

CH.4 SYSTÈME UNIX

SYSTÈMES DE FICHIERS

- 4.1 Quelques généralités sur UNIX
- 4.2 L'arborescence des fichiers : ls
- 4.3 La structure : les i-noeuds : ls -li
- 4.4 Les commandes utiles : cp, mv, ln, rm
- 4.5 Les liens symboliques et le montage
- 4.6 Les droits : chmod

Info S4 ch4 1

4.1 Quelques généralités sur UNIX

Système créé en 1969 aux Bell labs pour permettre la gestion multi-utilisateurs d'un mini-ordinateur.

La même équipe crée le langage C pour permettre le développement du système.

Le système UNIX est récrit en C en 1970.

Succès pour ces deux produits :

UNIX utilisé dans de nombreuses versions sur tous matériels ;

C utilisé comme langage de programmation de développement.

Le langage C est de plus en plus utilisé comme un langage d'assemblage évolué.

Le développement de normes (et le succès de Linux) diminuent les disparités dans les versions d'UNIX.

Info S4 ch4 2

Caractéristiques principales d'UNIX :

- Système multi-utilisateurs et multi-tâches ;
- Système de fichiers hiérarchisé cohérent ;
- Traitement uniforme des fichiers et des périphériques ;
- La notion de processus est dégagée, assurant la grande sûreté de fonctionnement ;
- Système ouvert et portable car écrit en C ;
- Accès possible aux couches matérielles ;
- Interpénétration du système et du langage C : appels-système intégrables dans un programme ;
- Disponibilité de nombreux outils de développement ;
- Commandes standard simples et langage de commandes simple et flexible, traité comme un processus en soi ;
- Développement d'interfaces graphiques conviviales (X-Window, KDE).

Info S4 ch4 3

4.2 L'arborescence des fichiers : ls

Exemple de session :

```
[desar@riesling fichiers]$ pwd
/home/institut/desar/fichiers
[desar@riesling fichiers]$ ls --help
Usage: ls [OPTION]... [FICHIER]...
Afficher les informations au sujet des FICHIERS (du répertoire
courant par défaut). Trier les entrées alphabétiquement si aucune
des options -cftuSUX ou --sort n'est utilisée.
```

.....

-a, --all	afficher les noms cachés débutant par .
-i, --inode	afficher le numéro d'index de chaque fichier
-l	utiliser le format long d'affichage
-R, --recursive	afficher les sous-répertoires récursivement

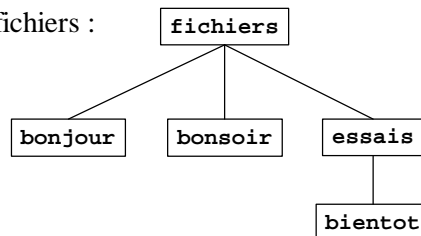
Info S4 ch4 4

```
[desar@riesling fichiers]$ ls
bonjour  bonsoir  essais
[desar@riesling fichiers]$ ls -l
total 3
-rw-r--r--  1 desar  institut  10 mar 20 15:57 bonjour
-rw-r--r--  1 desar  institut  10 mar 20 15:58 bonsoir
drwxr-xr-x  2 desar  institut 1024 mar 20 15:58 essais
[desar@riesling fichiers]$ ls -lR
total 3
-rw-r--r--  1 desar  institut  10 mar 20 15:57 bonjour
-rw-r--r--  1 desar  institut  10 mar 20 15:58 bonsoir
drwxr-xr-x  2 desar  institut 1024 mar 20 15:58 essais
```

```
essais:
total 1
-rw-r--r--  1 desar  institut  12 mar 20 15:58 bientot
```

Info S4 ch4 5

Exemple de hiérarchie de fichiers :



Arborescence constituée de répertoires (directories) et de fichiers. Les noeuds intérieurs sont des répertoires. Les feuilles sont en général des fichiers (ou des répertoires vides). Les fichiers sont des fichiers de texte, des fichiers binaires, des fichiers spéciaux (périphériques) ou des liens symboliques. La racine est /.

