QuizzGame [Propunere Continental]
Documentatie proiect Retele de calculatoare

Popescu Dimitrie, An 2: B7
Facultatea de Informatica Iasi
dimitrie.popescu@info.uaic.ro

1 Introducere

Există diferite tipuri de jocuri ce prezintă concursuri de cultură generală, ce se bazează pe întrebări si răspunsuri. Aceste tipuri de jocuri sunt întalnite sub numele de “Quizz Games” sau “Trivia Games”. Ideea de bază a acestor jocuri este: conținutul unui set de întrebări care se trimit aleatoriu la un anumit interval de timp iar apoi se anunță câștigatorul cu cele mai multe întrebări răspunse corect.

Proiectul **QuizzGame** presupune implementarea unei aplicații formată dintr-o componentă server si o componentă client. Componenta server va conține un server multithreading care va suporta un număr nelimitat de clienți, ei find puși să răspunda în ordinea în care s-au înregistrat.
Componenta client va conține un client care primește întrebarea si trebuie sa răs pundă într-un număr de secunde, iar serverul va verifica răspunsul clientului.

2 Tehnologii utilizate

2.1 Sisteme de operare compatibile si limbaje de programare utilizate

Aplicatia va rula pe sistemele de operare Linux (opțional, Ubunu), si va fi realizată utilizând cod scris în C/C++.

2.2 Comunicarea in rețea

Referitor la comunicarea în retea, proiectul va utiliza la nivel de **Aplicatie ,** un port ce va permite conectarea clienților la server-ul multithreading. Conexiunea dintre server si client este realizata prin protocolul **TCP**.

Alegerea utilizării **TCP** a fost una subiectivă, deoarece cred că se potrivește cel mai bine acestui proiect. Totusi, are si el problemele lui. El garantează precizia livrarii mesajelor, dar nu este foarte rapid in comparație cu **UDP**.

Spre deosebire de **TCP**, **UDP** nu este "**orientat-conexiune**", el nu oferă nici-o garanție ca mesajele vor ajunge la destinație, dar el este foarte rapid în schimb. Dacă aș fi ales un proiect de tip chat online, sau poate o aplicație DNS, atunci aș fi ales să folosesc **UDP** deoarece acolo chiar contează rapiditatea.

Am ales să folosesc TCP pentru proiectul meu, deoarece rapiditatea nu contează ci siguranta că datele vor fi livrate corespunzator. Atâta timp cât timpul de raspundere la întrebari si timpul de conectare al clientilor este corect nu vom întâmpina niciodată probleme. În schimb, cu UDP atunci ar fi existat șanse mari să nu ajungă întrebari la unii clienti, jocul nostru să nu mai ruleze cum ne doream defapt.

Comunicarea între server si client se face prin intermediul socket-urilor, ce realizează conexiunea dintre client si server. Atunci când se creează socket-urile, se realizează o conexiune orientate, ce garantează trimiterea pachetelor de date in ordine, iar dacă acestea sunt eventual pierdute să fie recuperate.

3 Arhitectura aplicatiei

Proiectul 3 părți importante :

Serverul este unul multithreading, folosind threaduri cu mutex. Am ales folosirea threadurilor cu mutex pentru a întâmpina problema la crearea thread-urilor, de exemplu problema majora a thread-urilor este posibilitatea de a folosi aceeasi memorie, folosind aceeași structură de către 2 threaduri diferite.

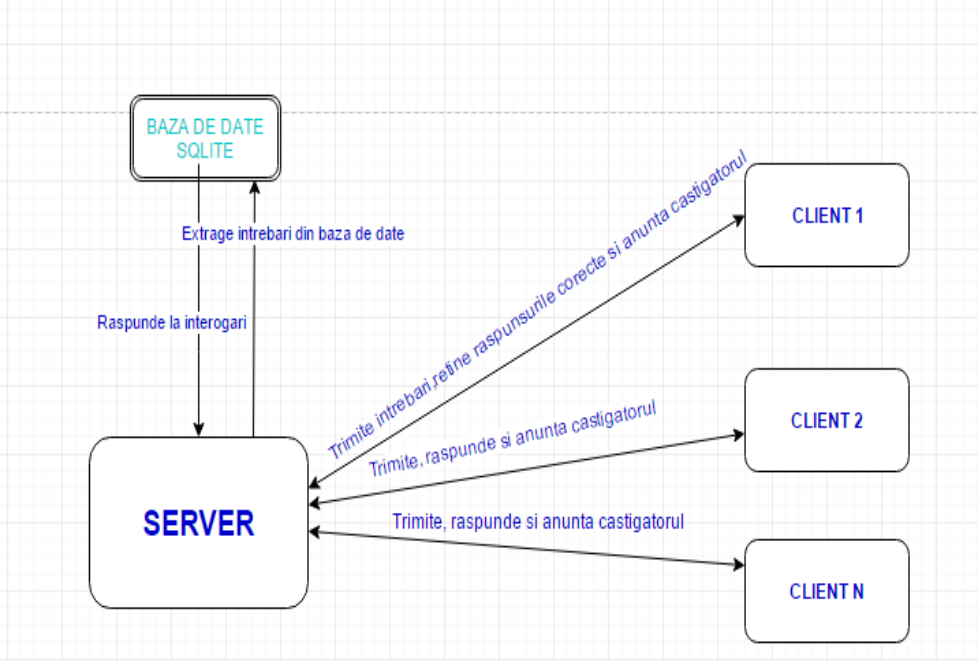
Mutex este un mecanism bazat pe pre-threaduri si rezolvă acească posibila problemă ca 2 threaduri sa foloseasca aceeasi structură.

