Fichiers II : système de fichiers

Fichiers : plan

On va observer en 3 niveaux

- arborescence

  `ls -ilah, cd, cp -lar, rm -rf, mv, dd, tar -cxzvf, pwd -P, mkdir -p, rmdir, ln -s, readlink, file, du, find, cat, touch`

- systèmes de fichiers

  `mkfs -t, fsck, mount, umount, findmnt, stat, df, sync`

- périphériques bloc (disque, partition, boucle, chiffrement, lvm, raid)

  `(g)parted, lsblk, cryptsetup, lvm, mdadm, losetup, dmsetup`

Système de fichiers

Montage

L'arborescence est un patchwork de systèmes de fichiers montés les uns sur les autres.

À la base n'existe que le répertoire racine `/`. Les systèmes de fichiers se montent (en: mount) sur des répertoires existants, et offrent à leur tour des répertoires, sur lesquels on peut monter de nouveaux systèmes de fichiers de façon récursive.

Démo, avec une clef USB et des fs exotiques (`sshfs` et `wikirofs`).

```
findmnt
mount
umount
/etc/fstab
```

Il n'est pas possible de démonter un système de fichier en cours d'utilisation, pour pas rester coincé·e :

```
$ man lsof
$ man fuser
```

Système de fichiers sur périphériques en mode bloc

Désormais, nous nous intéréssons aux systèmes de fichiers qui s'installent dans les péripheriques en mode bloc.

Comment organiser une suite de blocs d'un périphérique de sorte à pouvoir y stocker et manipuler des répertoires, des fichiers, etc ?

`mkfs`
`fsck`

Inode

- Un fichier = un inode.
- Un inode contient les métadonnées d'un fichier (type de fichier, user propriétaire, permissions, etc) et des pointeurs vers le contenu du fichier.
- Le numéro d'inode est l'identifiant unique d'un fichier.
- Les fichiers n'ont pas de nom, les répertoires nomment les fichiers, un fichier peut être nommé plusieurs fois dans l'arborescence des répertoires.
- Le contenu d'un fichier est interprété différemment selon son type.

Pour voir les informations contenues dans un inode :

```
$ stat
```

À quoi ça ressemble un fs (exemple de `ext2`) ? Comment agissent les commandes `cp`, `cp -l` = `ln`, `ln -s`, `mv`, `rm` au niveau du ou des filesystems en jeu.

-> Tableau !

Inode

Attributs stockés dans un inode (penser à `ls -l`) :

- type du fichier (-, d, l, p, s, b, c)
- droits d'accès (12 bits)
- nombre de liens (physiques) sur cet inode (`Nlinks`)
- numéro de l'utilisateur propriétaire (`uid`)
- numéro de groupe propriétaire (`gid`)
- taille du fichier
- horodatages (en: timestamps)
  - date/time dernier accès en lecture (`atime` = access time)
  - date/time création du fichier (`btime` = birth time)
  - date/time dernière modification de l'inode (`ctime` = change time)
  - date/time derniere modification du contenu (`mtime` = modify time)
- des pointeurs vers les blocs de données (pour les gros fichiers, certains de ces blocs peuvent à leur tour contenir des pointeurs vers des blocs de données, etc)
- (éventuellement des attributs étendus (`xattr`) : ACL, selinux, ...)

Voir : http://web.mit.edu/tytso/www/linux/ext2intro.html

Interprétation du contenu d'un fichier selon son type

Fichier régulier (fichier de type `-`)

- son contenu est le contenu du fichier (pas d'autre information)

Répertoire (fichier de type `d`) :

- son contenu est une liste de paires `(nom de fichier, inode correspondant)`. Une telle paire est appelé lien (en: link) ou lien physique (en: hardlink)
- En particulier, le nom de fichier est une info du répertoire, et n'est pas contenu dans l'inode du fichier.
- Les fichiers n'ont pas de nom, ce sont les répertoires qui leur donnent un (ou plusieurs) nom(s).
- Un même fichier (identifié par son inode) peut apparaître dans plusieurs répertoires et sous plusieurs noms.

Lien symbolique (fichier de type `l`)

- son contenu est le chemin vers lequel il pointe