Multithreaded Network Quiz Project Report

B22AI046,B22AI012

Contents

0.1 Requirements

To ensure smooth execution and avoid potential errors, it's recommended to run the following commands within the terminal of Visual Studio Code instead of the system command prompt:

1. Install the 'question' package using pip.
2. Install the 'home' package using pip.
3. Install PyQt5 using pip.

By executing these commands within the Visual Studio Code terminal, you can prevent any potential issues and ensure successful installation of the required packages for your project. Following the installation of the mentioned packages, the next step is to execute the following files in sequence:

1. Begin by running the 'server.py' file.
2. Subsequently, execute the 'client.py' file.

This order of execution ensures the proper functioning of the server-client communication setup for your application.

1 Problem Statement

1.1 Note

Before implementing, please ensure to replace the IP address in the `client.py` file with the IP address of your server to avoid any connectivity issues. This step is crucial for error-free communication between the clients and the server.

Overview: The task involves developing a multithreaded network quiz application where multiple clients connect to a server to participate in a quiz. The server coordinates the quiz, sends questions to clients, receives their responses, and calculates scores. This project not only demonstrates network communication but also showcases multithreading capabilities in Python.

Topics Covered:

1. **Computer Networks:**
   - **Socket Programming:** Utilizes socket programming to establish communication between the server and clients over a network. This involves creating TCP/IP socket connections for reliable data transfer.
   - **Network Protocols:** Implements a custom network protocol for transmitting quiz questions, options, and responses between the server and clients.
   - **Client-Server Architecture:** Demonstrates a client-server architecture where the server hosts the quiz and clients connect to participate.
2. **Multithreading:**
   - **Thread Management:** Utilizes the `threading` module to manage multiple client connections concurrently. Each client connection is handled in a separate thread, enabling parallel execution and preventing blocking.
   - **Concurrency Control:** Implements thread synchronization techniques such as locks (`thread_lock`) to ensure thread safety and prevent race conditions when accessing shared resources.
   - **Scalability:** The use of multithreading enhances scalability by allowing the server to handle multiple client connections simultaneously, thereby accommodating a larger number of participants.

Please Note:

- The file questionsBank.txt contains the questions that will be asked in the quiz program.
- Do not change the format of the document. Otherwise, it will lead to errors in the file I/O function of the program.
- Do not disturb these initial lines of note as well. The file I/O function is hardcoded to ignore these lines.
- Enter the questions in the format given.
- Each question can have only four options associated with it, of which only one can be the correct one.
- Do not change the relative address of the file.
- Do not change the file Name.

SAMPLE FORMAT [ONLY FOR REFERENCE]

- **Question:** Sample Question?
- **Options:**

  A.
  B. Correct: B
  C.
  D.

BEGIN

- **Question:** Which of the following is not a function of an Operating System?
- **Options:**

