UPMC
PARIS UNIVERSITAS

Préparation LPI

Exam 101

104.6 Liens hard et liens symboliques

104.6 – Domaines de connaissances

- Poids : 2
- Créer des liens
- Identifier les liens hard et liens symboliques
- Différence entre copie et création de lien
- Utilisation de liens par l'administrateur

104.6 – Liens

- Quelques slides pour expliquer ce que l'on entend par « lien » sous Linux....

Notions d'objets Unix

- Sous Unix, plusieurs type d'objets
  - fichiers
  - répertoires
  - objets associés aux disques durs, clefs USB, bandes
  - objets destinés à la communication entre applications
- On manipule le plus souvent
  - fichiers
  - répertoires
- Une règle que l'on vérifiera souvent
  - « Sous Unix, tout est fichier »

- Sous Unix, la « casse » est importante : le système fait la différence entre les majuscules et les minuscules

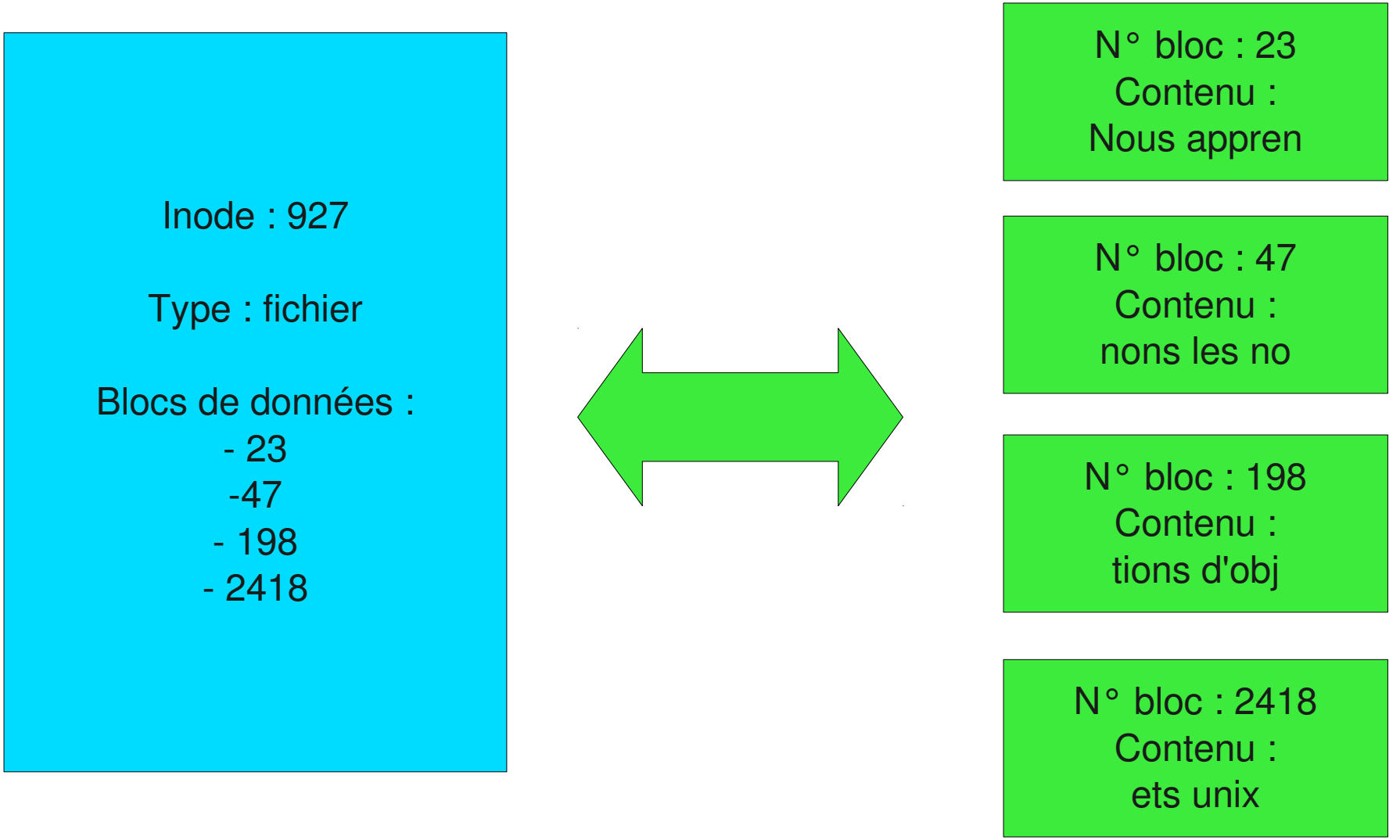
```
[aoi@test]$ ls -l
total 0
-rw-r--r-- 1 franck franck 0 mar 13 10:27 exemple.txt
-rw-r--r-- 1 franck franck 0 mar 13 10:26 exeMPLE.TXT
-rw-r--r-- 1 franck franck 0 mar 13 10:26 ExemplE.txt
-rw-r--r-- 1 franck franck 0 mar 13 10:26 EXEMPLE.txt
```

