The University of Melbourne

COMP90015 Distributed Systems

Semester 2, 2019

Assignment 1

Multi-threaded Dictionary Server

Student Name: Xinnan SHEN

Student ID: 1051380

Tutor: TianZhang He

1. Introduction

When reading articles or books, we inevitably encounter some words that we are not familiar with. A traditional way to get the meaning of words is to find their meanings in the dictionary. This way is useful to some extent, but it is not convenient for us to use a paper-based dictionary. At present, with the development of information technology, an increasing number of people has chosen to use e-dictionary, a dictionary that can be installed and used in the computer. In this project, I have designed a multi-threaded dictionary server so that users can use the dictionary in the computer simultaneously.

The dictionary has the following functions: Firstly, users can query the meaning of a given word; Secondly, users can add a new word into the dictionary; Thirdly, users can remove the existing word from the dictionary. The designing and implementation of the project are based on TCP protocol and multi-threaded method, with the use of the Java language.

2. System Components

The system uses a client-server architecture. The connection between client and server is based on the socket using TCP protocol so that the reliability of communication can be guaranteed. The method of thread-per-request is used to implement the multi-threaded server. Every time the client makes a request, the server builds a thread to communicate with the client. The request can be querying for a word, adding a new word and removing an existing word. These requests have access to the same shared resources, the dictionary data. To prevent the inconsistent state, I set the methods of handling the dictionary data to be synchronized. Having done that, multiple clients can access the server concurrently, and they do not need to worry about the inconsistent state.

As for the storage of the data, the system uses CSV format to store the data in the hard drive. The first line of the CSV file is ‘word’ and ‘meaning’, which will be ignored once read by the server. From the second line and so on, the first column is the word and the second column is the meaning of the word. When the server starts, it will automatically load the data from the hard drive to the memory. In the project, an array is utilized to store them in memory.

In terms of the design of GUI in both client and server, the system uses the WindowBuilder plugin to build the GUI in Java Swing framework so that the users can use the system much more conveniently. WindowBuilding is a useful plugin to help us to implement the GUI much more efficiently than purely coding to build GUI in Java Swing Framework.

3. System Design

3.1 Communication Design

Concerning system communication, the project utilizes the TCP protocol to communicate between the server and the client. When the client wants to send a request to the server, it first initializes the socket object with the server's IP address and port number. Then, the client creates I/O streams so that it can communicate with the server by the streams. After that, the client sends request messages to the server and handles the response from the server. Once the client finishes dealing with the request and response, it will close the socket, which means this request has already been handled well.

As for the server, it first initializes the server socket object and binds the server socket to the port. Then, it waits for the client to connect with it. Once it receives a connection from a client, it will accept the client's connection request. Just like the client, the server also needs to create I/O streams to communicate with clients. The messages are transmitted between clients and the server through the I/O streams, which is quite essential in socket communication. After that, the server begins to communicate with the client, so that the requests from clients can be handled by the server properly. When the server finishes dealing with the request from the client, it sends the result back to the client so that the user of the client is able to see the result of the request. Finally, when all of these has finished, the server will close the socket object.

The interaction diagram is shown below.

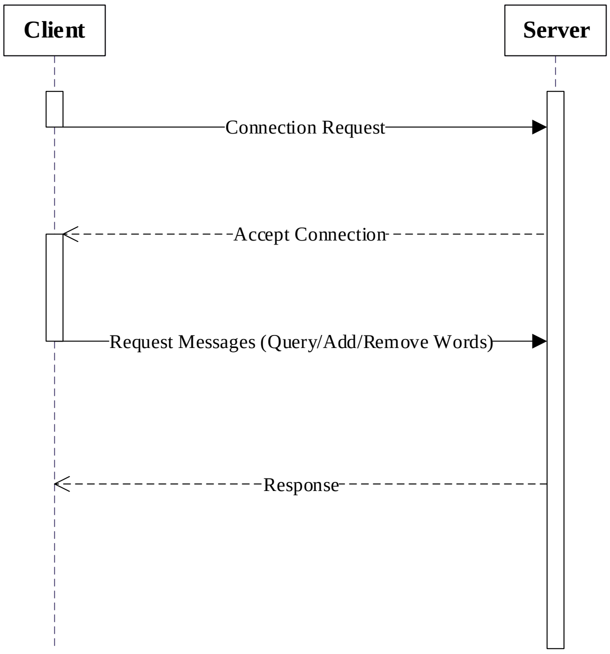


Figure 1: Interaction diagram

3.2 Class Design

There are multiple classes in both client and server. As for the client, there are two classes, namely DictionaryClient class and ClientThread class. The DictionaryClient class is used to deal with GUI problems such as the initialization of GUI and the action of the button in the GUI, whereas the ClientThread class is used to deal with problems regarding socket communication. When the client receives a message indicating the result of the request, the class is used to handle the result.