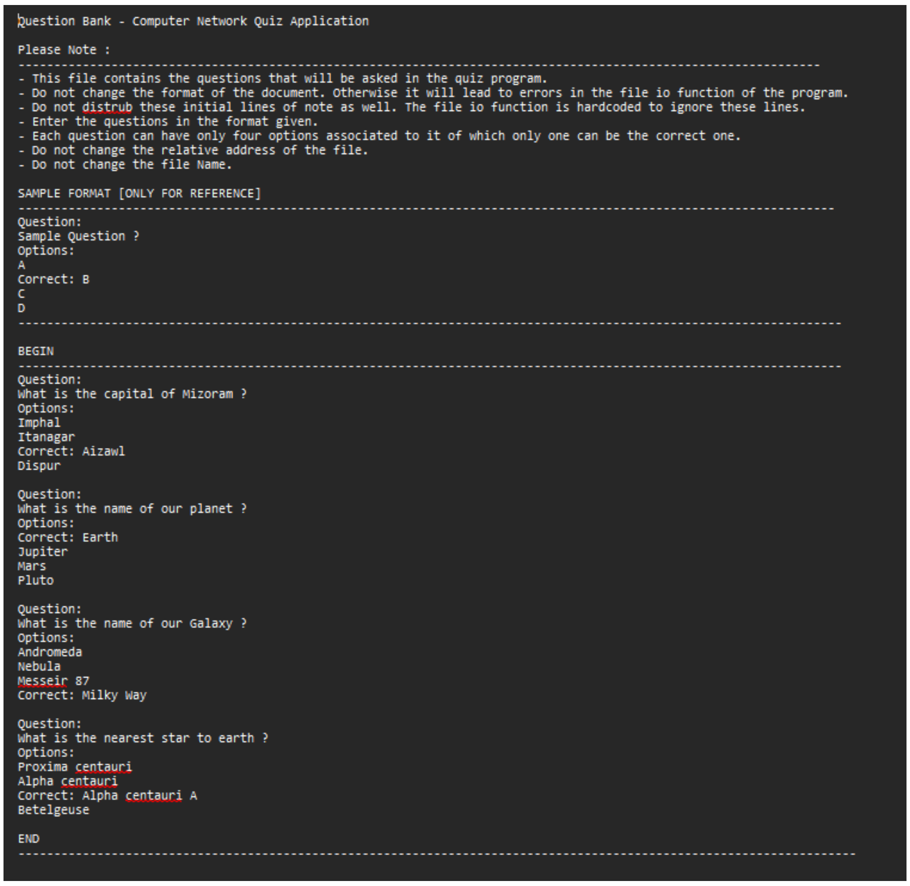
  A. Correct: Data Encryption
  B. Process Management
  C. File Management
  D. Memory Management

- **Question:** How many layers are there in the TCP/IP model?
- **Options:**

  A. 6
  B. 7
  C. 4
  D. Correct: 5

END

1.2 Image for the reference of txt file



```
Question Bank - Computer Network Quiz Application

Please Note :
----------------------------------------------------------------------------------------------------------
- This file contains the questions that will be asked in the quiz program.
- Do not change the format of the document. Otherwise it will lead to errors in the file io function of the program.
- Do not distrub these initial lines of note as well. The file io function is hardcoded to ignore these lines.
- Enter the questions in the format given.
- Each question can have only four options associated to it of which only one can be the correct one.
- Do not change the relative address of the file.
- Do not change the file Name.

SAMPLE FORMAT [ONLY FOR REFERENCE]
------------------------------------------------------------------------------------------------------------
Question:
Sample Question ?
Options:
A
Correct: B
C
D
-------------------------------------------------------------------------------------------------------------

BEGIN
-------------------------------------------------------------------------------------------------------------
Question:
What is the capital of Mizoram ?
Options:
Imphal
Itanagar
Correct: Aizawl
Dispur

Question:
What is the name of our planet ?
Options:
Correct: Earth
Jupiter
Mars
Pluto

Question:
What is the name of our Galaxy ?
Options:
Andromeda
Nebula
Messeir 87
Correct: Milky Way

Question:
What is the nearest star to earth ?
Options:
Proxima centauri
Alpha centauri
Correct: Alpha centauri A
Betelgeuse

END
---------------------------------------------------------------------------------------------------------------
```

Figure 1: Reference for Question Format

Summary of getQuestions Function

1. **Open File:** It opens the "questionBank.txt" file for reading.
2. **Find Beginning of Questions:** It reads lines from the file until it encounters the "BEGIN" marker, indicating the start of the questions section.
3. **Read Questions:** Once the beginning is found, it starts reading each question from the file until it reaches the "END" marker, denoting the end of the questions section.
4. **Extract Information:** For each question, it extracts the question itself, its options, and the correct solution. It formats the question and its options into a string.
5. **Populate Lists:** It appends the formatted question string to the 'questions' list and the correct solution (in the form of 'a', 'b', 'c', or 'd') to the 'solutions' list.
6. **Close File:** After all questions are processed, the file is closed.

