## Code Documentation:

First the code creates a main\_socket that listens for incoming connections on the port number that has been designated to it.

Then all the file descriptors that are currently active are added to readfds using FD\_SET(). After that the maximum value of the file descriptors is obtained for using the select() call. After the select call returns we check which file descriptor is now set using FD\_ISSET().

If it is the main\_socket, it denotes an incoming connection request. So we accept the connection using accept() and read the message sent using read()

We extract the details of the sender from the static user table list and add the details to the active connections list (conn\_list) and the file descriptor list (fd list). Finally we display the incoming message.

If it is the fd = 0 (STDIN) we first read from the console using read(). From the message read, we extract the name of the receiver (string before '/') and find his details from the static user table. Then we check if a connection to the user already exists among the list of active connections in conn info.

- If it exists we directly write the message into the socket corresponding to the recipient
- If it does not exists, we create a new socket and send a new connection request to the recipient using connect(). Once the request is accepted, the message is sent to the recipient using write()

If it is none of the above two, it indicates an incoming message from an already existing connection. So we find exactly which client has sent the message using the FD\_ISSET() call over all active file descriptors. Then we read the message from the corresponding condition using read().

- If read returns something <0, it implies that the client has closed the connection. So we free the corresponding entries in the conn\_list and fd\_list and close the connection from our end also.
- If it returns > 0, it means that it has received a message from a client. We extract the name of the client and the element in conn\_list corresponding to the sender and update its last active time.

Everytime we also check for timeouts. So we iterate over the list of active connections and find if for any transaction the value of current\_time - last\_time > TIMEOUT interval. If found we close that particular connection and free its entries in fd\_list and conn\_list.

## How to run:

It has enclosed 5 c++ codes corresponding to each client.

Run make command to compile all the codes and generate the executables. Run the executables using ./p2p (executable number).

- ./p2p1 is the first client (Sag)
- ./p2p2 is the second client (Roy)
- ./p2p3 is the third client (Avi)
- ./p2p4 is the fourth client (Sas)
- ./p2p5 is the fifth client (Soh)

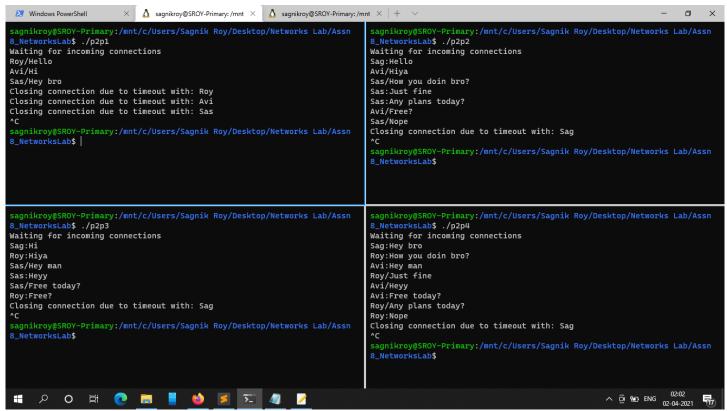
## Sample Input and Output:

In this sample input/output, we run using 4 clients, namely "Sag", "Roy", "Avi" and "Sas". Sag initiates by sending a Hello type message to all. The other three send messages to each other, but not Sag, letting this connection run to timeout. This also checks if the timeout feature is working correctly, along with the core functionality of enabling communication. The terminal results are copied as under:

```
sagnikroy@SROY-Primary:/mnt/c/Users/Sagnik Roy/Desktop/Networks
Lab/Assn8 NetworksLab$ ./p2p1
Waiting for incoming connections
Roy/Hello
Avi/Hi
Sas/Hey bro
Closing connection due to timeout with: Roy
Closing connection due to timeout with: Avi
Closing connection due to timeout with: Sas
sagnikroy@SROY-Primary:/mnt/c/Users/Sagnik Roy/Desktop/Networks
Lab/Assn8 NetworksLab$
sagnikroy@SROY-Primary:/mnt/c/Users/Sagnik Roy/Desktop/Networks
Lab/Assn8 NetworksLab$ ./p2p2
Waiting for incoming connections
Sag:Hello
Avi/Hiya
Sas/How you doin bro?
Sas: Just fine
Sas: Any plans today?
Avi/Free?
Sas/Nope
Closing connection due to timeout with: Sag
^C
sagnikroy@SROY-Primary:/mnt/c/Users/Sagnik Roy/Desktop/Networks
Lab/Assn8 NetworksLab$
sagnikroy@SROY-Primary:/mnt/c/Users/Sagnik Roy/Desktop/Networks
Lab/Assn8 NetworksLab$ ./p2p3
Waiting for incoming connections
Sag:Hi
Roy: Hiya
Sas/Hey man
Sas: Heyy
Sas/Free today?
Roy:Free?
Closing connection due to timeout with: Sag
^C
sagnikroy@SROY-Primary:/mnt/c/Users/Sagnik Roy/Desktop/Networks
Lab/Assn8 NetworksLab$
sagnikroy@SROY-Primary:/mnt/c/Users/Sagnik Roy/Desktop/Networks
Lab/Assn8 NetworksLab$ ./p2p4
Waiting for incoming connections
Sag: Hey bro
Roy: How you doin bro?
```

Avi:Hey man
Roy/Just fine
Avi/Heyy
Avi:Free today?
Roy/Any plans today?
Roy:Nope
Closing connection due to timeout with: Sag
^C
sagnikroy@SROY-Primary:/mnt/c/Users/Sagnik Roy/Desktop/Networks
Lab/Assn8 NetworksLab\$

## Screenshot of the terminal corresponding to above sample i/o:



In this picture:
Top Left: User Sag
Top Right: User Roy
Bottom Left: User Avi
Bottom Right: User Sas

By:

Sagnik Roy Debajyoti Kar