Documentatie -WeatherMonitor(C)

Palalae Stefan Tudor
Gr X2

1.Introducere

Scopul acestei documentii este sa puna la curent posibilul user al acestei aplicatii.
Aceasta este o aplicatie server/client scrisa in C++ si se numeste “WeatherMonitor”.
Nu este necesara logarea clientului pe server pentru a putea accesa featurile serverului.
Avem o baza de date care este administrata de catre server, acesta la randul lui urmand cerintele clientului/userului.

2.Tehnologii utilizate

Conexiunea dintre server si client va fi de tip TCP concurent, concurenta avand ca scop comunicarea in paralel a serverului cu mai multi clienti neexistanta idea de astaptare intre clienti, toti acestia fiind servici in paralel. Am realizat conexiunea print-un socket oferit de librariile deja existente din C/C++.

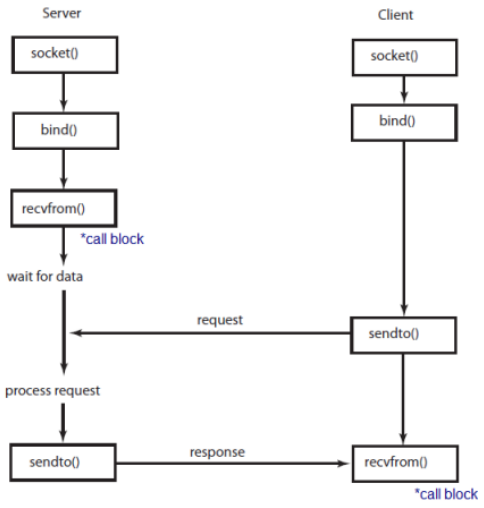
Alt tip de tehnologie folosita este cea de creare si administare a bazei de date, am ales Sqlite3 deoarece este una dintre cele mai light DB si simple de folosit, un dezavantaj ar fi ca este limita la 2GB max, dar nu este o problema in aceasta situatie.

Am folosit utilitatile din C/C++ pt a face serverul concurent, anume “fork()”, la fiecare alt client conectat este creat inca un proces copil.

