

## README – Event Driven Simulation (Bank Queuing)

Project Title: Event Driven Simulation – Bank Queuing

### 1. Introduction

This project simulates customers arriving at a bank and being served by multiple tellers. The simulation compares two queuing methods:

Single Queue: All tellers share one common line.

Multiple Queues: Each teller has a separate queue, and customers join the shortest one.

The program is event-driven, meaning events (customer arrivals, service completions, teller idle times) are scheduled and processed in order using a linked list-based event queue. Function pointers are used for event handling and logged to show execution.

### 2. Objectives

Implement an event-driven simulation using structures and function pointers.

Simulate customer-teller interactions for both single and multiple queues.

Compare performance metrics such as average waiting time, maximum wait, and teller utilization.

Generate a Gnuplot graph of Average Time vs Number of Tellers.

### 3. Folder Structure

```
bank-simulator2/  
├── bin/                # Compiled executable (qSim)  
├── include/            # Header files (.h)  
├── src/                # Source files (.c)  
├── output/            # Output data and logs  
├── Makefile  
├── gnudata.txt         # Data for plotting  
└── README.pdf         # This file (final submission)
```

### 4. Build Instruction:

```
gcc -Iinclude -Wall src/qSim.c src/event.c src/queue.c -o bin/qSim
```

### 5. Running the Program

Command format:

```
./bin/qSim <#customers> <#tellers> <simulationTime> <averageServiceTime>
```

Example:

```
./bin/qSim 100 4 60 2.3
```

This runs both the single-queue and multiple-queue modes and prints results in the terminal.

## 6. Test Cases

Run three test cases and save outputs in output/ folder.

Test 1

```
./bin/qSim 100 2 60 2.3 > output/test1.txt
```

Test 2

```
./bin/qSim 150 3 60 2.5 > output/test2.txt
```

Test 3

```
./bin/qSim 200 4 60 3.0 > output/test3.txt
```

## 7. Sample Results

Mode	Avg Time (min)	Max Time (min)	Teller Utilization
Single Queue	3.43	7.60	Teller0=55.41, Teller1=57.76, Teller2=53.36, Teller3=52.58
Multiple Queues	4.21	13.08	Teller0=54.96, Teller1=56.49, Teller2=53.33, Teller3=47.30

## 8. Analysis

Single Queue performs better in average and maximum waiting times because all tellers share the load equally.

Multiple Queues can cause longer waits for unlucky customers who join slower tellers.

Teller utilization is more balanced in the single-queue setup.

In real systems, multiple queues might be used for specialized tellers or customer preference, but for fairness and efficiency, a single queue is generally better.

## 9. Gnuplot Visualization

Create a simple data file gnudata.txt in your project folder:

```
#Tellers AvgTime
1 10.0
```

```
2 6.5
3 4.8
4 3.4
5 3.0
6 2.8
```

Open Gnuplot and run:

```
cd 'C:/Users/KIIT/OneDrive/Desktop/bank-simulator2'
plot 'gnudata.txt' using 1:2 with linespoints title 'Avg Time vs #Tellers'
```

To save as PNG:

```
set terminal pngcairo size 800,600
set output 'avg_time.png'
replot
set output
```

#### 10. Deliverables Checklist :

```
Source code files (src/, include/)
Makefile
Compiled executable (bin/qSim)
Output test files (output/test1.txt, etc.)
Gnuplot data file (gnudata.txt)
README.pdf
```

#### 11. Conclusion

This simulation successfully models bank queuing behavior under two service systems. Results clearly show the efficiency benefits of a single-queue approach, validated through event-driven logic, random arrivals, and function pointer-based event handling.