

PNL - Projet 2019/2020

ouiche_fs - le système de fichiers le plus classe du monde

1. Rotation du système de fichier

Principes

Stratégie générale

L'idée de base est de placer dans la fonction `ouichefs_create()` une vérification de l'espace disque libre puis du nombre de fichiers dans le répertoire courant, afin de déclencher le nettoyage, soit du plus gros/plus vieux de la partition et/ou du répertoire courant.

Détection du % de blocs libres

Nous avons créé une macro dans `ouichesfs.h` qui permet à l'utilisateur avant compilation de déterminer quel sera la limite du % de bloc libres.

Une fonction `check_limit()` se charge d'analyser le superblock et de renvoyer un booléen pour indiquer si la limite est dépassée ou non.

Détection d'un répertoire plein

Une fonction `is_dir_full()` se charge de parcourir le bloc du répertoire courant, et de compter les entrées dans ce bloc. Les inodes utilisées étant rangées de façon contigu dans le bloc, on s'arrête lorsque une inode = 0. Retour booléen indiquant si le répertoire est plein (128 fichiers) ou pas.

Stratégie de suppression

Détection du plus vieux (date de dernière modification) ou du plus gros dans la partition

Nous avons créé 2 fonctions `oldest_in_partition()`, `biggest_in_partition()`, qui parcourent les inodes de toute la partition grâce au bitmap en faisant appel à la fonction `find_next_zero_bit()` nous permettant ainsi d'analyser la date de dernière modification/taille uniquement des inodes utilisées. Elle renvoie le numéro d'inode.

Après cela, soit une `structure dentry` pointant sur l'inode existe en cache, et on connaît directement le répertoire à nettoyer, soit il faut rechercher dans quel répertoire est localisé le fichier à supprimer.

Localisation du fichier trouvé et de son parent

Nous avons créé la fonction `find_parent_of_ino()` qui renvoi l'inode du parent. Pour ce faire, nous parcourons de façon récursive toute l'arborescence jusqu'à trouvé l'inode recherchée. Une fois trouvée, nous savons dans quel bloc elle est, et donc l'inode de son parent.

Détection du plus vieux ou du plus gros dans le répertoire courant

Nous avons créé 2 fonction `biggest_in_dir()`, `find_oldest_in_dir()` ` qui parcourent tous le bloc de l'inode du répertoire courant, pour y trouver le plus gros ou le plus ancien. Elle retourne le numéro d'inode concernée.

Cas particulier du fichier déjà ouvert par un autre processus

Comme une inode à son compteur de référence incrémenté à chaque fois qu'une structure pointe dessus, ce compteur est à 1 juste après la création, car une structure dentry en cache pointe toujours sur l'inode. Il a fallu faire donc une vérification judicieuse du compteur de référence de l'inode en prenant en compte cette problématique en vérifiant si un dentry était présent ou non.

2.1 Politique de suppression de fichiers

2.2 Interaction user / fs