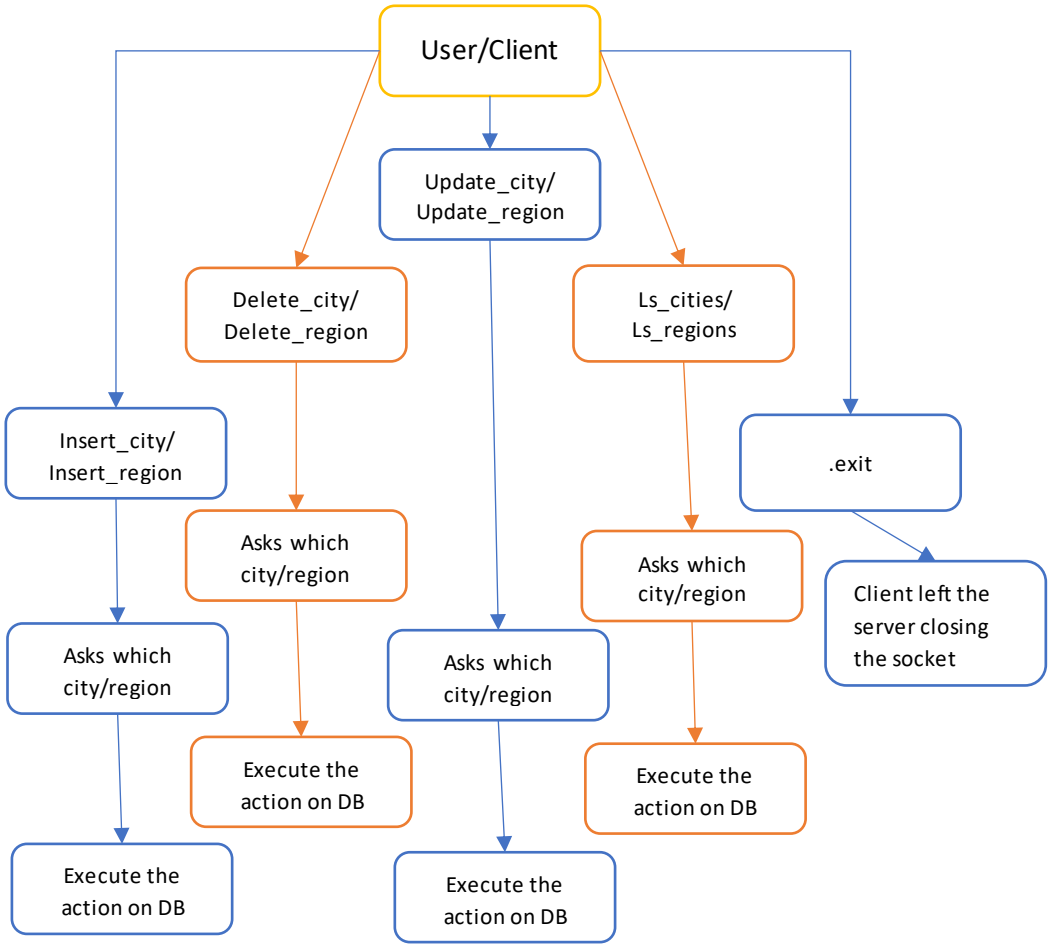
3. Arhitectura aplicatiei

Exista 3 componente ale acestei aplicatii: server, client si db. Intre aceste 3 componente exista 2 legaturi, legatura server-client si server-db, clientul neavand contactul direct cu db.
Schita DB-ul este si ea destul de simplista, avem 2 tabele, cities si regions, amandoua reprezinta zona mai mari sau mai mici de teren, deci nu prea exista diferente intre ele.

Arhitectura este exact ca la oricare alta aplicatie server-client.



Comenzi posibile



4.Detalii de implementare

Fiecare actiune are o comanda de executie care da request serverului, aceastea sunt:

a) Insert_city/Insert_region

Odata scrisa aceasta comanda serverul trimite formatul in care trebuie sa arate datele inserate

b) Delete_city/Delete_region

Odata scrisa aceasta comanda serverul intreba ce oras/regiune este dorita a fi stersa.

c) Update_city/Update_region

Odata scrisa aceasta comanda serverul intreba ce oras/regiune este dorita a fi schimbata, dupa care cere ce schimbari vor a fi aduse acelui lucru selectat.

d) Ls_cities/Ls_regions

Odata scrisa aceasta comanda serverul trimite raspuns o lista cu toate orasele/regiunile.

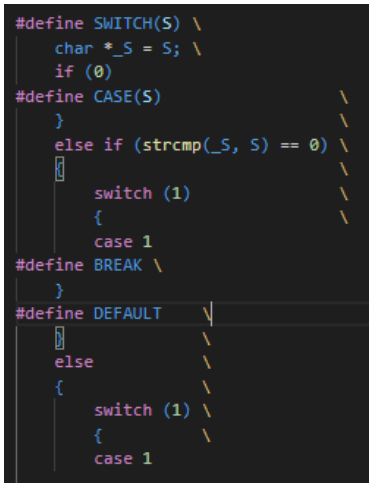
e) .exit

Odata scrisa aceasta comanda clientul inchide socketul din partea lui, serverul urmandu-l inchizand si el socketul.

