ICS 53, Winter 2018

Lab 5: Client/Server Address Book

You will write a networked client/server application which allows a client program to query a simple database which is managed by a server program. The client program sends a query message containing an email address to the server, and the server responds by sending back a message containing the name of the user who has that email address. For example, if the client sends a message containing the email address "harris@ics.uci.edu" then the server should return the name "Ian G. Harris".

1. Running the Client/Server Address Book

The application is actually two different programs, the client program and the server program. These two programs should both be running as two separate processes which are invoked by the user in a shell. These two programs may be running on the same machine but they may be running on two different machines. Since the client and server are communicating using TCP/IP, you will need to select a port to use for communication. ICS Computer Support has advised me that ports 30000 – 60000 are all available for your use, so you can use any one of those ports.

For the purposes of this explanation we will refer to the domain name of the server machine as "server.ics.uci.edu". Although we use this name in this description, your program should operate correctly for any domain name. We will also assume that the port used is 30000, but your program should be able to communicate on any port.

Starting the Server

If we assume that the name of the server's compiled executable is "add_server" then you would start the server by typing the following at a linux prompt, "add_server 30000". When the server is started it should print "Address server started" and then wait to receive messages from a client.

Starting the Client

If we assume that the name of the client's compiled executable is "add_client" then you would start the server by typing the following at a linux prompt, "add_client server.ics.uci.edu 30000". When the client starts it should print a ">" prompt on the screen and wait for input from the user.

User Interface of the Client

Requests for addresses are made by the user entering an email address at the client's prompt. When an email address is entered at the client's prompt, the client will send a request message containing the email address to the server, and the client will await a response from the server. The server will send a response

message containing the name of the user with the email address. The client will print the received name to the screen, print a prompt on a new line, and wait for more input from the user. An example of a user interaction with the client is shown below.

```
> harris@ics.uci.edu
Ian G. Harris
> joe@cnn.com
Joe Smith
>
```

The client will continue in this loop until the user enters “+++” which will cause the client’s process to exit.

User Interface of the Server

Once running, the server will only accept requests sent from a client over the network. When the server receives a request from a client, the server will print the email address in the message on the screen on a new line. An example of the printed output of the server when communicating with the client is shown below. We assume that the user has made the two address requests shown in the client example above.

```
Address server started
harris@ics.uci.edu
joe@cnn.com
```

The server will continue responding to requests and printing the associated email addresses until its process is killed externally, for instance by a ctrl-c typed at the keyboard.

2. Message Format

Each message sent between the client and the server must contain two pieces of information, a string, and the length of the string. The string will be the email address inside a request sent from the client to the server, and the string will be a name in a response sent from the server to the client. Each message must be shorter than 256 bytes long and must be formatted as follows:

Byte 0: Length of the string

Bytes 1 – n: Characters of the string

3. Address Database

You should hardcode the following address database into the server.

Email Address	Name
harris@ics.uci.edu	Ian G. Harris
joe@cnn.com	Joe Smith
jane@slashdot.org	Jane Smith

4. Implementation Details

1. If the email address typed into the client is not in the database which is known to the server, just return “unknown” as the name.
2. Don’t worry about invalid input typed into the client. Everything entered into the client can be assumed to be a valid email address.

5. Submission Instructions

There will be a dropbox on Canvas which you will use to submit your code. You should submit two C source code files, one for the client and one for the server. **The code must compile and execute on the openlab machines with no explicit options provided to gcc**. The name of your C source code files should be “Lab5_client.c” and “Lab5_server.c”.