Sean Patlan

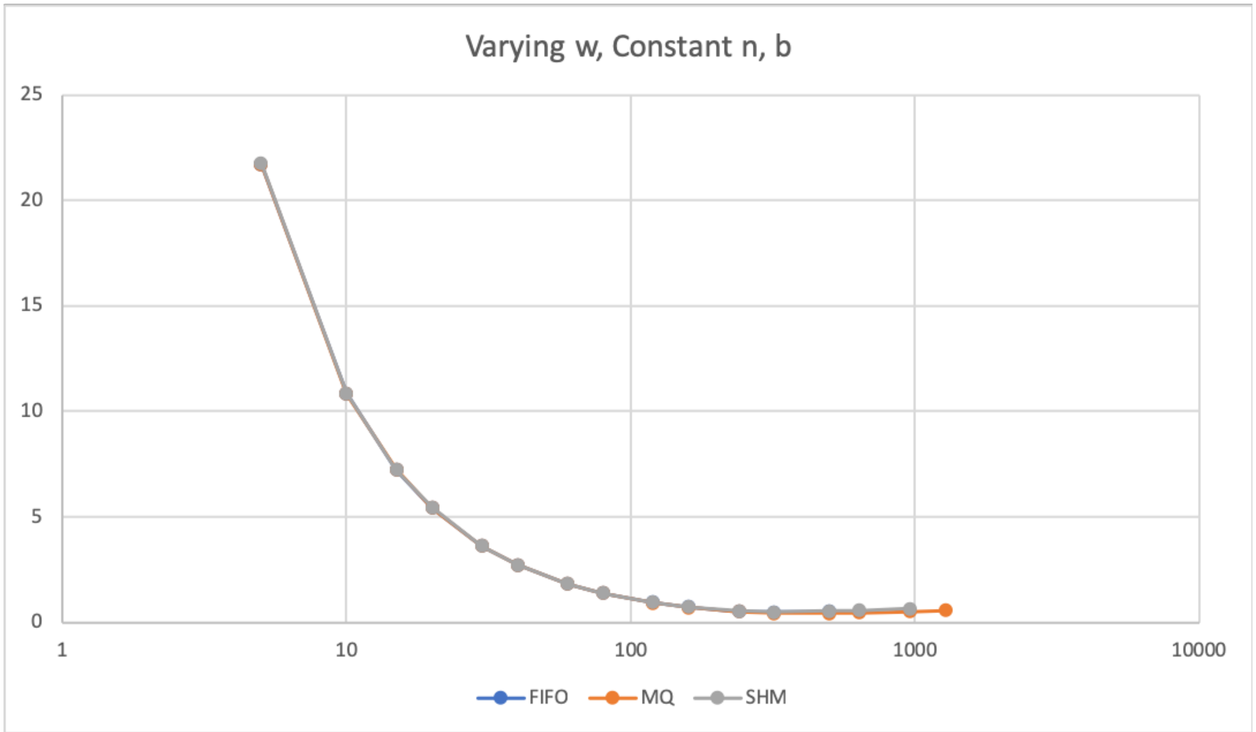
CSCE 313-508

Programming Assignment #6 Report

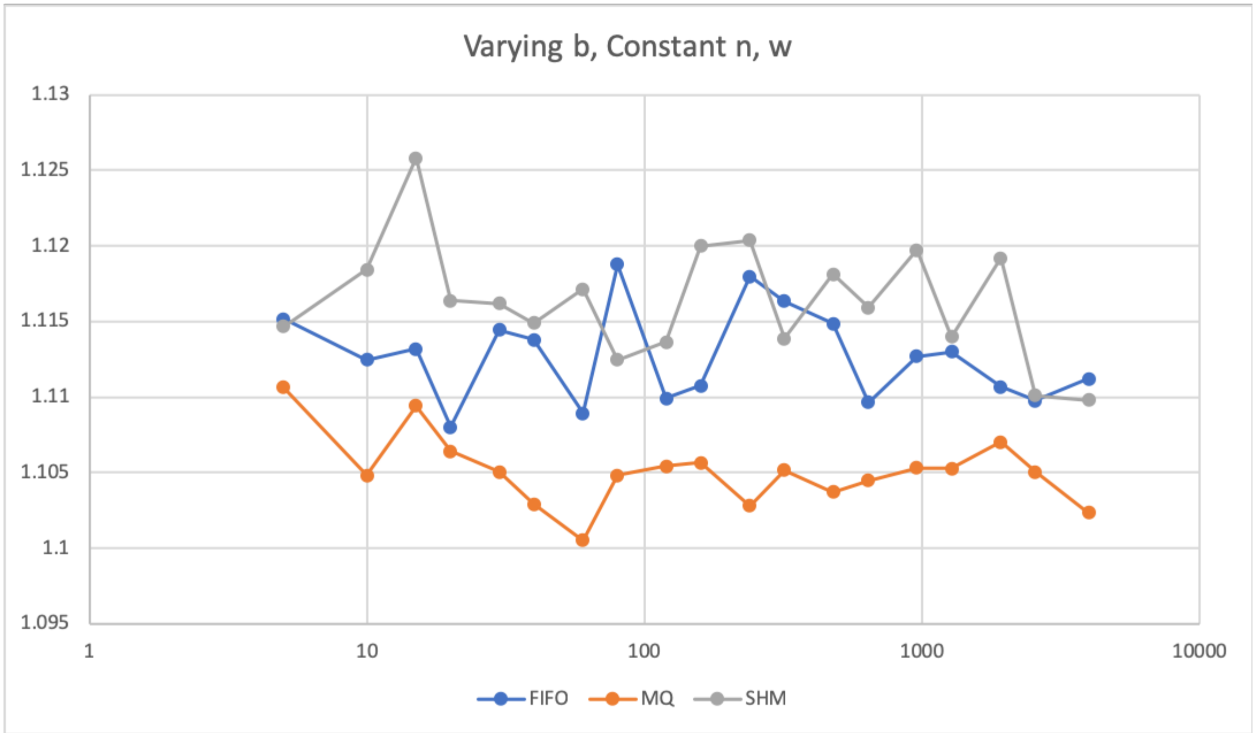
In this programming assignment we executed three different implementations of the server-client code we've been using for the past several PAs. The first implementation was identical to that of project 4, and the other two used IPCs to create the request channels.

The MQ implementation was fairly straightforward since it was already synchronized, but the SHM implementation required that we create an SHMBoundedBuffer class to push and pop requests, as well as a KernelSemaphore class so that the buffer would be not only thread-safe, but also inter-proccess-safe.

To test and compare these implementations, I started by increasing w while keeping n at 10000 and b at 300 and recording the time it takes for the program to complete. Each implementation had very similar results, but MQ was slightly faster than the other two (by a fraction of a second). MQ was also able to support the highest number of worker threads (1280). FIFO was only able to reach 500 (same as PA4) and SHM was able to reach 1000. Below is a graph of the time test results:



The results for an increasing b value with constant w and n is much less smooth of a curve as the w-curve, as has been the case for all of the recent PAs. It seems that b has much less of a direct effect on the time the program takes to finish than w does. Below is a graph of the results of the time test for a varied b value:


Varying b, Constant n, w
FIFO
MQ
SHM