

# Raport tehnic-ReadsProfiler

Ursache Ana-Maria [2B1]

Decembrie 2023

## 1 Introducere

În cadrul proiectului ReadsProfiler, se realizează o platformă similară cu o bibliotecă online, în care clienții pot căuta anumite titluri de cărți sau autori, în care aceștia se pot interesa de anumite genuri și subgenuri existente, dar și descărca acele cărți care sunt pe gustul lor.

Prin intermediul acestui proiect doresc să fac această platformă de tip bibliotecă să fie un loc în care se poate căuta și descărca anumite cărți într-un mod ușor, clar, ce nu impune probleme în utilizarea sa.

Câteva beneficii relevante ale acestui proiect sunt din punct de vedere al timpului și al ușurinței navigării prin diferitele informații stocate, deoarece algoritmul în sine va oferi detalii clare și concise pentru fiecare operațiune posibilă la un anumit moment.

## 2 Tehnologii aplicate

În cadrul acestei teme am decis să folosesc modelul client/server de tip TCP(Transmission Control Protocol).

Am ales această tehnologie pentru că, la acest proiect, este necesară asigurarea faptului că informația este primită în întregime și în ordinea inițială, fără nicio pierdere, la trecerea de la client la server, sau invers, motiv pentru care conceptul de Three-way Handshake este unul esențial pentru buna implementare a proiectului ales.

Spre deosebire de protocolul TCP pe care l-am ales, protocolul UDP(User Datagram Protocol) este de foarte mare ajutor în cazul în care este necesar multicasting-ul și broadcasting-ul, deoarece în aceste cazuri nu este nevoie să ne asigurăm că informația a fost trimisă și a fost primită în întregime.

Prin alegerea protocolului TCP, îmi voi asigura astfel transferarea corectă a informației și îmi voi ușura procesul de transmitere a mesajelor între client și server, sau invers.

De asemenea, în cadrul acestui proiect am utilizat librăria SQLITE3, ce a trebuit inițial instalată, pentru stocarea informațiilor necesare, precum datele referitoare la cărți, la istoric, la utilizatorii conectați, la ierarhiile de genuri și subgenuri, la numele autorilor și genurile pe care le abordează în operele lor, dar

și la rating. Prin intermediul bazelor de date am reușit să rețin informația într-un mod eficient și bine structurat, motiv pentru care alegerea acestei tehnologii îmbunătățește utilizarea acestei aplicații client/server.

Pe lângă această librărie pentru bazele de date, am mai folosit și librăria CURL pentru a putea clientul, cu ușurință, să descărcare o anumită carte care îi este pe plac.

### 3 Structura aplicației

În cadrul proiectului am utilizat conceptul de meniu de opțiuni pentru a oferi oportunitatea de a fi o navigare ușoară în cadrul acestei platforme ce mimează o bibliotecă online.

Am folosit modelul client/server de tip TCP concurent și astfel am format 2 documente: client.c(în care se va redirecționa ceea ce va scrie clientul) și server.c(în care serverul va identifica acele comenzi oferite de client și va întoarce rezultatul dorit).

Funcționalități ale bibliotecii:

- "Login: username"
- "Căutare după gen: gen"(doar după "Login: username")
- "Căutare după subgen: subgen"(doar după "Login: username" și după "Căutare după gen")
- "Căutare după autor: autor"(doar după "Login: username")
- "Căutare după titlu: titlu"(doar după "Login: username")
- "Căutare după an: an"(doar după "Login: username")
- "Căutare după ISBN: ISBN"(doar după "Login: username")
- "Căutare după rating: rating"(doar după "Login: username")
- "Istoric"(doar după "Login: username")
- "Recomandări"(doar după "Login: username")
- "Carti ce vor apare curand" (doar după "Login: username")
- "Descărcare: carte"(doar după "Login: username")
- "Afisare"(doar după "Login: username")
- "Tabela"(doar după "Login: username")
- "Rating"(doar după "Login: username" și după "Descărcare")
- "Quit"
- Va fi posibilă și verificarea comenzii(dacă nu este scrisă una din cele de mai sus, se va afișa un mesaj ce indică acest lucru)