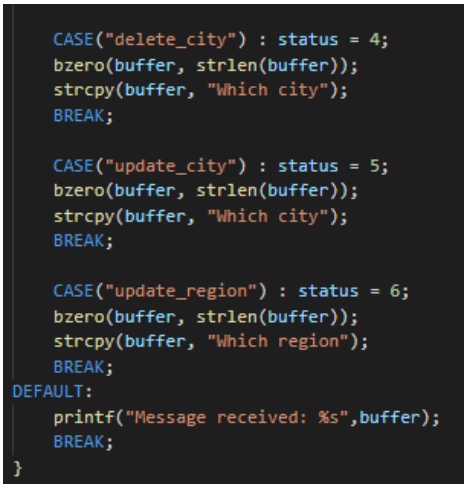
O alta chestie pe care am facut-o pentru a face codul mai citibil si pentru mine si pentru oricine e sa implementez niste #define-uri pentru a scapa de o gramada de if-uri si else-uri, codul arata de parca ar fi un switch, defapt sunt tot if-uri si else-uri care simuleaza un switch. Nu putea folosi switchul classic deoarece in conditie trebuia sa folosesc strcmp(), asta nefiind posibil in switch, switchul in mod normal doar compara 2 chestii prin operatorul "=".

Am 2 tipuri de switch pentru ca aveam nevoie si la variabile de tip int "status". Voiam sa evit folosirea comenzilor "break" clasice deoarece eu eram deja in while(1) pt a mentine comunicarea cu clientul.

Variabila "status" se ocupa de partea cu intrebatul clientului si practic setarea unei comenzi. De exemplu daca scrii "insert_city" atunci la urmatoarea comanda esti obligat sa inserezi orasul.
Mai jos se pot vedea niste poze ce arata implementarea:



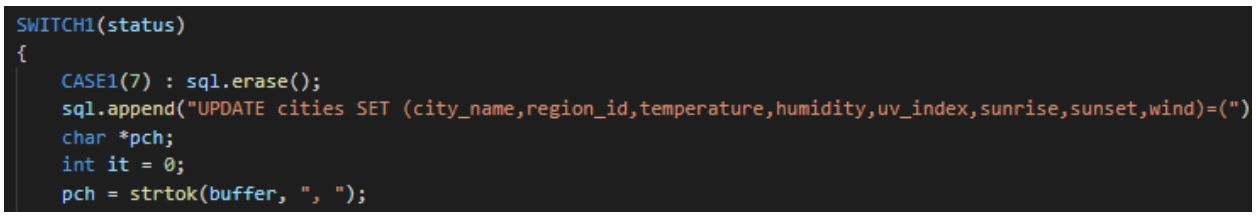
```
#define SWITCH(S) \
    char *_S = S; \
    if (0)
#define CASE(S)                 \
    }                           \
    else if (strcmp(_S, S) == 0) \
    {                           \
        switch (1)              \
        {                       \
        case 1
#define BREAK \
    }
#define DEFAULT     \
    }               \
    else            \
    {               \
        switch (1) \
        {          \
        case 1
```



