Using Java Sockets to Deliver Quotes

Henry Wilt

Ursinus College
hewilt@ursinus.edu

February 15, 2022

Abstract

The goal for the assignment was to use a linux environment and modify the two class, DataClient and DataServer, from the Textbook figures 3.27 and 3.28 [1]. We should change the two classes to have the DataServer respond to the DataClient with Quotes instead of the current date and time. There is another file with all the quotes that the server can respond to the Client. With the the Server reading them in at Startup or reading them during run time. The Client has to take in a command line argument for the number of quotes the user wants.

I. About the Project

There are two Java classes in the project, the QuoteServer and the QuoteClient [2]. We will take a look into the QuoteClient first as that is what the user communicates with, when you run the file, you have the option to put in the number of quotes you want returned back with the command line. It will then try to connect to the server using Sockets, for our example this just means connecting to local host. It will then send the number of quotes the user wants to the Server and if the user did not specify then it will send an -1 to represent only sending back one random quote. QuoteClient will then wait for the server to respond with the Quotes that was asked for and print them out to the terminal. and will finish off with closing the socket and ending the program.

As we start to talk about the Server side of the project, there is the file which stores all the quotes titled "quotes.txt" and the java class called "QuoteServer.java". These two files make up the Server aspect of the project. In the text file, there are 99 quotes in plain text format delimited by newline characters. In the QuoteServer file, there are two methods: getQuotes() and main(). For the getQuotes method, it will return an String array of the quotes from the file. The method is called right when the Server starts up to save time when the Client calls the server. In the main method, it will first call the getQuotes method and then for forever will wait to accept requests from the Client. If it gets a request from the Client, it will read in the number of quotes the Client wants and then check if the number is -1 if the user did not submit any number to the Client. If this is true then the server will send back only one random quote. If not, then it will loop through the number of quotes to send back and create a new random number that will choose from the array and send back. After, it goes through the loop it will keep waiting to accept another socket request.

II. Discussion on Accessing Quotes

There are two different ways of accessing the Quotes for the Server. Both ways has them being stored in a file to be read in a certain times but one way is being read in on startup and being stored in memory and the other is randomly accessing from the file during the run time whenever the server gets a request. In my implementation, I had the quotes being read in on startup of the server as my thought process was that it would save time when the client requests a new quote. The advantage is saving time but a big disadvantage is space, as quotes increase the space complexity increases as well. As this is only an small example for a class, there is not that big of a disadvantage withe space complexity. If this was implemented on a larger internet environment, with many thousands of quotes, then the amount of space used is too massive for the program to actually run. This can be fixed by randomly accessing the file during the request from a Client. This will save on storing all the quotes in an array. But, an disadvantage is time complexity as it is very time consuming to access a file multiple times. It will also use a lot of resources to move the file in and out memory to keep reading it. There are always trade-offs but it is better to implement accessing the file once when in smaller environment which will be used for testing or learning. While it is better off to implement the randomly accessing when in larger environments to keep the space complexity small as computers are fast enough nowadays to move files into memory and open them without losing a lot of time.

III. Conclusion

Using Java sockets to connect to a server that will respond with Quotes was interesting. I can see the connection between networks and the operating system are very similar. As the operating system is just a bunch of "local host" networks with other components in the machine.

References

[1] Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne. *Operating system concepts*. 10th ed. Wiley, 2018.

[2] Henry Wilt. *QuoteServer/Client Assignment Code*. URL: `https://github.com/hwilt/CS376/tree/main/HW2`. (accessed: 02.15.2022).