CS436/636 Programming Assignment 1 - 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 <r_port>? Can I just pick a port number and hard code this into my program? Or, do I need to choose it randomly?**

Ans: You can do it in any of the following two ways:

a) **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.**

b) You can use any language specific function which will return a free port.

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

Ans: You need to check 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 should close the TCP connection to that client (in the negotiation port) and will not create the UDP port. Additionally, server can print this exception in the console.

7. **What should the client do if it sends an incorrect req_code?**

Ans: The client should exit gracefully/in a controlled manner. In other words, it should not crash due to an unhandled exception or something.

8. **Can the string to be reversed contain the newline character or escape characters?**

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

9. **Can the string to be reversed contain space?**

Ans: Yes.

10. **What's the format of the <server_address> ?**

Ans: Either IP address or hostnames (e.g., ubuntu2204-008) or both are fine. Include in the readme what's supported.

11. **If there is only one client at a time in the system, why are we still listening on <n_port> when the client is in the stage 2?**

Ans: The server should continue listening after one communication (negotiation + transaction) is done. We want to keep the connection open on <n_port> to handle future client requests. If you close/reopen again, <n_port> might not be free. By continuing to listen on <n_port>, we get rid of this issue.

12. **When would the server terminate? Do we need to handle the termination of the server?**

Ans: The server should stay alive unless you kill it (e.g., ctrl+c). No marks will be deducted if you don't handle the termination of the server.

13. **How can I log into a particular linux.student.cs host? For example, ubuntu2204-008?**

Ans: ssh userid@ubuntu2204-008.student.cs.uwaterloo.ca

14. **Which programming language can I use?**

Ans: Any language available in the student.cs machines. Be sure to include instructions that TAs might need to run your programs.

15. **Have problems with external SSH access to CS systems ?**

Ans: IST no longer allowed password-only access to on-campus servers from off-campus. You need to use either a VPN, an SSH key or password & 2FA if accessing from off-campus.