- Sous Unix
  - on évite les caractères espace, apostrophe, guillemets, lettres accentuées dans les noms d'objet
  - on utilise généralement : les lettres minuscules, les lettres majuscules, le tiret « - », le point « . », l'underscore « _ »

Notions d'objets Unix

- Sur un système de fichier Unix, les objets (fichiers) sont gérés par l'intermédiaire d'une « meta-donnée » stockée dans le système de fichier appelée **inode**
  - chaque inode dispose d'un numéro identifiant le fichier
  - il renseigne sur le type d'objet (fichier, répertoire,...)
  - il possède la liste des blocs de données qui correspondent à son contenu
  - Attention : le nom de l'objet ne se trouve pas dans l'inode. Le nom du fichier est géré par le répertoire.

- Un fichier correspond à un inode de type fichier

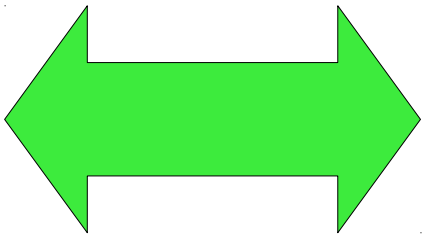


- Sous unix, un répertoire est aussi un fichier
  - ce fichier particulier contient une liste d'associations : nom de fichier - inode

Inode : 65

Type : répertoire

Blocs de données :
- 56
-72
- 12048
- 1768



N° bloc : 56
Contenu :
(toto, 789)

N° bloc : 72
Contenu :
(titi, 6754)

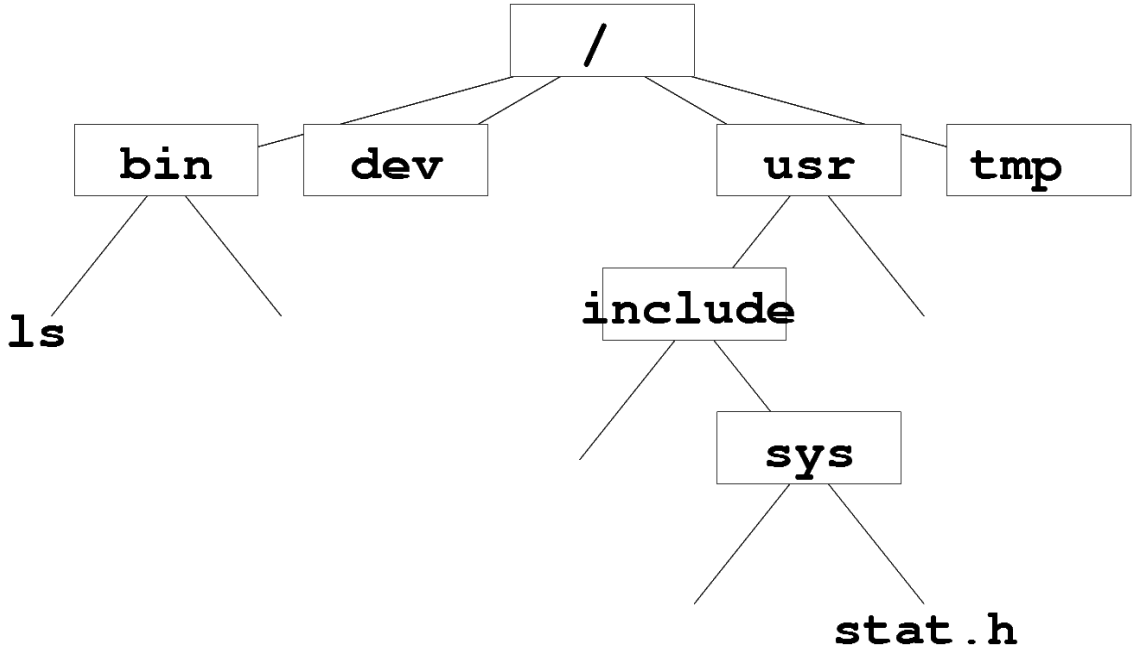
N° bloc : 12048
Contenu :
(tata, 8756)

