

Chat Client Server

Recently I got introduced myself with socket programming. I read through many books and articles to get more insight into the subject. I thought why not share it with the world what I have learned. For this purpose I created a simple console based chat application. I hope this will be a good reference code for you C++ programming. I am going to introduce the following concepts to you here.

- Sockets.
- Multi threading.

What is Chat Client Server?

The demo project given along with this article consists of two console-based applications.

Chat Server

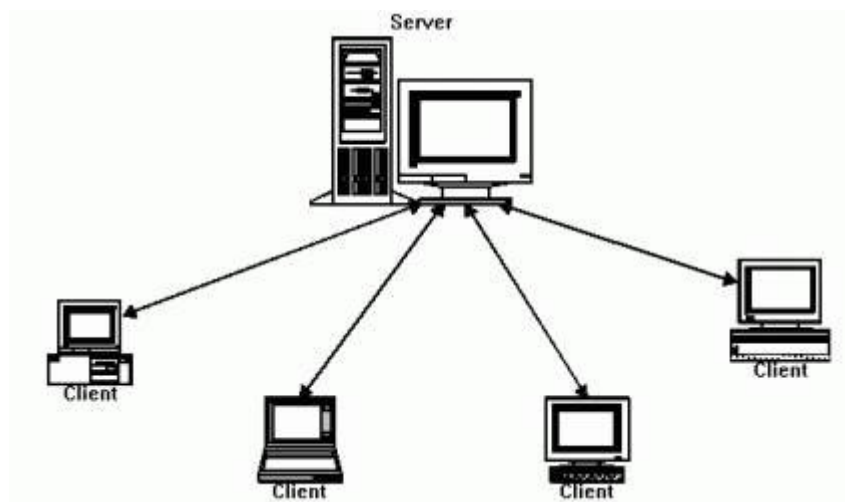
Chat server is an application, which does the following things.

- Listens for incoming calls from clients. Client running in any pc can connect to the server if IP address of the pc where server is running is known.
- Listens for messages from all the connected clients.
- Broadcast the message from clients to all the clients connected to the server.
- You can also type-in messages in the server, which will be broadcasted to all the clients.

Chat Client

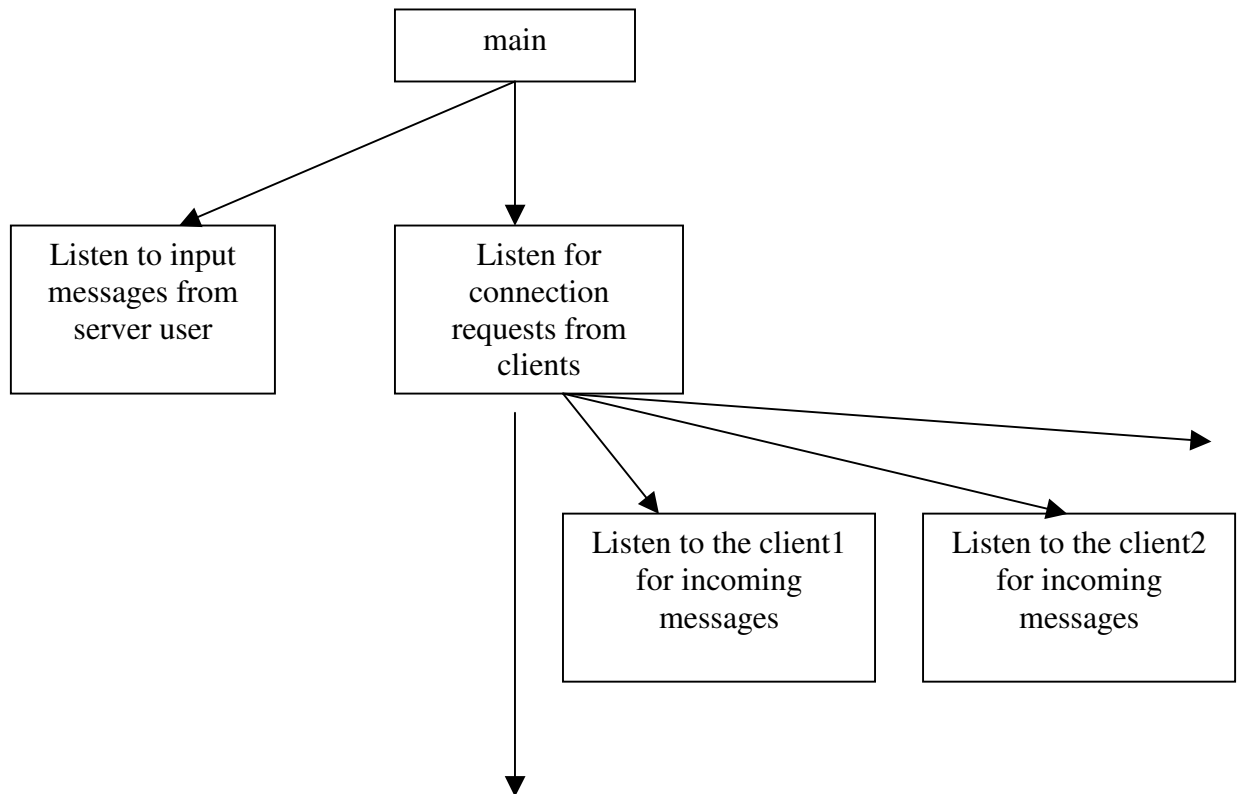
Chat client does the following things.