La navigation se fait avec la commande `cd`. L'utilisateur est dans un répertoire. Des pseudonymes désignent divers répertoires :

- Celui où on se trouve est `.`
- Son père est `..`

Info S4 ch4 6

Exemple de navigation :

```
[desar@riesling fichiers]$ cd essais
[desar@riesling essais]$ pwd
/home/institut/desar/fichiers/essais
[desar@riesling essais]$ cd ../../
[desar@riesling desar]$ pwd
/home/institut/desar
[desar@riesling desar]$ cd /
[desar@riesling /]$ cd ~
[desar@riesling desar]$ pwd
/home/institut/desar
```

Exemple de visualisation du contenu de fichiers-texte : cat et more

```
[desar@riesling fichiers]$ cat bonjour
Bonjour !
[desar@riesling fichiers]$ more bonsoir
Bonsoir !
```

Info S4 ch4 7

4.3 La structure : les i-noeuds : ls -i

On peut pratiquer des fonctions de base d'UNIX sans connaître plus de détails sur l'organisation des fichiers.

Une certaine compréhension permet néanmoins une plus grande efficacité de programmation et explique des particularités des commandes.

Les fichiers contiennent des *données*, stockées sur un *support physique* (disque). Ces données sont accessibles à travers des *noms* de fichiers.

Pour permettre une manipulation plus facile des données et des noms de fichiers, ainsi qu'une structure uniforme du système de fichiers, les données sont repérées de façon interne non pas par un nom, mais par un numéro, le *i-noeud*, indépendant de l'adresse réelle des données sur le disque.

Ce i-noeud assure une interface entre l'utilisateur et le système physique de fichiers.

L'option i de la commande ls permet de consulter les numéros de i-noeuds.

Info S4 ch4 8

Exemple d'affichage des i-noeuds :

```
[desar@riesling fichiers]$ ls -liR
total 3
 213235 -rw-r--r--   1 desari institut   10 mar 20 15:57 bonjour
 213237 -rw-r--r--   1 desari institut   10 mar 20 15:58 bonsoir
 213236 drwxr-xr-x   2 desari institut 1024 mar 20 15:58 essais

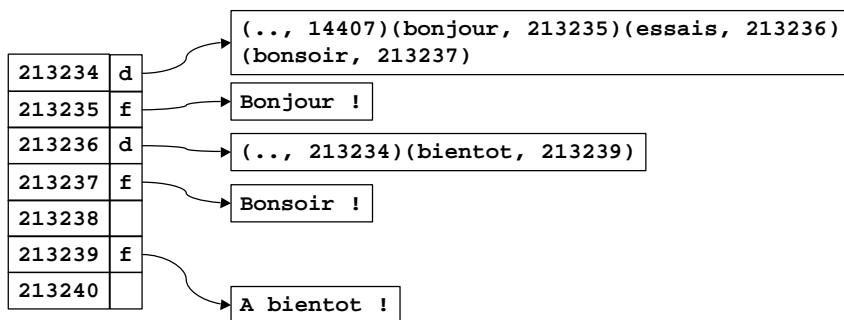
essais:
total 1
 213239 -rw-r--r--   1 desari institut   12 mar 20 15:58 bientot
```

La structure est la suivante :

- Une table des i-noeuds contient pour chaque numéro un pointeur vers l'information stockée ; elle contient aussi l'information sur le type de fichier, les droits, le propriétaire et le groupe, le compteur de liens, la date de modification, ...
- L'information relative aux répertoires contient, pour chacun des éléments qu'il contient, le numéro du i-noeud correspondant, et celui de son père.

Info S4 ch4 9

Exemple de table de i-noeuds :



De cette manière, il est possible de changer l'information sans changer le nom, ou de changer le nom sans modifier l'information, de créer, de supprimer, de renommer, de lier fichiers et répertoires.

Info S4 ch4 10

4.4 Les commandes utiles : cp, mv, ln, rm

Les trois premières ont une syntaxe commune : on peut copier, déplacer ou lier un fichier sur un autre fichier ou un ou plusieurs fichiers dans un répertoire.

Recopie le fichier bonjour en bonsoir ; l'information est dupliquée.

