COMPUTER PROJECT REPORT

Question:

There is a host who conducts the show and participants/players who provide answers.Let us say there are three players. The host has a list of 20 questions and correct answers with him. He randomly chooses one of the questions (makingn sure it is not a repeat of previous questions) and sends to all three players. The players receive the question, think about the answer for a while and press the buzzer. The first one to press the buzzer is given a chance to provide the answer. If the answer is correct, he is given 1 point, otherwise 0. The host then proceeds with the next question. The game stops when any player gets 5 points and that player is declared the winner.

Introduction

This deals with socket programming and multi-threading.
Sockets can be thought of as endpoints in a communication channel that is bi-directional, and establishes communication between a server and one or more clients. Here, we set up a socket on each end and allow a client to interact with other clients via the server. The socket on the server side associates itself with some hardware port on the server side. Any client that has a socket associated with the same port can communicate with the server socket.
A thread is sub process that runs a set of commands individually of any other thread. So, every time a user connects to the server, a separate thread is created for that user and communication from server to client takes place along individual threads based on socket objects created for the sake of identity of each client.
We will require two scripts to establish this chat room. One to keep the serving running, and another that every client should run in order to connect to the server.

A chat room is a designated virtual channel where users communicate with each other through the Internet, traditionally in plain text only. More recent developments in Web technology now allow the transmission of images and emoticons in a chat room as well. The term can mean online chatting, instant messaging and online forums using either synchronous or asynchronous conferencing. Some chat rooms require a username and password combination in order to log in or join a conversation, allowing for privacy among the users.

What does the code do?

Server Side Script

The server side script will attempt to establish a socket and bind it to an IP address and port specified by the user (windows users might have to make an exception for the specified port number in their firewall settings, or can rather use a port that is already open). The script will then stay open and receive connection requests, and will append respective socket objects to a list to keep track of active connections. Every time a user connects,
a separate thread will be created for that user. In each thread, the server awaits a message, and sends that message to other users currently on the chat. If the server encounters an error while trying to receive a message from a particular thread, it will exit that thread.

Client Side Script

The client side script will simply attempt to access the server socket created at the specified IP address and port. Once it connects, it will continuously check as to whether the input comes from the server or from the client, and accordingly redirects output. If the input is from the server, it displays the message on the terminal. If the input is from the user, it sends the message that the users enters to the server for it to be broadcasted to other users.

How to run the code

Open 4 terminals at the same time. One of the terminal acts as server and the other 3 act as clients.

Type server.py on the server terminal and client.py on the rest 3 client terminals.

Server gives questions randomly. The client that presses the buzzer first gets the chance to answer first. If the ansewr is wrong actiivate the buzzer option once again.

Else if answer is right increment the count value of that client. Before incrementing just check if the count variable of that client is >4. If so that client is declared as the winner and the game ends else just increment the counter variable and move to the next question.

Biblography

https://realpython.com/python-sockets/#background
https://www.geeksforgeeks.org/socket-programming-multi-threading-python/
https://www.geeksforgeeks.org/tcp-server-client-implementation-in-c/
https://www.geeksforgeeks.org/simple-chat-room-using-python/