# PA3: Interprocess Communication Mechanisms

*Brandon DeAlmeida*

*CSCE 313-508*

*Instructor: Tanzir Ahmed*

*TA: Feras Khemakhem*

# Design:

While writing this programming assignment I was able to simplify both my client and server executable greatly through abstraction by not needing to write separate code for each possible mode (FIFO, MQ, and SHM). This was accomplished through having each mode's respective "RequestChannel" class inherit certain traits from the parent *RequestChannel*, meaning that both client and server would run the same code regardless of what mode they were in.

As such, the differences between each mode are within each specific channel type class. The FIFO and MQ channels resemble each other in the fact that both require two separate "streams" (one going from the server to client and the other heading in the opposite direction), and differ only in the ipc-specific commands each needs to run. For example, the FIFO channel utilizes **mkfifo()** and **open()** where MQ uses **mq_open()**. SHM is the most strikingly different mode which only requires one collective stream and the use of Semaphones that allow for communication on who's turn it is to write or read. There are four Semaphones created for each SHM channel: two detail if the client has read or written while the final two detail the same info for the server, these allow both the client and server to 'talk' over one segment of shared memory.

# Data:

While testing my code, I found the following time data where the number of channels is always 5, and the times shown are averages of 5 consecutive runs of the test:

| | 1 Point | 1K Points | Filecopy (1.csv) |
|---|---|---|---|
| PA3_FIFO | *394.6μs* | *68.35ms* | *13.07ms* |
| PA3_MQ | *589.9ms* | *647.4ms* | *611.1ms* |
| PA3_SHM | *592.1ms* | *648.1ms* | *608.3ms* |

It can be seen that both the MQ and SHM modes shared very similar times across all three test cases, showing somewhat similar performance between them.

It's worth noting that the times given for both the **1 Point** and **1K Points** test cases are the summation of how long it took to get these points (each channel ran after the previous completed) while for **Filecopy** the work was split between each channel such that each channel would perform one filemsg request before handing off to the next.