

Program Design

The client-server program is designed to facilitate communication between multiple clients and a server using a custom protocol over a network. The program follows a client-server architecture, where clients can send requests to the server, and the server processes these requests and sends back appropriate responses.

Server Code explanation

Class Definition: TCPServer

TCPServer: Represents the TCP server class. It has a constructor (TCPServer), a destructor (~TCPServer), and two member functions (Start, HandleClient). It also contains some private members such as serverSocket (the server's socket), serverAddr (server's address information), clientAddr (client's address information), clientAddrLen (length of client's address), and clientThreads (a vector to store client threads).

Constructor: TCPServer::TCPServer

The constructor sets up the server socket, binds it to the specified port, and starts listening for incoming connections. If any step fails, it prints an error message and exits.

Destructor: TCPServer::~~TCPServer

The destructor closes the server socket when the object is destroyed.

Start Function

The Start function runs in an infinite loop, accepting incoming connections. For each new connection, it spawns a new thread (HandleClient) to handle communication with the client.

HandleClient Function

The HandleClient function runs in a separate thread for each connected client. It continuously receives messages from the client, processes them, and sends back appropriate responses. If the client sends "bye," the server replies with "Goodbye" and closes the connection.

main Function

The main function creates a TCPServer object, listens for incoming connections on port 6000, and starts the server using the Start method.

Client code explanation

Class Definition: TCPClient

TCPClient: Represents the TCP client class. It has a constructor (TCPClient), a destructor (~TCPClient), and three member functions (Connect, Start, SendMessage, ReceiveMessage). It also contains private members such as clientSocket (the client's socket), serverAddr (server's address information), and message (a character array for storing messages).

Constructor: TCPClient::TCPClient

The constructor sets up the TCP socket and connects to the server specified by the provided IP address (serverIP) and port number (port). If any step fails, it prints an error message and exits.

Destructor: TCPClient::~~TCPClient

The destructor closes the client socket when the object is destroyed.

SendMessage and ReceiveMessage Functions

SendMessage: Takes user input, sends the message to the server using send.

ReceiveMessage: Receives the server's response using recv and displays it.

Connect and Start Functions

Connect: Repeatedly sends messages to the server until the user enters "exit" or receives a "Goodbye" response from the server.

Start: Calls Connect and then closes the client socket.

main Function

The main function creates a TCPClient object, connects to the server at IP address "127.0.0.1" and port 6000, and starts the client using the Start method.