

Class: Fall 2022 - CS 4348.502
Student Name: Ahmet Mert Buyulu
Net ID: AMB190003
Date: 12/09/22

Project 3 - Write-up

1) **Question:** How did you approach the project?

Answer:

Since the purpose of this project is to demonstrate the understanding of how different scheduling algorithm from chapter 9 of our textbook operates on a uniprocessor system, I first read the section on the scheduling algorithms used in the project from the textbook, understand the fundamental difference between preemptive and non-preemptive scheduling algorithms, and how each of these algorithms approach process selection from the set of ready processes differently.

Having said that, I could not work on this project until the week it is due to being busy with other classes and exams. Thus, I knew I had to make a good plan to be able to finish the project on time. Luckily, gaining enough knowledge about the algorithms boosted the coding process by a lot since I did not need to research much; rather, I was able to just focus on the requirements of the project.

2) **Question:** How did you organize the project? Why?

Answer:

I organized the project by determining which action I should take in each step through the process of implementing the project to complete it with as minimum confusion and stress as possible. I know this project was a lot more complex than the previous ones. Thus, I first spent few hours on running the sample code, understanding the python script, which the professor has uploaded on e-learning, and thinking of the structure, in which my program should have to be able to demonstrate what is required.

After these steps, I started coding and first refactored the python script and created individual files for each of the following: simulator, scheduler, event, process, process statistics, and algorithms to be used in the project. These helped me a great deal later the project, as it was easier to navigate between files and to debug when needed. It also helped every file to simulate a single entity only.

Once I finished refactoring and completing the overall structure and the workflow of the project, I started implementing individual scheduling algorithms. I tested and checked the correctness of each of the algorithm's output against the sample runs provided by the professor.

Finally, I tested the functionality of overall program, cleared the comments used for debugging (left the other ones that are used for informing the reader), and concluded that the program is ready for submission.

3) **Question:** What problems did you encounter?

Answer:

1 - The output I was getting when running the program using highest-response-ratio-next algorithm did not match the one provided inside the sample-runs folder.

2 – I was having a problem with one process dispatching earlier than it was supposed to in the program while using VRR approach.

4) **Question:** How did you fix them?

Answer:

1 – I made research about it and realized that I made the wrong assumption on what waiting time meant in the selection function of the algorithm. I thought waiting time meant the time spent in the ready queue after getting unblocked until the time it is dispatched. However, I learned that waiting time means the total waiting time of a process, not the latest CPU wait time.

2 – I made researched and learned that when a process is dispatched from the auxiliary queue, it runs no longer than a time equal to the quantum minus the total time spent running since it was last selected from the ready queue. So, the problem was that the scheduler granted more execution time than it was supposed to the processes dispatched from the auxiliary queue. Once I updated that logic, the program worked without any problems and outputted the correct statistics to the console.

5) **Question:** What did you learn doing this project?

Answer:

1 - I learned how to use the concept of inheritance in python

2 - I learned about queue class in python and how to create my own priority queue from scratch using a list.

3 - I improved my understanding of popular process scheduling algorithms, how they differ from one another, and their advantages/disadvantages.

4 - I learned how each of the scheduling algorithm affects the performance of the processor.

6) **Question:** If you did not complete some feature of the project, why not?

Answer:

I implemented every scheduling algorithm required by the project except the shortest-remaining-time (SRT). The reason I could not finish it is mainly due to not having enough time. Regardless, I did complete all the other requirements aside from the one I mentioned above including input validation and handling program errors.

if I had more time, I would try to figure out when SRT algorithm preempts the currently running process and where to do this logic in my program. Luckily, I did not encounter any bugs during the partial implementation of this algorithm. Thus, I believe if I had a couple more hours to work on it, I would finish up implementing this feature as well.