

Raport Proiect - Part2Part (A)

Rotariu Cosmin - Andrei, IIA4

January 2021

1 Introducere

În această **fișă de raport** voi prezenta anumite detalii ce descriu funcționalitățile proiectului pe care am ales să îl implementez, proiect numit **Part2Part**.

Cerința proiectului constă în crearea unei aplicații de tip peer-to-peer ce permite partajarea fișierelor între clienții conectați în acel moment la server. Pentru comunicație va trebui să folosesc socket-uri, iar programul server va trebui să fie capabil să servească simultan mai multe cereri de transfer provenite de la clienți.

2 Tehnologii utilizate

Voi implementa o aplicație **server-client** de tip **TCP concurent**, folosind **socketi** pentru comunicare, și, implicit **thread-uri** pentru a asigura conectarea mai multor clienți la server și pentru a-i servi simultan.

Avantajul TCP-ului[1] față de UDP[1] ar fi faptul că TCP[1] furnizează un flux sigur de date între două sau mai multe calculatoare aflate în rețea și asigură stabilirea unei conexiuni permanente între acele calculatoare pe parcursul comunicării, având siguranța că informațiile trimise de un proces vor fi recepționate corect și complet la destinație, fără pierdere.

Pentru fiecare client care se va conecta, serverul TCP va crea câte un thread[3], după cum am menționat puțin mai sus, și astfel se va facilita servirea simultană a celor conectați în rețea.

Se va putea, astfel, comunica între server și client, și, implicit, între mai mulți clienți conectați în același timp la rețea.

Îmi propun să utilizez și o bază de date de tip **SQLITE**[7] pentru a stoca informații despre conturile existente și, implicit despre fișierele existente în sistemele clienților conectați la rețea, mai exact numele fișierului, adresa, și portul, așadar la fiecare cerere, serverul va ști la ce client se află fișierul căutat.

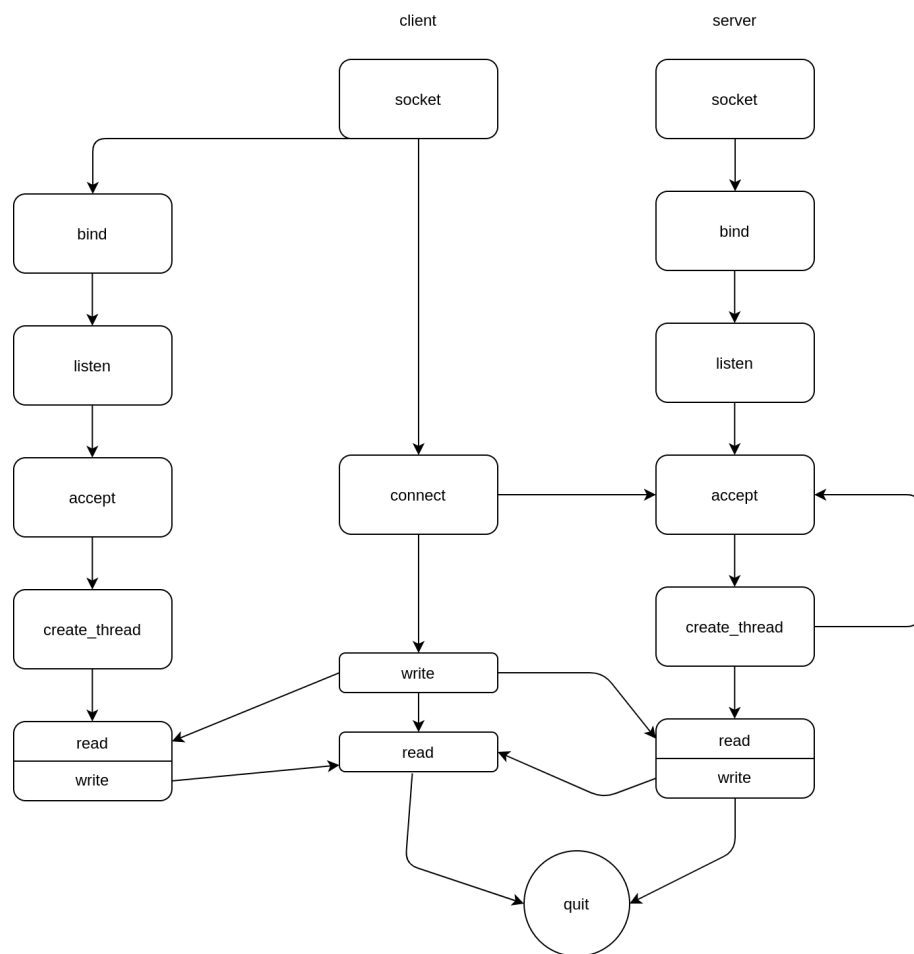


Figure 1: Diagramă TCP concurent cu thread-uri

3 Arhitectura aplicației

Descrierea diagramei de mai jos: În urma introducerii de către un client a comenzii ce face referire la intenția de a descărca un anumit fișier, serverul analizează baza de date și îi transmite clientului care a făcut request toate fișierele disponibile în acel moment în rețea, alături de adresa clientului țintă ce deține fiecare dintre acele fișiere. Eventual, clientul mai poate cere serverului să i se permită să partajeze și el un fișier în acea rețea.