Diagrama ce arată modul în care se va putea naviga în cadrul bibliotecii:

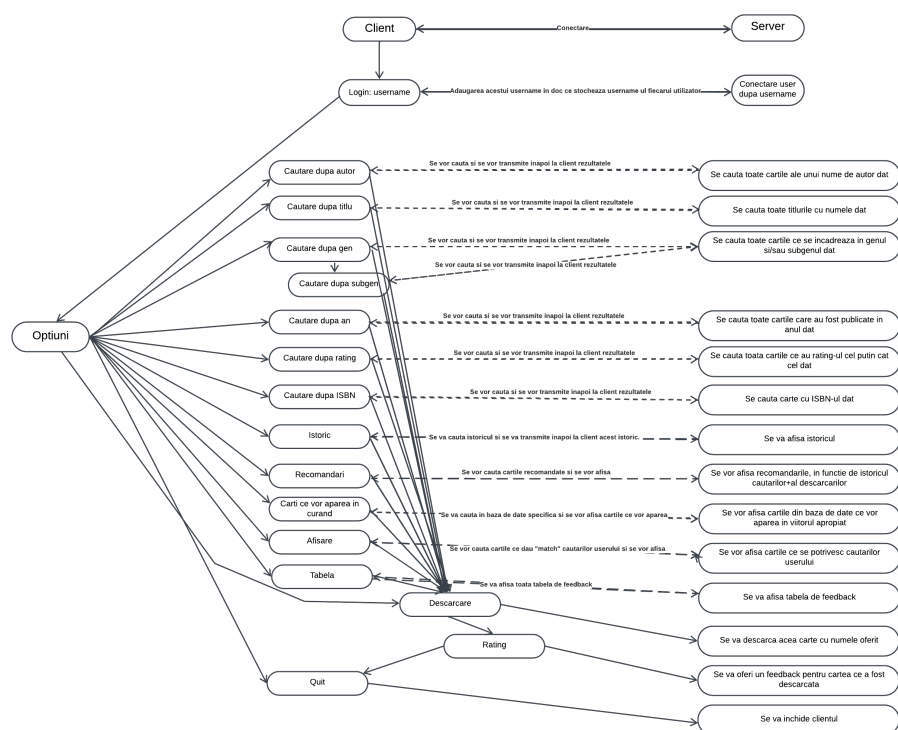


Figure 1: Diagrama 1

## Modelul client/server de tip TCP concurrent

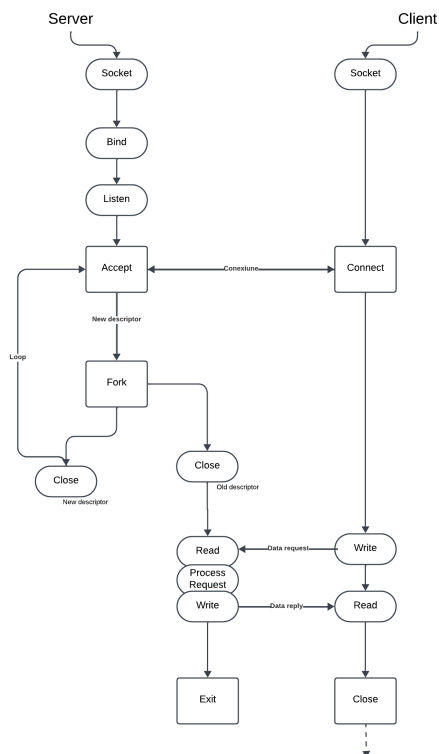


Figure 2: Diagrama 2

## 4 Aspecte de implementare

În cadrul acestei secțiuni voi expune anumite porțiuni de cod ce ajută ca implementarea algoritmului să fie eficientă.

Pentru a putea sincroniza cu ușurință utilizarea comenzilor, am folosit 3 mapări diferite.

- *login<sub>val</sub>*: prin care mă asigur că utilizatorul este logat înainte de a putea folosi alte comenzi;
- *gen<sub>val</sub>*: prin care mă asigur că utilizatorul poate cauta un subgen doar dacă acesta a căutat anterior un gen;
- *descarcare<sub>val</sub>*: prin care mă asigur că utilizatorul poate da rating la experiența sa/ la cartea descărcată, abia după ce a descărcat o carte.

