

REPORT 1

JINGMIAN ZHANG

2017 February

213568993

Table of Contents

Contents

1 Introduction	1
2 Overall Program Design	1
3 Possible Improvement	3
4 Situation that the program may not work	3
5 How to compile and run	4

1 Introduction

Learning about the TCP/IP protocol is essential in acquainting with communication network. In this first assignment, we build the skeleton of a very simple and basic client-server application, which allows multiple clients to exchange message with the server. Through the process of completing the assignment, we now have greater insight and understanding about how the client server model is constructed, implemented and functioned.

2 Overall Program Design

The client server programming application is a program that consists of three components, Echoclient, the client class, EchoServer, the server class, and MyThread, a thread class.

Sockets are heavily used to serve as an interface for clients and servers to communicate, passing information to each other. There are two types of sockets used: simple socket and server socket. Simple socket is used both in the client class and server class, while server socket is used in server class to be accepted by a simple socket, thus start the listening action for client. In the program, I specify the port number of the server to be localhost, which is '127.0.0.1', for simplicity.

With respect to the communication of client and server, `printstream` and `bufferedReader` objects are frequently used as input and output both for the clients and servers to pass information to each other. With all the input and output objects and sockets connections constructed, the echo client server system is thus complete. Then it is the task to implement the three client messages to the server: JOIN, LEAVE and LIST, as well as QUIT, which is used when a client decides to disconnect from the server. These requests are implemented by using an `arraylist`, which store all the ports number of the clients. For example, when a client joins the game and becomes a player, the `arraylist` is modified to include the port number of the client and the server send corresponding message to the client, stating that the join request is complete. Same process goes with LEAVE request. For LIST request, the server simply returns a list of all the port numbers of the clients so the client who requests the LIST command can see who is online. During all these messages, not only the `arraylist` will be modified accordingly but also the server and the client terminal will display messages concerning whether the request succeeds or not. In short, the client server program is built with sockets

object as a connection tool, with appropriate input, output objects as communicational tool between the client and the server, and an arraylist as an implementations to the three client requests.

3 Possible Improvement

Some possible improvement for this program would be to use a map that not only keeps track of all the port numbers of the players, but also the ip address of the players and the total count of the players, thus providing clients with more information available.

4 Situation that the program may not work

Since this program is written in a windows environment, there may be incidents where it couldn't work in a LINUX environment. If that were to occur, it would be best to change the code concerning getting the host name and port number.

When the program doesn't work, use the snippet of codes in Figure 1 to get host name and port number instead of the original snippet of codes (Figure 2):

```
/*
 * if (args.length != 2) { System.err.println(
 * "Usage: java EchoClient <host name> <port number>"); System.exit(1);}
 *
 *
 * String hostName = args[0]
 * String portNumber = Integer.parseInt(args[1]);
 */
```

Figure 1: code for LINUX

```
String hostName = "127.0.0.1"; // ip address for the server
int portNumber = 28993; // the port number of the server
```

Figure 2: code for WINDOWS

5 How to compile and run

It would be the most ideal to compile to run the program under a WINDOWS environment, using Eclipse to open and run it.