Report Lab 2

DVA218

Done by: Arafat Sulaiman & Jacob Johansson

A. Overview

A server and one or more clients are included in this program. Through sockets, the server process allows one or more clients to connect to it. When the server and the client(s) establish a connection, they can send and receive data between each other, client side can take input from the user and send it to the server, the server will then print it out on the screen. Since more than one client can be linked to the server, when a new client connects, the server uses broadcasting to send a message to the other clients telling that there is a new client connecting with it's specific IP-address. Using the IP-address, the server is able to ban/stop the client with that specific IP-address to connect.

B. Description

The server creates a socket and names and addresses it. The socket is placed in a state where it is listening for inbound connections. When the client is asking for connection, the server sees the data on the socket and accepts the request. At this moment the server is not expecting any messages on this socket, that means that a client trying to connect. The accept() function generates a socket, which is used for communicating with the client. The server sees the client's IP-address, and chooses to accept it or reject it. It looks if the IP-address is banned or not. With the help of the function strcmp(), that compares two strings, if the server keeps the connection, the server sends a message like ”I hear you dude”. If not, the server sends a message like ”You are banned” to the client. The socket then closes which means the connection stops. The server has 2 file sets, active-set and read-set, the active-set has file descriptors of the active sockets (The server sockets, and the client sockets). The server uses broadcast to all other clients when a new client has been connected, this is done by going through all the sockets in the active-set and sending them a message about the new connection. Client starts a socket and names it after the hostname-argument, the user input, and the port number. When the server accepts the connection, a thread is created to listen to the new input on the socket. The read data is printed out on the screen. The main thread waits for input that will be sent via the socket connected to the server. When user inputs ”quit”, client closes the socket, and quits.

C. Results

- Output connection (Two clients)

[waiting for connections...]
Server: Connect from client 127.0.0.1
>Incoming message: hej
Server: Connect from client 127.0.0.1
>Incoming message: hej2

- Output of when the IP-address is banned in the server.

Type something and press [RETURN] to send it to the server.
Type 'quit' to nuke this program.
>Server: You are banned

- Output of when first client connect and the second client connect.

Type 'quit' to nuke this program.
>Server: I hear you, dude...
>Server: Client 127.0.0.1 connected

- Output of when second client connect to the server

Type something and press [RETURN] to send it to the server.
Type 'quit' to nuke this program.
>Server: I hear you, dude...

- Output of the server, when banned client wants to connect

[waiting for connections...]
Server: client 127.0.0.1 rejected