

Tema 2

Buliga Nicu

December 2020

1 Introducere

Am ales proiectul [Monitorizarea traficului](A), pentru care trebuie să implementez un sistem (server) capabil să gestioneze traficul și să ofere informații șoferilor conectați (clienții). Clienții la rândul lor vor putea să raporteze accidente, după care serverul va trebui să trimită în timp real celorlalți șoferi informații despre accidentul produs. Fiecare șofer va trebui să trimită viteza cu care circulă la fiecare 60 *secunde*, de asemenea un client poate alege opțiunea de a primi informații despre vreme, evenimente sportive și prețuri pentru combustibili la stațiile peço.

Am decis să aleg acest proiect pentru că este foarte practic și mă va ajuta să înțeleg mai bine modelul **client-server** și utilizarea mai multor fire de execuție pentru a face mai multe task-uri în paralel și nu iterativ. De asemenea și interacțiunea cu baza de date este un concept foarte important pentru acest proiect și pentru aplicații mult mai complexe în general.

2 Tehnologii utilizate

2.1 C++

Am ales să utilizez limbajul C++, deoarece dispune de mult mai multe librării decât C: librării grafice ușor de utilizat, standart template library (STL) pentru gestionarea mult mai simplă a datelor, etc. Limbajul C++ permite programarea orientată pe obiecte și dispune de o comunitate mult mai largă și diversificată.

2.2 Modelul client-server [TCP]

Proiectul meu se bazează pe o conexiune client-server cât mai bună, iar modelul TCP asigură aceasta. Trimiterea și primirea mesajelor este foarte importantă, deoarece în caz de accident fiecare șofer trebuie anunțat și trebuie să primească mesajul complet (denumirea străzii, ora, etc.) fără pierderi. TCP garantează acest lucru, chiar dacă nu este foarte rapid (din cauza realizării conexiunii).

2.3 MySQL

MySQL este un sistem de gestiune a bazelor de date relaționale. Am ales să-l utilizez în proiectul meu, fiindcă e foarte ușor de folosit (chiar și în C++) și nu necesită o atenție sporită. Parsarea este foarte simplă, pentru că interogările returnează un tabel cu toate atributele și cu tipul de date specific (nu sunt necesare tranformări). Singura problemă este că nu poate executa mai multe interogări în același timp.

2.4 Threads

Am ales să implementez structura proiectului cu thread-uri, deoarece sunt ușor de utilizat și partajarea de informații între mai multe fire de execuție este simplă (prin variabile globale), și nu necesită canale noi de comunicare ca în cazul proceselor create de `fork()`.

2.5 SFML

SFML este o librărie multimedia pentru dezvoltarea aplicațiilor 2D. Este implementată în C++, iar aceasta va aduce o viteză mai mare a aplicației proiectate de mine, care de asemenea este în C++. Este ușor de utilizat și dispune de multe resurse pe internet.

3 Arhitectura aplicației

În această aplicație, clientul va dispune de o interfață grafică cu care va putea interacționa. Fiecare interacțiune va duce la transmiterea unui protocol către server, care la rândul său îl va recepționa și va decide ce să facă mai departe. Serverul va actualiza baza de date și va răspunde la mesajele clientului.

Pentru fiecare client conectat, serverul va crea un nou fir de execuție care se va ocupa de recepționarea mesajelor de la clientul respectiv, astfel serverul fiind concurrent. În paralel serverul va avea un al fir de execuție care va citi permanent din baza de date actualizări și în caz de accident va trimite la toți clienții mesaj de avertizare.

În client vor exista 2 fire de execuție: unul va trimite mesaje către server, iar celălalt va aștepta actualizări de la server, în caz de accident sau blocaj în trafic.

Procesele date sunt descrise în diagrama de mai jos:

