OPERATING SYSTEMS LAB

CSL-320

PROJECT REPORT

Project Title: **SCHEDULING ALGORITHMS**

Supervised By: **MUSLIM AHMED**

Group Members:

NAME	EMAIL	ENROLLMENT
Abdullah Abdul Wahid	Abdullahw72@gmail.com	02-134192-015
Fazeel Zafar	Fazeel332@gmail.com	02-134192-010
Shahood Sajid	Shahood.sajid@hotmail.com	02-134192-089

Department Of Computer Science

Bahria University, Karachi Campus

TABLE OF CONTENTS

Acknowledgement

After days of hard work and brainstorming we have successfully completed our project of “CPU Scheduling Algorithms”.

I would like to thank out teacher, Sir Muslim Ahmed, for helping us throughout the course. He showed great respect towards all the students and helped us in understanding all the concepts of our course and made sure that we implemented them properly in this project. We are thankful for his guidance and motivation

We would also thank Sir Shahid Khan for his support as well. He taught us all the concepts of Operating Systems very thoroughly and also made sure that we were implementing every concept perfectly in our assignments. He also took time out of his busy schedule to listen to our queries and helped us clear them out.

Finally we would like to thank each and every person who was involved in this project, whether they were our classmates or teachers or colleagues.

Introduction

We have understood the concept of scheduling algorithms in our theory class, what's their purpose, how they're implemented and which algorithm is best suited in a particular scenario.

This project is an implementation of those algorithms. It shows how these algorithms work in the backend of our Operating System. We have shown how each algorithm performs and the result it generates.

Problem Statement

Sometimes when there are too many processes running at a time, they all may try to access a resource or multiple resources at the same time, which can cause "resource starvation" or create a deadlock. This is why scheduling algorithms are used, to determine which algorithm, in the most efficient way, can schedule all processes seamlessly.

Objective

The primary objective of this project is to show our understanding of the concepts of this project, scheduling algorithms, and to also show how all these algorithms work in the backend of an operating system. The user will enter the burst and arrival times and the program will show and output, along with Gantt chart. This will help a user understand how an algorithm is performing and which one would be the best suited for their problem.

Project Summary

In this project, we are going to design scheduling algorithms. On the basis of these scheduling algorithms, different processes are assigned to the CPU. These algorithms are pre-emptive or non-pre-emptive. Pre-emptive algorithms may process high priority against low priority process. Non-pre-emptive algorithms works in a way that it does not preempt the process until it is completed in its time. The main objectives achieved through these algorithms are to get the max CPU utilization, CPU memory allocation, min turnaround time, min waiting time. We will be using 4 different algorithms in Linux operating system and it will be implemented in Python 3 language.

Purpose

Our project is to make a CPU Scheduler that will be used in an operating system. All processes in a system wait
for some kind of I/O operation to occur like fetching data from storage, requesting a resource to be allocated
etc.
The purpose of CPU scheduler is to keep allocating processes to a CPU whenever it is idle and to make sure
that all resources are fairly allocated to each process for a certain time to avoid any deadlocks, and to make
the system efficient, responsive and fast.
We will implement 4 types of Scheduling Algorithms:

- **First-Come-First-Serve (FCFS)**
- **Shortest Job First Scheduling (SJF)**
- **Priority-Based Scheduling**
- **Round Robin Scheduling**

Project Scope

This project can be used in two situations, one where we can test the efficacy of each algorithm and determine which algorithm will be the best for our use.
Another use of this program is to demonstrate students how every algorithm performs. Through this, we can show the results of each algorithm with a visual representation of the Gantt chart.

