How to Run:

To compile use the Makefile ‘make’

To run type ‘./aubatch’

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

Theo Zinner

Tvz0001@auburn.edu

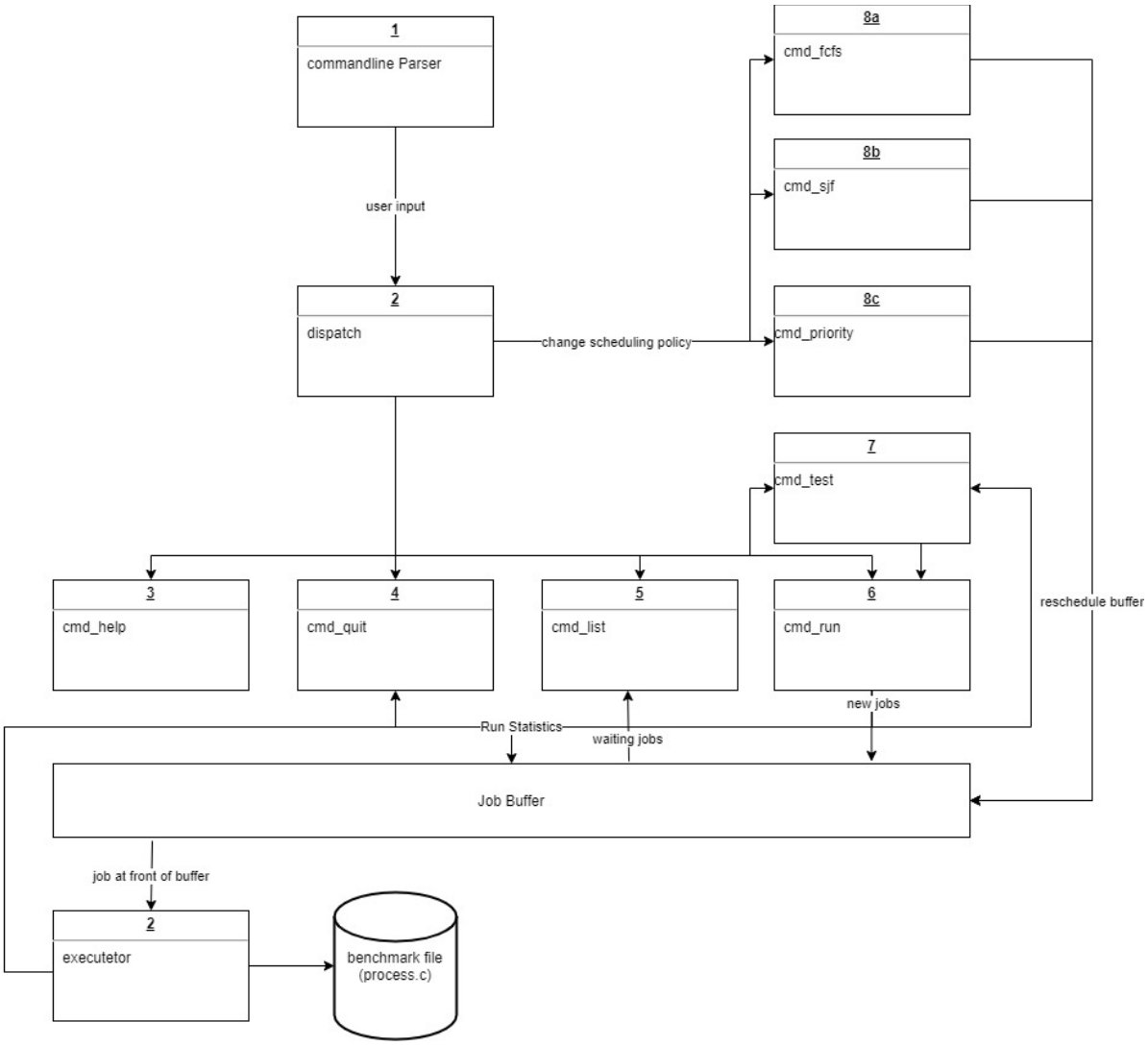
903502834

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.
2) Dispatch
   This takes the input from the commandline and references the command table to send the command to the appropriate function.
3) Cmd_help
   This command prints out the different input options for the user.

```
>help
run <job> <time> <priority>: submit a job named <job>,
                             execution time is <time>,
                             priority is <priority>.
list: display the job status.
fcfs: change the scheduling policy to FCFS.
sjf: change the scheduling policy to SJF.
priority: change the scheduling policy to priority.
test: <benchmark> <policy> <num_of_jobs> <priority_levels>
      <min_CPU_time> <max_CPU_time>
quit: exit AUbatch
help: print this menu
```

4) Cmd_quit

This command quits the program and prints out statistics.

```
>quit
Total number of jobs submitted: 6
Average turnaround time:        34.50 seconds
Average CPU time:               13.83 seconds
Average waiting time:           20.67 seconds
Throughput                      0.072 No./second
```

5) Cmd_list

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

```
>list

Total Number of jobs in the queue: 2
Scheduling Policy: FCFS.
Name    CPU_Time        Priority        Arrival_time    Progress
b       20              30              18:00:16        RUN
c       40              14              18:00:20
d       8               17              18:00:24
```

6) Cmd_run

This command adds a new job to the buffer.

```
>run a 5 10
Job a was submitted.
Total number of jobs in the queue: 0
Expected waiting time: 5 seconds
Scheduling Policy: FCFS.
```

7) Cmd_test

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

```
>test testAUbatch sjf 5 3 10 20
Total number of jobs submitted: 5
Average turnaround time:        39.40 seconds
Average CPU time:               14.60 seconds
Average waiting time:           24.80 seconds
Throughput                      0.068 No./second
```

8) Cmd_priority

This command changes the scheduling policy to priority.

```
>priority
Scheduling policy is switched to Priority. All the 1 waiting jobs
have been rescheduled.
```

9) Cmd_sjf

   This command changes the scheduling policy to SJF.

```
>sjf
Scheduling policy is switched to SJF. All the 1 waiting jobs
have been rescheduled.
```

10) Cmd_fcfs

    This command changes the scheduling policy to FCFS.

```
>fcfs
Scheduling policy is switched to FCFS. All the 0 waiting jobs
have been rescheduled.
```

11) Executor

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

12) Job buffer

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

13) Benchmark file (process.c)

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

14) Main:

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

```
Welcome to Theo's batch job scheduler version 1.0
Type 'help' to find more about AUbatch commands.
```

Performance Metrics

FCFS:

```
>test testFCFS fcfs 5 3 10 20
Total number of jobs submitted: 5
Average turnaround time:        46.60 seconds
Average CPU time:               14.60 seconds
Average waiting time:           32.00 seconds
Throughput                      0.068 No./second
```

SJF:

```
>test testSJF sjf 5 3 10 20
Total number of jobs submitted: 5
Average turnaround time:        39.40 seconds
Average CPU time:               14.60 seconds
Average waiting time:           24.80 seconds
Throughput                      0.068 No./second
```

Priority:

```
>test testPriority priority 5 3 10 20
Total number of jobs submitted: 5
Average turnaround time:        45.00 seconds
Average CPU time:               14.60 seconds
Average waiting time:           30.40 seconds
Throughput                      0.068 No./second
```

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.