Programming Assignment #1

CSCE 5580 - Computer Networks

Spring 2018

100 Points

Due: 02/23/2018, 11:59 PM

Instructions:

- Compile the C programs and make sure they are working.
- Comment your code.
- Create a readme text file that describes the working and usage of the code.
- Please create a zip archive of your assignment folder (readme, code, and header files) and upload the zip file.
- Not following the above instructions could result up to 20% deduction from your assignment score.

Objective:

Create a math server that can compute simple math numerical expressions that is requested by a client.

Requirements:

1. Create a C-based client-server architecture using sockets
2. The math server should run on cse01.cse.unt.edu machine and the client should run on cse02.cse.unt.edu machine.
3. The math server should be able to accept and service at least one client's requests.
4. The math server should be able to compute simple numerical expression with at most six non-zero integer values.
5. The numerical expression could have spaces between operations and operands.
6. The numerical expression cannot have multiple inner brackets or inner expression.

7. The math server should be able to perform addition, subtraction, multiplication, divide, square root, power (x^y), exponential (e^x), and log.
8. Square root and log does not compute for negative values as input.
9. The results should have at least four places of precision.
10. The math server should be able to quit computation when a quit expression is given as input.
11. The math server should parse the input expression and identify the operands and operations and perform the computation.
12. Do not use functions for expression evaluations or use any Linux shell commands or code from the internet. You could use pre-defined functions for square root, power (xy), exponential (ex), and log computations.

Procedure:

1. Create a C-based server that can accept at least one client's request using sockets.
2. Make sure the math server runs on cse01.cse.unt.edu and the format to start the math server as follows

   mserver <port_number>

   where mserver is the math server executable and port_number is the port number on which the math server listens.
3. Create a C-based client that can connect to the math server using sockets
4. Make sure the client runs on cse02.cse.unt.edu and connects the math server. The user can request the numerical expression using the below format

   client <port_number>

   exprsn:(5+6)*(6+5)

   where client is the client executable, port_number is the port number on which the client connects the server and exprsn is the requested numerical expression
5. Once the math server gets a request from the client, it computes the value of the expression and replies it to the client.
6. The math server quits computation when a quit expression is given as input.
7. An example client interaction with the server is shown below.

Client side input/output example:

```
client 1234
exprsn: (-5*6) + (5+5)
result: -20.0000
exprsn: (10^2) + (5/5)
result: 101.0000
exprsn: (e^2) + (10 - 5)
result: 12.3890
exprsn: sqrt(2) + log(2)
result: 1.7152
exprsn: (5*6) + (5+)
result: Invalid expression
exprsn: (5*6) + (-5)
result: 25.0000
exprsn: sqrt(2) + log(-2)
result: Invalid expression
exprsn: quit
result: bye bye!
```

Deliverables:

1. Commented server and client C code
2. A readme file that describes how to compile, execute, and test the C codes.