

Documentatie Proiect Monitorizarea Traficului

Rosca Alexandru-David

Universitatea "Alexandru Iona Cuza", Iasi, Romania

December 13, 2023

Abstract

Acest document a fost creat cu scopul de a prezenta tehnologiile folosite, modul de functionare dar si diferite particularitati care pot fi imbunatatite in proiectul "Monitorizarea Traficului"

1 Introducere

Proiectul "Monitorizarea Traficului" oferă clienților aferenți capacitatea de a observa viteza maximă de pe o anumită porțiune de drum, viteza care este influentată atât de posibile accidente cât și de vremea manifestată pe respectiva stradă pe care se află aceștia, împreună cu un indice care le transmite dacă ei depășesc această viteză. Pe lângă aceasta clienții au posibilitatea de a primi știri legate de vreme, sporturi cât și preturile la gaz dacă aceștia doresc acest lucru, știrile fiind actualizate periodic. În final, clienții au opțiunea de a raporta un accident pe strada pe care se află acest lucru apărând pe clientul fiecărui participant la trafic și viteza maximă de pe porțiunea respectivă de drum fiind diminuată.

2 Tehnologii Utilizate

2.1 TCP - Transmission Control Protocol

Am optat pentru utilizarea protocolului TCP (concurrent) datorită capacității sale de a facilita comunicarea între client și server, oferind în același timp asigurarea că pachetele ajung la destinație fără pierderi și sunt reasamblate în ordine, acest lucru fiind crucial datorită necesității de organizare a mesajelor în GUI-ul aplicației precum și păstrării consistenței comunicării (client-server). Alegerea acestui protocol se datorează prioritizării securității ridicate în transferul de date, în ciuda potențialei sale mai scăzute în ceea ce privește viteza, care în cazul de față nu este cel mai important factor din cadrul proiectului, iar TCP oferă un nivel de fiabilitate mult mai înalt în detrimentul altor protocoale precum UDP.

2.2 SQLite

SQLite reprezintă o bibliotecă specifică limbajului C, iar în cadrul aplicației noastre, va fi folosită pentru stocarea informațiilor despre străzi și vitezele maxime de pe acestea. Utilizarea SQLite ne oferă posibilitatea de a manipula într-un mod facil aceste date. Prin natura sa, această bibliotecă este ușor de integrat, rapidă și furnizează un mod eficient de vizualizare și gestionare a datelor, comparativ cu stocarea în fișiere text simple. Alegerea de a implementa SQLite în aplicația noastră este motivată de facilitarea integrării în cadrul proiectului și al accesului la date, performanța sa, făcându-l un partener potrivit pentru necesitățile specifice ale aplicației.

2.3 GTK3

Interfața grafică a aplicației va fi dezvoltată utilizând biblioteca GTK 3, una dintre cele mai extinse și folosite biblioteci grafice de pe piață, deoarece aceasta furnizează trei caracteristici esențiale: accesibilitate, performanță și o documentație extinsă. Deși alte biblioteci, cum ar fi Qt, pot oferi funcționalități similare, am optat pentru GTK 3 datorită flexibilității sale și a faptului că este bine integrată în

mediul grafic al sistemului de operare. În comparație cu alternativele, precum FLTK, GTK 3 este mai potrivită pentru dezvoltarea interfețelor grafice complexe, oferind o gamă extinsă de instrumente pentru personalizarea aspectului și comportamentului aplicației.