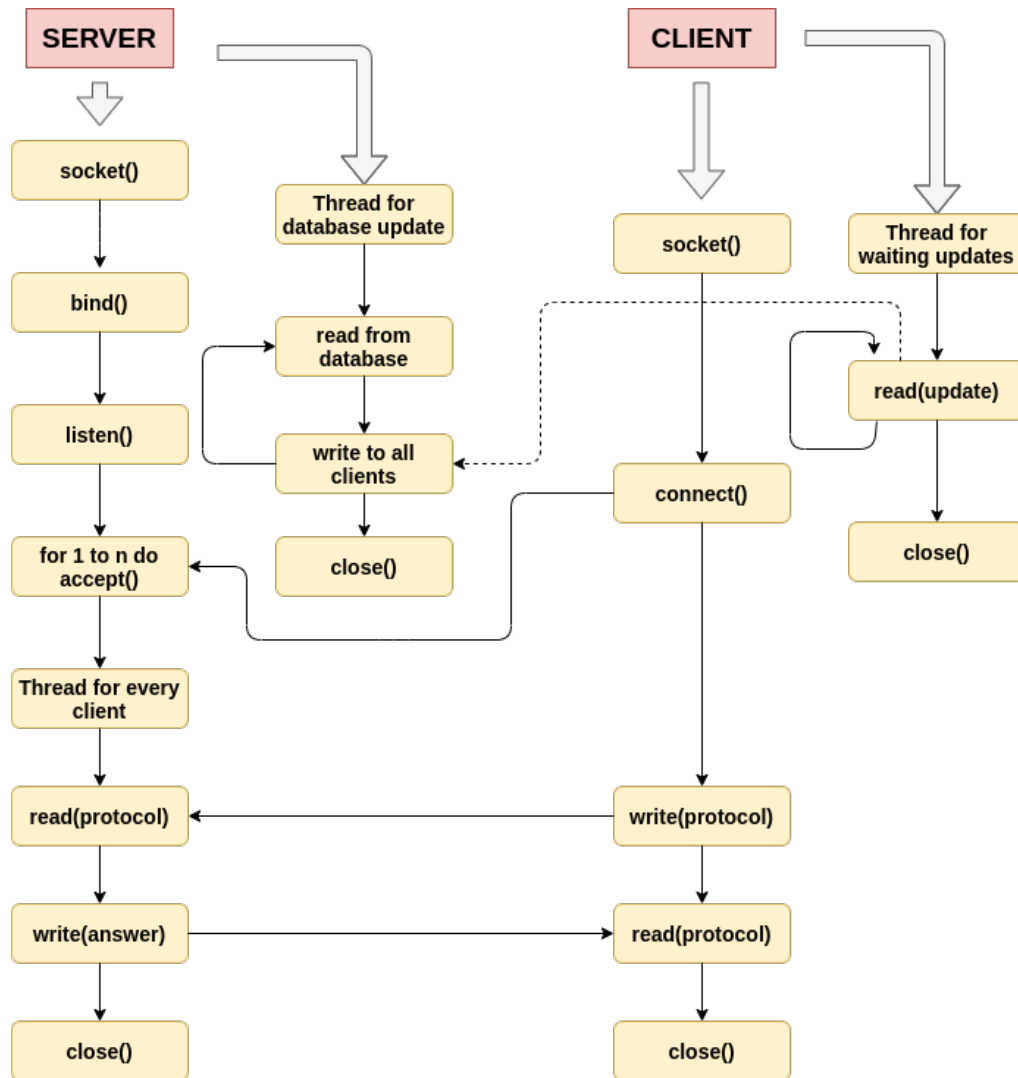


Figure 1: Diagrama aplicației detaliată

4 Detalii de implementare

Pentru utilizarea bazei de date este necesară conectarea, care se realizează în felul următor:

```

16 // Structuri de date pentru interactiunea cu MySQL db
17 MYSQL *conn;
18 MYSQL_RES *res;
19 MYSQL_ROW row;
20
21 /* Parametri de conectare la baza de date */
22 char *server = "localhost";
23 char *user = "Nicu";
24 char *password = "Nicu.123";
25 char *database = "users";

```

Figure 2: Variabilele necesare pentru conectare

```

41 void connectToDataBase()
42 {
43     conn = mysql_init(NULL);
44
45     /* Connect to database */
46     if (!mysql_real_connect(conn, server,
47         user, password, database, 0, NULL, 0)) {
48         fprintf(stderr, "%s\n", mysql_error(conn));
49         exit(1);
50     }
51 }

```

Figure 3: Conectarea la baza de date

Clientul și serverul vor putea comunica prin următoarele protocoale:

- *< login >* - Clientul se va putea loga în sistem cu date existente în baza de date.
- *< register >* - Clientul se va putea înregistra cu un login și parolă care nu există în baza de date.
- *< accident >* - Clientul va raporta un accident pe o anumită stradă.
- *< minPath >* - Clientul va cere cel mai scurt drum de la locația curentă la orice alt punct.
- *< newSpeed >* - Clientul va trimite viteza curentă la fiecare 60 secunde.
- *< join >* - Clientul se va putea înscrie să primească informații despre vreme, evenimente sportive, prețuri pentru combustibili la stațiile peco.
- *< logout >* - Clientul va ieși din aplicație.

5 Concluzie

În concluzie, proiectul dat utilizează foarte multe tehnologii, iar folosirea acestora împreună necesită mult timp de lucru și atenție. Îmbunătățiri ar putea fi aduse la partea de optimizare (codul să ruleze mai rapid și să nu folosească multe resurse) prin folosirea instrucțiunii `select()` care supraveghează inputul și nu e necesar ca execuția să stea blocată într-un apel `read()`. Ar putea fi introduse noi protocoale care să extindă posibilitățile unui șofer de a se folosi de aplicație: să vadă toți participanții la trafic, să existe opțiunea de a căuta restaurante, magazine, etc. prin apropiere și multe altele. În final, manipularea erorilor este foarte importantă și e o îmbunătățire considerabilă pentru acest proiect.

References

- [1] Model de client-server TCP cu thread-uri,
[https://profs.info.uaic.ro/ computernetworks/index.php](https://profs.info.uaic.ro/computernetworks/index.php)
- [2] Conectarea la baza de date,
<https://www.cyberciti.biz/tips/linux-unix-connect-mysql-c-api-program.html>
- [3] Utilizarea SFML,
<https://www.sfml-dev.org/tutorials/2.5/>
- [4] Exemplu de diagramă,
[https://profs.info.uaic.ro/ computernetworks/files/6rc.ProgramareaInReteaII.Ro.pdf](https://profs.info.uaic.ro/computernetworks/files/6rc.ProgramareaInReteaII.Ro.pdf)
- [5] Crearea diagramei,
<https://www.draw.io/>
- [6] Diferențe dintre TCP și UDP,
<https://www.geeksforgeeks.org/differences-between-tcp-and-udp/>