IMPLEMENTING TWO WAY CHAT APPLICATION USING TCP/IP

Heli Bhalodiya

1Student, MCA Department, Bhavnagar

Abstract: Now days various chat software are very popular. There is two type of chat is available 1.one way 2. Two way. here we coded the duplex chat code for two way communication. This code is use TCP/IP protocol to submit and received the message.

Key words: Duplex, chat, TCP/IP

I. INTRODUCTION
Chat communication is the process of exchanging messages between two systems continuously. Anyone can break the communication. Both systems come with the following same responsibilities.[1,4]
1. Reading from keyboard. Uses an input stream connected to System
2. Sending data to the other system what is read from keyboard. Uses an output stream method of Socket.
3. Receiving data from the other system. Uses an input stream method of Socket.
As the responsibilities are same, both client and server programs contain the same stream objects and same code. The order of using stream objects varies in the while loop.

II. TCP/IP
TCP/IP is a two-layer program. The higher layer, Transmission Control Protocol, manages the assembling of a message or file into smaller packets that are transmitted over the Internet and received by a TCP layer that reassembles the packets into the original message. The lower layer, Internet Protocol, handles the address part of each packet so that it gets to the right destination. Each gateway computer on the network checks this address to see where to forward the message. Even though some packets from the same message are routed differently than others, they'll be reassembled at the destination.[1,2]
TCP/IP uses the client/server model of communication in which a computer user (a client) requests and is provided a service (such as sending a Web page) by another computer (a server) in the network. TCP/IP communication is primarily point-to-point, meaning each communication is from one point (or host computer) in the network to another point or host computer. TCP/IP and the higher-level applications that use it are collectively said to be "stateless" because each client request is considered a new request unrelated to any previous one ( unlike ordinary phone conversations that require a dedicated connection for the call duration). Being stateless frees network paths so that everyone can use them continuously. [2]

III. METHODOLOGY
Steps to making program at server side are as follows
• Include the necessary header files.
• Create a socket using socket function with family AF_INET, type as SOCK_STREAM.
• Initialize server address to 0 using the bzero function.
• Assign the sin_family to AF_INET, sin_addr to INADDR_ANY, sin_port to dynamically assigned port number.
• Bind the local host address to socket using the bind function.
• Listen on the socket for connection request from the client.
• Accept connection request from the Client using accept function.
• Fork the process to receive message from the client and print it on the console.
• Read message from the console and send it to the client.

Steps to making program at client side are as follows
• Include the necessary header files.
• Create a socket using socket function with family AF_INET, type as SOCK_STREAM.
• Initialize server address to 0 using the bzero function.
• Assign the sin_family to AF_INET.
• Get the server IP address and the Port number from the console.
• Using gethostbyname function assign it to a hostent structure, and assign it to sin_addr of the server address structure.
• Request a connection from the server using the connect function.
• Fork the process to receive message from the server and print it on the console.
• Read message from the console and send it to the server.

IV. RESULTS

Results at Server:
(Host Name:Root1)
[root@localhost 4ita33]# vifserver.c
[root@localhost 4ita33]# cc fserver.c
[root@localhost 4ita33]# ./a.out
Server is running....... Enter the input data:
Received message from the client:hi
how are u
Data sent… Enter the input data:
Received message from the client:i am fine

Results at Client:
(Host Name:Root2)
[root@localhost 4ita33]# vifclient.c
[root@localhost 4ita33]# cc fclient.c
[root@localhost 4ita33]# ./a.out 127.0.0.1
Enter the input data:
hi
Data sent… Enter the input data:
Received message from the server:how are u
i am fine
Data sent… Enter the input data:

V. INFERENCE
Thus the chat application full duplex communication is established by sending the request from the client to the server, server gets the message and gives response to the client and prints it.
VI. CONCLUSION

There are two hosts, Client and Server. Both the Client and the Server exchange message i.e. they send messages to and receive message from the other. There is a two way communication between them.

To implement a full duplex application, where the Client establishes a connection with the Server. The Client and Server can send as well as receive messages at the same time. Both the Client and Server exchange messages.

REFERENCES

1. http://searchnetworking.techtarget.com/definition/TCP-IP