2 Video Demonstration

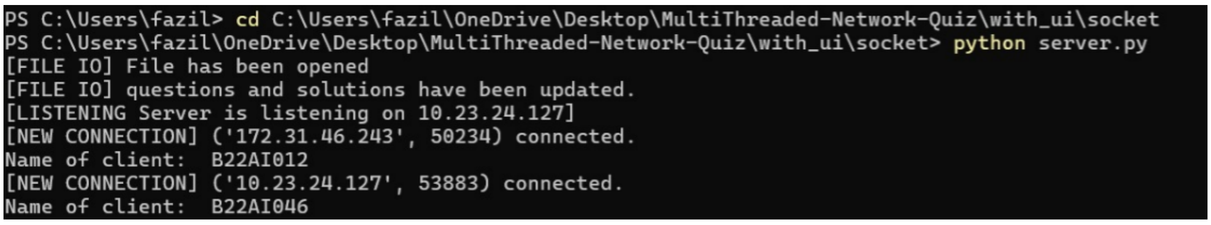
Here is a video:

Please clink on the tab for video demonstration : Click here for video link

3 Server Implementation

3.1 Socket Setup

- Establishes a TCP/IP socket connection to listen for client connections on a specified port.



```
PS C:\Users\fazil> cd C:\Users\fazil\OneDrive\Desktop\MultiThreaded-Network-Quiz\with_ui\socket
PS C:\Users\fazil\OneDrive\Desktop\MultiThreaded-Network-Quiz\with_ui\socket> python server.py
[FILE IO] File has been opened
[FILE IO] questions and solutions have been updated.
[LISTENING Server is listening on 10.23.24.127]
[NEW CONNECTION] ('172.31.46.243', 50234) connected.
Name of client:  B22AI012
[NEW CONNECTION] ('10.23.24.127', 53883) connected.
Name of client:  B22AI046
```

Figure 2: Printing the User Ids of clients

3.2 Question Management

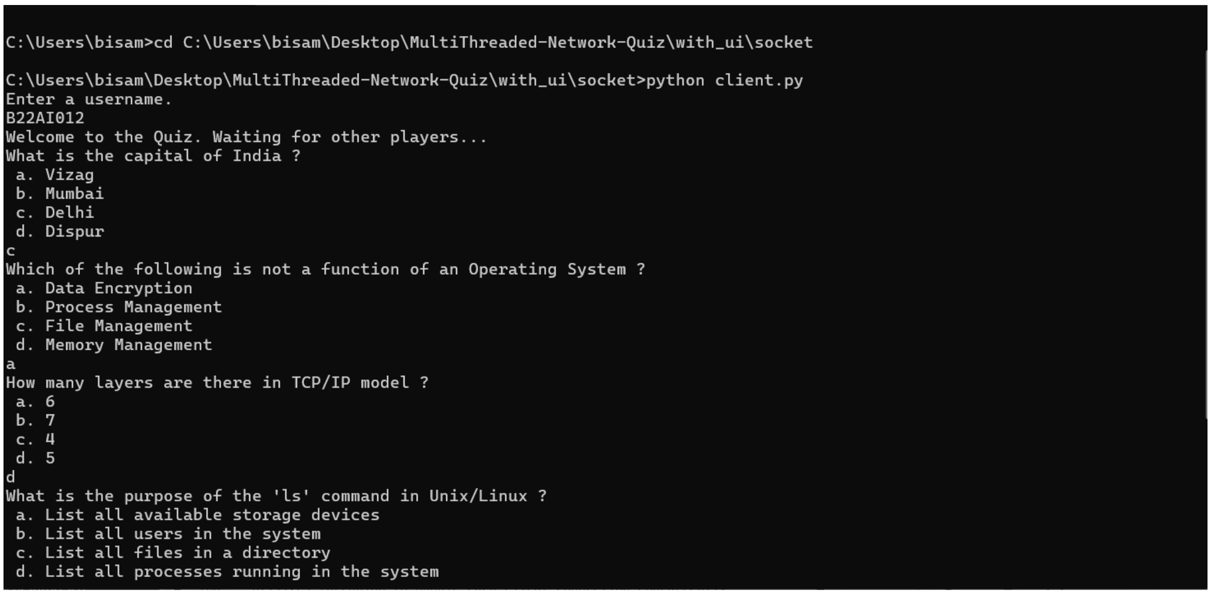
- Retrieves questions and solutions from a text file using the `getQuestions` function, providing a dynamic and extensible question bank.

3.3 Multithreading

- Utilizes threading to handle each client connection concurrently, ensuring responsiveness and efficient resource utilization.

