Documentation

Terms and Functions used:

Node: In computer science a node is a basic unit of a data structure, such as a tree data structure. Nodes contain data and also may link to other nodes usually by a pointer.

API (Application Programming Interface) : In simple terms, API is a software intermediary that establishes communication between two programs.

Winsock - The library used for handling sockets on the Windows OS.

Socket- A socket is an endpoint of a two way communication. It is bound to an ip address and a port to establish the communication.

TCP - Transmission control protocol is a standard that defines how to establish and maintain a network conversation through which application programs can exchange data. TCP has a three part handshake process where the connection is first acknowledged, setup and then closed (once we are done). The other alternative to TCP is UDP which is faster, but less secure and reliable when transferring data across the network.

Client - The “client” is the one who requires a specific resource or service.

Server - The one providing the resource of service to the client.

AF\_INET - It is an address family that is used to designate the type of addresses that your socket can communicate with. AF\_INET represents the IPv4 addresses.

AF\_INET6 - The same as AF\_INET, but for IPv6 address.

htons()-This function is used to make sure that our data is in the correct order. The scientific term for this correct order is big-endian.

A computer can only understand data in the big-endian format.

inet\_ntop(): This takes the address(from the address structure) and converts it into a string.

send() - The function used to send out data to other clients/server.

recv() - The function used to receive data over the network.

SOCK\_STREAM() - A reliable two way communication scheme. Sending data in this format, it will arrive in the same order it was sent. Uses the TCP connection format to achieve this. (We are gonna be using this in our program)

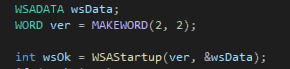
SOCK\_DGRAM() - Known as connectionless sockets, using the connection type UDP (User datagram protocol). No connection required. Unlike TCP, no open connection is maintained. Packet is just given a destination and sent.

Code Explanation:

Here are some key points to explain what exactly is happening. Mostly the server is discussed.

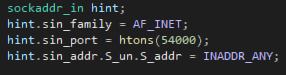
Note: The features are not discussed here. The basic working is discussed.

The project consists of a server and a client program.

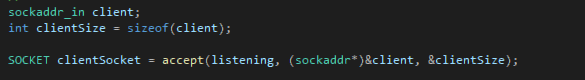
* In order to create a socket and to establish TCP, we need to initialize the winsock library.
* Now we will create our listening socket.



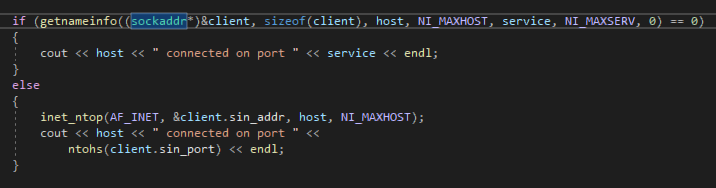
* We need to connect our socket to the port and the ip address. To do that we have created a structure containing information regarding this which will be used as an argument of the bind() function.



* Now we are going to attempt connection between client and server. For that we declare a structure which is supposed to contain the client's information and we use a function accept() which makes the server wait until the client enters their local host/ip address.



* The client has a parallel function to accept() called connect(). Now the client and server are connected.
* The getnameinfo() function translates a socket address to a node name and service location which is a requirement in socket programming. If getnameinfo is unable to do so then the program will use inet\_ntop() which has similar functionality to getnameinfo(). This was the last step in establishing a one way communication between server and client.



* Now we can echo messages back and forth in a loop with the send() and recv() functions.
* Lastly we have closed the client socket and broken the communication and shut down winsock with WSAcleanup(). Although this is not necessary, it certainly is a good practice.

**Features:**

**1)Filter function:**

We have created a function that filters out any swear words in the chat. For implementation of this, we are comparing the message with the contents of a file. This file is included with the project and can be edited as long as the format of the words stays the same.

**2)File transfer:**

A feature for file transferring(txt files) has been added. The client can give the file command and then select the file by writing its name. This file will be copied over to the server.

**3)User names:**

The client can select a username and it will be displayed along with their messages in the server console.

**4)Chat log:**

A chat log is created every session and it displays all of the messages of that particular session and the past ones. With each message is written the time it was sent on.