

# CNT4704: Analysis of Computer Communication Networks (Fall 2015)

## Programming Assignment 1: Socket programming

(Assigned Sept. 21<sup>st</sup> ; due: Oct. 4<sup>th</sup> midnight)

In this programming assignment, you will implement a client process on **eustis.eecs.ucf.edu** and a server process on **eustis2.eecs.ucf.edu**, respectively, to communicate with each other. If you never SSH login to eustis and eustis2 machines, your account should have the usual NID as your username and default password (Pyymmdd) with your birth year, month and day. If your computer is out of campus network, you need to first VPN to UCF then connect to Eustis machines.

You can use either C, C++, or Java, or Python to program this project. Eustis and eustis2 servers support all these programming languages.

Your client program and server program should operate as follows. Your server runs first on eustis2 machine waiting for your client to connect. The server should use a server port number choosing by yourself and must be bigger than 1024 (smaller than 65535). Your client runs next on Eustis machine.

**Objective**---Simple math calculator: The client program repeatedly requires the user to input a basic math computing question, which will be sent to server program where it will be computed. The result is sent back by the server program and the client program displays it out.

Math computing question has this format: “ $x \text{ op } y =$ ”, where  $x$  and  $y$  are positive real numbers, “ $op$ ” represents math operator of “+”, “-“, “\*”, “/”. The user ends the repeated input procedure by typing “0/0=”. If the user types in an invalid math question, the client should show an error message “**Input error. Re-type the math question again.**”

The server should output the math questions sent by the client, and show the end of client input when receiving “0/0=”, and then shut down. An example interaction between the client and the server is shown below (red colored text are user input parts):

| <b>Client Side:</b>  | <b>Server Side:</b>   |
|--|---|
| Connected with server on [IP address]<br>$20 + 10.5 =$<br>Answer from server: 30.5<br>$13 / 4 =$<br>Answer from server: 3.25<br>$0 / 0 =$<br>User input ends; end the client program | Connected by client on [IP address]<br>Received question “ $20 + 10.5 =$ ”; send back answer 30.5<br>Received question “ $13 / 4 =$ ”; send back answer 3.25<br>Received question “ $0 / 0 =$ ”; end the server program |

When you submit this assignment to me, please submit the following (put in a .zip file):

- A brief report: write a brief description of your program codes, then copy the **screen output images** in your report, showing your client running screen and server running screen (must be screen shot images, not the text you directly copy and paste onto your report). Your client and server interaction **must show each of those four types of math operations at least once**.
- Submit client source code and server source code as attachments in the webcourse submission page. The TA will try to download and run your code by himself. Be sure to explain clearly in your report **how to compile and run your code!**

## Programming notes

Here are a few tips to help you with the assignment:

- On Eustis or Eustis2 machine, use this command line to compile your C program to generate the executable file:  
`gcc -o client client.c (or gcc -o server server.c)`
- You must choose a server port number greater than **1024** (to be safe, choose a server port number larger than 5000). If you choose a port that is currently being used by another user's server program, you have to choose another port.
- Make sure you close every socket that you use in your program before your program ends. If you abort your program, the socket may still hang around and the next time you try to bind a new socket to the same port number you previously used (but never closed), you may get an error "Address already in use". If this happens, you have to wait a minute or so to let the hanging socket time out, or use a different port number.
- Be aware that port IDs, when bound to sockets, are system-wide values and thus other students may be using the port number you are trying to use.