Assignment 2

CmpE 250, Data Structures and Algorithms, Fall 2017

Instructors: H. L. Akın, A. T. Cemgil
TA: Alper Alimoğlu, Özlem Şimşek
SA: Yusuf Hakan Kalaycı, Abdullatif Köksal

Due: 23 Nov. 2017, 23:59 Sharp

1 Discrete Event Simulation

You are a businessman who wants to build new Starbucks in Etiler. There are two types of employees in Starbucks, one of them is cashier who takes the order of the customer and the other one is barista who prepares the coffee for the customer.

There are two different models in Starbucks. Each model has N cashiers and N/3 baristas(N is divisible by 3 in examples). First model has one queue for cashiers and one queue for baristas(Fig. 1). The other model has one queue for cashiers and N/3 queues for baristas. First 3 cashiers send the orders to the first barista and next 3 cashiers send the orders to the second barista and goes on like that(Fig. 2). In both systems, cashier queue works by first come first serve design and barista queue works by most expensive coffee served first.

You are going to simulate these two different models and collect the following statistics for each model:

1. Total running time of the system

2. Utilization of each cashier and barista

3. Turnaround time of each order

4. Maximum length of each queue

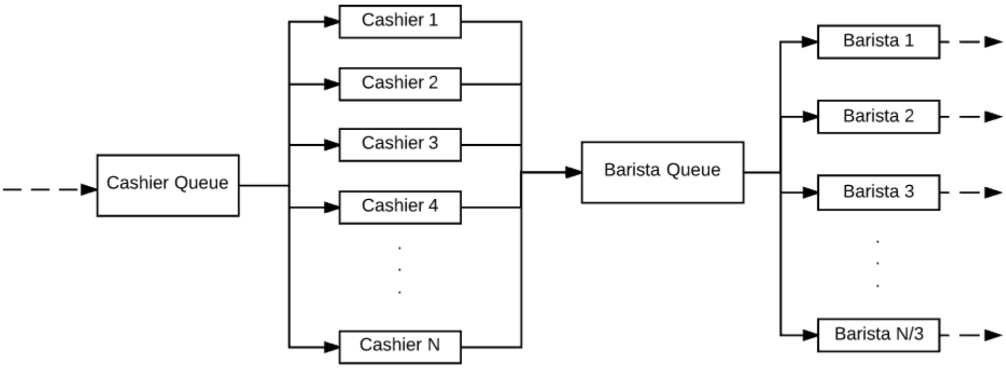

Figure 1: First Model

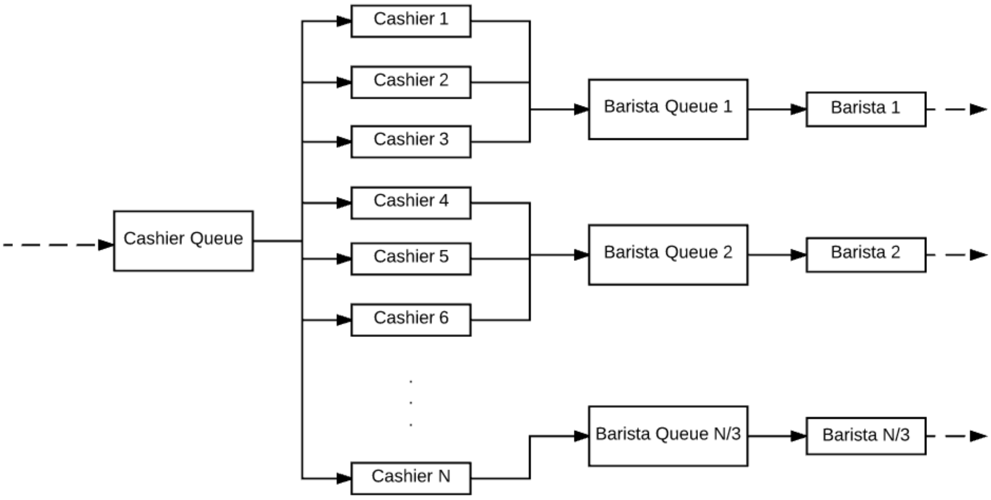

Figure 2: Second Model

2 Input\Output Format

The input file has the following format:

1. First line is number of cashiers (N) which is always divisible by 3.
2. Second line is the number of orders (M).
3. Following M lines contain information about orders. There are 4 variables separated with space. They are:
   (a) Arrival Time: The time when customer enters to the coffee shop. (in seconds)
   (b) Order Time: The time required to give coffee order. (in seconds)
   (c) Brew Time: The time required to make coffee. (in seconds)
   (d) Price of Order: Price of the coffee.