Le fichier bonjour existe encore.

```
[desar@riesling fichiers]$ cp bonjour salut
[desar@riesling fichiers]$ cat salut
Bonjour !
```

Lie le fichier bonsoir sur adieu : ce sont deux accès distincts à *la même* information.

```
[desar@riesling fichiers]$ ln bonsoir adieu
[desar@riesling fichiers]$ cat adieu
Bonsoir !
```

Crée un lien bonsoir dans le répertoire essais.

```
[desar@riesling fichiers]$ ln bonsoir essais
```

Renomme bonsoir en bonsoir_n.

```
[desar@riesling fichiers]$ mv bonsoir bonsoir_n
```

Info S4 ch4 11

État du système de fichiers :

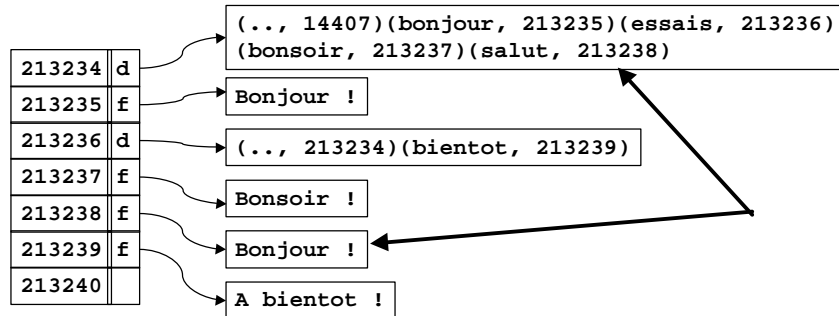
```
[desar@riesling fichiers]$ ls -liR
total 5
213237 -rw-r--r--  3 desar  institut    10 mar 20 15:58 adieu
213235 -rw-r--r--  1 desar  institut    10 mar 20 15:57 bonjour
213237 -rw-r--r--  3 desar  institut    10 mar 20 15:58 bonsoir_n
213236 drwxr-xr-x  2 desar  institut   1024 mar 20 16:18 essais
213238 -rw-r--r--  1 desar  institut    10 mar 20 16:18 salut

essais:
total 2
213239 -rw-r--r--  1 desar  institut    12 mar 20 15:58 bientot
213237 -rw-r--r--  3 desar  institut    10 mar 20 15:58 bonsoir
```

Effet des opérations sur la structure des i-noeuds :

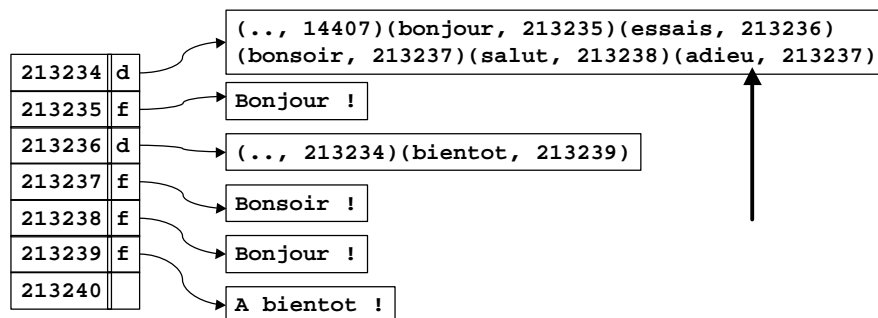
Info S4 ch4 12

```
[desar@riesling fichiers]$ cp bonjour salut
```



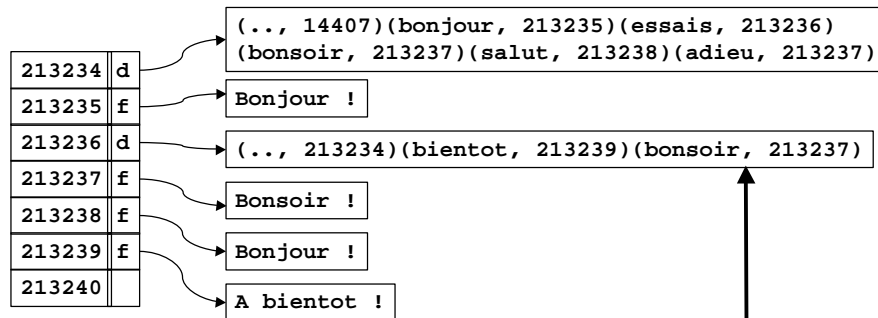
Info S4 ch4 13

```
[desar@riesling fichiers]$ ln bonsoir adieu
```



