Jaime Riley

CIS 3207- Project 4

Lab Section 4

27 April 2020

**Execution and Analysis- Project 4**

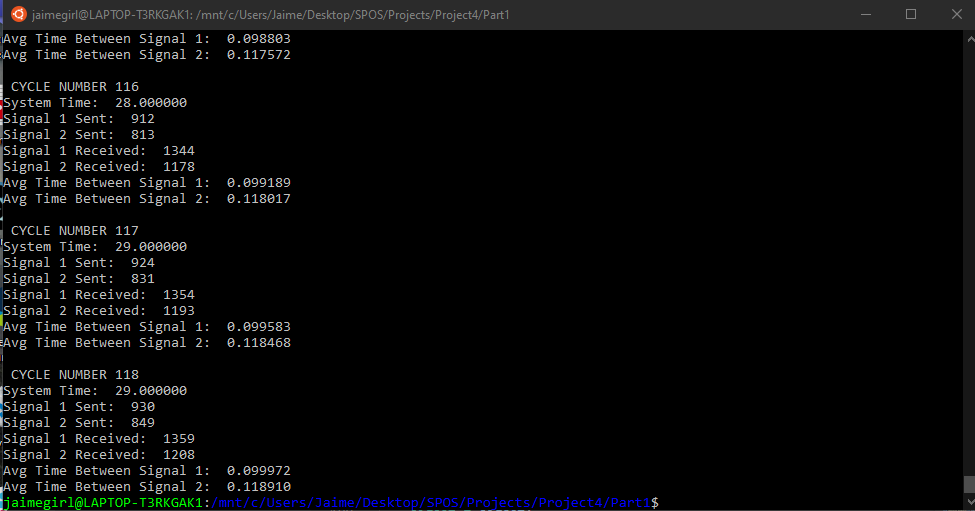
**Process Program Execution Results**

Figure : Process program time trial with test mode off.

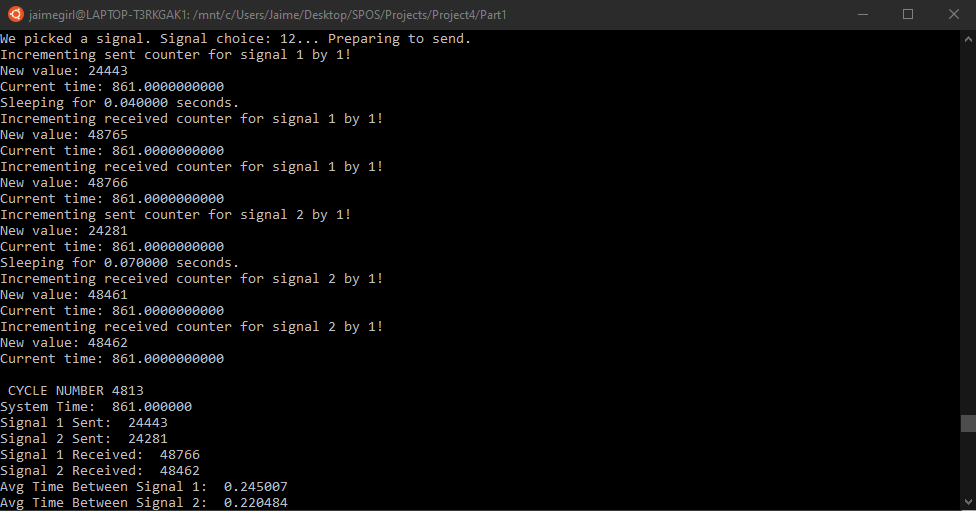


Figure : Process program time trial with test mode on.

