Rețele de calculatoare - TEMA 2

Bocăneț Raluca-Andreea

13.12.2023

Championship (B)

Sa se realizeze o aplicatie client/server ce va administra diferite campionate. Aplicatia server va avea urmatoarele functionalitati: inregistrarea utilizatorilor(diferite tipuri de utilizatori: obisnuiti, administratori). Toate comenzile vor fi restrictionate de sectiunea de logare. Dupa logare utilizatorii vor putea primi informatii despre ultimele campionate inregistrate, avand posibilitatea de a se inscrie la ele. Comenzile ce trebuie implementate: inregistrarea unui campionat, specificarea jocului, numarului de jucatori, diferite reguli sau structuri de campionat(single-elimination, double elimination), modul de extragere a partidelor(deciderea partidelor), inregistrarea unui utiliziator intr-un campionat, utilizatorul va fi informat via e-mail daca a fost acceptat in campionatul respectiv, si va primi informatii aditionale despre partidele sale (ora, adversarul, etc...) Utilizatorul are posibilitatea sa reprogrameze o sesiune de joc. Utilizatorii administratori vor detine un istoric al scorurilor partidelor.

1 Introducere

Proiectul *Championship* se bazează pe comunicare de tip client/server prin intermediul căreia putem administra diferite campionate. Utilizatorii vor avea la dispoziție două metode de conectare (login și înregistrare), urmând a fi repartizați după anumite clirerii la partidere de joc. De asemenea, administratorii vor avea și ei aceleași modalități de conectare, însă aceștea vor fi cei ce supraveghează jocurile, adică vor avea acces la tabelul cu punctaje a utilizatorilor.

2 Tehnologii aplicate

- Proiectul va fi implementat cu ajutorul protocolului de comunicare TCP, un protocol orientat conexiune. Am făcut această alegere deoarece TCP asigură flexibilitate în livrarea datelor (garantează trimitera datelor fără erori și în ordinea corectă), primesc la fiecare pas confirmare cum că informația trimisă a fost primită în siguranță și oferă o conexiune orientată (comunicare bidirecțională și continuă).
- Pentru a stoca informațiile referitoare la utilizatori și administratori voi folosi un fisier XML deoarece am acces mult mai rapid la informațiile legate de persoanele logate sau care vor să se conecteze spre deosebire de

utilizarea fișierelor care sunt mai greu de gestionat atunci când informația este vastă.

- Codul acestui proiect va fi scris cu ajutorul limbajelor de programre C/C++.
- Codul pentru trimiterea email-urile va folosi SMTP.

3 Structura aplicatiei

Proiectul *Championship* este format din client și server. Mai mulți utilizatori sau administratori se pot conecta simultan pe server, fiecare fiind pus să furnizeze anumite informații pentru a avea acces la următoarele comenzi. Pentru a înțelege mult mai bine diferența între utilizator și administrator, am această listă de comenzi și descrierea lor pentru fiecare:

Pentru utilizatori:

- **Înregistrarea**
  Fiecare utilizator se va putea înregistra cu ajutorul unui username și a unei adrese de email. Fără aceste informații înregistrarea se va considera invalidă. Comanda pentru inregistrare este : login utilizator : username , adresaEmail
- **Înscriere**
  În cazul în care un utilizator este înregistrat în baza de date, acesta se va putea avea acces la campionate introducând username-ul. Comanda este : autentificare utilizator : username ;
- **Înscrierea într-un campionat**
  După ce utilizatorul este înregistrat cu succes sau este logat acesta va putea să se înscrie la campionate, înscrierea acestuia într-un campionat reprezentând oferirea unui punctaj și a unui email cu detaliile competițiilor. Comanda este : inscriere campionat : numeCampionat ;
- **Posibilitatea de a reprograma un campionat daca vrea**
  Dacă un utilizator nu poate sa participe la un campionat, poate utiliza camnda reprogramare campionat : numeCampionat , data , ora pentru a reprograma campionatul;
- **Renuntare la un campionat**
  Daca un utilizator nu mai doreste sa participe la un campionat, acesta poate scrie comnada renuntare campionat : numeCampionat si datele legate de inscrierea sa la campionat, cat si punctele ii vor si anulate/sterse.
- **Vizualizare profil**
  Aceasta comanda poate sa iti ofere informatii despre profilul tau in orice moment si pentru a le vedea poti scrie comanda "vizualizare profil".

