Operating Systems

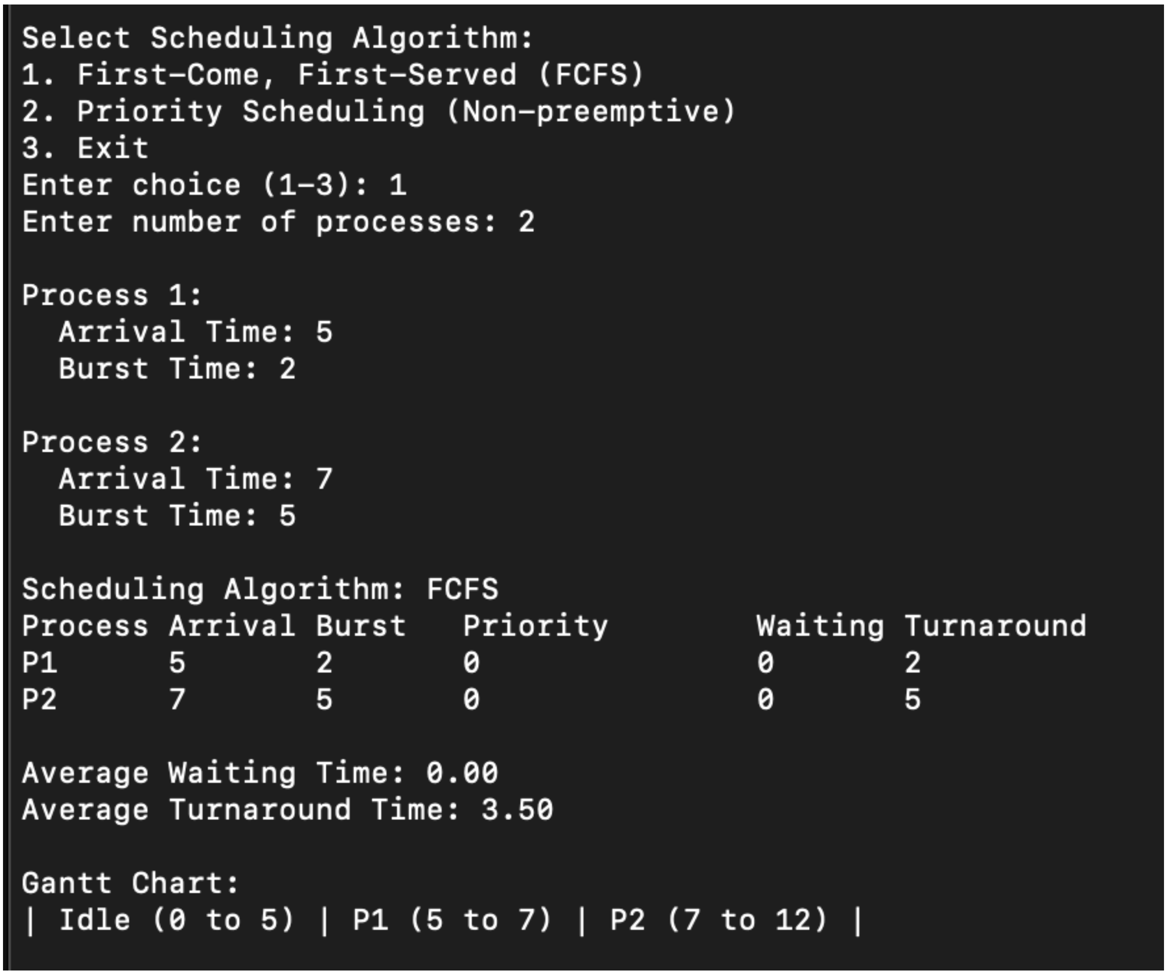
Anna Testova

GH1025428

The link to GitHub:https://github.com/AnnaTestova/annatestova.github.io.git

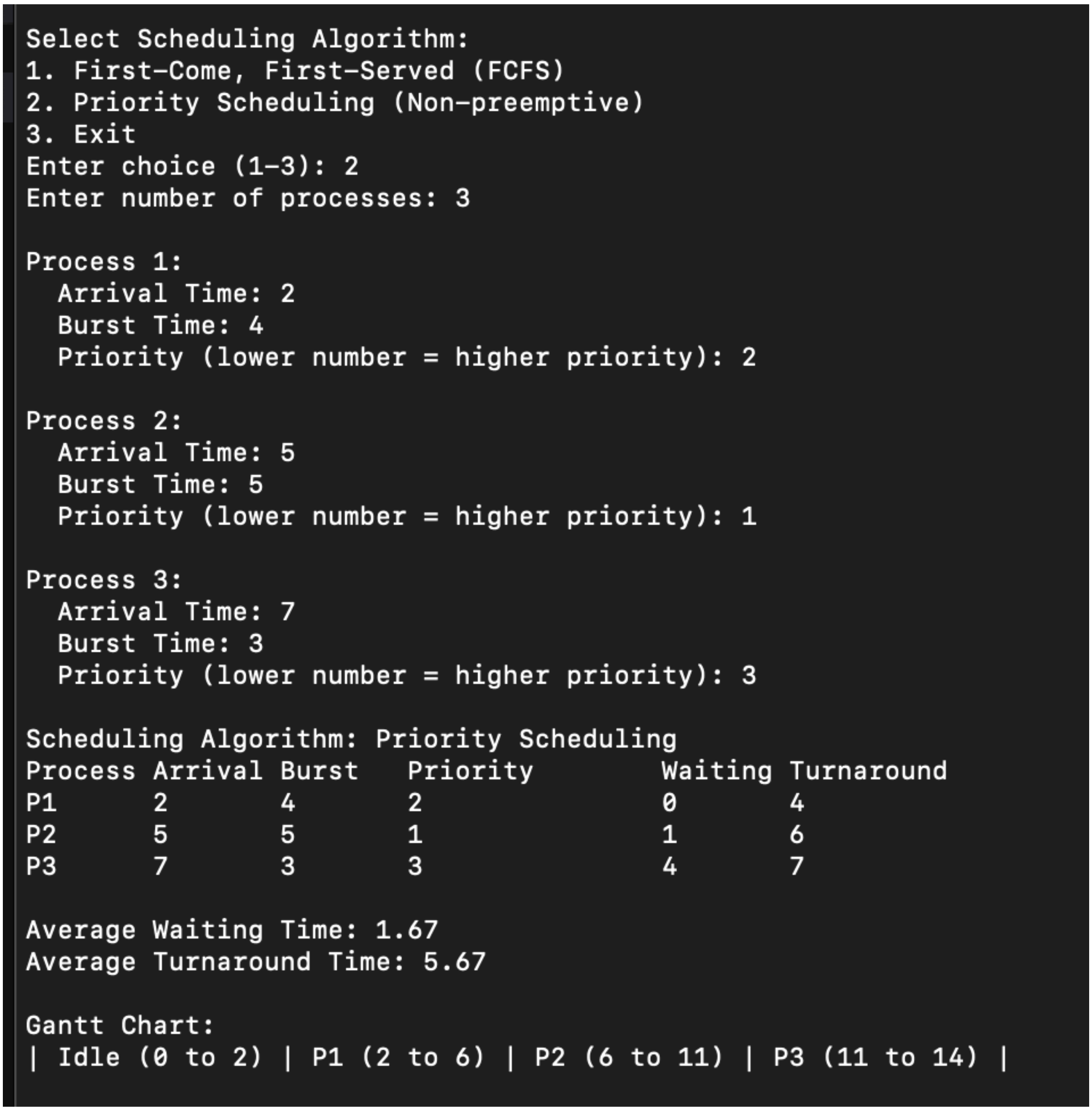
Screenshots of my actual Programm(both scheduling algorithms)

1. First-Come, First-Serve(FCFS):



```
Select Scheduling Algorithm:
1. First-Come, First-Served (FCFS)
2. Priority Scheduling (Non-preemptive)
3. Exit
Enter choice (1-3): 1
Enter number of processes: 2

Process 1:
  Arrival Time: 5
  Burst Time: 2

Process 2:
  Arrival Time: 7
  Burst Time: 5

Scheduling Algorithm: FCFS
Process Arrival Burst   Priority        Waiting Turnaround
P1      5       2       0               0       2
P2      7       5       0               0       5

Average Waiting Time: 0.00
Average Turnaround Time: 3.50

Gantt Chart:
| Idle (0 to 5) | P1 (5 to 7) | P2 (7 to 12) |
```

2. Priority Scheduling:



```
Select Scheduling Algorithm:
1. First-Come, First-Served (FCFS)
2. Priority Scheduling (Non-preemptive)
3. Exit
Enter choice (1-3): 2
Enter number of processes: 3

Process 1:
  Arrival Time: 2
  Burst Time: 4
  Priority (lower number = higher priority): 2

Process 2:
  Arrival Time: 5
  Burst Time: 5
  Priority (lower number = higher priority): 1

Process 3:
  Arrival Time: 7
  Burst Time: 3
  Priority (lower number = higher priority): 3

Scheduling Algorithm: Priority Scheduling
Process Arrival Burst   Priority        Waiting Turnaround
P1      2       4       2               0       4
P2      5       5       1               1       6
P3      7       3       3               4       7

Average Waiting Time: 1.67
