**Software Entrance Assignment – Part 1**

Pipes and shared memory are two main ways for programs to communicate with each other. Pipes are used like message passing; someone sends a piece of information to the recipient and the recipient can receive it. Shared memory, on the other hand, is more like publishing data; someone puts data in shared memory and the readers must use synchronization. With pipes, the synchronization is simple and built into the pipe mechanism itself. With shared memory, it is easier to work asynchronously and check for new data only once in a while, but it requires more complex code. Also, debugging of pipe-based communication is easier than debugging shared memory. In addition, shared memory also gives you more control over buffering and resource use. With pipes, the OS controls things automatically, so once again you lose some flexibility but are relieved of much work. In summary, pipes allow for one-to-one communication, less coding, and let the OS handle things. Meanwhile, shared memory allows for more manual control over things but at the cost of more work and harder debugging. Due to these differences between pipes and shared memory, I would probably use pipes because they seem more convenient to use.