Computer Networks Final Project 2023

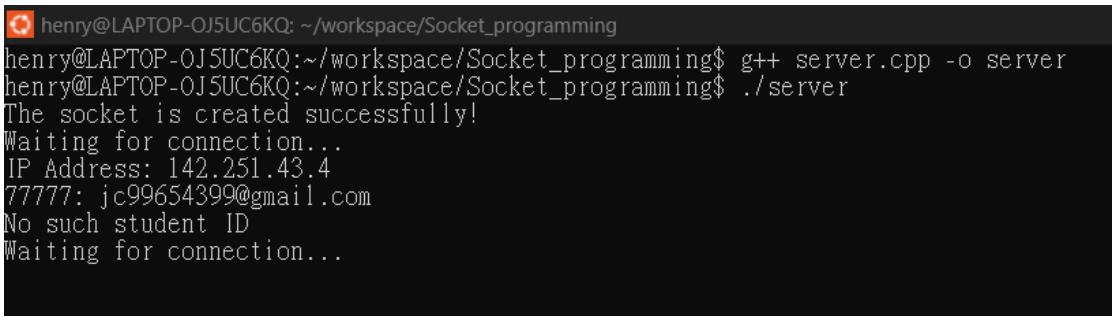
Student: 109070025 林泓錩

PART 1: Result

Execution result using Ubuntu and tmux:

The left side is the server；right side is the client.

The bigger screenshot of server:



The bigger screenshot of client:

PART 2: Experience

server.cpp:

1. The code begins by including necessary header files for input/output, networking, and vector data structure.
2. The code defines various constants such as buffer size and port number.
3. A “Student” class is defined to store student ID and email.
4. A vector “st_list” of type `Student` is declared to store the data read from the "query.txt" file.
5. The function “Student_List()” is defined to read the "query.txt" file and populate the “st_list” vector with student data.
6. In the “main()” function, various variables and structures are declared to set up the server socket.
7. The socket is created using the “socket()” function, and an error check is performed to ensure successful socket creation.
8. The “setsockopt()” function is used to set the “SO_REUSEADDR” option to avoid "address already in use" errors.
9. The “bind()” function is called to bind the server port and address to the socket.
10. An error check is performed to ensure successful binding.
11. The “listen()” function is called to listen to incoming connections on the socket, with a maximum queue length of 10.
12. The “Student_List()” function is called to read and populate the student data from the "query.txt" file.
13. The server enters a loop and waits for incoming connections using the “accept()” function.
14. Upon successful connection, the server sends a menu message to the client and waits for a response.
15. The server receives the client's choice and handles different cases based on the choice:
    - If the choice is "1", the server requests the client to input a URL address and performs a DNS lookup using “gethostbyname()”.
    - If the choice is "2", the server requests the client to input a student ID and searches the “st_list” vector for a matching ID to retrieve the corresponding email.
    - If the choice is "3", the client wants to quit the connection, so the server breaks the inner loop.

- If the choice is invalid, the server sends an error message to the client.

16. The server closes the connection and continues to wait for new connections in the outer loop.

client.cpp:

1. The code begins by including necessary header files for input/output, networking, and vector data structure.
2. The code defines various constants such as buffer size, IP address, and port number.
3. In the “main()” function, a variable “server_addr” of type “sockaddr_in” is declared to store the server's address.
4. The socket is created using the “socket()” function, and an error check is performed to ensure successful socket creation.
5. The server address is set up with the IP address and port number.
6. The “connect()” function is called to connect to the server using the created socket and the server address.
7. An error check is performed to ensure successful connection.
8. If the connection is successful, a message is printed to indicate the successful connection.
9. The client enters a loop and waits for messages from the server.
10. The client receives the message from the server and displays it.
11. The client prompts the user to enter a message and sends it to the server.
12. If the message sent is "3", the client wants to quit the connection, so it breaks the loop.
13. The client continues receiving and sending messages until it decides to quit.
14. Once the loop is exited, the client closes the socket and terminates the program.