Modul în care le-am inițializat: acestea au fost folosite sub forma unui OK(OK1, OK2, OK3) și au avut doar valoarea 0, inițial, și, mai apoi, 1.

```
int *login_val = mmap(NULL,
                      sizeof(int),
                      PROT_READ | PROT_WRITE,
                      MAP_SHARED | MAP_ANONYMOUS,
                      -1,
                      0);
if (login_val == MAP_FAILED)
{
    handle_error("Eroare la mapare!", 19);
}
*login_val = 0;

int *gen_val = mmap(NULL,
                   sizeof(int),
                   PROT_READ | PROT_WRITE,
                   MAP_SHARED | MAP_ANONYMOUS,
                   -1,
                   0);
if (gen_val == MAP_FAILED)
{
    handle_error("Eroare la mapare!", 19);
}
*gen_val = 0;

int *descarcare_val = mmap(NULL,
                          sizeof(int),
                          PROT_READ | PROT_WRITE,
                          MAP_SHARED | MAP_ANONYMOUS,
                          -1,
                          0);
if (descarcare_val == MAP_FAILED)
{
    handle_error("Eroare la mapare!", 19);
}
*descarcare_val = 0;
```

Figure 3: Inițializare

Modul în care le-am utilizat:

```
else if ((strstr(msg, "Cautare dupa subgen")))
{
    if (*OK == 1&&*OK2==1)
    {
        char msgrasp[100];
        bzero(msgrasp, 100);
        strcpy(msgrasp, "S-a gasit comanda de cautare dupa subgen.");
        if (write(client, msgrasp, 100) <= 0)
        {
            perror("[server]Eroare la write() catre client.\n");
            return;
        }
        else
            printf("[server]Mesajul a fost trasmit cu succes.\n");
    }
    else
    {
        printf("Nu se poate utiliza aceasta operatie.\n");
        char msgrasp[100];
        bzero(msgrasp, 100);
        strcpy(msgrasp, "Nu se poate utiliza aceasta operatie");
        if (write(client, msgrasp, 100) <= 0)
        {
            perror("[server]Eroare la write() catre client.\n");
            return;
        }
        else
            printf("[server]Mesajul a fost trasmit cu succes.\n");
    }
}
```

Figure 4: Folosire mapări

Modul în care le-am folosit în cadrul fork-ului din loop-ul inițial:

```
while (1)
{
    int client;
    int length = sizeof(from);

    client = accept(sd, (struct sockaddr *)&from, &length);
    if (client < 0)
    {
        perror("[server]Eroare la accept().\n");
        continue;
    }

    int pid;
    if ((pid = fork()) == -1)
    {
        close(client);
        continue;
    }
    else if (pid > 0)
    {
        close(client);
        while (waitpid(-1, NULL, WNOHANG))
        {
            ;
        }
        continue;
    }
    else if (pid == 0)
    {
        close(sd);
        while (1)
        {
            operatii(client, login_val, gen_val,descarcare_val);
        }
        exit(0);
    }
    close(client);
    exit(0);
}
```

Figure 5: Loop

Modul în care le-am închis pentru a mă asigura că nu intervin probleme:

```
    if (munmap(gen_val, sizeof(int)) == -1)
    {
        handle_error("Eroare la inchiderea maparii!", 39);
    }
    if (munmap(descarcare_val, sizeof(int)) == -1)
    {
        handle_error("Eroare la inchiderea maparii!", 39);
    }

    if (munmap(login_val, sizeof(int)) == -1)
    {
        handle_error("Eroare la inchiderea maparii!", 40);
    }
}
```

Figure 6: Inchidere mapări

În cadrul funcției de Login, în timp ce se folosește această comandă, numele utilizatorului se retine în baza de date.

```
strncpy(username, msg + strlen("Login: "), strlen(msg) - strlen("Login: "));

if (sqlite3_open("useri_conectati.db", &useri) != SQLITE_OK)
{
    perror("Nu se poate deschide aceasta baza de date.\n");
}

char *eroare = 0;
char intrare[1000];
sprintf(intrare, "INSERT INTO useri (Username) VALUES ('%s');", username);
sqlite3_exec(useri, intrare, 0, 0, &eroare);

sqlite3_close(useri);

char mesaj[100];
strcpy(mesaj, "A fost conectat user-ul cu numele: ");
strcat(mesaj, username);
strcat(mesaj, ".");
strcpy(msggrasp, mesaj);
```

