Ryan Russell

Dr. Ahmed

CSCE 313.504

8 September 2020

Programming Assignment One Report

The design of this program involved having a client being able to speak with a server and the server sending back information back to the client. This involved using a FIFOreqchannel (or FIFP pipe for short) which is the medium of communication since both the client and server are two separate processes. The FIFOreqchannel could send messages to the client from the server or vice versa using a function called cwrite which allows for the channel to know what the message is and where it is going. The FIFOreqchannel could receive these messages from one of the processes using a function called cread which allows for the channel to hold the information and disperse that information to the correct location. Along with these two functions, the FIFOreqchannel also handled messages (examples being data, file, new channel, and quit messages) that both processes can receive from the channel.

When it comes to the case of data points, requesting one data point takes about 4800 to 6200 microseconds to accomplish using the getimeofday function. However, the amount of time it took for 1000 data points to be requested and put into a file x1.csv was 2.06471\*10^7 microseconds. When it comes to transferring files of various sizes, the amount of time to accomplish can vary if the buffer capacity is standard MAX\_MESSAGE (which is 256). In the graph below titled “Transferring File: Size vs. Time (μs)”, the data on the graph shows that as the file size increases, the amount of time the transfer takes to complete increases as well. This is also proven to be reliable data as the trendline of the graph is y = 0.0962x + 194.86 (where x is the size of the file and y is the time it takes to transfer a file) with a correlation coefficient of 0.9971. Since the correlation coefficient is close to equaling one, this means that the data fits very well with the trendline function given for the graph.

In the case of transferring large files (for example, files sized at 100MB), it can take a long period of time especially when the buffer capacity is set to MAX\_MESSAGE (which is equal to 256). However, by changing the buffer capacity, the transfer of large files is made quicker. In the graph below titled “Transferring Large File: Time vs. Buffer Capacity”, the data on the graph shows that as the buffer capacity increases, the amount of time the transfer of the 100MB file takes to complete decreases as well. This is also proven to be reliable data as the trendline of the graph is y = (7\*10^6)\*(e^((-7\*10^-4)\*x)) (where x is the buffer capacity and y is the time it takes to transfer the file) with a correlation coefficient of 0.8451. Since the correlation coefficient is close to equaling one, this means that the data fits very well with the trendline function given for the graph.

Demo Video Link:

https://drive.google.com/file/d/1wiLFu4XE1hNCO67QhZYxpkecQkyjRNv1/view?usp=sharing