In terms of the server, there are four classes. The first one is the DictionaryServer class, which is the main class to load the data from the file into the array in the memory. The second one is the DictionaryServerWindow class, and its function is initializing GUI and dealing with the action of buttons in the GUI. The third one is the ServerThread class, which is most important, and the main function is to deal with the request received from clients and send the outcome back to the clients. The last one is the StartThread class, which is used to deal with the socket connection between the server and clients. Every time a client sends a request to the server, the server creates a thread to deal with the request.

The structure of the classes is shown below.

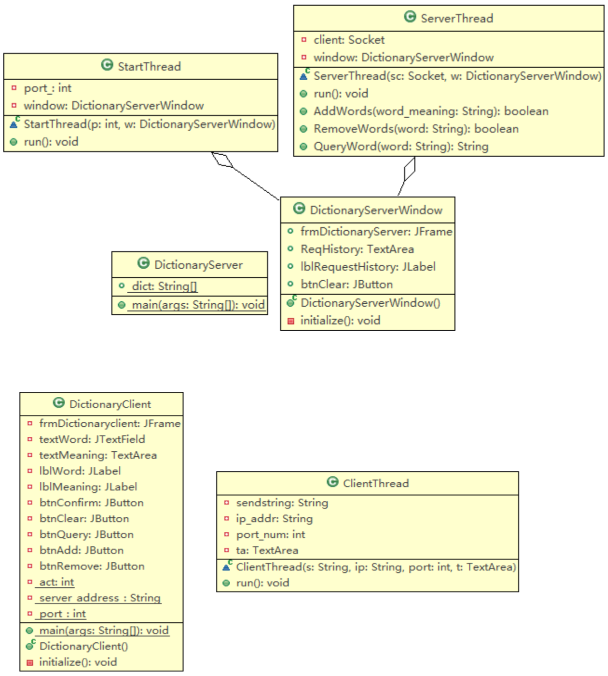


Figure 2: Classes Structure

3.3 Functional Design

The function of this project includes querying, adding and deleting words. The designing method of these functions is shown below.

(1) Query the meaning of a given word

When the server receives the word, it searches the array where the word and its meaning are stored. If the word already exists in the dictionary, the server returns the meaning of the word to the client. Otherwise, the server returns an error. The flow chart of this function is shown as Figure 3.

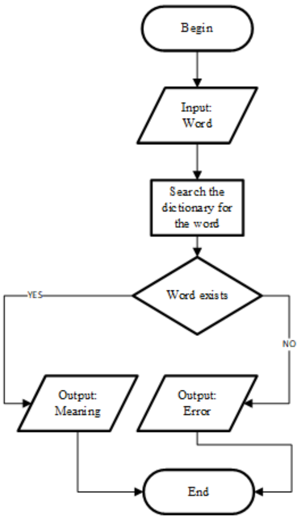


Figure 3: Query Word Flow Chart

(2) Add a new word

When the server receives the new word and the meaning, it first searches the array for the word. If the word has already existed in the dictionary, the server returns an error. Otherwise, the server adds the new word and its meaning into the array and returns success. The flow chart of this function is shown as Figure 4.

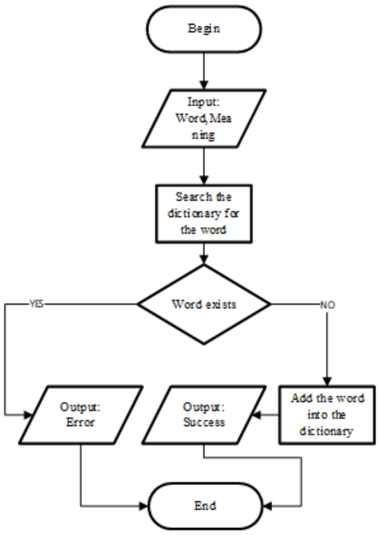


Figure 4: Add Word Flow Chart

(3) Remove an existing word

When the server receives the word, it searches the array where the word and its meaning are stored. If the word does not exist in the dictionary, the server returns an error. Otherwise, the server removes the word from the array and returns success. The flow chart of this function is shown as Figure 5.

