

Thomas Gorney
Prof. Zhang
Operating Systems
22 April 2019

Assignment 4 Report

Producer-Consumer Problem

This assignment is to create a program that solves the producer-consumer problem using pthreads API. The program creates multiple producer and consumer threads, synchronized and adds them to a buffer, as well as removes them. The items are removed in order of being added i.e. FIFO.

Submitted Files

main.cpp - main cpp file with everything in it to drive the program.

buffer.h – Defines buffer size and the buffer

Buffer.cpp – Includes functions for inserting, removing, and displaying the buffer

Makefile - makefile to make the file into an executable

Mytest - executable file for testing

Instructions

The program runs using arguments from the console. To run the program simply type in the console,

`./mytest arg1 arg2 arg3` **Example: “./mytest 5 3 5”**

arg1 – How long the main thread sleeps before terminating

arg2 – The number of producer threads

arg3 – The number of consumer threads

Results and Design Decisions

This is a solution for the producer-consumer problem using a pthreads with mutex locks and synchronization with semaphores. The results turn out similar to what I expect. The idea is that the semaphore protects the insertion or removal of items in the buffer. The producer and consumer move items to and from the buffer with separate threads that are synchronized.

Improvements

There are a few improvements that I would like to make in the future. First of all I would like to add in some error checking given more time. That way I can make sure the user only enters in 3

arguments and not less or more. Second, I would have liked to produce an “empty” message within the brackets when I am displaying the buffer so it is not just an empty pair of open brackets. Thirdly I would like to implement the bounded buffer as a monitor.