficient in scenarios where minimizing waiting times and maximizing throughput are critical. Meanwhile, the DDF algorithm introduced a strategic focus on deadline-centric tasks, ensuring that critical orders are processed timely, which is essential for maintaining stringent delivery schedules.

12. Innovative solution/Algorithm

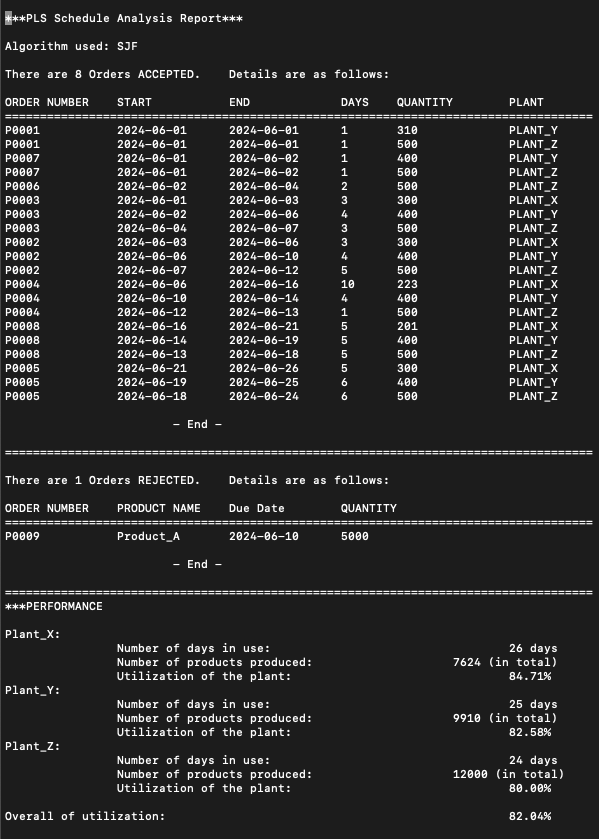
Test case:



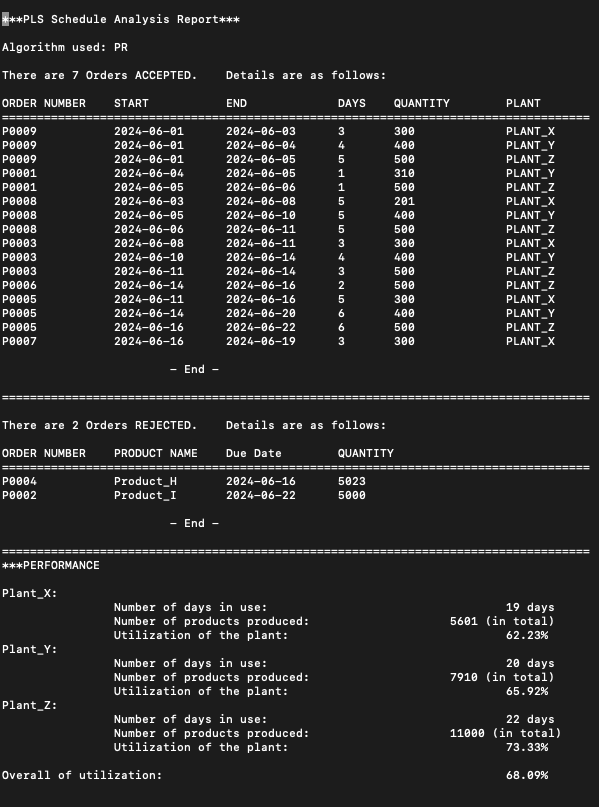
FCFS:



SJF:



PR:



Based on the percentages, the utilization rates for FCFS, SJF, and PR scheduling algorithms are as follows: FCFS - 79.26%, SJF - 82.04%, and PR - 68.09% in our test case. These percentages represent the total number of days each plant is in use for production compared to the total number of days in the analyzed period.

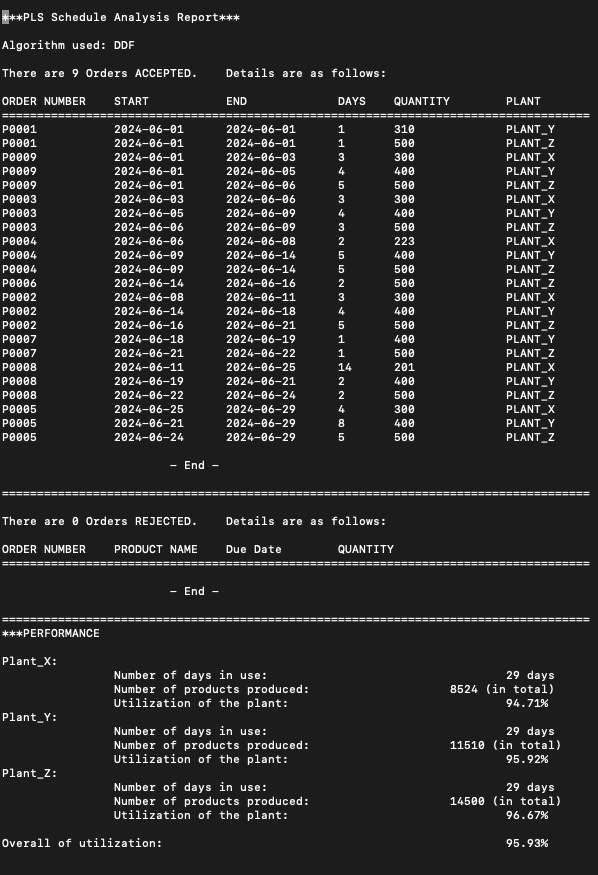
From these results, it can be observed that the SJF algorithm has the highest utilization rate (82.04%), followed by FCFS (79.26%), and PR (68.09%). This indicates that SJF utilizes the manufacturing plants more efficiently, making better use of available resources compared to FCFS and PR algorithms.

Therefore, based on these utilization rates, the order of efficiency for the given scheduling algorithms can be ranked as follows: SJF > FCFS > PR.

We have an innovative algorithm to increase the percentages which is the Due Date First algorithm.

DDF:

The DDF (Due Date First) scheduling algorithm is an innovative method designed to enhance the efficiency of production scheduling by prioritizing tasks or orders according to their deadlines. This approach ensures that tasks with the nearest due dates are addressed first, promoting the timely completion of critical.



13.Appendix

GitHub:

<https://github.com/DanielWong612/PLS_G43>

