

Queue practice

```
// This program simulates jobs which are of random assigned lengths
// and have random delays between their arrival. All time values are
// actual time values using <chrono> at sub-millisecond units.
//
// The Simulation object creates a vector of Virtual Machine (VM) objects, each
// one containing a vector of CPU objects.
//
// The simulation also creates a Run Queue object for jobs that need to be
// processed, and a Completed Queue object for jobs that are done. Both are
// custom Template Linked Lists that hold Job objects.
//
// For each CPU of each VM, the following logic is performed:
//
// - Test if a new job has "arrived" based on time comparison against
// delay. If it has:
// Create the job object, stamp time, place it into the Run Queue, and
// increase Run Queue length. Write information about the queue event to
// Queue Event CSV file (including the size of the queue) - for import
// into Excel to visualize time and queue size with scatterplot diagram.
// Also test to see if the new job is the smallest or biggest so far.
//
// - Test if the current CPU has a job and is done processing it (elapsed
// time on cpu is longer than assigned length). If so, stamp time and move
// job to Completed Queue. Also, if the final job was just moved to
// completed queue, set simulation Done flag to true.
//
// - Test if the current processor is ready and there is a job available
// in the Run Queue. If so, stamp time, move job from Run Queue to the CPU,
// and decrease Run Queue length. Write information about the queue event to
// CSV file (including the new size of the queue).
//
// - If the simulation is not done, move to next Machine/CPU
```

Program Files:

```
main.cpp      // Main routine
Sim.cpp       // Simulation Object
Sim.h
MyList.h      // Templated Linked List
MyQueue.h     // Extends MyList
VM.cpp        // Virtual Machine Obj
VM.h
CPU.cpp       // CPU Obj
CPU.h
Job.cpp       // Job Object
Job.h
```

Program Generated Output Files:

```
QueueSequence.csv  // The order jobs enter and exit Run queue
Done_Queue.txt     // Entire listing of Job objs in done order
```

Additional Files for Analysis and Visualization:

```
doneQueue300.xlsx  // Excel graph of 300 job sequence
doneQueue4000.xlsx // Excel graph of 4000 job sequence
```

Requirements:

Calculate the following information about each CPU utilization:

1. Total busy time
2. Total idle time
3. Number of jobs served by the CPU.

Use actual time as hours, minutes, and seconds instead of integer value to process each job and assign different jobs length.

Also used csv file generated to import into Excel and graph queue size in relation to time.

Machine Configuration:

Summary of configuration including number of machines, CPUs per machine, number of jobs requested, and the limits used for random creation of job arrival delays and requested job lengths.

CPU Statistics:

Listing of each CPU on each Machine, Total jobs run, total busy time, and total idle time.

The screenshot displays the Visual Studio IDE with the following components:

- Project Explorer (Left):** Shows the project structure for `DataStruct_HW03`. The file `QueueSequence.csv` is highlighted.
- Run Window (Top Right):** Displays the output of the program. The output includes:


```
=====
                        MACHINE CONFIGURATION
                        -----

NUMBER OF VIRTUAL MACHINES: 2
NUMBER OF CPUS PER VIRTUAL MACHINE: 4

NUMBER OF JOBS SIMULATED: 300

- RANDOM LIMITS (MICROSECONDS) -
MIN DELAY:      5      MIN RUN REQUEST:      1
MAX DELAY:     200     MAX RUN REQUEST:     5000

=====

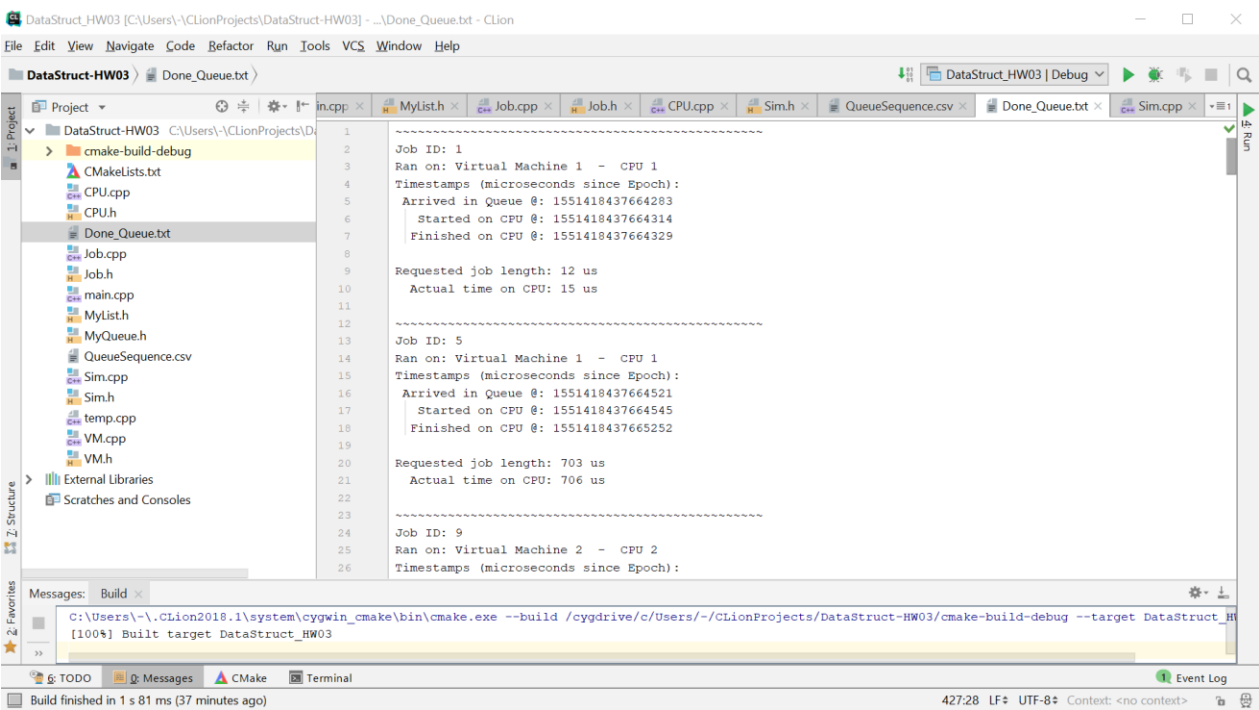
                        CPU STATISTICS
                        -----

=====

****  VM 1 - CPU 1  ****
Total Jobs Served: 36
Total Processing Time: 95818 us
Total Idle Time: 1796 us
```
- Messages Window (Bottom):** Shows the build log. The message indicates that the target `DataStruct_HW03` was built successfully.


```
C:\Users\~\Clien2018.1\system\cygwin_cmake\bin\cmake.exe --build /cygdrive/c/Users/~\ClienProjects/DataStruct-HW03/cmake-build-debug --target DataStruct_HW03
[100%] Built target DataStruct_HW03
```

