CS570 Spring 2014 Assignment 1

This page last modified 4 Feb 2014

You shall implement a program that creates processes, threads, and uses a semaphore to manage a shared file

1. When your program starts, it will prompt the user to either run the program or exit
    1. If the user selects exit, then have your program gracefully terminate itself (using the exit() system call), your program is done.
    2. If the user elects to run, continue and do the following:
2. Your program shall create a new (child) process (using fork()); the parent process will wait until this child process completes all of it's work and then it will terminate itself.
3. The new child process shall do the following:
    1. Create a file, PTAB.txt, in the current directory (cwd).
    2. Write it's pid (followed by a Carriage Return and Newline) in the file.
    3. Close the file PTAB.txt
    4. Create a semaphore LOCK for the file PTAB.txt.
    5. Create 4 threads. Each thread shall have a unique identifier that you establish when you create the thread. Use the POSIX version of threads (i.e., pthread_create())
    6. Block/wait for all ten threads to complete their work
    7. Destroy the semaphore, then exit
4. Each thread shall perform the following (note, each thread is running concurrently):
    1. Once every second get the semaphore LOCK; once the thread has LOCK, it will proceed to do the following:
        1. open the file PTAB.txt and write the unique thread identifier (*threadid*) of that thread (followed by a Carriage Return and Newline)
        2. Write to the shell (print to stdout) "Thread <*threadid*> is running" followed by a newline
        3. Close the file PTAB.txt
        4. Release the semaphore LOCK
    2. Repeat 9 more times (each thread writes to the file and console a total of 10 times)
    3. exit

You will need to use the following POSIX system calls for creating and managing the semaphores with: sem_init(), sem_wait(), sem_post(), and sem_destroy().

I will test your program by compiling it and executing it on rohan. Your program shall be written such that it compiles and executes cleanly when using the cc, gcc, CC, or g++ compiler You shall create a sub-directory named "a1" in your home directory. In it, you shall place your source file(s), your header file, your Makefile (see Blackboard for examples on Makefiles), and a README file (see Blackboard for README requirements). Your source files SHALL CONTAIN sufficient comments for making the source easy to read. Points will be taken off for poorly (or non) commented source. Name the executable "sem".

- Create ~/a1 by hand.
- Create source files, an include file, a Makefile, and a README file. Put them into ~/a1.
- The Makefile shall create an executable named "sem" in this same directory (~/a1).
- Here is a nice overview of threads [http://www.llnl.gov/computing/tutorials/pthreads/#Overview]
- The system call "system()" will NOT be allowed

The assignment is due 1730 (5:30 PM) on Wednesday, 26 Feb 2014

TURNING IN YOUR WORK:
Make sure that all of your files (all source files, Makefile, README file, test files, etc) are in the a1 sub-directory of your class account and then follow the class assignment turnin procedures as specified in the instructions in Blackboard