- Send messages to server as well as all the connected the clients.
- View the messages from all the clients and server.



Server Design

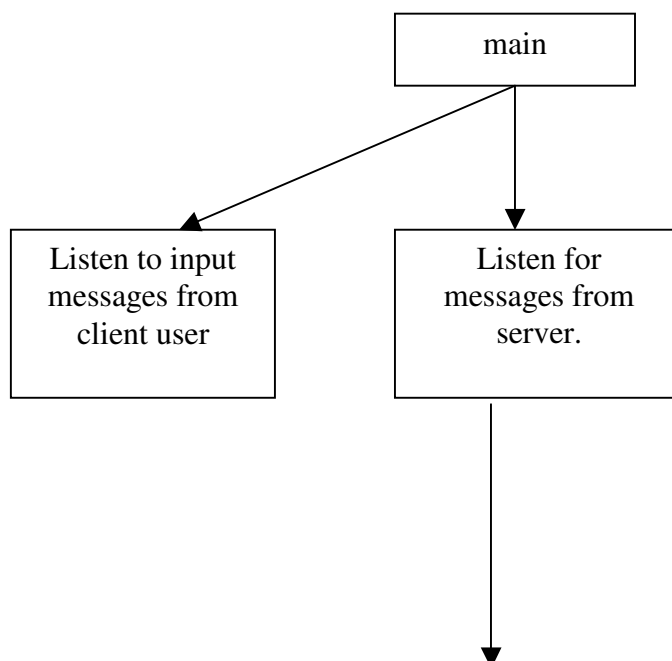
Following diagram explains the operations performed by the server application.



As shown in the diagram above, there are two threads always running in the server application. First thread handles the message inputs to the server if any. Second thread wait for incoming calls to the server. A new thread will be created for each client connection, which will be terminated when client disconnects from the server. That means if there are three clients connected to the server, 5 threads will be running in the server. The port used in the client side is 8084(In digital display, 8084 corresponds to BOBY). This has been hard coded in the server as well as clients.

Client design

Client design is very simple compared to server.



In Client, only two threads are running. First one waits for messages from server. Second one waits for user inputs.

Note: Update server.ini along with the client application with IP address of the server machine. If you don't have only one pc, still you can run this application. Specify the IP address as 127.0.0.1(local host).

Sockets

The functions used for sockets are declared in winsock.h and you need to link your application with the static library ws2_32.lib.

The following are the main functions required for socket programming.

```
SOCKET SClient; // creates a handle to a socket. SOCKET is nothing but an unsigned
integer, which represents a socket.

SClient = socket(AF_INET, SOCK_STREAM, 0); // Creates a socket.

connect(SClient, (struct sockaddr*)&server, sizeof(server)); // Connects to a socket
where server is struct of type sockaddr_in.

The structure sockaddr_in contains the following details.

struct sockaddr_in {
    short        sin_family;
    u_short      sin_port;
    struct       in_addr sin_addr;
    char         sin_zero[8];
};

You need to fill in the details of sockaddr_in before using it with
"connect" function.

send(sSocket, sMessage.c_str(), sMessage.size()+1, 0); // sends a message to the
socket sSocket.

recv(sSocket, acRetData, 4096, 0); // receive a message from the socket sSocket.
```

Go through the demo project to get a better overview about sockets and how it can be used in applications.

Summary

Sockets can be used for communication of information between applications running in different PCs. Please write to me for