

Daniel Borges (dborges@etudiant.univ-mlv.fr)

Projet Système

FUSE-L3

Présentation

Ce projet est un système de fichiers implanté en espace utilisateur à l'aide de la bibliothèque FUSE, pour le système GNU/Linux.

Fonctionnalités

- support des fichiers de taille infinie en interne
- répertoires aux noms de taille infinie
- liens matériels
- 256 inodes maximum

(voir inode.h pour changer ce nombre, il devra être multiple de 8 et maximum de 2^{32})

- système de fichiers « mappé » en mémoire (1Go)
- système de fichier contenu dans un fichier du système de fichiers sous-jacent.
- système de fichiers miroir
(le système de fichiers est contenu dans 2 fichiers identiques, si un fichier est détecté comme étant corrompu, l'autre système de fichiers lui est immédiatement recopié)

Structure du projet

Le système de fichiers doit être capable de gérer des fichiers de taille infinie et ce pour 256 inodes. J'ai donc séparé le système de fichiers en 2 parties : les inodes et les blocks.

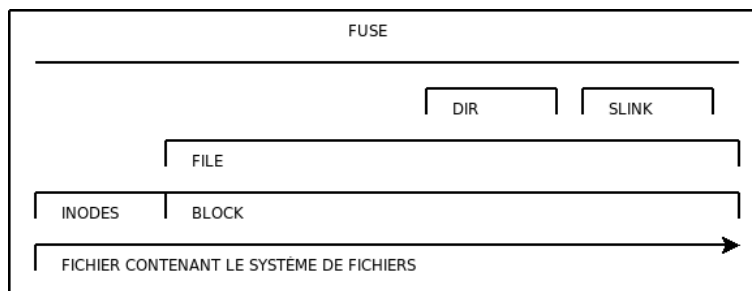
Les inodes occupent la première partie du fichier contenant le système de fichier. Ils commencent donc à l'octet 0. Une inode contient ici toutes les informations requises par un appel à stat (ou presque, des informations comme l'ID du périphérique contenant le fichier étant inutile) ainsi qu'un numéro de block correspondant au premier block de la partie données du fichier stocké. Toute la partie inode est gérée par les fichiers inode.c et inode.h.

Les blocks contiennent toutes les données stockées qui font les fichiers ; suivant la philosophie UNIX que « tout est fichier », les répertoires et les liens symboliques sont aussi contenus dans les blocks. Nous avons donc ici une première abstraction, toute la gestion des blocks se fait donc par les fichiers block.h et block.c.

Les fichiers file.c et file.h contiennent un système, au dessus du système de blocks, permettant de gérer les données associées à un fichier. Il s'agit donc d'ensembles de blocks. Les fichiers devant être de taille infinie, la structure pour gérer les fichiers a été réalisée sous la forme d'une liste de tableaux d'adresses de blocks.

Un répertoire est un simple fichier contenant des couples de valeur <numéro d'inode, nom de fichier>. Le nom de fichier s'arrête par un caractère nul, comme les chaînes de caractères en C, la taille d'un nom de fichier est donc infinie. Les répertoires sont gérés par les fichiers dir.c et dir.h.

Les liens symboliques sont aussi de simples fichiers contenant tout simplement une chaîne de caractères pour stocker le fichier pointé. (on ne stocke pas le numéro d'inode, puisqu'un lien symbolique peut pointer en dehors du système de fichier ou même pointé sur un fichier qui n'existe plus et qui n'a donc plus de numéro d'inode) La gestion des liens symboliques est contenue dans slink.c et slink.h



Description détaillée

1. Inodes

Les inodes sont des structures contenant toutes les informations qu'on retrouve dans la structure stat, sauf celles qu'on peut calculer, plus l'adresse du début du fichier dans le système de blocks.

Il y en a 256 enregistrées au début du fichier contenant le système de fichiers. Avant ces 256 inodes se trouve un tableau de 256 bits permettant de savoir si une inode est occupée (bit à 1) ou non.

2. Blocks

Un block est un ensemble de 512 octets. Ils sont enregistrés après les inodes.

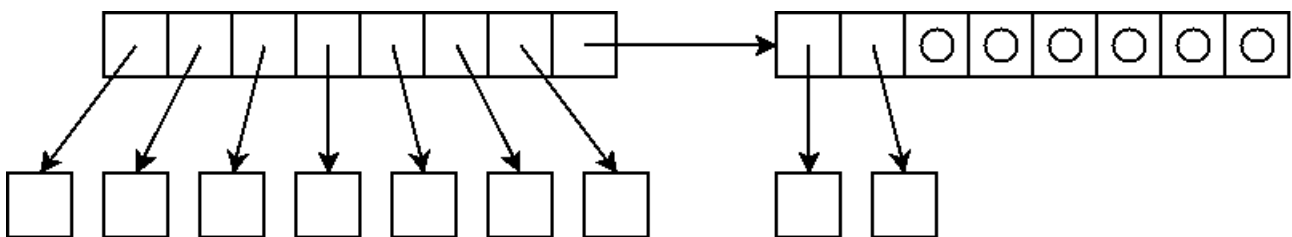
Le système permettant de gérer les blocks se décompose en une série de superblocks. Un superblock contient un block représentant un tableau de bits permettant de savoir si tous les blocks suivants, dans la limite du nombre de bits enregistrable dans un block (soit donc 512×8 bits), sont utilisés ou non. Pour accélérer la recherche de block libre, le premier bit sert à savoir si le superblock est plein ou non. Ainsi, un superblock contient exactement 4096 blocks dont 4095 contenant des données.

Le fichier contenant le système de fichiers grossit automatiquement d'un superblock à chaque fois qu'on a besoin d'un superblock supplémentaire pour stocker des données.

3. Ensemble de blocks (fichiers)

Comme nous avons besoin de stocker des fichiers de plus de 512 octets, et plus particulièrement des fichiers de taille infinie, le système de fichiers est conçu sous la forme d'une liste de tableaux de blocks. Le fichier commence par un block contenant les adresses des blocks suivants composant le fichier. Le dernier numéro de block pouvant être contenu dans un block correspond en fait à l'adresse d'un autre super block contenant des adresses. On peut donc ainsi avoir une infinité de blocks de données et donc des fichiers de taille infinie.

Le block 0 n'est jamais utilisé par le système d'ensemble de blocks, pour avoir un numéro de block invalide.



4. Répertoires

Un répertoire est un fichier spécial contenant les données suivantes :

- les 4 premiers octets contiennent le nombre d'entrées dans le répertoire
- les éléments suivants du répertoire sont des couples de données :
 - 4 octets pour stocker le numéro de l'inode correspondant permettant de retrouver les données
 - une chaîne de caractères terminée par un caractère nul.

On peut ainsi avoir jusqu'à $2^{32}-1$ entrées dans un répertoire, les noms de fichiers n'étant pas limités en taille.