Systèmes d'exploitation, 2ème année

Système de gestion des fichiers - Partie intermédiaire

Yves Stadler

Université Paul Verlaine - Metz

8 décembre 2011

Agenda

Plan du cours

- Arborescence des fichiers
- Liaison decripteur de fichier avec fichier
- Montages

Liaison avec la partie symbolique

Organisation connue

- Nos fichiers sont utilisables grâce aux descripteurs de fichiers
- Nous n'avons toujours pas lié nom de fichier avec le fichier réel
- Les fichiers sont organisés en arborescence
- Répertoire : collections de fichiers (dont répertoires)

Organisation sur le dique

- Sous unix un fichier est associé à un numéro *i-node*
- La commande `ls -i` affiche les *i-nodes*
- On verra qu'il existe des tables d'*i-nodes*

Modifications de l'arborescence

Liens

- On peut créer des liens dans l'arborescence
- Liens symboliques (un nom associé à un autre)
- Liens matériels (plusieurs nom pour une donnée)

Montage

- Accrocher une arborescence dans une autre arborescence
- Abstraction de partitions
- Raccourcis, accès confinés.

Modifications de l'arborescence

Liens symboliques - Soft links

- Peut pointer sur quelque-chose d'inexistant
- Modifier le contenu du lien revient à modifier le fichier
- Enlever le lien ne détruit pas le fichier
- *i-nodes* différents.

Commande

- ln -s <cible> [<destination>]
- par défaut, répertoire en cours même nom

Modifications de l'arborescence

Liens matériels - hard links

- Doit pointer sur un fichier existant
- Modifier le contenu d'un lien revient à modifier le fichier
- Enlever le lien peut détruire le fichiers
- Notion de nombre de liens pointant sur le fichier
- Un *i-node* unique.

Commande

- ln <cible> [<destination>]
- par défaut, répertoire en cours même nom

Modifications de l'arborescence

Liaison entre abstraction et i-node

- Une entrée de la table des fichiers ouverts pointe vers un *i-node*
- Elle contient aussi :
  - Le mode d'ouverture
  - L'offset

Fonctions systèmes en C

Création, effacement

- `link(2)`
- `unlink(2)`
- `symlink(2)`
- `rename(2)`

Répertoires

Contenu

- Les répertoires contiennent des fichiers
- En fait, il s'agit d'une liste de (nom, *i-nodes*)

Fonctions systèmes en C

- `mkdir(2)`
- `rmdir(2)`

Répertoires

Manipulation des répertoires

- `opendir(3)`
- `readdir(3)`
- `rewinddir(3)`
- `closedir(3)`

Usage

- Une entrée permet de connaître : *i-node*, longueur, type, nom et décalage pour l'entrée suivante
- On rappelle la fonction read pour accéder à l'entrée suivante
- Ces fonctions ne sont pas des appels systèmes. C'est une surcouche provenatn d'une bibliothèque.
- Les appels systèmes sous-jacents sont `open(2)`,`read(2)`,`write(2)`,`close(2)`

Montages

Concept

- Les arboresences peuvent être distinguées en fonctions des supports (windows : un lecteur par disque, clef USB, ...)
- UNIX introduit la notion de montage, qui efface cette notion de support
- Un montage est l'association d'un répertoire à une arborescence
- Le format de l'arborescence montée peut être différente de celui de l'hôte.
- On ne peut monter que des volumes.

Utilisation

- `mount(8)` montage par ligne de commande
- `mount(2)` montage par programme.

Montages

Montages automatiques

- Au démarrage de linux, certains montages peuvent être configurés pour être automatisés.
- `/etc/fsatb`

Exemple

```
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    nodev,noexec,nosuid 0       0
# / was on /dev/sda4 during installation
UUID=1d0c6490-24c8-4af6-a37c-9605c7546774 / ext4 errors=remount-ro 0 1
```