TP4 - KEY VALUE STORE
(1)

"playing with uthash"

INTRO

Greetings!

From this point and on, you will start developing a small key-store server, strictly using C. The assignments are going to be chained together as specific parts of the application for every TP. They will cover:

- simple dynamic memory management with uthash functionality
- string parsing and string manipulation for key store commands/transactions
- simple file access and manipulation
- sockets
- maybe more things if we have time...

Playing with uthash

In this TP you will get acquainted with uthash (the container for key-store transactions). It is a hash written in C with heavy use of MACROS and inline calls, in the form of a simple header file (provided).

more information (read through please):
http://troydhanson.github.io/uthash/userguide.html#_a_hash_in_c

The point of this TP is to implement the prototype functions provided in the store.h header:

openStore(..) : allocate your container.
closeStore(..) : close container with proper cleanup.
setValue(..) : insert value with key - replace old value.
getValue(..) : retrieve value with key.
removeValue(..) : remove value with proper cleanup.
storeSize(..) : number of elements in Store - check uthash MACROs.

1. You will have to declare your hash **Record** structure as well as the container for this record using the uthash logic (**Store**).
2. The “value” of the Record is going to be a simple C string (null terminated).
3. Memory management for this container has to be dynamic with the use of **malloc(..)** and **free(..)**
4. Remember! you are wrapping around uthash MACROS.
5. Provide a **main** function for testing your implementation with example calls - print records.
6. Your code has to compile without any errors.
7. A project structure is provided in https://chamilo.unige.ch/

GOOD LUCK!