Greg MacGown, Tim Ryan and Josh Rosenthal

Term Project Progress Report

Timetable

11/11 - Progress report complete, presentation slides begun, paper outline complete,early draft begun but not significant enough to submit, see below
11/16 - Paper draft submitted to Dr. Teresco , presentation slides done, begin practicing
11/19 - Presentation thoroughly rehearsed, time to make last minute modifications on paper
11/21 - Final Paper submitted and Presentation 100% ready

Paper Outline and Sources

How Modern Machines Can Solve Synchronization Problems

Tim Ryan, Greg MacGown, Josh Rosenthal

Abstract

In this study, we have examined some classic synchronization problems (Namely the Dining Philosophers problem, the Bounded Buffer problem, the Readers and Writers problem, the Sleeping Barber problem, and the Cigarette Smokers problem) and researched some ways in which said problems can be solved. In our studies, we have found that a majority of the problems are solvable using semaphores; however, there are usually other variations of solutions that use mutex locks in addition. We then explain why these solutions are viable and necessary.

1. Overview

One of the largest issues in computing is the issues of synchronization: A lack of correct synchronization cause massive issues in a program, and it is extremely hard to consistently get the same errors as the timing of most synchronization issues is hard to predict due to the fact that the error can happen at a random interval based on how the resources are getting allocated. Our goal in this research paper is to find the solutions to a few famous (or infamous) synchronization problems, explore these solutions, and explain both why a solution that requires semaphores or mutex locks is necessary as well as demonstrate both how and why the solution given works.

2. Dining Philosophers

Main problem: If everyone picks up a single chopstick, then they all are waiting to pick up the second chopstick, resulting in deadlock

Resource Hierarchy solution references:
https://www.mathworks.com/help/simevents/ug/dining-philosophers-problem.html

Waiter solution references:
https://www.stolaf.edu/people/rab/pdc/text/dpsolns.html#:~:text=The%20waiter%20solution%20to%20Dining%20Philosophers&text=Strategy%3A,first%2C%20then%20their%20right%20chopstick.

Chandry/Missa solution references:
https://www.stolaf.edu/people/rab/pdc/text/dpsolns.html#:~:text=The%20waiter%20solution%20to%20Dining%20Philosophers&text=Strategy%3A,first%2C%20then%20their%20right%20chopstick.

Semaphores references:
https://www.geeksforgeeks.org/classical-problems-of-synchronization-with-semaphore-solution/

3. Bounded Buffer

Main issue: Both that the producer can produce while the place to put the thing it produced is already full and that the consumer can attempt to consume when the place to put produced things is empty

Semaphore references:
https://www.geeksforgeeks.org/producer-consumer-solution-using-semaphores-java/

4. Readers and Writers

Main problem: Writers can write over the file while readers are attempting to read

Semaphore references:
https://www.geeksforgeeks.org/readers-writers-problem-set-1-introduction-and-readers-preference-solution/

5. Cigarette Smokers

Main problem: If 2 smokers each grab 1 ingredient at the same time, it leads to deadlock

Semaphore references:
http://www.cs.umd.edu/~hollings/cs412/s96/synch/smokers.html

6. Sleeping Barber

Main issue: Multiple customers can try to sit in the same seat, as well as a customer might see the barber awake when they enter the shop but the barber finishes before they are seated and goes back to sleep because he doesn't see anyone sitting leaving both the barber and customer waiting for each other

Semaphore References:
https://www.geeksforgeeks.org/sleeping-barber-problem-in-process-synchronization/

7. Conclusion

Insert conclusion here