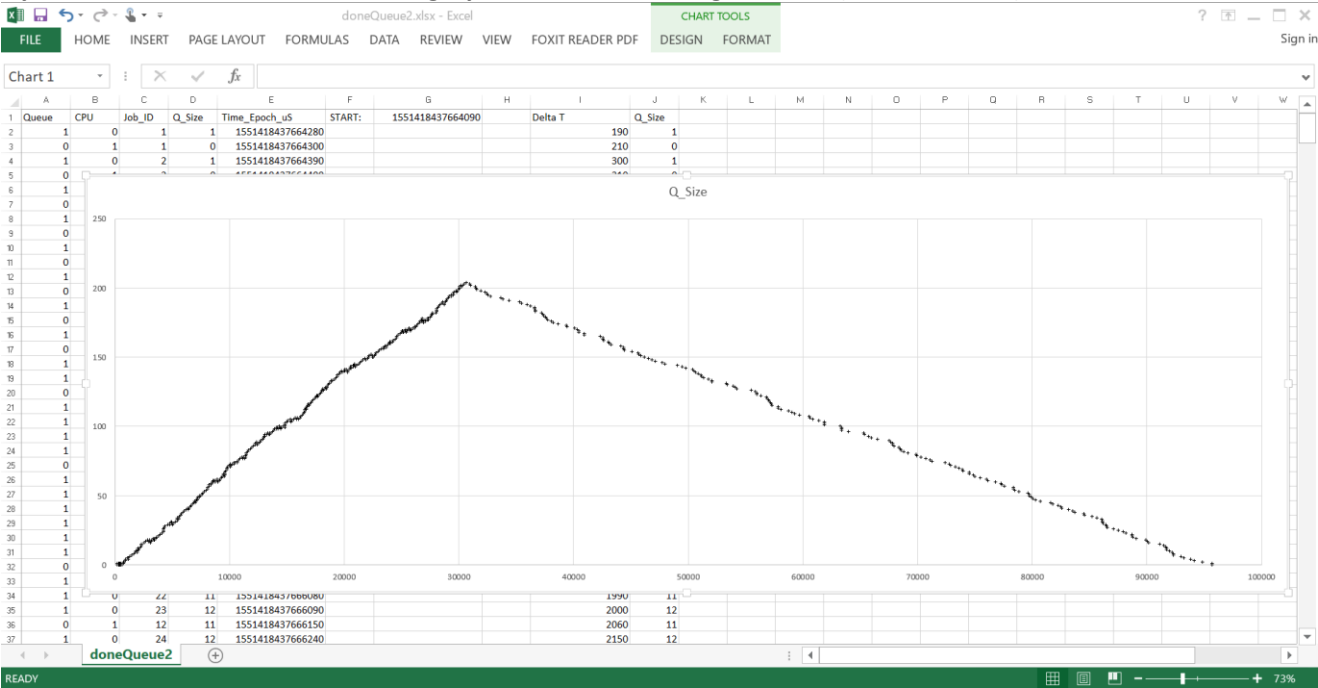
Completed job information written to Done_Queue.txt.
Includes ID number, which Machine and CPU it ran on, arrival times in run queue, CPU, & done queue, random length assigned, and actual run time.