3.4 Question Broadcasting

- Broadcasts questions to all connected clients, facilitating synchronized quiz participation.



```
C:\Users\bisam>cd C:\Users\bisam\Desktop\MultiThreaded-Network-Quiz\with_ui\socket

C:\Users\bisam\Desktop\MultiThreaded-Network-Quiz\with_ui\socket>python client.py
Enter a username.
B22AI012
Welcome to the Quiz. Waiting for other players...
What is the capital of India ?
 a. Vizag
 b. Mumbai
 c. Delhi
 d. Dispur
c
Which of the following is not a function of an Operating System ?
 a. Data Encryption
 b. Process Management
 c. File Management
 d. Memory Management
a
How many layers are there in TCP/IP model ?
 a. 6
 b. 7
 c. 4
 d. 5
d
What is the purpose of the 'ls' command in Unix/Linux ?
 a. List all available storage devices
 b. List all users in the system
 c. List all files in a directory
 d. List all processes running in the system
```

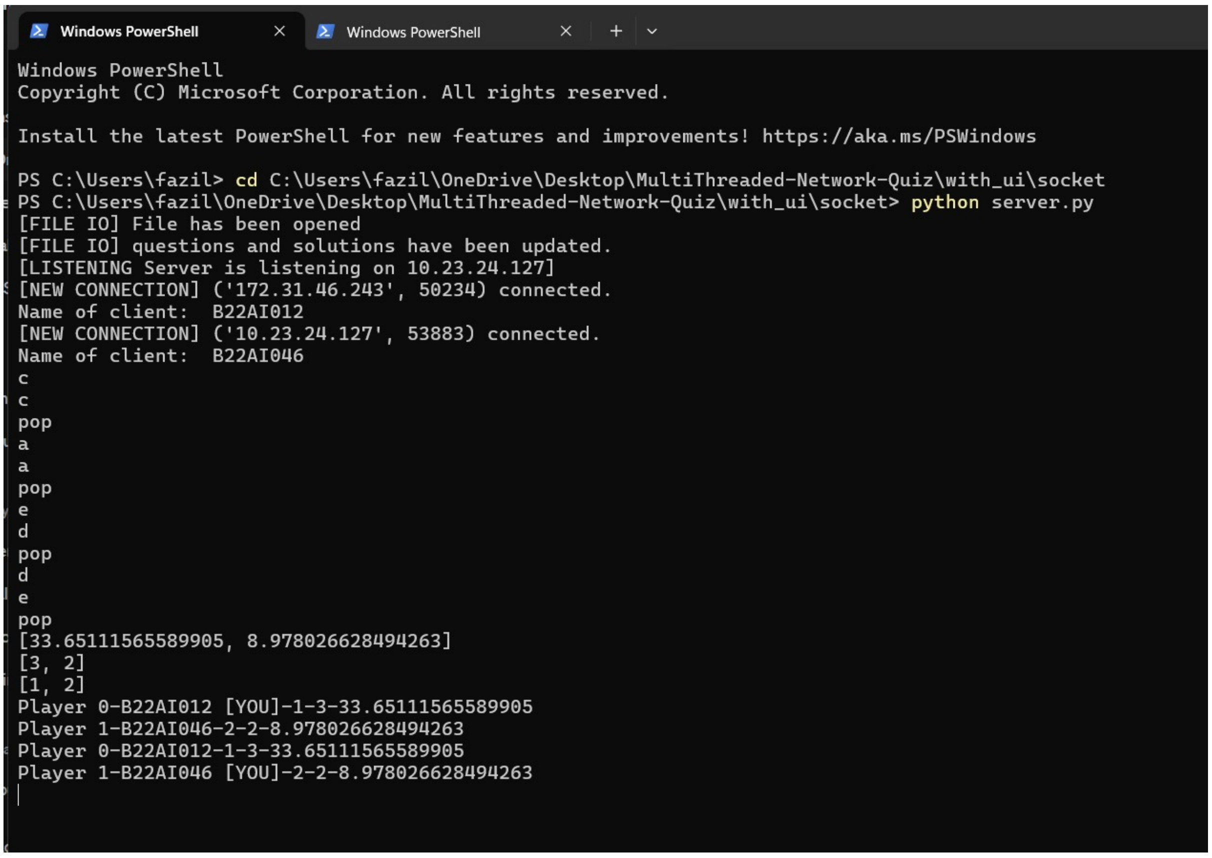
Figure 3: Broadcast questions to clients

3.5 Response Handling

- Receives and processes client responses, updating scores and managing quiz flow.