Pentru administratori:

- **Înregistrarea**
  Fiecare administrator se va putea înregistra cu ajutorul unui username. Comanda pentru inregistrare este : login utilizator : username ;
- **Înscriere**
  Dacă un administrator se află în baza de date acesta se poate loga cu ajutorul username-ului. Comanda este : autentificare utilizator : username;
- **Înregistrarea unui campionat**
  Un administrator poate crea un campionat, dacă dorește, respectând cerințele înregistrării unui campionat. Acesta trebuie să specifice reguli, numărul de partide și punctajul oferit. Comanda pentru inregistrarea unui campionat este : inregistrare campionat : TipCampionat , nrPartide , punctajul-FiecareiPartide (ex 11-12) , nrMembriiPartida (ex 2—3) ;
- **Posibilitatea de a vizualiza scorului utilizatorilor**
  Un administrator poate vizulializa în orice moment istoricul scorurilor. Comanda este : vizualizare scor;

Functionalități suplimentare:

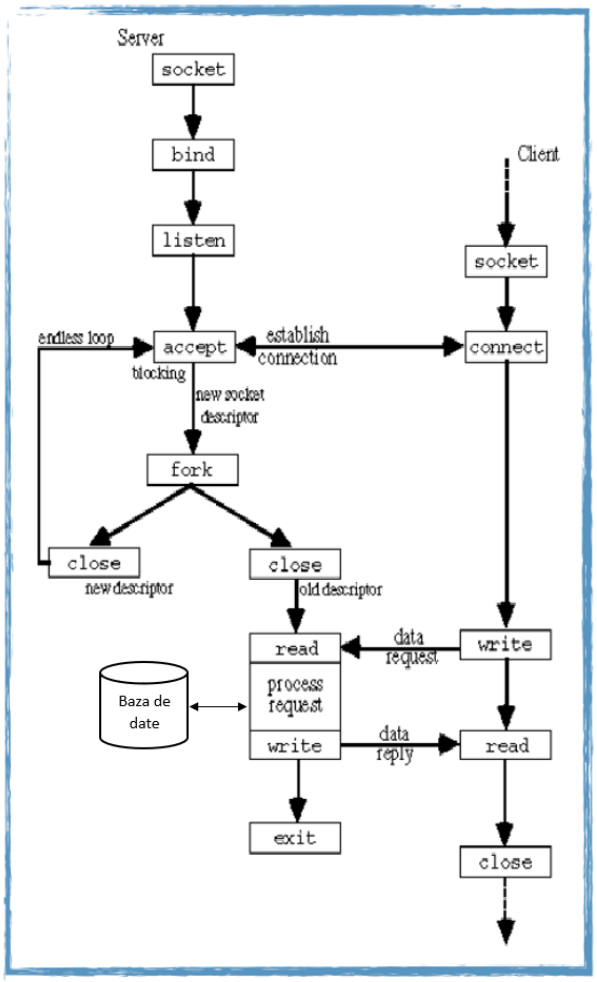
- **Trimiterea de email-uri**
  Trimite utilizatorilor e-mailuri despre partidele la care tocmai s-au înscris.
- **Extragerea automată a partidelor**
  Implementarea unei funcționalități care extrage automat partidele în funcție locurile disponibile.
- **Logout**
- **Quit**

- În cadrul *process request* se vor implementa functionalitățile menționate mai sus. Schema "Figure 1: legăturile cu campionatele" descrie în mare cum va arăta fisierul XML și legătura între ele.