Done Queue sequence of events written to QueueSequence.csv :

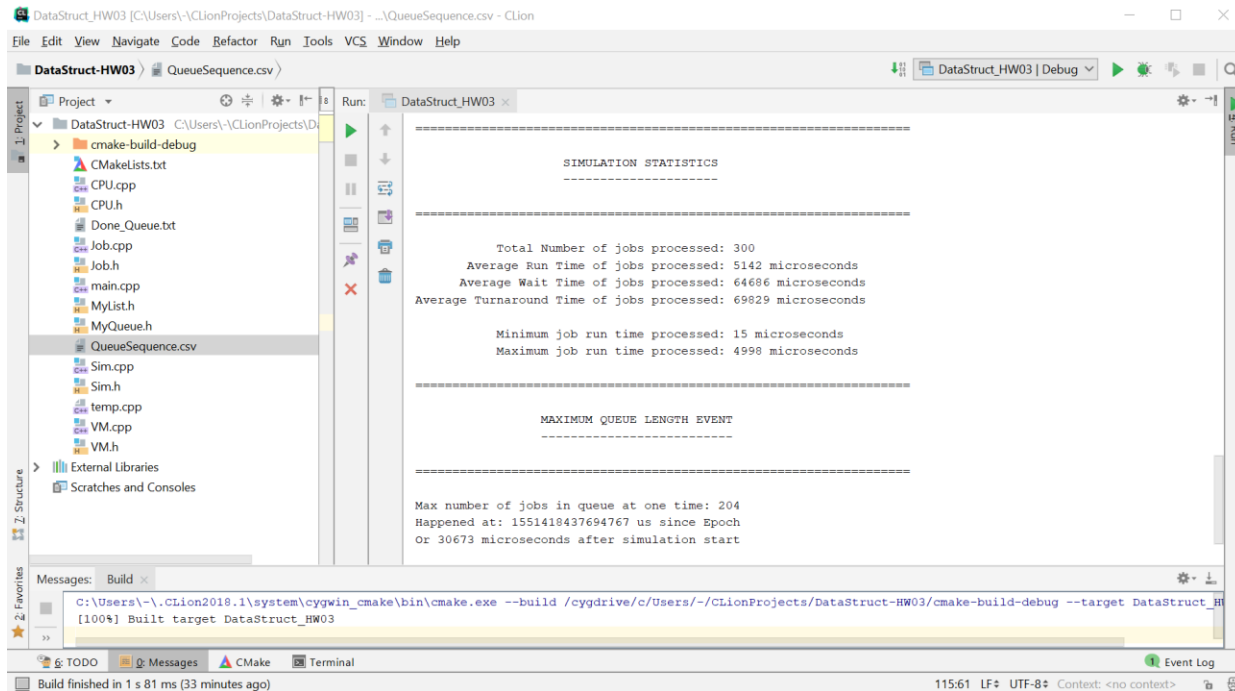
```
toQueue, toCPU, Job_ID, Q_Size, Time_Epoch_uS, START:, 1551418437664094
1, 0, 1, 1, 1551418437664286
0, 1, 1, 0, 1551418437664305
1, 0, 2, 1, 1551418437664392
0, 1, 2, 0, 1551418437664401
1, 0, 3, 1, 1551418437664469
0, 1, 3, 0, 1551418437664480
1, 0, 4, 1, 1551418437664504
0, 1, 4, 0, 1551418437664513
```

Opened with Excel and Queue size graphed based on change in time (microseconds)



Overall simulation statistics: number of jobs run, average run time, average wait time, average turnaround time, minimum run time, and maximum run time.

Maximum number of jobs in queue at once and time it occurred. Maximum size and time match the information in the graph.



