

## Project 3 Report – A Batch Scheduling System

by Sajith Muralidhar (szm0227)

Steps to run the program: -

In your centos terminal, type the makefile command(The make will create all executable files) run  
“make -f makerfile.txt”

Execute the program by ./aubatch

The output is as follows : -

### 1) Help:

```
Welcome to Sajith's batch job scheduler version 1.0
Type 'help' to find more about AUBatch commands.
>help
////////////////////////////////////
How to use 'run': run <job> <time> <priority>: submit a job named <job>,
                    execution time is <time>,
                    priority of jobs is <priority>.
What 'list' displays, list: display the job status.
[First Come First Serve] fcfs: change the scheduling policy to FCFS.
[Shortest Job First] sjf: change the scheduling policy to SJF.
[Priority scheduling] priority: change the scheduling policy to priority.
How to use 'test': test <benchmark> <policy> <num_of_jobs> <priority_levels> <min
_CPU_time> <max_CPU_time>
help: Prints the same prompt again
quit: Exit AUBatch Batch Scheduling System
////////////////////////////////////
>
```

### 2) Run: -

```
>run a 9 8
Job a was submitted.
Total number of jobs in the queue: 1
Expected waiting time: 9 seconds
Scheduling Policy: FCFS.
>
```

### 3) Sjf: -

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

### 4) Priority: -

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

5) Fcfs: -

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

Note: - if the scheduling policy is set and if we type the same command again, the prompt returns nothing. It only returns if the policy is changed.

6) List: -

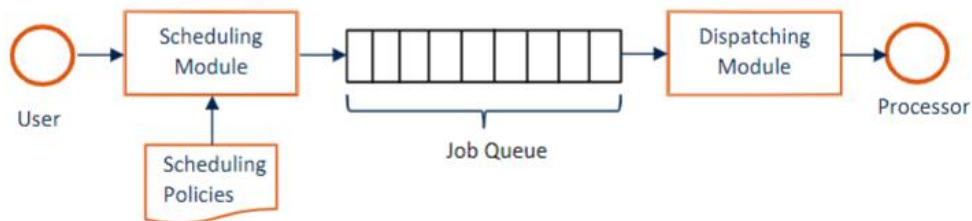
```
>list

Total Number of jobs in the queue: 2
Scheduling Policy: FCFS.
Name    CPU_Time    Priority    Progress
a       5           10         RUN
b       45          20
c       3          20
```

