

IF110 - Système de Fichiers



Laurent Réveillère

Université de Bordeaux

`laurent.reveillere@u-bordeaux.fr`
`http://www.reveillere.fr/`

D'après le cours d'introduction aux systèmes d'exploitation de Télécom SudParis



Système de Fichiers

- ❑ **Besoin de mémoriser des informations**
 - Photos, PDF, données brutes, exécutables d'applications, etc.
- ❑ **Organisation du stockage sur mémoire de masse**
 - Localisation abstraite grâce à un chemin dans une arborescence
 - Unité de base = fichier
- ❑ **Exemples de types de systèmes de fichiers**
 - NTFS pour Windows, ext2, ext3, ext4 pour Linux, HFSX pour Mac-OS
 - FAT pour les clés USB, ISO pour les CD
 - ... et des myriades d'autres types de systèmes de fichiers



Plan

- ❑ Système de fichiers vu par un processus
- ❑ Système de fichiers sur disque
- ❑ Commandes utilisateurs
- ❑ Droits d'accès

IF110

3



Qu'est-ce qu'un fichier

- ❑ Un fichier est la réunion de
 - Un contenu, c'est-à-dire un ensemble ordonné d'octets
 - Un propriétaire
 - Des horloges scalaires (création, dernier accès, dernière modif)
 - Des droits d'accès (en lecture, en écriture, en exécution)
- ❑ Attention :
 - le nom ne fait pas parti du fichier
 - un même fichier peut avoir plusieurs noms

IF110

4



On stocke de nombreux de fichiers

- ❑ Facilement plusieurs centaines de milliers de fichiers dans un ordinateur
 - Plusieurs milliers gérés/utilisés directement par l'utilisateur
 - Plusieurs centaines de milliers pour le système et les applications
- ❑ Problème : comment retrouver facilement un fichier parmi des centaines de milliers ?
- ❑ Solution : en rangeant les fichiers dans des répertoires (aussi appelés dossiers)

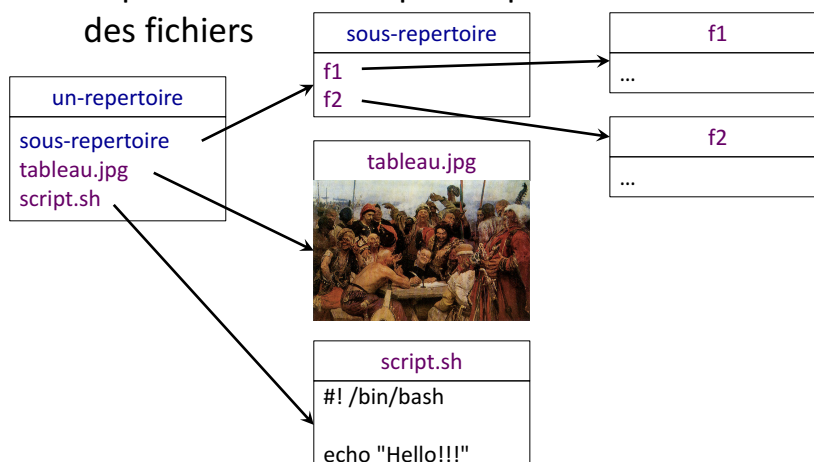
IF110

5



Organisation en répertoires

- ❑ Répertoire = fichier spécial qui associe des noms à des fichiers



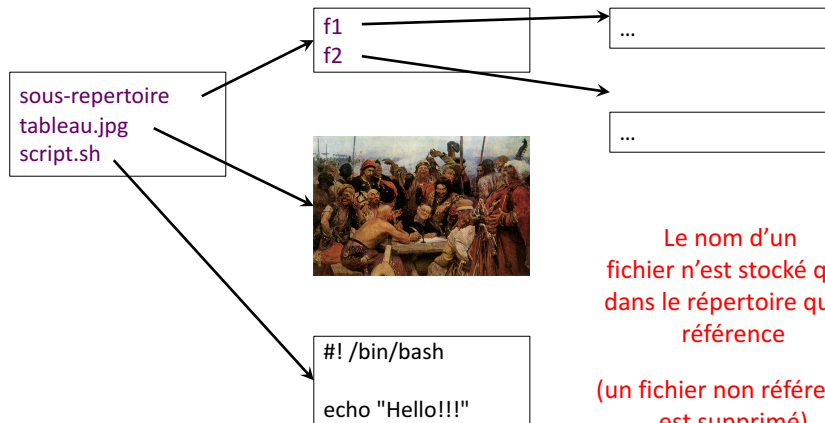
IF110

6



Organisation en répertoires

- ❑ Répertoire = fichier spécial qui associe des noms à des fichiers



Le nom d'un
fichier n'est stocké que
dans le répertoire qui le
réfère
(un fichier non référencé
est supprimé)

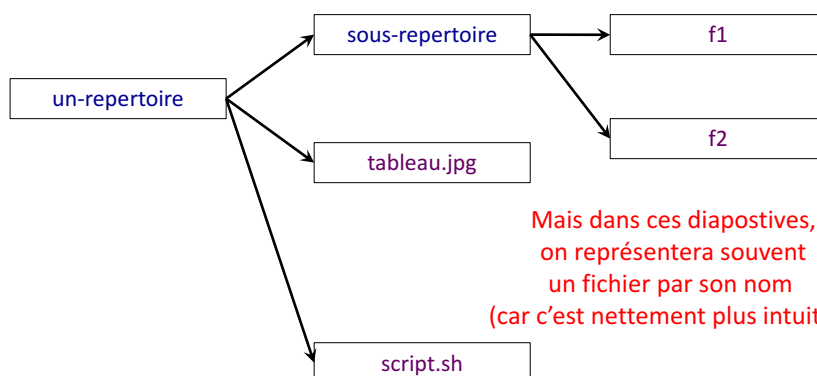
IF110

7



Organisation en répertoires

- ❑ Répertoire = fichier spécial qui associe des noms à des fichiers



Mais dans ces diapositives,
on représentera souvent
un fichier par son nom
(car c'est nettement plus intuitif !)