The screenshot shows the CLion IDE interface with the file `QueueSequence.csv` open. The main editor displays the following simulation statistics:

```
=====
SIMULATION STATISTICS
=====

Total Number of jobs processed: 300
Average Run Time of jobs processed: 5142 microseconds
Average Wait Time of jobs processed: 64686 microseconds
Average Turnaround Time of jobs processed: 69829 microseconds

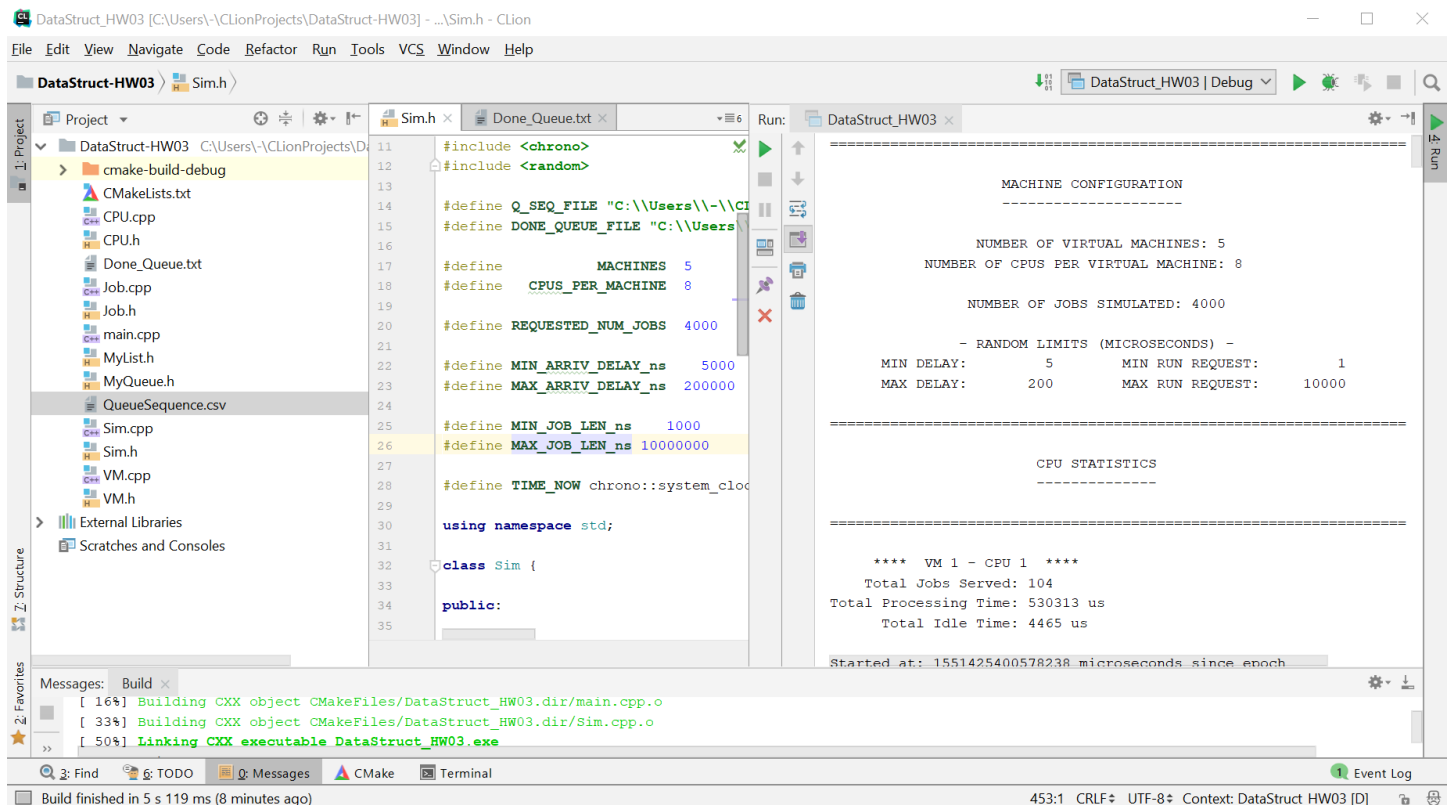
Minimum job run time processed: 15 microseconds
Maximum job run time processed: 4998 microseconds

=====

MAXIMUM QUEUE LENGTH EVENT
=====

Max number of jobs in queue at one time: 204
Happened at: 1551418437694767 us since Epoch
Or 30673 microseconds after simulation start
=====
```

The bottom status bar indicates the build finished in 1 s 81 ms (33 minutes ago).



The screenshot shows the CLion IDE interface with the file `Sim.h` open. The main editor displays the following machine configuration and CPU statistics:

```
=====
MACHINE CONFIGURATION
=====

NUMBER OF VIRTUAL MACHINES: 5
NUMBER OF CPUS PER VIRTUAL MACHINE: 8

NUMBER OF JOBS SIMULATED: 4000

- RANDOM LIMITS (MICROSECONDS) -
MIN DELAY: 5 MIN RUN REQUEST: 1
MAX DELAY: 200 MAX RUN REQUEST: 10000

=====

CPU STATISTICS
=====

**** VM 1 - CPU 1 ****
Total Jobs Served: 104
Total Processing Time: 530313 us
Total Idle Time: 4465 us

Started at: 1551425400578238 microseconds since epoch
=====
```

The bottom status bar indicates the build finished in 5 s 119 ms (8 minutes ago).

4,000 job run completed successfully.