System Design and its Specification

Hardware Limitations

- Quad Core x64 processor or above
- 4gb RAM minimum or above
- Minimum 20gb hardware required

Software Limitations

- Visual Studio Code or any other code editor
- Virtual Machine Workstation with Linux OS
- Microsoft Windows 7 or above

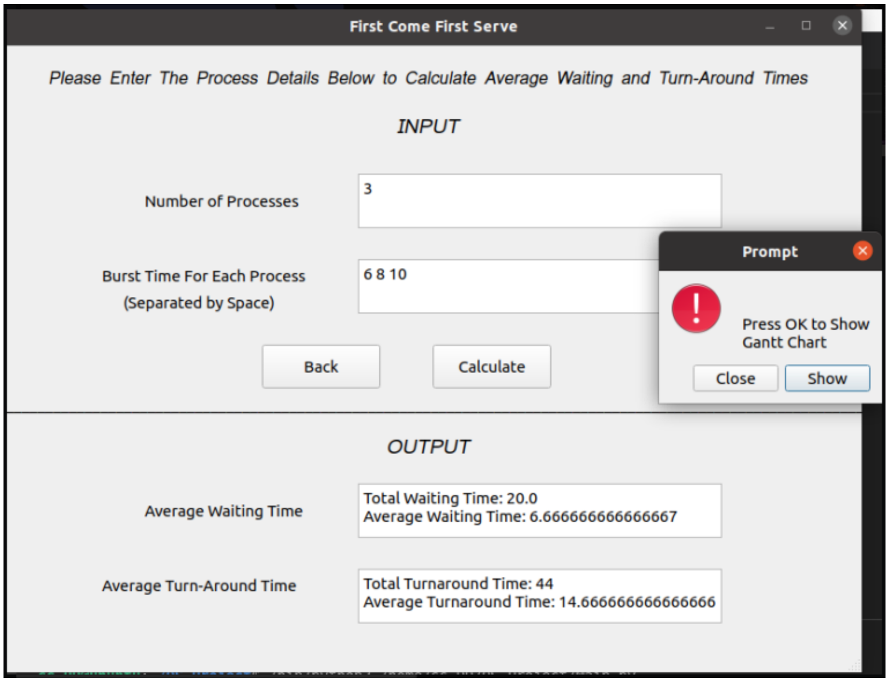
Features

- ✦ Main page to select an algorithm

- ✦ First-Come-First-Serve (FCFS)
- ✦ Shortest Job First Scheduling (SJF)
- ✦ Priority-Based Scheduling
- ✦ Round Robin Scheduling

- ✦ Gantt Charts for all the Algorithms

Screenshots



Please Enter The Process Details Below to Calculate Average Waiting and Turn-Around Times