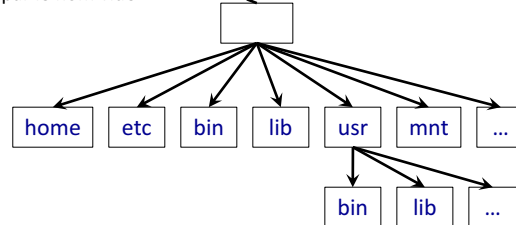
IF110

8



Arborescence standard UNIX

La racine est référencée
par le nom vide



La plupart des systèmes d'exploitation Unix
(GNU/Linux, BSD, MacOS...) utilisent une arborescence
de base standardisée
(seul Windows utilise une arborescence réellement différente)

Vous pouvez la consulter en faisant : `man hier` (pour hierarchy)

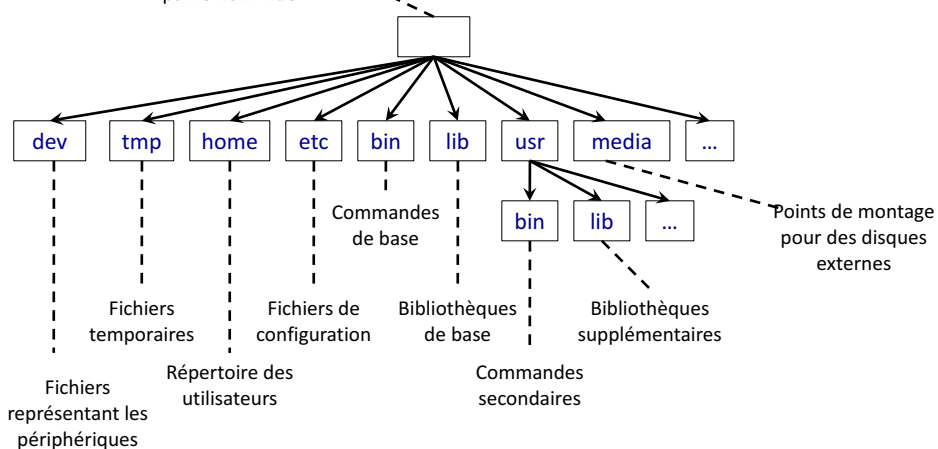
IF110

9



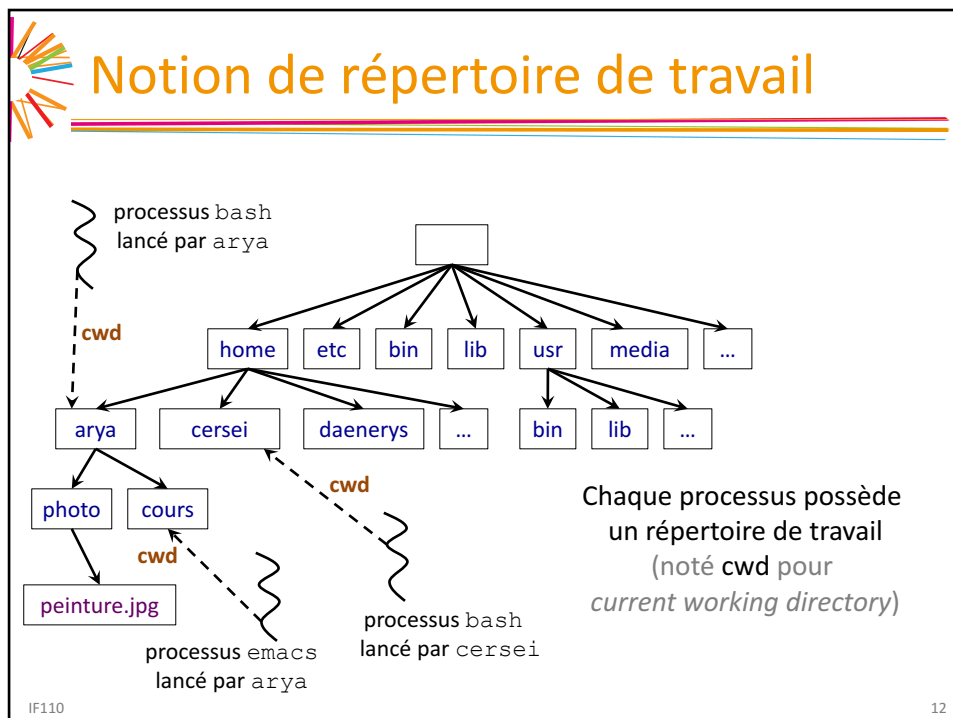
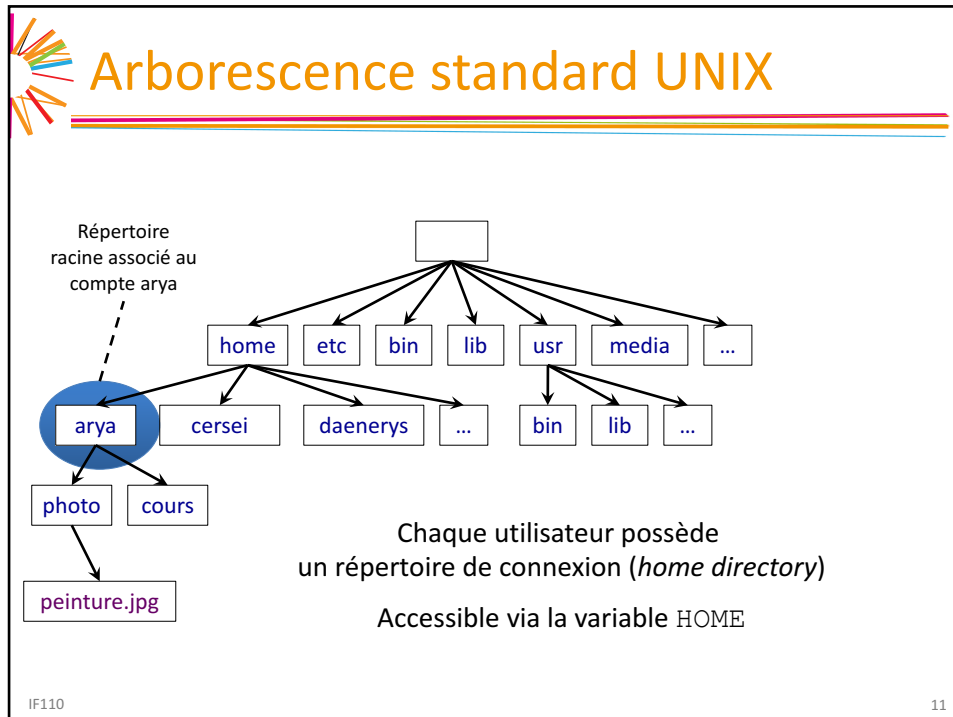
Arborescence standard UNIX

