

CSCI203/CSCI803 ASSIGNMENT 2 Due 23:55 21/09/2018

Your task for this assignment is to investigate some of the properties of queues.

You should write a program which uses discrete, event based, simulation to model the queuing and service of a set of requests.

Input consists of the following data:

- The number of primary servers in the system.
- The number of secondary servers in the system.
- A set of service requests each consisting of an arrival time and two service times in the order primary followed by secondary. This set is terminated by a dummy record with arrival time and service times all equal to 0. (Note: the arrival times are sorted in ascending order).

Your program should read the name of the data file from standard input and then read the data in the named file into the simulation as needed.

The simulation is to be of a system with two sets of servers, primary and secondary, each set of servers associated with a queue. Customers arrive in the system and are served first by a primary server and, on completion of this service, by a secondary server. If all servers of a particular type are busy, the customer will enter either the primary or secondary queue as appropriate.

The simulation should be run until the last customer has left the system.

Output, to standard output will consist of the following data:

- Number of people served.
- Time last service request is completed.
- Average total service time.
- Average total time in queue(s). Both overall and separate.
- Average length of queue. For each queue and overall.
- Maximum Length of queue. For each queue and overall.
- Total idle time for each server.

Notes:

- The simulation starts at time=0, not when the first customer arrives.
- You may assume that there are no more than 10 servers of each type, 20 total.
- You should attempt to use dynamic data sparingly or not at all.
- Your submission will be compiled and run using a script. Because of this the correct file name for your source code is essential. Check your submission receipt.
- The sample data provided for testing will not be used for marking your program, a second data file will be used for this purpose.

Programs must compile and run under gcc (C programs), g++ (C++ programs) java or python. Programs which do not compile and run will receive no marks.

Programs should be appropriately documented with comments.

All coding must be your own work. Standard libraries of data structures and algorithms such as STL may not be used, nor may code be sourced from textbooks, the internet, etc.

In addition to the code, a pdf file describing your solution should also be produced. This file should contain at least:

1. A high-level description of the overall solution strategy:
2. A list of all of the data structures used, where they are used and the reasons for their choice.
3. A list of any standard algorithms used, where they are used and why they are used.

Marking Guide:

- Your program will receive a maximum of 5 marks, the other 5 will be awarded for the documentation.
- Programs submitted must work! A program which fails, to compile or run will receive a mark of zero for the programming component.
- A program which produces the correct output, no matter how inefficient the code, will receive a minimum of 50% for the programming component.
- Additional marks beyond this will be awarded for the appropriateness, i.e. efficiency for this problem, of the algorithms and data structures you use.
- Programs which lack clarity, both in code and documentation, will lose marks.
- If you have used a data structure or algorithm which you know is inefficient then you should note this in the documentation pdf and indicate what you feel might be better.

Submission:

Assignments should be typed into a single text file called `ass2.ext` where *ext* is the appropriate file extension for the chosen language.

Both files should be submitted via the submit program.

```
submit -u user -c csci203 -a 2 ass2.ext ass2.pdf
```

where your unix userid should appear instead of *user*.