CS25610 / CSM5610 Worksheet One 1999-00

David Price

March 3, 2000

1 Introduction

This worksheet is intended to be a simple first introduction to the production of programs that make use of TCP based communications from Java.

2 Your Task

This worksheet asks you to produce first a “client” program and then later a “server” program. The problem being solved is an extension of the PoemServer example presented in lectures.

I provide a sample version of the server which is running on the computer moin.dcs.aber.ac.uk and listening on TCP port number 4000.

The server obeys a very simple protocol which is an extension of the protocol supported by the lecture example.

3 The PoemService Protocol

3.1 Client connection and disconnection

The protocol is very simple. The server is sitting listening for incoming TCP calls on port number 4000. On receiving an incoming call, the server will extract one line of text from the socket, split the line of text into three tokens and use these to decide how to respond. The server will then send some messages back to the client and then close the socket. If the client wishes to send another command line into the server, then the client must open another TCP connection to the server and send the new command on that new connection.

3.2 Message Formats

All the messages which a client can legally send to the server consist of three components which must be separated by white space.

Currently, the server supports just three commands as follows

- GET POEM NUMBER
- GET POEM TITLES
- GET POEM n

The first message will return some text of the form

```
There are 9 poems available
```

The second message will return the titles of the available poems.

The third message will return the title and the text of the poem whose number is supplied in place of “n”.

Poems are numbered from 0 up to some number one less than the value returned in the first message.

4 The Client Program

You should construct a client program that can contact my sample server and exchange messages, obeying the protocol, with the server and display the returned results.

The client program should commence by prompting the user for a hostname to contact to reach the PoemServer.

The client program should then prompt the user for the port number being used by the PoemServer.

The client program should enable the user to specify commands, either by typing them on the keyboard or by some other means that you define. The user should be allowed to type a sequence of commands and see their respective results via a single run of your client program.

NOTE: as mentioned above, the server closes the socket after every individual command has been processed. You will thus have to think carefully about how you design your program to satisfy my requirement stated above about the user being able to submit multiple commands from a single run of the client program.

5 The Server Program

When you have completed your client program, you should then attempt to develop a server which obeys EXACTLY the same protocol as that used by my sample server. Your client program should be able to interact with your own server without making any code changes to your client whatsoever.

6 Effort Allocated to the Worksheet

It is unlikely that you will complete all of this worksheet in your two hour practical this week. You should expect to spend some “own time” work to complete the worksheet. Another, new worksheet will be provided for the practical next week.

7 Assessment of this Worksheet

These worksheets for the modules CS25610 and CSM5610 are NOT assessed.

The worksheets and the demonstrated practicals are provided as part of the educational offering of the module. The examination questions for these modules may be based on material covered in practicals in addition to lecture material and background reading drawn to your attention by the lecturers or by links on the course materials web site.