7) Quit: -

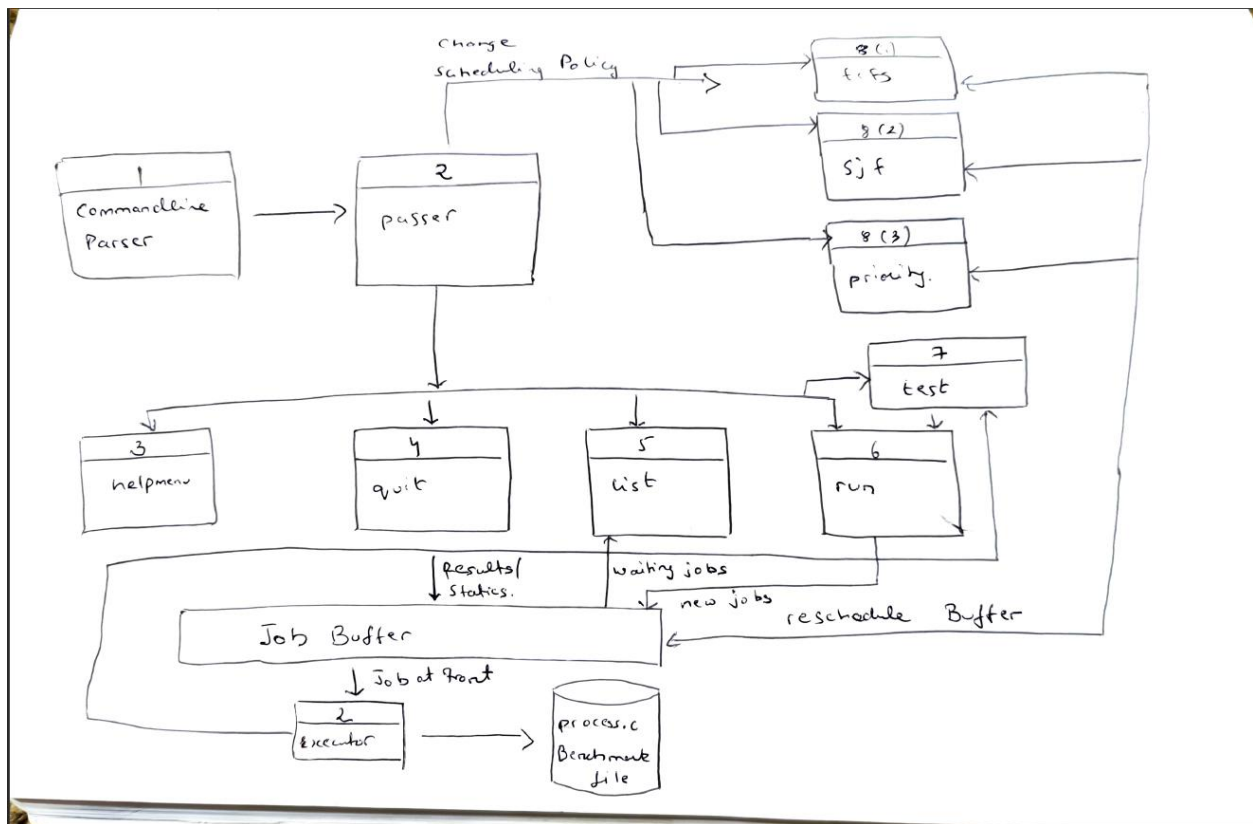
```
>quit
Total number of jobs submitted: 4
Average turnaround time:33.00 seconds
Average CPU time:15.50 seconds
Average waiting time:17.50 seconds
Throughput:0.06 No./second
[centos@localhost ~]$
```

System Design: -



**Fig. 1 The system architecture of the AUBatch scheduler**

Dataflow diagram: -



Performance Metrics: -

Calculated turnaround time, CPU time, waiting time  
Passed 6 jobs in different scheduling algorithms.

Priority Scheduling: -

```

>test testpriority priority 6 8 14 20
Total number of jobs submitted: 18
Average turnaround time: 53.17 seconds
Average CPU time: 16.28 seconds
Average waiting time: 36.89 seconds
Throughput: 0.06 No./second
  
```

Shortest Job First: -

```

>test testsjf sjf 6 8 14 20
Total number of jobs submitted: 12
Average turnaround time: 50.00 seconds
Average CPU time: 15.42 seconds
Average waiting time: 34.58 seconds
Throughput: 0.06 No./second
  
```

First Come First Serve: -

```
>test testfcfs fcfs 6 8 14 20
Total number of jobs submitted: 6
Average turnaround time: 63.17 seconds
Average CPU time: 18.00 seconds
Average waiting time: 45.17 seconds
Throughput:      0.06 No./second
```

Performance Evaluation: -

The waiting time of FCFS is higher than the other two, which makes the performance of FCFS the worst of the three. SJF had the quickest turnaround time and best performance followed by priority scheduling policy but priority being randomly assigned to each job, FCFS could have performed better. Arrival rate, number of jobs and processing time for each job are factors that determine the throughput(jobs/sec).

Lessons learnt: -

The project gives a clear picture on how scheduling algorithms function in coding or practically, where factors like PC performance effect in values from theoretical values. Also, this project taught me how to use pthread conditions to reduce system latency. Synchronization shows how it increases the performance of the system. Makefile makes it easier to execute bunch of files together and gives an order for execution of files.

(Note: - we can remove all executable files after running the application by make -f makerfile.txt clean)