N° bloc : 1768
Contenu :
(tutu,59)

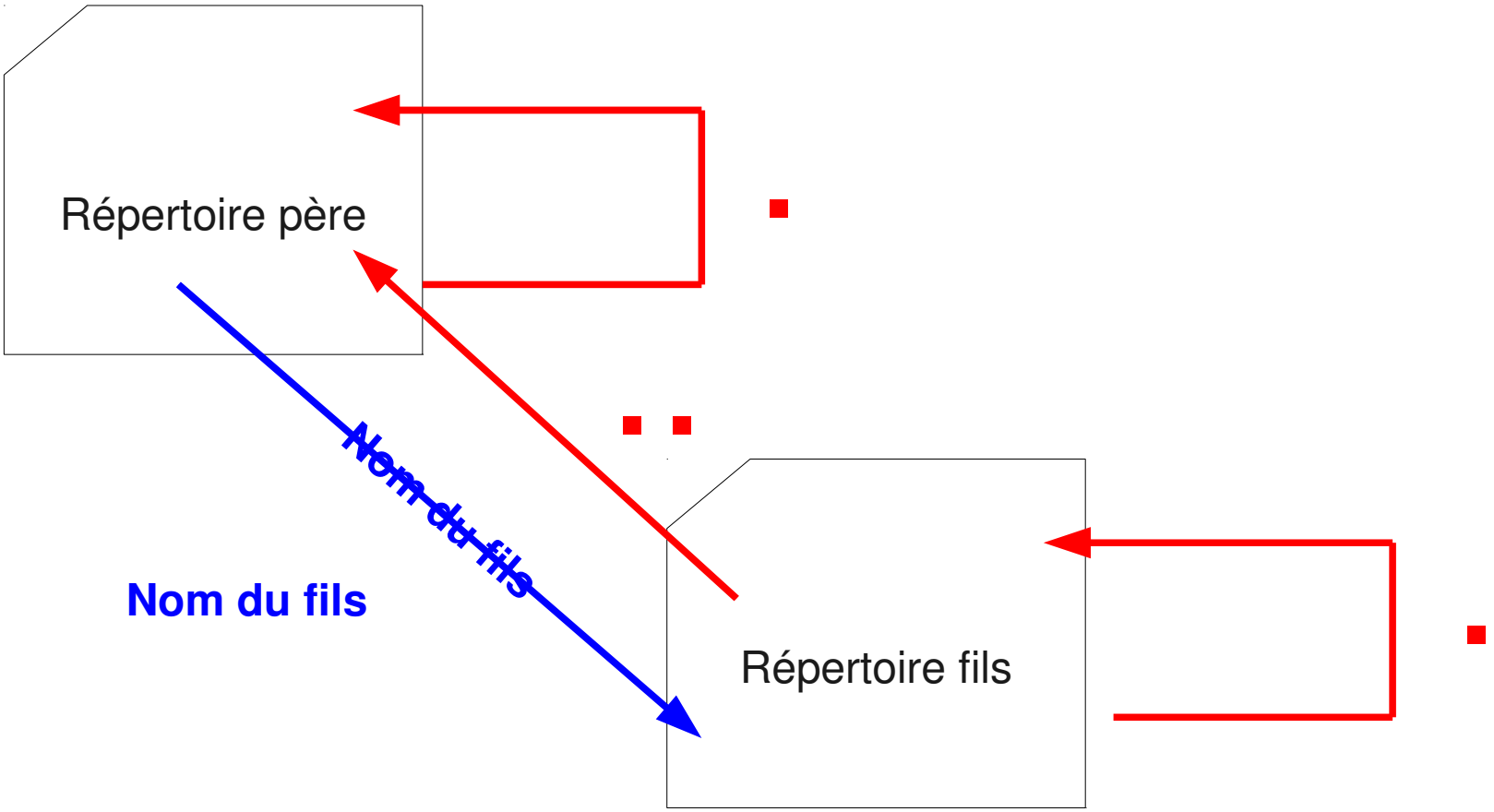
- Le répertoire contient les noms de fichiers ainsi que l'inode correspondant permettant d'y accéder
  Cette association est un lien

```
$ vi repertoire1
"
==========================================================
===============
" Netrw Directory Listing                                 (netrw v98)
"   /home/franck/UNIX/repertoire1
"   Sorted by      name
"   Sort sequence: [\/]$,*,\.bak$,\.o$,\.h$,\.info$,\.swp$,\.obj$
"   Quick Help: <F1>:help  -:go up dir  D:delete  R:rename  s:sort-by  x:exec
"
==========================================================
===============
../
./
tata
titi
toto
```

