Jacob Kurian

11/05/2016

Section 3

Threads: Lab Report 3

**First: Description of the Project**

* Normal Description: The objective of this project is make different threads and run it concurrent and parallel without corrupting the data of each thread. My program has nine threads some threads are running concurrently and some others are running parallel. In order to prevent the corruption of data, I use semaphore and mutex locks.
* Things that are given: The example of locks and handler are provided by TA and the professor.
* Things those are required: Program should have nine threads including the main thread. There should 3 signal generating thread, 4 handler thread and 1 reporting thread. Signal thread will sent the SIGUSR1 or SIGUSR2 to the handler threads. Also it have to send the signal to the reporting thread. Also I have to figure out the syntax for locks, and wait.
* Suggested Methods: Handler(), Reporter(), Generator().
* Steps I followed: I Make 1 generator and 1 handler at first and tested them. Then finish making rest of the generator and Handler threads. At the very end I make the reporter thread.

**Second: Solution, Approaches Methods for Projects**

* High Level Description of my solution: Create three signal threads and lock the critical section so that one thread can access the critical section at a time. It also protect the global variable in the functions. Create two Handler function and lock it with different lock. To make sure the global variables will not get corrupted. And make 2 threads for each type of signal. There is a signal reporting thread that will get signals from generator threads.
* Sudo-Code of main methods:

Int main(){

Create set1 and block SIGUSR1 for handler1

Create set2 and block SIGUSR2 for handler2

Create set3 and block SIGUSR1 and SIGUSR2 for reporter.

Pthread\_create(Handler1)

Pthread\_create(handler2)

Pthread\_create(repoter)

Pthread\_create(signalGenerator1)

Pthread\_create(signalGenerator2)

Pthread\_create(signalGenerator3)

Pthread\_join-> join all the handlers, reporter and signal generating threads

Print(“Total Sent Signal: ”)

Print(“Total Received signal”)

Printf(“Total Reported SIgnal”)

}

* Methods: There are only four important methods in this project. Two Handler methods that handle either SIGUSR1 or SIGUSR2. One signal generating method that sent signal to handlers thread and reporter thread. Reporter methods take those signals from the generator and calculate the ‘average Signal time’ count.

**Third: Test and Results**

* Bugs:
  + Infinite loop or infinite wait is the one bug that I encounter in this project. It is solved when I used the function ‘sigtimedwait’.
  + Corruption of global variables is another bug I faced. It is solved by using mutex locks.
  + Loss of Signals: The loss of signal is another problem that I faced. I solved that by putting my generator thread in to sleep for 0.05 to 0.1 seconds.
* Requirement I have finished:
  + Create nine threads (1 main, 3 signal generator threads, 4 handler threads, 1 reporter thread). Signal generating thread will randomly picks a signal and randomly sent signal to one of the two corresponding handlers. Then it sent appropriate signal to the reporter thread. After generating each signal it will randomly wait for 0.05 to 0.1 second before it sent the next signal.
  + There are four handler threads. Two of them for SIGUSR1 and two of them for SIGUSR2. There are separate counter for each signals.
  + Reporter thread have counter for each signal. Also it calculate the average time between the reception of SIGUSR1 and SIGUSR2.

**Fourth: Conclusion and Thoughts**

The lab seems hard at the beginning because of the unfamiliar syntax and concepts. But it get fairly easier after I get used to the syntax. The lab help me to understand how does thread works and how to protect the data inside the threads without corrupting it.

**Result to Report**

* Run the program for 30 seconds: Because my wait time after each signal is from 0.05 second to 0.1 second, the program takes 30 second in order to deal with 600 signals. Signal generating threads generated 600 signal and Handler and reporter thread received 600 signals each. The total average time to generate total of 331 SIGUSR1 is about 94080 micro seconds. And total average time to create 269 signal is 1115578 microseconds. The program also print Average time of SIGUSR1 and SIGUSR2.
* Run the program for 10,000 signals: Because my program take relatively long time I didn’t run for 100,000 signals. I just run for 10,000 signals. Signal handler thread and reporter thread got all the signals. It takes about approximately 9 minutes to finish the program. The average time of SIGUSR1 and SIGUSR2 is approximately 0.09 and 0.1 seconds.