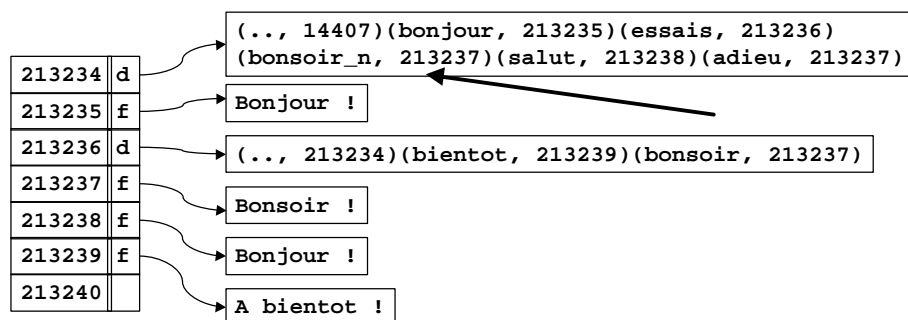
Info S4 ch4 14

```
[desar@riesling fichiers]$ ln bonsoir essais
```



Info S4 ch4 15

```
[desar@riesling fichiers]$ mv bonsoir bonsoir_n
```



Info S4 ch4 16

La commande rm permet d'effacer. En fait, l'opération est plus subtile :

- La référence nom-i-noeud est effacée ;
- Le compteur de liens du i-noeud est diminué d'une unité ;
- Si ce compteur vaut 0, le i-noeud et l'espace mémoire vers lequel il pointe sont récupérés.

```
[desar@riesling fichiers]$ rm --help
```

```
Usage: rm [OPTION]... FICHIER
```

```
Enlever (unlink) les FICHIER(s).
```

```
-i, --interactive    demander une confirmation avant chaque destruction  
-r, -R, --recursive enlever le contenu des répertoires récursivement
```

Attention à l'option récursive : il n'y a aucun recours !

Les commandes mkdir et rmdir sont utilisées pour créer un répertoire et effacer un répertoire vide. S'il n'est pas vide, utiliser rm -r (attention !) ou le vider d'abord.

Info S4 ch4 17

```
[desar@riesling fichiers]$ ls -liR
```

```
total 5
```

```
213237 -rw-r--r--  3 desar  institut    10 mar 20 15:58 adieu  
213235 -rw-r--r--  1 desar  institut    10 mar 20 15:57 bonjour  
213237 -rw-r--r--  3 desar  institut    10 mar 20 15:58 bonsoir_n  
213236 drwxr-xr-x  2 desar  institut   1024 mar 20 16:18 essais  
213238 -rw-r--r--  1 desar  institut    10 mar 20 16:18 salut
```

```
essais:
```

```
total 2
```

```
213239 -rw-r--r--  1 desar  institut    12 mar 20 15:58 bientot  
213237 -rw-r--r--  3 desar  institut    10 mar 20 15:58 bonsoir
```

```
[desar@riesling fichiers]$ rm adieu
```

```
[desar@riesling fichiers]$ ls -liR
```

```
total 4
```

```
213235 -rw-r--r--  1 desar  institut    10 mar 20 15:57 bonjour  
213237 -rw-r--r--  2 desar  institut    10 mar 20 15:58 bonsoir_n  
213236 drwxr-xr-x  2 desar  institut   1024 mar 20 16:18 essais  
213238 -rw-r--r--  1 desar  institut    10 mar 20 16:18 salut
```

```
essais:
```

```
total 2
```

```
213239 -rw-r--r--  1 desar  institut    12 mar 20 15:58 bientot  
213237 -rw-r--r--  2 desar  institut    10 mar 20 15:58 bonsoir
```