INPUT

Number of Processes: 3

Burst Time For Each Process (Separated by Space): 6 8 10

Press OK to Show Gantt Chart

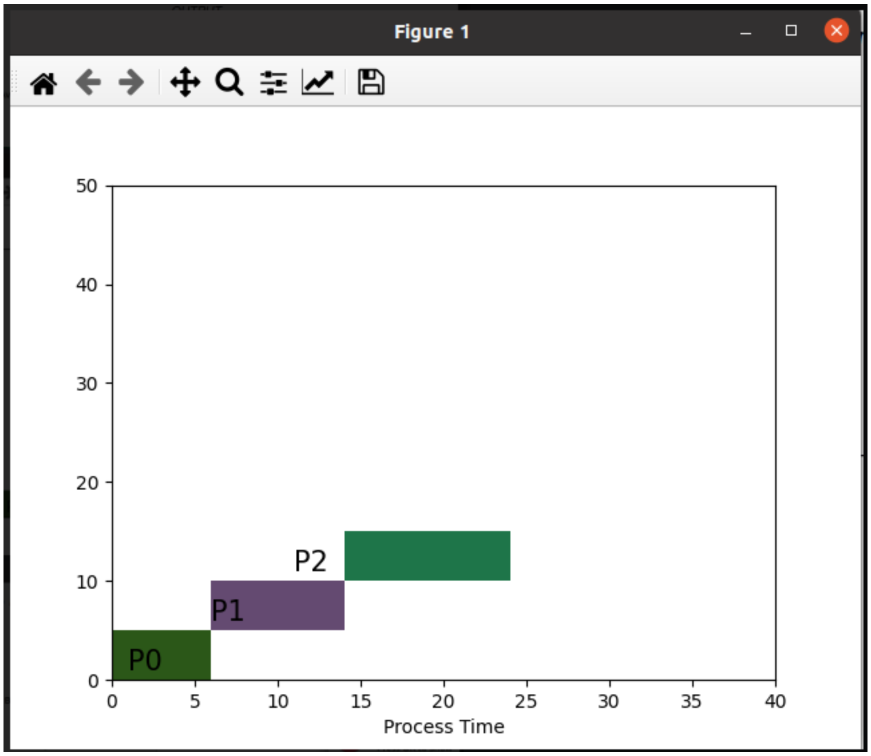
OUTPUT

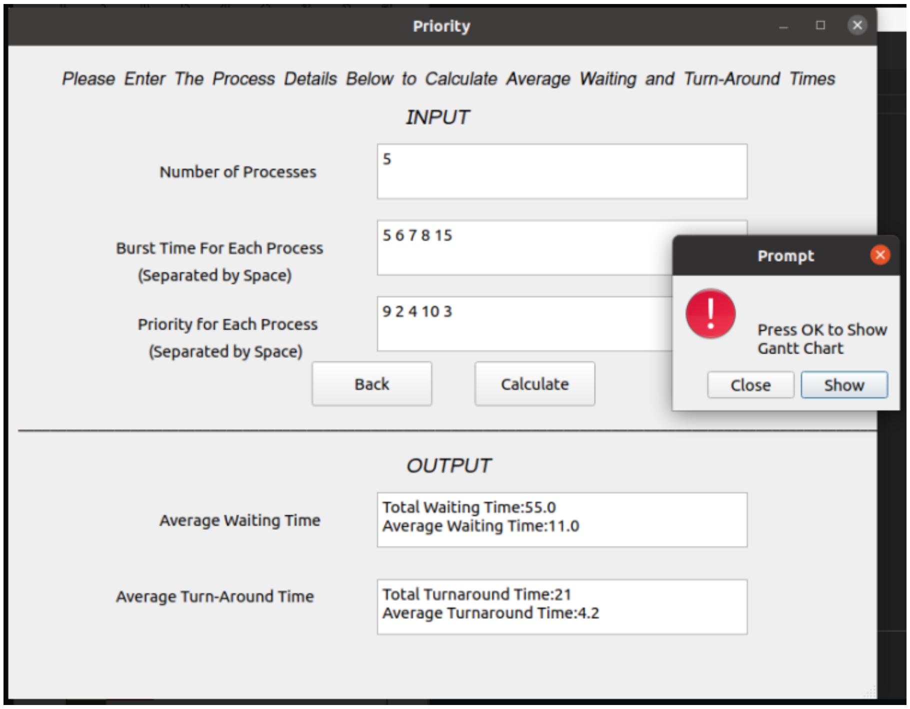
Average Waiting Time:

```
Total Waiting Time: 20.0
Average Waiting Time: 6.666666666666667
```

Average Turn-Around Time:

```
Total Turnaround Time: 44
Average Turnaround Time: 14.666666666666666
```





Please Enter The Process Details Below to Calculate Average Waiting and Turn-Around Times

