**Result/ Analysis Doc**

Firstly, my entire code was written on an online compiler from replt.it

**RESULTS**

***Process Program ran for 30 seconds***

*Run 1*

System Time = 0.17s

Total Signals sent = 2999

Total Signals received = 8635

Average SIGURS1 reception time by receiving thread in 10 signal intervals = 0.000068s

Average SIGURS2 reception time by receiving thread in 10 signal intervals = 0.000084s

*Run 2*

System Time = 0.15s

Total Signals sent = 2072

Total Signals received = 5792

Average SIGURS1 reception time by receiving thread in 10 signal intervals = 0.000060s

Average SIGURS2 reception time by receiving thread in 10 signal intervals time = 0.000079s

*Run 3*

System Time = 0.16s

Total Signals sent = 2419

Total Signals received = 7072

Average SIGURS1 reception time by receiving thread in 10 signal intervals = 0.000069s

Average SIGURS2 reception time by receiving thread in 10 signal intervals time = 0.000084s

***Thread Program ran for 30 seconds***

*Run 1*

System Time = 0.45s

Total Signals sent = 1886

Total Signals received = 5549

Average SIGURS1 reception time by receiving thread in 10 signal intervals = 0.000249s

Average SIGURS2 reception time by receiving thread in 10 signal intervals = 0.000245s

*Run 2*

System Time = 0.39s

Total Signals sent = 2477

Total Signals received = 7300

Average SIGURS1 reception time by receiving thread in 10 signal intervals = 0.000217s

Average SIGURS2 reception time by receiving thread in 10 signal intervals time = 0.000193s

*Run 3*

System Time = 0.43s

Total Signals sent = 1951

Total Signals received = 5652

Average SIGURS1 reception time by receiving thread in 10 signal intervals = 0.000236s

Average SIGURS2 reception time by receiving thread in 10 signal intervals time = 0.000228s

***Process Program ran for 10,000 signals***

*Run 1*

System Time = 0.694s

Total Signals sent = 10000

Total Signals received = 28053

Average SIGURS1 reception time by receiving thread in 10 signal intervals = 0.000065s

Average SIGURS2 reception time by receiving thread in 10 signal intervals = 0.000080s

*Run 2*

System Time = 0.654s

Total Signals sent = 10000

Total Signals received = 27561

Average SIGURS1 reception time by receiving thread in 10 signal intervals = 0.000068s

Average SIGURS2 reception time by receiving thread in 10 signal intervals time = 0.000081s

*Run 3*

System Time = 0.702s

Total Signals sent = 10000

Total Signals received = 29584

Average SIGURS1 reception time by receiving thread in 10 signal intervals = 0.000069s

Average SIGURS2 reception time by receiving thread in 10 signal intervals time = 0.000084s

***Thread Program ran for 10,000 signals***

*Run 1*

System Time = 1.90s

Total Signals sent = 10000

Total Signals received = 28668

Average SIGURS1 reception time by receiving thread in 10 signal intervals = 0.000208s

Average SIGURS2 reception time by receiving thread in 10 signal intervals = 0.000205s

*Run 2*

System Time = 2.034s

Total Signals sent =10000

Total Signals received = 29562

Average SIGURS1 reception time by receiving thread in 10 signal intervals = 0.000210s

Average SIGURS2 reception time by receiving thread in 10 signal intervals time = 0.000198s

*Run 3*

System Time = 1.87s

Total Signals sent = 10000

Total Signals received = 28642

Average SIGURS1 reception time by receiving thread in 10 signal intervals = 0.000205s

Average SIGURS2 reception time by receiving thread in 10 signal intervals time = 0.000202s

**ANALYSIS**

* Signals received were about 3 times signals sent. This is because when signal generator sends out 1 signal, 3 other processes or threads are handling these signals for each signal type.
* Regardless of how the programs were ran (30 seconds or 10000 signals), process program behaved similarly because 10000 signals are just about 2 minutes
* Multithreading leads to faster computing speed. However, the ratio of total sent/received for both programs is about the same. This is most likely because signals in our threads are received synchronously, which is the trade of when dealing with threads. So, although the threads were generated. If our thread program were asynchronous, a lot more work would have been done
* The system time for threads were at least double that of the processes for each case so it took more time for the threads to generate signals, but approximately same amount signals were handled
* There is signal loss because number of signals handled is not exactly 3 times the number of signals sent. So, some signals are stuck on send mode. This is because the counters are in mutex locks so every single signal sent cannot be received at the same time Also for our non- reporting processes the other signal is blocked. Furthermore, out thread program handles signal synchronously, so all signals sent cannot be handled