COMP90015 Distributed Systems

2020 A1: Multi-threaded Server

Student ID: 1066384

Name: Yu Ting Liu

22/04/2020

Problem Context

Utilize the client-server architecture to design and implement a multi-threaded dictionary server. This server allows clients to implement some actions on the dictionary such as search, add, and remove. In this project, TCP is used as a reliable communication protocol. Plus, the message's I/O are applied directly on txt. Files. When it comes to failure handling, this system can identify errors such as invalid inputs from the command line, network communication error, I/O error, and so on. Last but not least, this system can also identify and indicate our clients that some illegal commands have occurred.

Components of the system

1. Server

Thread-per-request architecture is applied to this server. That is to say, the server will keep listening to the client's request and open a new thread whenever there is a new request from the client. Moreover, with the synchronized methods, the server can keep the data up to date. As a result, the server not only can handle multiple clients at the same time but also synchronize the changed data.

To execute the server, we must enter two parameters into the command line, a port number for clients to build the connection and a file path for the server to access the dictionary file.

2. Client

The client will take two parameters as well, the server address(IP) and a port number to build connection. After the client's socket is opened and the connection is established, the client can send commands to the server with GUI and receive the result from the server. Then print it on the GUI.

3. GUI

This GUI is within the client and will display a user interface on the screen. This interface contains three main buttons such as search, add, and remove, which are the actions could be done by the client. Moreover, the interface indicates the users that where should they enter the word and meaning. And of course, it will display the result in a result block to show our users if their actions are completed successfully or the meaning of the queried word.

● Design & Interaction Diagram

➢ Dictionary

There are two dictionary txt. files. "TestDict.txt" is the one for clients to query. And "tempDict.txt" is the one for remove action to temporarily store the dictionary, and then copy back to "TestDict.txt". Besides, the data format inside the dictionary will look like this "school : a place where children go to be educated", with " : " between the word and the meaning.

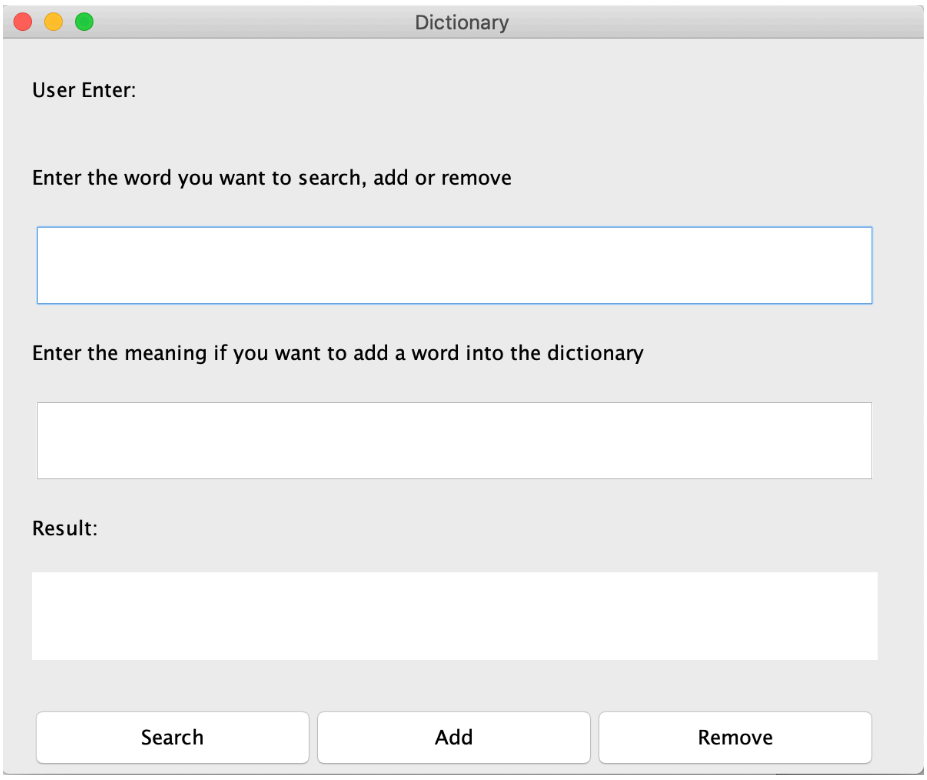
➢ Server

Request a port number and a file path as input parameters. Once the parameters are confirmed to be valid, the server will open the socket and keep listening to clients' requests. After the connection is established, the server will create a new thread for each new request. Then, it will execute the requested action from clients and synchronize the data. Eventually, it will send the result back to clients.