3.6 End of Quiz

- Calculates rankings based on scores and time taken, broadcasts results to clients, and gracefully closes connections.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\fazil> cd C:\Users\fazil\OneDrive\Desktop\MultiThreaded-Network-Quiz\with_ui\socket
PS C:\Users\fazil\OneDrive\Desktop\MultiThreaded-Network-Quiz\with_ui\socket> python server.py
[FILE IO] File has been opened
[FILE IO] questions and solutions have been updated.
[LISTENING Server is listening on 10.23.24.127]
[NEW CONNECTION] ('172.31.46.243', 50234) connected.
Name of client:  B22AI012
[NEW CONNECTION] ('10.23.24.127', 53883) connected.
Name of client:  B22AI046
c
c
pop
a
a
pop
e
d
pop
d
e
pop
[33.65111565589905, 8.978026628494263]
[3, 2]
[1, 2]
Player 0-B22AI012 [YOU]-1-3-33.65111565589905
Player 1-B22AI046-2-2-8.978026628494263
Player 0-B22AI012-1-3-33.65111565589905
Player 1-B22AI046 [YOU]-2-2-8.978026628494263
```

Figure 4: Recieves Responces

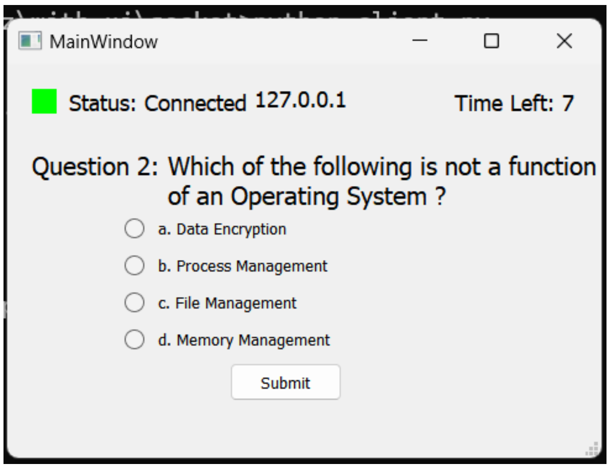
4 Client Implementation

4.1 Socket Connection

- Establishes a TCP/IP socket connection with the server to participate in the quiz.

4.2 User Interface

- Displays questions and options using a graphical user interface (GUI) implemented with PyQt5, providing an interactive quiz experience.



🟩 Status: Connected 127.0.0.1 Time Left: 7

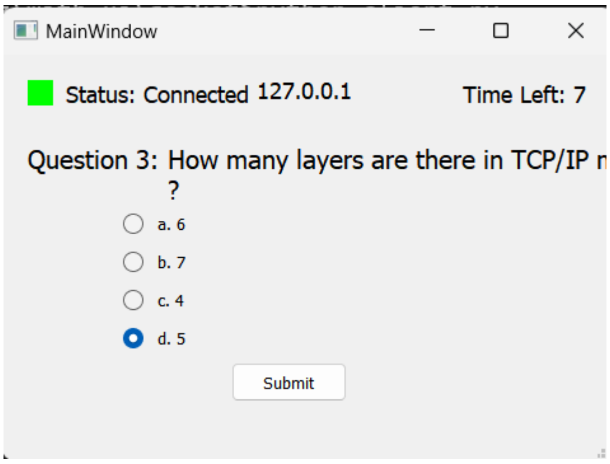
Question 2: Which of the following is not a function of an Operating System ?

- a. Data Encryption
- b. Process Management
- c. File Management
- d. Memory Management

Figure 5: Display of Question

4.3 Message Reception

- Receives questions from the server and displays them to users, allowing them to select options.



🟩 Status: Connected 127.0.0.1 Time Left: 7

Question 3: How many layers are there in TCP/IP r ?

- a. 6
- b. 7
- c. 4
- [✓ d. 5]

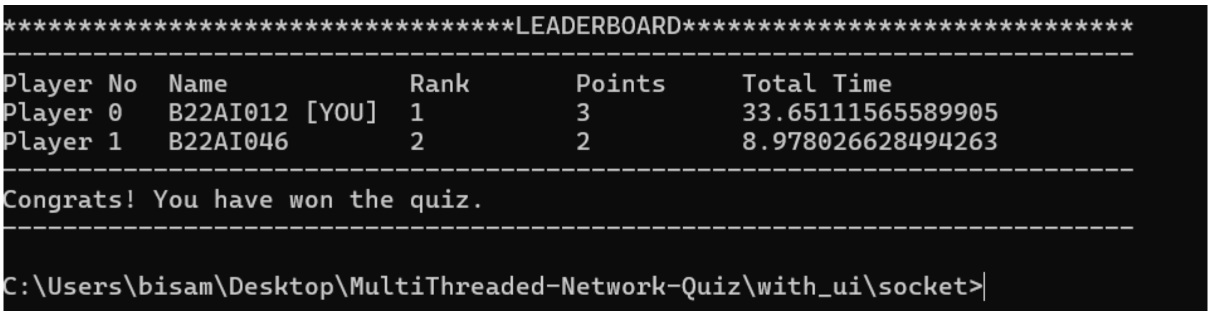
Figure 6: Selecting the option

4.4 Response Transmission

- Sends selected options to the server for scoring and updates the UI based on quiz progress.

4.5 Leaderboard Display

- Receives and displays the final leaderboard upon completion of the quiz.



