

# 3 - Système de Fichiers

Laurent Tichit

12 avril 2011

# Plan

- 1 Fichiers et répertoires
- 2 Structure du système de fichiers
- 3 Répertoires spéciaux
- 4 Commandes
- 5 Chemins

# Fichiers et répertoires

# Les fichiers

- Unix gère des **fichiers** : toute structure accessible par le système d'exploitation sera considérée comme un fichier.
- Les fichiers standards :
  - ▶ Programmes source (texte)
  - ▶ Programmes exécutables (binaires)
  - ▶ Bibliothèques (binaires)
  - ▶ Fichiers de données (texte, images, sons, documentation, ...)
- Les fichiers spéciaux (pseudo-fichiers) :
  - ▶ répertoires (dossiers)
  - ▶ liens symboliques (raccourcis)
  - ▶ fichiers périphériques, terminaux, réseaux ( /dev/xxx)

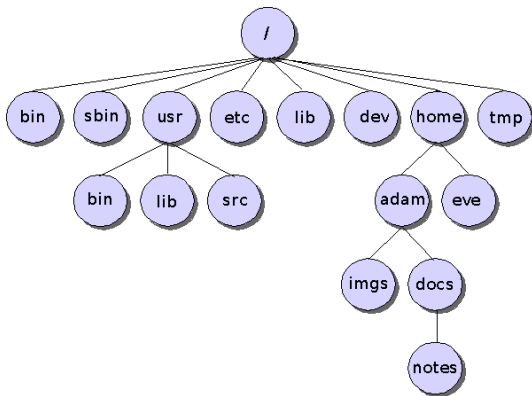
# Les fichiers et répertoires

- Un fichier permet :
  - ▶ De conserver de l'information
  - ▶ De traiter de l'information
- La quantité d'information contraint les systèmes d'exploitation à la structurer.
- La solution est de constituer des containers hiérarchisés : les répertoires.
- Un répertoire peut-être vu comme une boîte contenant des fichiers (dont certains peuvent être des répertoires).

# Structure du système de fichiers

# Structure du système de fichiers

- La hiérarchie des fichiers est représentée sous forme d'arborescence.



- Le repertoire `/` est celui qui englobe tous les autres. Il s'agit de la **racine**.
- Le chemin absolu du fichier `notes` est `/home/adam/docs/notes`

# Répertoires spéciaux



# Le répertoire de travail

- Le répertoire courant ou **répertoire de travail** (working directory) est “celui où l'on se trouve”.
- Se déplacer dans le système de fichier revient à changer de répertoire courant.
- On utilise la commande : **cd nom-de-répertoire** (pour *change directory*).
- Pour connaître le chemin absolu du répertoire de travail : **pwd** (pour *path working directory*)  
tichit@iml230:~\$ **cd Test**  
tichit@iml230:~/Test\$ **pwd**  
/home/tichit/Test

Attention : l'accès à un fichier ne peut s'effectuer que vers un fichier visible à partir du répertoire de travail.

# Le répertoire d'accueil

- Chaque usager dispose d'un **répertoire d'accueil** (*home directory*) qui lui appartient.
- Le nom de ce répertoire est contenu dans la variable **HOME**.
- La commande **cd** permet de revenir dans son répertoire d'accueil.  
tichit@iml230:~\$ **cd /etc**  
tichit@iml230:/etc\$ **pwd**  
/etc  
tichit@iml230:/etc\$ **cd**  
tichit@iml230:~\$ **pwd**  
/home/tichit
- Sous le Bash, ~ est un *alias* pour /home/tichit.

# Exercice

Tapez les commandes suivantes :

- `tichit@iml230:~$ echo $HOME`
- `tichit@iml230:~$ echo ~`

# Les répertoires . et ..

Tout répertoire contient deux répertoires particuliers qui ont pour noms . et ..

- . représente le répertoire de travail (répertoire courant),
- .. représente le répertoire parent du répertoire courant.