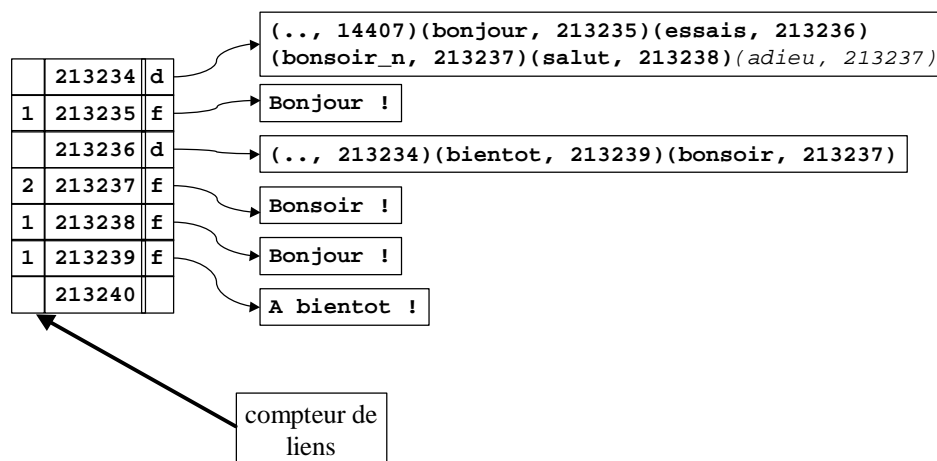
Info S4 ch4 18

```
[desar@riesling fichiers]$ rm essais/bientot
[desar@riesling fichiers]$ ls -liR
total 4
 213235 -rw-r--r--  1 desar    institut    10 mar 20 15:57 bonjour
 213237 -rw-r--r--  2 desar    institut    10 mar 20 15:58 bonsoir_n
 213236 drwxr-xr-x  2 desar    institut   1024 mar 20 16:18 essais
 213238 -rw-r--r--  1 desar    institut    10 mar 20 16:18 salut

essais:
total 1
 213237 -rw-r--r--  2 desar    institut    10 mar 20 15:58 bonsoir
```

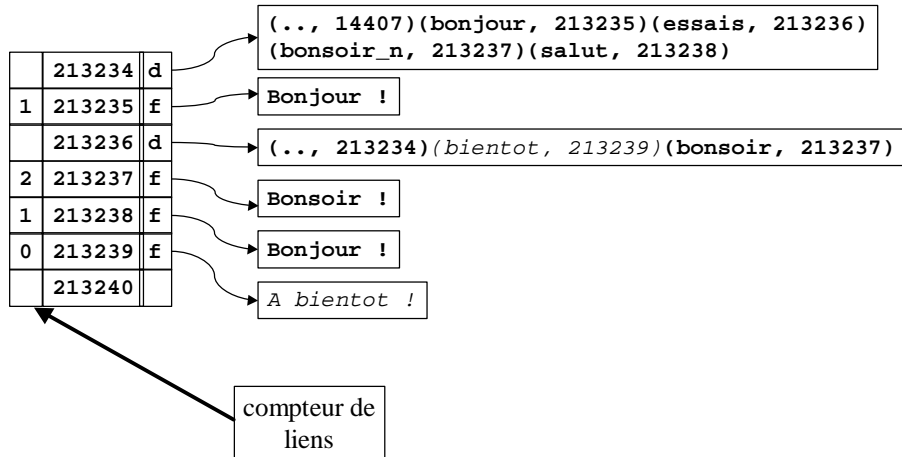
Info S4 ch4 19

```
[desar@riesling fichiers]$ rm adieu
```



Info S4 ch4 20

```
[desar@riesling fichiers]$ rm essais/bientot
```



Info S4 ch4 21

4.1 Les liens symboliques et le montage

Les liens ordinaires sont une façon de créer plusieurs références au même objet. Une fois créées, ces références sont rigoureusement équivalentes.

Ces liens ne sont applicables qu'aux fichiers (non aux répertoires).

