# Operating System
# Assignment 3 :
# FAT32 File system

Sacha Allan BRON, Jérémy RABASCO

May 20, 2014

## 1   State of implementation

Everything was implemented and should work quite well. IN particular our file system should be able to:

- Check that the volume is really a FAT32 volume

- Retrieve and list every entry in a directory while ignoring deleted and hidden files

- Read a file

- Verify that one don't try to access a file or directory that does not exist

- Set correctly the dates of creation, modification and last access to a file

- Handle long file names

As far we tested, all these features work.

## 2   Experiences

### 2.1   Documentation

An important part of the work on this assignment was to learn how the FAT32 File system is organized. We spent time on reading some parts of the Microsoft documentation on FAT, Wikipedia articles about FAT and some websites as StackOverflow for particular problems. The hard thing about this part is that a file system is a complex thing and the precise documentation is long and sometimes hard to read and there are a lot of specific values for each specific case, like the ones for the deleted file, hidden file etc... And to make sure that we understood well, we sometimes read some parts of the provided FAT32 volume (with a C program or an hexadecimal editor) to see ourselves how the values were set.

We also documented ourselves a little about fuse in order to use it well.

## 2.2  Implementation

We chose to use the skeleton you provided us for the implementation. The hardest part of the implementation, is linked with the fact, as mentioned above, that there are many specific values in a lot of bytes that mean a lot of things. This fact forced us to constantly look at the documentation each time we wanted to implement something because each operation on a file system requires to check a lot of values in order to make the right choices to perform the required action. This makes the process of coding much slower than it could be but it is necessary if one wish to read the file system properly and retrieve the correct informations for fuse.

## 2.3  Testing

We mainly tested our implementation in the following ways:

- Calling the function currently being implemented with well chosen argument to see the behavior of the program with some calls to `printf` to see the state of execution

- Exploring the given FAT32 volume with an hexadecimal editor to verify the consistency of the values printed when calling a function

- Mounting the volume with our implementation and comparing the result with the one obtained with the `mount` command of linux (we explored the directories and read the files to see if they were the same in both cases)