INPUT

Number of Processes: 5

Burst Time For Each Process (Separated by Space): 5 6 7 8 15

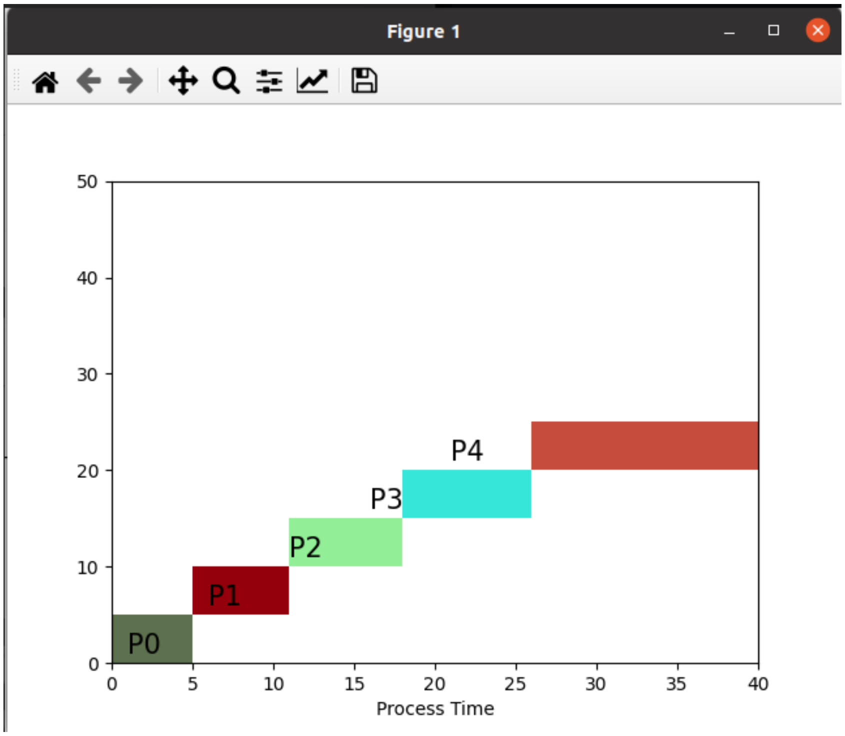
Priority for Each Process (Separated by Space): 9 2 4 10 3

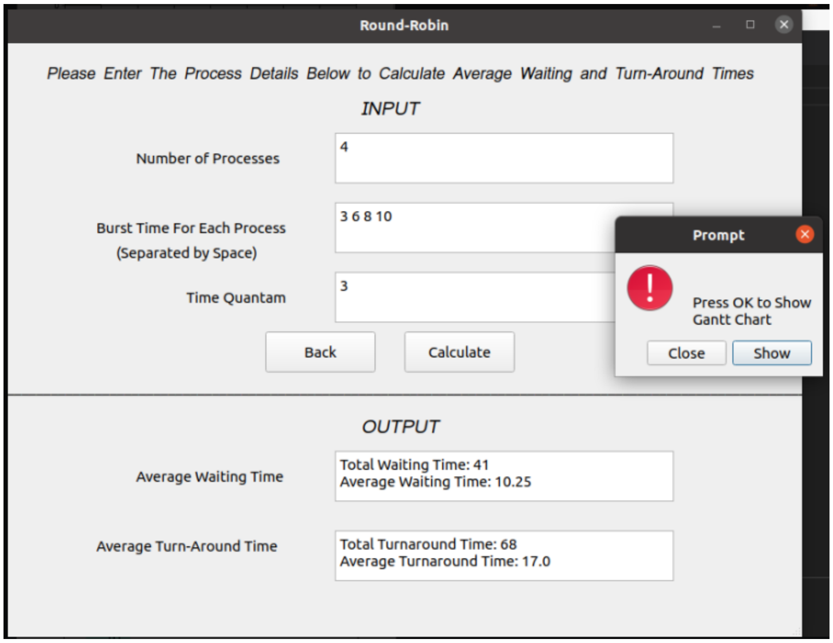
Press OK to Show Gantt Chart

OUTPUT

Average Waiting Time:
Total Waiting Time:55.0
Average Waiting Time:11.0

Average Turn-Around Time:
Total Turnaround Time:21
Average Turnaround Time:4.2





Round-Robin

Please Enter The Process Details Below to Calculate Average Waiting and Turn-Around Times

