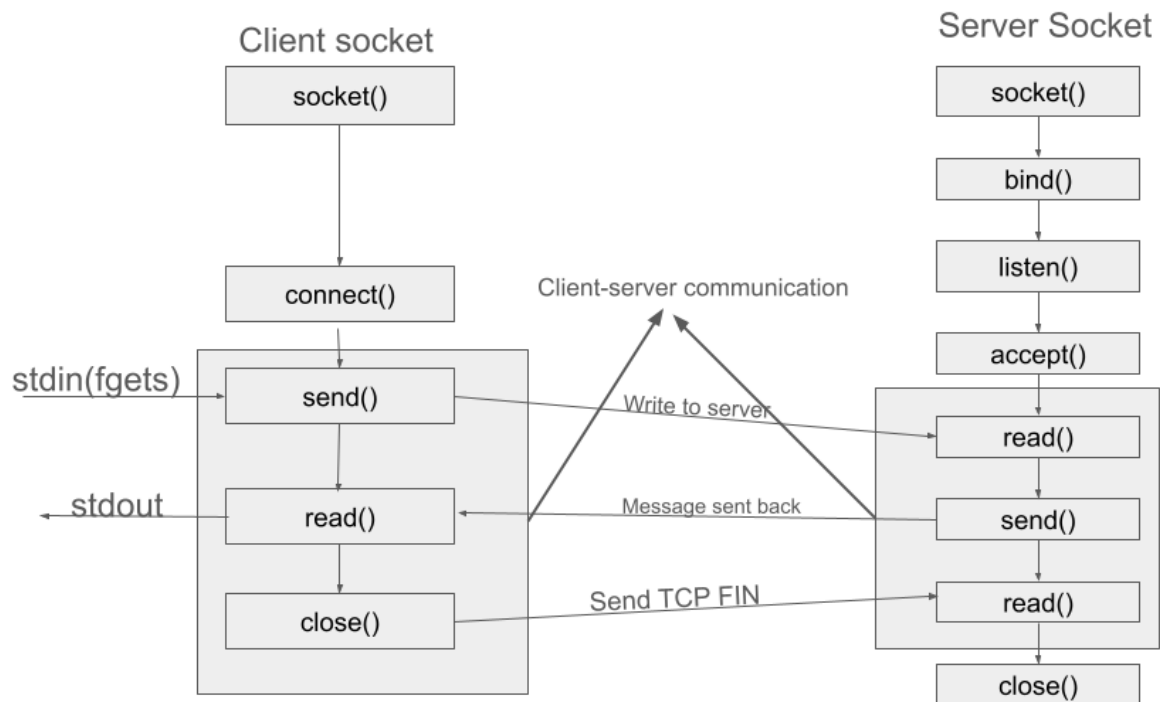


Team 8: ECEN602-socket_programming

ECEN 602 Machine Problem-1 by Team 8

Socket programming introduction and architecture

In this project, we have implemented a client server TCP socket where multiple clients can echo to the server.



Functions used by both client and server

- 1. Socket():** A socket descriptor is defined containing the socket family(whether IPv4 or IPv6) and socket type(TCP or UDP). In our case we have implemented a TCP stream socket and IPv4 address scheme.
- 2. read():** This function reads a line of text from the client and stores them in a buffer with a specified size. read() is also used to get the TCP_FIN packet from the client.
- 3. send():** This function writes data buffer from the client to the server.
- 4. close():** Closes the client/server instance. When the client closes the connection, the TCP_FIN packet is received by the server.

Functions used by the server

- 1. bind():** Binds the socket to the specified IP address and port number to listen
- 2. listen():** Listens for clients at the specified IP address and port. It is used to accept incoming clients

3. accept(): The server waits for client connection to complete. This function also hold the clients information(e.g. address).

Functions used by the client

1. connect(): Initiates a connection to the socket at the specified IP address and port number.

Steps to run the code

Main code is present in "/Original_work" directory which contains the original code. below are the commands to run the makefile.

```
cd Original_work/
```

```
make all
```

```
make echos PORT="PORT NUMBER"
```

Open another terminal and enter the following

```
make echo SERVER_ADDR="IP ADDRESS" PORT="PORT NUMBER"
```

Example

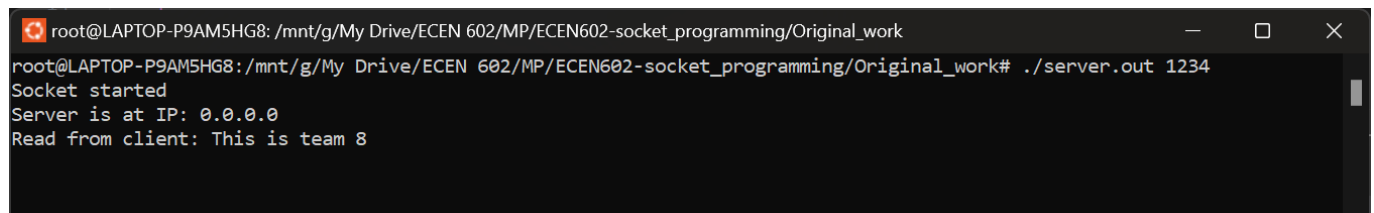
```
make echos PORT=1234  
make echo SERVER_ADDR=127.0.0.1 PORT=1234
```

Test Case Execution

1. Line of text executed by newline

In this test case, once the client has established connection with the server, the client sends a single line of text, For e.g. "This is team 8" and the message is checked on the server.

