Assignment 1 Marking Scheme and FAQs

Hi all,

Please find below marking scheme and some FAQs related to Assignment 1 and their corresponding answers. Please go through them first before posting any question.

Best of luck!

CS456/A1 Marking Scheme

Note: Each bullet indicates the penalty of marks deducted if that requirement is not satisfied. However, the total deduction should not exceed the maximum marks allocated to each sub-heading.

[10] Assignment Presentation (NOT awarded if everything else fails)

- [5] Makefile and README
  - -3 Makefile not present or not working
  - -2 How to compile and make the programs
  - -1 Where (which machines)?
  - -2 What are the executables and parameters?
- [5] Code Style & Comments
  - -4 Un-modular
  - -2 More complicated than needed
  - -3 Poor comments

[10] General Features

- -3 Ports are built in as numbers
- -2 If fixed ports fail the program doesn't use a different port
- -3 No invalid/ missing parameter warnings
- -3 Wrong input format

[40] Negotiation Stage Using UDP Sockets

- -30 Negotiation stage not working

-20 Client cannot send the negotiation signal

-20 Server cannot receive the signal

-10 Client does not confirm <r_port>

-10 Server does not acknowledge the receipt of <r_port>

-5 Server does not print the <n_port>

[40] Transaction Stage Using TCP Sockets

-30 Transaction stage not working

-25 Client cannot send the message

-25 Server cannot reply back

-10 Message is NOT reversed

-5 Client does not print the reversed message

Generic-FAQ

1. Do I need to worry about UDP being unreliable in my code?

Ans: No need to worry about packet loss, connection instability, and so on. That means, you can consider that the message will be transferred as it is over the network.

2. How should the server choose the <n_port>? Can I just pick a port number and hard code this into my program? Or do I need to choose it randomly?

Ans: Start with a fixed port (e.g., 1234). If it is occupied, then try with the next one (i.e, 1235) and so on until you find a free one. Note when you are going to test whether the port is available or not (by actually trying to use it in the bind method for C++ or in the DatagramSocket method for Java), it will result an exception if the port is not free. So, if-else condition checking will not work here. Rather, you need to use try-catch in such cases.

3. How should the server choose the <r_port>?

Ans: You can use the same approach as mentioned in FAQ#2. Alternatively, you can use any language specific function which will return a free port. For example, in Java, ServerSocket call with parameter 0 will automatically allocate a free port.

3. What exception/error handling do we have to do in this assignment?

Ans: You need to check only the number and formats of the command line arguments passed to the server and the client.

4. What should the size of the buffer for the input string be? Will 1024 byte suffice?

Ans: 1024 bytes is sufficient.

5. Is there a limit to how big our messages can be?

Ans: Size of message will be at most 1024 bytes.

6. What is the purpose of the request code in this assignment?

Ans: Request code needs to be checked in the server. If the client does not send the exact request code, then the server does nothing and will not create the TCP port <r_port>. Additionally, server can print this exception in the console.

7. How many Makefiles should be there?

Ans: A single make file should be used to compile both client and server simultaneously.

8. Should we make server mutli-threaded?

Ans: No need to do for this assignment. But this is an additional feature that you can include. Note no marks will be given for that

9. A client might crash after creating a TCP connection with the server. Do we need to handle this exception?

Ans: No need to do for this assignment. But this is an additional feature that you can include. Note no marks will be given for that.

10. Can the string to be reversed contain the newline character? For example

./client.sh <server_add> <n_port> <req_code> "Hello \n World"

Ans: There will be no newline character within the string.

11. I cannot run the server.sh or client.sh scripts.

Ans: Run the following commands:

chmod +x server.sh
chmod +x client.sh

C++-FAQ

1. Is there any recommended socket library?

Ans: For C++ implementation, the socket implementation through GNU C library (using sys/socket.h header) is preferable.

2. What resource can I use if I'm trying to do the assignment in C++?

Ans:
You can go through these tutorials:
http://www.tutorialspoint.com/unix_sockets/socket_server_example.htm
http://www.linuxhowtos.org/C_C++/socket.htm
http://beej.us/guide/bgnet/output/html/singlepage/bgnet.html

3. When I send 'elppA' to the server, I am expecting 'Apple' as the reversed string from the server. However, instead, the reversed string occasionally looks like this (getting some weird symbols added at the end of the message):
Apple?)?
or
Apple?O
So far, I figured that the server returns a correct reversed message and this is what's causing this:

char output[len]; // len: size of the message
recvFrom(...) // receive the msg from the server and store it to output
cout << output << endl; <-- problem on this line

This is only happening on some random messages with random sizes.

Ans: The output array is not null terminated. This causes cout to continue reading from memory until it gets to the null terminating character.

See:
http://www.cplusplus.com/reference/string/string/c_str/

Java-FAQ

1. What does the Makefile do?

Ans: A Java Makefile will only compile the *.java files (e.g. server.java and client.java) and produce server.class and client.class That is, it will run all required javac commands for the user.

Python-FAQ

1. Do we need to submit a Makefile for python?

Ans: No need to submit a Makefile with Python implementation of the assignment. Marks allocated for

Makefile will be distributed among the README, coding Style, and Comments.

Ruby-FAQ

1. Do we need to submit a Makefile for Ruby?

Ans: No need to submit a Makefile with Ruby implementation of the assignment. Marks allocated for Makefile will be distributed among the README, coding Style, and Comments.