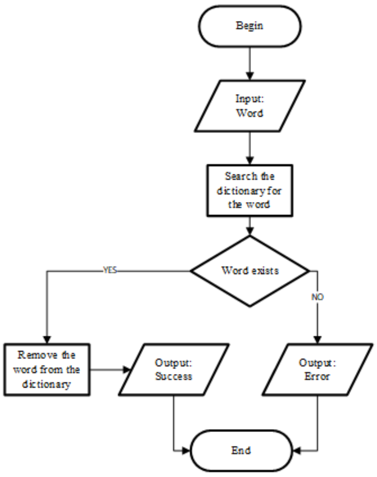
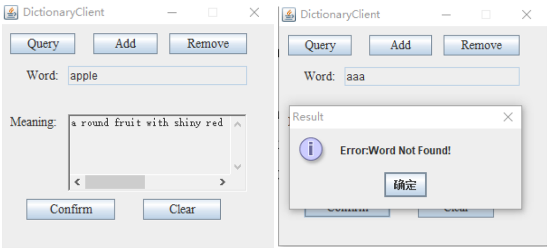

Begin
Input: Word
Search the dictionary for the word
Word exists
YES
NO
Remove the word from the dictionary
Output: Success
Output: Error
End

Figure 5: Remove Word Flow Chart

4. System GUI and Running Result

(1) Query the meaning of a given word

Figure 6 is the running result of querying words. If the word exists, the user will see the meaning of the word. If the word does not exist, the user will see the error message.



Word: apple

Meaning: a round fruit with shiny red

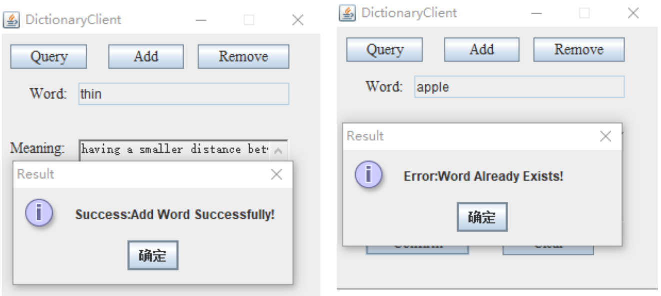
Word: aaa

Error:Word Not Found!

Figure 6: Query Word Running Result

(2) Add a new word

Figure 7 is the running result of adding words. If the word does not exist, the user will see the message of success. If the word already exists, the user will see the error message.



Word: thin

Meaning: having a smaller distance bet

Success:Add Word Successfully!

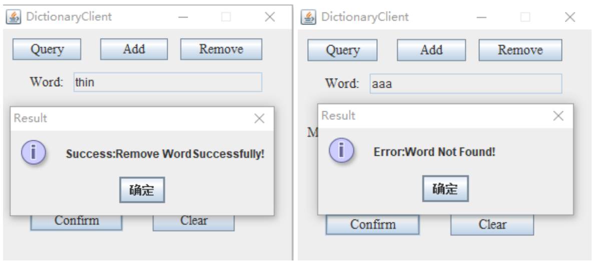
Word: apple

Error:Word Already Exists!

Figure 7: Add Word Running Result

(3) Remove an existing word

Figure 8 is the running result of removing words. If the word exists, the user will see the message of success. If the word does not exist, the user will see the error message.



Word: thin

Success:Remove WordSuccessfully!

Word: aaa

Error:Word Not Found!

Figure 8: Remove Word Running Result

5. Excellence and Creativity

5.1 Project Excellence

(1) The notification of errors: When encountered with error, the system tries to tell users the error in GUI. A message box that shows the error message will occur when an error happens.

(2) The record of request: In the server, all the request from clients can be viewed. If possible, the server is also able to delete all the records of requests.

5.2 Project Creativity

(1) Server GUI: In the server, the GUI is designed and implemented to help the server understand the request from clients and the result of each requests (success or failure).

(2) Memory from the client: The action of the user is memorized in the client unless the user chooses another action. For example, the user first chooses to query the word and already gets

the result. Next time he does not need to click on the query button to confirm his action, and he only needs to input the word so that the system can memorize his action.

6. Conclusion

In this project, I used Java programming language to design and implement a multi-threaded server. During the designing and implementation, I used the multi-threaded method and TCP protocol to make reliable communication between server and client. After I finished the project, I have become familiar with multi-threaded programming and socket programming, as well as the method of designing GUI in Java.

References

[1] Couloris, G, Dollimore, J. Kinberg, T, and0 Blair, G, Distributed Systems - Concepts
and Design, 5th Edition, Addison-Wesley, Pearson Education, UK, 2011, ISBN
0132-143-011. http://www.cdk5.net

[2] R. Buyya, T. Selvi, X. Chu, “Object Oriented Programming with Java: Essentials and Applications”, McGraw Hill, New Delhi, India, 2009, USA.