Programming assignment 1 Grading Rubric

(2020-10-01, 10.20 PM EDT) Some addition and clarifications:

- I missed the `exit` command in the instruction. That command is supposed to close the frontend only. If you don't have an `exit` command implemented for the frontend, it's fine. The grader should be able to close the command line by just sending an EOF character.
- The `quit` command in this rubric also refers to the `shutdown` action in the original instructions. This command is supposed to close the frontend and send a shutdown signal to the backend. We'll accept both keywords.
- When the last client arrives (the client that comes after the shutdown signal is issued), the backend can either close this connection right away or choose to serve it like other existing clients. But no more clients should be accepted after this point.
- For the 3 points mentioned above, we'll make sure to be flexible when grading those parts of the program. For the most part you don't need to change anything in your current program. Sorry for the inconvenience.

Shell commands

Commands	Description	Output	Error
`add x y`	Returns the sum of two integers x and y	A single integer	None
`multiply x y`	Returns the product of two integers x and y	A single integer	None
`divide x y`	Returns the result x / y	A floating point number with exactly 6 digits after the decimal point or `Error: Division by zero`	Division by zero

Commands	Description	Output	Error
`factorial x`	Returns the factorial of x	An integer	x is guaranteed to be in the range [0, 20]
`sleep x`	Sleeps for x seconds	None	None
`quit` or `shutdown`	Terminates the frontend and send a shutdown signal to the backend	Optional goodbye message	None
`exit`	Terminates the frontend only	Optional goodbye message	None
Other	Unsupported commands	`Error: Command "<name>" not found`	Command not found

For the arithmetic calculations, you can assume that there is no overflow.

Shell requirements

- Prompt user for input: print 2 greater than signs and a space for a prompt, i.e `>>`
- Read a single line.
- Execute the command and prints the result back to stdout.
- Remember to flush stdout everytime you output something.
- We will not check for output from backend, so you can log any message there.

An example is provided at the end of this document.

Grading rubric

	Points
Implement `add`, `multiply`, `divide`, `factorial`, `sleep`	10%
Parse user input	5%
Establish client-server connection	5%
Message passing	5%
Handle division by zero and command not found	5%
Handle 1 connection	10%
Handle 5 concurrent connections	35%
`quit` or `shutdown` properly shuts down the server	10%
Memory management	5%
Code quality (commenting, error handling, code style, Makefile)	10%

There is no specific requirements for the RPC_* functions, but try to follow the suggestions that the Professor has given. You can decide on the function parameters and return types if it helps simplify your implementation.

Makefile

Add your compile target to the provided Makefile similar to the example. We should be able to compile your program by running `make rpc`.

Your frontend executable should be named `frontend` and backend executable should be named `backend`.

Running the backend

```
./backend <host_ip> <host_port>
Server listening on <host_ip>:<host_port>
```

Running the frontend

```
./frontend <host_ip> <host_port>
>> add 10 2
12
>> multiply 4 5
20
>> divide 99 10
9.900000
>> divide 99 0
Error: Division by zero
>> factorial 6
720
```

```
>> subtract 10 30
Error: Command "subtract" not found
>> print hello world
Error: Command "print" not found
>> quit
Bye!
```

Submission

Submit your source code, Makefile in a zipfile with the format `StudentName_ID.zip`. You can add an optional README in the zipfile. The README can include instructions to run your program, and some difficulties you encountered when doing the assignment. Please don't submit any executable or the example client/server files from the starter code.

A sample directory structure looks like this

```
----
      a1_lib.c
      a1_lib.h
      backend.c
      backend.h
      frontend.c
      frontend.h
      rpc.h
      Makefile
      <Optional README.txt>
      <other *.c and *.h files>
```

Questions

Please post your questions in the discussion board for assignment 1. We will try to answer as quickly as possible.