La racine est référencée
par le nom vide



IF110

10





Notion de chemin

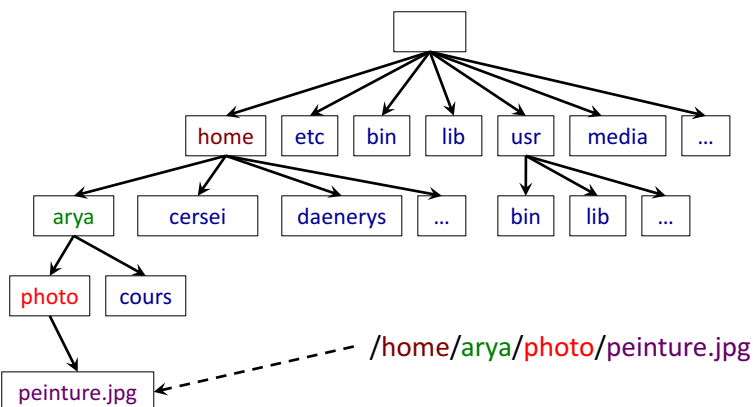
- ❑ En `bash`, le séparateur de répertoire est le caractère `/`
- ❑ Un chemin s'écrit sous la forme `a/b/c` qui référence
 - le fichier `c`
 - se trouvant dans le répertoire `b`
 - se trouvant lui même dans le répertoire `a`
- ❑ Un **chemin absolu** part de la racine du système de fichiers
Commence par le nom vide (racine), par exemple `/a/b/c`
- ❑ Un **chemin relatif** part du répertoire de travail du processus
Commence par un nom non vide, par exemple `a/b/c`

IF110

13



Exemple de chemin absolu (1/2)



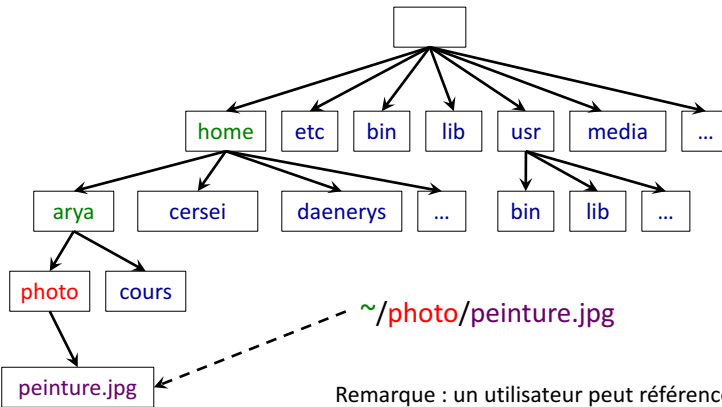
IF110

14



Exemple de chemin absolu (2/2)

Un utilisateur peut utiliser ~ comme raccourci pour son répertoire de connexion (répertoire personnel)



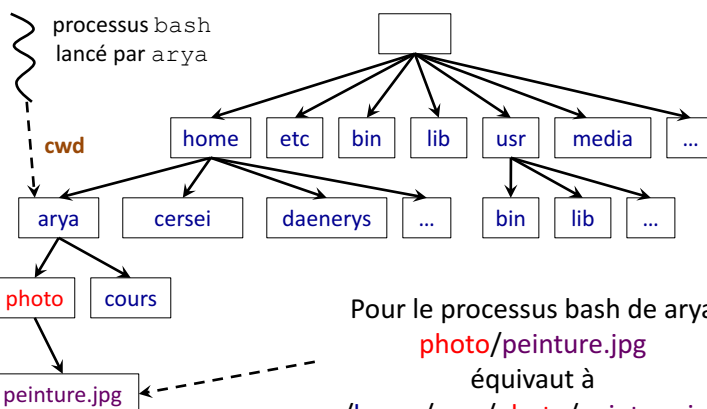
Remarque : un utilisateur peut référencer le répertoire de connexion d'un autre utilisateur avec ~name (par exemple `~arya/photo/peinture.jpg`)

IF110

15



Exemple de chemin relatif (1/3)



Pour le processus bash de arya
`photo/peinture.jpg`
équivalut à
`/home/arya/photo/peinture.jpg`

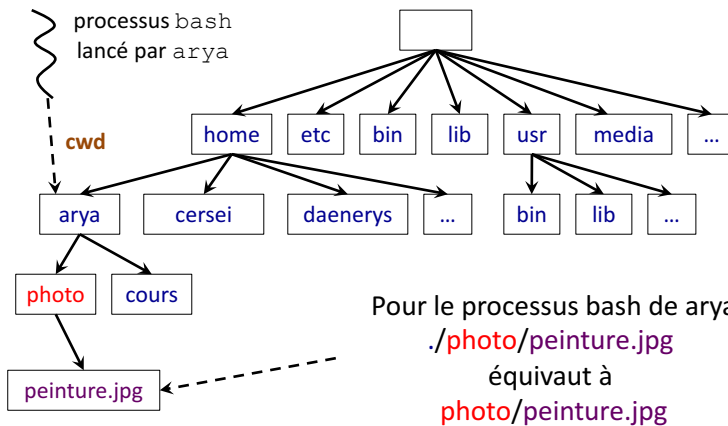
IF110

16



Exemple de chemin relatif (2/3)

