

# Projet — Système de Gestion de Fichiers

Antoine de ROQUEMAUREL (Groupe 2.2)

---

## 1 Complétion des fonctions de base du SGF

### 1.1 Lecture et écriture du SGF

**lire\_superbloc** Afin de lire le superbloc, cette fonction positionne tout d'abord le curseur au premier bloc du fichier puis lit dans `mon_SGF->superbloc` les données du superbloc, sa taille est connue grâce à `sizeof(superbloc_t)`.

*Si une erreur se produit, lors du `read`, la fonction retourne `EXIT_READ_PB`.*

**lire\_table\_inoeuds** La lecture de la table des inoeuds est le même principe que pour le superbloc, nous savons que celle-ci est écrite à partir du bloc 1<sup>1</sup>, ainsi il faut se positionner au bloc 1, puis faire lire `sizeof(inoeud_t)` octets.

*En cas d'erreur du `read`, la fonction retourne `EXIT_READ_PB`.*

**lire\_SGF** La lecture du SGF ne peut se faire que si le SGF est ouvert correctement, cette pré-condition est vérifiée à l'aide de `assert`. Afin de lire le contenu du SGF, il nous suffit d'utiliser les deux fonctions écrites précédemment, cependant avant d'appeler ses fonctions, un `malloc` doit être fait.

*En cas d'erreur d'allocation mémoire, la fonction retourne `EXIT_MEM_ALLOC`.*

**ouvrir\_SGF** Tout d'abord, il faut allouer la mémoire pour la structure de type `SGF_t` contenant notre SGF, si ce `malloc` s'effectue correctement, il faut ouvrir le fichier `DEVICE_NAME` à l'aide de la fonction `open`.

Nous ne gérons pas le cas où le `open` retourne `NULL`, en effet ce cas est systématiquement géré après l'appel de notre fonction.

### 1.2 Gestion des inoeuds et blocs

**bloc\_libre\_suivant** La liste des blocs libre est représentée par une liste chaînée. Sur le disque dur, à la position de chaque bloc libre est stocké le numéro du bloc libre suivant.

Ainsi, en se déplaçant de `num_bloc` blocs<sup>2</sup>, nous pourrons ensuite lire le numéro du bloc libre suivant à l'aide d'un `read`.

*En cas de problème au niveau du `read`, la fonction retourne `EXIT_READ_PB`.*

**allouer\_n\_blocs\_dans\_inoeud**

**lire\_donnees\_dans\_inoeud**

**ecrire\_donnees\_dans\_inoeud**

**creer\_inoeud**

**inoeud\_libre** Le premier inoeud libre est trouvé en parcourant la table des inoeuds jusqu'à trouver un inoeud ayant pour type `INOEUD_LIBRE`, si nous arrivons à la fin de la table sans avoir trouvé d'inoeud libre, alors il faut renvoyer `NO_INOEUD`.

**liberer\_blocs\_du\_inoeud**

---

1. le bloc 0 étant réservé au superbloc

2. la taille d'un bloc étant connue dans le superbloc

**liberer\_inoeud** Pour libérer un inoeud, il faut appeler la fonction développée précédemment, **liberer\_blocs\_du\_inoeud**, celle-ci libérera tous les blocs de l'inoeud. Une fois les blocs libérés correctement, la taille et le type de l'inoeud sont modifiés. Ensuite il ne faut pas oublier de mettre à jour la table des inoeuds dans le fichier à l'aide de la fonction **ecrire\_table\_inoeuds**.

## 2 Commandes d'accès au SGF

### 2.1 `ls -l`

**afficherInoeud**

### 2.2 `du`

**tailleSousRep**

**tailletotalrepertoire**

## 3 Exemples d'exécution