

Universitatea Alexandru Ioan Cuza IASI

Facultatea de Informatica

Strîmbu Andrada-Ştefana

Grupa 2B4

An universitar: 2022-2023

`andrada.strimbu@gmail.com`

1 Introducere

Proiectul "Offline Messenger" dezvoltă o aplicație client/server care să permită schimbul de mesaje între utilizatorii conectați la server. Aceștia pot trimite mesaje către alți utilizatori conectați cât și către cei offline. Cei offline au posibilitatea să citească mesajul când se vor conecta la server.

2 Tehnologii utilizate

În dezvoltarea aplicației, voi alege să implementez un server TCP concurent, fiind mai sigur decât un server UDP, deoarece servere UDP nu garantează primirea completă și corectă a datelor transmise.

Voi folosi thread-uri deoarece ocupă mai puțină memorie și sunt mai rapide, iar comunicarea între ele se realizează ușor.

Pentru a reține și organiza informațiile despre utilizatori, dar și istoricul conversațiilor, voi utiliza biblioteca Json. Json gestionează datele din fișier în mod similar fișierelor XML.

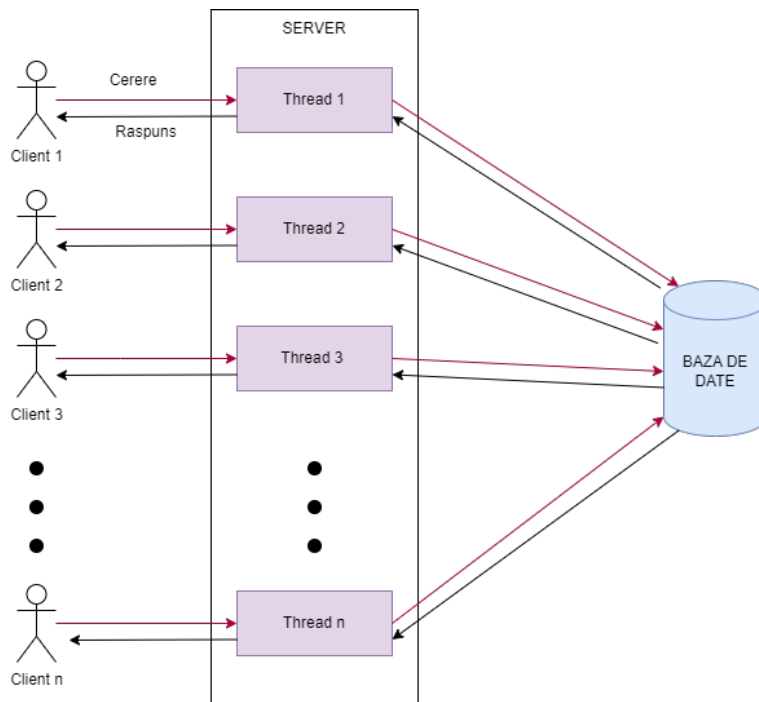
Am ales să utilizez fișiere Json în detrimentul fișierelor txt, deoarece acest tip de fișiere reține informația într-un mod organizat, ocupând mai puțin spațiu. Accesarea mai facilă a datelor este un alt avantaj.

3 Arhitectura aplicației

3.1 Conceptele implicate

Pentru implementarea aplicației, voi utiliza următoarele noțiuni: server, client și bază de date. Server-ul poate gestiona mai mulți clienți în același timp. Acesta comunică cu baza de date și validează cererile primite cu ajutorul bazei de date create. Dacă informația cerută de server se găsește în baza de date, acesta trimite răspunsul înapoi serverului. În final serverul trimite informația înapoi la client.

3.2 Diagrama aplicației detaliată



4 Detalii de implementare

4.1 Cod relevant proiectului

Comunicarea între server și client o voi realiza cu ajutorul unui socket. Am ales socket în detrimentul pipe-urilor deoarece este bidirecțional, deci nu trebuie să creăm două canale de comunicare separate.

Pentru thread-uri voi utiliza biblioteca "pthread" și mă voi folosi de funcția detach() pentru a detașa thread-ul curent.

În main() vom găsi funcția "raspunde". Tot în main() vom regăsi și socket-ul utilizat pentru comunicarea server/client.

4.2 Scenarii de utilizare

Un utilizator se va conecta cu ajutorul username-ului memorat în fișierul "username.txt".

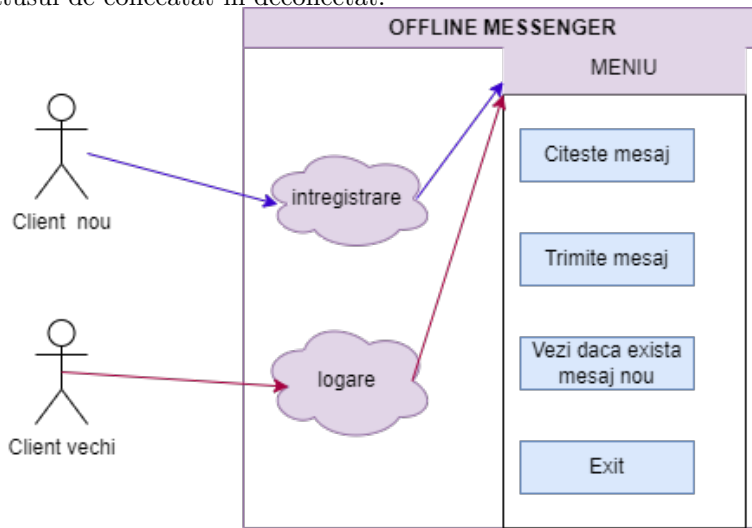
Odată conectat utilizatorul va avea de ales din mai multe opțiuni: fie să vadă mesajele, fie să trimită un anumit mesaj cuiva, fie să se deconecteze și să iasă.

Dacă un utilizator dorește să vadă mesajele cu o anumită persoană trebuie să îi scrie username-ul, iar serverul va trimite informația la funcția "read-mesage". Funcția citește din fișierul Json doar mesajele ce sînt ale utilizatorului.

Dacă un utilizator dorește să trimită un mesaj către un utilizator îi va scrie username-ul și mesajul pe care dorește să îl transmită.

Dacă un utilizator dorește să își citească mesajele noi va scrie username-ul persoanei care crede ca i-a trimis un mesaj nou.

Dacă un utilizator dorește să se delogheze va scrie numarul corespunzator iar in server se va schimba statusul de conectat in deconectat.



5 Concluzii

Pentru a îmbunătăți aplicația am putea realiza o interfață grafică. Un alt mod de îmbunătățire ar putea fi oferirea de reacții la mesaje, trimiterea de fișiere între utilizatori sau crearea unui grup cu mai mulți utilizatori.

References

1. <https://profs.info.uaic.ro/computernetworks/cursulaboratorul.php>
2. <https://profs.info.uaic.ro/andreis/index.php/computernetworks/>
3. <https://www.geeksforgeeks.org/differences-between-tcp-and-udp/>
4. <https://learn.microsoft.com/en-us/host-integration-server/core/iterative-vs-concurrent-tcp-ip-models1>
5. <https://www.geeksforgeeks.org/thread-in-operating-system/>
6. <https://www.geeksforgeeks.org/thread-in-operating-system/>
7. <https://stackoverflow.com/questions/tagged/json>