Chaque répertoire possède un fichier nommé `.` s'auto-référençant



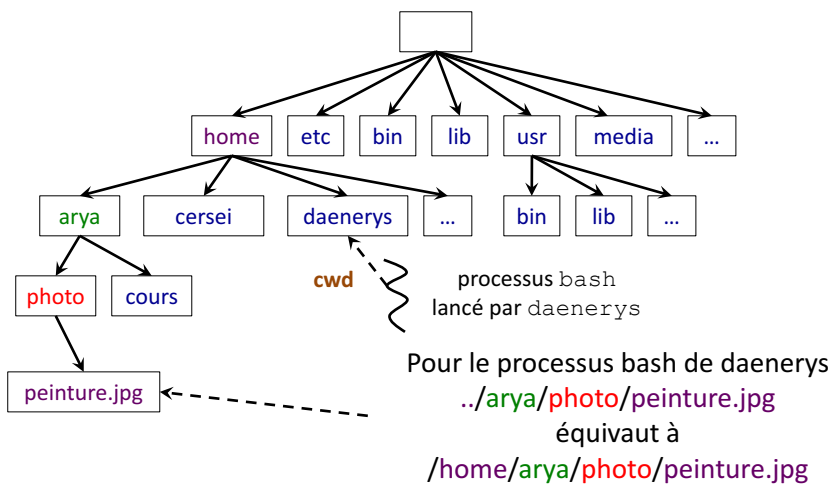
IF110

17



Exemple de chemin relatif (3/3)

Chaque répertoire possède un fichier nommé `..` référençant son parent



IF110

18



Remarque

- Dans `bash`, quand vous écrivez `./script.sh`, vous référencez le fichier `script.sh` du répertoire de travail du processus `bash` de votre terminal

IF110

19



Exemple

```
#!/bin/bash
```

```
echo "Bonjour, vous êtes dans le répertoire $PWD"  
echo "Votre maison se trouve en $HOME"  
echo "Et vous avez lancé le script $0"
```

```
/home/reveillere/tmp/script.sh
```

```
$ ./script.sh  
Bonjour, vous êtes dans le répertoire /home/reveillere/tmp  
Votre maison se trouve en /home/reveillere  
Et vous avez lancé le script ./script.sh  
$
```

IF110

20



Explorer l'arborescence de fichiers

❑ `cd chem`

⇒ change le répertoire courant vers `chem`

Exemple : `cd ../cersei; cd /home/arya/photo`
(sans argument, `cd` va dans votre répertoire de connexion)

❑ `pwd` (*print working directory*)

⇒ affiche le répertoire de travail (↔ `echo $PWD`)

IF110

21



Explorer l'arborescence de fichiers

❑ `ls chem`

⇒ Affiche le contenu d'un répertoire

- Si `chem` absent : affiche le contenu du répertoire de travail
- Si `chem` répertoire : affiche le contenu du répertoire `chem`
- Sinon si `chem` est un fichier : affiche le nom du fichier
- Options utiles :
 - a : affiche les fichiers cachés (c.-à.d., commençant par '.')
 - l : affichage long (propriétaire, droits d'accès, taille etc.)

IF110

22



Plan

- ❑ Système de fichiers vu par un processus
- ❑ Système de fichiers sur disque
- ❑ Commandes utilisateurs
- ❑ Droits d'accès

IF110

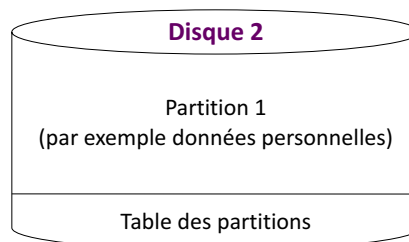
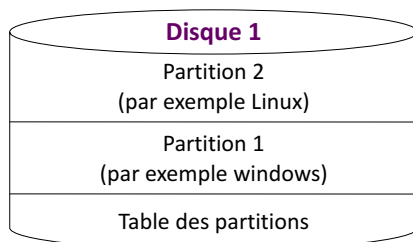
23



Organisation des disques

- ❑ Une machine peut posséder plusieurs disques
- ❑ ... chaque disque peut être scindé en plusieurs partitions
- ❑ ... chaque partition possède son système de fichiers indépendant

Parfois utile de séparer les données utilisateurs et le système d'exploitation



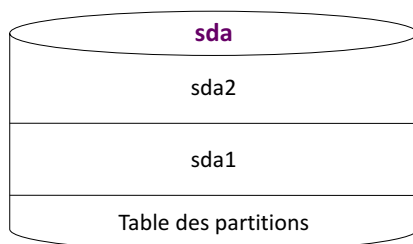
IF110

24

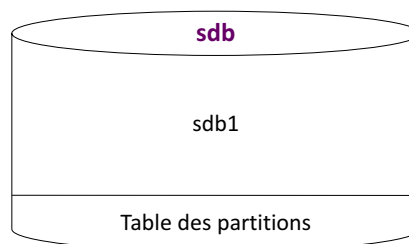


Partitions dans les systèmes UNIX

- ❑ Un disque est identifié par le préfixe `sd` (*scsi drive*)
- ❑ Les disques sont numérotés `a`, `b`, `c`...
- ❑ Les partitions sont numérotées `1`, `2`, `3`...
(vous pouvez voir les disques/partitions en faisant `ls /dev`)



IF110



25



Système de fichiers sur disque (1/2)

- ❑ 3 concepts fondamentaux
 - Le bloc : unité de transfert entre le disque et la mémoire
(souvent 4096 octets)
 - L'inode (*index node*) : descripteur d'un fichier
 - » Type du fichier (ordinaire, répertoire, autres)
 - » Propriétaire, droits, dates de création/modification/accès
 - » Taille
 - » Liste des blocs du contenu du fichier
 - » ...
 - Donc, dans ce cours : fichier = inode + blocs du fichier