➢ Client

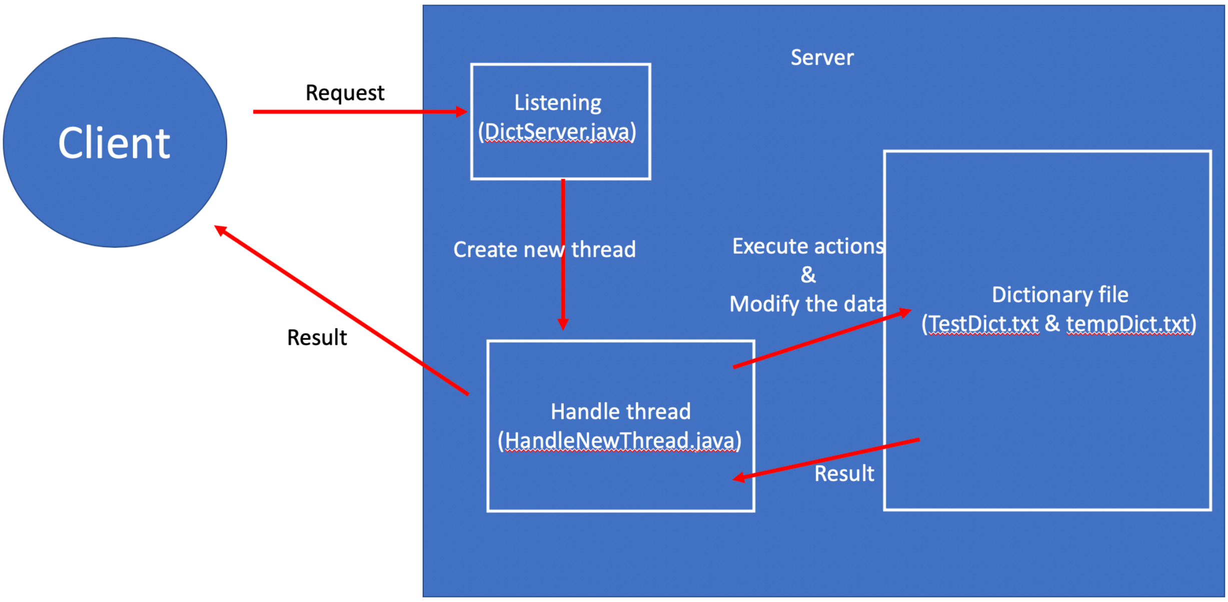
Request a server address and a port number as input parameters. Once the parameters are confirmed to be valid, a socket will be open and attempt to build the connection with the server. If the connection is failed, the GUI will print "Unable to connect". Otherwise, the connection will be established and the client can start sending commands from GUI to the server.

- GIU



Dictionary
User Enter:
Enter the word you want to search, add or remove
Enter the meaning if you want to add a word into the dictionary
Result:
Search
Add
Remove

- Interaction Diagram



- **Analysis**

  - Advantages

    A. The system provides a reliable environment with TCP

    B. Be able to manage multiple clients at the same time and ensure the data consistency. Because of the design of thread-per-request.

  - Disadvantages

    A. The main disadvantage is that a leak of the socket open-close management. To be more specific, the system creates a new socket every time after returning the result to our user. This should be fixed.

    B. Thread-per-request is more expensive than other methods. This is because it is expensive to create threads. Plus, threads will consume the server's resources.

    C. This project does not consider the security issue. As a result, everyone who knows the port and server address can access the server and modify the data which makes the server very attackable.