

Programming Assignment 3 Report

Submitted Files:

- **rr.cpp**: a Client file which implements a round robin scheduling algorithm and outputs the wait time, avg wait time, turn around time, and avg turn around time based on input & time quantum provided
- **fcfs.cpp**: a Client file which implements a first come first serve scheduling algorithm and outputs the wait time, avg wait time, turn around time, and avg turn around time based on an inputted text file.
- **priority.cpp**: a Client file which runs a priority based scheduling and outputs the wait time, avg wait time, turn around time, and avg turn around time based on an inputted text file.
- **sjf.cpp**: a Client file which implements a shortest job first scheduling algorithm and outputs the wait time, avg wait time, turn around time, and avg turn around time based on inputted text file.
- **priority_rr.cpp**: a Client file which sorts a text file based on priority and then runs a round robin scheduling algorithm on the sorted queue. It outputs the wait time, avg wait time, turn around time, and avg turn around time based on the time quantum provided.

How to Compile and Run the Program:

To compile the program, use the command: “make” followed by the name of the scheduling algorithm you wish to run.

For example:

make rr

make sjf

make priority_rr

To run the program, use the command: “./” followed by the name of the scheduling algorithm, followed by the input file. If the algorithm is a round robin, include the time quantum.

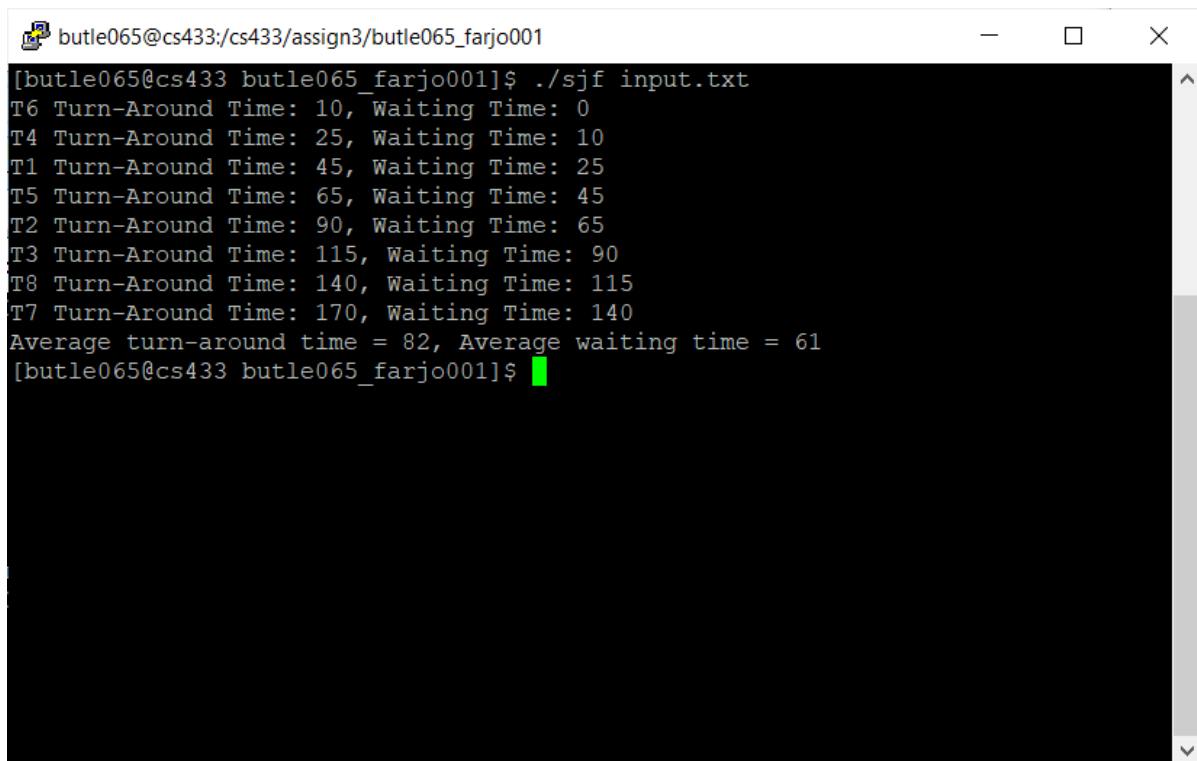
For example:

