CSCE 3600: Systems Programming

Minor Assignment 7 – Using Linux Sockets

Due: 11:59 PM on Monday, April 11, 2016

PROGRAM DESCRIPTION:

In this assignment, you will write a complete C program to support a client/server model using Linux sockets for a ticket system (movies, theatre, sports, concert, etc.). Start by reviewing the tutorial found here (http://www.linuxhowtos.org/C_C++/socket.htm). Along with some discussion about sockets, it includes some code for a client/server application using sockets.

This program will consist of a "main" ticket outlet (i.e., the server) that will provide services to "BUY" and "RETURN" tickets to two "local" ticket distributors (i.e., the clients) as follows:

- **Server**
  - The server will generate N tickets, where $N = 20$, and each ticket will be a randomly generated (and unique) 5-digit number. In this case, the server will therefore maintain an array of 20 tickets (i.e., unique 5-digit integers).
  - When asked for a ticket using the "BUY" request, the server will look to see if any remain. If there are available tickets, the server will return the unique 5-digit ticket number to the client. If no tickets remain, the server will return a value (such as a sentinel value) to indicate that all tickets are gone. However, since a client may return tickets using the "RETURN" request, the server must remain available to accept additional requests.
  - When a client returns a ticket using the "RETURN" request, the server will "place" it back into the ticket pool, hopefully to be used later when another "BUY" request comes in from a client.
  - When no clients remain "connected" to the server, the server will complete its operations and shut down.
- **Clients** (note that there are 2 clients)
  - The client will request a ticket from the server using the "BUY" request.
  - The client may also, from time to time, return a ticket to the server using the "RETURN" request. The client should provide the ticket number that is being returned in this request.
  - Each client will make 15 requests to the server, where 80% of the requests should be to "BUY" a ticket and 20% of the requests should be to "RETURN" a ticket. When all of the client requests have been granted, the client will disconnect from the server.

Your program should run so that the server and each of the two clients execute on three different CSE machines at the same time.

Given the randomness of the requests, it could be that not all of the client requests will be met, so you will need to make sure you are able to handle error cases, such as a client trying to return an invalid ticket (why would this happen?) or the server running out of tickets.

SAMPLE OUTPUT (user input shown in **bold green**)**:**

To be updated shortly.

REQUIREMENTS:

- Your code should be well documented in terms of comments. For example, good comments in general consist of a header (with your name, course section, date, and brief description), comments for each variable, and commented blocks of code.
- Your programs should be named "**`svrMinor7.c`**" and "**`cliMinor7.c`**", without the quotes, for the server and client code, respectively.
- Your program will be graded based largely on whether it works correctly on the CSE machines (e.g., `cse01`, `cse02`, …, `cse06`), so you should make sure that your program compiles and runs on a CSE machine.
- Please pay attention to the **SAMPLE OUTPUT** for how this program is expected to work. If you have any questions about this, please contact your instructor, TAs, or IA assigned to this course to ensure you understand these directions.
- This is an individual programming assignment that must be the sole work of the individual student.

SUBMISSION:

- You will electronically submit your two source code files to the **Minor Assignment 7** dropbox in Blackboard by the due date.