ANSI C Post Office Simulation

Barnaby Gill

February 2021

1 Design Decisions

For this simulation, I knew I would have to use a data structure to store the instances of customers at the post office, along with their relevant data.
My first design choice was deciding which structure to use; I decided to implement a doubly linked list. I chose this so I could easily store several associated instances of customers, with links (pointers) from one to the next automatically assigning a priority, since newly added customers will be at the end of the list and those who have been waiting longest with be at the beginning.
I used two different structs for this system. Firstly, a struct I named 'QUEUE' (using typedef), which stored the pointers to the next and previous nodes, as well as a variable named data of type 'INFO', my second struct.
The INFO struct (also named using typedef) was used to store customer-specific information, including the unique ID of the customer as well as their current waiting time, patience (the maximum time they would be willing to wait), position in the queue, time their task will take to complete, and a string containing their current status in text. (This string is not output in the current program but is still saved in case individual customer information is required.)

The post office was to be implemented in a way where several things would happen in each of a number of time intervals. The actions that occur each time interval are as follows:

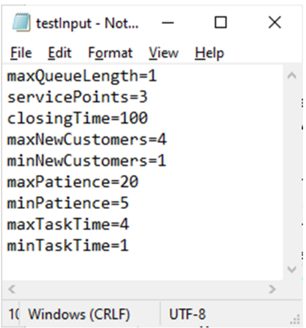
- Zero or more fulfilled customers have finished being served and leave the system.
- A service point that is not serving a customer will start to serve the next available customer, and that customer will leave the queue.
- If a customer in the queue has reached their waiting tolerance limit, then they leave the queue and become a timed-out customer.
- Zero or more customers arrive into the system, and either join the back of the queue, or, if the queue is full, leave the system as an unfulfilled customer.

I decided the best representation for a real system would come with each time interval representing five minutes in the real world.
Assuming the regular business is open from 9am-5pm, this is 8 hours which is 96 intervals. I thought it was fair to round this to 100 intervals representing one full day, and allowed the user to enter the input parameter of 'closing time' which was a count of the total intervals to simulate.

Finally, I made the decision not to remove anyone from my queue struct until the end of each simulation, even if they finished their business in the post office simulated queue. Instead, I assigned each customer a specific position relating to their status. For example, fulfilled customers have the position code of 999 and timed-out customers had 404, where customers in the queue just had their normal queue position from 1 to maxQueueLength.
I made this decision to firstly make it much easier to determine the number of each type of customer in the simulation. The queue can be traversed and the positional code of each customer can let it be added to a tally of customers of that type. Additionally, I kept the customers in the queue should there be a need to inspect a specific customer rather than just the overall standings, since customers who have finished their business in the post office still have their data stored until the end of the current simulation.

2 Simulated Experiment

The experiment I chose to perform using my simulation was to find the shortest required queue length to be used where running a simulation over 1000 iterations would consistently result in less than 5% of the customers left unfulfilled. I also attempted to identify the optimum queue length based on the number of customers unfulfilled and the number timed-out in the process. An unfulfilled customer is one that wanted to perform a task but was turned away due to the queue being at max capacity, and a timed-out customer had to wait in the queue for so long they became impatient and left.
The choice I made for the other input categories selectable by the user are as follows:



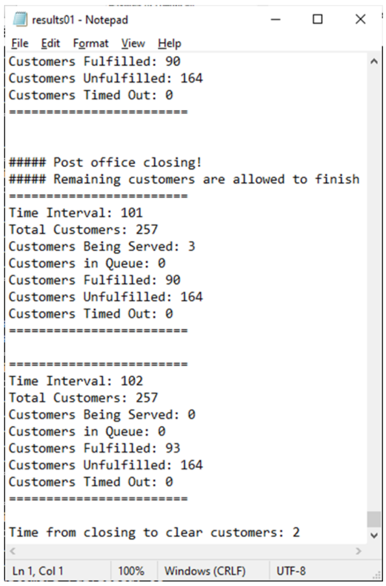
testInput - Not...

