Final Project Report

TJ Maynes and Chris Migut

Abstract—The purpose of this project is to build an echo server with client using pthreads and semaphores.

Index Terms—Operating Systems, Shared Memory, Multithreading, pthreads, C, USF

1 Introduction

In this assignment we are supposed to learn how to use semaphores to protect a limited size resource. A circular buffer with 15 positions, each position storing one character, is used to communicate information between two threads (producer and consumer). The producer thread will read characters, one by one from a file and place it in the buffer and continue to do that until the end of file marker is reached. Make sure to place a one second sleep in the consumer thread between reads.

There should be no more than 150 characters in the mytest.dat file, when submitting the program. Also, the producer must inform the consumer when it has been finished processing the last character in the buffer. The producer could do this by placing a special character * in the shared memory buffer or by using a shared memory flag that the producer sets to true and the consumer reads at the appropriate time. The consumer thread will read each one of the characters (from shared memory buffer) and print it to screen.

1.1 Multithreading w/ pthreads

In specific situations, a programmer may want to write programs that do multiple tasks at once (as opposed to one at a time). For instance, a web server needs to be set up as a multithreaded application since incoming requests from client (to server) needs have its own thread to service the request. The benefits of multithreaded programming being responsiveness, resource sharing, economy, and scalability. Pthreads refers to the POSIX standard (IEEE 1003.1c) defining an API for thread creation and synchronization, which are found in the pthread.h library.

1.2 Client

1.3 Server

1.4 Analysis

References

[1] Silberschatz, Galvin, and Gagne *Operating System Concepts*, 8th ed. John Wiley and Sons.