

QuizzGame

Martisca Filip

Decembrie 2020

1 Introducere

În acest raport vom prezenta caracteristicile și implementarea aplicației QuizzGame. QuizzGame este o aplicație în care clientul primește întrebări de la server având un anumit timp pentru a răspunde, ulterior clientul va fi evaluat în funcție de răspunsurile lui va primi o notă.

Întrebările vor fi trimise în ordinea înregistrării utilizatorilor iar când ultimul utilizator înregistrat va termina de răspuns la întrebări toți utilizatorii vor primi un mesaj cu utilizatorul care a avut scorul cel mai mare.

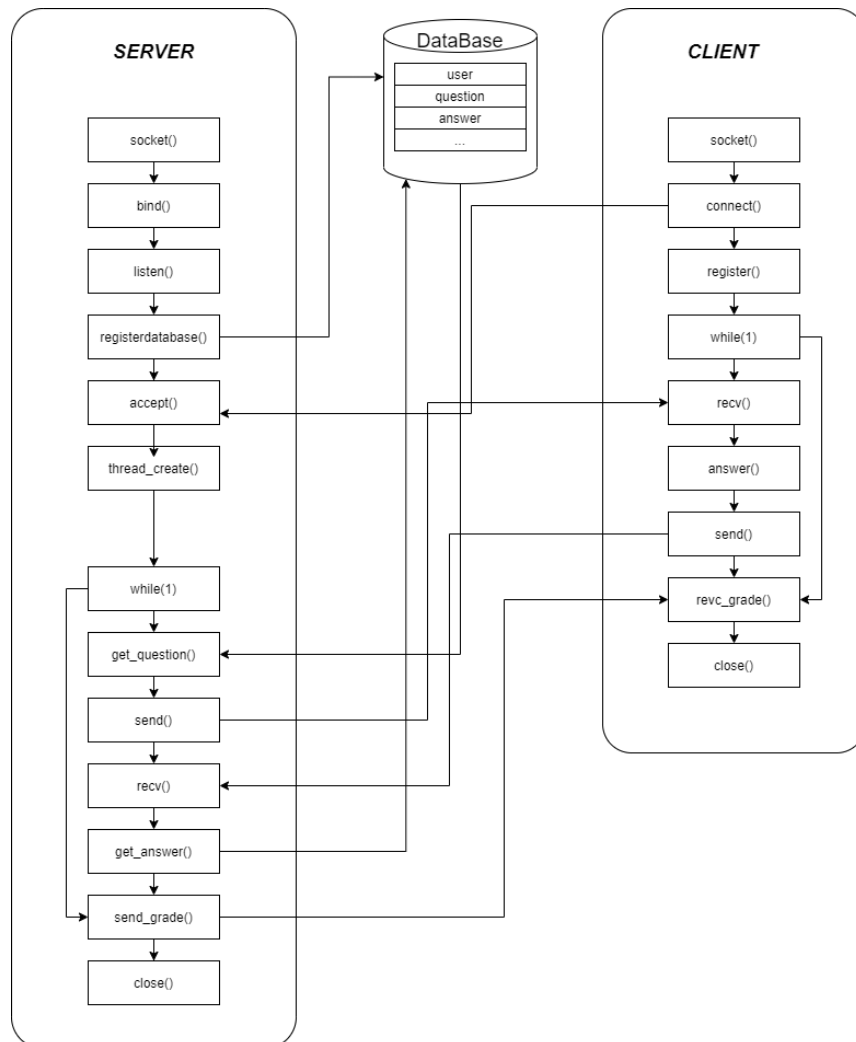
2 Tehnologii utilizate

În realizarea conexiunii dintre server și client am folosit modelul TCP. Am folosit modelul TCP care este orientat-conexiune deoarece avem siguranța că datele noastre vor ajunge intacte și în ordinea în care le-am trimis. Atunci când un client se va deconecta ceilalți participanți pot continua să răspundă la întrebări.

Serverul nostru va fi multithreading (va crea câte un thread pentru fiecare client) deci fiecare client va comunica independent cu serverul. Threadurile utilizează mai puține resurse decât procesele iar comunicarea între threaduri se poate face mai ușor.

3 Arhitectura aplicației

Toți utilizatorii, întrebările și răspunsurile vor fi stocate într-o bază de date SQLite în care vom avea mai multe tabele: user, question, answer ...



4 Detalii implementare

Atunci cand clientul realizeaza conexiunea cu serverul trebuie sa se inregistreze cu un username si cu un email, care apoi este trimis la server iar acesta le memoreaza intr-o baza de date impreuna cu un id unic.

In server se va selecta o intrebare cu variante de raspuns din baza de date care va fi trimisa clientului. Intr-un numar de secunde stabilit clientul trebuie sa raspunda la intrebare, in caz contrar va primi 0 puncte(dupa ce trece timpul apare automat urmatoarea intrebare). Se trimite raspunsul la server iar acesta ii calculeaza punctajul.

Cand se epuizeaza setul de intrebari clientii vor primi punctajul iar dupa ce au terminat toti clientii se va trimite tuturor un mesaj cu numele clientului care a castigat.

Clientul poate iesi oricand din joc trimitand "exit" ca raspuns la orice intrebare.

```
//primire raspuns/intrebare
    bzero(buf, sizeof(buf));
if (read (sd, &buf, sizeof(buf)) < 0)
{
    perror ("[client] Eroare la read() de la server.\n");
    return errno;
}
cout<<buf<<endl;
}

//trimitere raspuns
printf ("[client] Introduceti un raspuns: ");
fflush (stdout);
bzero(buf, sizeof(buf));
scanf("%s", buf);

if (write (sd, &buf, sizeof(buf)) <= 0)
{
    perror ("[client] Eroare la write() spre server.\n");
    return errno;
}

\\selectare id pentru crearea unui nou id
while((rc=sqlite3_step(stmt))==SQLITE_ROW)
{
    sent= sqlite3_column_int(stmt, 0);
}
sqlite3_finalize(stmt);
return sent;
```

5 Concluzii

Aceasta aplicatie poate avea ca scop evaluarea unor persoane (test) dar poate sa fie si un joc de cultura generala pe care il poti juca cu prietenii.

O varianta mai buna in materie de timp ar fi utilizarea modelului TCP cu `select()` si I/O neblocaante in loc de TCP multithreading.

O imbunatatire semnificativa a aplicatiei ar fi realizarea unei interfete deoarece va fi mult mai interactiva si mai placuta de client. O alta imbunatatire ar fi adaugarea unui timer pentru a vedea timpul ramas.

References

- [1] <https://profs.info.uaic.ro/computernetworks/cursullaboratorul.php>
- [2] https://en.wikipedia.org/wiki/Transmission_Control_Protocol
- [3] <https://www.backblaze.com/blog/whats-the-diff-programs-processes-and-threads/>
- [4] <https://stackoverflow.com/questions/3756882/detached-vs-joinable-posix-threads>
- [5] <https://www.geeksforgeeks.org/sql-using-c-c-and-sqlite/?ref=rp>
- [6] <https://stackoverflow.com/questions/263116/c-waiting-for-all-threads-to-complete>
- [7] https://www.youtube.com/watch?v=It0OFCbbTJEab_channel=JacobSorber
- [8] <https://computing.llnl.gov/tutorials/pthreads/>
- [9] <https://www.sqlite.org/index.html>
- [10] <https://www.geeksforgeeks.org/tcp-3-way-handshake-process/?ref=rp>