

Rapport Tp3

Dans ce TP, le but était de reproduire en bas niveau le programme *rsync*, qui est un programme utilisé pour faire des backups à travers le réseau. En particulier, nous avons dû reproduire la partie de *rsync* qui permet de lister les fichiers d'une manière récursive et de copier le contenu dans un autre dossier toujours de manière récursive.

Nous avons donc séparé notre code en 2 module: un module `listing.c`, qui va lister le contenu d'un dossier d'une manière très similaire à la commande "ls -l". Ensuite, le deuxième module (`copy.c`), allait nous permettre de faire une copie du contenu du dossier dans un autre dossier, qui pouvait être existant ou à créer. Enfin le `main.c` allait rassembler nos fonctions principales pour que le programme puisse agir en fonction de ce que l'utilisateur souhaite obtenir.

Dans le premier module, `listing.c`, notre fichier `.c` est divisé en deux fonction:

- La première fonction va nous permettre de récupérer, pour chaque dossier ou fichier présent dans le dossier demandé, les informations du fichier (droits, taille, date d'accès, chemin). Pour se faire, on a utilisé les inodes, qui sont des structures de données contenant des informations sur le fichier voulu (ex: l'inode `ctime` va nous donner la date de la dernière modification);
- La deuxième fonction est une méthode qui va nous permettre de lire les droits de fichier. Nous avons utilisé une deuxième fonction pour éviter de surcharger la première et donc un simple appel de fonction dans la première fonction va nous permettre d'utiliser la deuxième à l'endroit souhaité.

Enfin, pour vérifier quel type de fichier et les différents droits selon si l'on est un utilisateur, si on fait partie d'un groupe ou si l'on est un root, on va toujours utiliser les inodes pour déterminer les permissions d'un inode et donc connaître les permissions des utilisateurs et quel type de fichier on est train de lire (fichier, dossier ou lien symbolique)

Dans le deuxième module, `copy.c`, on a une seule fonction qui va nous permettre en quelque sorte de faire un “backup” du dossier demande. En effet, ce module va copier le contenu du dossier demande dans un autre dossier, qui peut être inexistant a la base mais qui va pouvoir être créé par notre module.

Question: dans quel cas les droits ne pourront pas être les mêmes ?

Les droits ne pourront pas être les mêmes dans le cas où notre processus va essayer d’agir sur un fichier ou dossier sur lequel le fichier/dossier a des droits plus élevés que l’utilisateur qui a lancé le processus. Dans ce cas, notre processus va tout simplement s’arrêter en renvoyant un code d’erreur.