

LS4 100414

Rappels systeme

SGF : Systeme de Gestion de Fichier

Les répertoires organisent le systeme des fichiers en arborescence où :

- les noeuds de l'arbre sont des i-noeuds (inode, noeud d'index) des fichiers
- les aretes de l'arbre ont des liens sur ces noeuds

Les noeuds

Un inoeud est associé à un fichier et contient les informations suivantes :

- un numero renseignant sur la zone mémoire où s trouve le fichier
- les droits d'accès
- le type de fichier
- le nomvre de liens sur le fichier
- les dates -

schema d'une arborescence

```
| - A/  
| ---> .  
| ---> ..  
| --- B/  
| ----> .  
| ----> ..  
| ----> fic1  
| ----> fic2  
| --- C/  
| ----> .  
| ----> ..  
| ----> fic3  
| ---- D/  
| -----> .  
| -----> ..  
| -----> fic4
```

Les liens

Chaque répertoire a **au moins** 2 liens

- un lien vers le père de nom '..'
- un lien vers lui meme de nom '.'

Sur chaque fichier non repertoire pointe au moins un lien **physique**, mais il peut y en avoir plus (voir la *commande ln*)

Les droits

Le droit en **lecture** sur

- un repertoire permet de lister le contenu du repertoire
- un fichier non repertoire permet de lire le contenu du fichier

Le droit en **écriture** sur :

- un repertoire permet de modifier le contenu du repertoire (créer, effacer)
- un fichier non repertoire permet de modifier le contenu du fichier

Le droit en **exécution** sur

- un repertoir donne le droit de "traverser"
- un fichier non repertoire de le rendre executable

Les processus

Tout programme est exécuté par un processus qui est créé au lancement du programme et meurt avec la fin du programme.

Le premier processus est lancé à l'ouverture de votre ordinateur. C'est le processus ancêtre de tous les autres processus. On obtient donc une arborescence des processus.

Chaque processus est identifié par un numéro. Le numéro permet à l'utilisateur d'interagir avec le processus en lui envoyant des signaux (ex : SIGSTOP, SIGKILL, SIGINT...)

Les redirections

On peut rediriger l'entrée standard sur une commande :

```
$cmd < entree
```

La sortie standard :

```
$cmd > sortie  
$cmd 1> sortie
```

rappel que sortie est un nom de fichier

sortie standard erreur

```
$cmd 2>sortie_err
```

Exemple :

```
ls > listing.txt  
$ls toto > listing.txt
```

si le fichier toto n'existe pas le message d'erreur s'affiche sur la sortie standard

```
$cmd1 | cmd2 | cmd3
```

3 processus fils du shell sont lancés en parallèle, chacun gérant une commande. De plus ces processus communiquent par tube

Le 1er tube recupere la sortie de cmd1 et l'ecrit sur l'entrée de la cmd2 Le 2em tube recupere la sortie de cmd2 et l'ecrit sur l'entrée de la cmd3