DataStruct_HW03 [C:\Users\...\CLionProjects\DataStruct-HW03] - ...Sim.h - CLion

File Edit View Navigate Code Refactor Run Tools VCS Window Help

DataStruct-HW03 > Done_Queue.txt

Project: DataStruct-HW03

- cmake-build-debug
 - CMakeLists.txt
 - CPU.cpp
 - CPU.h
 - Done_Queue.txt
 - Job.cpp
 - Job.h
 - main.cpp
 - MyList.h
 - MyQueue.h
 - Sim.cpp
 - Sim.h
 - VM.cpp
 - VM.h
- External Libraries
- Scratches and Consoles

Sim.h

```
11 #include <chrono>
12 #include <random>
13
14 #define Q_SEQ_FILE "C:\\Users\\...\\CL
15 #define DONE_QUEUE_FILE "C:\\Users
16
17 #define
18 #define MACHINES 5
19 #define CPUS_PER_MACHINE 8
20
21 #define REQUESTED_NUM_JOBS 4000
22
23 #define MIN_ARRIV_DELAY_ns 5000
24 #define MAX_ARRIV_DELAY_ns 200000
25
26 #define MIN_JOB_LEN_ns 1000
27 #define MAX_JOB_LEN_ns 10000000
28
29 #define TIME_NOW chrono::system_clock
30
31 using namespace std;
32
33 class Sim {
34 public:
35
```

Run: DataStruct_HW03 | Debug

SIMULATION STATISTICS

Total Number of jobs processed: 4000
Average Run Time of jobs processed: 10605 microseconds
Average Wait Time of jobs processed: 115413 microseconds
Average Turnaround Time of jobs processed: 126019 microseconds

Minimum job run time processed: 20 microseconds
Maximum job run time processed: 22452 microseconds

MAXIMUM QUEUE LENGTH EVENT

Max number of jobs in queue at one time: 917
Happened at: 1551425400989175 us since Epoch
Or 409711 microseconds after simulation start

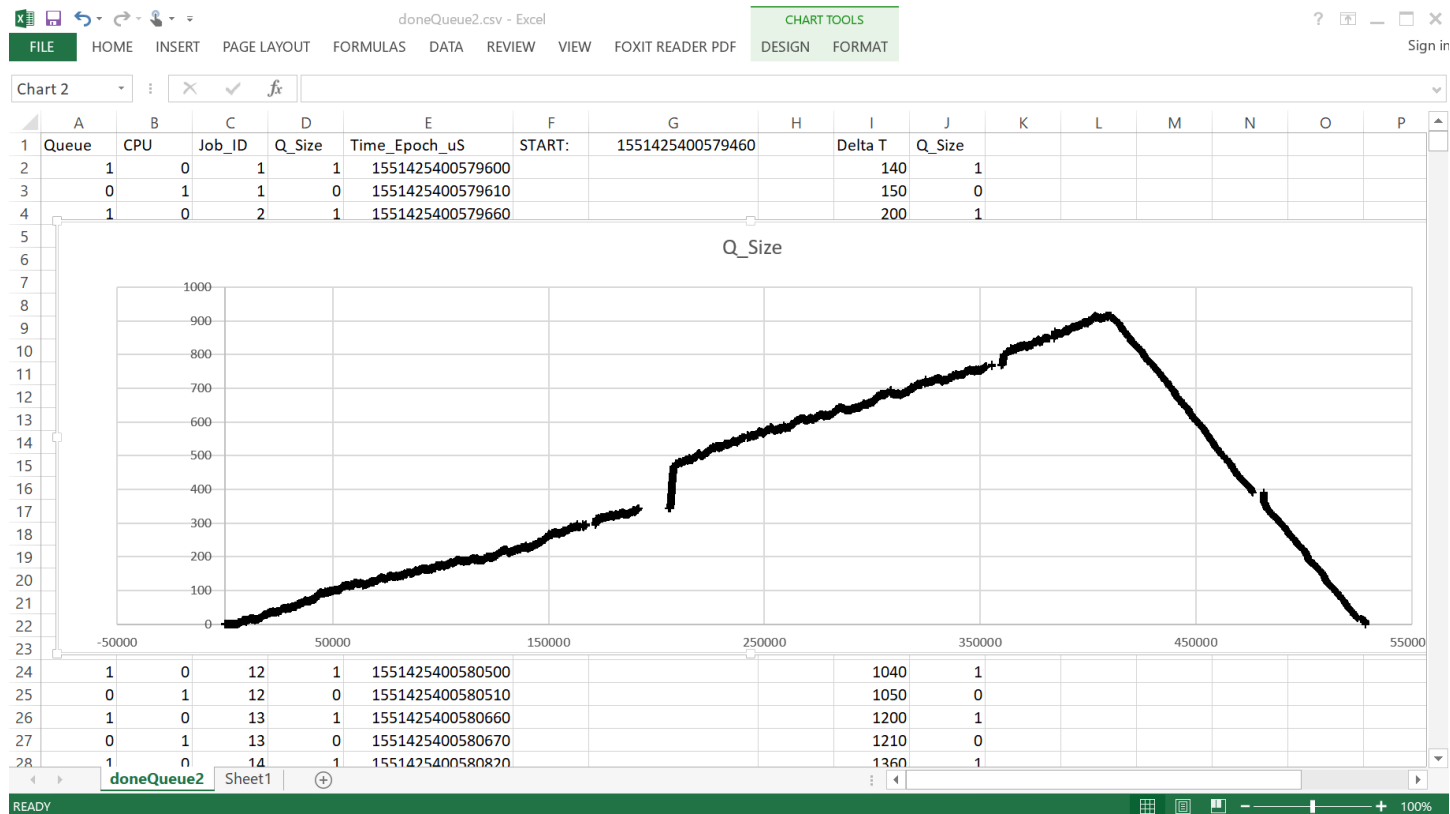
Messages: Build

- [16%] Building CXX object CMakeFiles/DataStruct_HW03.dir/main.cpp.o
- [33%] Building CXX object CMakeFiles/DataStruct_HW03.dir/Sim.cpp.o
- [50%] Linking CXX executable DataStruct_HW03.exe