3 Arhitectura Aplicației

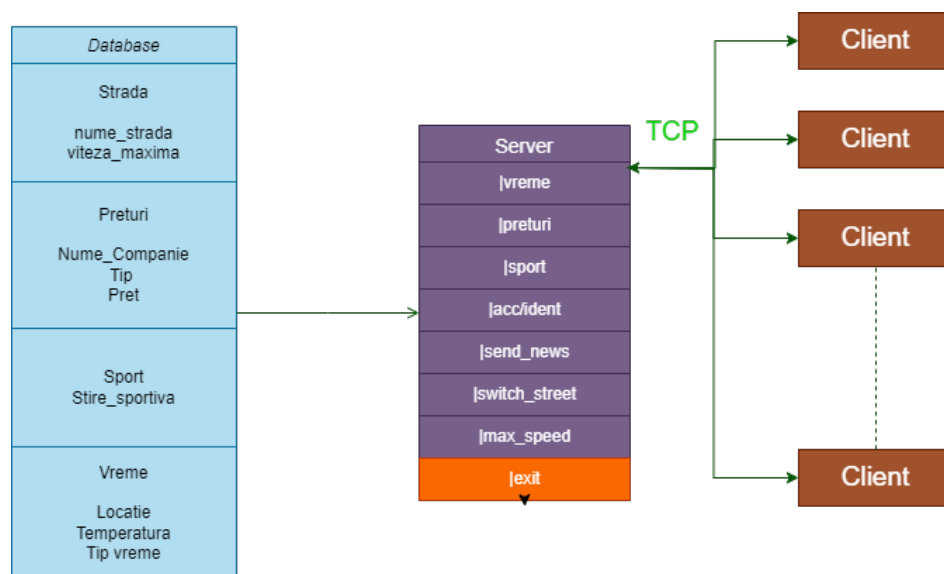
3.1 Concepte implicate

Aplicația "Monitorizarea Traficului" asigură comunicarea client/server într-un mod concurrent. Aceasta va folosi o bază de date SQLite pentru a memora datele legate de strazi precum numele și viteza aferentă și o interfață grafică creată cu biblioteca GTK3 pentru vizualizarea mesajelor returnate de server. Serverul va oferi capacitatea de a returna date despre vreme, anunțuri sportive, cât și preturi la carburanți, pe lângă aceasta serverul returnează date despre viteza cu care poți merge maxim pe un tronson de drum, lucru influențat de temperatura dar nu numai, de asemenea serverul permite schimbarea strazii împreună cu actualizarea valorii vitezei, toate acestea sunt răspunsuri ale serverului la un prompt primit de la client însă mai există o funcționalitate a serverului, aceea fiind cea de a trimite tuturor participanților la trafic un mesaj în cazul în care unul dintre ei a raportat un accident (lucru care duce la modificarea vitezei pe strada respectivă de asemenea).

Comenzile valabile pentru server sunt:

- * |vreme (Switch-ul pentru vreme)
- * |sport (Switch-ul pentru sport)
- * |preturi (Switch-ul pentru preturile carburanților)
- * |send_news (Răspunde cu stiri noi care au fost selectate anterior)
- * |acc\ident (Anunță un accident pe o anumită stradă)
- * |switch_street (Schimbă strada din cea actuală în cea dată ca parametru și ne spune cu ce viteză să circulăm pe ea)
- * |exit (Închide clientul)
- * |max_speed (Trimite viteza și returnează viteza cu care trebuie circulat pe tronsonul de drum aferent)

3.2 Diagrama aplicației



4 Detalii de Implementare

Pasii urmati de client

1. Crearea socket-ului
2. Conectarea la server
3. Inchiderea conexiuni

```
if (argc != 3)
{
    printf ("[client] Sintaxa: %s <adresa_server> <port>\n", argv[0]);
    return -1;
}

port = atoi (argv[2]);

if ((sd = socket (AF_INET, SOCK_STREAM, 0)) == -1)
{
    perror ("[client] Eroare la socket().\n");
    return errno;
}

server.sin_family = AF_INET;
server.sin_addr.s_addr = inet_addr(argv[1]);
server.sin_port = htons (port);

if (connect (sd, (struct sockaddr *) &server, sizeof (struct sockaddr)) == -1)
{
    perror ("[client]Eroare la connect().\n");
    return errno;
}
```