IF110

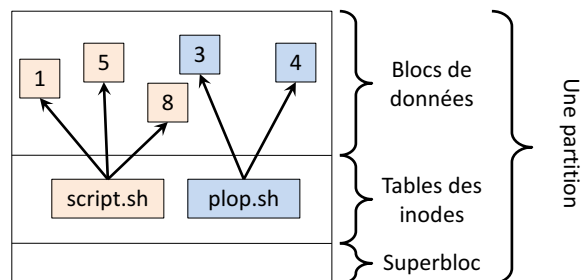
26



Le système de fichiers sur disque (2/2)

- ❑ Avec `ext`, utilisé sous GNU/Linux, trois zones principales
 - Le superbloc, au début, décrit les autres zones
 - La table des inodes contient les inodes (inode 0 = racine)
 - La zone des blocs de données contient les données des fichiers

Par exemple,
contenu de `script.sh` :
4096 octets du bloc 1 puis
4096 octets du bloc 5 puis
312 octets du bloc 8



IF110

27



Montage d'une partition (1/2)

- ❑ Le système maintient une table des montages qui associe des chemins (points de montage) et des disques
 - `/` $\Rightarrow$ `sda1`
 - `/home` $\Rightarrow$ `sdb1`
 - `/mnt/windows` $\Rightarrow$ `sdb2`

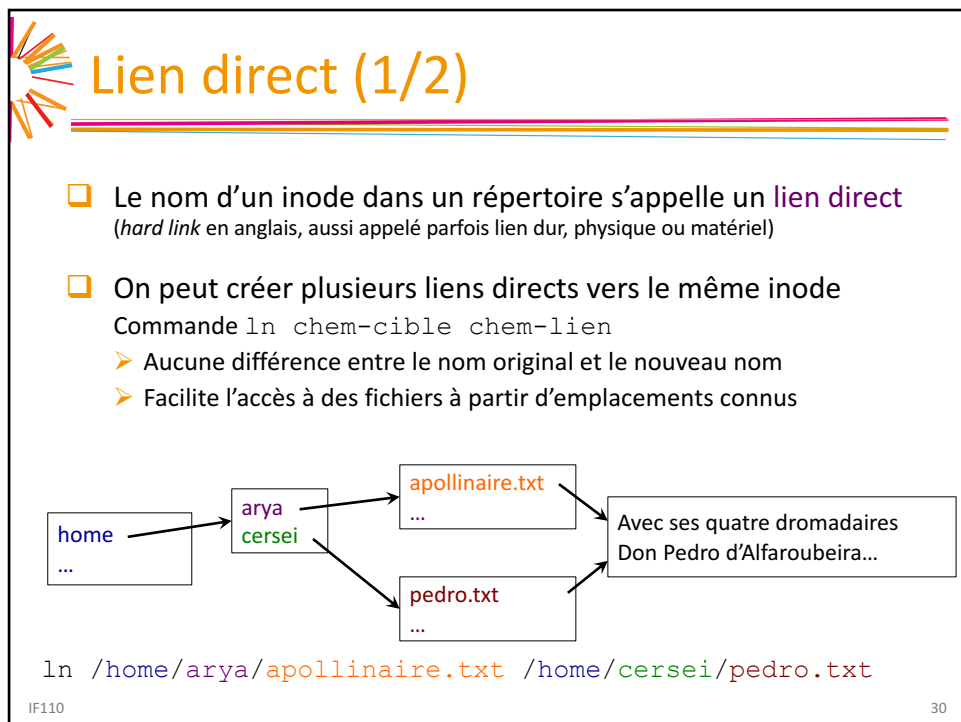
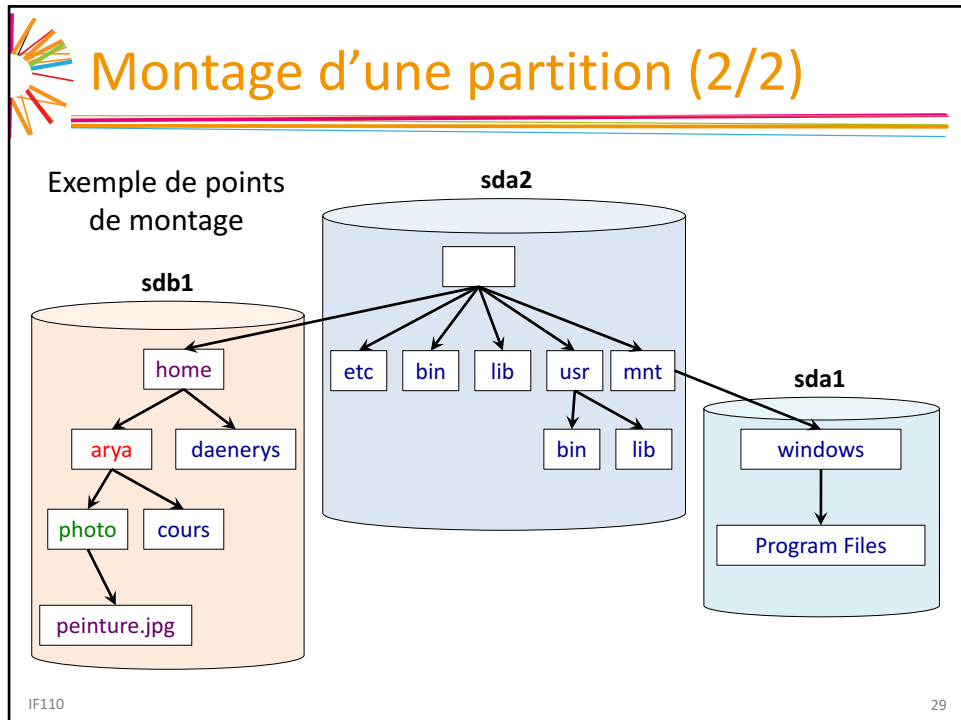
Remarque : les partitions du disque dur peuvent se trouver sur une autre machine (typiquement Network File System, comme en salle TP, voir <https://doc.ubuntu-fr.org/nfs>)

- ❑ Lorsqu'un processus accède à un point de montage, il accède à l'inode racine du disque indiqué dans la table des montages

Par exemple `cd /mnt/windows` accède à l'inode racine de `sdb2`

IF110

28





Lien direct (2/2)