```
*********************************LEADERBOARD*****************************
-------------------------------------------------------------------------
Player No  Name            Rank       Points     Total Time
Player 0   B22AI012 [YOU]  1          3          33.65111565589905
Player 1   B22AI046        2          2          8.978026628494263
-------------------------------------------------------------------------
Congrats! You have won the quiz.
-------------------------------------------------------------------------

C:\Users\bisam\Desktop\MultiThreaded-Network-Quiz\with_ui\socket>
```

Figure 7: Leader Board Display

5 Additional Insights

5.1 Customizable Parameters

- Parameters such as maximum waiting time for clients to join, maximum number of players allowed, and timing allocated for answering each question are declared for easy customization and configuration.

5.2 Scalability

- The application is scalable and can accommodate more participants by adjusting the maximum number of players allowed, making it suitable for a wide range of quiz sizes.

5.3 User Experience

- By providing a time limit for answering each question, the application enhances user engagement and adds a sense of urgency to the quiz, contributing to a more immersive experience.

5.4 Error Handling

- Robust error handling mechanisms ensure graceful handling of unexpected client disconnections and maintain the stability and reliability of the quiz application.

5.5 Performance Optimization

- Efforts to optimize communication and response processing minimize latency, providing a seamless and responsive quiz experience for all participants.

6 Conclusion

The multithreaded network quiz application effectively demonstrates the integration of computer networks and multithreading concepts in Python. By combining socket programming, multithreading, and GUI development, it provides a comprehensive solution for hosting interactive quizzes over a network. With its customizable parameters, scalability, and emphasis on user experience and error handling, the application meets the requirements of a real-world quiz scenario, delivering an engaging and enjoyable experience for participants.

7 Execution

Run the Server:

- Open a terminal or command prompt.
- Navigate to the directory containing the `server.py` file.
- Run the following command:

```
python server.py
```

- This will start the server, and it will begin listening for client connections.

Run the Clients:

- Open a new terminal or command prompt window for each client you want to run.
- Navigate to the directory containing the `client.py` file.
- Run the following command in each terminal window:

```
python client.py
```

- Each client will connect to the server and wait for the quiz to start.

Participate in the Quiz:

- Once all clients have connected, the server will start the quiz and begin sending questions to the clients.
- Clients will display the questions and options using the provided GUI.
- Participants can select their answers within the allotted time frame and submit them to the server.

Quiz Completion:

- After all questions have been answered, the server will calculate the scores and rankings.
- The final leaderboard will be displayed to all clients, indicating their performance in the quiz.
- Clients can view their ranks and scores before the server closes the connections.