Rapport du projet : Mini S.G.F

# Introduction

Le but ce de projet était de comprendre le fonctionnement d’un système de gestion de fichier en simulant un mini-système.

Certaines fonctionnalités étaient implémentées par défaut, notamment celles très « bas-niveau », d’autres telles que l’affichage du contenu du disque, la lecture d’un fichier ou l’écriture bloc par bloc étaient à réaliser.

Ce projet s’est révélé, par moment, complexe et difficile à appréhender mais nous a permis de vraiment nous former et d’assimiler les concepts essentiels de la gestion des fichiers sur une machine. C’est un composant absolument essentiel d’un système d’exploitation, et pas assez souvent mis en avant ; avant ces TPs et ce projet nous n’avions pas forcement conscience d’à quel point un SGF était important et difficile a mettre en œuvre.

Bien évidemment notre « mini-SGF » est très basique, mais il nous a ouvert les yeux une fois de plus sur la puissance et la complexité de nos systèmes informatique modernes, nous imaginons maintenant à quel point une simple ouverture-écriture d’un document PDF implique des traitements immenses du côté de la machine.

# Arborescence du projet

Notre archive de rendu possède un dossier CODE qui, comme son nom l’indique, rassemble les fichiers du « mini-SGF ».

Nous avons principalement modifié les fichiers :

* main.c : Initialisation du SGF et lancement d’une fonction de test (voir la section Tests).
* sgf-io.c : Comme son nom l’indique (IO), c’est ici que se rassemblent l’ensemble des fonctionnalités d’entrées et de sorties. C’est dans ce fichier que la plupart du travail a été réalisé.
* Makefile : Pour la compilation ; Ajout des options « -w », « -g » et « -std=c99 » à des fins de débuggage ou d’autoriser certains formes de code ( commentaires en //, déclaration du int dans la condition d’un for, etc… )

# Lecture dans un fichier

La fonction list\_directory consiste, comme vu en TD, à parcourir toutes les entrées de la **FAT** afin de chercher les fichiers présents sur notre disque. Ces fichiers sont décrits par un **inode**, par conséquent, au cours de notre parcours, des lors que nous tombons sur un bloc **inode**, nous le lisons afin d’extraire les informations du fichier comme sa longueur, son premier et son dernier bloc de contenu.

La fonction sgf\_read\_bloc consiste à chercher le bloc logique *n* d’un fichier *f.* Pour cela, on part du premier bloc (ciblé par *f->first*), et on passe n fois au bloc suivant (adresse retournée par get\_fat(bloc\_actuel)). Si aucune erreur n’a été rencontrée jusqu’à la fin de ce *while*, alors nous avons l’adresse du bloc logique n et nous pouvons alors charger son contenu dans le buffer du fichier f.

## Acces direct en lecture