4 Aspecte de Implementare

Crearea unui socket:

```
server_socket = socket(AF_INET, SOCK_STREAM, 0);
if (server_socket == -1)
{
    perror("Eroare la crearea socketului");
    exit(EXIT_FAILURE);
}
```



Reutilizarea adresei IP si a portului după închiderea socket-ului:

```
int opt = 1;
if (setsockopt(server_socket, SOL_SOCKET, SO_REUSEADDR, &opt, sizeof(opt)) < 0)
{
    perror("Eroare la setarea optiunii SO_REUSEADDR");
    exit(EXIT_FAILURE);
}
```

- După cum se observă în imaginea atașată mai sus comunicarea între client și server se realizează pe bază unui socket. Utilizarea sa este mult mai avantajoasă datorită bidirecționalității spre deosebire de canalele de comunicare învățate (fifo,pipe).

- Pentru a exemplifica mai bine legătura dintre comenzi, atât pentru fiecare dintre utilizatori, cât și pentru administratori avem următoarea schemă (Figure 2 :utilizatori și administratori).
- Stocarea informațiilor se va face cu ajutorul bazei de date XML deoarece este o alegere mult mai rapidă și mai ușor de vizualizat spre deosebire de fișiere text, unde parcurgerea informațiilor se face linie cu linie, caracter cu caracter. Baza de date XML are o structură arborescentă după cum se poate vedea în codul de mai jos. Aici se regăsește structura fisierului XML mentionata in imaginea "Figure 2 : utilizatori si campionate".

```
<database>
        <utilizatori>
                <utilizator>
                        <username>admin1</username>
                        <email>bocantralucaandreea@gmail.com</email>
                        <punctaj>115</punctaj>
                        <status>nu_sunt_logat</status>
                </utilizator>
                <utilizator>
                        <username>admin2</username>
                        <email>bocantralucaandreea@gmail.com</email>
                        <punctaj>50</punctaj>
                        <status>nu_sunt_logat</status>
                </utilizator>
                <utilizator>
                        <username>ion</username>
                        <email>ion@gmail.com</email>
                        <punctaj>20</punctaj>
                        <status>nu_sunt_logat</status>
                </utilizator>
                <utilizator>
                        <username>mara</username>
                        <email>ralucaandreea313@gmail.com</email>
                        <status>nu_sunt_logat</status>
                </utilizator>
        </utilizatori>
        <administratori>
                <administrator>
                        <username>ad1</username>
                        <status>nu_sunt_logat</status>
                </administrator>
                <administrator>
                        <username>Raluca</username>
                        <status>nu_sunt_logat</status>
                </administrator>
```

```
        <administrator>
                <username>marius</username>
                <status>nu_sunt_logat</status>
        </administrator>
</administratori>
<championships>
        <championship_single_elimination>
                <informatii>
                        <ora>12:00</ora>
                        <data>07.04.2024</data>
                </informatii>
                <matches>
                        <match>
                                <name>Partida1</name>
                                <score>10</score>
                                <members>2</members>
                        </match>
                        <match>
                                <name>Partida2</name>
                                <score>5</score>
                                <members>2</members>
                        </match>
                </matches>
        </championship_single_elimination>
        <championship_double_elimination>
                <informatii>
                        <ora>20:00</ora>
                        <data>04.03.2024</data>
                </informatii>
                <matches>
                        <match>
                                <name>Partida1</name>
                                <score>10</score>
                                <members>2</members>
                        </match>
                        <match>
                                <name>Partida2</name>
                                <score>20</score>
                                <members>2</members>
                        </match>
                </matches>
        </championship_double_elimination>
        <championship_single_elimination_1>
                <informatii>
                        <ora>19:00</ora>
                        <data>12.04.2024</data>
```

```
                </informatii>
                <matches>
                        <match>
                                <name>Partida1</name>
                                <score>5</score>
                                <members>2</members>
                        </match>
                        <match>
                                <name>Partida2</name>
                                <score>6</score>
                                <members>2</members>
                        </match>
                </matches>
        </championship_single_elimination_1>
        <championship_double_elimination_1>
                <informatii>
                        <ora>19:00</ora>
                        <data>23.02.2024</data>
                </informatii>
                <matches>
                        <match>
                                <name>Partida1</name>
                                <score>9</score>
                                <members>2</members>
                        </match>
                        <match>
                                <name>Partida2</name>
                                <score>10</score>
                                <members>3</members>
                        </match>
                </matches>
        </championship_double_elimination_1>
    </championships>
</database>
```