`./rr input.txt 10` → runs round robin on input.txt with a time quantum of 10
`./sjf input.txt`
`./priority input.txt`

Tests of each scheduling algorithm:

Our groups C++ program accurately runs each of the following scheduling algorithms. All the functions work to the extent required of the assignment. Below I have included screenshots of each scheduling algorithm on the same input file.

SJF:

A screenshot of a terminal window with a black background and white text. The window title is 'butle065@cs433:/cs433/assign3/butle065_farjo001'. The prompt is '[butle065@cs433 butle065_farjo001]\$' followed by the command './sjf input.txt'. The output lists turn-around and waiting times for processes T6, T4, T1, T5, T2, T3, T8, and T7, followed by the average turn-around time (82) and average waiting time (61). The prompt is then shown again with a green cursor.

```
butle065@cs433:/cs433/assign3/butle065_farjo001
[butle065@cs433 butle065_farjo001]$ ./sjf input.txt
T6 Turn-Around Time: 10, Waiting Time: 0
T4 Turn-Around Time: 25, Waiting Time: 10
T1 Turn-Around Time: 45, Waiting Time: 25
T5 Turn-Around Time: 65, Waiting Time: 45
T2 Turn-Around Time: 90, Waiting Time: 65
T3 Turn-Around Time: 115, Waiting Time: 90
T8 Turn-Around Time: 140, Waiting Time: 115
T7 Turn-Around Time: 170, Waiting Time: 140
Average turn-around time = 82, Average waiting time = 61
[butle065@cs433 butle065_farjo001]$
```

FCFS:

```
butle065@cs433:/cs433/assign3/butle065_farjo001
[butle065@cs433 butle065_farjo001]$ ./fcfs input.txt
T1 Turn-Around Time: 20, Waiting Time: 0
T2 Turn-Around Time: 45, Waiting Time: 20
T3 Turn-Around Time: 70, Waiting Time: 45
T4 Turn-Around Time: 85, Waiting Time: 70
T5 Turn-Around Time: 105, Waiting Time: 85
T6 Turn-Around Time: 115, Waiting Time: 105
T7 Turn-Around Time: 145, Waiting Time: 115
T8 Turn-Around Time: 170, Waiting Time: 145
Average turn-around time = 94, Average waiting time = 73
[butle065@cs433 butle065_farjo001]$
```

Priority:

```
butle065@cs433:/cs433/assign3/butle065_farjo001
[butle065@cs433 butle065_farjo001]$ ./priority input.txt
T6 Turn-Around Time: 10, Waiting Time: 0
T2 Turn-Around Time: 35, Waiting Time: 10
T3 Turn-Around Time: 60, Waiting Time: 35
T7 Turn-Around Time: 90, Waiting Time: 60
T1 Turn-Around Time: 110, Waiting Time: 90
T4 Turn-Around Time: 125, Waiting Time: 110
T5 Turn-Around Time: 145, Waiting Time: 125
T8 Turn-Around Time: 170, Waiting Time: 145
Average turn-around time = 93, Average waiting time = 71
[butle065@cs433 butle065_farjo001]$
```

