Yoni Stern

Assignment 2 writeup

Structure:

Fifo: I gave each thread its own semaphore of size 0. In my code I put the bulk of the logic into an infinite loop. I started the loop by putting all the threads to sleep by calling sem_wait (on all besides for thread 0). Next I sent a request to the server along with changing the current file. Once that finished I checked if it was the last thread, if it wasn't I would call sem_post on the next thread semaphore to wake it up and have it do its work. After this I would finish off by receiving the information and printing to stdout. After this it would hit a barrier (to make sure all threads finished their work before restarting).

Concur: Concur threads do their work concurrently. I did this by having each thread run in an infinite loop. The loop would start by sending a request and then receiving the information. Afterwards it would switch the current file (each thread had their own file number which I would switch from 0 to 1 and 1 to 0). Last I added a barrier to make sure that all n threads completed their work before restarting and doing the work again.

Specifications:

The specifications in the assignment doc were pretty straightforward.

Bugs / Problems:

I debugged the code and there are no bugs that I know of.

Debugging

The way I debbugged my code was very similar to the first assignment. I used printf's to make sure the right code is running at the correct time. I also looked over my code to make sure the logic was correct.