The output file has the following format:

1. First line is the total running time of the coffee shop. (in seconds)
2. Next line is the maximum length of the cashier queue.
3. (a) If the coffee shop has one barista queue, next line is maximum length of this queue.
   (b) If the coffee shop has N/3 barista queues, next N/3 lines are maximum lengths of each barista queue.
4. Next N lines are the utilization of the cashiers.
5. Next N/3 lines are the utilization of the baristas.
6. Next M lines are turnaround times of orders.

You will print these values for first and second models, respectively. You will put an empty line between models. Example of input and output files can be seen in the table below.

Input	Output
	37.52
	1
	5
	0.40
	0.46
	0.27
	0.15
	0.19
	0.21
	0.85
	0.82
	45.64
	134.31
	72.82
	38.66
	124.54
6	25.18
10	102.93
0 20.75 24.89 10.30	93.81
3.21 22.47 18.12 7.57	73.20
5.17 19.83 32.35 13.93	49.21
8.16 21.10 11.32 25.95	
9.54 26.05 17.14 8.15	191.42
10.32 11.10 14.08 3.74	1
16.47 34.68 41.41 16.98	4
23.13 17.33 16.22 9.86	2
27.52 40.68 23.43 20.97	0.29
28.08 16.80 30.47 11.32	0.33
	0.19
	0.11
	0.14
	0.15
	0.89
	0.31
	45.64
	188.21
	72.82
	38.66
	70.64
	25.18
	126.36
	39.91
	73.90
	145.22

3 Implementation Details

1. Whenever a cashier or barista finishes his/her job, he/she immediately fetches an order waiting in corresponding queue. If the queue is empty, he/she goes idle.

2. If more than one cashier is available, the cashier with smallest ID takes the order. For example, if cashier1 and cashier3 are available at the same time, the next order will be taken by cashier1. This principle is same with baristas, too.

3. If any statistics have more than 2 decimal places, you need to write the number to the output file with 2 decimal values with rounding. You will use even rounding.(You can use printf("%.2lf", variable_name) to print double variables.)

4. If you can't implement a certain statistic value, you should write "-1" in place of that statistics to the output file. This way you can implement other statistics and get points.

5. $$\text{Unit Utilization} = \frac{\text{Busy Time of the Unit}}{\text{Total Running Time of The Coffee Shop}}$$

   $$\text{Turnaround Time} = \text{Time when the coffee is done} - \text{Arrival time of the customer}$$

   $$\text{Total Running Time} = \text{Time when all coffee orders are done}$$

6. Your program will be compiled with **cmake CMakeLists.txt && make** command. Therefore, if you add new files, you have to check CMakeLists.txt is updated accordingly so that your code auto-compiles. Note that it is also fine to code in a single file in this project (Because I am assuming you already know what .h and .cpp files are and how to use those)

7. I will execute your program with **./project2 inputFile outputFile** command. So, use command line arguments in your main function accordingly.

4 Project Guidelines

1. Warning: All source codes are checked automatically for similarity with other submissions and also the submissions from previous years. Make sure you write and submit your own code.

2. Your program will be graded based the correctness of your output and the clarity of the source code. Correctness of your output will be tested automatically so make sure you stick with the format described above.

3. There are several issues that makes a code piece 'quality'. In our case, you are expected to use C++ as powerful, steady and flexible as possible. Use mechanisms that affects these issues positively.

4. Make sure you document your code with necessary inline comments, and use meaningful variable names. Do not over-comment, or make your variable names unnecessarily long.

5. Try to write as efficient (both in terms of space and time) as possible. Informally speaking, try to make sure that your program completes in meaningful amount of time.