Build finished in 5 s 119 ms (15 minutes ago)

453:1 CRLF+ UTF-8+ Context: DataStruct_HW03 [D]

Another graph of output file values showing a break, indicating the computer was doing something else during simulation



For Testing: set high and low limits of random values to the same value in order to test cumulative and averaging functions for reasonable values.

Set delay values high to eliminate any waiting in queue and set requested job length to 4 milliseconds. Maximum queue length correctly reported as 1, Average runtime correct at 4.005 milliseconds, Average wait time fairly close to 0 (.021 milliseconds), Average Turnaround time correct at 4.026 milliseconds, Minimum job length correct, Maximum job length correct and indicates other processes are happening on computer.

```
#include <vector>
#include "VM.h"
#include "MyQueue.h"
#include <chrono>
#include <random>

#define Q_SEQ_FILE "C:\\Users\\-\\CLionProjects\\DataStruct-HW03\\Queue"
#define DONE_QUEUE_FILE "C:\\Users\\-\\CLionProjects\\DataStruct-HW03\\Queue"

#define MACHINES 5
#define CPUS_PER_MACHINE 8
#define REQUESTED_NUM_JOBS 100

#define MIN_ARRIV_DELAY_ns 10000000
#define MAX_ARRIV_DELAY_ns 10000000

#define MIN_JOB_LEN_ns 4000000
#define MAX_JOB_LEN_ns 4000000

#define TIME_NOW chrono::system_clock::now()

using namespace std;

class Sim {
public:
    // ...
}
```

SIMULATION STATISTICS

Total Number of jobs processed: 100
Average Run Time of jobs processed: 4005 microseconds
Average Wait Time of jobs processed: 21 microseconds
Average Turnaround Time of jobs processed: 4026 microseconds

Minimum job run time processed: 4000 microseconds
Maximum job run time processed: 4261 microseconds

MAXIMUM QUEUE LENGTH EVENT

Max number of jobs in queue at one time: 1
Happened at: 1551487827930321 us since Epoch
Or 10054 microseconds after simulation start

Process finished with exit code 0

Set delay values to 1 microseconds to ensure high queue congestion (built up to 60), left job length at 4 milliseconds. Maximum queue length looks reasonable, Average runtime correct at 4.043 milliseconds, Average wait time increased to 2.8 milliseconds, Average Turnaround time value within expected values, Minimum job length correct, Maximum job length within expected values.

```
#include <vector>
#include "VM.h"
#include "MyQueue.h"
#include <chrono>
#include <random>

#define Q_SEQ_FILE "C:\\Users\\-\\CLionProjects\\DataStruct-HW03\\Queue"
#define DONE_QUEUE_FILE "C:\\Users\\-\\CLionProjects\\DataStruct-HW03\\Queue"

#define MACHINES 5
#define CPUS_PER_MACHINE 8
#define REQUESTED_NUM_JOBS 100

#define MIN_ARRIV_DELAY_ns 1000
#define MAX_ARRIV_DELAY_ns 1000

#define MIN_JOB_LEN_ns 4000000
#define MAX_JOB_LEN_ns 4000000

#define TIME_NOW chrono::system_clock::now()

using namespace std;

class Sim {
public:
    // ...
}
```

SIMULATION STATISTICS

Total Number of jobs processed: 100
Average Run Time of jobs processed: 4043 microseconds
Average Wait Time of jobs processed: 2777 microseconds
Average Turnaround Time of jobs processed: 6821 microseconds

