

Eric Smith  
March 27<sup>th</sup>, 2017

## CSE7343 – Semester Project Phase 1

### Introduction

My semester project was written in C++. I chose this language because it's the one I love most! I also have past experience using C to create the necessary data structures.

### Compilation Instructions

This project was compiled and tested on both Linux and MacOS machines.

To run the program, simply compile all .CPP files using the C++11 standard compiler. Then, call the resulting executable.

```
[Eric's-MacBook-Pro:CSE7343 - Semester Project ericsmith$ g++ -std=c++11 *.cpp  
Eric's-MacBook-Pro:CSE7343 - Semester Project ericsmith$ ./a.out
```

### The Queues

The application has two queues: the waiting and ready queues. The user has the ability to add to, delete from, and print both queues. However, the waiting queue only exists as a proof of concept. Those processes that are added to the *ready* queue will be the ones that are executed under “[3] Execute processes”.

To add to the ready queue, enter “1” when prompted at the main menu. You will be shown the three options for interacting with the ready queue: print, add, and delete.

```
#####  
# Welcome to EricOS #  
# Version 1.0 #  
# #  
# Main Menu: #  
# #  
# [1] Edit Ready Queue #  
# [2] Edit Waiting Queue #  
# [3] Execute processes #  
# #  
# [0] Exit #  
# #  
#####  
  
Please select a mode: 1
```

<pre>##### #                                     # # Edit queue:                         # #                                     # # [1] Print Queue                     # # [2] Add to Queue                   # # [3] Delete from Queue               # #                                     # # [0] Go Back                         # #                                     # ##### Select operation to perform on Ready queue: 2</pre>	<pre>Queue is empty.  ##### #                                     # # Add to queue:                       # #                                     # # [1] Add from file                   # # [2] Add from command line           # #                                     # # [0] Go Back                         # #                                     # #####  Please select a mode: 1 Please enter a file to parse: ([0] to go back) input.txt</pre>
---	---

When you enter “2”, if any queues are in the ready queue, they will be printed out. You’ll then have the choice of adding to the queue from a file or the command line. Above is an example of adding from a file.

Please enter processes in the following format:  
Process ID (1-99999), Arrival Time (0-9999), Burst Time (0-9999), Priority (1-4)

If there are any errors in the file passed to the program, they will be displayed above the menu. All other processes’ whose lines were valid will be loaded into the ready queue.

<pre>5 processes successfully entered. 0 errors generated.  ##### #                                     # # Edit queue:                         # #                                     # # [1] Print Queue                     # # [2] Add to Queue                   # # [3] Delete from Queue               # #                                     # # [0] Go Back                         # #                                     # ##### Select operation to perform on Ready queue: </pre>	<pre>Error: Duplicate process on line 2. Command ignored.  4 processes successfully entered. 1 errors generated  ##### #                                     # # Edit queue:                         # #                                     # # [1] Print Queue                     # # [2] Add to Queue                   # # [3] Delete from Queue               # #                                     # # [0] Go Back                         # #                                     # ##### Select operation to perform on Ready queue: </pre>
---	--

Processes can also be added from the command line. From the edit queue menu, press “2” again to add to the queue, then select “2” again to add from the command line. You’ll then have the option to add the process at an index of the queue (starting at zero) or add to the tail of the queue by entering “default”.

```
#####
#                                     #
# Add to queue:                       #
#                                     #
# [1] Add from file                   #
# [2] Add from command line           #
#                                     #
# [0] Go Back                         #
#                                     #
#####

Please select a mode: 2
Please enter a process to parse: ([0] to go back)
6,0,2,1

Enter the position to add the process:
(Positions indexed at 0. Enter "default" to add to tail.)
5

Process added to the Ready queue at position: 5.
Please enter a process to parse: ([0] to go back)

```

```
----- Ready Queue -----
0 - ID: 1, ArrivalTime: 8, BurstTime: 6, Priority: 2 (Head)
1 - ID: 2, ArrivalTime: 8, BurstTime: 2, Priority: 1
2 - ID: 3, ArrivalTime: 0, BurstTime: 8, Priority: 1
3 - ID: 4, ArrivalTime: 2, BurstTime: 5, Priority: 3
4 - ID: 5, ArrivalTime: 6, BurstTime: 10, Priority: 4
5 - ID: 6, ArrivalTime: 0, BurstTime: 2, Priority: 1 (Tail)

#####
#                                     #
# Edit queue:                         #
#                                     #
# [1] Print Queue                     #
# [2] Add to Queue                    #
# [3] Delete from Queue               #
#                                     #
# [0] Go Back                         #
#                                     #
#####

Select operation to perform on Ready queue:

```

The same error checking rules apply to the command that apply to entering a file, any logical or syntactical errors will prevent the process from being added to the queue.

```
----- Ready Queue -----
0 - ID: 1, ArrivalTime: 8, BurstTime: 6, Priority: 2 (Head)
1 - ID: 2, ArrivalTime: 8, BurstTime: 2, Priority: 1
2 - ID: 3, ArrivalTime: 0, BurstTime: 8, Priority: 1
3 - ID: 4, ArrivalTime: 2, BurstTime: 5, Priority: 3
4 - ID: 5, ArrivalTime: 6, BurstTime: 10, Priority: 4 (Tail)

#####
#                                     #
# Add to queue:                       #
#                                     #
# [1] Add from file                   #
# [2] Add from command line           #
#                                     #
# [0] Go Back                         #
#                                     #
#####

Please select a mode: 2
Please enter a process to parse: ([0] to go back)
1, 0, 3, 4

Enter the position to add the process:
(Positions indexed at 0. Enter "default" to add to tail.)
0
Error: Duplicate process. Command ignored.
```

To delete from a queue, select “3” from the edit queue menu. You can then enter the ID of the process to delete. (Process 3 was deleted in the example below)

<pre>----- Ready Queue ----- 0 - ID: 1, ArrivalTime: 8, BurstTime: 6, Priority: 2 (Head) 1 - ID: 2, ArrivalTime: 8, BurstTime: 2, Priority: 1 2 - ID: 3, ArrivalTime: 0, BurstTime: 8, Priority: 1 3 - ID: 4, ArrivalTime: 2, BurstTime: 5, Priority: 3 4 - ID: 5, ArrivalTime: 6, BurstTime: 10, Priority: 4 (Tail)  ##### #                                     # # Edit queue:                         # #                                     # # [1] Print Queue                     # # [2] Add to Queue                    # # [3] Delete from Queue               # #                                     # # [0] Go Back                         # #                                     # #####  Enter the ID of process to delete: (Enter [0] for default position.) 3</pre>	<pre>----- Ready Queue ----- 0 - ID: 1, ArrivalTime: 8, BurstTime: 6, Priority: 2 (Head) 1 - ID: 2, ArrivalTime: 8, BurstTime: 2, Priority: 1 2 - ID: 4, ArrivalTime: 2, BurstTime: 5, Priority: 3 3 - ID: 5, ArrivalTime: 6, BurstTime: 10, Priority: 4 (Tail)  ##### #                                     # # Edit queue:                         # #                                     # # [1] Print Queue                     # # [2] Add to Queue                    # # [3] Delete from Queue               # #                                     # # [0] Go Back                         # #                                     # #####  Select operation to perform on Ready queue: </pre>
---	--

## Scheduling Algorithms

Now that the ready queue has some processes, let’s run the scheduling algorithms on the ready queue. Choose “3” at the main menu to run the processes, then enter a desired time quantum for round robin.

```
#####
# Welcome to EricOS                  #
# Version 1.0                        #
#                                    #
# Main Menu:                         #
#                                    #
# [1] Edit Ready Queue               #
# [2] Edit Waiting Queue             #
# [3] Execute processes              #
#                                    #
# [0] Exit                           #
#                                    #
#####

Please select a mode: 3
Please enter a quantum for Round Robin: <1-999>
5
```

You'll then be shown the results of all the scheduling algorithms at the same time. Please note that SJF, FCFS, and Priority scheduling all happen NON-preemptively. In the case of first come first serve, processes are chosen by arrival time. If two arrival times are the same, priority is used as a tie breaker. For round robin, the next process to have arrived is selected to run. The column "Wait Time" shows the wait time with that individual process burst. The column "Total Wait Time" shows the wait time associated with the process as a whole.

----- Shortest Job First -----			
Process	Start Time	End Time	Wait Time
1 - P3	0	8	0
2 - P2	8	10	0
3 - P4	10	15	8
4 - P1	15	21	7
5 - P5	21	31	15

Average waiting time: 6

----- First Come First Serve -----			
Process	Start Time	End Time	Wait Time
1 - P3	0	8	0
2 - P4	8	13	6
3 - P5	13	23	7
4 - P2	23	25	15
5 - P1	25	31	17

Average waiting time: 9

----- Priority -----			
Process	Start Time	End Time	Wait Time
1 - P3	0	8	0
2 - P2	8	10	0
3 - P1	10	16	2
4 - P4	16	21	14
5 - P5	21	31	15

Average waiting time: 6.2

----- Round Robin -----				
Process	Start Time	End Time	Wait Time	Total Wait Time
1 - P3	0	5	0	17
2 - P4	5	10	5	3
3 - P5	10	15	10	14
4 - P1	15	20	15	17
5 - P2	20	22	20	12
6 - P3	22	25	17	17
7 - P5	25	30	10	14
8 - P1	30	31	10	17

Average wait time: 12.6

Enter [0] to go back.

█