File Edit Format View Help

```
maxQueueLength=1
servicePoints=3
closingTime=100
maxNewCustomers=4
minNewCustomers=1
maxPatience=20
minPatience=5
maxTaskTime=4
minTaskTime=1
```

1(Windows (CRLF) UTF-8

As shown in the screenshot, the initial maximum length of the queue is 1.
The number of available service points is 3, and the minimum and maximum values for customer data are also displayed, with the actual value being a random selection from this range.



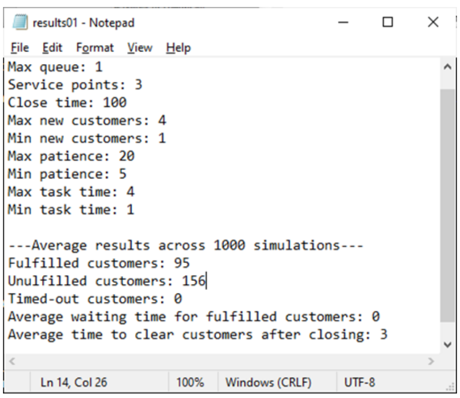
results01 - Notepad

File Edit Format View Help

```
Customers Fulfilled: 90
Customers Unfulfilled: 164
Customers Timed Out: 0
======================


##### Post office closing!
##### Remaining customers are allowed to finish
======================
Time Interval: 101
Total Customers: 257
Customers Being Served: 3
Customers in Queue: 0
Customers Fulfilled: 90
Customers Unfulfilled: 164
Customers Timed Out: 0
======================

======================
Time Interval: 102
Total Customers: 257
Customers Being Served: 0
Customers in Queue: 0
Customers Fulfilled: 93
Customers Unfulfilled: 164
Customers Timed Out: 0
======================

Time from closing to clear customers: 2
```

Ln 1, Col 1 100% Windows (CRLF) UTF-8

I decided to first run this test with a single simulation and see the results.
As expected with a very low queue length, the majority of customers were turned away unfulfilled. 93 customers completed their tasks, and 164 could not join the queue due to it being full. The time intervals previous to 100 can be seen in the results file when scrolled above the displayed screenshot.
Also, 0 customers timed-out in this simulation. Due to the queue being so short, customers never had to wait behind other customers and so never reached their patience tolerance level, although this is clearly still a bad queue model and unrealistic simulation of real life.



results01 - Notepad

File Edit Format View Help

```
Max queue: 1
Service points: 3
Close time: 100
Max new customers: 4
Min new customers: 1
Max patience: 20
Min patience: 5
Max task time: 4
Min task time: 1

---Average results across 1000 simulations---
Fulfilled customers: 95
Unulfilled customers: 156
Timed-out customers: 0
Average waiting time for fulfilled customers: 0
Average time to clear customers after closing: 3
```

Ln 14, Col 26 100% Windows (CRLF) UTF-8

A test of the same system and queue length calculating the average result over 1000 simulations gives a similar result. This test gives a count of 156/251 people unfulfilled, a proportion of 62%. To reach the required limit of 5% unfulfilled, there must be 12 or less unfulfilled customers.
The next step was to gradually increase the length of the queue and continually run the simulation, checking the number of unfulfilled customers each time. I chose to rely on the averages generated from 1000 simulations for this test to avoid outliers affecting the results.

I changed the maximum queue length to 5, leaving the other inputs the same, and ran 1000 simulations on these settings. With the increase in queue size, the average results made changes as expected. Firstly, the number of fulfilled customers compared to unfulfilled is now almost even, as opposed to the large majority being unfulfilled in the first test. Also, while the number of timed-out customers remains at 0, the average time to clear the customers from the post office after closing time has increased from 3 to 7. This makes sense considering there is likely to be more customers between the service points and queue at any given time if there are more spots in the queue, and so these people would take longer to leave.

```
Max queue: 5
Service points: 3
Close time: 100
Max new customers: 4
Min new customers: 1
Max patience: 20
Min patience: 5
Max task time: 4
Min task time: 1

---Average results across 1000 simulations---
Fulfilled customers: 125
Unfulfilled customers: 126
Timed-out customers: 0
Average waiting time for fulfilled customers: 1
Average time to clear customers after closing: 7
```

```
Max queue: 10
Service points: 3
Close time: 100
Max new customers: 4
Min new customers: 1
Max patience: 20
Min patience: 5
Max task time: 4
Min task time: 1

---Average results across 1000 simulations---
Fulfilled customers: 129
Unfulfilled customers: 106
Timed-out customers: 15
Average waiting time for fulfilled customers: 2
Average time to clear customers after closing: 10
```

