Università della Svizzera italiana

Faculty of Informatics

Distributed Systems **Autumn 2022/23**

Code Assignment 2 — Chat Server Part 2: Server Operator

The deadline to uploading your source code to iCorsi is on the 20th of October, at 16:30. Late submission policy will apply; see the Intro slides for details. Upload a single Java project, with separate packages for each exercise. If you are using a language other than Java, include instructions on how the execute the code and list any dependencies needed.

Exercise 1 - (3 points)

Spawn a thread in the server to work as the server operator. This way it will be possible interact with the server while it is running. The operator should display the number of users connected via the command "num_users".

To test the operator, start the server and check "num_users" when different amounts of clients are connected.

Exercise 2 - (3 points)

The interaction between clients and servers involved only sending text messages so far, it is time to create a message format to enable the sending of structured data. We are going to use Google's protobuf to design our format. Install protobuf both for Java (or the language of your choice) and for your operating system, and then compile the format file provided using the command "protoc --java_out=. msgFormat.proto". In our format a normal message will have the following fields:

1. int32 **fr**, specifying who's sending the message;
2. int32 **to**, specifying to whom the message is addressed;
3. string **msg**, specifying the message content.

Integrate protobuf to your project and test the client-server interaction. The server should echo the message back to the client, as done in the previous assignments.

Java tip: if you want to run your code from the CLI, compile all files using your IDE and run a command similar to "java -cp protobuf-java-3.21.7.jar:./ exercise1.Server" in the parent folder of where the `.class` files are located; "cp" stands for classpath and includes the jar and the current folder, separated by a colon. Remember to have the protobuf jar in the folder!

Exercise 3 - (4 points)

Create another message in the format file, with the goal of enabling a handshake. It should have the following fields:

1. int32 **id**;
2. bool **error**.

When interacting with the server, the client should first automatically do the handshake, with the server answering with the same handshake message. If there is an error we should have "error = True", otherwise "error = False". Make the client and server work as per the specification above.