Networked Systems Programming (A.Y. 2023/24)

Lab n. 6 – November 9, 2023

Prof. Eugenio Zimeo

Exercise 6.1

By exploiting the concurrency model provided by Java threads, implement a Full Duplex version of the Chat implemented with exercise 4.3. To this end, consider that the code to read from stdin and to write onto the network channel must be executed concurrently with the code used to read from the network channel and to write onto the standard output, since both code fragments use a blocking method (read). To this end, you can implement two thread classes: `Writer` and `Reader`. The former (run method) reads strings from the standard input and writes them onto the network channel whereas the latter (run method) reads strings from the network channel and writes them onto the standard output.

Exercise 6.2

Write a Java multi-user application for "instant messaging". The application should be composed of a server able to serve several client processes. Every client can send messages (strings) to the other ones connected to the server. Client-to-client communication must be handled by the server: whenever a client sends a string, it is received by the server and forwarded to all clients connected to the server (broadcast). To this end, the server should use a thread (`MultiUserChatServer`) for accepting clients connection and a thread (`MultiUserChatHandler`) to handle each connected client. `MultiUserChatServer` and `MultiUserChatHandler` should share an object of type `Room`, which is used to handle the active connections by storing them in a collection.

The client of this application can be implemented by leveraging the code of the full-duplex chat of exercise 6.1: the client handler should terminate its execution when its user writes '.' at end of the input string.

Test the application by running both the server and the clients on the same computer.

Upload the exercises of this lab into your shared folder before November 13, 2023.