CO324 Lab 02

Sockets and Message Framing in Transmission Control Protocol (TCP)

In this lab, we will look at the way of creating a simple TCP client/server application and we will understand the need of message framing in TCP with different implementation methods.

TCP Echo Server

An echo server is that, which sends back whatever the data it has been received.

Exercise 01: Complete the given TCP server and client codes in such a way that the server acts as an echo server.

Exercise 02: Note the way the try block is opened. What is the advantage of this method over the normal way we open a try block?

Message Framing

Data is not sent as a continuous stream through a network. Instead, data is separated into packets of a particular size defined by protocols and configurations of the system. Which means, a large message would be separated into small chunks before being sent. Due to the unreliability of the network, certain packets might have a delay in delivery.

This might become a problem since the receiver has no way to figure out how much data is sent by the sender as one message. Message framing is used to handle this problem. Two basic methods of handling this situation are:

01 - Delimiter based approach.

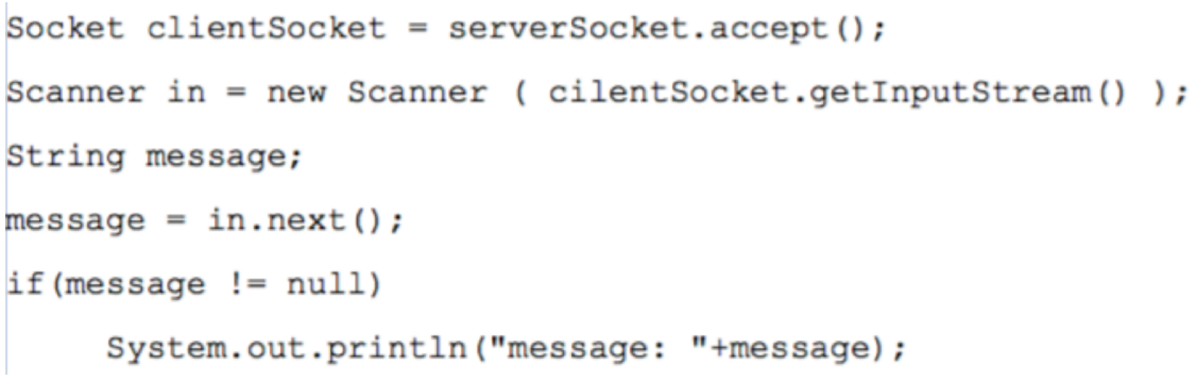
02 - Header based approach.

Delimiter Based Approach

In this method, an end of one message is denoted by a special character called a 'delimiter'. The sender attaches this special character at the end of the string. The receiver then reads characters until this special character is reached.

Exercise 03: Compile and run the given code. FrameServer.java and FrameClient.java. Explain what will happen if the client did not append the delimiter at the end of the message. And what will happen if the user decided to have a dash (-) character in one of the messages entered.

Exercise 04: What is the delimiter used for framing in the following code?



```
Socket clientSocket = serverSocket.accept();
Scanner in = new Scanner ( cilentSocket.getInputStream() );
String message;
message = in.next();
if(message != null)
     System.out.println("message: "+message);
```

Now, consider the given FrameClient2.java and FrameServer2.java codes. This application allows users to send multi-line messages yet delimiting the messages with some characters that can be entered with a keyboard. Modify the program so that this application will do the same task with non-printing delimiters.

Ex:
String line = “Hello World” + `\u0003`