**Programming Assignment 2 – Multithread Programming**

Jefferson Parker

|  |
| --- |
|  |

**Instructions:**

* Please name your file as CS575\_yourusername\_p2.doc Please also update the filename in the header.
* All questions must be answered on an INDIVIDUAL basis. If your answer is inspired by the discussion with other students, you need mention their names in your acknowledgement section.
* State clearly all your assumptions if something seem ambiguous.
* Always ask if you have any questions about the following problems or stuck on any of them.
* Please submit your homework through blackboard on the due date.
* Thank You!

**Multithread Application:**

Consider a pizza cafe where there are only N seats. When a customer enters the store, (s)he buys a pizza and grabs a seat to eat. When the customer finishes eating, (s)he leaves the store. If there is any unsold pizza left, (s)he can get it immediately. Otherwise, (s)he need wait for the chef to make one. A chef in the kitchen just keeps making pizzas until there are M unsold pizzas left in the kitchen. (S)he resumes the work when a customer comes to buy one.

After the customer gets the pizza, (s)he tries to find a seat. If there is any available seat, (s)he grabs the seat and eats the pizza. Otherwise, (s)he waits until another customer finishes eating and leaves the café..

You are asked to simulate this problem as following:

1. Your program should first ask user to input M (the number of plates to hold unsold pizza) and N (the number of seats). You can also pass these two numbers through command line parameters. The maximum number should be 10 for both parameters. (10%)
2. You need create two types of threads besides the main thread: chef and customer. There is only one chef thread but multiple customer threads. The number of customers should be randomly generated. Assuming the maximum number is 20. (For example, you can use random()/srandom() or drand48()/srand48() to generate a random number in C) (10%)
3. Suppose each customer only buys one pizza. Their eating time is randomized between 1 to 10 time units. Assume it takes the chef 5 time units to make a pizza and the chef only makes enough pizzas for all customers. Implement the chef and the customer threads. The program should stop when all customers finish eating their pizzas, and no pizza should be left in the kitchen at the end. (You can define a time unit by yourself, e.g. a second, or a microsecond, or a fixed number of loops) (30%)
4. Synchronize the chef and customers so that at any time, there are no more than M unsold pizza and no more than N customers sitting and eating the pizza. (50%)
5. You can choose any programming language (e.g. C/C++/Java/python) to implement this program. You can find some sample C and Java code on the blackboard.
6. Your output may look like the following:

*Please input M: 5*

*Please input N: 8*

*The total number of customers is 17*

*Create a chef thread*

*Create 17 customer threads*

*Customer 1 arrives  
Chef makes pizza 1. 1 pizza unsold*

*Chef makes pizza 2. 2 pizzas unsold.*

*Customer 1 buys a pizza. 1 pizza unsold.*

*Customer 2 arrives*

*Customer 2 buys a pizza, 0 pizza unsold.*

*Customer1 gets a seat, 7 seats left.*

*Chef makes pizza 3. 1 pizza unsold.*

*Customer 1 finishes eating and leaves. 8 seats left.*

*Chef makes pizza 4. 2 pizzas unsold.*

*Customer 3 arrives*

*Customer 3 buys a pizza, 1 pizza unsold.*

*....*

**What to submit:** please submit all source code files, and a simple readme file to describe how to compile your code and also include some sample output. Please zip all files into a single zip file and submit it through the blackboard.

**Homework Feedback (1%)**

1. How long did you take to complete this homework? Is it too hard, too easy or OK?

It took about five hours to do this homework. It was the right level of difficulty.

1. Are the lectures clear? Does this homework help you learn this topic? How well do you understand this topic?

The lectures were clear, but this assignment really helped solidify my understanding of synchronization and locks.

1. Do you have any other feedback?

No.