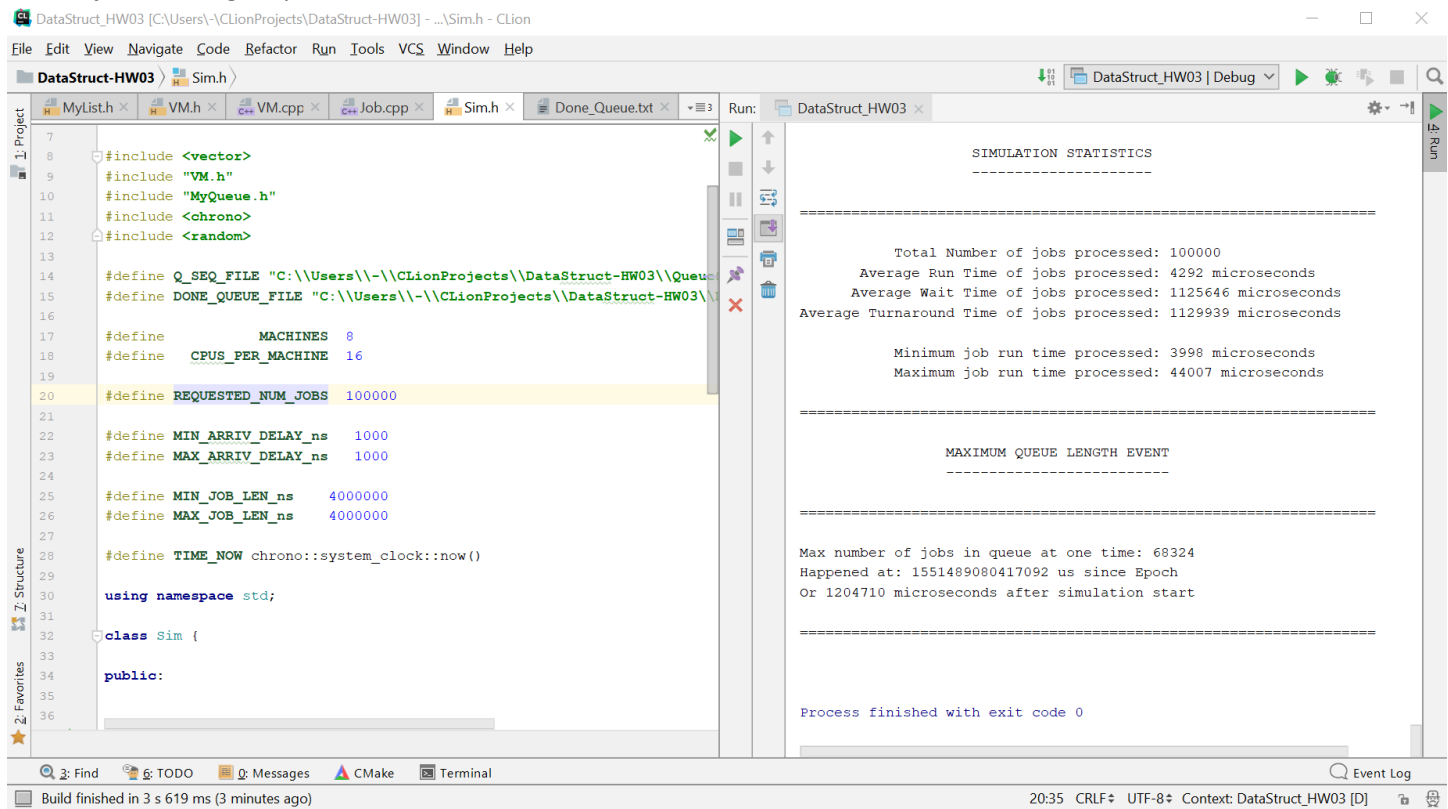
Minimum job run time processed: 4000 microseconds
Maximum job run time processed: 4179 microseconds

MAXIMUM QUEUE LENGTH EVENT

Max number of jobs in queue at one time: 60
Happened at: 1551488681188618 us since Epoch
Or 1736 microseconds after simulation start

Process finished with exit code 0

Requested 100,000 jobs processed on 8 virtual machines with 16 CPUs each. Queue congestion caused a maximum of 68,324 jobs waiting in queue.



```
#include <vector>
#include "VM.h"
#include "MyQueue.h"
#include <chrono>
#include <random>

#define Q_SEQ_FILE "C:\\Users\\-\\CLionProjects\\DataStruct-HW03\\QueueSeq.txt"
#define DONE_QUEUE_FILE "C:\\Users\\-\\CLionProjects\\DataStruct-HW03\\DoneQueue.txt"

#define MACHINES 8
#define CPUS_PER_MACHINE 16

#define REQUESTED_NUM_JOBS 100000

#define MIN_ARRIV_DELAY_ns 1000
#define MAX_ARRIV_DELAY_ns 1000

#define MIN_JOB_LEN_ns 4000000
#define MAX_JOB_LEN_ns 4000000

#define TIME_NOW chrono::system_clock::now()

using namespace std;

class Sim {
public:
    // ...
}
```

SIMULATION STATISTICS

Total Number of jobs processed: 100000
Average Run Time of jobs processed: 4292 microseconds
Average Wait Time of jobs processed: 1125646 microseconds
Average Turnaround Time of jobs processed: 1129939 microseconds