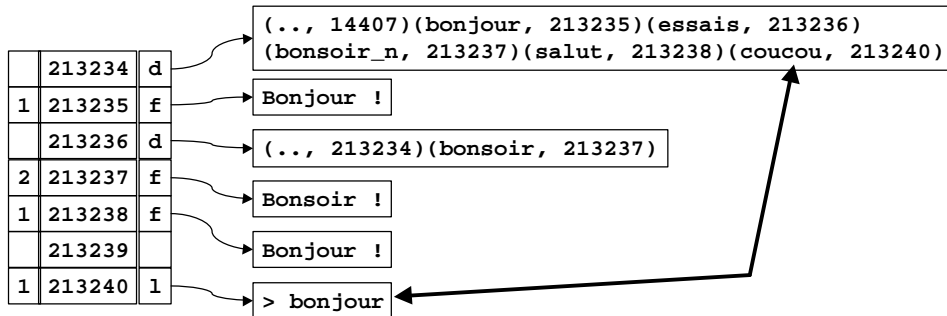
Il existe un autre système de liens dits *liens symboliques*, obtenus par la commande `ln -s`. Le résultat obtenu ressemble aux alias de Windows.

Un lien symbolique est un nouveau fichier qui contient l'adresse à laquelle renvoie ce fichier (cela peut être un répertoire).

```
[desar@riesling fichiers]$ ln -s bonjour coucou
[desar@riesling fichiers]$ ls -li
total 4
213235 -rw-r--r-- 1 desar institut 10 mar 20 15:57 bonjour
213237 -rw-r--r-- 2 desar institut 10 mar 20 15:58 bonsoir_n
213236 drwxr-xr-x 2 desar institut 1024 mar 20 16:18 essais
213240 lrwxrwxrwx 1 desar institut 7 mar 20 16:23 coucou ->
bonjour
213238 -rw-r--r-- 1 desar institut 10 mar 20 16:18 salut
```

Info S4 ch4 22

```
[desar@riesling fichiers]$ ln -s coucou bonjour
```



Info S4 ch4 23

Le nom coucou renvoie à bonjour.

```
[desar@riesling fichiers]$ cat coucou
Bonjour !
```

Mais il n'existe aucune garantie d'intégrité :

```
[desar@riesling fichiers]$ rm bonjour
[desar@riesling fichiers]$ ls -li
total 3
213237 -rw-r--r--  2 desar    institut    10 mar 20 15:58 bonsoir
213240 lrwxrwxrwx  1 desar    institut     7 mar 20 16:23 coucou ->
                                bonjour
213236 drwxr-xr-x  2 desar    institut   1024 mar 20 16:18 essais
213238 -rw-r--r--  1 desar    institut    10 mar 20 16:18 salut
[desar@riesling fichiers]$ cat coucou
cat: coucou: Aucun fichier ou répertoire de ce type.
```

Les commandes cp, ln, mv, rm agissent sur le lien.

Info S4 ch4 24

On peut faire un lien symbolique sur un répertoire :

```
[desar@riesling fichiers]$ rm coucou
[desar@riesling fichiers]$ ls -li
total 3
 213237 -rw-r--r--  2 desar    institut    10 mar 20 15:58 bonsoir
 213236 drwxr-xr-x  2 desar    institut   1024 mar 20 16:18 essais
 213238 -rw-r--r--  1 desar    institut    10 mar 20 16:18 salut
[desar@riesling fichiers]$ ln -s essais nouveau
[desar@riesling fichiers]$ ls -li
total 3
 213237 -rw-r--r--  2 desar    institut    10 mar 20 15:58 bonsoir
 213236 drwxr-xr-x  2 desar    institut   1024 mar 20 16:18 essais
 213240 lrwxrwxrwx  1 desar    institut     7 mar 20 16:35 nouveau ->
                                essais
 213238 -rw-r--r--  1 desar    institut    10 mar 20 16:18 salut
[desar@riesling fichiers]$ cd nouveau
[desar@riesling nouveau]$ pwd
/home/institut/desar/fichiers/nouveau
[desar@riesling nouveau]$ ls -li
total 1
 213237 -rw-r--r--  2 desar    institut    10 mar 20 15:58 bonsoir
```

Info S4 ch4 25

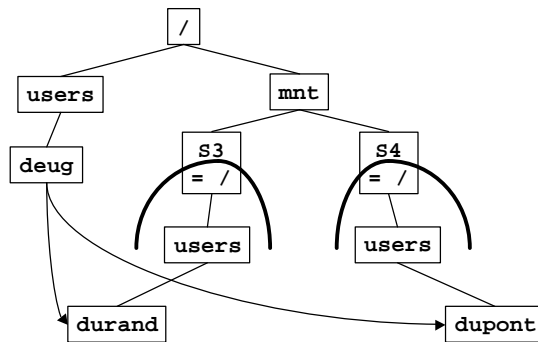
Les systèmes de fichiers différents peuvent être assemblés en un seul :
disques physiques, disquettes, CD-rom, disques amovibles (zip).
Chacun est un système de fichiers complet, avec une table de i-noeuds par
disque. L'assemblage est fait par le superutilisateur avec la commande
mount, qui attache la racine d'un système de fichiers en un point de
l'arborescence d'un autre système.

