COSC 4330/6310—Operating System Fundamentals
Assignment #2 for Summer 2016: Managing priorities
Due on Thursday, July 14 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 a single message requesting a priority number;
2. A server program that will wait for connection requests from client processes and send them back their priority number starting at one and incremented by one at each new request.

The Server Program

Your server should start by prompting the user for a port to listen to as in

Enter server port number: 2468

It should 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 it accepts a connection request, it will read in the user name of a client, reply with a priority number and store into a table the user name of the client and its priority number.

In addition, a special user request **"Terminate"** will force the server to print out the contents of the table containing the user name of the clients and their priority number then terminate:

Alice 1
Carol 2
Bob 3
Francis 4
Gina 5
Henry 6
Isabel 7
Done.

The Client Program

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

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

Please note that the only correct answer to the first prompt is the name of the local host as provided by **gethostname( ).** Your client should ***reject*** any other entry and prompt for a new entry.

It will then create a stream **socket**, do a **connect( )** request to the specified server, and prompt for a user name as in:

Enter your user name: Jorge

After that, the client will request a priority number from the server and print it out as in:

Jorge, your priority number is: 25

unless the user entered the user name **"Terminate."**

Hints

1. Please refer to "BSD Sockets: A Quick and Dirty Primer" on the Piazza resource page. It contains a general introduction to sockets. You can include any code from that document in your assignment.
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. You will need to come with a way to let the sender specify either the length of each message or its end using a non printable character.
3. 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.
4. Yes, you will have to turn in two different programs, namely a client program and a server program.
5. User names will be short and never contain spaces or tabs.
6. Each client process will perform a single server request.
7. The server table should be able to store up to 64 requests.

This document was updated last on Tuesday, June 28, 2016.