Programming Assignment 1

Instructions:

- Complete the project in groups of 2-3
- The grade for this project is weighted as follows:
  - 50% Implementation
  - 20% Functional Tests
  - 30% Write-up
- Late penalty: up to 24 hrs: 20%, 25 - 48hrs: 40%, etc.

Due Date: October 5, at the beginning of class.
(Late penalties apply after the beginning of class.
Reference time is timestamp for email received.)

Objective: Implement an Addressbook service.

Problem Description:

- The Addressbook service is a client-server application. A client sends a query message to the server containing an email address. The server responds to the client with a message that contains the full name which corresponds to the email address.

- A client process Addressbook_client submits query messages to the server. The query message contains an email address.

- The server process Addressbook_Server runs on a known workstation (say, `cobra.cs.virginia.edu`) and listens on a well-known port number for client requests. When a request comes in, the server looks in a database to find the full name which corresponds to the email address, and sends the full name to the requesting client.

- The following messages format is used for query and response:

<table>
<thead>
<tr>
<th>Message Type (1 byte)</th>
<th>AString Length (1 byte)</th>
<th>AString (≤ 255 bytes)</th>
</tr>
</thead>
</table>

- Message Type is set to “Q” (ASCII 82) for queries, and “R” (ASCII 83) for responses.
- AString Length is an unsigned 8-bit integer.
• AString is a character string with up to 255 characters.

Example (simplified):

Requirements:
• Addressbook_Server and Addressbook_Client must be able to run on different machines on the Internet.
• The Address_Client must be able to access a test server, which is running on mamba.cs.virginia.edu on port 5678.
• The database of the server must be able to handle and arbitrary number of entries.
• The server must be able to process multiple queries in a row.
• You get 10% extra credit if your server can handle multiple clients simultaneously.

Your Task:
1. Implement the Addressbook service systems for Unix workstations. Both server and clients are to be implemented exclusively with sockets (Internet domain, stream sockets). The implementation must be done in C or C++.
2. Test your client implementation with the server specified in the requirements.
3. Deliver a max 1 page write-up which describes your implementation approach, a description of your test strategy, and known bug list.

Hints:
1. Implement the server process as a background process (``daemon’’), which is listening on a well-known port number.
2. A client process contacts the server by connecting to the well-known port number on the server machine.
3. The server process accepts the connection request and then handles the Addressbook request. (Note: Alternatively, the server process may fork a child process, where the child process handles the request using a new socket (with a number that is different from the well-known port number.) What are the advantages of such a strategy?

Submission Guidelines:
You will submit your homework electronically using the guidelines posted on the website.