Castillo, Javillo CSCI 51.02/CS 162A B-Q4

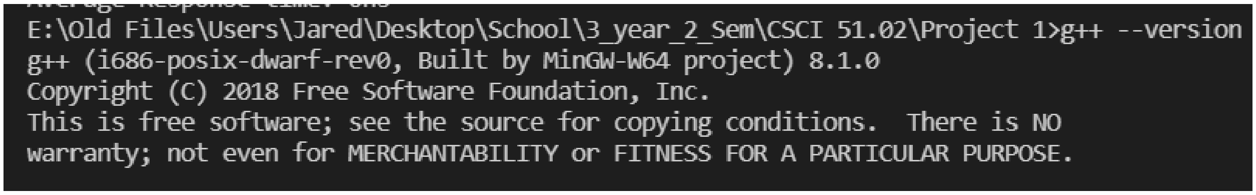
README/Manual

Technical Assumptions:

1. Users will not have errors in input. For example, there will be no errant spaces, tabs or characters in input.
2. Users will use a text file for input or will type inputs manually.
3. A while loop is used to simulate what a process scheduler and the CPU do for 1 tick or 1 ns.

Installation of c++:

1. Go to https://sourceforge.net/projects/mingw-w64/files/
2. Download and Install MinGW-W64-install.exe
3. In cmd, type g++ --version
4. Output should be:



```
Average Response time: 0ns
E:\Old Files\Users\Jared\Desktop\School\3_year_2_Sem\CSCI 51.02\Project 1>g++ --version
g++ (i686-posix-dwarf-rev0, Built by MinGW-W64 project) 8.1.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

How to run the program:

1. To run the program, you must first compile the main.cpp file by typing "g++ -o run main.cpp". You can do this by opening a command prompt or a terminal (if you are using a mac) from the folder the main file is located in.
2. After typing the command in step 1, you will have an executable file called run.exe and using this you will run the code. Type ".\run" or "run" to run the code. If you desire to run the code with premade input from a text file type "./run < _____.txt" or "run < _____.txt". (You can test the code with GP1Sample1In.txt or GP1Sample2In.txt)
3. When running the code with a text file, it will automatically print out the output for the test cases in the given file. If you will not run the code with a text file, you will need to manually input the test cases.
4. (Manually) The first line of input should be an integer that will determine the number of test cases you want to perform. Click enter after inputting the integer.
5. (Manually) The second line of input will ask you to input the first test case. Type the number of processes (N) you want in the test case. Then, separated by a space, type your desired process scheduling algorithm. It can either be:

a. FCFS (First Come First Serve)
b. SJF (Shortest Job First)
c. SRTF (Shortest Remaining Time First)
d. P (Priority)
e. RR (Round Robin)

For RR you will need to input an extra input Q which will represent the time quantum. Then click enter.

6. (Manually) At this point you will be inputting the processes for your test case. The format for the processes are as follows: A B P. The first number A represents the arrival time, B represents the burst time, and P represents the priority value. Then click enter. You will be doing this for N times (N being the number of processes you specified in step 5).
7. (Manually) Once you have finished inputting the processes you will automatically be shown the output for each test case. The first line of each test case output will show what test case number it is and its process scheduling algorithm. The lines that follow will show the processes running, either if they finished or were preempted, and the details of the CPU running the processes, that being the total time that has passed, the utilization of the CPU in percentage, the total CPU burst time, the Throughput, the waiting, turnaround and response times of each process, and the averages of all those times.