

# CISC 361 Final Project

Chris Bellotti and Ashley Gold

## How to Run:

To run our program that is written in C, you can type `"gcc -std=c99 -o exe main.c"` into a terminal running bash to compile it first, then run it using the command `"./exe"`. We have included all sample inputs as .txt files (`"sample_input.txt"`, `"test_input1.txt"`, and `"test_input2.txt"`). To test the different inputs, change the `FILE_NAME` in line 9. It is set to `"test_input1.txt"` by default.

## Description of Approach:

Our program has three source files: `main.c`, `queues.c`, and `types.h`. Starting with the types header file, we created a struct called `"job"` that contains fields for every aspect of the job: its arrival time, completion time, turnaround time, weighted turnaround time, job number (id), priority, memory units required, memory units allocated, runtime, remaining time, maximum devices, devices allocated, devices requested, and whether a process exists for that job or not. Then, we created a struct called `"node"` where each node contains a job and a next pointer so that we could easily put jobs in queues and move them around. Finally, we created a struct called `"queue"` with first and last pointers as well as a size field so that we could create all the queues we needed.

Moving on to `queues.c`, this is where we created functions related to creating, sorting, and removing jobs, queues and nodes. First there are three functions to create a new job, node and queue. Then, there are functions to add a job to a queue, delete the first node of a queue, and delete a specific job from a queue. Finally, there is a function to sort a queue's jobs by runtime so that the Shortest Job First scheduling algorithm can be appropriately used. Additionally, there is a swap helper function to switch the jobs of a node that is used in both `sortByRuntime` and `removeFromQueue`.

The most important part of the project is `main.c`. In the main function there, we first create all the queues that we need to simulate our system (submit, hold 1, hold 2, ready, wait, running and complete). We also created an extra queue to store all of the jobs that we have accepted over the course of the simulation. Next, we open the input file for reading, and created a function to read the file by line. Each line of the file is parsed to find the key associated with a given command, and the values passed in are used accordingly. Once we read in a new command, we start simulating to the time of that event using the provided quantum. During this time, all ready processes are swapped into the cpu in a round robin fashion. Completed jobs are dealt with during this

time as well. When we reach the time of the complete job command, we call the corresponding helper functions to perform that action. Upon request of devices while running on the cpu, the devices are allocated and we check if we are in a safe state using Banker's algorithm. If we are not in a safe state, the devices are deallocated and we save the request for when it can be fulfilled. Banker's algorithm is also used whenever we release devices or complete jobs to see if these previously unfulfilled requests can now be filled. Finally, when a 'D' is parsed from input, if there are one or more jobs completed, the average turnaround time and average weighted turnaround time will be computed using helper functions. Also when a 'D' is parsed, the program prints out a summary of the system at the given time using output(), including all system variables, all queues, and all jobs. This same output is also written and saved to a .json file using the generateJSON() function.

It should be noted that we ran into some difficulties during the project. The first two sample inputs, sample\_input.txt and test\_input1.txt compile and execute nearly flawlessly. The console output generated by those is 100% correct. The third sample input, test\_input2.txt, caused us problems and we are aware that the console output is incorrect, however after attempting to debug for around ten hours we decided to call it. Additionally, the .json outputs are finicky. We find that json files are always created for test\_input1 but sometimes problems arise with sample\_input and test\_input2.

## Sample Output

"sample\_input.txt"

```
*****System Information*****
  Current Time: 9999
  Total Memory: 200
  Available Memory: 200
  Total Devices: 12
  Available Devices: 12
  Quantum: 4
  Turnaround Time: 29
  Weighted Turnaround Time: 32
*****ReadyQueue*****
  []
*****RunningQueue*****

*****SubmitQueue*****
  []
*****HoldQueue2*****
```

```

[]
*****HoldQueue1*****
[]
*****CompleteQueue*****
[3, 1, 2, 4, 5, 6, ]
*****WaitQueue*****
[]
*****Jobs*****
[
    {
        Arrival Time: 3,
        Job Number: 1,
        Remaining Time: 0,
        Completion Time: 29,
    },
    {
        Arrival Time: 4,
        Job Number: 2,
        Remaining Time: 0,
        Completion Time: 33,
    },
    {
        Arrival Time: 9,
        Job Number: 3,
        Remaining Time: 0,
        Completion Time: 19,
    },
    {
        Arrival Time: 13,
        Job Number: 4,
        Remaining Time: 0,
        Completion Time: 56,
    },
    {
        Arrival Time: 24,
        Job Number: 5,
        Remaining Time: 0,
        Completion Time: 57,
    },
    {
        Arrival Time: 25,
        Job Number: 6,
        Remaining Time: 0,
        Completion Time: 61,
    },
]
}

```

“test\_input1.txt”

\*\*\*\*\*System Information\*\*\*\*\*

Current Time: 10

Total Memory: 200

Available Memory: 0

Total Devices: 12

Available Devices: 12

Quantum: 4

\*\*\*\*\*ReadyQueue\*\*\*\*\*

[1, 5, ]