- ❑ Mais faire de multiples liens directs pour faire des raccourcis peut poser problème
 - Pour supprimer un fichier, il faut supprimer tous les liens directs vers son inode, mais les utilisateurs sont distraits et en oublient
 - Un lien direct ne peut référencer qu'un inode de la même partition

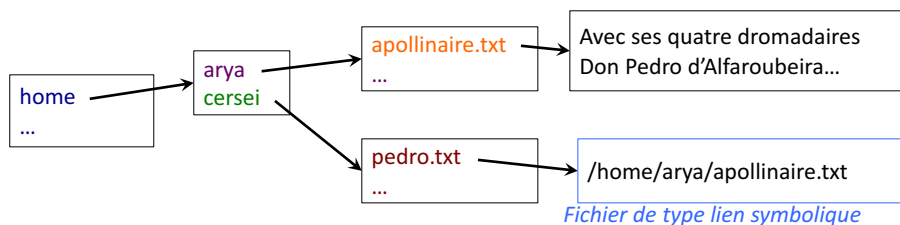
IF110

31



Notion de lien symbolique (1/2)

- ❑ Pour faire des raccourcis on utilise aussi des liens symboliques
 - Comme `ln -s chem-cible chem-lien`
 - Fichier spécial (type lien) dont le contenu est un chemin cible
 - Lorsque le système doit ouvrir le fichier, il ouvre la cible à la place de l'original



```
ln -s /home/arya/apollinaire.txt /home/cersei/pedro.txt
```

IF110

32



Notion de lien symbolique (2/2)

□ Avantages des liens symboliques

- Dès que le fichier cible est détruit, son espace est libéré (ouvrir le lien symbolique engendre alors une erreur)
- Un lien symbolique peut référencer un fichier quelconque, y compris appartenant à une autre partition

□ Principal inconvénient des liens symboliques

- En cas de déplacement du fichier cible, le lien symbolique peut devenir invalide

IF110

33



Il existe de nombreux types de fichiers

- Fichier ordinaire
- Répertoire
- Lien symbolique
- Device : un fichier qui représente un périphérique (disque dur, carte son, carte réseau, ...)
 - » Par exemple `/dev/sda1`
- Tube nommé : fichier spécial étudié plus loin
- Socket : fichier spécial proche des tubes (non étudié dans ce cours, voir cours prog. réseau en T2)

IF110

34



Plan

- ❑ Système de fichiers vu par un processus
- ❑ Système de fichier sur disque
- ❑ Commandes utilisateurs
- ❑ Droits d'accès

IF110

35



Commandes utilisateur

- ❑ Commandes de base sur les fichiers
 - Création
 - Suppression
 - Copie
 - Déplacement / renommage
 - Consultation
 - Recherche
- ❑ Commandes utilitaires bien pratiques
 - Principales vues en CI et TP

IF110

36



Création d'un fichier

❑ Création d'un fichier ordinaire :

- Au travers de logiciels
 - » en particulier des éditeurs : emacs, vi, gedit, nano, etc...
- `touch chem` : crée fichier vide + mise à jour heures modif.

❑ Création d'un répertoire :

- `mkdir rep`

❑ Création d'un lien :

- Lien dur : `ln chem-cible chem-lien`
- Lien symbolique : `ln -s chem-cible chem-lien`

IF110

37



Suppression d'un fichier (1/5)

❑ Supprimer un fichier (tout type, sauf répertoire)

`rm chem`

- Suppression de l'entrée associée au chemin dans le répertoire parent
 - » Décrémenter le compteur de liens directs de l'inode
 - » Libère le fichier (inode + données) si compteur tombe à zéro

IF110

38



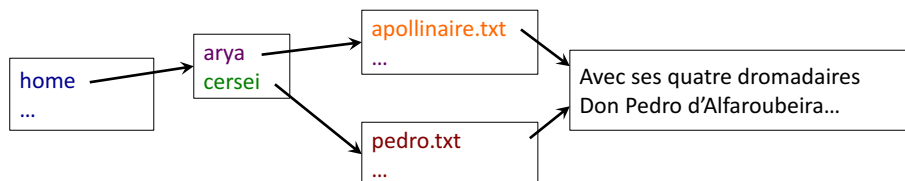
Suppression d'un fichier (2/5)

❑ Supprimer un fichier (tout type, sauf répertoire)

```
rm chem
```

➤ Suppression de l'entrée associée au chemin dans le répertoire parent

- » Décrémentation du compteur de liens directs de l'inode
- » Libère le fichier (inode + données) si compteur tombe à zéro



IF110

39



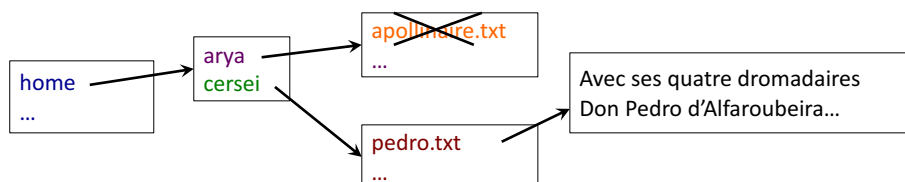
Suppression d'un fichier (3/5)

❑ Supprimer un fichier (tout type, sauf répertoire)

```
rm chem
```

➤ Suppression de l'entrée associée au chemin dans le répertoire parent

- » Décrémentation du compteur de liens directs de l'inode
- » Libère le fichier (inode + données) si compteur tombe à zéro



```
rm /home/arya/apollinaire.txt
```

IF110

40



Suppression d'un fichier (4/5)

❑ Supprimer un fichier (tout type, sauf répertoire)

`rm chem :remove`

➤ Suppression de l'entrée associée au chemin dans le répertoire parent

- » Décrémentation du compteur de liens directs de l'inode
- » Libère le fichier (inode + données) si compteur tombe à zéro



`rm /home/cersei/pedro.txt`

IF110

41



Suppression d'un fichier (5/5)

❑ Supprimer un fichier (tout type, sauf répertoire)

`rm chem`

➤ Suppression de l'entrée associée au chemin dans le répertoire parent

- » Décrémentation du compteur de liens directs de l'inode
- » Libère le fichier (inode + données) si compteur tombe à zéro

❑ Supprimer un répertoire

- `rmdir <rep>` : suppression d'un répertoire vide
- `rm -r <rep>` : suppression récursive d'un répertoire et de tous les sous-fichiers (sous-répertoires inclus)

(faites très attention avec cette commande !)

