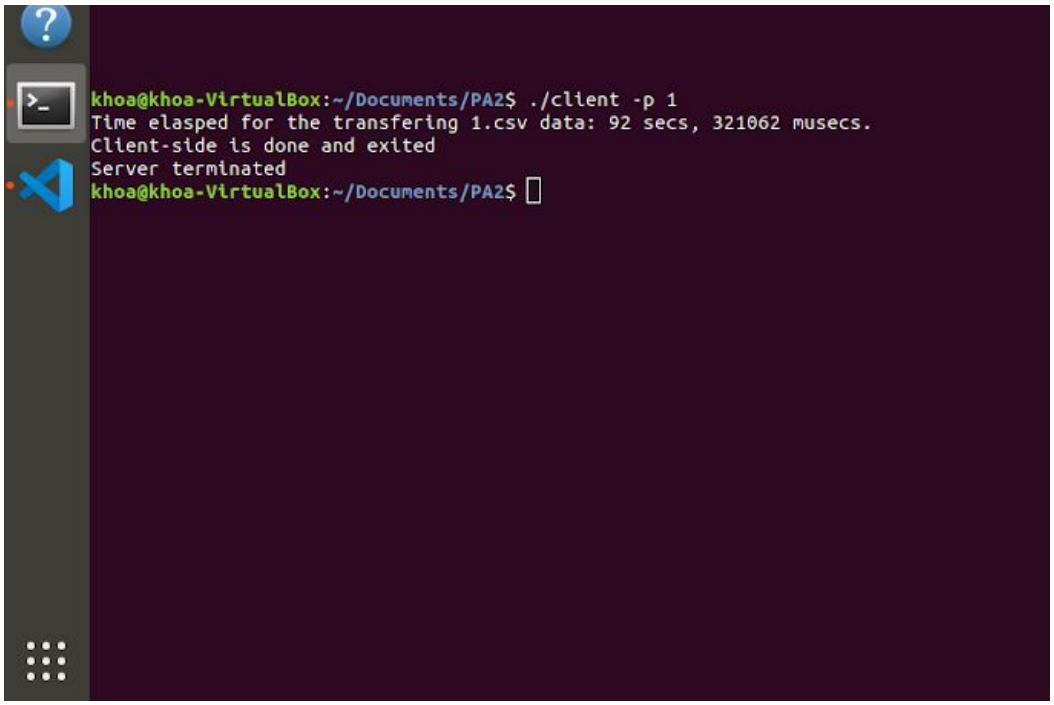
Khoa Diep
CSCE 313 - 501
2/7/20

Requesting Data Points

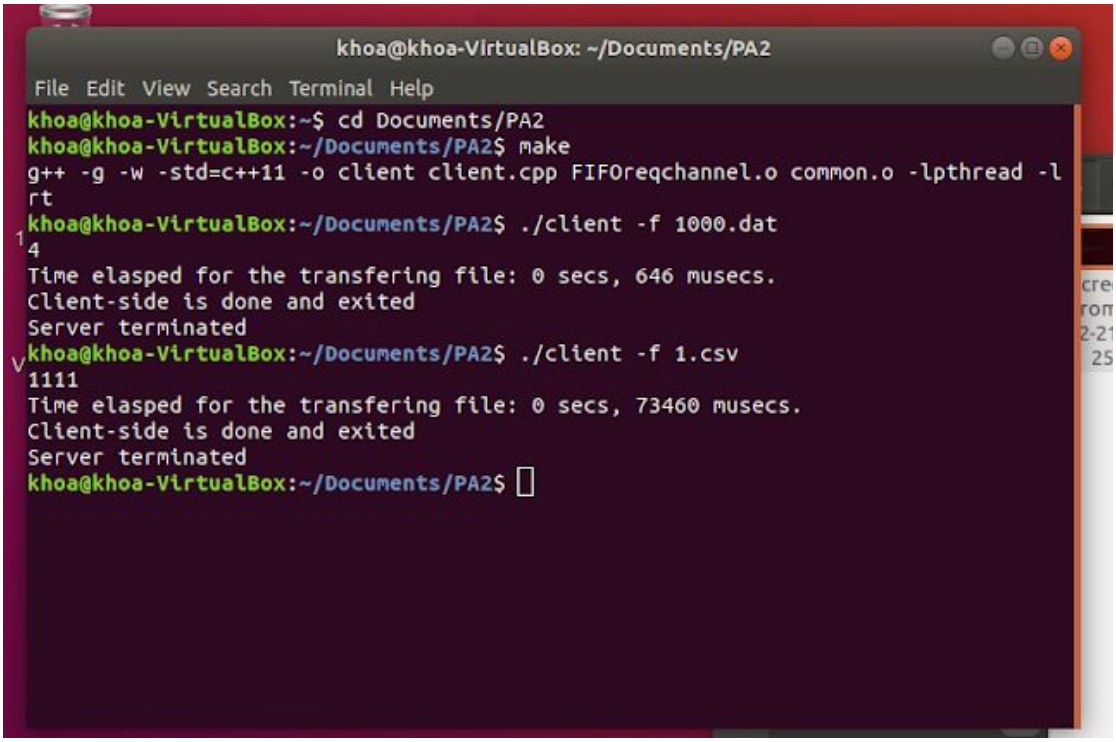
Data points were requested from the command argument -p. The argument value after it relates to which csv it requests from. A single data point can also be received by using the -t and -e, where the -t relates to the time and -e relate the ECG value. The datamsg is written in the buffer, and then the client reads the buffer the server sent it and typecast it to a double. This took a while and took 92.322 seconds.

.



Requesting Files

Files can be requested similarly to how data can be requested, but instead of a datamsg, a filemsg must be sent. The -f flag will let the user request which file the client will request. The file name must also be sent to the server in order for the server to know which file to send. A file message with 0,0 as its parameters will return the file's length. The server will send the file in chunks. This is done by dividing the size of the file by the buffer capacity of the server. By default, the buffer capacity is 256 bytes by can be changed by using the -m argument. The client can request any file to be sent including text files and binary files. This will be significantly faster than the datamsg method since much fewer requests are made between the server and the client. The time it took to request 1.csv into y1.csv (requesting a file) took much less than it took to request 1.csv into x1.csv.



Requesting a New Channel

To request a new channel, all we have to do is send a NEWCHANNEL_MSG from the client to the server. After that, you simply open the channel with “data1_” as the first parameter and close the channel after done using it.

System:

- Intel Core i7, 7th gen
- 16 GB RAM
- OS - Ubuntu and Windows(to write on)

Bottlenecking:

The bottleneck of this system is obviously the MAX_MESSAGE limitation. This states that the max buffer capacity we can use is 256 bytes. Since the communication between the server and the client will happen in a linear fashion, if we can increase the buffer capacity, the rate of data between the server and the client will increase. We could also communicate between multiple servers so the client can have more than one server sending data to it, which will speed up the process of transferring data.