COSC 6360—Operating Systems
Assignment #2:A Public Key Server
Due Wednesday, November 6 at 11:59:59 PM

Objective

You will learn to use stream sockets.

Your Programs

You are to write two programs:

1. A client program that will connect with your server and send it requests for a specific public key.
2. A server program that will wait for connection requests from your client and exchange one-line text messages with it.

The Server Program

Your server must start by reading in a file named **`keys20.txt`** that will contain the email addresses of various users and the public keys of these users:

```
aperez4@uh.edu  0x123456789ABDCEF
bschmidt@uh.edu 0xABDCEF 123456789
…
```

It should then prompt for a port to listen to as in

```
Enter server port number: 2468
```

It will then create a stream **`socket`**, **`bind`** it to the specified port number, do a **`listen()`** to specify a maximum number of queued connection requests and do an **`accept()`** that will let it wait for connection requests.

Whenever the server accepts a connection request, it will receive the email address of a user and reply to the client with the public key of that user. Additionally—and for debugging purposes—it should print each email address it receives as in:

```
Received request for public key of
bschmidt@uh.edu
```

The Client Program

Your client should start by prompting the user for a server host name and a server port number as in:

```
Enter server host name:
program.cs.uh.edu
Enter server port number: 2468
```

or

```
Enter server host name: localhost
Enter server port number: 2468
```

You will have to use the second option if you test your program on a computer that does not have a full Internet address such as **`program.cs.uh.edu`**.

It should then create a stream **`socket`**, do a **`connect()`** request to the specified server, and prompt the user for an email address:

```
Enter the email address of a user:
aperez4@uh.edu
The public key of aperez4@uh.edu is
0x123456789ABDCEF
```

or

```
Enter the email address of a user:
nguyen@utsa.edu
The database had no public key for
user nguyen@utsa.edu
```

Hints

1. Please refer to the two online socket tutorials at: http://www.cs.rpi.edu/~moorthy/Courses/os98/Pgms/socket.html/ http://www.cs.uh.edu/~paris/3360/Sockets.html or through the course Piazza page. It contains a general introduction to sockets. You can include any code from these documents in your submissions.
2. Keep in mind that server and client processes read the messages byte by byte and have no way to know how many bytes they should read. The easiest way to do it is to put your messages into fixed size buffers. Both **`sprintf()`** and **`sscanf()`** could come handy.
3. The table will be short but public keys are likely to fairly big. While these keys will actually be large hexadecimal numbers (128 or 192 digits), your best bet is to treat them as strings.
4. Use a *single-threaded server* to keep things simple. You will not have to not worry about zombies and can safely ignore the **`fireman()`** call in the primer.
5. Yes, you will have to turn in two different programs, namely a client program and a server program.

This document was updated last on Saturday, October 12, 2019.