Les liens ordinaires ne permettent pas de passer d'un système de fichiers
à un autre monté (pourquoi ?). Les liens symboliques le permettent de
façon transparente pour l'utilisateur.

Voici un exemple où l'utilisateur Durand a ses fichiers sur le disque
S3 et l'utilisateur Dupont a les siens sur le disque S4.

Les disques externes S3 et S4 sont montés et des liens symboliques renvoient
sur les répertoires de base de Durand et de Dupont.

Info S4 ch4 26



Il suffit de créer deux liens :

```
ln -s /mnt/S3/users/durand /users/deug/durand
ln -s /mnt/S4/users/dupont /users/deug/dupont
```

si bien que Durand se retrouve chez lui par `cd /users/deug/durand`
et Dupont par `cd /users/deug/dupont`

Info S4 ch4 27

4.6 Les droits : chmod

Revenons à la situation suivante :

```
[desar@riesling fichiers]$ ln bonsoir adieu
[desar@riesling fichiers]$ ls -li
total 4
213237 -rw-r--r--  3 desar    institut    10 mar 20 15:58 adieu
213238 -rw-r--r--  1 desar    institut    10 mar 20 16:18 bonjour
213237 -rw-r--r--  3 desar    institut    10 mar 20 15:58 bonsoir
213236 drwxr-xr-x  2 desar    institut   1024 mar 20 16:18 essais
```

Le premier caractère identifie le type de fichier, puis viennent deux groupes de trois `rw`, certains remplacés par un trait.

Le premier se réfère au propriétaire `u`, le second au groupe `g`, le troisième aux autres `o`.

Pour un fichier, `r` autorise la lecture, `w` l'écriture et `x` l'exécution.

Pour un répertoire, `r` autorise la lecture de la liste des fichiers contenus, `w` la modification de cette liste et `x` l'accès aux fichiers contenus.

Info S4 ch4 28

Le propriétaire et *lui seul* (et le superutilisateur) peut modifier ces droits par la commande `chmod`.

```
[desar@riesling fichiers]$ chmod --help
Usage: chmod [OPTION]... MODE[,MODE]... FICHIER...
       ou: chmod [OPTION]... MODE_OCTAL FICHIER...
```

Usuellement, la syntaxe est `chmod [ugoa][+--][rwx] FICHIER`.

La commande `chmod a-w FICHIER` protège en écriture.

La commande `chmod u+x FICHIER` rend un fichier exécutable.

Les droits sont attachés au i-noeud.

```
[desar@riesling fichiers]$ chmod a-w adieu
[desar@riesling fichiers]$ ls -li
total 4
213237 -r--r--r--  3 desar    institut    10 mar 20 15:58 adieu
213238 -rw-r--r--  1 desar    institut    10 mar 20 16:18 bonjour
213237 -r--r--r--  3 desar    institut    10 mar 20 15:58 bonsoir
213236 drwxr-xr-x  2 desar    institut   1024 mar 20 16:18 essais
```

Info S4 ch4 29

Par défaut, seul le propriétaire peut modifier un de ses fichiers ou la liste des fichiers contenus dans un répertoire. Mais tous peuvent les lire.

La commande `chmod go-x REPERTOIRE` empêche les tiers d'accéder aux fichiers contenus dans un répertoire ; ils peuvent toujours en consulter la liste.

La commande `chmod go-r REPERTOIRE` empêche les tiers de consulter la liste des fichiers contenus dans un répertoire ; mais ils peuvent y accéder, s'ils en connaissent l'adresse (si le droit `x` est activé).

Enfin, la commande `chmod go-rwx REPERTOIRE` assure la confidentialité complète vis-à-vis des tiers.

Info S4 ch4 30