- În cerință proiectului este menționată trimiterea unor email-uri între participanții unei partide. Acest lucru se va realiza cu protocolul SMTP (Simple Mail Transfer Protocol) deoarece oferă securitate, gestionează erorile, este flexibil (mesajul poate conține fișiere diverse) și fiabil (dacă nu poate să trimită un mesaj imediat va încerca mai târziu, întorcând o eroare dacă nu reușește).

```
struct upload_status
{
    int lines_read;
    const char *data;
```

```
    size_t data_size;
};

static size_t payload_source(void *ptr, size_t size,
size_t nmemb, void *userp)
{
    FILE *file = (FILE*)userp;
    size_t retcode = fread(ptr, size, nmemb, file);
    return retcode;
}

void trimitere_email(char email[])
{
    CURL *curl;
    CURLcode res = CURLE_OK;
    FILE *file = fopen("test_email.txt", "rb");

    if (!file) {
        fprintf(stderr, "Eroare la deschidea fisierului!\n");
        return;
    }

    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL,
        "smtps://smtp.gmail.com:465");
        curl_easy_setopt(curl, CURLOPT_USE_SSL, CURLUSESSL_ALL);
        curl_easy_setopt(curl, CURLOPT_USERNAME,
        "ralucaandreea313@gmail.com");
        curl_easy_setopt(curl, CURLOPT_PASSWORD,
        "cpycedfdnjxauybt");
        curl_easy_setopt(curl, CURLOPT_MAIL_FROM,
        "ralucaandreea313@gmail.com");

        struct curl_slist *recipients = NULL;
        recipients = curl_slist_append(recipients, email);
        curl_easy_setopt(curl, CURLOPT_MAIL_RCPT, recipients);

        curl_easy_setopt(curl, CURLOPT_READFUNCTION, payload_source);
        curl_easy_setopt(curl, CURLOPT_READDATA, file);
        curl_easy_setopt(curl, CURLOPT_UPLOAD, 1L);

        res = curl_easy_perform(curl);
        curl_slist_free_all(recipients);
        curl_easy_cleanup(curl);
    }
```

```
    fclose(file);
    if (res != CURLE_OK) {
        printf("Mesaj esuat!\n");
    } else {
        printf("Mesaj trimis!\n");
    }
}
```

5 Concluzii

O primă îmbunătățire a aplicației ar o interfață grafică pentru a oferi celor ce utilizează aplicația o experiență plăcută. O altă îmbunătățire ar fi alegea partidelor în funcție de mai multe criterii decât punjatul fiecărui utilizator (de exemplu: după dorințele utilizatorilor sau în funcție de rezultate excepționale la alte campionate ce nu sunt înregistrate sau oferirea de sugestii utilizatorilor după evaluarea istoricului lor). De asemenea, o îmbunătățire ar consta în înlocuirea procesului fork cu thread-uri pentru o comunicare mai ușoară (acestea pot comunica folosindu-se de aceeași zonă de memorie) și pentru o sincronizare eficientă.

