CLIENT SOCKET

COMPUTER NETWORK

Time: 60 mins

Introduction

In this class, the student/s create the client sockets to enable chat functionality between end user clients.

New Commands Introduced

- `client.send(message.encode('utf-8'))` Sends the encoded message from the client
- `clients.append(conn)` `Append the new client connection`
- `Thread(target=clientThread,args=(conn, addr))` `Creates a thread entity with the target function and argument list`
- `conn.recv(2048).decode('utf-8')` `Receives the encoded message`
- `message = '{}: {}'.format(nickname, input(''))` Creates a formatted string for the nickname and the input string

Vocabulary

- **Multitasking** involves working on two or more tasks simultaneously, switching back and forth from one thing to another, or performing a number of tasks in rapid succession.
- **A thread** is a sequence of such instructions within a program that can be executed independently of other code.
- **Broadcast messaging** is a one-to-many communication in which you create a list of contacts and then send them all the same message simultaneously.

Learning Objectives

Student/s should be able to:

- ***Recall*** achieving connection between server and client and sending messages using send() method.

- ***Explain*** creating threads to write and receive the messages between the server and the client.
- ***Demonstrate*** client - client chat between multiple users through the server.

Activities

1. **Class Narrative:** *(3 mins)*

    - Recall how the client and server could be connected using the socket programming.
    - Brief the student/s that the characters plan to design a chat window to enable chat between multiple users and it is possible only through the server.

2. **Concept Introduction Activity:** *(4 mins)*

    - Let the student/s undertake the explore-activity so the socket server can handle message requests from multiple clients parallelly.
    - Explain that the client sockets can be created to enable client-client chat functionality.
    - Using the slides, explain that the student/s will learn:
        - to add multiple clients to the server
        - to send messages from the client to the server
        - to enable client-client chat

3. **Activity 1: Add Multiple Clients to the Server***(14 mins)*

    Teacher Activity: *(7 mins)*

    - Relate website accessible by multiple users to server accessible to multiple clients. Give real life examples of chefs cooking multiple dishes at a time.
    - Explain how threading divides tasks and parallelly performs it.
    - Demonstrate creating a thread to receive messages on connected and save the users name.

    Student Activity: *(7 mins)*

    - Guide the students to name and connect the multiple clients to the server.

4. **Activity 2: Send Messages from the Client to the Server** *(12 mins)*

    Teacher Activity: *(7 mins)*

    - Recall and relate the chat apps. Introduce the format for the chat windows for multiple users.
    - Explain how to create a thread to store and send the username and input message to be displayed on the screen.

    Student Activity: *(6 mins)*

- Guide the student/s to create a thread to send the username and input message to the server from the clients.

5. **Activity 3: Enable Client-Client Chat** *(12 mins)*

   Student Activity: *(12 mins)*

   - Explain to the student(s) about iterating through each client to broadcast the message.
   - Guide the students to broadcast the message to all the clients.
   - Guide the students to create a thread to receive the message broadcasted.

6. **Introduce the Post class project:** *(2 min)*

   - Use multi threading to connect multiple clients to the server, send and receive messages.

7. **Test and Summarize the class learnings:** *(5 mins)*

   - Check for understanding through quizzes and summarize learning after respective missions.
   - Summarize the overall class learning towards the end of the class.

8. **Additional activities:**

   - Encourage the student/s to remove the clients from the list that are not connected to the server.
   - Encourage the student/s to debug the code to enable clients receive messages sent from other clients.

9. **State the Next Class Objective:** *(1 min)*

   - In the next class, student/s will learn to build the graphical interface for the chat app.

U.S. Standards:

CSTA: 2-AP-11, 2-AP-12, 2-AP-13, 2-AP-14, 2-AP-19

Links Table		
Activity	**Activity Name**	**Link**
Class Presentation	Client Socket	https://s3-whjr-curriculum-uploads.whjr.online/6ff009c9-49da-4ad1-8e2c-a7b8242d9b2e.html

Explore Activity	Client Socket	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-SAS-BP
Teacher Activity 1	Add Multiple Clients to the Server	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-TAS-BP
Teacher Reference: Teacher Activity 1 Solution	Add Multiple Clients to the Server	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-TAS
Student Activity 1	Add Multiple Clients to the Server	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-SAS-BP
Teacher Reference: Student Activity 1 Solution	Add Multiple Clients to the Server	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-SAS
Teacher Activity 2	Send Messages from Client to Server	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-TAS-BP
Teacher Reference: Teacher Activity 2 Solution	Send Messages from Client to Server	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-TAS
Student Activity 2	Send Messages from Client to Server	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-SAS-BP
Teacher Reference: Student Activity 2 Solution	Send Messages from Client to Server	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-SAS
Student Activity 3.1	Enable Client-Client Chat	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-SAS-BP
Teacher Reference: Student Activity 3.1 Solution	Enable Client-Client Chat	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-SAS
Student Activity 3.2	Enable Client-Client Chat	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-SAS-BP
Teacher Reference: Student Activity 3.2 Solution	Enable Client-Client Chat	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-SAS
Student's Additional Activity 1	Remove the Disconnected Clients	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-SAS-BP
Teacher Reference: Student's Additional Activity 1 Solution	Remove the Disconnected Clients	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-SAS
Student's Additional Activity 2	Debug the Code	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-SAS-BP
Teacher Reference: Student's Additional Activity 2 Solution	Debug the Code	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-SAS
Post Class Project	Chat with Multiple Clients	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-PCP-BP
Teacher Reference: Post Class Project Solution	Chat with Multiple Clients	https://github.com/Tynker-Computer-Networks/TNK-M14-C110-PCP