

CMPE142 : Operating Systems

Assignment 4: Concurrency

Version 2

Due: Before the final exam

In this assignment you will use concurrency primitives to implement a simple producer/consumer system. A producer/consumer system consists of multiple threads “producing” something and multiple threads “consuming” the things produced.

The scenario in this assignment is as follows:

- There is a fast food burger joint with one chef and three infinitely hungry customers
- In order for a customer to eat, he/she must assemble three meal components:
 - Hamburger
 - Fries
 - Soda
- The chef produces two random meal components at a time, then goes to sleep until a customer tells him to make something else

Customer #1 brings their own infinite supply of hamburgers, and are only waiting for fries and soda.
Customer #2 brings their own infinite supply of fries, and are only waiting for hamburgers and soda.
Customer #3 brings their own infinite supply of soda, and are only waiting for hamburgers and fries.

The scenario begins with the chef producing two random meal components. After producing the items, the chef sleeps. This means two of the customers will (eventually) be able to eat. Once the second customer has taken his/her item, they ring a bell at the counter. That wakes up the chef who then produces two other random items, then goes back to sleep. The chef cannot produce duplicate items (eg, two sodas). After consuming the items, the customers wait until they are presented with another opportunity to eat. Only after both items are eaten should the chef be awakened.

Your assignment is to implement the above scenario, using 4 threads (1 chef thread and 3 customer threads). You must use proper concurrency primitives to ensure that the items produced are properly distributed and that no deadlocks occur.

Run the restaurant scenario 100 times in your program (eg, the chef produces 200 total items). For each customer, keep track of how many times that customer got to eat. Print these statistics at the end of the run.

You may work in teams of up to three students. Each team member can receive up to 20 points. It is expected that groups of more than one student will find an equitable way to distribute the work outlined in this assignment. Groups of fewer than three people will still be expected to do the same assignment (eg, no skipping parts if you are working solo).

Prerequisites

- You will need a Linux development environment. We outlined how to create this in class.
- A github repository to store your code (see below)

The Assignment

Start by creating a new github repository for your code. You can sign up for a free github account online if you don't have one already. Each team member should have their own account, but it does not matter which account owns the repository for this assignment. The repository can be either public or private.

Implement the restaurant scenario as described above. You may use whatever programming language you wish, provided said language has the proper concurrency primitives to ensure correct operation.

Teams are **strongly** urged to “commit early, commit often” - this means as soon as you have something working, push your changes to your team’s repository. This ensures that everyone has access to the latest code, and also serves as a backup in case your copy of the code gets messed up.

Submission and Grading

Your team will be awarded up to 15 points based on the correctness of the implementation. Each team member will be awarded up to 5 points based on the answers to the questions below.

On or before the due date, send answers to the questions below via E-Mail or Canvas, and grant me access to your team’s github repository (my github ID is ‘mlarkin2015’). **Each team member must answer the questions separately.**

The answers to the questions below shall be made via email to the email addresses listed in the class syllabus/green sheet, or via Canvas. DO NOT WAIT UNTIL LATE ON THE DUE DATE, as email server lags or delays may result in a late submission. This is one area that I am extremely picky with – even 1 second late will result in a zero score for that part of the assignment.

I will be comparing all submissions to ensure no collaboration has taken place. Make sure you do not copy another group's work. If you copy another group's work, members of both groups will receive an F in the class and be reported to the department chair for disciplinary action. If you are working in a group, make sure your partners do not copy another group's work without your knowledge, as all group members will be penalized if cheating is found.

Make sure your final changes are pushed to github before the class’ scheduled final exam date (as per the university catalog). I will be cloning your repository “as of” that time on that date. This means any changes made afterward will not be considered.

Questions

1. Provide the name of the github repository (including the owning account, eg “mlarkin2015/142-assignment-4”)
2. Provide instructions I need to follow to build your shell.
3. Comment on the distribution of how many times each customer got to eat. Was the distribution equal? Why, or why not?
4. For each member in your team, provide 1 paragraph detailing what parts of the lab that member implemented / researched. This may seem obvious but this is the way I use to ensure that each team member was actually contributing usefully to the team. If you worked on the assignment by yourself, you can just send me an email saying “I did the project by myself”, and that will be sufficient.