Pour remonter dans le répertoire parent : **cd ..**

# Commandes du système de fichiers

# Lister les fichiers

- Lister le nom des fichiers du répertoire courant : **ls** (pour *list*)  
tichit@iml230:~/LaserBeam\$ **ls**  
Diffusion\_LaserBeam.class Diffusion\_LaserBeam.java
- **ls -a** permet de visualiser les fichiers cachés par défaut. Tout fichier dont le nom commence par "." est caché (., .., fichiers de configuration)
- Tapez **ls**, puis **ls -a** dans votre répertoire d'accueil.

# Supprimer des fichiers

- Détruire un fichier : `tichit@iml230:~/LaserBeam$ rm [fichier]`  
(pour *remove*)  
Attention, sous Unix, par défaut, un fichier détruit ne peut pas être récupéré.

## Informations associées aux fichiers

Des informations spécifiques sont attachées aux fichiers :

```
tichit@iml230:~/Test$ ls -l
```

```
total 36
```

```
-rw-r--r-- 1 tichit tichit 9145 2009-04-03 11:44 Test.txt  
-rw-r--r-- 1 tichit tichit 21702 2009-04-03 11:50 Titi.pl
```

Les informations sont dans l'ordre :

- Type du fichier et protections
- Nombre de liens sur le fichier
- Le nom du propriétaire
- Le nom du groupe auquel appartient le propriétaire
- Taille exprimée en octets
- Date et heure de la dernière modification
- Nom du fichier



# Exercice

On peut conjuguer les options **-a** et **-l**

Tapez :

- tichit@iml230:~/Test\$ **ls -la**

- tichit@iml230:~/Test\$ **ls -al**

et comparez.

## Copie, déplacement, renommage

- Afficher le contenu d'un fichier sur la sortie standard (le terminal actif) : **cat fichier** (pour *catalog*).

```
tichit@iml230:~ cat HelloWorld.pl
```

```
#!/usr/bin/perl  
print("Hello World!\n");
```

- Dupliquer un fichier : **cp** (pour *copy*).

```
tichit@iml230:~/Test$ ls
```

```
Test.txt
```

```
tichit@iml230:~/Test$ cp Test.txt Toto.txt
```

```
Test.txt Toto.txt
```

- Renommer/Déplacer d'un fichier : **mv** (pour *move*).

```
tichit@iml230:~/Test$ mv Toto.txt Titi.pl
```

```
Test.txt Titi.pl
```

# Création de fichier, type du fichier

- **touch fichier :**

- ▶ Modifie la date de dernier accès au fichier
- ▶ Crée le fichier s'il n'existe pas.

```
tichit@iml230:~$ ls -l
```

```
-rw-r--r-- 1 tichit tichit 21702 2009-04-03 11:50 Titi.pl
```

```
tichit@iml230:~$ touch Titi.pl
```

```
tichit@iml230:~$ ls -l
```

```
-rw-r--r-- 1 tichit tichit 21702 2009-04-03 12:41 Titi.pl
```

- **file [-f] fichier :** Décrit le contenu du fichier.

```
tichit@iml230:~$ file script
```

```
script: Bourne-Again shell script text executable
```

## Création/destruction d'un répertoire

- **mkdir répertoire...** : crée des répertoires.

```
tichit@iml230:~/Test$ mkdir rep1
```

```
tichit@iml230:~/Test$ ls
```

```
rep1/ Test.txt Titi.pl
```

- **rmdir répertoire...** : supprime des répertoires (vides!).

```
tichit@iml230:~/Test$ rmdir rep1
```

```
tichit@iml230:~/Test$ ls
```

```
Test.txt Titi.pl
```

- Pour supprimer un répertoire et son contenu :

**rm -fr répertoire** (r = recursive ; f=force).

ATTENTION quand vous utilisez **rm** !

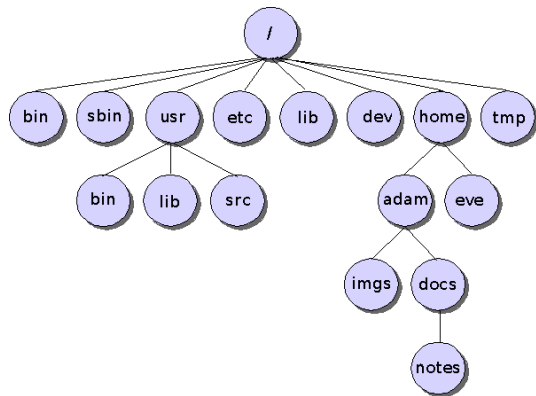
# Chemins

# Chemin des fichiers

Jusqu'à présent nous avons travaillé sur les fichiers du répertoire de travail. On peut travailler de la même façon sur n'importe quel fichier du système.