Increasing the maximum queue length from 5 to 10 made less drastic changes than 1 to 5, but you can see the trend is the same. The number of fulfilled customers increased, the number of unfulfilled customers decreased, and for the first time some customers timed-out and left the queue. In this set of simulations, the average number of customers this happened to was 15.

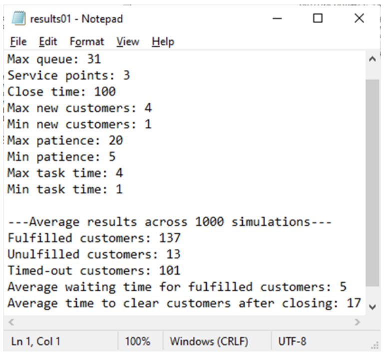
With a maximum queue of 20 customers, the number of unfulfilled customers dropped significantly to 58, and the number of timed out customers rose again, to just passed that at 59. The average wait time of fulfilled customers increased from 2 to 4 and the time to clear customers increased too, although these are not strictly negatives as the proportion of fulfilled customers increased as well.

```
Max queue: 20
Service points: 3
Close time: 100
Max new customers: 4
Min new customers: 1
Max patience: 20
Min patience: 5
Max task time: 4
Min task time: 1

---Average results across 1000 simulations---
Fulfilled customers: 134
Unfulfilled customers: 58
Timed-out customers: 59
Average waiting time for fulfilled customers: 4
Average time to clear customers after closing: 14
```

```
Max queue: 30
Service points: 3
Close time: 100
Max new customers: 4
Min new customers: 1
Max patience: 20
Min patience: 5
Max task time: 4
Min task time: 1

---Average results across 1000 simulations---
Fulfilled customers: 137
Unfulfilled customers: 17
Timed-out customers: 97
Average waiting time for fulfilled customers: 5
Average time to clear customers after closing: 17
```

At a max queue length of 30 people, the unfulfilled counter averages at a low 17/251. This is approximately 7%, but it isn't quite under the 5% target.
In this scenario however, even with a significantly longer queue, the number of fulfilled customers has only marginally increased from 134 to 137 per simulation, but the count of timed-out customers rose to 97, almost 40% of the test customers. The average wait time and clear time after closing have also increased.

The average unfulfilled customers with a max queue of 35 was 5, equivalent to a very low 2% of the customers. The lowest maximum queue length appeared to be between 30 and 35.
A queue of 31 people gave an average count of 13 people unfulfilled, putting it just above 5%.
A queue of 32, which gives an average of 11 unfulfilled customers is therefore the lowest value under 5%, sitting at 4.38% to be exact.

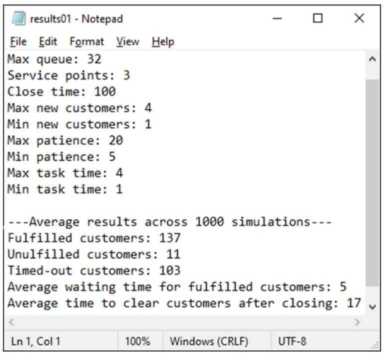


results01 - Notepad

File Edit Format View Help

```
Max queue: 31
Service points: 3
Close time: 100
Max new customers: 4
Min new customers: 1
Max patience: 20
Min patience: 5
Max task time: 4
Min task time: 1

---Average results across 1000 simulations---
Fulfilled customers: 137
Unfulfilled customers: 13
Timed-out customers: 101
Average waiting time for fulfilled customers: 5
Average time to clear customers after closing: 17
```