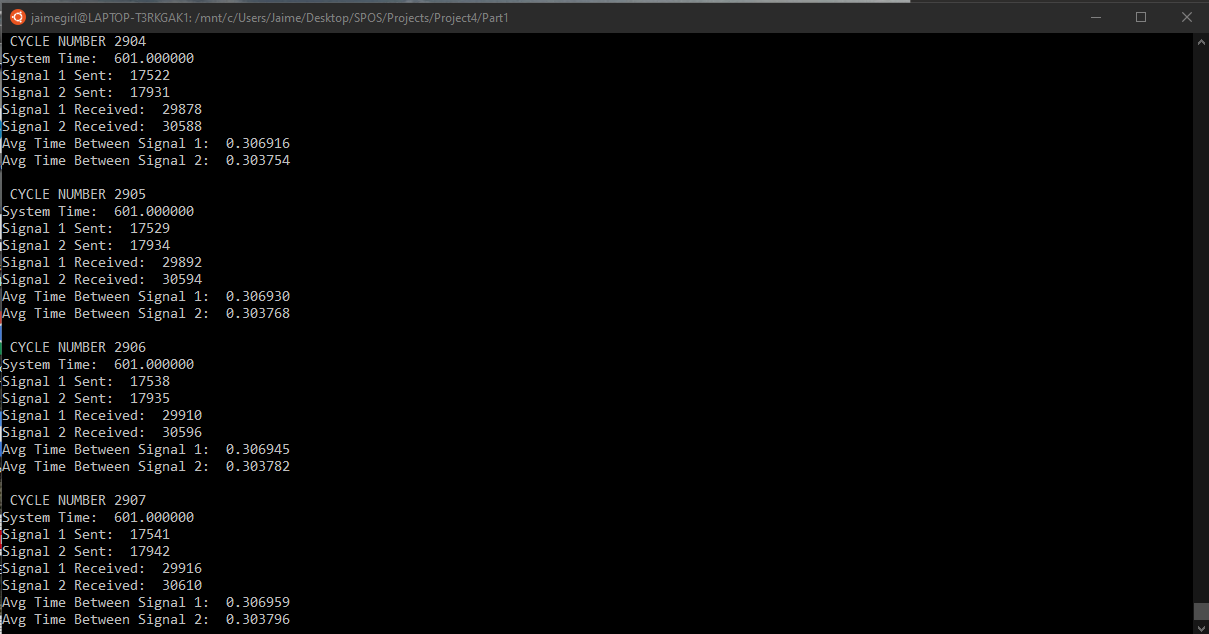


Figure : Process program cycle trial with test mode off.

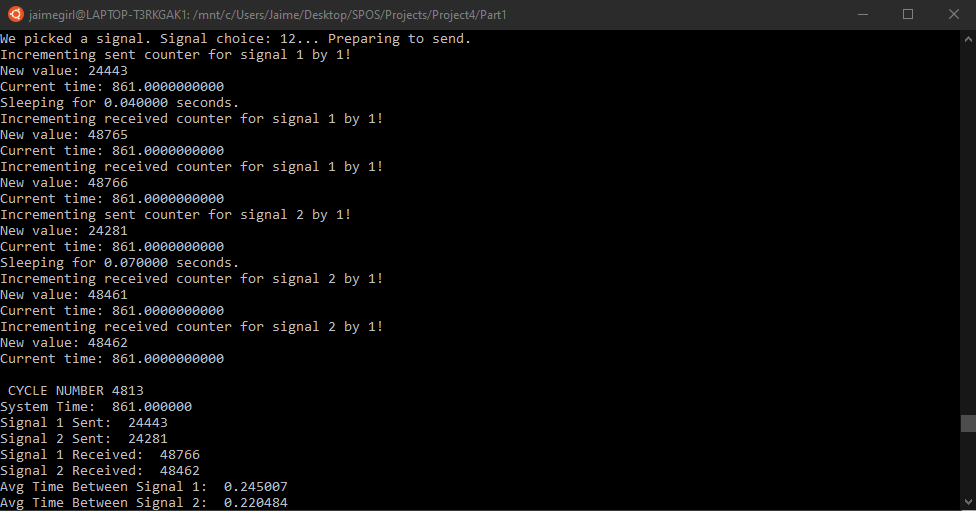


Figure : Process program cycle trial with test mode on.

