Carlos Romeo Clemente Del Castillo III - *CC22-1H_A*

Dale Cofulan Kerchaten - *CC22-1H_A*

Introduction to Platform Technologies • *Final Project: Analysis*

32PT-FP-AN • 12 Aug. 2025 **Tue**

1 Analysis

1. **Why did you choose the specific algorithms for each queue?**
   - We believe that combining the two scheduling algorithms (SJF and NPPS) would allow us to strike a balance between urgency and efficiency.

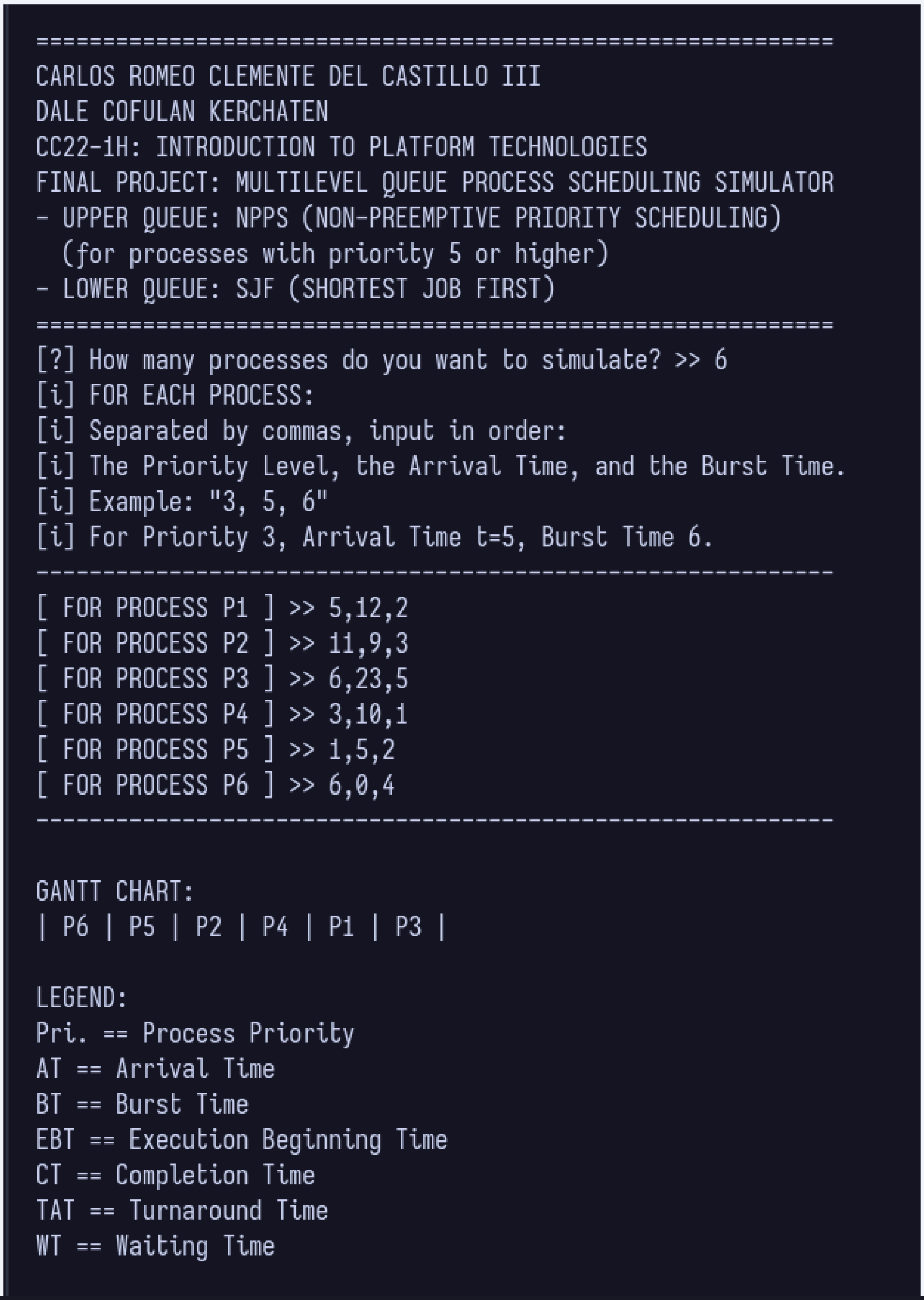
2. **What are the advantages and disadvantages you observed? (can be based on TAT and WT)**
   - Using NPPS and SJF made important tasks done first and short once finished quickly for a balanced workflow. There aren't much disadvantages to mention since there's barely any unfairness and starvation taking place.

3. **Did starvation or unfairness occur in any of the queues? If yes, explain why.**
   - Since our system only executes tasks one at a time, there is little unfairness that occurs, except in cases where a process is particularly lengthy. Starvation will only happen if a process with a long burst time is present.

4. **What challenges did you encounter while simulating multilevel scheduling?**
   - We ran into numerous problems whilst writing code to simulate multilevel scheduling. Most notably, while writing code to simulate the ticking of CPU time, the ticker went into integer overflow multiple times for reasons yet to be concluded, which was only fixed by entirely rewriting the code for simulating process execution.