- Un répertoire peut renvoyer sur un autre répertoire et ainsi de suite
  - le système de fichier unix peut être ainsi représenté par un arbre où chaque noeud correspond à un répertoire et chaque feuille à un fichier
  - la racine de l'arbre s'appelle « / » (« slash »)



- Il existe une notion de « descendance » dans les répertoires



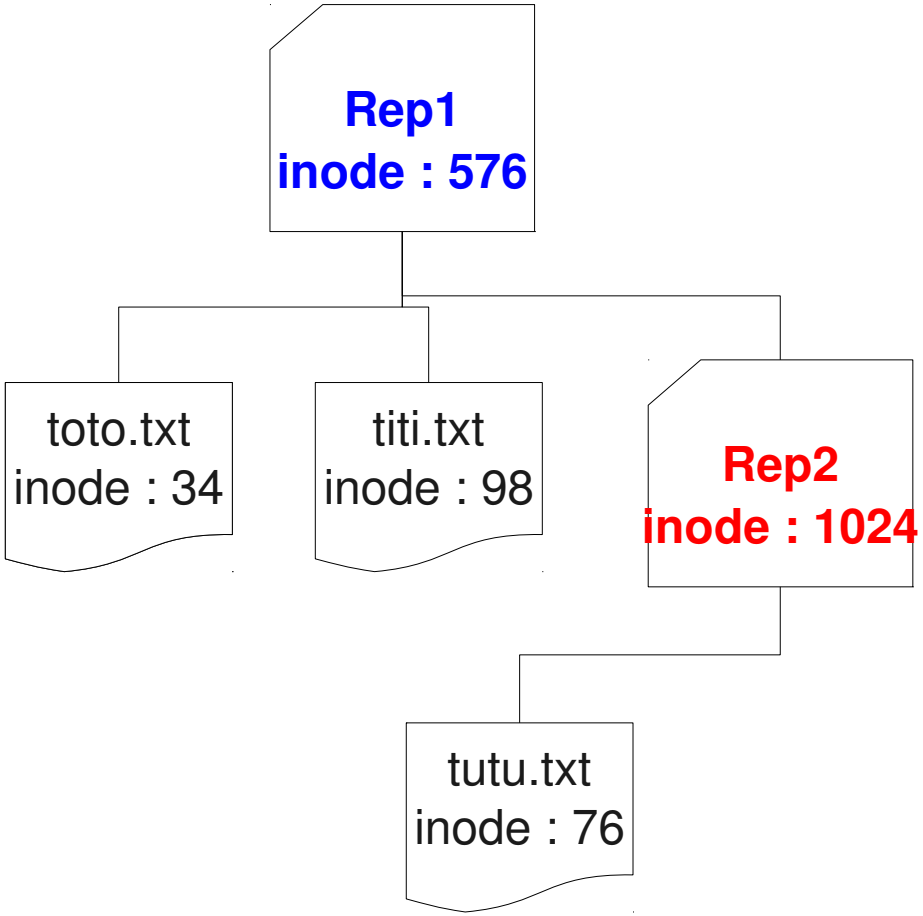
- Donc où se trouve stocké le nom d'un répertoire s'il ne se trouve pas dans l'inode ?...