Ln 1, Col 1 100% Windows (CRLF) UTF-8

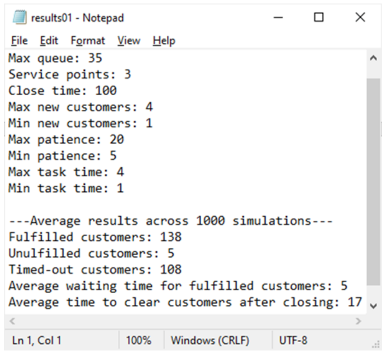


results01 - Notepad

File Edit Format View Help

```
Max queue: 32
Service points: 3
Close time: 100
Max new customers: 4
Min new customers: 1
Max patience: 20
Min patience: 5
Max task time: 4
Min task time: 1

---Average results across 1000 simulations---
Fulfilled customers: 137
Unfulfilled customers: 11
Timed-out customers: 103
Average waiting time for fulfilled customers: 5
Average time to clear customers after closing: 17
```

Ln 1, Col 1 100% Windows (CRLF) UTF-8



results01 - Notepad

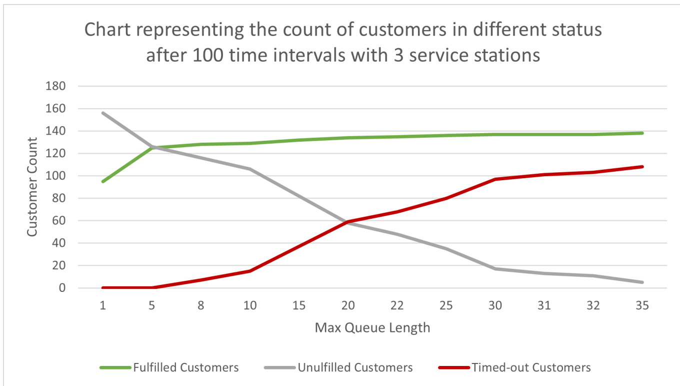
File Edit Format View Help

```
Max queue: 35
Service points: 3
Close time: 100
Max new customers: 4
Min new customers: 1
Max patience: 20
Min patience: 5
Max task time: 4
Min task time: 1

---Average results across 1000 simulations---
Fulfilled customers: 138
Unfulfilled customers: 5
Timed-out customers: 108
Average waiting time for fulfilled customers: 5
Average time to clear customers after closing: 17
```

Ln 1, Col 1 100% Windows (CRLF) UTF-8

As demonstrated by the above tests, the answer to my experiment - "Finding is the shortest maximum queue length which consistently gives a rate of unfulfillment less than 5%" is 32 given the other inputs I chose.
To better decide what I thought the 'ideal' maximum queue length would be based on this experiment, I plotted the results shown as well as data from some other queue lengths on a graph.

Max Queue Length	Fulfilled Customers	Unfulfilled Customers	Timed-out Customers
1	95	156	0
5	125	126	0
8	128	116	7
10	129	106	15
15	132	82	37
20	134	58	59
22	135	48	68
25	136	35	80
30	137	17	97
31	137	13	101
32	137	11	103
35	138	5	108



Technically, if the only point to consider is how many customers are fulfilled, then the longer the maximum queue length the better. However, as shown in this experiment, there are diminishing gains in extending the queue past a length of between 5 and 10. If this simulation is to be realistic then it is unreasonable to have to allocate space for a 30 person queue for 137/251 fulfilled customers when a 10 person queue still has 129/251.
Even running the simulation with an input parameter of -1 for the maximum queue length, meaning there is no maximum, only yields a rate of 138/251. This is the same return as with a queue of length 35.
I also decided a customer running out of patience and becoming 'timed-out' was less severe in a realistic simulation than a customer being unable to even queue due to it being at maximum capacity, and so was more accepting of this.
Considering these points I concluded that the optimal queue length limit, given the input parameters chosen for this experiment, was 8. A queue of length 8 has very few timed-out customers (7), and almost as many fulfilled customers as the much longer queue lengths. In addition, a length of 8 is very feasible for a post office queue implementation in the real world.