Assignment #5
Scheduling
130 Points

Problem

In this assignment, you will write a C/C++ program that accepts process data and simulates the scheduling of those processes using various scheduling algorithms. You will be implementing the following scheduling algorithms: First Come First Serve (FCFS), Round Robin (RR) with a quantum of 10ms and 40ms, Shortest Process Next (SPN), Shortest Remaining Time (SRT), Highest Response Ratio Next (HRRN), and Feedback with a quantum of 10ms.

Your solution should accept the input process data file as a required command line argument and display an appropriate error message if the argument is not provided or the file does not exist. The command to run your application will look something like this:

Form: assignment_5 <path_to_source_file>
Example: assignment_5 process_data.txt

Output Rules

Each line of the output file will contain the identifier for the process that would be running in each 10ms increment. Your solution will need to produce a single output file for each scheduling policy being implemented. The name of each output file and the scheduling policy being utilized for that output file can be found below:

Output File Name	*Policy*
fcfs.out	First Come First Serve
rr_10.out	Round Robin (quantum = 10 ms)
rr_40.out	Round Robin (quantum = 40 ms)
spn.out	Shortest Process Next
srt.out	Shortest Remaining Time
hrrn.out	Highest Response Ratio Next
feedback.out	Feedback (quantum = 10 ms)

Input / Output File Specifications

Input File Specification

The input file will adhere to the following specifications:

1. The file will be a plain text file.
2. Each line of the input file will contain the following three fields separated by tabs:
    a. Process identifier (an alphanumeric string)
    b. Arrival time (an integer stating the process's arrival time in ms)
    c. Service time (a non-zero, positive integer stating the process's service time in ms)

Example File:

```
P1      0       20
P2      10      50
P3      30      10
P4      50      30
```

Output File Specification

Your output files must adhere to the following specifications:

1. The output file for each scheduling policy should be a plain text file.
2. Each line of the output file represents a 10ms window in time.
    a. i.e. The first line would represent the first 10ms of execution, the second line would represent the next 10ms of execution, and so forth.
3. The process identifier that is running during each 10ms window is indicated on the corresponding line(s) in the output file.
4. If a process has a service time longer than 10ms, its identifier will be printed on multiple lines to indicate which process is running in each 10ms window.

Example File:

fcfs.out

```
P1
P1
P2
P2
P2
P2
P2
P3
P4
P4
P4
```

Additional Solution Rules and Hints

Your solution should abide by the following rules:

- Your solution will be tested in the HPCC using a bash script to compile your code, run your solution against a test case, and then assign a correctness score to your solution.
- Your solution must be able to compile using GNU v5.4.0.
- The grading script will request the following resources for your job, these should be considered your maximum limits when determining how to write your solution:
    - 1-hour maximum runtime
    - 2 cores
    - 64,440 MB of memory

Helpful Links:

- Command Line Arguments in C/C++
- Makefile Tutorial
- HPCC User Guides

What to turn in to BlackBoard

A zip archive (.zip) named <FirstName>_<LastName>_R<#>_Assignment5.zip that contains the following files:

- Your C / C++ source code
    - You can select any names you want, your makefile will dictate the executable name.
- `makefile`
    - The makefile required to compile your application – ensure your final executable is called *assignment_5.*