

Offline Messenger

Romascu Raluca-Loredana, grupa B3, anul II

Universitatea Alexandru Ioan Cuza, Iasi
An Universitar 2021-2022

1 Introducere

Am ales proiectul “Offline Messenger” , proiect ce implementează o aplicație client/server care va permite schimbul de mesaje între utilizatori. Pentru schimbul de mesaje, utilizatorii trebuie să fie înregistrați și conectați, dar va fi funcțională și trimiterea mesajelor către utilizatorii care sunt offline, acestora aparându-le mesajele atunci când se conectează la server. Aplicația va oferi istoricul conversațiilor pentru și cu fiecare utilizator în parte și de asemenea, utilizatorii vor avea posibilitatea să trimită un răspuns în mod specific la un anumit mesaj primit (reply).

2 Tehnologii utilizate

Având în vedere că principalul scop al aplicației create este de a transmite mesaje, iar aplicația este de tip client-server, aleg să implementez un server TCP concurrent.

TCP (Transmission Control Protocol) este un protocol pentru comunicarea client-server care controlează transferul de date astfel încât acesta să fie de încredere.

Astfel, TCP asigură fiabilitatea în timpul transmiterii datelor. Toate datele ajung la destinație, nu se pierde nimic pe parcurs, ceea ce este necesar aplicației noastre, căci dorim transmiterea integrală/completă a mesajelor (asemenea protocolului de email). Totodată pachetele de date sunt reasamblate în ordine, o altă proprietate necesară aplicației “Offline messenger”, căci textul trebuie să fie transmis în aceeași formă, fără amestecarea literelor/cuvintelor și în aceeași ordine cronologică, fiind nevoiți să gestionăm un istoric al conversației.

Pentru realizarea arhitecturii de server concurrent, am ales să utilizez `fork()`, creând pentru fiecare client câte un proces copil care va rezolva cererea, astfel clienții for putea fi “serviți” simultan .

Cu ajutorul bibliotecii SQLite3, voi implementa o baza de date SQL pentru a gestiona informațiile necesare aplicației. SQLite3 este o mică bibliotecă C care implementează un motor de baze de date SQL încapsulat, oferă posibilitatea de a-l introduce în diverse sisteme și necesită zero-configurare.

3 Arhitectura aplicatiei

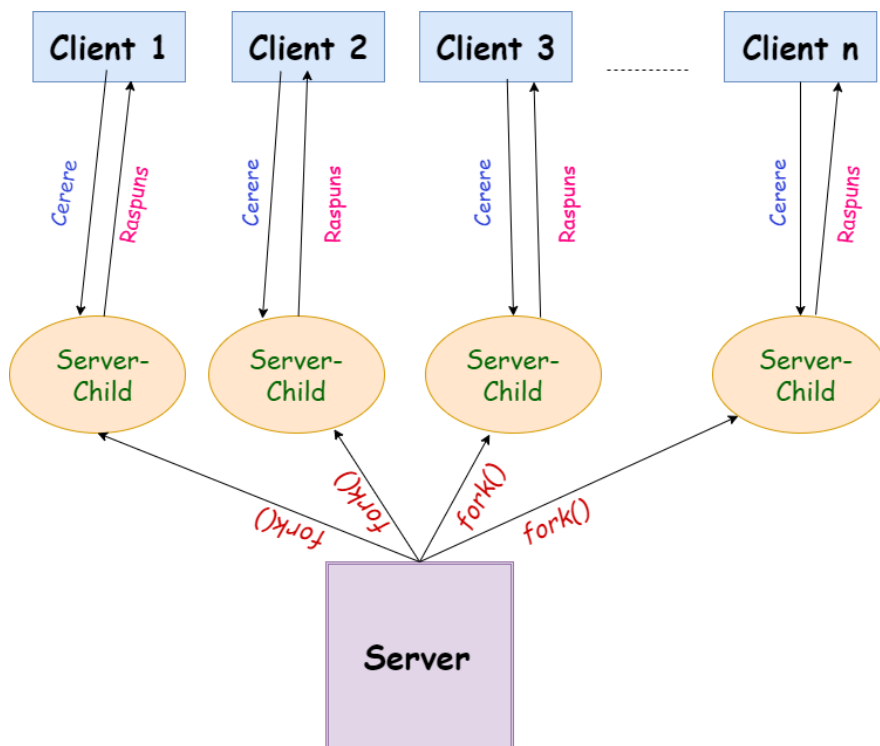
3.1 Concepte implicate

Arhitectura aplicației implică: serverul, clienții (pot fi mai mulți, conectați simultan) și baza de date.

Baza de date va conține două tabele, un tabel Users care va gestiona utilizatorii înregistrați și dacă aceștia sunt sau nu conectați și un tabel Messages care va reține în câte o coloană mesajele transmise, utilizatorul care transmite și utilizatorul care primește, momentul de timp la care a fost trimis mesajul.

Conform modelului clasic client-server TCP, clienții fac cereri pe care serverul le va prelua. Serverul va face legătură cu baza de date și va răspunde clienților, aceștia având posibilitatea de a formula în continuare alte cereri.

3.2 Diagrama aplicatiei



4 Detalii de implementare

4.1 Cod relevant proiectului

Pentru a asigura comunicarea între server și client vom utiliza socket. Utilizarea socketului ne asigură comunicarea bidirecțională și este suficient unul singur pentru a transmite mesajele de la client la server și invers. Așa cum am precizat mai sus, concurența este asigurată de `fork()`. Se va crea un copil pentru fiecare client, astfel cerințele clienților pot fi îndeplinite de către server simultan, acestea fiind rezolvate de copiii creați.