5. **How does multilevel scheduling affect overall performance compared to using a single algorithm?**
   - A multilevel schedule makes mixed workloads faster by giving urgent tasks priority and finishing small tasks quickly.

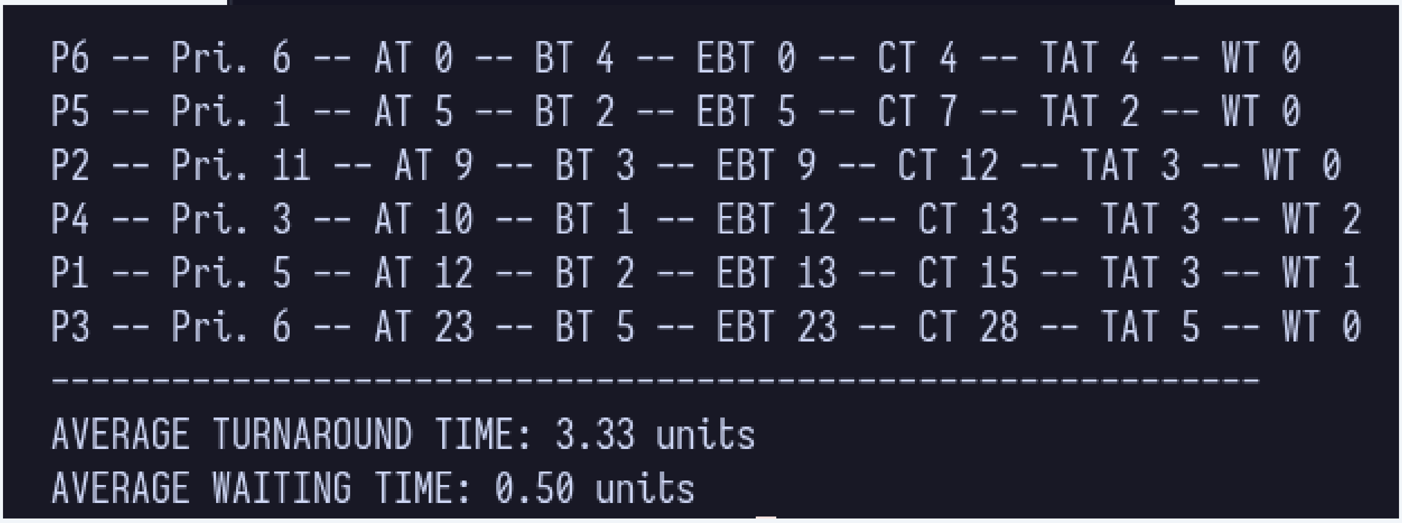
2 Screenshots



```
===============================================================
CARLOS ROMEO CLEMENTE DEL CASTILLO III
DALE COFULAN KERCHATEN
CC22-1H: INTRODUCTION TO PLATFORM TECHNOLOGIES
FINAL PROJECT: MULTILEVEL QUEUE PROCESS SCHEDULING SIMULATOR
- UPPER QUEUE: NPPS (NON-PREEMPTIVE PRIORITY SCHEDULING)
  (for processes with priority 5 or higher)
- LOWER QUEUE: SJF (SHORTEST JOB FIRST)
===============================================================
[?] How many processes do you want to simulate? >> 6
[i] FOR EACH PROCESS:
[i] Separated by commas, input in order:
[i] The Priority Level, the Arrival Time, and the Burst Time.
[i] Example: "3, 5, 6"
[i] For Priority 3, Arrival Time t=5, Burst Time 6.
---------------------------------------------------------------
[ FOR PROCESS P1 ] >> 5,12,2
[ FOR PROCESS P2 ] >> 11,9,3
[ FOR PROCESS P3 ] >> 6,23,5
[ FOR PROCESS P4 ] >> 3,10,1
[ FOR PROCESS P5 ] >> 1,5,2
[ FOR PROCESS P6 ] >> 6,0,4
---------------------------------------------------------------


GANTT CHART:
| P6 | P5 | P2 | P4 | P1 | P3 |

LEGEND:
Pri. == Process Priority
AT == Arrival Time
BT == Burst Time
EBT == Execution Beginning Time
CT == Completion Time
TAT == Turnaround Time
WT == Waiting Time
```



```
P6 -- Pri. 6 -- AT 0 -- BT 4 -- EBT 0 -- CT 4 -- TAT 4 -- WT 0
P5 -- Pri. 1 -- AT 5 -- BT 2 -- EBT 5 -- CT 7 -- TAT 2 -- WT 0
P2 -- Pri. 11 -- AT 9 -- BT 3 -- EBT 9 -- CT 12 -- TAT 3 -- WT 0
P4 -- Pri. 3 -- AT 10 -- BT 1 -- EBT 12 -- CT 13 -- TAT 3 -- WT 2
P1 -- Pri. 5 -- AT 12 -- BT 2 -- EBT 13 -- CT 15 -- TAT 3 -- WT 1
P3 -- Pri. 6 -- AT 23 -- BT 5 -- EBT 23 -- CT 28 -- TAT 5 -- WT 0
------------------------------------------------------------
AVERAGE TURNAROUND TIME: 3.33 units
AVERAGE WAITING TIME: 0.50 units
```