\*\*\*\*\*RunningQueue\*\*\*\*\*

2

\*\*\*\*\*SubmitQueue\*\*\*\*\*

[]

\*\*\*\*\*HoldQueue2\*\*\*\*\*

[]

\*\*\*\*\*HoldQueue1\*\*\*\*\*

[3, ]

\*\*\*\*\*CompleteQueue\*\*\*\*\*

[]

\*\*\*\*\*WaitQueue\*\*\*\*\*

[]

\*\*\*\*\*Jobs\*\*\*\*\*

[

{

Arrival Time: 4,

Job Number: 2,

Remaining Time: 9,

},

{

Arrival Time: 5,

Job Number: 3,

Remaining Time: 4,

},

{

Arrival Time: 3,

Job Number: 1,

Remaining Time: 6,

},

{

Arrival Time: 8,

Job Number: 5,

Remaining Time: 4,

},

]

}

```

*****System Information*****
    Current Time: 26
    Total Memory: 200
    Available Memory: 30
    Total Devices: 12
    Available Devices: 12
    Quantum: 4
    Turnaround Time: 16
    Weighted Turnaround Time: 18
*****ReadyQueue*****
    [3, ]
*****RunningQueue*****
    2
*****SubmitQueue*****
    []
*****HoldQueue2*****
    []
*****HoldQueue1*****
    []
*****CompleteQueue*****
    [5, 1, ]
*****WaitQueue*****
    []
*****Jobs*****
    [
        {
            Arrival Time: 4,
            Job Number: 2,
            Remaining Time: 3,
        },
        {
            Arrival Time: 5,
            Job Number: 3,
            Remaining Time: 4,
        },
        {
            Arrival Time: 3,
            Job Number: 1,
            Remaining Time: 0,
            Completion Time: 25,
        },
        {
            Arrival Time: 8,
            Job Number: 5,
            Remaining Time: 0,
            Completion Time: 19,
        },
    ],

```

```
]
}
```

“test\_input2.txt”

\*\*\*\*\*System Information\*\*\*\*\*

Current Time: 8  
Total Memory: 200  
Available Memory: 10  
Total Devices: 12  
Available Devices: 2  
Quantum: 4

\*\*\*\*\*ReadyQueue\*\*\*\*\*

[1, ]

\*\*\*\*\*RunningQueue\*\*\*\*\*

1

\*\*\*\*\*SubmitQueue\*\*\*\*\*

[]

\*\*\*\*\*HoldQueue2\*\*\*\*\*

[]

\*\*\*\*\*HoldQueue1\*\*\*\*\*

[]

\*\*\*\*\*CompleteQueue\*\*\*\*\*

[]

\*\*\*\*\*WaitQueue\*\*\*\*\*

[2, ]

\*\*\*\*\*Jobs\*\*\*\*\*

[

{

Arrival Time: 3,  
Devices Allocated: 10,  
Job Number: 1,  
Remaining Time: 6,

},

{

Arrival Time: 4,  
Devices Allocated: 0,  
Job Number: 2,  
Remaining Time: 11,

},

]

}

\*\*\*\*\*System Information\*\*\*\*\*

Current Time: 11  
Total Memory: 200  
Available Memory: 0  
Total Devices: 12

```

    Available Devices: 2
    Quantum: 4
    *****ReadyQueue*****
    [3, 1, ]
    *****RunningQueue*****
    1
    *****SubmitQueue*****
    []
    *****HoldQueue2*****
    []
    *****HoldQueue1*****
    []
    *****CompleteQueue*****
    []
    *****WaitQueue*****
    [2, ]
    *****Jobs*****
    [
        {
            Arrival Time: 3,
            Devices Allocated: 10,
            Job Number: 1,
            Remaining Time: 3,
        },
        {
            Arrival Time: 4,
            Devices Allocated: 0,
            Job Number: 2,
            Remaining Time: 11,
        },
        {
            Arrival Time: 10,
            Job Number: 3,
            Remaining Time: 4,
        },
    ]
}

```

```

*****System Information*****
    Current Time: 20
    Total Memory: 200
    Available Memory: 130
    Total Devices: 12
    Available Devices: 9
    Quantum: 4
    Turnaround Time: 12
    Weighted Turnaround Time: 14
    *****ReadyQueue*****

```

```

[]
*****RunningQueue*****
2
*****SubmitQueue*****
[]
*****HoldQueue2*****
[]
*****HoldQueue1*****
[]
*****CompleteQueue*****
[1, 3, 1, ]
*****WaitQueue*****
[]
*****Jobs*****
[
    {
        Arrival Time: 3,
        Job Number: 1,
        Remaining Time: 0,
        Completion Time: 18,
    },
    {
        Arrival Time: 4,
        Devices Allocated: 3,
        Job Number: 2,
        Remaining Time: 9,
    },
    {
        Arrival Time: 10,
        Job Number: 3,
        Remaining Time: 0,
        Completion Time: 18,
    },
]
}

```