SCSSE, University of Wollongong
CSCI203 Data Structures and Algorithms
Spring 2016

Assignment 2 (Due 11:59 PM Fri 9th September)

Objectives

Design and implement a discrete event simulation
Choose and implement appropriate data structures
Adapt standard algorithms to a specific problem

Task

You should write a single program which simulates the queuing and service of customers under two different queuing strategies and compare the results.

Program

Your program should run two **discrete event** simulations, one for each strategy. Each simulation should start at time 0 and run until all customers have been served.

The two queueing strategies are as follows:

1. Use a single queue from which each server will take the next customer as soon as the server becomes available.

2. Use a queue for each server. Customers will choose the server with the shortest queue on arrival and will not move from queue to queue.

Your program should be readable and well commented.

Data Structures and Algorithms

Part of the purpose of this subject is to gain an in-depth understanding of data structures and algorithms. As such, all programming tasks in this subject require you to *choose appropriate data structures and algorithms*, and to *implement them yourself*. You may not take advantage of any built in data structures more complex than an array or list, but instead should provide your own implementation. If you use any references other than the lecture notes, ensure you cite them or you may receive 0 for plagiarism. A clear comment in your code is sufficient.

Readme

Write a text file named readme. Include clear instructions on how to compile and run your algorithm. Ensure that your program compiles and runs correctly on ubuntu, in the lab. If your program does not compile, it will receive 0 marks. If it doesn't run according to the specification, you will receive very few marks. You may also include a makefile if you wish.

Analysis

Once your program is complete, you should run it for some sample scanarios and examine the results. Write a file named analysis.pdf containing a discussion the differences you observe in statistics between the two queueing strategies. Which strategy is most efficient?

Input

Both simulations will use identical inputs. Your program should read a **file name** from **standard input**, and run each simulation using the named file as the input file.

The input file has the following format:

- The number of servers.
- A set of lines each consisting of a customer arrival time followed by the corresponding service time.

Times are in seconds, from the start of the simulation. Although a sample input file has been provided, your program should still run sucessfully substantially larger inputs.

Output

Output should be to standard output.

For each of the two simulations the output should consist of the following data, clearly labelled:

- Number of people served.
- Time that the last customer finished being served (total simulation time).
- Average service time (note this should depend only on the input file).
- Average time a customer spends in queue.
- Maximum time a customer spends in queue.
- Average length of queue in any given second. For each queue and in total for the multi-queue version of the simulation.
- Maximum Length of queue. For each queue and in total for the multi-queue version of the simulation.
- Total idle time for each server.

Submission instructions

Your completed assignment should contain your source code, readme and analysis.pdf in a zipped folder named ass2.zip. To submit your assignment transfer your ass2.zip file to banshee, and run the command:

```
turnin -c csci203 -a 2 ass2.zip
```

from the directory containing your file.

1. Late submissions will be marked with a 25% deduction for each day.
2. Submissions more than three days late will not be marked, unless an extension has been granted.
3. If you need an extension apply through SOLS, if possible **before** the assignment deadline.
4. Plagiarism is treated seriously. If we suspect any work is copied, all students involved are likely to receive zero for the entire assignment.