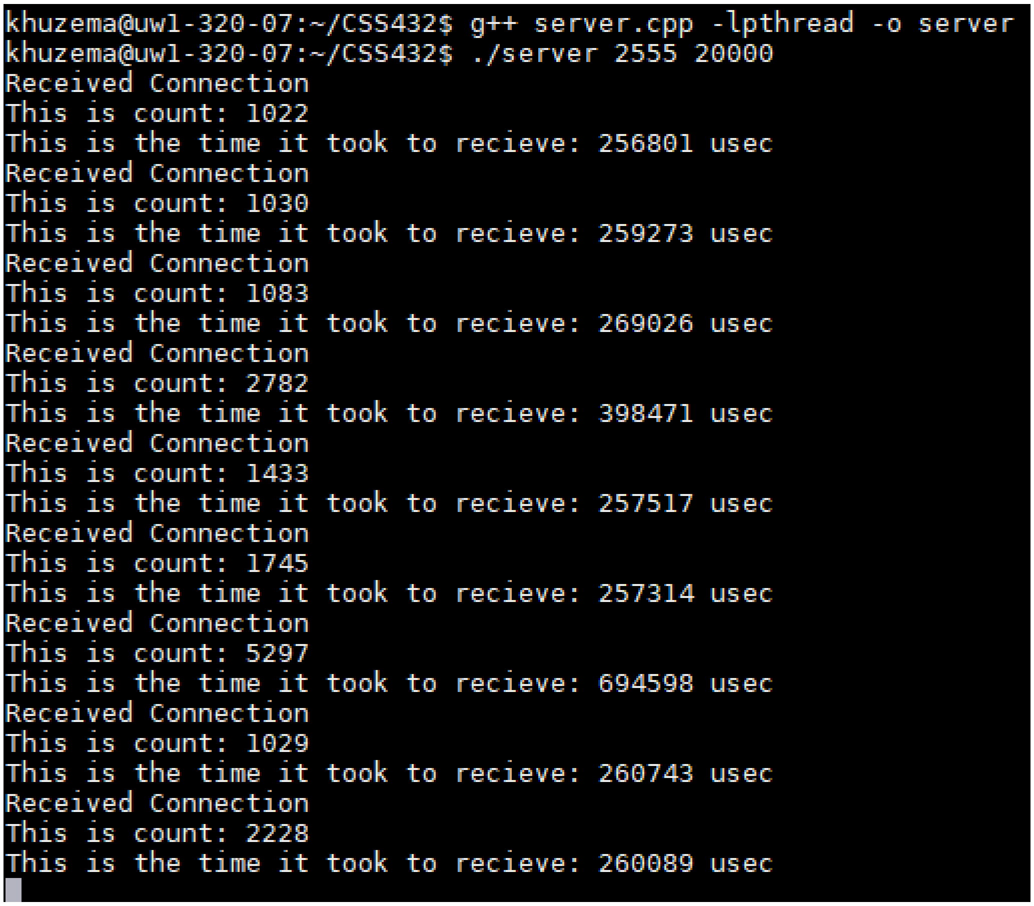
Khuzaima Mushtaq

CSS 432

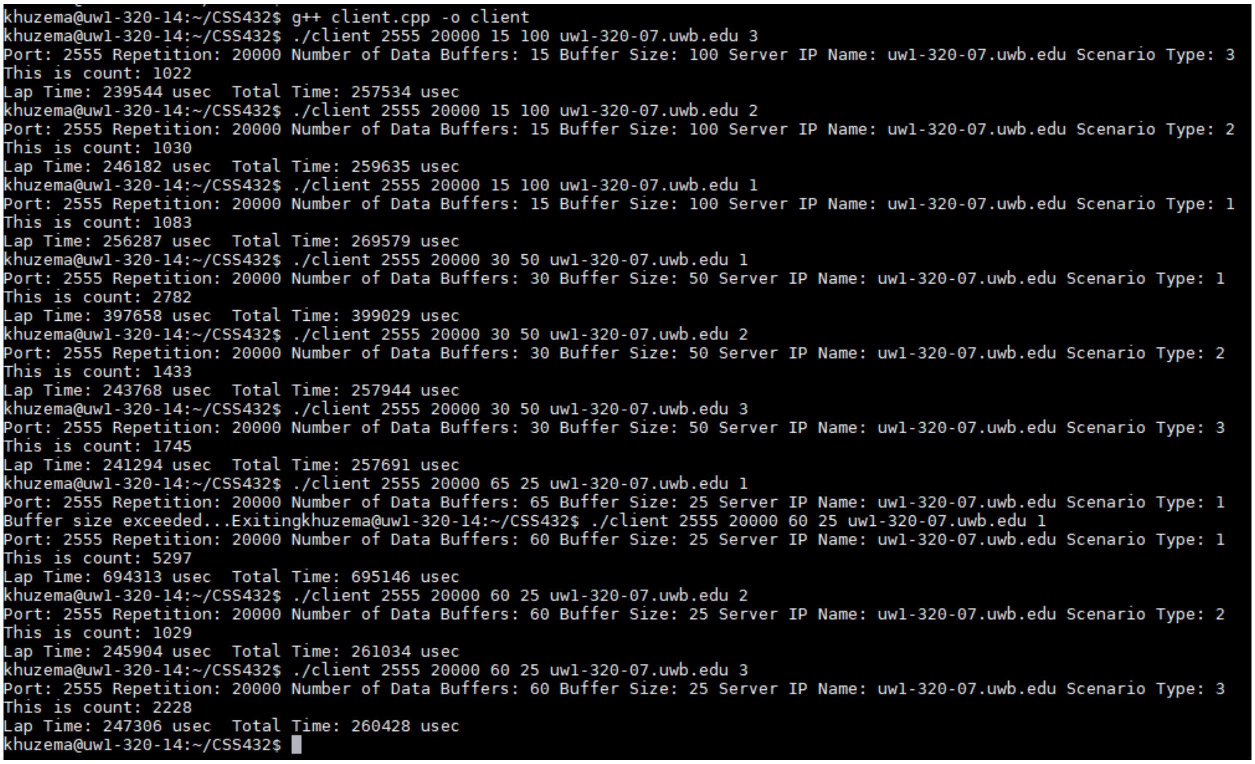
Program 1: Intro to Network Programming

My program follows the guide in the assignment very closely. I use the recommended socket code for setting up and connecting my sockets. In client, I use the args to set up my local variables then set up a socket connection to the proper port and server host provided in args. Once the connection is established, based on the test scenario type (1, 2, or 3) I write to the clientSd, number of repetition times. Then I read back from the server the number of times it used the call read. I use gettimeofday to keep track of how long it takes to send all the data and then how long it takes for the server reply to get back. In server, I use the args to get the number of repitiontions to call read and the port number to listen on. I set up the socket and listen for connections. When a connection is spawned I take the socket descriptor for that connection, create a new thread and pass the SD into it, where that connection is handled. When a new thread is spawned I get the SD from the parameter passed in, then I create a buffer size 1500 as specified. Then I read from the socket repetition times and keep track of how much time it takes to read everything, and how many times read is called. Then I output the time it took to read to screen and write back to the socket the number of times read was called, which is then read from the client.