Pasii urmati de server

1. Asculta daca vin clienti sa se conecteze
2. Serverul primeste comanda si da un raspuns(comenzile posibile sunt:

```
|max_speed
|vreme
|sport
|preturi
|send/news
|switch_street
|acc/ident
|exit

char strada_client[MSG_LENGTH], viteza_client[MSG_LENGTH];
int i = 0;

bzero(strada_client, MSG_LENGTH);
bzero(viteza_client, MSG_LENGTH);

for(int j = 0 ; j < strlen(msg) ; j++)
{
    if(msg[j] != ' '){
```

```

        {
            strada_client[i] = msg[j];
            i++;
        }
    else
    {
        strada_client[i] = ' ';
        strada_client[i + 1] = '\0';
        break;
    }
}

/////Restul codului

fclose(filee);
fclose(file);

FILE *srcfile, *destfile;
char buffer[MSG_LENGTH];

srcfile = fopen("strazi/decoy_accidente", "rb");
if (srcfile == NULL) {
    perror("Error opening source file");
}

destfile = fopen("strazi/strada_km", "wb");
if (destfile == NULL) {
    perror("Error opening destination file");
    fclose(srcfile);
}

size_t bytesRead;
while ((bytesRead = fread(buffer, 1, sizeof(buffer), srcfile)) > 0) {
    fwrite(buffer, 1, bytesRead, destfile);
}

fclose(srcfile);
fclose(destfile);

//////////

srcfile = fopen("strazi/decoy_accidente", "w");
if (srcfile == NULL) {
    perror("Error opening source file");
}

fclose(srcfile);

return bytes;

```

3. Serverul trimite la toti clientii mesaj legat de accident

```

char strada_client[MSG_LENGTH], viteza_client[MSG_LENGTH];
int i = 0;

bzero(strada_client, MSG_LENGTH);

```

```

bzero(viteza_client,MSG_LENGTH);

for(int j = 0 ; j < strlen(msg) ; j++)
{
    if(msg[j] != '|')
    {

        strada_client[i] = msg[j];
        i++;
    }
    else
    {
        strada_client[i] = ' ';
        strada_client[i + 1] = '\\0';
        break;
    }
}

        /////Restul codului

fclose(filee);
fclose(file);

FILE *srcfile, *destfile;
char buffer[MSG_LENGTH];

srcfile = fopen("strazi/decoy_accidente", "rb");
if (srcfile == NULL) {
    perror("Error opening source file");
}

destfile = fopen("strazi/strada_km", "wb");
if (destfile == NULL) {
    perror("Error opening destination file");
    fclose(srcfile);
}

size_t bytesRead;
while ((bytesRead = fread(buffer, 1, sizeof(buffer), srcfile)) > 0) {
    fwrite(buffer, 1, bytesRead, destfile);
}

fclose(srcfile);
fclose(destfile);

//////////

srcfile = fopen("strazi/decoy_accidente", "w");
if (srcfile == NULL) {
    perror("Error opening source file");
}

fclose(srcfile);

return bytes;

//////////

```

```

char msgrasp[MSG_LENGTH]=" ";

bzero(msgrasp,MSG_LENGTH);
strcat(msgrasp,":A fost raportat un incident\npe strada ");
strcat(msgrasp,message);
strcat(msgrasp, ".\nReduceti viteza si conduceti\ncu mare atentie!");

printf("%s\n",msgrasp);

if (write (fd, msgrasp, strlen(msgrasp)) < 0)
{
    perror ("[server] Eroare la write() catre client.\n");
    return 0;
}

```

4. Inchidere server:

5 Concluzii

Proiectul "Monitorizarea Traficului" permite clientilor sa participe la o simulare virtuala de trafic din cadrul unui oras, unde acestia pot primi in mod periodic striri cat si valoarea vitezei cu care acestia ar trebui sa circule, si in caz de accident pot anunta restul participantilor la trafic despre acest lucru.

Imbunatatiri posibile ale aplicatiei pot fi, adaugarea unui chat intre participantii la trafic, o harta digitala care sa ne permita sa vedem unde ne aflam pe strada in comapratie cu restul participantilor, dar si o alta imbunatatire ar putea fi adaugarea unui sistem de encriptarea a datelor pentru a putea pastra in siguranta detalii importante legate de participantii la trafic, precum locatia acestora.

6 Bibliografie

References

<https://profs.info.uaic.ro/computernetworks/cursullaboratorul.php>
<https://docs.gtk.org/gtk3/>
<https://www.jetbrains.com/datagrip/features/>
https://ro.wikipedia.org/wiki/Transmission_Control_Protocol