System Programming
C programming manual: lab 7
2015 - 2016

Bachelor Electronics/ICT

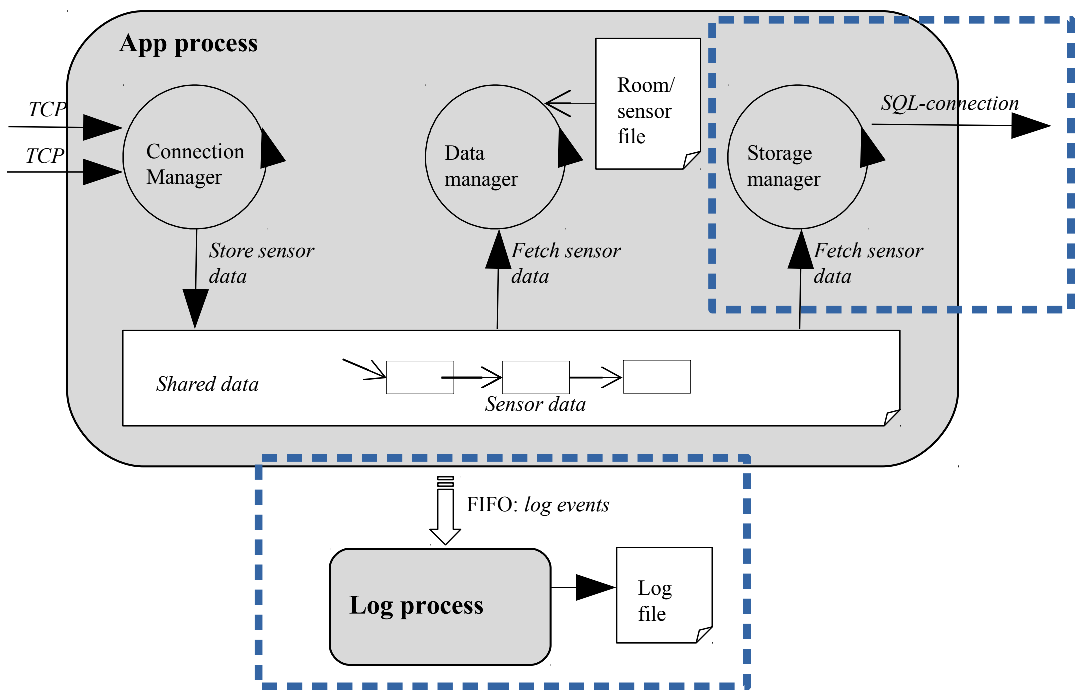
Course coördinator: Luc Vandeurzen
Lab coaches: Jeroen Van Aken
Stef Desmet
Tim stas
Luc Vandeurzen

Last update: March 14, 2016

C programming

Lab targets:file I/O, SQL database interfacing, processes and IPC (pipe) communication

For your information
The picture below visually sketches the final assignment of this course. The relationship of this lab to the final assignment is indicated by the dashed blue line.



Exercise 1: *sqlite*

Sqlite is a self-contained, embeddable SQL database system written in C and completely open source. Because of it's embeddable approach, it's a very popular database used by applications for local storage. It's used as a local storage system in Android and by many other famous applications such as firefox, chrome, dropbox, facebook, … You find more information on sqlite on its home page: www.sqlite.org.

We will use 'sqlite3' in this lab. Install it on your system. It could be interesting to also install the 'sqlitebrowser', a GUI editor for sqlite databases. After the installation of sqlite3, the sqlite3 library should be installed on your system and you can use already the sqlite shell interface: just type 'sqlite3' on the command line and follow the help.

Exercise 2: *file I/O and sqlite interfacing*

THE SOLUTION OF THIS EXERCISE NEEDS TO BE UPLOADED AS A ***ZIP FILE*** *ON syssoft.groept.be BEFORE THE NEXT LAB.*

YOUR SOLUTION IS ONLY ACCEPTED IF THE CRITERIA FOR THIS EXERCISE AS DESCRIBED ON syssoft.groept.be ARE SATISFIED!

Find on Toledo a 'sensor_db.h' and 'config.h' file. The 'sensor_db.h' file contains the function prototypes to insert sensor measurements read from the file 'sensor_data' (see lab 6) into the database and to search the database on sensor value or timestamp. Write the corresponding 'sensor_db.c' file and implement a 'main.c' file to test the 'sensor_db.c/.h' code.

The 'init_connection()' creates or opens a database with one table. This table should have the following columns:

- id: automatically generated unique id (AUTOINCREMENT)
- sensor_id (INT)
- sensor_value (DECIMAL(4,2))
- timestamp (TIMESTAMP)

The default name of the database and table are "Sensor.db" and "SensorData", respectively. But it should be possible to set other names as preprocessor directives, as indicated in sensor_db.h.

Some information sources that will help you a lot:

- A tutorial that will guide you all the way through this exercise: http://zetcode.com/db/sqlitec/

- The sqlite C API reference manual: https://www.sqlite.org/c3ref/intro.html

You can compile your code using the sqlite library as follows:

```
> gcc -Wall main.c sensor_db.c -lsqlite3
```

But it's also possible to compile sqlite3 from source. Download from the sqlite home page the sqlite source code – the so called 'amalgamation'. Compile now as follows:

```
> gcc -Wall main.c sensor_db.c sqlite3.c -pthread -ldl
```

The option -pthread is used to include the POSIX pthread library. More on threads will follow in lab 9. And what about the option '-ldl'? Well, you should know this one already (hint: on demand loading).

Also notice that if you compile the above with the code size optimization option (-Os) on, the executable is remarkable smaller. Try it with and without this option and compare the sizes!

Both compilation options (library or source code) should work for your solution. Test it!

Exercise 3:

Change the program above (= storage manager) such that a new log process is created and a communication pipe between the storage manager and the newly created log process is installed. The log process receives log-events from the storage manager and writes them to a text file (the log file).
A log-event contains an ASCII info message describing the type of event. A few examples of log-events are:

- Connected to SQL server.
- Unable to connect to SQL server.
- New table <name-of-table> created.
- Data insertion failed.

For each log-event received, the log process writes an ASCII message of the format <sequence number> <timestamp> <log-event info message> to a new line on a log file called "gateway.log".