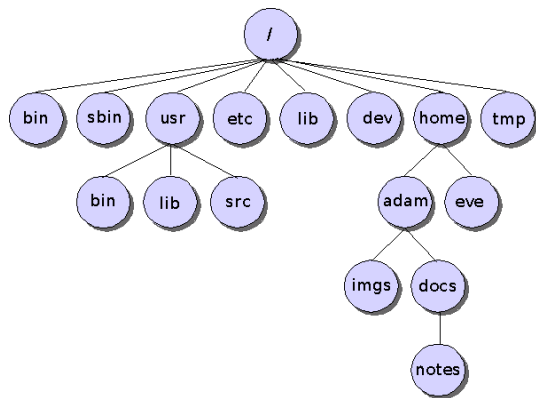
- On peut indiquer son chemin **absolu** : chemin à partir de la racine (/) du système de fichiers.
- On peut indiquer son chemin **relatif** : chemin à partir du **répertoire de travail**.

# Chemin absolu



- Le chemin absolu du fichier *notes* est : **/home/adam/docs/notes**
- Le chemin absolu du répertoire *eve* est : **/home/eve**

# Chemin relatif



- Le chemin relatif du fichier *notes* à partir du répertoire *adam* est : **docs/notes**
- Le chemin relatif du fichier *notes* à partir du répertoire *eve* est : **../adam/docs/notes**



# Chemin des fichiers

- Plutôt que de recourir tout le temps à la commande **cd**, il est plus rapide de se servir des chemins.

- Ex : Préférer :

```
tichit@iml230:~/Test$ cat /etc/fstab
```

à

```
tichit@iml230:~/Test$ cd /etc
```

```
tichit@iml230:~/Test$ cat fstab
```

On a aussi :

- `tichit@iml230:~/Test$ mv ~/Desktop/image.jpg ~/Images/`
- `tichit@iml230:~/Test$ mv ~/Desktop/image.jpg`  
`~/Images/image1.jpg`
- ...

# Exercice

En utilisant le manuel :

- Que fait la commande **mkdir A/B/C** ?
- Que fait la commande **mkdir -p A/B/C (man mkdir)** ?

# Exercice

En utilisant le manuel :

- Que fait la commande **mkdir A/B/C**?
- Que fait la commande **mkdir -p A/B/C (man mkdir)**? C'est plus rapide que **mkdir A ; mkdir A/B ; mkdir A/B/C** ou que **mkdir A ; cd A ; mkdir B ; cd B ; mkdir C**

# Chemin des fichiers

Pour obtenir le nom d'un fichier à partir de son chemin complet, on utilise la commande suivante :

- **basename chaîne suffixe** : supprime le plus long préfixe se terminant par '/' de la chaîne et le suffixe (s'il se trouve en fin de chaîne).

```
tichit@iml230:~$ basename /home/tichit/Test/test.txt  
test.txt
```

- **dirname chemin-de-fichier** : donne le chemin du répertoire dans lequel se trouve le fichier. tichit@iml230:~\$ **dirname**

```
/home/tichit/Test  
/home/tichit
```

# Commandes relatives au système de fichiers

- **du [option] fichier-ou-répertoire** : place disque utilisée par un fichier/répertoire en kilo-octets.

```
tichit@iml230:~$ du Test
```

```
40 Test/
```

- **df [répertoire]** : place libre et occupée sur le(s) système(s) de fichiers.

```
tichit@iml230:~$ df .
```

Sys. de fich.	1K-blocs	Occupé	Disponible	C
/dev/sda6	14136684	9092060	4326524	68% /h

Note : . représente le répertoire courant.

# Exercice

- Que fait la commande **df -h** (**man df**) ?
- Combien d'espace disque reste-t'il sur votre partition principale (/) ?