Project 4: Signal and Thread Management

CIS 3207

Reed Ceniviva

Testing Notes:

Average SIGUSR1 and SIGUSR2 times are calculated by taking the total time of the program and dividing it by the number of each signal typethat completed within that time. Though one of the signals may not have started at the moment that the program started, the difference in result likely does not account for much error in the calculation of the time for an average signal between receiving one and clearing the same type. Some of the information in the log.csv file is printed to the screen showing the average time for each handling process for every 10 signals processed.

Misses are calculated from the number of threads that are set to be created subtracted by the total number of threads that actually completed. Leaving the number of processes that missed.

N\_HANDS and N\_GENS hold the number of generating and handling threads to be created for the program. handTA and genTA then as Thread Arrays hold the number of threads generated for the program. reportT is the reporter Thread that keeps track of the other threads.

Each of the handling threads between hand1Threads, hand2Threads and reportThread use a signal mask to catch the signals that that thread is meant to record. hand1Threads and hand2Threads use their respective signal (SIGUSR1 and SIGUSR2) for their mask while the reporter thread utilizes both of them to be able to catch either of the signals and thus report on both of them as it should.

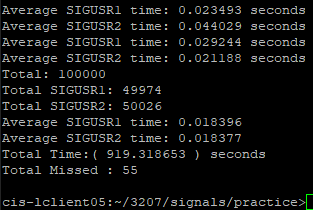
There were a lot of headaches with getting the times to be recorded properly, there is a lot of room for error when it comes to dealing with microseconds and seconds and converting the two back and forth. The time for each signal to be processed is decided in genThreads and had to be a random number between 1 and 10 inclusive and adjusted to be 1000-10000 so that usleep interpreted it correctly.

mkThreads and clrThreads handles the initialization of the threads from main and the takes care of joining the threads at the end of the threads running.

Results are recorded into a CSV file that contains the number of cycles that took place in order to reach the max signals, as well as the average time of each signal type from a 10 signal cycle. The data must be added to the document

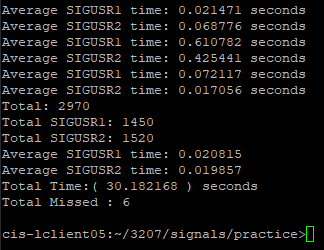
Results:

Project 2: 100,000 run:



It makes sense that the resulting average times would be nearly identical, over enough time they should approach the middle of the range of possible times for the threads to take. The issue is that the average time of these processes should be closer to 5 with a range of 0.1-9.9. I believe that this is a result of multiple sleeping SIGUSR1 or SIGUSR2 signals being woken at closer intervals that they are set to sleep individually, resulting in a fast completion time of signals over all. This may be a result from my choice in where signals are recorded and communication from the reporter thread. The 55 misses I think is a decent record as far as things go. 55/100000 ~= 0.00055% misses. A total time of 919 seconds ( 15.316 min) was better than the expected ~1050 seconds as thought might be the case based on smaller test sizes.

Project 2: 30 sec run:



In the thirty second run there were 6 misses and SIGUSR2 signals were noticeably higher in frequency than SIGUSR1. 6 misses in this case provides 0.00202% chance of a miss which is magnitudes worse that the previous run, but this is a much smaller set that the 100,000 run.

Note from the Student:

I apologize for handing in literally half of the assignment, I feel bad about it as it represents my work in a way but I’ve just been really struggling with things since everything changed due to COVID-19. Not looking for sympathy, just wanted to apologize.