

Lab 8: Eliza - Designing a Client-Server Application

Objectives

- Learn the basics of network programming.
- Write a client server application.

You may complete this lab individually or with one colleague from this class using pair programming rules (see the course Canvas page).

1 Introduction

In this lab you will get familiar with socket programming by walking through example application and then making modifications to it. Although somewhat simplified, this example application is similar to the core of many instant-messaging applications.

2 Warmup (6 Points)

Study the example simplex-talk program provided in Ch. 1.4 of the book. Once you have a general idea of its work, it's time to run the program. Start one server and one client, in separate windows. While the first client is running, start 5 other clients that connect to the same server; these other clients should most likely be started in the background with their input redirected from a file.

In a text file, write down one-sentence answers to the following questions:

1. What happens to these 5 clients? Do their connect(s) fail, or time out, or succeed?
2. Do any other calls block?
3. Now let the first client exit. What happens?

Computer psychotherapist

ELIZA is a computer program that emulates a Rogerian psychotherapist. In this assignment you will implement a simple version of ELIZA. Your simplified version will be able to talk almost like a standard ELIZA but won't be able to learn from the conversations. You can read more about her

<http://en.wikipedia.org/wiki/ELIZA>

and talk to a sample implementation of ELIZA (although not too long, please):

<http://psych.fullerton.edu/mbirnbaum/psych101/Eliza.htm>

3 Assignment (14 Points)

Lab 8: Eliza - Designing a Client-Server Application

Eliza Warmup

Modify the simplex-talk socket program so that each time the client sends a line to the server, the server sends the line back to the client. The client (and server) will now have to make alternating calls to `recv()` and `send()`.

Assignment

Now you will work on ELIZA. The client program will act ummm... like a client and the server will act like ELIZA. For every line the client sends to the server, the server should do either of the following two actions (chosen randomly).

1) Either it should return that line with another line appended to it and a question mark attached at the end (as in

Client: It's late.

Eliza: What makes you think It's late?

The appended line should be randomly chosen among the following phrases: What makes you think ?

How long since started?

Did you come to me because ?

Attention: if a client asks a question (i.e. their line ends with a question mark), ELIZA should not respond with her own question. Rather, she should say "Tell me more" or something similarly plausible.

2) Or the server should randomly pick and return one of the following lines:

Tell me more...

I'm sorry to hear that.

How does that make you feel?

Is it really?

Are you sure?

These are the minimum requirements. Feel free to add more phrases to your automatic therapist (for instance, she can ask every so often: Are we going to chat? I can't help you without a dialog!)

Submission

Lab 8: Eliza - Designing a Client-Server Application

Submit your client and server program, and text responses on Canvas.