Figure 7: Login



În cadrul funcției de Descărcare, prin faptul că în comanda clientul vă introduce un nume, vă face să se descarce automat cartea cu acel nume.

```

else if ((strstr(comenzi, "Descarcare: ")))
{
    char msg[100];
    bzero(msg, 100);

    if (read(sd, msg, 100) < 0)
    {
        handle_error("Eroare la read", 14);
    }
    else
    {
        strcpy(carte, msg);

        if (strcmp(carte, "1984") == 0)
        {
            Descarcare_carte_dorita("1984.pdf", "https://rauterberg.employee.id.tue.nl/lecturenotes/DDM10%20CAS/Orwell-1949%201984.pdf");
        }
        else if (strcmp(carte, "SecretulMenajerei") == 0)
        {
            Descarcare_carte_dorita("SecretulMenajerei.pdf", "https://101books.club/pdf/2288/Freida-McFadden-Secretul-menajerei.pdf");
        }
        else if (strcmp(carte, "OFataDinBucati") == 0)
        {
            Descarcare_carte_dorita("OFataDinBucati.pdf", "https://101books.club/pdf/1435/o-fata-din-bucati-kathleen-glasgow.pdf");
        }
        else if (strcmp(carte, "OViataMarunta") == 0)
        {
            Descarcare_carte_dorita("OViataMarunta.pdf", "https://101books.club/pdf/1748/o-viata-marunta-hanya-yanagihara.pdf");
        }

        strcpy(msg, "Cartea s-a descarcat.\n");
        printf("[client]%%s\n", msg);
    }
}

```

Figure 8: Descărcare-cărți

Tabela principală în care vom retine informațiile legate de cărți:

```

char *tabela_carti = "CREATE TABLE IF NOT EXISTS biblioteca("
                    "NR INTEGER PRIMARY KEY AUTOINCREMENT, "
                    "Titlu TEXT NOT NULL, "
                    "Autor TEXT NOT NULL, "
                    "Gen TEXT NOT NULL, "
                    "Subgen TEXT NOT NULL, "
                    "Rating INTEGER NOT NULL, "
                    "ISBN INTEGER NOT NULL, "
                    "An INTEGER NOT NULL"
                    ");";

```

Figure 9: Tabelă-cărți

Afișare recomandărilor în funcție de cărțile pe care acel ușor deja le-a căutat și instalat:

```
if (!strstr(continut_istoric, "1984"))
{
    nr++;
}
if (!strstr(continut_istoric, "SecretulMenajerei"))
{
    nr++;
}
if (!strstr(continut_istoric, "OFataDinBucati"))
{
    nr++;
}
if (!strstr(continut_istoric, "OViataMarunta"))
{
    nr++;
}

if (nr != 0)
{
    char meesaj[100];
    strcpy(meesaj, " ");
    if (!strstr(continut_istoric, "1984"))
    {
        strcat(meesaj, "|1984|");
    }
    if (!strstr(continut_istoric, "SecretulMenajerei"))
    {
        strcat(meesaj, "|SecretulMenajerei|");
    }
    if (!strstr(continut_istoric, "OFataDinBucati"))
    {
        strcat(meesaj, "|OFataDinBucati|");
    }
    if (!strstr(continut_istoric, "OViataMarunta"))
    {
        strcat(meesaj, "|OViataMarunta|");
    }
    strcpy(msgrasp, meesaj);
}
else
    strcpy(msgrasp, "Nu avem recomandari pentru dumneavoastra la acest moment.");
```