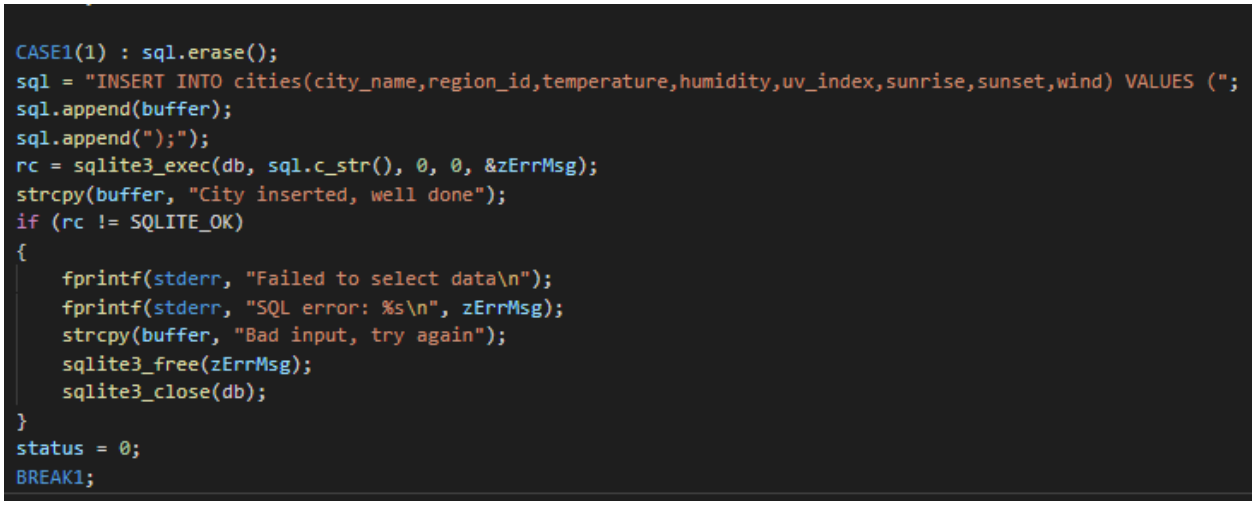
```
    CASE("delete_city") : status = 4;
    bzero(buffer, strlen(buffer));
    strcpy(buffer, "Which city");
    BREAK;

    CASE("update_city") : status = 5;
    bzero(buffer, strlen(buffer));
    strcpy(buffer, "Which city");
    BREAK;

    CASE("update_region") : status = 6;
    bzero(buffer, strlen(buffer));
    strcpy(buffer, "Which region");
    BREAK;
DEFAULT:
    printf("Message received: %s",buffer);
    BREAK;
}
```



```
SWITCH1(status)
{
    CASE1(7) : sql.erase();
    sql.append("UPDATE cities SET (city_name,region_id,temperature,humidity,uv_index,sunrise,sunset,wind)=(")
    char *pch;
    int it = 0;
    pch = strtok(buffer, ", ");
```



```
CASE1(1) : sql.erase();
sql = "INSERT INTO cities(city_name,region_id,temperature,humidity,uv_index,sunrise,sunset,wind) VALUES (";
sql.append(buffer);
sql.append(");");
rc = sqlite3_exec(db, sql.c_str(), 0, 0, &zErrMsg);
strcpy(buffer, "City inserted, well done");
if (rc != SQLITE_OK)
{
    fprintf(stderr, "Failed to select data\n");
    fprintf(stderr, "SQL error: %s\n", zErrMsg);
    strcpy(buffer, "Bad input, try again");
    sqlite3_free(zErrMsg);
    sqlite3_close(db);
}
status = 0;
BREAK1;
```

```c
int callback(void *NotUsed, int argc, char **argv, char **azColName)
{
    NotUsed = 0;
    for (int i = 0; i < argc; i++)
    {
        strcat(buffer, argv[i] ? argv[i] : "NULL");
        strcat(buffer, ",");
    }

    printf("\n");
```

5.Concluzie

Aplicatia functioneaza pentru cea ce cere cerinta insa poate fi imbunatatita, o idee ar fi fost sa fac request si parsing cu date reale de pe un site de vreme si sa le actualizeze periodic in DB. Legat de tehnologia TCP folosita, este necesara pentru a asigura ajungerea la destinatie a datelor, UDP aducea mai multa viteza insa nu putea avea incredere ca ajung mereu datele la destinatie.

6. Bibliografie

Course&Laboratory - Computer Networks 2020-2021 (uaic.ro)
Explicitly assigning port number to client in Socket - GeeksforGeeks
tcp - different socket buffer size for client and server - Stack Overflow
c++ - Sending and receiving std::string over socket - Stack Overflow
Multiple Client Server Program in C using fork | Socket Programming - YouTube
linux - How to completely destroy a socket connection in C - Stack Overflow
c - Bind error while recreating socket - Stack Overflow

Link catre proiect:

PalalaeStefan301/WeatherMonitor-Linux: An server/client application who administrate a DB of cities/regions and there meteorological infos (github.com)