Frank, Wednesday February 5, 2020
Suh6, 001411435

COMPSCI-3SH3 Operating Systems Assignment 2

1. Full source code: I will attach the source code as files when I submit the assignment
2. Short description of your implementation and advantage/disadvantage of each IPC mechanism. The description should not be longer than half a page

Q2 Answer:

For my shared memory implementation, my strategy was to save the memory mapping of the shared memory region to a timeval struct pointer. This made it much simpler because I could use gettimeofday() to directly write the time value to the shared memory, without using sprintf() and dealing with pointer increment/decrement which could easily cause segmentation faults and garbage values. I used a similar approach for pipes, where I would write to the pipe using a timeval pointer, and reading it in a similar fashion.

In terms of advantages, pipes has synchronization built into the mechanism itself; either reading/writing will freeze it. However, with shared memory, you must implement synchronization for read/writes (i.e. using semaphores) to make sure that only one process has access to the shared memory at once. However, an advantage for shared memory is that it gives you more control over buffering and reserource use, as you explicitly declare how much memory to allocate and how to use it. For pipes, on the other hand, the OS does much of this automatically behind the scenes.

Thus, we can see that generally there is a tradeoff between shared memory and pipes; pipes are better for one-to-one communication, and is generally simpler. However, shared memory allows more flexibility, but the implementation requires more care and code in general to avoid potential errors such as segmentation faults.