Rep1

.	**576**
..	9999
toto.txt	34
titi.txt	98
Rep2	**1024**

Rep2

.	**1024**
..	**576**
tutu.txt	76



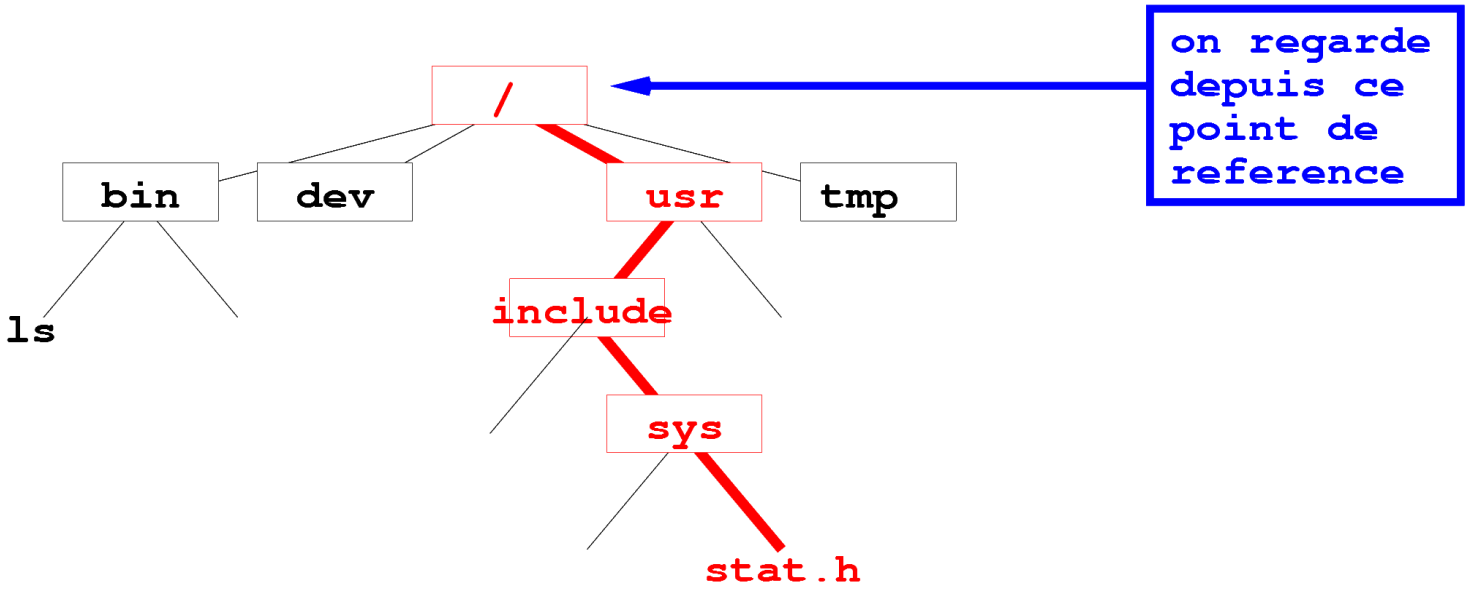
Notions de chemins

- On accède à un fichier à travers son « chemin » dans l'arborescence
  - le chemin est constitué d'une liste de noms de répertoires et se termine par le nom du fichier
  - le caractère « / » permet de séparer les différents répertoires qui constituent le chemin

```
/repertoire1/répertoire2/fichier
```

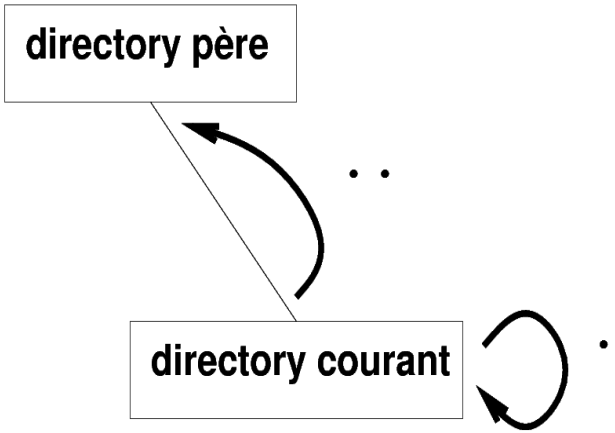
Notions de chemins

- Chemin d'accès « **absolu** »
  - C'est le chemin qui permet d'accéder à un fichier et **qui commence par la racine de l'arbre**
  - **Un chemin absolu doit toujours commencer par « / »**
  - exemple : « /usr/include/sys/stat.h »



