

Top Music

Georgescu Florin-Adrian, X4

14-01-2020

1 Introducere

Proiectul consta in crearea unei aplicatii pe principiul client/server. Fiecare client se va putea conecta la server, loc in care va putea executa diferite comenzi pentru a opera cu diferite melodii. Aplicatia se apropie foarte mult ca structura cu aplicatia Spotify sau orice alta aplicatie pentru organizarea melodiilor in topuri.

2 Tehnologiile utilizate

Pentru realizarea acestei aplicatii, vom folosi protocolul TCP/IP, care are ca si scop principal, transmiterea cat mai exacta a informatiilor. Spre deosebire de protocolul UDP, unde informatiile sunt transmise rapid, dar cu sanse de pierderi, protocolul TCP, ne va transmite informatiile in siguranta, complete, dar mai greu.

Necesitatea protocolului TCP este una triviala, deoarece avem nevoie de o transmitere exacta a comenzilor executate de client, respectiv in aceeasi ordine ca cea data de client. De asemenea, dorim ca serverul sa nu ai pierderi de informatii si sa recunoasca toate comenzile efectuate de catre client.

3 Arhitectura aplicatiei

Aplicatia necesita un server TCP concurent, deoarece, vom avea conectati la server, mai multi clienti simultan, care vor executa comenzi. Concurenta am realizat-o prin crearea mai multor *threaduri* pentru fiecare client in parte.

Odata ce conexiunea dintre client si server a fost stabilita, serverul va prelua comenzile de la client, le va procesa si ii va intoarce rezultatele bazate pe comenzile efectuate.

Datele folosite in aplicatie au fost stocate intr-o baza de date de tipul SQLITE. Am ales acest sistem de gestiune a bazelor de date deoarece are o compatibilitate foarte buna cu limbajul C, are o viteza buna de procesare

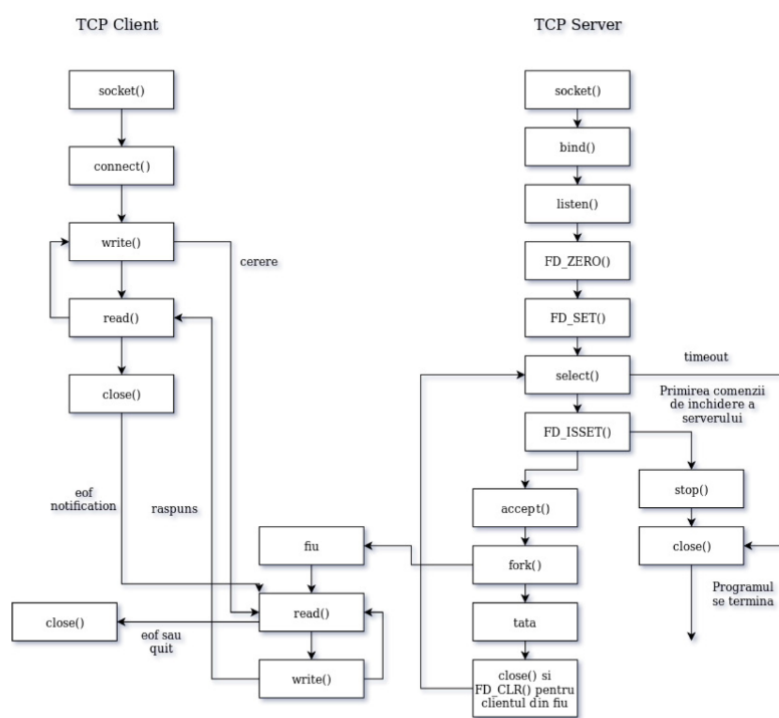


Figure 1: Diagrama aplicatiei

a datelor si pentru ca a fost usor de configurat. In aceasta baza de date se regasesc urmatoarele tabele:

- songs- care contine numele tuturor melodiilor
- users- care contine toti utilizatorii inregistrati in baza de date
- genres- care contine toate genurile melodiilor din tabela songs
- comments- care contine toate comentariile adaugate melodiilor
- votes- care contine punctajul acumulat de fiecare melodie in urma voturilor acordate

4 Detalii de implementare

Aplicatia va avea nevoie de un mod de a recunoaste clientii, asa ca vom face doua functii, precum *login* si *register*, functii care ii vor permite clientului fie sa se logheze fie sa se inregistreze, asta daca nu are deja un cont creat.