EXECUTION OUTPUT:



```
khuzema@uw1-320-07:~/CSS432$ g++ server.cpp -lpthread -o server
khuzema@uw1-320-07:~/CSS432$ ./server 2555 20000
Received Connection
This is count: 1022
This is the time it took to recieve: 256801 usec
Received Connection
This is count: 1030
This is the time it took to recieve: 259273 usec
Received Connection
This is count: 1083
This is the time it took to recieve: 269026 usec
Received Connection
This is count: 2782
This is the time it took to recieve: 398471 usec
Received Connection
This is count: 1433
This is the time it took to recieve: 257517 usec
Received Connection
This is count: 1745
This is the time it took to recieve: 257314 usec
Received Connection
This is count: 5297
This is the time it took to recieve: 694598 usec
Received Connection
This is count: 1029
This is the time it took to recieve: 260743 usec
Received Connection
This is count: 2228
This is the time it took to recieve: 260089 usec
```



```
khuzema@uw1-320-14:~/CSS432$ g++ client.cpp -o client
khuzema@uw1-320-14:~/CSS432$ ./client 2555 20000 15 100 uw1-320-07.uwb.edu 3
Port: 2555 Repetition: 20000 Number of Data Buffers: 15 Buffer Size: 100 Server IP Name: uw1-320-07.uwb.edu Scenario Type: 3
This is count: 1022
Lap Time: 239544 usec  Total Time: 257534 usec
khuzema@uw1-320-14:~/CSS432$ ./client 2555 20000 15 100 uw1-320-07.uwb.edu 2
Port: 2555 Repetition: 20000 Number of Data Buffers: 15 Buffer Size: 100 Server IP Name: uw1-320-07.uwb.edu Scenario Type: 2
This is count: 1030
Lap Time: 246182 usec  Total Time: 259635 usec
khuzema@uw1-320-14:~/CSS432$ ./client 2555 20000 15 100 uw1-320-07.uwb.edu 1
Port: 2555 Repetition: 20000 Number of Data Buffers: 15 Buffer Size: 100 Server IP Name: uw1-320-07.uwb.edu Scenario Type: 1
This is count: 1083
Lap Time: 256287 usec  Total Time: 269579 usec
khuzema@uw1-320-14:~/CSS432$ ./client 2555 20000 30 50 uw1-320-07.uwb.edu 1
Port: 2555 Repetition: 20000 Number of Data Buffers: 30 Buffer Size: 50 Server IP Name: uw1-320-07.uwb.edu Scenario Type: 1
This is count: 2782
Lap Time: 397658 usec  Total Time: 399029 usec
khuzema@uw1-320-14:~/CSS432$ ./client 2555 20000 30 50 uw1-320-07.uwb.edu 2
Port: 2555 Repetition: 20000 Number of Data Buffers: 30 Buffer Size: 50 Server IP Name: uw1-320-07.uwb.edu Scenario Type: 2
This is count: 1433
Lap Time: 243768 usec  Total Time: 257944 usec
khuzema@uw1-320-14:~/CSS432$ ./client 2555 20000 30 50 uw1-320-07.uwb.edu 3
Port: 2555 Repetition: 20000 Number of Data Buffers: 30 Buffer Size: 50 Server IP Name: uw1-320-07.uwb.edu Scenario Type: 3
This is count: 1745
Lap Time: 241294 usec  Total Time: 257691 usec
khuzema@uw1-320-14:~/CSS432$ ./client 2555 20000 65 25 uw1-320-07.uwb.edu 1
Port: 2555 Repetition: 20000 Number of Data Buffers: 65 Buffer Size: 25 Server IP Name: uw1-320-07.uwb.edu Scenario Type: 1
Buffer size exceeded...Exitingkhuzema@uw1-320-14:~/CSS432$ ./client 2555 20000 60 25 uw1-320-07.uwb.edu 1
Port: 2555 Repetition: 20000 Number of Data Buffers: 60 Buffer Size: 25 Server IP Name: uw1-320-07.uwb.edu Scenario Type: 1
This is count: 5297
Lap Time: 694313 usec  Total Time: 695146 usec
khuzema@uw1-320-14:~/CSS432$ ./client 2555 20000 60 25 uw1-320-07.uwb.edu 2
Port: 2555 Repetition: 20000 Number of Data Buffers: 60 Buffer Size: 25 Server IP Name: uw1-320-07.uwb.edu Scenario Type: 2
This is count: 1029
Lap Time: 245904 usec  Total Time: 261034 usec
khuzema@uw1-320-14:~/CSS432$ ./client 2555 20000 60 25 uw1-320-07.uwb.edu 3
Port: 2555 Repetition: 20000 Number of Data Buffers: 60 Buffer Size: 25 Server IP Name: uw1-320-07.uwb.edu Scenario Type: 3
This is count: 2228
Lap Time: 247306 usec  Total Time: 260428 usec
khuzema@uw1-320-14:~/CSS432$
```

PERFORMANCE EVALUATION:

Repetition	nbuf	bufsize	Type 1	Type 2	Type 3
20000	15	100	count:1083 LapTime:256287 TotalTime:269579	count:1030 LapTime:246182 TotalTime:259635	count: 1022 LapTime: 239544 usec TotalTime:257534 usec
20000	30	50	count: 2782 LapTime: 397658 TotalTime:399029	count: 1433 LapTime: 243768 TotalTime:257944	count: 1745 LapTime: 241294 usec TotalTime:257691 usec
20000	60	25	count: 5297 LapTime: 694313 TotalTime:695146	count: 1029 LapTime: 245904 TotalTime:261034	count: 2228 LapTime: 247306 usec TotalTime:260428 usec

DISCUSSION:

This results would be different when using a slower network especially in terms of the difference between TotalTime and LapTime. I think count ie. The number of times read is called would still be the same and the data would be read in similar chunks since we are using a small size buffer of 1500 bytes. Overall, the time it takes to complete the same actions would be slower but the biggest difference will be noticeable in the Total Time. Between the different test cases however, Type 1 would be affected the least by a slower connection since it makes multiple writes and sends smaller chunks of data individually which can travel as needed on a slower network. Type 3, on the other hand, sends all data in a single write, and on a slower network this has a chance of causing packet loss due to buffer overload if the data being sent exceeds that capacity and also might max out the bandwidth on a slower connection. In the case of Type 2, which sends all the data in the form of a iovec, the downside would be similar to Type 3. Additionally, we want to use multiple threads to handle separate connections because that way our server can handle multiple connections as a server should be able to do, This also makes it so that each connection is a separate entity and has its own line of execution thereby not messing up different connections as it might if it was not multi-threaded. This also allows us to put our server is a while loop forever and still have it be able to service new threads without needing multiple variables for expected connections.