Writeup 2

CS 444
Spring 2017

Abstract

This writeup is the writeup for Project 3 for CS444 Operating Systems 2. The write-up includes the Linux Yocto kernal on os-class.engr.oregonstate.edu block encryption device and also the concurrency problem of mutual exlucisve searchers/inserters/deleters.

Contents

1 Concurrency writeup

Concurrency: the problem was implemented using posix threads, which were subdivided into three categories, inserters, searchers, and deleters. The problem was solved by using 3 mutex, one for searchers, inserters, and deleters. As per the assignment, each thread would only try certain mutex. For example, since deleters can not run in parallel with another deleter, it would try the deleter mutex. Another case would be that inserters can proceed in parallel with any searcher but mutually exclusive with another inserter. Thus, inserters would not check searcher mutexes but would check an inserter mutex.

2 Block Driver Stuff

A simple block driver simulates a block device with a region of kernel memory. Essentially a very simple ramdisk driver. It starts by making the disk available to the system and setting up an internal data structure.

3 Questions

3.1 Main point of assignment

The main points of this assignment were to determine how the scheduler worked and how there are different methods of scheduling.The concurrency assignment was to help us remember how to use semaphores that we used only once or twice in CS344. In addition, we learned how to use some networking tactics in order to work with the VM and how to test the i/o using io generation scripts.

3.2 Personal approach to problem

The approach to the block device and encryption came from thorough research on block devices and some of the basics that were cover in tophat and expanded further from our own research. We used teh AES encryption which incorporated the cipher into the __init function. We then called the cipher to encrypt or decrypt in the transfer function.

3.3 Verification of solution

Verification of the encryption was if the output was not gibberish when the correct key was given, otherwise either there would be gibberish if there was an incorrect key or you would be unable to read the drive.
The important parts that needed to be verified were when the counter applied and which thread grabbed control of the mutex. Verificaiton of the solution helped when we added ID's to each of the threads using pthread self. After going through the output, it also was a big help when we printed out which threads were blocked. So we could ensure that the threads were working properly.

The makefile for this assignment performs this automatically if the VM is booted.

3.4 What We learned

We learned about using multiple mutexes to achieve selective exclusion, both from the assignment and the tophat reading. Using mutexes again helped this process to remember to not allow multiple threads to access the same data. The assignment helped remind us how semaphore works and some of the intricacies of i/o. It also taught about devices and drivers, and how a block layer works.

4 Version control log

4.1 Git Logs

35225a9 - Isaac Stallcup, 19 hours ago : added demo script. If this isn't good enough for Fuz literally nothing will be c799b8c - Isaac Stallcup, 20 hours ago : finished encrypted block driver with crypto! Now working on script be7a0bd - Isaac Stallcup, 24 hours ago : added in skeleton block driver file d5dc8a4 - Isaac Stallcup, 2 days ago : added in makefile to build module b84f54a - Austin, 4 days ago : added id's to threads using pthread_self 47d481c - Austin, 4 days ago : added thread blocked 698164f - Isaac Stallcup, 4 days ago : added in script to transfer flies to VM; usage is ./transfer_file.sh [file to transfer] [location on VM] 4f9bc5d - Austin, 5 days ago : optimized 12dc4f7 - Austin, 5 days ago : new version uploaded 68d74d4 - Austin, 5 days ago : test concurrency probably works b4050a0 - Austin, 5 days ago : Worked on concurrency 384d1b2 - Austin, 6 days ago : Started assignment3 concurrency 09edd54 - Austin, 6 days ago : Created project 3 folder

5 Work log

5.1 May 16-17

Austin created project 3 folder sometime. Worked on most of the concurrency

5.2 May 17-17

Austin finished concurrency problem with new specs

5.3 May 19-17

Alex and Isaac worked on block device. Isaac worked on scripts

5.4 May 21-17

Austin did final git log update and writeup.