Server side

A terminal window with a dark background. The title bar shows a red icon and the path "/mnt/g/My Drive/ECEN 602/MP/ECEN602-socket_programming/Original_work". The terminal text shows the user running './server.out 1234', the server starting, and receiving a message from the client.

```
root@LAPTOP-P9AM5HG8: /mnt/g/My Drive/ECEN 602/MP/ECEN602-socket_programming/Original_work  
root@LAPTOP-P9AM5HG8:/mnt/g/My Drive/ECEN 602/MP/ECEN602-socket_programming/Original_work# ./server.out 1234  
Socket started  
Server is at IP: 0.0.0.0  
Read from client: This is team 8
```

Client side

```
root@LAPTOP-P9AM5HG8: /mnt/g/My Drive/ECEN 602/MP/ECEN602-socket_programming/Original_work
root@LAPTOP-P9AM5HG8:/mnt/g/My Drive/ECEN 602/MP/ECEN602-socket_programming/Original_work# ./client.out 1234
Connection to the server successful
This is team 8
Read from Server: This is team 8
```

2. line of text the maximum line length without a newline In this test case we build two scenarios. In the first scenario(2.1), the buffer size is limited to 5 on the client size and we send a message of size 15 from client too server. While in the second scenario (2.2), we limit the buffer size on the server side to 5, and we send a message of size 15 from client to server.

Server side terminal for 2.1

```
ibrahim_shahbaz@LAPTOP-EGIHJND:/mnt/c/Users/ibrah/OneDrive/Desktop/PhD- TAMU/Year 1/FALL 2024/Comp Networks - ECEN 602/Assignments/MP1/ECEN602-socket_programming/Original_work$ make echos PORT=1234
./server.out 1234
Socket server started
Server is at IP: 0.0.0.0
Accepted a new connection
Total clients connected: 1
Number of characters read = 4
Read from client 1: 1234
Number of characters read = 4
Read from client 1: 5678
Number of characters read = 4
Read from client 1: 9101
Number of characters read = 4
Read from client 1: 1112
```

Client side terminal for 2.1

```
ibrahim_shahbaz@LAPTOP-EGIHJND:/mnt/c/Users/ibrah/OneDrive/Desktop/PhD- TAMU/Year 1/FALL 2024/Comp Networks - ECEN 602/Assignments/MP1/ECEN602-socket_programming/Original_work$ make echo SERVER_ADDR=127.0.0.1 PORT=1234
./client.out 127.0.0.1 1234
Connection to the server successful
123456789101112
Recieved from Server: 1234
Recieved from Server: 5678
Recieved from Server: 9101
Recieved from Server: 1112
```

Server side terminal for 2.2

```
ibrahim_shahbaz@LAPTOP-EGIHJND:/mnt/c/Users/ibrah/OneDrive/Desktop/PhD- TAMU/Year 1/FALL 2024/Comp Networks - ECEN 602/Assignments/MP1/ECEN602-socket_programming/Original_work$ make echos PORT=1235
./server.out 1235
Socket server started
Server is at IP: 0.0.0.0
Accepted a new connection
Total clients connected: 1
Number of characters read = 5
Read from client 1: 12345.0
Number of characters read = 5
Read from client 1: 67891.0
Number of characters read = 5
Read from client 1: 01112.0
```

Client side terminal for 2.2

```
ibrahim_shahbaz@LAPTOP-EGIHJND:/mnt/c/Users/ibrah/OneDrive/Desktop/PhD- TAMU/Year 1/FALL 2024/Comp Networks - ECEN 602/Assignments/MP1/ECEN602-socket_programming/Original_work$ make echo SERVER_ADDR=127.0.0.1 PORT=1235
./client.out 127.0.0.1 1235
Connection to the server successful
123456789101112
Recieved from Server: 12345.0
```

3. line with no characters and EOF

Server side

```
ibrahim_shahbaz@LAPTOP-EGIHJND:/mnt/c/Users/ibrah/OneDrive/Desktop/PhD- TAMU/Year 1/FALL 2024/Comp Networks - ECEN 602/Assignments/MP1/ECEN602-socket_programming/Original_work$ ./out_server.o 9999
Socket server started
Server is at IP: 0.0.0.0
Accepted a new connection
Total clients connected: 1
TCP_FIN packet received. Client disconnected
Exiting child process
```

