

CS 512 - Programming Assignment 1

Summary

In this assignment, you will implement a client/server RPC protocols. The goal is to use simple TCP to build a communication channel so a client can request services using simple RPCs.

Description

The following client.c (p1/client.c) and server.c (p1/server.c) are a simple implementation of TCP socket connection which is described in the textbook. You can compile and run client and server on two different putty windows (both connected to uxb4.wiu.edu). You should get the following output.

On the server side

```
hb117@uxb4:~$ gcc server.c -Wall -o server
hb117@uxb4:~$ ./server
Socket successfully created..
Socket successfully binded..
Server listening..
server acccept the client...
From client: Hello, world!
      To client : Hi there,
From client: exit
      To client : exit
Server Exit...
```

On the client side:

```
hb117@uxb4:~$ gcc client.c -Wall -o client
hb117@uxb4:~$ ./client
Socket successfully created..
connected to the server..
Enter the string : Hello, world!
From Server : Hi there,
Enter the string : exit
From Server : exit
Client Exit...
```

As you can see, the client and the server establish a socket connection and start exchanging messages using read(), write() system calls.

If you see the error: Address already in use, someone is already running their server and using PORT 8080, just change the value of PORT in both client.c and server.c to a random port and try again.

```
hb117@uxb4:~$ gcc server.c -Wall -o server
hb117@uxb4:~$ ./server
Socket successfully created..
socket bind failed...: Address already in use
```

The server should understand the following RPCs: *add*, *subtract*, *multiple* and *division* and send the results back to the client.

If the client requests something else, "not supported" will be sent back.

These RPCs: *add*, *subtract*, *multiple* and *division* should only accept 2 arguments. It is the job of the **client** to validate the user input, and print out an error message.

To make things more interesting, the server will not send a raw number back the client, it will send a string instead. For example:

negative one instead of -1
seven instead of 7
one nine instead of 19
three zero eight instead of 308

...

Your implementation should work like this:

On the client side:

```
Socket successfully created..  
connected to the server..  
Enter the request : add 3 4  
From Server : 7  
Enter the request : add 3 2 100  
add only supports 2 arguments!  
Enter the request : sub 33 14  
From Server : 19  
Enter the request : sub 33 34  
From Server : -1  
Enter the request : mul 3 4  
From Server : 12  
Enter the request : div 30 10  
From Server : 3  
Enter the request : mod 30 10  
From Server : not supported  
Enter the request : exit  
From Server : exit  
Client Exit...
```

On the server side:

```
Socket successfully created..  
Socket successfully binded..  
Server listening..  
server accept the client...  
From client: add 3 4  
From client: sub 33 14  
From client: sub 33 34  
From client: mul 3 4  
From client: div 30 10  
From client: mod 30 10  
From client: exit  
Server Exit...
```

Server and client will terminate when client sends server "exit" message.

You should use `strtok` (<http://linux.die.net/man/3/strtok>) to tokenize the string that the server gets from the client. Take a look at this example (`p1/input.c`) to see how *strtok* works.

Additionally, you may want to use `strncmp` (<http://linux.die.net/man/3/strncmp>) to compare 2 strings, use `atoi` (<http://linux.die.net/man/3/atoi>) to convert a string to an integer, and use `sprintf` (<http://linux.die.net/man/3/sprintf>) to construct a string to send the result back to the client.

Grading

This assignment is worth 100 points. Grade will be based on:

1. Correct implementation of client capability: 35 points.
2. Correct implementation of server capability: 55 points.
3. Indent/comment your code properly: 10 points.

Turning in

Upload your source code (`client.c` and `server.c`) to Western Online Dropbox. Programming Assignment 1 is due on Friday, September 25th at midnight.

