(IME603: Introduction to Computing)

Command Line Based Chatting Application

Socket programming & Multi-Threading

JAVA

Submitted by –

Nitesh Sharma	20114013
Mansi Varshney	20114010

Preface

It is a command line based chatting application, using which two or more than two persons can communicate with each other. It includes the concepts of multithreading and Socket Programming implemented in JAVA programming language.

How to use???

Connect all the systems you want to connect for chatting on same network (LAN)
then -

step1 : On the system you want to make as SERVER

1. compile 'UserThread.java'
2. compile 'ChatServer.java'
3. run 'ChatServer' as -

```
java ChatServer 8989
```

where 8989 is just a random port number which you can specify.

Step2 : On the systems you want to make as CLIENT

1. compile 'ReadThread.java'

2. compile 'WriteThread.java'
3. compile 'ChatClient.java'
4. run 'ChatClient' as -

   java ChatClient hostname/IP_address 8989

   need to provide the IP_Address and port no of the server

Using Step 2 we can connect as many users to the chat.
[Test snapshots are shared in TEST folder]

What are various Classes Used?

SERVER is implemented by two classes: 'ChatServer' and 'UserThread'

The ChatServer class starts the server, listening on a specific port. When a new client gets connected, an instance of UserThread is created to serve that client. Since each connection is processed in a separate thread, the server is able to handle multiple clients at the same time.

The UserThread class is responsible for reading messages sent from the client and broadcasting messages to all other clients. First, it sends a list of online users to the new user. Then it reads the username and notifies other users about the new user.

Then it enters a loop of reading message from the user and sending it to all other users, until the user sends 'bye' indicating he or she is going to quit. And finally it notifies other users about the disconnection of this user and closes the connection.

CLIENT is implemented by three classes: ChatClient, ReadThread and WriteThread.

The ChatClient starts the client program, connects to a server specified by hostname/IP address and port number. Once the connection is made, it creates and starts two threads ReadThread and WriteThread.

The ReadThread is responsible for reading input from the server and printing it to the console repeatedly, until the client disconnects.

The WriteThread is responsible for reading input from the user and sending it to the server, continuously until the user types 'bye' to end the chat.

The reasons for running these two threads simultaneously is that the reading operation always blocks the current thread (both reading user's input from command line and reading server's input via network). That means if the current thread is waiting for the user's input, it can't read input from the server. Therefore, two separate threads are used to make the client responsive: it can display messages from other users while reading message from the current user.

References

Java - Networking - Tutorialspoint

Multithreading in Java - javatpoint