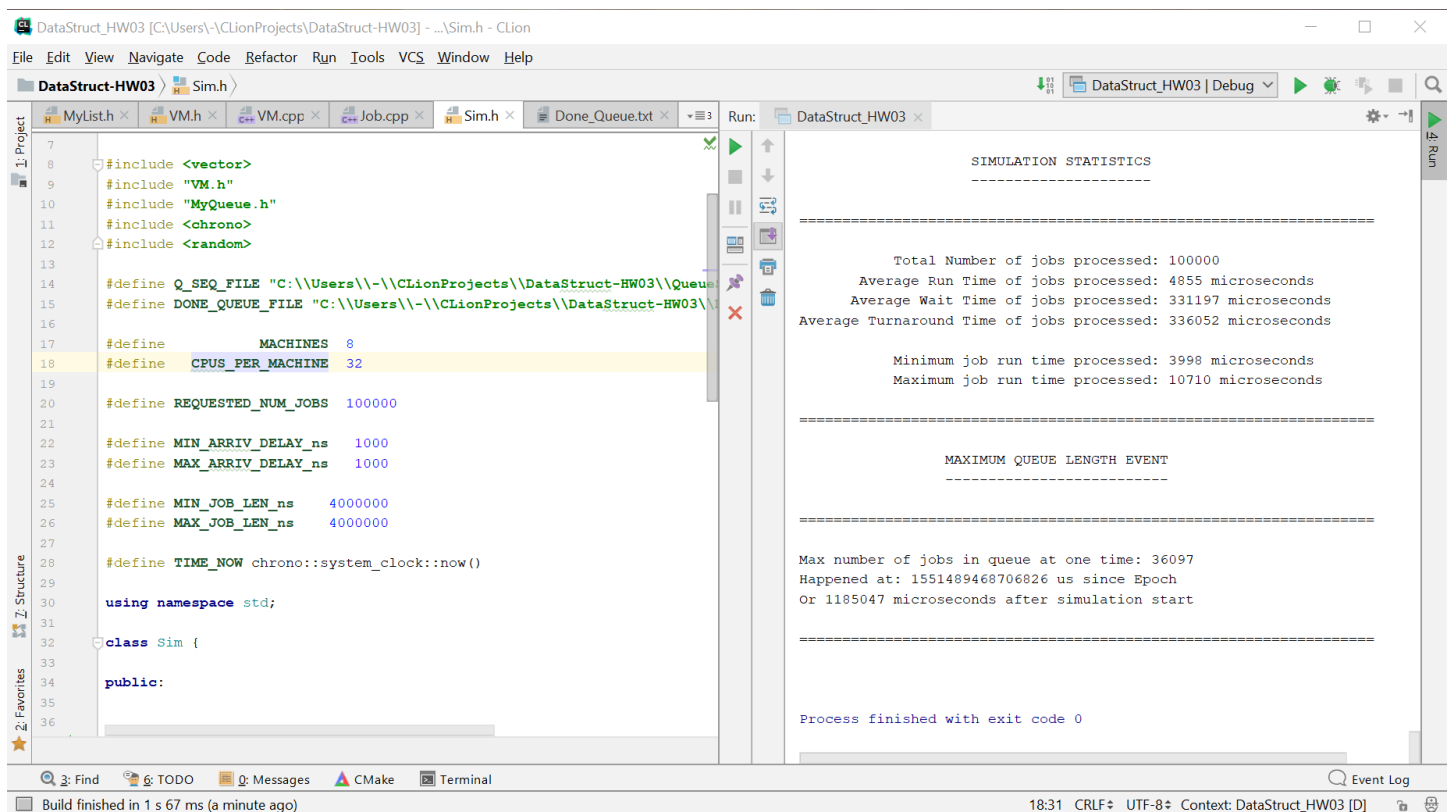
Minimum job run time processed: 3998 microseconds
Maximum job run time processed: 44007 microseconds

MAXIMUM QUEUE LENGTH EVENT

Max number of jobs in queue at one time: 68324
Happened at: 1551489080417092 us since Epoch
Or 1204710 microseconds after simulation start

Process finished with exit code 0

Doubled the CPUs to 8 machines of 32 CPUs. Reduced wait queue max size to 36,097 jobs.



```
#include <vector>
#include "VM.h"
#include "MyQueue.h"
#include <chrono>
#include <random>

#define Q_SEQ_FILE "C:\\Users\\-\\CLionProjects\\DataStruct-HW03\\QueueSeq.txt"
#define DONE_QUEUE_FILE "C:\\Users\\-\\CLionProjects\\DataStruct-HW03\\DoneQueue.txt"

#define MACHINES 8
#define CPUS_PER_MACHINE 32

#define REQUESTED_NUM_JOBS 100000

#define MIN_ARRIV_DELAY_ns 1000
#define MAX_ARRIV_DELAY_ns 1000

#define MIN_JOB_LEN_ns 4000000
#define MAX_JOB_LEN_ns 4000000

#define TIME_NOW chrono::system_clock::now()

using namespace std;

class Sim {
public:
    // ...
}
```

SIMULATION STATISTICS

Total Number of jobs processed: 100000
Average Run Time of jobs processed: 4855 microseconds
Average Wait Time of jobs processed: 331197 microseconds
Average Turnaround Time of jobs processed: 336052 microseconds

Minimum job run time processed: 3998 microseconds
Maximum job run time processed: 10710 microseconds

MAXIMUM QUEUE LENGTH EVENT

Max number of jobs in queue at one time: 36097
Happened at: 1551489468706826 us since Epoch
Or 1185047 microseconds after simulation start

Process finished with exit code 0

Graph of queue size (building up to 36,097) vs time using file output of QueueSequence.csv .

