Bank Queue Simulation (Event-Driven)

Introduction

This project simulates customers arriving at a bank and being served by tellers using event-driven

simulation.

The program compares two queuing models:

1. Single queue for all tellers.

2. Separate queue per teller.

The goal is to measure waiting times, service times, idle times, and compare the efficiency of each

model.

Folder Structure

qSim/

src/ # source code (qSim.c)

include/ # header files (qSim.h)

bin/ # compiled binary (qSim)

output/ # simulation results

Makefile # build script

REaCompilation

To build the project, run:

make

This creates the executable:

bin/qSim

To clean up:

make clean

Running the Program

Usage:

./bin/qSim #customers #tellers simulationTime avgService

Example:

./bin/qSim 100 4 60 2.3

Test Cases

Three sample test cases were executed:

1. 100 customers, 2 tellers, 60 minutes, avg service 2.0

./bin/qSim 100 2 60 2.0 > output/test1.txt

2. 200 customers, 4 tellers, 120 minutes, avg service 3.0

./bin/qSim 200 4 120 3.0 > output/test2.txt

3. 500 customers, 6 tellers, 180 minutes, avg service 1.5

./bin/qSim 500 6 180 1.5 > output/test3.txt

Analysis

- Single Queue: Performs better when customer load is high because tellers stay busy and

customers are served in fair orde

- Multiple Queues: May cause imbalance (some tellers idle while others overloaded). Slightly faster

under very low load.

Conclusion:

- High load → Single queue is better.

- Low load → Multiple queues can be acceptable.

Problems Faced

- Linking error with -lm (solved by adding -lm at the end of gcc command).

- Folder structure setup confusion (solved by organizing into src, include, bin, output).

- Implementing function pointers in events was tricky but handled