INPUT

Number of Processes: 4

Burst Time For Each Process (Separated by Space): 3 6 8 10

Time Quantam: 3

Prompt

Press OK to Show Gantt Chart

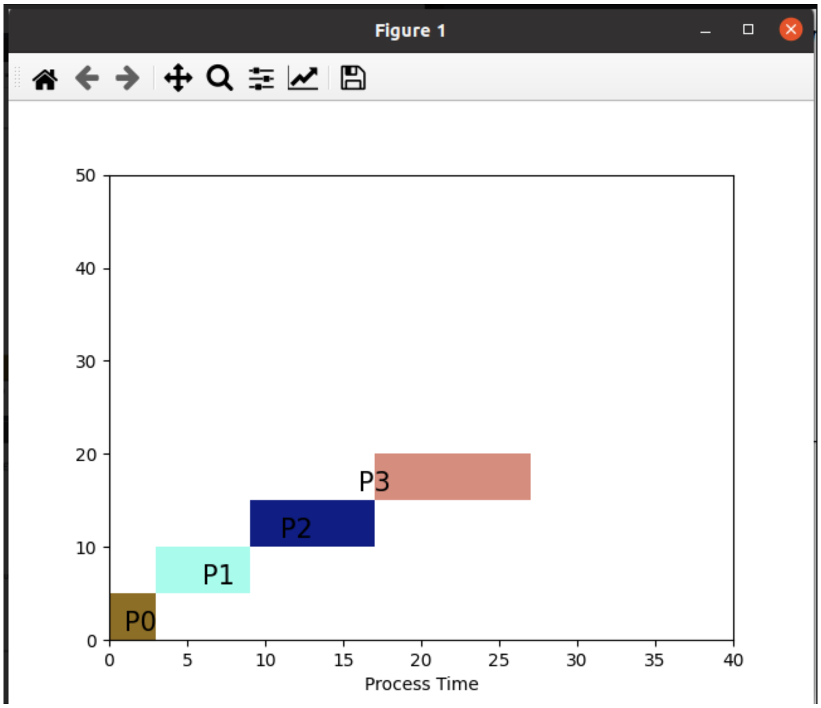
OUTPUT

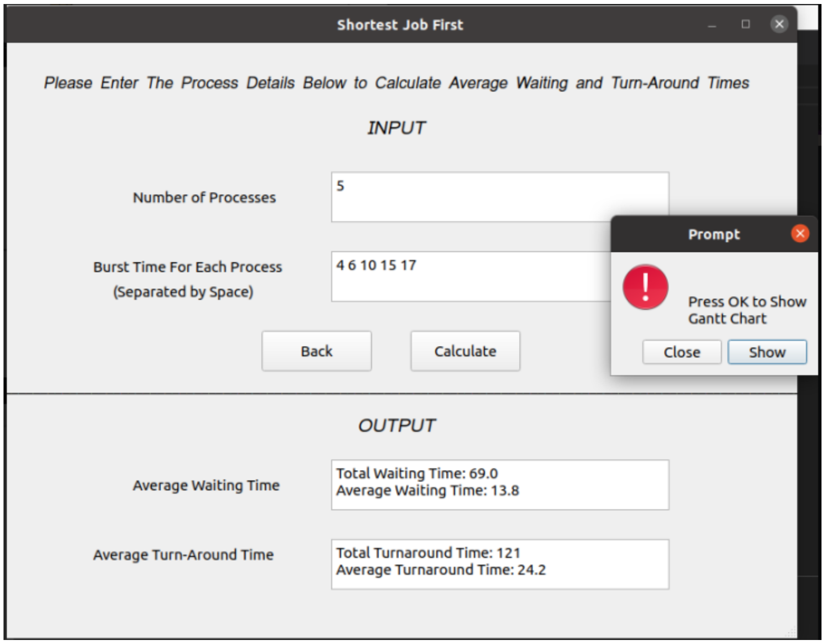
Average Waiting Time:

```
Total Waiting Time: 41
Average Waiting Time: 10.25
```

Average Turn-Around Time:

```
Total Turnaround Time: 68
Average Turnaround Time: 17.0
```





Please Enter The Process Details Below to Calculate Average Waiting and Turn-Around Times

