

Dependencies:

Python 2.7

Run the server

start.sh [port number]

Chat server task:

This repository has the implementation of the centralised chat server using Sockets in Python. Chat server is based on the Tcp protocol, designed for stream connections between clients and server, with stream connections used to provide a call-back capability from client to server. This repository contains two files stest and the client file.

1. stest- this file contains the implementation of chat server.
2. Client-this file contains the implementation of clients.

Functionality of server

Server contains different functionalities:-

1. Server handles multiple clients using threads. Server connects with the client based on the address and assigns the port to client for transferring messages by including client to the chat room.
2. Server has the additional feature of authority to add a client in a multiple chat rooms. Also, Server can add multiple clients to single room as well as multiple clients to single room or into multiple rooms.
3. Server process the client request of client based on the connection and processes the chat request for joining the chat rooms based on the requests from the clients.
4. Server provides the functionalities of leaving the chat rooms as well.
5. Server communicates with the single or with multiple client by transferring messages using the input and output streams.
6. After successful communication server disconnects with the clients which were connected earlier.

Working of server

- Socket - Create a new communication endpoint for a specific transport protocol.
- Bind - Attach a local address to the newly created socket.
- Listen - Announce willingness to accept connections.
- Accept-

1. Block caller until a connection request arrives.

2. Create a new socket with the same properties as the original one and returns it to the caller.

3. Wait for another connection request on the original socket

- Close - Release the connection.

Working of Client:

- Socket - Create a new communication endpoint for a specific transport protocol, but bind is not necessary
- Connect
 1. Actively attempt to establish a connection, with the transport- level address
 2. Block the caller until a connection has been set up successfully
- Send - Send some data over the connection
- Receive - Receive some data over the connection
- Close - Release the connection