Laboratory Exercise 2 - Sockets

Client-Server Communication Using Sockets

The exercise involves understanding of the inner workings of two java programs: ServerApp.java and ClientApp.java. Download files from our Google Drive. The programs implement a simple GUIed Java chat program using sockets. The main objective is to successfully run the Server and the Client on separate machines.

Part I: Setting up the Server and Client

1. The java programs can be run on a single machine but to illustrate the client-server communication clearly, you will be using two or more separate machines. You need to find a partner to accomplish this exercise.

2. Study the codes for the chat program. You are to analyze how the two machines communicate with each other when using separate machines using sockets. You are to modify some parameters of the functions to be able to communicate with the desired computer. By this time, you should know how sockets work. Hint: IP Add + Port.

   Questions/To do:
   a. Which statement in the Server.java program is crucial for the socket connection? Explain in your own words and understanding.
   b. Which statement in the Client.java program is crucial for the sockec connection? Explain in your own words and understanding.

3. Compile Server.java in one computer and Client.java in another computer. If you have accomplished Part I-2 successfully and with the correct modifications, run the programs to have a feel of how the program works.
   Questions/To do:
   a. Run the Server first then the Client. Start sending messages. Does the program work?
   b. Turn off the previously run client and server. Run the Client first then the Server, in that order, again. Start sending messages. Does the program work?
   c. Which statement(s) (in Server.java and Cleint.java) is important in holding the text(s) being passed from one computer to another? What do these statements do? Explain in your own words and understanding.
   d. Work with three computers. Run the server and one client. While the first client is running, run another client. Send message(s) from the server, do the second client receive these messages? How many clients can the server support?

4. Modify the codes to improve the chat program:
   a. such that the text that is being sent by one computer is not only seen on the recipient but also on the local chat window. Also, modify the code
   b. such that when the “send” button is clicked, the textfield is emptied of the text typed.

5. Deliverables : Submit the answers to questions in #2 and #3 in a pdf file (filename should be <surname>_sockets.pdf (e.g. gamot_sockets.pdf)). Submit the code/s in #4 as well. Zip the code/s and the pdf file in a zip file (filename should be <surname>_sockets.zip (e.g. gamot_sockets.zip)). Submit the zip file in the Canvas assignment bin for this lab exercise.

How to determine IP Address of your machines

In Linux:
1. Open terminal
2. Type

```
> /sbin/ifconfig
```

In Windows:
1. Open command prompt
2. Type

```
> ipconfig /all
```

Port Number Ranges

A port number uses 16 bits and so can therefore have a value from 0 to 65535.
Port numbers are divided into ranges as follows:

Well known ports (Port numbers 0-1023)- These are allocated to server services by the Internet Assigned Numbers Authority (IANA). e.g Web servers normally use port 80 and SMTP servers use port 25.

Registered Port (Ports 1024-49151)-These can be registered for services with the IANA and should be treated as semi-reserved. User written programs should not use these ports.

Ports 49152-65535– These are used by client programs and you are free to use these in client programs. When a Web browser connects to a web server the browser will allocate itself a port in this range. Also known as ephemeral ports.

The netstat command will allow one to determine the ports currently being used by the system.