Topic: Future Work

Name: Aikaterini Karanasiou

Improvements

[1] In the current implementation, *ConnectionWeatherAPI* class includes a list of static functions. I would prefer to use a namespace instead of class for the definition of functions.

[2] *currentDate()* function is defined two times in two different files (for two processes): *Command.cpp* and *WeatherData.cpp files*. I could define this function in one source file in order to be used by both processes.

[3] *"WeatherData"* stores the current and historical Weather Data. When this app starts, I could check if DB already exists. If it exists, I could start storing data from the latest stored date (and not from 01-01-2021).

[4] I would like to add the following changes on the makefile (that is used for "*WeatherData"* process) for adding and removing files of DB.

```
WeatherData: .....

    .....

    mkdir ../../Database

clean_db:

    rm ../../Database/WeatherDatabase.db*
```

[5] I observed that if *"WeatherData"* and *"WeatherCLI"* are executing in parallel, then it is possible to execute parallel write and read requests to the database.

In case that there are parallel read and write requests to the DB, the read requests are failing.

Due to this, I added the following command: ***PRAGMA journal_mode=WAL;***

I observed that using the *"journal_mode=WAL"* command, several parallel read-write processes are executed correctly.

In order to ensure that all parallel read and write processes are executing successfully I could provide one of the following changes:

[5a] Executing manually the ***PRAGMA locking_mode=EXCLUSIVE.*** I tested and it works correctly only If the database is open.

[5b] *"WeatherData"* could store the weather data both on the DB and the shared memory and the *"WeatherCLI"* could read the weather data only from the shared memory.

Corrections

[1] I initialize a new curl session on *GetHourlyAverageTemperature()* and *GetCurrentTemperature()* functions using the following command:

CURL* curl = curl_easy_init();

After I start a curl session, I remove the memory that is allocated for this session using the following command:

curl_easy_cleanup(curl);

I forgot to call this command on the 2nd function (*GetCurrentTemperature()*).

[2] *SendGETrequest(CURL* curl, string url, const string &receivedData, long &result_httpCode, CURLcode &result_curlCode)* is called for storing the response of HTTP GET request on the *receivedData* argument.

Due to this, the *receivedData* should not be passed as *const* argument on this function.

Although I pass this argument as *const*, the response is stored successfully to the *receivedData.*

I think that this is happening because *receivedData* is passed as argument on the callback() function and due to this, there is a typecasting from *const string &* to the *string*.* This is possible on C++ but maybe it is should not be used.

[3] On the *doubleResultFromSelectQuery()* and *intResultFromSelectQuery()* functions, I forgot to put the following output displays before the ***if (argc == 1 && argv[0]!=nullptr)*** condition:

cout << endl << resName[0] << " = " << ((argv[0]) ? argv[0] : "NULL") << endl;

[4] On the *ConnectionWeatherAPI* class, I define a *const* parameter that it is never used. (I forgot to remove this declaration)