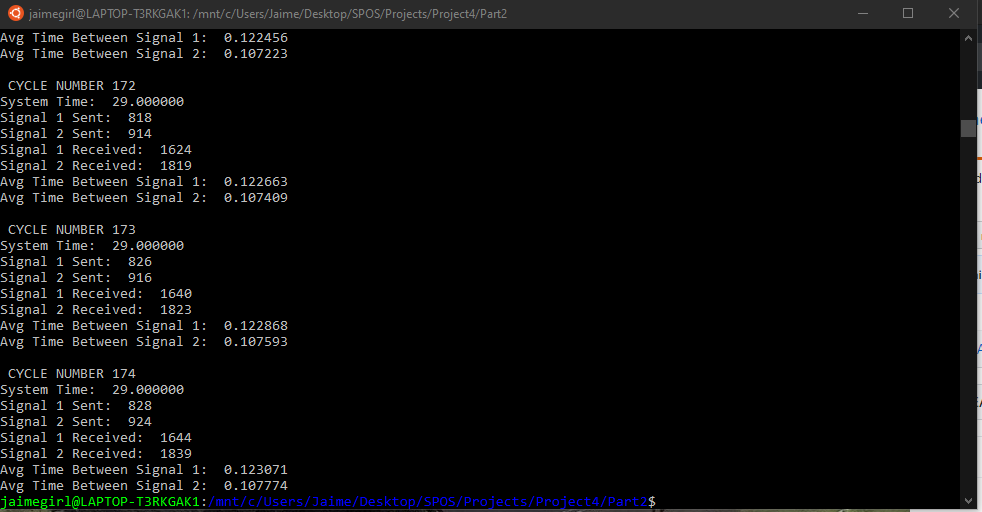
Analysis:

Figure : Process program time trial with test mode on.

In the above screenshots, I have provided the results of both my timed and cycle run of my process program. We can see that in Figure 1, when test mode is off, there are much more signal losses than in Figure 2, when test mode is on. After working with my code for awhile, I was not able to get this fixed until I completely removed all of my test conditionals. Therefore, I will be analyzing the results provided in Figures 2 and 4. We can see that in these cases, although there are signals lost, they are not as prominent in the timed trial then in the cycle trial. The longer the program runs for, the more signals are lost, which makes sense given there are more opportunities for mutual exclusion errors and memory leak. Also, as time went on, we can see that the average interval between signals and receptions went up. This is also fairly intuitive given that the longer the program runs, the slower the CPU is able to respond. Personally, when running the cycle trials, my computer was audibly disagreeing with me despite the sleep() calls between cycles.

**Thread Program Execution Results**

Figure : Thread program time trial with test mode off.



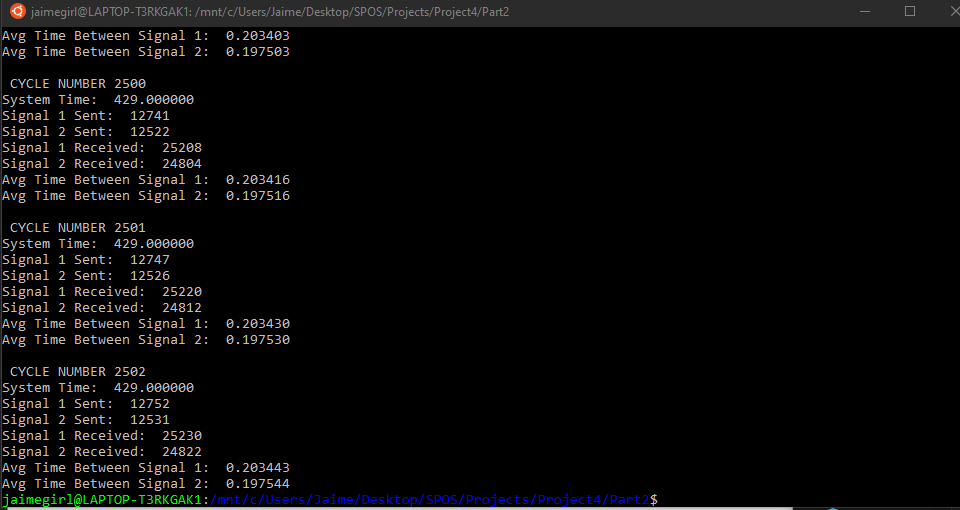
****

Figure : Thread program cycle trial with test mode off.

Analysis:

In the above figures for the thread program, we can see that the results were much more accurate even when test mode was off. I did not have the same problem I ran into during my process program execution. The thread program had less signals missed and executed more quickly than the process program. The average interval between signals being received and signals being sent was also somewhat smaller in this program. Even with very large numbers of signals being sent, there were not many being missed in the multi-thread program.