TCP-based-chat

A TCP-based chat program composed of a client for the end users and a server program which will be used to connect clients so that they can exchange messages.

README

Name: Shuli He
Email: she77@ucsc.edu
Using C++

Files

bin - compiled program
doc - report: design details and some test
src - source file

Usage

Use command 'make' (Makefile) to compile the source code.(Using g++)

Run:
client:
./client <ipaddress> <port> <id>
server:
./server <listen-port>

Design

Server - Client

1. Client will send a connection message to server with its id at the beginning.
2. When the server received a connection from client it will create a new thread to handle the command from this client.

3. /list: The server maintains a list of the waiting client when client call /list the server will send this list.
4. /wait: The client will create a new port to listen the chat connection. First will use select to accept () in a new listen thread then it send the new port to server and server will add this client’s information to the waiting list.
5. /connect <id>: When a client request connection the server will search the waiting list and return the result back to the client.
6. /left: When a client stops waiting it will send /left to server and the server can remove it from waiting list.
   Client – Client
7. /connect <id>: A client get the information from server it can connect to another client by the new port number and start chat. It will start a message receive thread to recv () and type send message in the main thread.
8. Ctrl-c: the client can use ctrl-c to leave the waiting state and chatting state with a state change.
9. /quit will always exit the program.

Potential problem:

1. There might some memory leak with the socket or thread.
2. Client might cannot connect to an old port when the waiting one update a new one.
3. Ctrl-c sometimes has some issue.
4. Connection between client might lost.
5. Waiting list might no clear the leave client in some cases.

Test result in report...