Pentru a funcționa aplicația, este necesar ca minim 2 clienți să fie conectați la server.

Clienții vor avea posibilitatea de a alege:

- să afle ce fișiere sunt disponibile în rețea, iar apoi să se conecteze și să descarce un anumit fișier de la un client
- să distribuie și ei un fișier în acea rețea

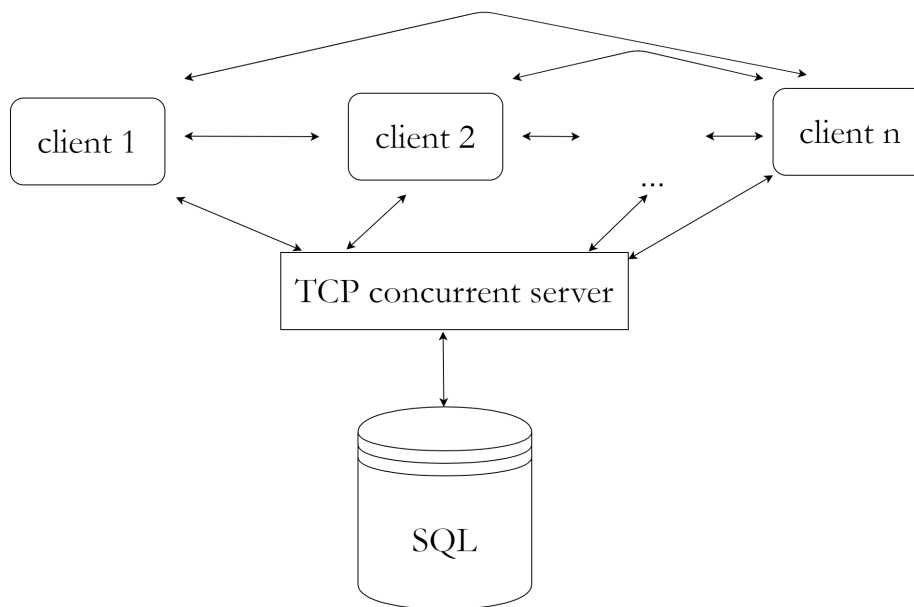


Figure 2: Arhitectura aplicației

Serverul:

- primește informația cum că un client dorește să afle ce fișiere sunt disponibile în rețea
 - accesează baza de date *files*
 - trimite clientului toate informațiile din baza de date
- îi oferă posibilitatea unui client de a partaja un fișier cu întreaga rețea
- apelează baza de date *login* pentru a-i oferi acces clientului la funcționalitățile aplicației

4 Detalii de implementare

Pentru implementarea acestui protocol de comunicare pentru **Part2Part**, am creat un fișier *server.c* și unul *client.c*.

Comunicarea dintre *server* și *client* se realizează prin *socketi de comunicare* și *thread-uri*. Am ales să folosesc *thread-uri* în loc de *fork-uri* deoarece *thread-urile* din motive de performanță și deoarece se recepționează mai rapid mesajele dintre *server* și *client*.

Astfel, *clientul* se conectează la *server*, se autentifică folosind una dintre comenzile *login* sau *register* și i se afișează un meniu cu diferite comenzi pe care le poate executa.

Meniul va avea 4 comenzi: "**share**", "**list**", "**download**" și "**exit**".

- Dacă clientul va alege **prima** comandă, și anume *share*, acesta va trebui să introducă numele fișierului pe care dorește să îl partajeze în rețea, acel fișier fiind astfel vizibil și pentru ceilalți clienți conectați la rețea în acel moment. După primirea datelor de la client, serverul va uploada în baza de date numele fișierului, alături de adresa clientului deținător, apoi va întoarce un mesaj pozitiv, caz în care uploadarea s-a efectuat cu succes, iar în caz contrar, va întoarce un mesaj care indică faptul că fișierul există deja în baza de date.
- Dacă clientul va alege **a doua** comandă, și anume *list*, acesta va primi de la server lista cu toate fișierele share-uite la momentul actual în rețea, alături de portul clientului deținător, după care este îndrumat de către server să introducă a treia comandă
- Dacă clientul va alege **a treia** comandă, și anume *download*, acesta va trebui să introducă de la tastatură numele fișierului pe care dorește să îl descarce din rețea, alături de portul clientului la care dorește să se conecteze. Clientul cu adresa respectivă va fi notificat că cineva s-a conectat și că dorește să descarce un fișier, astfel începând descărcarea.

- Dacă clientul va alege a **patra** comandă, și anume *exit*, se va deconecta de la server și, în plus, i se vor șterge din baza de date toate fișierele partajate în rețea de către acesta.

