Experiment 4

C Programming Language--

Producer-Consumer Model Using Multiple Threads

By Jonathan Day

University of Southern Maine

Spring 2024

COS 350 System Programming

Contents

Problem

This project implements an advanced version of the classic Producer-Consumer problem, a fundamental concept in operating systems related to thread synchronization and inter-process communication. In this scenario, the challenge is expanded to include two producers and two consumers, each operating in their own threads. The producers generate mathematical expressions as their products, while the consumers solve these expressions. This setup serves as an excellent demonstration of managing access to a shared resource (a buffer) in a multi-threaded environment, ensuring data integrity and synchronization between threads.

Implementation

Provide detailed descriptions for your code, including the purpose of each function, the execution flow (how functions call each other, which can be illustrated with a flowchart), and any additional explanations necessary to make your code more readable and understandable.

Random Operator Function:

Called as a helper method by the Random Equation function, this method supplies a random operator(+,-,*,/) based on the corresponding random integer value(1,2,3,4).

Random Equation Function:

This function is called by the producer it generates 2 random values between 1 and 100. Then the random operator function is called. The values are combined and then returned.

Producer Function:

The producers call the Random Equation function and adds the results to the buffer.

Solve Equation Function:

This function is called by the consumer and returns the results of the equation.

Consumer Function:

The consumer calls the Solve Equation Function after popping an equation from the buffer and printing the results to the console. Lastly the memory allocation is freed.

Main Function:

The main function creates the threads and controls when the program ends.

Results

Include a screenshot of the execution results in this section, similar to the example provided in the problem statement document. Make sure the screenshot captures both the outcomes of the execution and the process you used to compile and run the code.

Conclusion

Did you encounter any challenges during the execution of this experiment? If yes, state the issue and explain how to solve it. If not, elaborate on the insights gained from this experiment.

Adapted to using multithreading in C was difficult. I've used the threading library in python as well as asynchronous functions which I thought was more straight forward but would provide less low level control. I also had an issue dividing by zero which I fixed by excluding zero the random integer values. My last issue was using floats which I could not figure out the so values were limited to integers only.