Figure 10: Recomandări

Modul în care se afișează conținutul unei tabele:

```
if (sqlite3_open("biblioteca.db", &biblioteca) != SQLITE_OK)
{
    perror("Nu se poate deschide aceasta baza de date.\n");
}
struct Continut_baza_de_date select;
select.continut = calloc(1, sizeof(char));

char *eroare = 0;
char *intrare2 = malloc(sizeof(char) * (strlen(intrare) + 1));

strcpy(intrare2, intrare);

int rez = sqlite3_exec(biblioteca, intrare2, parsare, &select, &eroare);
if (rez != SQLITE_OK)
{
    fprintf(stderr, "Eroare: %s\n", eroare);
    sqlite3_free(eroare);
}
strcpy(msgrasp, select.continut);
free(select.continut);
```

Figure 11: Afișare-conținut-tabela

În cadrul funcției de afișare, se actualizează și istoricul cu ultima comandă de căutare pe care a introdus-o clientul:

```
char intrare10[1000];
char *eroare10 = 0;

if (sqlite3_open("istoric.db", &istoric) != SQLITE_OK)
{
    perror("Nu se poate deschide aceasta baza de date.\n");
}

sprintf(intrare10, "INSERT INTO istoric (Username, Descarcare, Titlu, Autor, Gen, Subgen) VALUES ('%s',0,'%s', '%s', '%s', '%s');", username, c.titlu, c.autor, c.gen, c.subgen);
printf("intrare: %s\n", intrare10);

sqlite3_exec(istoric, intrare10, 0, 0, &eroare10);

sqlite3_close(istoric);
```

Figure 12: Actualizare-istoric

În cadrul funcției de descărcare, se actualizează și istoricul cu această carte pe care acesta a descărcat-o:

```
char intrare10[1000];
char *eroare10 = 0;

if (sqlite3_open("istoric.db", &istoric) != SQLITE_OK)
{
    perror("Nu se poate deschide aceasta baza de date.\n");
}

sprintf(intrare10, "INSERT INTO istoric (Username, Descarcare, Titlu, Autor, Gen, Subgen) VALUES ('%s',1,'%s', '', '', '');", username, c.titlu);
printf("intrare: %s\n", intrare10);

sqlite3_exec(istoric, intrare10, 0, 0, &eroare10);

sqlite3_close(istoric);
```

Figure 13: Actualizare-istoric-descărcare

## 5 Concluzii

Prin intermediul acestui proiect se poate vedea modul în care se poate face o platforma după modelul client/server de tip TCP concurent, dar, totodată,

arată și modul în care se poate folosi o bibliotecă online.

Unele lucruri ce ar aduce un plus acestui proiect ar fi:

- formarea unei interfațe grafice ce va îmbunătăți modul în care se navighează în cadrul bibliotecii, deoarece va fi mai mult o activitate vizuală și nu ar mai implica scrierea în terminal, scriere ce poate fi puțin complicată pentru unii utilizatori;
- asigurarea siguranței userului în momentul navigării, prin implementarea unei metode ce impune securitatea clienților;
- utilizarea firelor de execuție/threads pentru o mai bună eficiență a codului.

## 6 Referințe bibliografice

1. [https://profs.info.uaic.ro/~computernetworks/files/5rc\\_ProgramareaInReteaI\\_ro.pdf](https://profs.info.uaic.ro/~computernetworks/files/5rc_ProgramareaInReteaI_ro.pdf)
2. [https://profs.info.uaic.ro/~computernetworks/files/6rc\\_ProgramareaInReteaII\\_Ro.pdf](https://profs.info.uaic.ro/~computernetworks/files/6rc_ProgramareaInReteaII_Ro.pdf)
3. [https://profs.info.uaic.ro/~computernetworks/files/7rc\\_ProgramareaInReteaIII\\_Ro.pdf](https://profs.info.uaic.ro/~computernetworks/files/7rc_ProgramareaInReteaIII_Ro.pdf)
4. [https://docs.google.com/presentation/d/1reUzYxEYVd1WjqvNNy-wKNurJsS57\\_0xn7ciMss6KU8/edit#slide=id.g29c6eba0386\\_0\\_5a](https://docs.google.com/presentation/d/1reUzYxEYVd1WjqvNNy-wKNurJsS57_0xn7ciMss6KU8/edit#slide=id.g29c6eba0386_0_5a)