

Programming 3 (P3): Customer Service Center

Instructor: Dr. Shengquan Wang

Due Time: 10PM, 4/2/2023

In this programming assignment, you will use `semaphore` together with `pthread` to implement a customer service center problem.

1 Instruction

`semaphores` represent resources which are being competed for: 4 waiting chairs, a semaphore to decide to wait or leave, etc.

A customer service center consists of an awaiting room with 4 awaiting chairs, 2 assistants, and 2 service chair (a customer will be served when he/she sits on this chair). If there are no customers sitting on any awaiting chair, the assistant who is not busy should go to sleep. If a customer enters the center and all chairs are occupied, then the customer leaves the center. If all assistants are busy but chairs are available, then the customer sits in one of the free chairs. If one or all assistants is/are asleep, the customer wakes up one of the assistants to receive the service. We assume that all customers are receiving the service in FCFS order. Use semaphore to coordinate the assistants and the customers.

2 Testing

In the main thread, you need to create one thread for each customer and one thread for each assistant. The arrival time and service time for all customers are showed as follows:

cid	arrival_time	service_time
1	3	15
2	7	10
3	8	8
4	9	5
5	11	12
6	12	4
7	14	8
8	16	14
9	19	7
10	22	2
11	34	9
12	39	3

Associated with every customer is a unique id (`cid`) as the customer ID, the arrival time (`arrival_time`) in seconds, and the service time (`service_time`) in seconds. Make sure that your simulation outputs information that clearly shows that your solution works. In particular, messages should be printed at the following times:

- Whenever a customer arrives at the custom service center;

- Whenever a customer leaves the custom service center without receiving any service;
- Whenever a customer starts to receive service;
- Whenever a customer is done with the service.

This is the right output:

```

Time 3: Customer 1 arrives
Time 3: Customer 1      starts
Time 7: Customer 2 arrives
Time 7: Customer 2      starts
Time 8: Customer 3 arrives
Time 9: Customer 4 arrives
Time 11: Customer 5 arrives
Time 12: Customer 6 arrives
Time 14: Customer 7 arrives
Time 14: Customer 7      leaves
Time 16: Customer 8 arrives
Time 16: Customer 8      leaves
Time 17: Customer 2      done
Time 17: Customer 3      starts
Time 18: Customer 1      done
Time 18: Customer 4      starts
Time 19: Customer 9 arrives
Time 22: Customer 10 arrives
Time 23: Customer 4      done
Time 23: Customer 5      starts
Time 25: Customer 3      done
Time 25: Customer 6      starts
Time 29: Customer 6      done
Time 29: Customer 9      starts
Time 34: Customer 11 arrives
Time 35: Customer 5      done
Time 35: Customer 10     starts
Time 36: Customer 9      done
Time 36: Customer 11     starts
Time 37: Customer 10     done
Time 39: Customer 12 arrives
Time 39: Customer 12     starts
Time 42: Customer 12     done
Time 45: Customer 11     done

```

3 Coding and Hints

You are recommended to use threading programming in C/C++. The programming should be tested in the Ubuntu environment in your P1 and P2 as you ran all the simulations in homework assignments.

Use the sample code from <https://gist.github.com/tausen/4261887> as the basic framework.

You could use `usleep` for the interarrival time between different threads and assistants' sleep.¹ You could use the `Spin` function² to simulate the service time.

4 Submission

You are going to report the following things:

- (a) Describe in details how you implemented it in the report.
- (b) Write a detailed report with the screenshots of the testing results. Each screenshot should include your username and the current time, which show that you did it by yourself. **If your output is different from the expected one, provide a reason for the cause.**
- (c) Specify the contribution made by each member if you work as a group.

The report should be written in a “.docx”, “.doc”, or “.pdf” format. All source codes should be well formatted with good comments. Submit the report and the source code to Programming P3 on Canvas. Any compression file format such as `.zip` is not permitted.

¹<http://linux.die.net/man/3/usleep>

²Download this: <http://pages.cs.wisc.edu/~remzi/OSTEP/Code/code.intro.tgz>. You can find the `Spin` function in `common.h`.