How to Run:

Theo Zinner

[Tvz0001@auburn.edu](mailto:Tvz0001@auburn.edu)

903502834

To compile use the Makefile ‘make’

To run type ‘./aubatch’

For more info on commands type ‘help’ into the command line

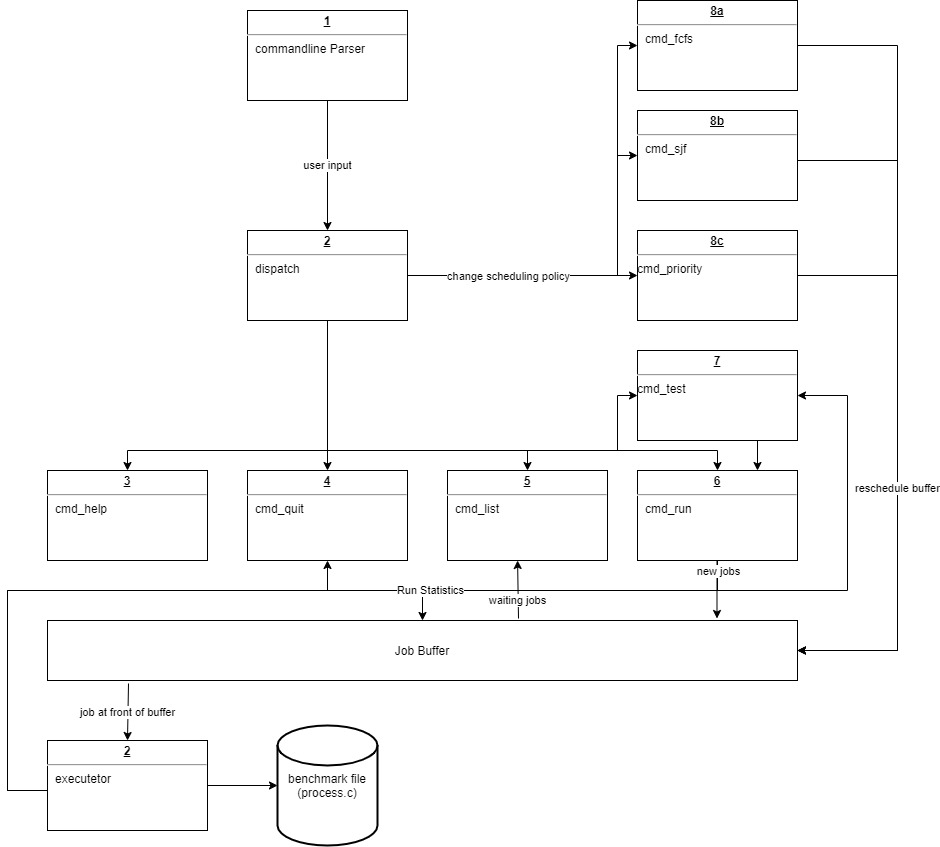
Design and Implementation:

For this project I started with Dr. Qins code and I created the following diagrams to base my project on:

System Diagram:



Data Flow Diagram:



1. Commandline

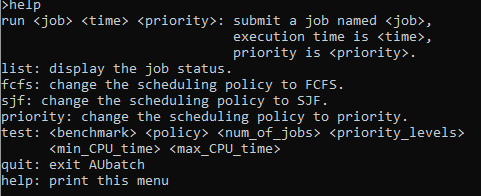
This takes the input from the user and sends it to dispatch.

1. Dispatch

This takes the input from the commandline and references the command table to send the command to the appropriate function.

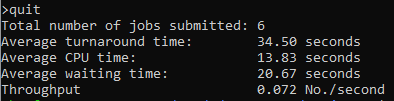
1. Cmd\_help

This command prints out the different input options for the user.



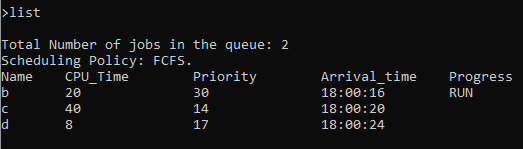
1. Cmd\_quit

This command quits the program and prints out statistics.



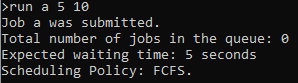
1. Cmd\_list

This command lists the current job being run as well as the buffer.



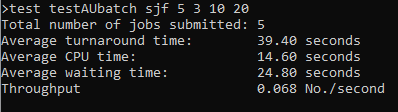
1. Cmd\_run

This command adds a new job to the buffer.



1. Cmd\_test

This command tests out the program by adding multiple jobs with randomized parameters specified by the user.



1. Cmd\_priority

This command changes the scheduling policy to priority.



1. Cmd\_sjf

This command changes the scheduling policy to SJF.



1. Cmd\_fcfs

This command changes the scheduling policy to FCFS.



1. Executor

This executes jobs in the buffer and report statistics that can be read by quit or test.

1. Job buffer

This is the data structure that the jobs are stored in.

1. Benchmark file (process.c)

This is a simple program that is called by execv in the executor.

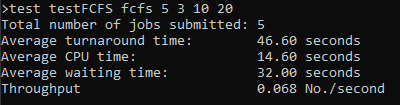
1. Main:

This creates the two threads which are the command line and the executor.

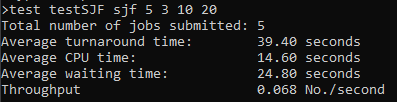


Performance Metrics

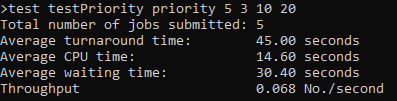
FCFS:



SJF:



Priority:



Performance Evaluation:

FCFS:

This policy performed the worst out of the three but only by a little bit compared to priority.

SJF:

This policy performed the best out of the three with the lowest average waiting time and turnaround time.

Priority:

This policy performed marginally better than FCFS however I believe it could easily be the other way around as priority is arbitrarily assigned to each process.

Lessons Learned

1. Start the project on time. I started mine 2 weeks before the original due date and barely got something working by then. Luckily, the due date was generously extended. If I had started in the last 3 days I would not have gotten it done.
2. I used separate compilation for the first time in this class and learned that it is very easy.
3. I learned that concurrent programing is very difficult.

References:

I started with Dr. Qins reference code from canvas. I also looked up syntax on geeksforgeek.org, cplusplus.com, and tutorialspoint.com.