

This project is due at 11:59:59pm on January 21, 2015 and is worth 1% of your grade. You must complete it with a partner. You may only complete it alone or in a group of three if you have the instructor's explicit permission to do so for this project.

1 Description

This assignment is intended to familiarize you with writing simple network code. You will implement a client program which communicates with a server using sockets. The server will ask your program to solve hundreds of simple mathematical expressions. If your program successfully solves all of the expressions, then the server will return a secret flag that is unique for each student. If you receive the secret flag, then you know that your program has run successfully, and you will receive full credit for the assignment.

2 Language

You can write your code in whatever language you choose, as long as your code compiles and runs on unmodified CCIS Linux machines on the command line. Do not use libraries that are not installed by default on the CCIS Linux machines. Similarly, your code must compile and run on the command line. You may use IDEs (e.g. Eclipse) during development, but do not turn in your IDE project without a Makefile. Make sure your code has no dependencies on your IDE.

3 Protocol

The server runs on the machine `cs3600tcp.ccs.neu.edu` and listens for requests on a TCP socket bound to port 27993. This exercise has four types of messages: HELLO, STATUS, SOLUTION, and BYE. Each message is an ASCII string consisting of multiple fields separated by spaces (0x20) and terminated with a line feed (0x0a, \n). The maximum length of each message is 256 bytes. Messages are case sensitive.

The protocol works as follows. The client initiates the protocol by creating a TCP socket connection to the server. Once the socket is connected, the client sends a HELLO message to the server. The format of the HELLO message is:

```
cs3700spring2015 HELLO [your NEU ID]\n
```

In your program you should replace [your NEU ID] with your NEU ID (including any leading zeroes). You must supply your NEU ID so the server can look up the appropriate secret flag for you. The server will reply with a STATUS message. The format of the STATUS message is:

```
cs3700spring2015 STATUS [a number] [a math operator] [another number]\n
```

The three variable fields represent a simple mathematical expression, e.g. "5 + 10". The server may return plus, minus, multiplication, or division expressions. All numbers will be between 1 and 1000. Your program must solve the mathematical expression and return the answer to the server in a SOLUTION message. The SOLUTION message has the following format:

```
cs3700spring2015 [the solution]\n
```

It is okay for the solution to be negative. In the case of division, round the answer down to the nearest integer (do not send floating point numbers to the server).

The server will respond to the SOLUTION message with either another STATUS message, or a BYE message. If the server terminates the connection, that means your solution was incorrect. If the server sends another STATUS message, your program must solve the expression and return another SOLUTION message. The server will ask your program to solve hundreds of expressions; the exact number of expressions is chosen at random. Eventually, the server will return a BYE message. The BYE message has the following format:

```
cs3700spring2015 [a 64 byte secret flag] BYE\n
```

Once your program has received the BYE message, it can close the connection to the server. If the server returns "Unknown_Husky_ID" in the BYE message, that means it did not recognize the NEU ID that you supplied in the HELLO message. Otherwise, the 64-byte string is your secret flag; write this value down, since you need to turn it in along with your code.

4 Your client program

Your client program must execute on the command line using the following command.

```
$ ./3700client <-p port> [hostname] [NEU ID]
```

Your program must follow this command line syntax exactly, i.e. your program must be called `3700client` and it must accept these exact parameters in exactly this order. If you cannot name your program `client` (i.e. your program is in Java and you can only generate `client.class`) then you must include a script called `client` in your submission that accepts these parameters and then executes your actual program. Keep in mind that all of your submissions will be evaluated by grading scripts; if your program does not conform exactly to the specification then the grading scripts may fail, which will result in loss of points.

The `-p port` parameter is optional; it specifies the TCP port that the server is listening on. If this parameter is not supplied on the command line, assume the port is 27993. The `[hostname]` parameter is required, and specifies the name of the server (either a DNS name or an IP address in dotted notation). The `[NEU ID]` parameter is required. Your code must support NEU IDs that have leading zeroes (do not strip them!).

You should manually record the secret flags that are given to all team members into a file `secret_flags`. You should include one flag per line, with one line for each team member. You will submit this file along with your project.

Your program should print exactly one line of output: the secret flag from the server's BYE message. If your program encounters an error, it may print an error message before terminating. Your program should not write any files to disk, including writing to the `secret_flags` file.

5 Other Considerations

You may test your client code with our server as many times as you like. Your client should conform to the protocol described above, otherwise the server will terminate the connection silently.

Your client program must verify the validity of messages by strictly checking their format, i.e. the server may send corrupted messages just to try and crash your software. If a received message is not as expected, such as an incorrect field or wrong message type, you must assert an error and terminate your program. You should be strict; if the returned message does not exactly conform to the specification above, you should assert an error. Remember that network-facing code should be written defensively.

6 Grading

This project is worth 1% of your grade. If your program compiles, and you successfully submit the secret flags of all group members, then you will receive full credit. We will check all teams' code to make sure that it works correctly. All student code will be scanned by plagiarism detection software to ensure that students are not copying code from the Internet or each other.

7 Submitting your project

7.1 Registering your team

You and your partner should first register as a team by running the `/course/cs3700sp15/bin/register` script. You should pick out a team name (no spaces or non-alphanumeric characters, please) and run

```
/course/cs3700sp15/bin/register project0 <teamname>
```

This will either report back success or will give you an error message. If you have trouble registering, please contact the course staff.

You must register your team by 11:59:59pm on January 14, 2015.

7.2 Submission

For the final submission, you should submit your (thoroughly documented) code along with a plain-text (no Word or PDF) README file. In this file, you should describe your high-level approach, the challenges you faced, a list of properties/features of your design that you think is good, and an overview of how you tested your code. You should also submit your `secret_flags` file.

You should submit your project by running the `/course/cs3700sp15/bin/turnin` script. Specifically, you should create a `project0` directory, and place all of your code and README files in it. Then, run

```
/course/cs3700sp15/bin/turnin project0 <dir>
```

Where `<dir>` is the name of the directory with your submission. Again, the script will print out every file that you are submitting, make sure that it prints out all of the files you wish to submit! You should receive an email confirmation of your submission.

You must submit your project by 11:59:59pm on January 21, 2015.

8 Advice

A few pointers that you may find useful while working on this project:

- Check the Piazza forum for question and clarifications. You should post project-specific questions there first, before emailing the course staff.
- Finally, get started early and come to the instructor's office hours and TA lab hours. You are welcome to come to the lab and work, and ask the TA any questions you may have.