IF110

42



Copie d'un fichier (1/3)

- ❑ `cp [-r] <src> <dest>`
Création d'un nouvel inode et duplication des blocs de données
 - `src` correspond au chemin du fichier à copier
 - `dest`, au chemin où doit être copiée `src`
- ❑ Deux fonctionnements différents
 - Si `dest` est un répertoire, copie `src` dans le répertoire `dest` (dans ce cas, multiples copies possibles avec `cp fic1 fic2 ... rep`)
 - Sinon, copie `src` sous le nom `dest`
- ❑ L'option `-r` permet de copier récursivement un répertoire (sans `-r`, si `src` est un répertoire, erreur)

IF110

43



Copie d'un fichier (2/3)

- ❑ `cp [-r] <src> <dest>`
Création d'un nouvel inode et duplication des blocs de données
 - `src` correspond au chemin du fichier à copier
 - `dest`, au chemin où doit être copiée `src`



IF110

44

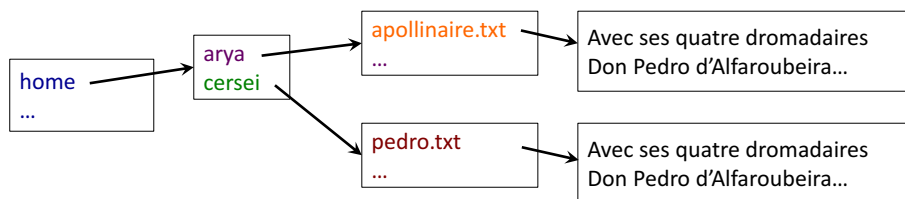


Copie d'un fichier (3/3)

❏ `cp [-r] <src> <dest>`

Création d'un nouvel inode et duplication des blocs de données

- `src` correspond au chemin du fichier à copier
- `dest`, au chemin où doit être copiée `src`



```
cp /home/arya/apollinaire.txt /home/cersei/pedro.txt
```

IF110

45



Déplacement / renommage d'un fichier

❏ `mv <src> <dest>` (déplace ou renomme)

- `src` : *fichier de type quelconque*
- Si `dest` est un répertoire, déplace `src` dans le répertoire `dest`
(dans ce cas, multiples déplacements possibles avec `mv fic1 fic2 ... rep`)
- Sinon, déplace `src` sous le nom `dest`

❏ Fonctionnement :

- Déplacement dans la même partition
 - » Crée un lien direct à partir de `src` puis supprime `src`
- Déplacement sur une autre partition
 - » Copie `src` vers `dest` puis supprime `src`

IF110

46



Plan

- ❑ Système de fichiers vu par un processus
- ❑ Système de fichiers sur disque
- ❑ Commandes utilisateurs
- ❑ Droits d'accès

IF110

47



Droits d'accès

- ❑ Toute opération sur un fichier est soumise à droits d'accès
 - Message d'erreur « *Permission non accordée* »
- ❑ 3 types d'accès
 - **r** : *read*
 - » Si répertoire, consultation de ses entrées (c.à.d, `ls` autorisé)
 - » Sinon, consultation du contenu du fichier
 - **w** : *write*
 - » Si répertoire, droit de création, de renommage et de suppression d'une entrée dans le répertoire
 - » Sinon, droit de modification du contenu du fichier
 - **x** : *execute*
 - » si répertoire, droit de traverser (c.à.d., `cd` autorisé)
 - » sinon, droit d'exécution

IF110

48



Droits d'accès

3 catégories d'utilisateurs:

- Propriétaire (u)
- Groupe propriétaire (g)
- Tous les autres (o)

Consultation des droits d'accès d'un fichier :

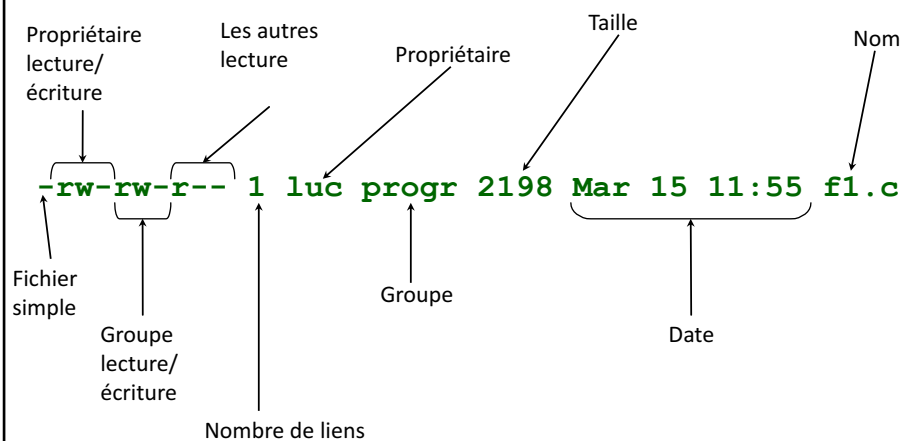
```
ls -ld chemin
```

IF110

49



Exemple : fichier simple



IF110

50



Droits d'accès – modification

- ❑ Modification sur une entrée existante
chmod droit entrée : change mode
- ❑ Droits à **appliquer!** à l'entrée
 - Catégories : u, g, o ou a (= all c.à.d., ugo)
 - Opérations : Ajout (+), retrait (-), affectation (=)

```
$
```

IF110

51



Droits d'accès – modification

- ❑ Modification sur une entrée existante
chmod droit entrée
- ❑ Droits à **appliquer!** à l'entrée
 - Catégories : u, g, o ou a(= all = ugo)
 - Opérations : Ajout (+), retrait (-), affectation (=)

```
$ ls -ld fichier
-rwx r-- --- fichier
$
```