INPUT

Number of Processes: 5

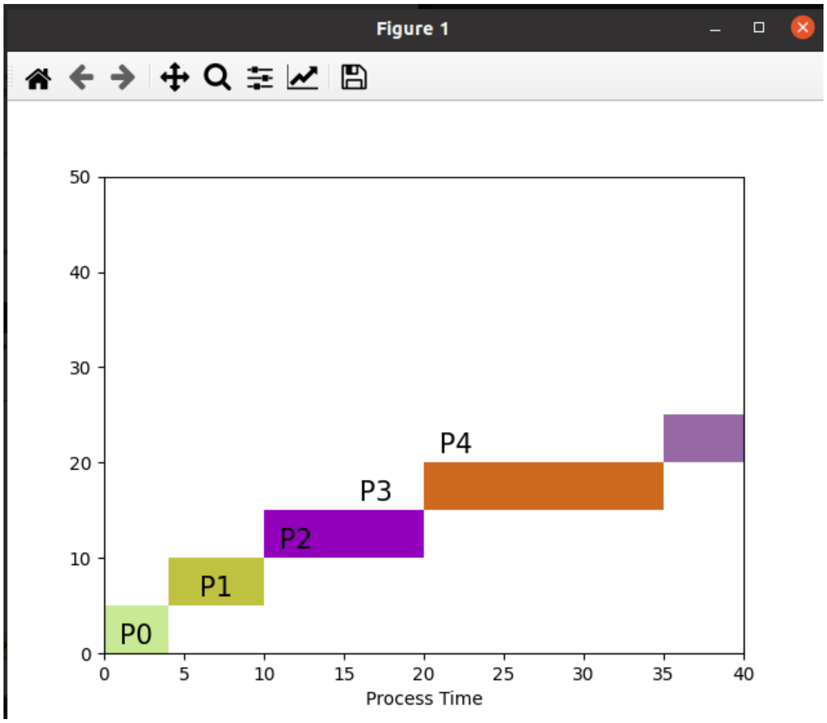
Burst Time For Each Process (Separated by Space): 4 6 10 15 17

Press OK to Show Gantt Chart

OUTPUT

Average Waiting Time:
Total Waiting Time: 69.0
Average Waiting Time: 13.8

Average Turn-Around Time:
Total Turnaround Time: 121
Average Turnaround Time: 24.2



Description of Services

Following is an overview of the 4 types of Scheduling Algorithms we have implemented in this Project:

First-Come First-Serve (FCFS): In First-Come First-Serve (FCFS) algorithm jobs are executed on first come, first serve basis. It is a non-preemptive, pre-emptive scheduling algorithm. Easy to understand and implement. Easy to understand and implement. Its implementation is based on FIFO queue. Poor in performance as
average wait time is high.

Shortest-Job-First (SJF): In Shortest-Job-First (SJF) jobs algorithm Jobs are executed on shortest Job First basis. This is a non-preemptive, pre-emptive scheduling algorithm. It is best approach to minimize waiting time. It is easy to implement in Batch systems where required CPU time is known in advance. Impossible to implement in interactive systems where required CPU time is not known. The processor should know in advance how much time process will take.

Priority-Based Scheduling: Priority based scheduling is a non-preemptive algorithm and one of the most common scheduling algorithms in batch systems. Each process is assigned a priority. Process with highest priority is to be executed first and so on. Processes with same priority are executed on first come first served basis. Priority can be decided based on memory requirements, time requirements or any other resource requirement.

Round Robin Scheduling: Round Robin is the preemptive process scheduling algorithm. Each process is provided a fix time to execute, it is called a quantum. Once a process is executed for a given time period, it is pre-empted and other process executes for a given time period. Context switching is used to save states of preempted processes.

Conclusion

Using this scheduling algorithm program gives us an idea of which algorithm would be best suited for our particular problem. By running this program, we can enter our data values and see which algorithm produces the best turn-around time and has the least waiting time, and then select the best algorithm. This program can also be used to demonstrate students. We can run each algorithm and show them the results along with the Gantt Charts.

References

PyQt5 Documentation:

https://doc.qt.io/qtforpython/

Other References:

https://www.javatpoint.com/os-fcfs-scheduling

https://www.javatpoint.com/os-sjf-scheduling

https://www.javatpoint.com/os-preemptive-priority-scheduling

https://www.studytonight.com/operating-system/round-robin-scheduling