6 Referințe Bibliografice

- Nivelul de transport - PDF
- Cod de implentare cu modificări a unui TCP concurent
- Informații despre email-uri
- Protocoale pentru email-uri
- Tutorial pentru XML

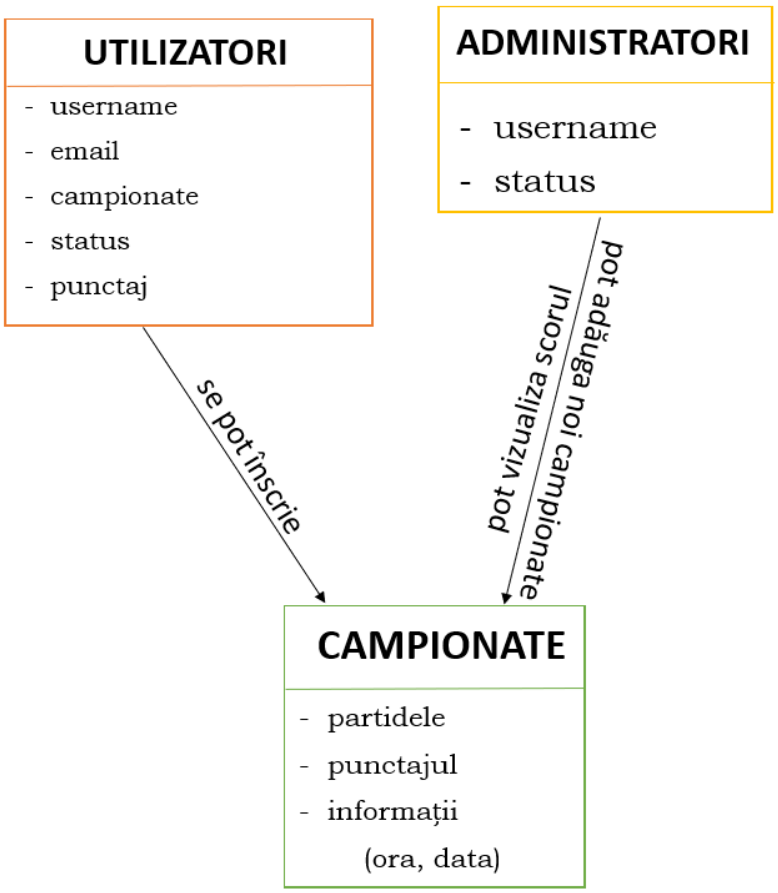


Figure 1: legăturile cu campionatele

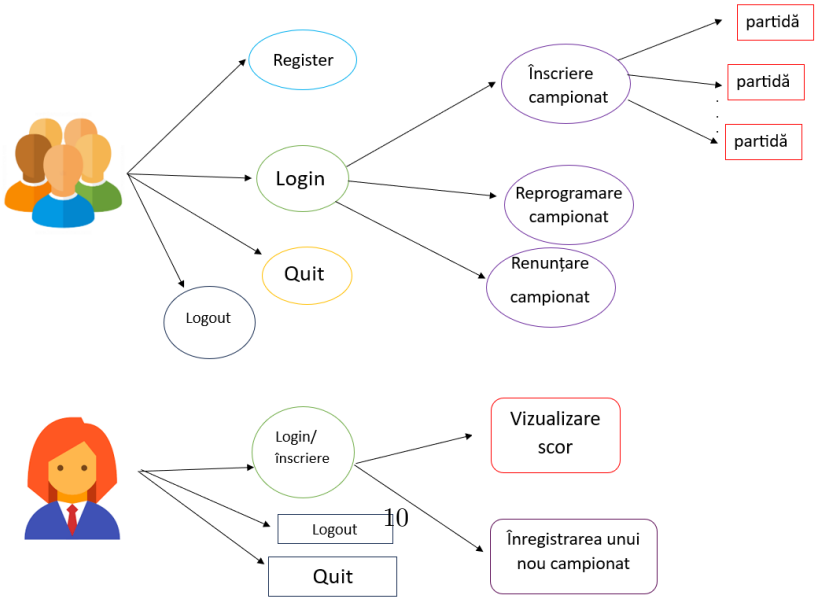


Figure 2: utilizatori și administratori