Client side

```
ibrahim_shahbaz@LAPTOP-EGIHJND:/mnt/c/Users/ibrah/OneDrive/Desktop/PhD- TAMU/Year 1/FALL 2024/Comp Networks - ECEN 602/Assignments/MP1/ECEN602-socket_programming/Original_work$ ./out_client.o 9999
Connection to the server successful
Exiting... Closing the Client socket Due to EOF.
Closing the connection
```

4. client terminated after entering text

Server side terminal

```
ibrahim_shahbaz@LAPTOP-EGIHJND:/mnt/c/Users/ibrah/OneDrive/Desktop/PhD- TAMU/Year 1/FALL 2024/Comp Networks - ECEN 602/Assignments/MP1/ECEN602-socket_programming/Original_work$ ./out_server.o 9999
Socket server started
Server is at IP: 0.0.0.0
Accepted a new connection
Total clients connected: 1
TCP_FIN packet received. Client disconnected
Exiting child process
```

Client side terminal

```
ibrahim_shahbaz@LAPTOP-EGIHJND:/mnt/c/Users/ibrah/OneDrive/Desktop/PhD- TAMU/Year 1/FALL 2024/Comp Networks - ECEN 602/Assignments/MP1/ECEN602-socket_programming/Original_work$ ./out_client.o 127.0.0.1 9999
Connection to the server successful
exit
Closing the Client socket due to Termination by user.
Closing the connection
```

5. Three clients connected to the server. In the last test case, we test our server for handling requests from 3 different clients at a time. The server is able to differentiate which client is sending the echo request.

Server side terminal

```
ibrahim_shahbaz@LAPTOP-EGIHJND:/mnt/c/Users/ibrah/OneDrive/Desktop/PhD- TAMU/Year 1/FALL 2024/Comp Networks - ECEN 602/Assignments/MP1/ECEN602-socket_programming/Original_work$ ./out_server.o 127.0.0.1 1234
Socket server started
Server is at IP: 0.0.0.0
Accepted a new connection
Total clients connected: 1
Number of characters read = 13
Read from client 1: Howdy from C1
C1 you are all set
Accepted a new connection
Total clients connected: 2
Number of characters read = 13
Read from client 2: Hello from c2
Accepted a new connection
Total clients connected: 3
Number of characters read = 20
Read from client 3: Hi server this is C3
Number of characters read = 0
TCP_FIN packet received. Client disconnected
Exiting child process
Number of characters read = 0
TCP_FIN packet received. Client disconnected
Exiting child process
```

Client side terminals

```
ibrahim_shahbaz@LAPTOP-EGIHJND:/mnt/c/Users/ibrah/OneDrive/Desktop/PhD- TAMU/Year 1/FALL 2024/Comp Networks - ECEN 602/Assignments/MP1/ECEN602-socket_programming/Original_work$ ./out_client.o 127.0.0.1 1234
Connection to the server successful
Howdy from C1
Recieved from Server: Howdy from C1
exit
Closing the Client socket due to Termination by user.
Closing the connection
ibrahim_shahbaz@LAPTOP-EGIHJND:/mnt/c/Users/ibrah/OneDrive/Desktop/PhD- TAMU/Year 1/FALL 2024/Comp Networks - ECEN 602/Assignments/MP1/ECEN602-socket_programming/Original_work$ 
ibrahim_shahbaz@LAPTOP-EGIHJND:/mnt/c/Users/ibrah/OneDrive/Desktop/PhD- TAMU/Year 1/FALL 2024/Comp Networks - ECEN 602/Assignments/MP1/ECEN602-socket_programming/Original_work$ ./out_client.o 127.0.0.1 1234
Connection to the server successful
Hello from c2
Recieved from Server: Hello from c2
Exiting... Closing the Client socket Due to EOF.
Closing the connection
ibrahim_shahbaz@LAPTOP-EGIHJND:/mnt/c/Users/ibrah/OneDrive/Desktop/PhD- TAMU/Year 1/FALL 2024/Comp Networks - ECEN 602/Assignments/MP1/ECEN602-socket_programming/Original_work$ 
ibrahim_shahbaz@LAPTOP-EGIHJND:/mnt/c/Users/ibrah/OneDrive/Desktop/PhD- TAMU/Year 1/FALL 2024/Comp Networks - ECEN 602/Assignments/MP1/ECEN602-socket_programming/Original_work$ ./out_client.o 127.0.0.1 1234
Connection to the server successful
Hi server this is C3
Recieved from Server: Hi server this is C3
```

Contributors

Shubham Santosh Kumar(UIN: 835008989)

1. Developed server code.

2. Added capabilities to handle multiple clients.
3. Contributed to documentation of code.

Ibrahim Shahbaz(UIN: 236001615)

1. Developed Client code and documented it.
2. Worked on handling test cases.