Average Turnaround Time: 5.67

Gantt Chart:
| Idle (0 to 2) | P1 (2 to 6) | P2 (6 to 11) | P3 (11 to 14) |
```

Report

Introduction:

This project focuses on simulating CPU scheduling. The scheduler manages the execution order of processes, impacting system efficiency and responsiveness. I’ve chose

2 algorithms: FCFS and non-preemptive Priority Scheduling, to demonstrate different scheduling strategies.

System Design:

Each process contains information such as: it’s ID, arrival time, burst time and priority. The program run on the terminal, where users can easily enter the process details and choose a scheduling algorithm. Then the program calculates the results and shows a timeline in a Giantt chart, highlighting process execution and idle time.

Implementation:

Processes are stored in list and sorted according Schosen algorithm. The programm calculates how long each process waited and it’s turnaround times and generates Gantt chart to illustrate scheduling.

Changes and Sollutions:

One of the main challenges was handling idle CPU time required careful simulation of time progression, when no processes were ready. What is more, implementing Priority Scheduling was another challenge, which was solved using sorting and maintaining process completion flags.

Results:

The program shows the average waiting time and turanaout time for each set of processes and clearly displays the CPU scheduling.

Conclusion and Future Work:

The project simulates basic CPU scheduling and highlights scheduling impacts on process execution. Few of the future improvements might be implementing preemptive algorithms and memory management.