Clientul nu va putea executa orice alta comanda daca nu este logat pe contul sau.

Ulterior am implementat urmatoarele functii si functionalitati:

- addSong()- functie care va permite adaugarea unei noi melodii in baza de date. Utilizatorului i se vor cere mai multe detalii despre melodie precum, numele melodiei, o descriere, un link catre Youtube al melodiei respective, numarul de genuri carora le apartine melodia respectiv genurile.
- voteSong()- functie care va permite votarea unei anumite melodii
- generalTop()- vom afisa topul actual al melodiilor in functie de numarul de voturi
- genreTop()- afisarea topului pentru un anumit gen de muzica(ex: dance, rock, hip-hop, etc...)

```

void addSongToTop(int clientSocket, MYSQL *con)
{
    char *successMessage = "success";

    char *name = readData(clientSocket);
    char *description = readData(clientSocket);
    char *link = readData(clientSocket);
    int numberOfGenres = atoi(readData(clientSocket));

    int insertSongQueryLength = strlen("insert into songs (name, description, link) values ();") + strlen(name) + strlen(description) + strlen(link) + 10; // 10 pentru virgule si spatii
    char *insertSongQuery = malloc(sizeof(char) * insertSongQueryLength);

    strcpy(insertSongQuery, "insert into songs (name, description, link) values ('");
    strcat(insertSongQuery, name);
    strcat(insertSongQuery, "\", '");
    strcat(insertSongQuery, description);
    strcat(insertSongQuery, "\", '");
    strcat(insertSongQuery, link);
    strcat(insertSongQuery, "');");

    printf("Query: %s\n", insertSongQuery);

    if (mysql_query(con, insertSongQuery))
    {
        printf("Can't run the query! Line: %d\n", __LINE__);
        sendData(clientSocket, "Can't run the query!");
        free(insertSongQuery);
        return ;
    }

    if (numberOfGenres == 0)
    {
        printf("Invalid number of genres! Line: %d\n", __LINE__);
        sendData(clientSocket, "Invalid number of genres!");
        return;
    }
}

```

Figure 2: Functia addSong

```

for (int i = 0; i < numberOfGenres; i++)
{
    char *genre = readData(clientSocket);
    char *baseQuery = "insert into genres (songName, genre) values ('", "');";
    int addGenresQueryLength = strlen(baseQuery) + strlen(name) + strlen(genre);

    char* insertGenreQuery = malloc(sizeof(char) * addGenresQueryLength);
    strcpy(insertGenreQuery, "insert into genres (songName, genre) values ('");
    strcat(insertGenreQuery, name);
    strcat(insertGenreQuery, "\", '");
    strcat(insertGenreQuery, genre);
    strcat(insertGenreQuery, "');");

    printf("Query: %s\n", insertGenreQuery);

    if (mysql_query(con, insertGenreQuery))
    {
        printf("Can't run the query! Line: %d\n", __LINE__);
        sendData(clientSocket, "Can't run the query!");
        free(insertSongQuery);
        return;
    }
}

sendData(clientSocket, successMessage);

```

Figure 3: Functia addSong

- `comment()`- adaugarea unui comentariu unei anumite melodii
- `restrictVote()`- ca admin avem posibilitatea de restrictiona dreptul de a mai vota unui anumit user
- `deleteSong()`- ca admin avem dreptul de a sterge orice melodie inregistrata in baza de date
- `getComments()`- afisarea tuturor comentariilor melodiilor, inclusiv userul care a pus acel comentariu si melodia la care a fost pus

5 Concluzii

Ca si concluzie as spune ca nu mi se pare un proiect greu de implementat, ba chiar relativ interesant, avand functionalitati de uz practic, ”contemporane”. Functiile sunt foarte asemanatoare ca structura, ceea ce diferea de la o functie la alta, sunt datele. Am avut diferite probleme la implementare, mai ales la partea de transmisie a datelor pe retea.

Ca functionalitati suplimentare, care nu erau specificate in descrierea cerintei, am adaugat functia de afisare a tuturor comentariilor. Timpul nu mi-a permis adaugarea mai multor functionalitati, asa cum precizasem in prima documentatie.

6 Bibliografie

-<https://profs.info.uaic.ro/computernetworks/cursullaboratorul.php>
 -<https://www.geeksforgeeks.org/basics-computer-networking>
 -<https://www.w3schools.com/sql/>