Server Socket

SOCKET PROGRAMMING

Time: 60 mins

Introduction

In this class, the student/s will learn how to write a server socket and send messages to client sockets.

New Commands Introduced

- socket()   : Creates the sockets (required on both server end and client end)

- bind()   : Binds the socket to the ip address and port specified

- listen()   : Enables the server to accept connections

- connect()   : Connects to a remote address specified as parameter

- accept()   : Accepts a connection. It returns a pair of values (conn, address)

- close()   : Closes the connection and marks the socket as closed

Vocabulary

- **Socket** is one endpoint of a two way communication link between two programs running on the network.
- **Stream** refers to the data that is transmitted when the content is consumed.

Learning Objectives

Student/s should be able to:
- *Recall* how to transfer a file from one device to another using FTP server.
- *Explain* sockets and its usage to establish a data transfer connection between the server and a client.
- *Demonstrate* how to connect server socket and client socket and send messages to and fro.

Activities

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

   - Introduce the hook where the game players find a need to chat among themselves. Sarah and friends decide to solve this by creating a chatroom in the game for players to communicate.

- Brief the student/s about how chat systems work using socket programming on server and client sides.

2. **Concept Introduction Activity:** *(5 mins)*

   - Brief the student/s that the workings of the chat application will be explored in three stages.

     - Create a Socket Server

     - Create Socket Clients

     - Create Chat App GUI.

   - Discuss the applications of Socket programming and how it ensures the message is delivered to the right device without a loss.

   - Let the student/s undertake the explore-activity to observe the phase-1 of the chat app i.e. Create a socket server and send messages to clients on successful connection.

   - Using the slides, explain that the student/s will learn:

     - Setup the Server

     - Connect with Client

     - Communicate with Client

3. **Activity 1: Setup the Server***(12 mins)*

   Teacher Activity: *(6 mins)*

   - Enquire students to figure out which transfer protocol between TCP and UDP should be used for a chat application.

   - Introduce student(s) to socket programming and basic flow of a TCP Socket.

   - Demonstrate how to deploy a socket server and bind it to the right IP and port number.

   Student Activity: *(6 mins)*

   - Guide the student/s to deploy the socket server and listen to a request.

4. **Activity 2: Connect with Client** *(10 mins)*

   Teacher Activity: *(5 mins)*

   - Encourage student(s) to observe the issue that the server stops as soon as it starts, and try to find a solution to this issue.

   - Explain to the student(s) that the server needs to listen to the request infinitely while the chat application is on.

- Demonstrate how to enable the server to listen to requests infinitely and send a message to the client on successful connection.

  Student Activity: *(5 mins)*

- Guide the student/s to listen and accept client requests on server and send a "Thank you" message on successful connection.

5. **Activity 3: Communicate with Client** *(15 mins)*

   Teacher Activity: *(7 mins)*

- Explain to the student(s) that a similar socket needs to exist on the client side to receive messages from the server.

- Demonstrate students how to receive the message on the socket and print it.

  Student Activity: *(8 mins)*

- Guide the student/s to write a code on the client socket to receive the message from the server and print the same on client end.

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

- Create a Quiz app server which sends a question to the client on a successful connection.

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 count the number of clients connected to the server and display it on the server.

- Encourage the student/s to send a message from the client to the server.

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

- In the next class, student/s will learn to create a multi-client chat application system.

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	Server Socket	https://s3-whjr-curriculum-uploads.whjr.online/1ee6d28a-d432-4539-bcaa-779149c63549.html
Explore Activity	Server Socket	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-SAS-BP
Teacher Activity 1	Setup the Server	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-TAS-BP
Teacher Reference: Teacher Activity 1 Solution	Setup the Server	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-TAS
Student Activity 1	Setup the Server	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-SAS-BP
Teacher Reference: Student Activity 1 Solution	Setup the Server	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-SAS
Teacher Activity 2	Connect with Client	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-TAS-BP
Teacher Reference: Teacher Activity 2 Solution	Connect with Client	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-TAS
Student Activity 2	Connect with Client	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-SAS-BP
Teacher Reference: Student Activity 2 Solution	Connect with Client	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-SAS
Teacher Activity 3	Communicate with Client	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-TAS-BP
Teacher Reference: Teacher Activity 3 Solution	Communicate with Client	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-TAS
Student Activity 3	Communicate with Client	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-SAS-BP
Teacher Reference: Student Activity 3 Solution	Communicate with Client	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-SAS
Student's Additional Activity 1	Display Client Count	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-SAS-BP
Teacher Reference: Student's Additional Activity 1 Solution	Display Client Count	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-SAS
Student's Additional Activity 2	Send Message to Server	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-SAS-BP
Teacher Reference: Student's Additional Activity 2 Solution	Send Message to Server	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-SAS
Post Class Project	Quiz App - 1	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-PCP-BP
Teacher Reference: Post Class	Quiz App - 1	https://github.com/Tynker-Computer-Networks/TNK-M14-C109-PCP

Project Solution		