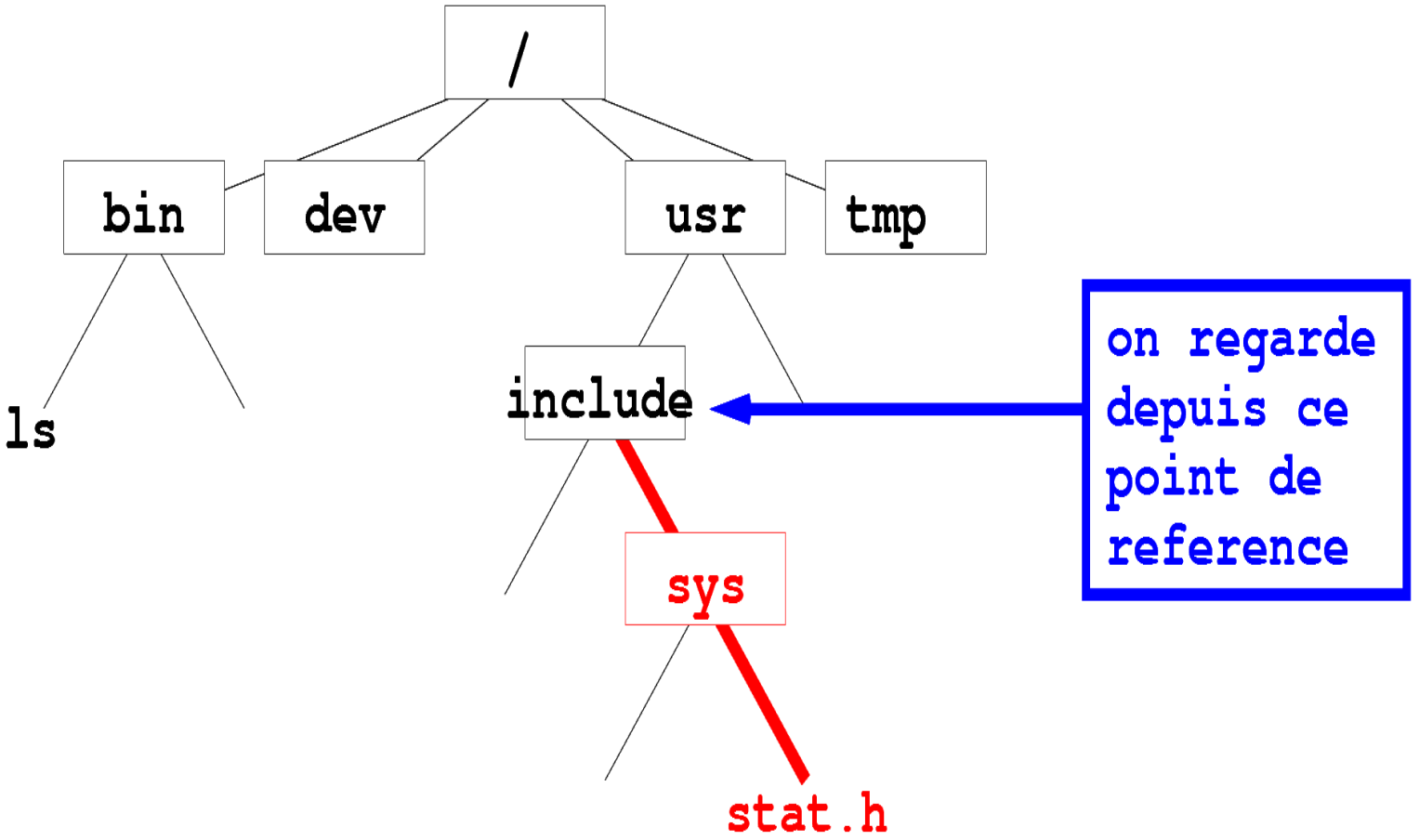
Notions de chemins

- Chemin d'accès « **relatif** »
  - C'est le chemin qui permet d'accéder à un fichier et **qui peut commencer à n'importe quel endroit de l'arbre excepté la racine**
  - **Un chemin relatif est « relatif » à la position de référence (le répertoire courant)**
  - le répertoire courant est noté « . »
  - le répertoire parent du répertoire courant est noté « .. »