Priority round robin:

```
butle065@cs433:/cs433/assign3/butle065_farjo001
[butle065@cs433 butle065_farjo001]$ ./priority_rr input.txt 10
T6 Turn-Around Time: 10, Waiting Time: 0
T2 Turn-Around Time: 20, Waiting Time: 10
T3 Turn-Around Time: 30, Waiting Time: 20
T7 Turn-Around Time: 40, Waiting Time: 30
T1 Turn-Around Time: 50, Waiting Time: 40
T4 Turn-Around Time: 60, Waiting Time: 50
T5 Turn-Around Time: 70, Waiting Time: 60
T8 Turn-Around Time: 80, Waiting Time: 70
T2 Turn-Around Time: 90, Waiting Time: 80
T3 Turn-Around Time: 100, Waiting Time: 90
T7 Turn-Around Time: 110, Waiting Time: 100
T1 Turn-Around Time: 120, Waiting Time: 110
T4 Turn-Around Time: 125, Waiting Time: 120
T5 Turn-Around Time: 135, Waiting Time: 125
T8 Turn-Around Time: 145, Waiting Time: 135
T2 Turn-Around Time: 150, Waiting Time: 145
T3 Turn-Around Time: 155, Waiting Time: 150
T7 Turn-Around Time: 165, Waiting Time: 155
T8 Turn-Around Time: 170, Waiting Time: 165
Average turn-around time = 228, Average waiting time = 206
[butle065@cs433 butle065_farjo001]$
```

Round Robin:

```
butle065@cs433:/cs433/assign3/butle065_farjo001
[butle065@cs433 butle065_farjo001]$ ./rr input.txt 10
T1 Turn-Around Time: 10, Waiting Time: 0
T2 Turn-Around Time: 20, Waiting Time: 10
T3 Turn-Around Time: 30, Waiting Time: 20
T4 Turn-Around Time: 40, Waiting Time: 30
T5 Turn-Around Time: 50, Waiting Time: 40
T6 Turn-Around Time: 60, Waiting Time: 50
T7 Turn-Around Time: 70, Waiting Time: 60
T8 Turn-Around Time: 80, Waiting Time: 70
T1 Turn-Around Time: 90, Waiting Time: 80
T2 Turn-Around Time: 100, Waiting Time: 90
T3 Turn-Around Time: 110, Waiting Time: 100
T4 Turn-Around Time: 115, Waiting Time: 110
T5 Turn-Around Time: 125, Waiting Time: 115
T7 Turn-Around Time: 135, Waiting Time: 125
T8 Turn-Around Time: 145, Waiting Time: 135
T2 Turn-Around Time: 150, Waiting Time: 145
T3 Turn-Around Time: 155, Waiting Time: 150
T7 Turn-Around Time: 165, Waiting Time: 155
T8 Turn-Around Time: 170, Waiting Time: 165
Average turn-around time = 227, Average waiting time = 206
[butle065@cs433 butle065_farjo001]$
```

Features Implemented (by class):

- All classes have very similar feature sets:
 - Most Notably:
 - Parsing shell input for file name
 - Calculating wait time, avg wait time, turn around time, and avg turn around time.
 - Checking if the given text file contains valid inputs
 - Scheduling Algorithm Specific features:
 - Priority, Priority_RR
 - Included a sort feature to sort a given text file by priority value
 - SJF
 - Sorts a given text file by burst time
 - RR & Priority_RR
 - Parses shell for time quantum, and calculates with a time quantum.

Design and Implementation choices:

For FCFS we implemented this with the standard library queue data structure. We chose to use a queue as it is first-in first-out. For SJF, Priority, and Round Robin Priority we chose to implement them with vectors (all though SJF is passed as a queue after it's sorted). Vectors are an ideal choice because they are easy to write sort functions for, and the standard library provides features to mimic the features of queues which allowed us to re-use some code from FCFS.

Lessons Learned / re-Learned

I tried to implement a comparator and a priority queue initially to sort our processes. However, I had issues with this as I had never used the priority queue data structure in C++. after fiddling around with this for a little I decided that it would be fast and efficient to just write a quick bubble sort, and take in the processes as a vector. Sometimes choosing simpler data structures you are more familiar with can save you a lot of headaches.

References:

I used the website Geeks for Geeks a lot in this project to review queues and vectors. Specifically [this page](#) and [this page](#).