El are rolul de a oferi :

- "**atomicity**", blocarea unui mutex este o operatie atomică, ce ineseamna că sistemul de operare asigură faptul că daca ai blocat un mutex, atunci niciun alt thread nu poate fi folosit în același timp
- "**singularity**", dacă un thread a reușit să blocheze un mutex, niciun alt thread nu va putea debloca pana cand threadul inițial va face acest lucru
- "**non-busy wait**", dacă un thread reușește sa blocheze un thread care a fost blocat de threadul secundar atunci primul thread va fi suspendat (si nu va consuma nici-o resursa CPU) până când lacătul este luat de catre secondul secundar. În acel moment, primul thread se va trezi si va continua executia, având mutex-ul blocat.

Severul joacă un rol important și se afla în legătură cu baza de date **SQLite**. Baza de date are rolul de menține întrebările, variantele de raspuns, scorul in tabelele respective. Client-ul este unul obișnuit, la care se adaugă procesul de înregistrare, atunci când el se conectează la server, va trebui să se înregistreze si va fi alocat

thread-ului respectiv. În cazul in care, un client pierde conexiunea la joc sau o închide intenționat sau din greșeală atunci el va fi eliminat de la joc.



4 Detalii de implementare

Pentru a facilita implementarea efectivă a jocului, am implementat un protocol ce are rolul de a coordona comunicarea dintre server si client.

Acesta are rolul de a verifica mai multe comenzi necesare proiectului cum ar fi : înregistrarea clienților, trimiterea întrebărilor către clienți, primirea răspunsurilor și verificarea dacă un client părăsește jocul.

```
#DEFINE REGISTER_USER_COMMAND "RegisterUser:"
#DEFINE FAILED_USER_REGISTER  "Failed:RegisterUser:"
#DEFINE SUCCESS_USER_REGISTER "Sucess:RegisterUser:"
#DEFINE GET_QUESTION_COMMAND  "GetQuestion:"
#DEFINE ANSWER_OPTION_COMMAND "AnswerOption:"
#DEFINE ANSWER_COMMAND        "Answer:"
```

Întrebările vor fi menținute intr-o bază de date SQLite. Această bază de date va fi formată din 2 tabele, una numită Answers si una numită Questions. Tabela Answers va conține varianta corectă de răspuns si scorul respective întrebării, iar tabela Questions va mentine întrebarea, variantele de răspuns, și timpul permis pentru a răspunde

la întrebare. De asemenea, se va verifica mereu dacă baza de date a fost deschisa corect pentru a parcurgere corect jocul. Dacă se va întâmpla ca una din tabele sa fie gresită, atunci nu se va merite crearea ei.

```
int rc;

   rc = sqlite3_open("QuizzGame.db", &db);

   if( rc ){
      fprintf(stderr, "Can't open database: %s\n",
sqlite3_errmsg(db));
      return(0);
   }else{
      fprintf(stderr, "Opened database successfully\n");
```

```
rc = sqlite3_exec(db, sql, NULL, 0, &zErrMsg);
      if (rc != SQLITE_OK) {
        fprintf(stderr, "SQL error: %s\n", zErrMsg);
        sqlite3_free(zErrMsg);
      } else {
        fprintf(stdout, "Table created successfully\n");
```

Serverul aplicației este unul multithreading, iar concurența este asigurată prin folosirea de pre-threaduri. Atunci cand un client se înregistrează, folosim un mutex prin a permite accesul lui la joc.

Structura thread-urilor este următoarea :

```
#include <pthread.h>

typedef struct {
    pthread_t id; // id-ul thread-ului
    int thCount;  // nr de conexiuni servite
} Thread;
```

```
unregisterUser(index);
pthread_mutex_lock(&dlock);
```