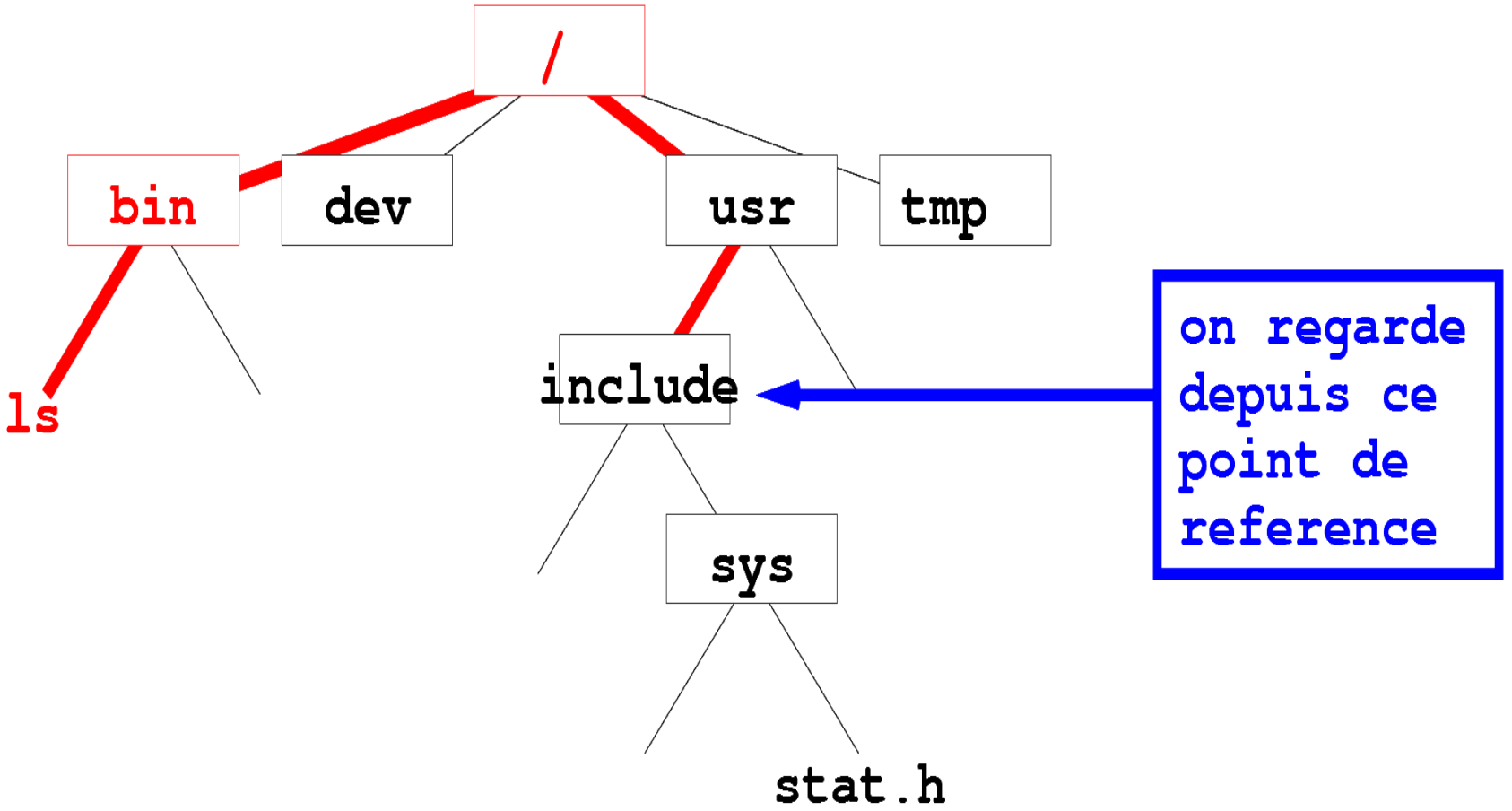


Notions de chemins

- exemple 1 : depuis « /usr/include/ », le chemin relatif du fichier « stat.h » est « sys/stat.h »



- exemple 2 : depuis « /usr/include/ », le chemin relatif du fichier « ls » est « ../../bin/ls »



Notions de chemins

- Importance des écritures « . » et « .. »
  - commande « `find` » pour lancer une recherche à partir de l'endroit courant

```
$ find . -name fichier -print
```

