FCFS CPU scheduler Code and Report

Project objective: To learn more about OS CPU scheduling through a hands-on simulation programming experience. To simulate, compare, and evaluate CPU scheduling algorithms using a consistent set of data.

Directions

The following information should be included in your document, **numbered and in order:**

Names:
Date:
Language used:

1. Provide clear instructions on how to compile, build, and run the simulator (this will indicate that the application has been tested and works on the engineering student desktop)

2. Introduction

3. **Insert one table** (See the ASSIGNMENT EXAMPLE) that includes the entire simulation results for CPU Utilization, Response Time (RT) , Waiting Time (WT), Turnaround Time (TT) **PER PROCESS and Averages f**or both FCFS(710 time units) and SJF (810 time units)

4. Answer the following questions in full sentences with a brief explanation, IN YOUR OWN WORDS:
    a. Which algorithm (FCFS or SJF) has the best (highest) CPU utilization, why do you think that algorithm has a higher CPU utilization?
    b. How many context switches are in the simulation of FCFS ?
    c. How many context switches are in the simulation of SJF ?
    d. How does the number of context switches effect the performance of the algorithm?
    e. Which algorithm (FCFS or SJF) has the lowest average waiting time?
    f. Which algorithm (FCFS or SJF) has the lowest average response time ?
    g. Which algorithm (FCFS or SJF) has the lowest average turnaround time?

5. Insert a partial or complete Gantt chart for FCFS. (NOTE: IF YOU DID NOT GET YOUR CODE TO WORK CORRECTLY, OR YOUR SIMULATOR DOES NOT PRODUCE THE CORRECT OUTPUT YOU MUST COMPLETE AND SUBMIT A FULL GANTT CHART IN THIS REPORT – DO NOT INSERT MY GANTT CHART, CREATE YOUR OWN)

6. Insert the calculated results that were produced by the simulation
7. Insert the FCFS Program Output.

Requirements

- There will be only 2 files submitted for this assignment
    1. Document must be submitted in a PDF format
        - The document should be numbered and in order, see FCFS ASSIGNMENT EXAMPLE
        - Save the file as **LastName_FCFSreport.pdf**
        - (Use the same last name as the Implementation plan assignment)
    2. The source code file for FCFS
        - The source code MUST be submitted as a plain text file
        - The source code MUST all be included in only one file
        - Save the file as **LastName_FCFS.txt**
        - (Use the same last name as the Implementation plan assignment)
- One assignment per student

- DO NOT COMPRESS (ZIP) the files
- Assignments with zipped files will not be graded
- All report information, code, images tables, and instructions must be in ONE PDF file
- Your simulator should NOT read data from an input file, all input data needs to be included in the program
- Only one PDF file will be graded **LastName_FCFSreport.pdf**
- Only one plain text source code file should be submitted **LastName_FCFS.txt**

NOTE: The code will be tested using the cloud on the Engineering student desktop.
Instructions must be included for the grader to be able to use the correct tool on the Engineering student desktop to build the application using the plain text source code file submitted and run the simulation.

Remote Connection - https://portal.eng.fau.edu/
Description and instructions: http://tsg.eng.fau.edu/

Algorithms and data :

1. **FCFS non preemptive**
2. **MLFQ**

   Multilevel Feedback Queue (preemptive – absolute priority in higher queues)
   Queue 1 uses RR scheduling with Tq = 6
   Queue 2 uses RR scheduling with Tq = 12
   Queue 3 uses FCFS

   All processes enter first queue 1. If time quantum (Tq) expires before CPU burst is complete, the process is downgraded to next lower priority queue. Processes are not downgraded when preempted by a higher queue level process. Once a process has been downgraded, it will not be upgraded.

Assumptions:
1. All processes are activated at time 0
2. Assume that no process waits on I/O devices.
3. After completing an I/O event, a process is transferred to the ready queue.
4. Waiting time is accumulated while a process waits in the ready queue.
5. Turnaround time is a total of (Waiting time) + (CPU burst time) + (I/O time)
6. Response time is the first measure of waiting time from arrival at time 0 until the first time on the CPU

Process Data:
process goes {CPU burst, I/O time, CPU burst, I/O time, …….., last CPU burst}

```
P1 { 6, 21, 9, 28, 5, 26, 4, 22, 3, 41, 6, 45, 4, 27, 8 , 27, 3 }
P2 { 19, 48, 16, 32, 17, 29, 6, 44, 8, 34, 21, 34, 19, 39, 10, 31, 7 }
P3 { 12, 14, 6, 21, 3, 29, 7, 45, 8, 54, 11, 44, 9 }
P4 { 11, 45, 5, 41, 6, 45, 8, 51, 4, 61, 13, 54, 11, 61, 10 }
P5 { 16, 22, 15, 21, 12, 31, 14, 26, 13, 31, 16, 18, 12, 21, 10, 33, 11 }
P6 { 20, 31, 22, 30, 25, 29, 11, 44, 17, 34, 18, 31, 6, 22, 16 }
P7 { 3, 44, 7, 24, 6, 34, 5, 54, 4, 24, 7, 44, 6, 54, 5, 21, 6, 43, 4 }
P8 { 15, 50, 4, 23, 11, 31, 4, 31, 3, 47, 5, 21, 8, 31, 6, 44, 9}
```

Sharing code includes posting completed work (code) before the assignment official deadline onto sites such as GitHub, emailing code to other students, allowing any access to your work before the official deadline has passed. Other code sharing offenses include submitting another person's work as your own, this includes taking code off sites such as GitHub, Chegg, etc.

Modifying code and submitting it as your own is a fraudulent practice—specifically, plagiarism—and is no different than copying paragraphs of information from a book or journal article and calling it your own

(make sure that you work independently within your group and submit only your own code)

Students at Florida Atlantic University are expected to maintain the highest ethical standards. Academic dishonesty is considered a serious breach of these ethical standards, because it interferes with the University mission to provide a high quality education in which no student enjoys an unfair advantage over any other. Academic dishonesty is also destructive of the university community, which is grounded in a system of mutual trust and places high value on personal integrity and individual responsibility. Harsh penalties are associated with academic dishonesty. For more information, see https://www.fau.edu/ctl/AcademicIntegrity.php