Universidad de los Andes

ISIS-4218 CPD
Actors
Fecha: April 7, 2024

For this assignment we will develop a chat room (90's version of direct messaging). The behavior of a chat room is quite simple. (1) Users can join a chat room at any time, immediately getting access to messages posted (from that moment on), (2) any user can post messages in the chat room, and (3) at any time users can leave the chat room, no longer receiving messages.

Your task is to implement a chat room with N users, so that they all can exchange messages, without any of the messages getting lost.

Task 1. For the third (Elixir) implementation of the chat room you must explore the use of Actors as a model to solve the concurrency problems that may appear.

For this task you are required to:

- The main module for the application must be called `Chat`. The application execution must start from here, creating a chat room
- The function to add users to a chatroom must be called `add_user`, passing the user as parameter
- The function to remove a user from the chat must be called `user_delete`, passing the user as parameter
- The function to write a message in the chat must be called `write_message`, passing the message to write. this function must be executed by the user

As there are many ways to implement actor behavior in elixir (*e.g.,* agents, supervisors) We will strive to use the low level implementation of actors (*i.e.,* `send` and `receive`).

Task 2. Write a report with the analysis on the similarities and differences in the solutions (*i.e.,* mutex, STM, CSP). This report is meant to be a reflection on the concurrency models covered in the course based on your hands on experience.

This work is to be developed in groups (two, max three people). You must hand in a project/file with the implementation of our work (with all group members), as well as the instructions on how to execute it, and an example program for the execution. Additionally you should hand in a pdf file with the concurrency models analysis.