Inserarea întrebărilor si a timpului de raspuns în tabela Questions, se va face în următorul mod :

```
"CREATE TABLE Questions(id INT, text TEXT, timeToAnswer
INT);"
                "INSERT INTO Questions VALUES(1, 'In ce
an a murit Albert Einstein?', 12);"
                "INSERT INTO Questions VALUES(2, 'In ce
an s-a nascut Albert Einstein?', 12);"
                "INSERT INTO Questions VALUES(3, 'Ce drog
se gaseste in ceai si cafea?', 12);"
                "INSERT INTO Questions VALUES(4, 'Care
este elementul de baza in compozitia stelelor?', 12);"
                "INSERT INTO Questions VALUES(5, 'Ce
sport se practica in NBA?', 10);"
                "INSERT INTO Questions VALUES(6, 'Care
este capitala Japoniei?', 8);"
                "INSERT INTO Questions VALUES(7, 'Care
este capitala Portugaliei?', 8);"
                "INSERT INTO Questions VALUES(8, 'Care
este capitala Suediei?', 8);"
                "INSERT INTO Questions VALUES(9, 'In ce
tara a fost inventat bumerangul?', 10);"
                "INSERT INTO Questions VALUES(10, 'Care
este capitala Belgiei?', 8);"
                "INSERT INTO Questions VALUES(11, 'Care
este simbolul ceriului, element chimic?', 12);"
                "INSERT INTO Questions VALUES(12, 'Care
dintre urmatoarele nu este vecin al Ungariei?', 10);"
                "INSERT INTO Questions VALUES(13, 'Care
este a cincea litera din alfabetul latin?', 6);"
                "INSERT INTO Questions VALUES(14, 'Care
este bautura marinarilor?', 10);"
               "INSERT INTO Questions VALUES(15, 'Care este
capitala Bulgariei?', 8);"
               "INSERT INTO Questions VALUES(16, 'Care este
numarul de sateliti ai planetei Venus?', 12);"
```

Inserarea întrebărilor și scorul primit pentru răspuns corect, se va face astfel :

```
"CREATE TABLE Answers(id INT, question INT, text TEXT,
score INT, FOREIGN KEY(question) REFERENCES Ques-
tions(id));"
                "INSERT INTO Answers VALUES(101, 1,
'1955', 5);"
                "INSERT INTO Answers VALUES(102, 1,
'1962', 0);"
                "INSERT INTO Answers VALUES(103, 1,
'1950', 0);"
                "INSERT INTO Answers VALUES(104, 1,
'1870', 0);"
                "INSERT INTO Answers VALUES(201, 2,
'1900', 0);"
                "INSERT INTO Answers VALUES(202, 2,
'1879', 5);"
                "INSERT INTO Answers VALUES(203, 2,
'2000', 0);"
                "INSERT INTO Answers VALUES(204, 2,
'1850', 0);"
                "INSERT INTO Answers VALUES(301, 3, 'her-
oina', 0);"
                "INSERT INTO Answers VALUES(302, 3, 'co-
caina', 5);"
                "INSERT INTO Answers VALUES(303, 3,
'zahar', 0);"
                "INSERT INTO Answers VALUES(304, 3,
'nimic', 0);"
                "INSERT INTO Answers VALUES(401, 4,
'Oxigen', 0);"
                "INSERT INTO Answers VALUES(402, 4, 'Acid
sulfuric', 0);"
                "INSERT INTO Answers VALUES(403, 4,
'Apa', 0);"
                "INSERT INTO Answers VALUES(404, 4, 'Hi-
drogen', 5);"
                "INSERT INTO Answers VALUES(405, 4, 'Ar-
senic', 0);"
                "INSERT INTO Answers VALUES(501, 5, 'Bil-
iard', 0);"
                "INSERT INTO Answers VALUES(502, 5, 'Bi-
atlon', 0);"
                "INSERT INTO Answers VALUES(503, 5, 'Bas-
chet', 5);"
                "INSERT INTO Answers VALUES(504, 5,
'Baseball', 0);"
```

```
                "INSERT INTO Answers VALUES(601, 6,
'Yokohama', 0);"
                "INSERT INTO Answers VALUES(602, 6, 'Yam-
ashina', 0);"
                "INSERT INTO Answers VALUES(603, 6, 'To-
kyo', 5);"
                "INSERT INTO Answers VALUES(604, 6, 'Na-
goya', 0);"
```

5 Concluzii

Proiectul ar putea beneficia de câteva inbunătătiri, atât pe partea de client, cât si pe cea de server. Pe partea de server, s-ar putea înbunătăți baza de date ce conține întrebările,clasificarea întrebarile pe categorii (Matematică,Logică,Programare,Istorie, Geografie,etc). Pe partea de client, s-ar putea adauga diverse opțiuni de a răspunde mai tarziu la o intrebare, de a ști în timp real câte întrebări au răspuns si ceilalți clienți.

6 Bibliografie

- http://profs.info.uaic.ro/~adria/teach/courses/net/cursullaboratorul.php
- http://stackoverflow.com/questions/5970383/difference-between-tcp-and-udp
- http://stackoverflow.com/questions/5201852/what-is-a-thread-really
- https://en.wikipedia.org/wiki/Transmission_Control_Protocol
- https://www.sqlite.org/cintro.html
- https://www.sqlite.org/quickstart.html

- http://stackoverflow.com/questions/34524/what-is-a-mutex
- https://www.sqlite.org/c3ref/exec.html