IF110

52



Droits d'accès – modification

❑ Modification sur une entrée existante

chmod droit entrée : *change mode*

❑ Droits à **appliquer!** à l'entrée

➤ Catégories : u, g, o ou a(= all = ugo)

➤ Opérations : Ajout (+), retrait (-), affectation (=)

```
$ ls -ld fichier
-rwx r-- --- fichier
$ chmod u-x fichier
$ ls -ld fichier
-rw- r-- --- fichier
$
```

IF110

53



Droits d'accès – modification

❑ Modification sur une entrée existante

chmod droit entrée : *change mode*

❑ Droits à **appliquer!** à l'entrée

➤ Catégories : u, g, o ou a(= all = ugo)

➤ Opérations : Ajout (+), retrait (-), affectation (=)

```
$ ls -ld fichier
-rwx r-- --- fichier
$ chmod u-x fichier
$ ls -ld fichier
-rw- r-- --- fichier
$ chmod u+x fichier
$ ls -ld fichier
-rwx r-- --- fichier
```

IF110

54



Démonstration

```
$ cp /etc/passwd .  
$
```

IF110 55



Démonstration

```
$ cp /etc/passwd .  
$ ls -l  
total 4  
-rw-r--r-- 1 reveillere users 1927 sept. 28 11:45 passwd  
$
```

IF110 56



Démonstration

```
$ cp /etc/passwd .
$ ls -l
total 4
-rw-r--r-- 1 reveillere users 1927 sept. 28 11:45 passwd
$ chmod u-r passwd
$
```

IF110 57



Démonstration

```
$ cp /etc/passwd .
$ ls -l
total 4
-rw-r--r-- 1 reveillere users 1927 sept. 28 11:45 passwd
$ chmod u-r passwd
$ cat passwd
cat: passwd: Permission non accordée
$
```

IF110 58



Démonstration

```
$ cp /etc/passwd .
$ ls -l
total 4
-rw-r--r-- 1 reveillere users 1927 sept. 28 11:45 passwd
$ chmod u-r passwd
$ cat passwd
cat: passwd: Permission non accordée
$ mkdir rep
$
```

IF110

59



Démonstration

```
$ cp /etc/passwd .
$ ls -l
total 4
-rw-r--r-- 1 reveillere users 1927 sept. 28 11:45 passwd
$ chmod u-r passwd
$ cat passwd
cat: passwd: Permission non accordée
$ mkdir rep
$ ls -l
total 8
--w-r--r-- 1 reveillere users 1927 sept. 28 11:45 passwd
drwxr-xr-x 2 reveillere users 4096 sept. 28 11:47 rep
$
```

IF110

60



Démonstration

```
$ cp /etc/passwd .
$ ls -l
total 4
-rw-r--r-- 1 reveillere users 1927 sept. 28 11:45 passwd
$ chmod u-r passwd
$ cat passwd
cat: passwd: Permission non accordée
$ mkdir rep
$ ls -l
total 8
--w-r--r-- 1 reveillere users 1927 sept. 28 11:45 passwd
drwxr-xr-x 2 reveillere users 4096 sept. 28 11:47 rep
$ cd rep/
$
```

IF110

61



Démonstration

```
$ cp /etc/passwd .
$ ls -l
total 4
-rw-r--r-- 1 reveillere users 1927 sept. 28 11:45 passwd
$ chmod u-r passwd
$ cat passwd
cat: passwd: Permission non accordée
$ mkdir rep
$ ls -l
total 8
--w-r--r-- 1 reveillere users 1927 sept. 28 11:45 passwd
drwxr-xr-x 2 reveillere users 4096 sept. 28 11:47 rep
$ cd rep/
$ cd ..
$
```

IF110

62



Démonstration

```
$ cp /etc/passwd .
$ ls -l
total 4
-rw-r--r-- 1 reveillere users 1927 sept. 28 11:45 passwd
$ chmod u-r passwd
$ cat passwd
cat: passwd: Permission non accordée
$ mkdir rep
$ ls -l
total 8
--w-r--r-- 1 reveillere users 1927 sept. 28 11:45 passwd
drwxr-xr-x 2 reveillere users 4096 sept. 28 11:47 rep
$ cd rep/
$ cd ..
$ chmod u-x rep
$
```

IF110

63



Démonstration

```
$ cp /etc/passwd .
$ ls -l
total 4
-rw-r--r-- 1 reveillere users 1927 sept. 28 11:45 passwd
$ chmod u-r passwd
$ cat passwd
cat: passwd: Permission non accordée
$ mkdir rep
$ ls -l
total 8
--w-r--r-- 1 reveillere users 1927 sept. 28 11:45 passwd
drwxr-xr-x 2 reveillere users 4096 sept. 28 11:47 rep
$ cd rep/
$ cd ..
$ chmod u-x rep
$ cd rep
bash: cd: rep: Permission non accordée
```

IF110

64



Droits d'accès initiaux

- ❑ Masque de droits d'accès **retirés!** à la création de tout fichier
 - Commande `umask` (*user mask*)
 - Le masque est donné en octal (base 8) avec 3 chiffres (u, g, o)
 - En standard, masque par défaut = 022
 - » $r = 100$ en binaire = 4 en octal, $w = 010 = 2$
 - » Le droit d'exécution n'est pas affecté par le `umask`
 - » Si droits retirés --- -w- -w-, alors droits appliqués `rw- r-- r--`
 - Modification du masque grâce à la commande `umask`
 - » **Attention** : `umask` sans effet rétroactif sur les fichiers préexistantes
 - » **Attention** : `umask` n'a d'effet que sur le `bash` courant

IF110

65



Conclusion

- ❑ Concepts clés :
 - Arborescence, racine du système de fichier, répertoire de connexion, répertoire de travail
 - Chemin absolu, chemin relatif
 - Droits d'accès
 - Partition, inode
 - Fichier, répertoire, liens (direct et symbolique)
- ❑ Commandes clés :
 - `pwd`, `cd`, `ls`
 - `chmod`, `umask`
 - `mkdir`, `ln`, `rm`, `rmdir`, `cp`, `mv`

IF110

66