4.2 Scenarii de utilizare

Voi descrie protocolul folosit de aplicație, adică sub ce formă va forma cererile clientul și ce răspunsuri va primi. Un utilizator care va deschide aplicația “Offline Messenger” va avea posibilitatea de a se înregistra sau autentifica (cu nume utilizator și parolă), după caz. Dacă autentificarea are loc cu succes, va avea posibilitatea să scrie un mesaj nou pentru alt utilizator, să vadă mesajele noi primite cât timp a fost deconectat sau poate vizualiza conversația cu alt utilizator și răspunde la un anumit mesaj al acestuia. De asemenea există posibilitatea de deconectare-logout și de închidere a aplicației-quit.

Login: utilizatorul va trebui să introducă un nume de utilizator și o parolă. Dacă numele de utilizator este inexistent în baza de date, i se va cere să se înregistreze prin comandă “Register:”. Dacă parola este introdusă greșit, se transmite mesajul: “conectare eșuată” și se va reface login-ul. Dacă username-ul și parola corespund unui utilizator înregistrat în baza de date, clientul primește mesajul “conectare efectuată cu succes” și va putea efectua alte comenzi.

Register: dacă un utilizator nou va dori să se înregistreze în aplicație, va introduce un username și o parolă. Dacă username-ul este deja existent se va transmite un mesaj corespunzător, altfel clientul primește mesajul “utilizator logat cu succes” și va putea apoi să efectueze comanda login și celelalte comenzi disponibile.

Logout: utilizatorul va fi deconectat.

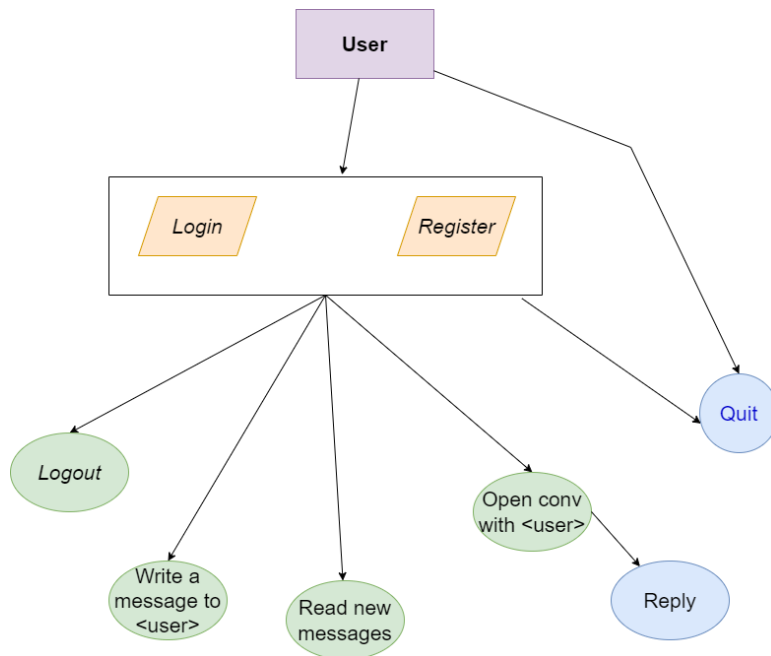
Quit: se va încheia conexiunea cu serverul.

Write a message to <user>: dacă utilizatorul nu este logat va primi un mesaj conform căruia comanda nu poate fi procesată. Dacă utilizatorul este logat, dar <user> nu a fost găsit în lista utilizatorilor va primi un mesaj corespunzător, altfel va introduce textul dorit.

Read new messages: dacă utilizatorul nu este logat va primi un mesaj conform căruia comanda nu poate fi procesată. Dacă utilizatorul este logat, I se vor afișa mesajele primite, împreună cu utilizatorii de la care acestea au fost trimise.

Open conv with <user>: dacă utilizatorul nu este logat va primi un mesaj conform căruia comanda nu poate fi procesată. Dacă <user> nu se găsește în baza de date a utilizatorilor, utilizatorul primește un mesaj corespunzător, altfel se va afișa istoricul conversației .

Reply at <number>: după ce a fost afișat istoricul conversației cu un <user>, utilizatorul are posibilitatea de a răspunde la un anumit mesaj, precizat prin <number>, un număr de ordine ce identifică fiecare mesaj din conversație.



5 Concluzii

Așadar aplicația creată , "Offline Messenger" asigură transmiterea de mesaje între mai mulți utilizatori conectați la un server, păstrează istoricul conversației și oferă posibilitatea de reply, putând fi folosită cu succes în comunicare. O serie de îmbunătățiri care ar putea fi aduse proiectului ar fi :

- modificarea parolei în cazul în care aceasta a fost uitată
- crearea unei conversații secrete(criptate) cu o anumită persoană
- prezența unei interfețe grafice

-posibilitatea conversației dintre un număr mai mare decât 2 (grupuri de 3 sau mai multe persoane).

6 Bibliografie

1. <https://ro.wikipedia.org/wiki/SQLite>
2. <https://www.sqlite.org/index.html>
3. https://profs.info.uaic.ro/~computernetworks/files/7rc_ProgramareaInReteaIII_Ro.pdf
4. https://profs.info.uaic.ro/~computernetworks/files/5rc_ProgramareaInReteaI_ro.pdf
5. <https://profs.info.uaic.ro/~ioana.bogdan/>
6. <https://stackoverflow.com/>