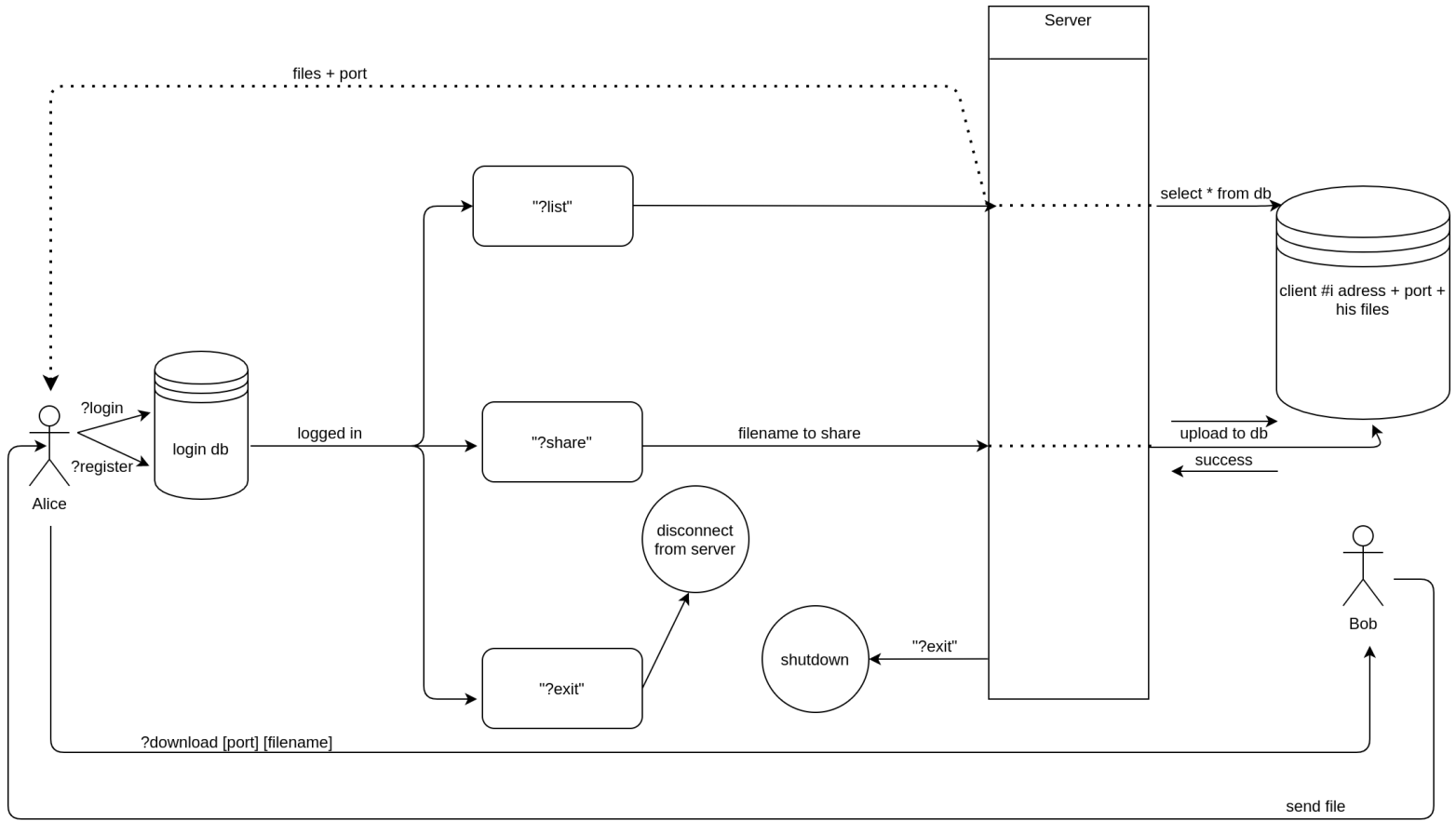


Figure 3: Use Cases

5 Concluzie

Această aplicație este utilă mai ales persoanelor care doresc să partajeze anumite date confidențiale și nu doresc să folosească software-urile deja existente pe piață. De asemenea, permite partajarea de fișiere într-un mod sigur și fără pierderi.

Soluția propusă ar putea fi îmbunătățită prin adăugarea pe viitor a unei interfețe, fiindu-le mult mai ușor clienților să interacționeze cu funcțiile programului.

References

- [1] https://ftp.utcluj.ro/pub/users/civan/CPD/2_LABORATOR/06_Socket/6_SK.pdf
- [2] <https://app.diagrams.net/>
- [3] <https://sites.google.com/view/fii-rc/>
- [4] <https://profs.info.uaic.ro/computernetworks/>
- [5] <https://www.overleaf.com/>
- [6] <https://notes.shichao.io/unp/>
- [7] <https://www.sqlite.org/cintro.html>
- [8] http://www.student.apamaravathi.in/materials/32cnlab/07_32cnlabexp.pdf
- [9] <https://www.youtube.com/watch?v=-C5HSdPU3TI>