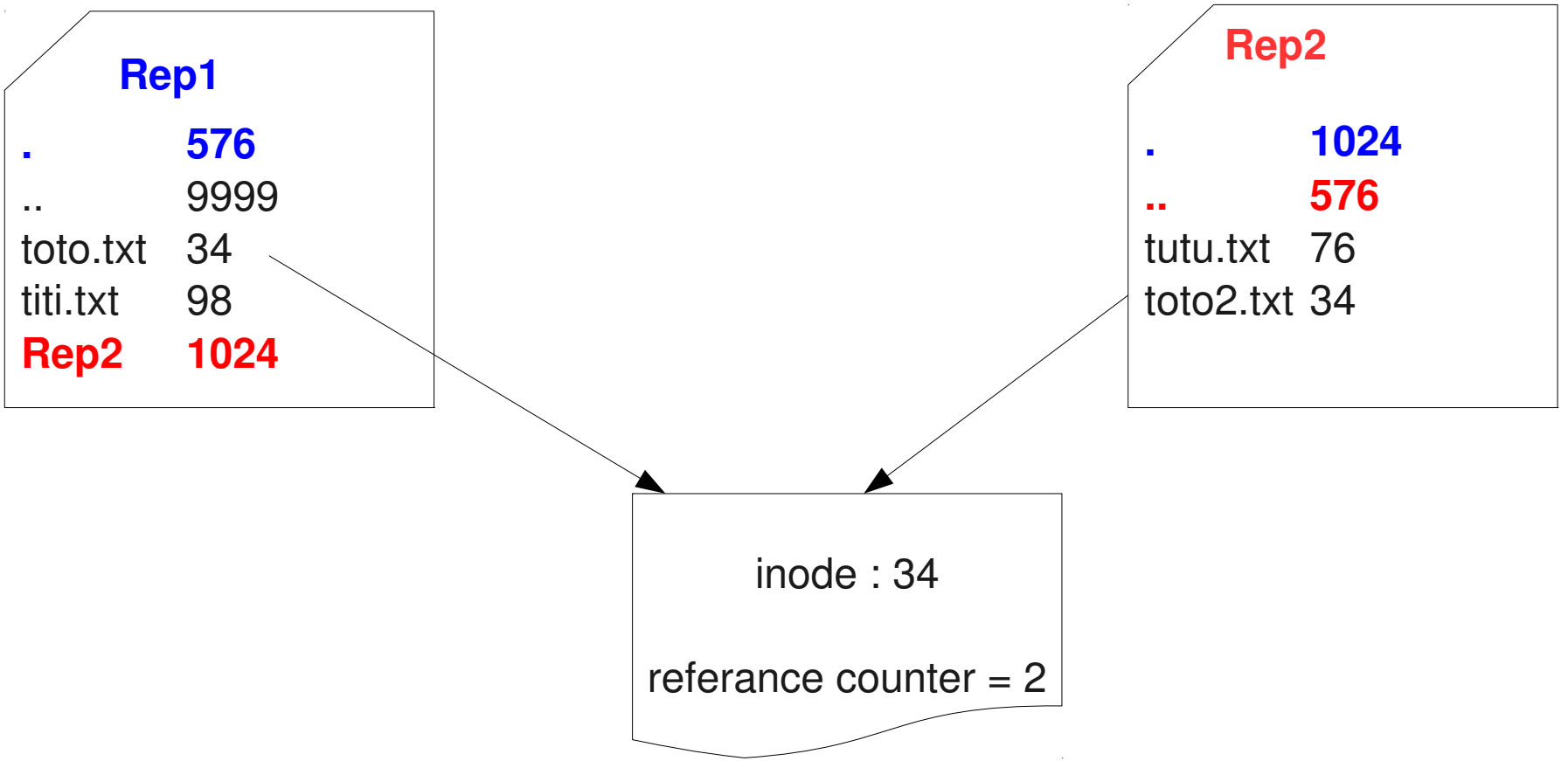
  - pour lancer une commande qui se trouve dans le répertoire courant

```
$ ./macommande
```

- Comparaison Windows/Unix
  - Windows : plusieurs volumes « C:, D: »; « \ » comme séparateur de répertoires
  - Unix : une arbre unique; « / » comme séparateur de répertoires

Liens sur objets

- Rappel : les noms sont stockés dans les répertoires
- un nom est appelé *un lien* sur l'objet
- Sur Unix, chaque objet peuvent être associés plusieurs noms



Liens sur objets

```
$ file unix
unix: directory

$ less unix
```

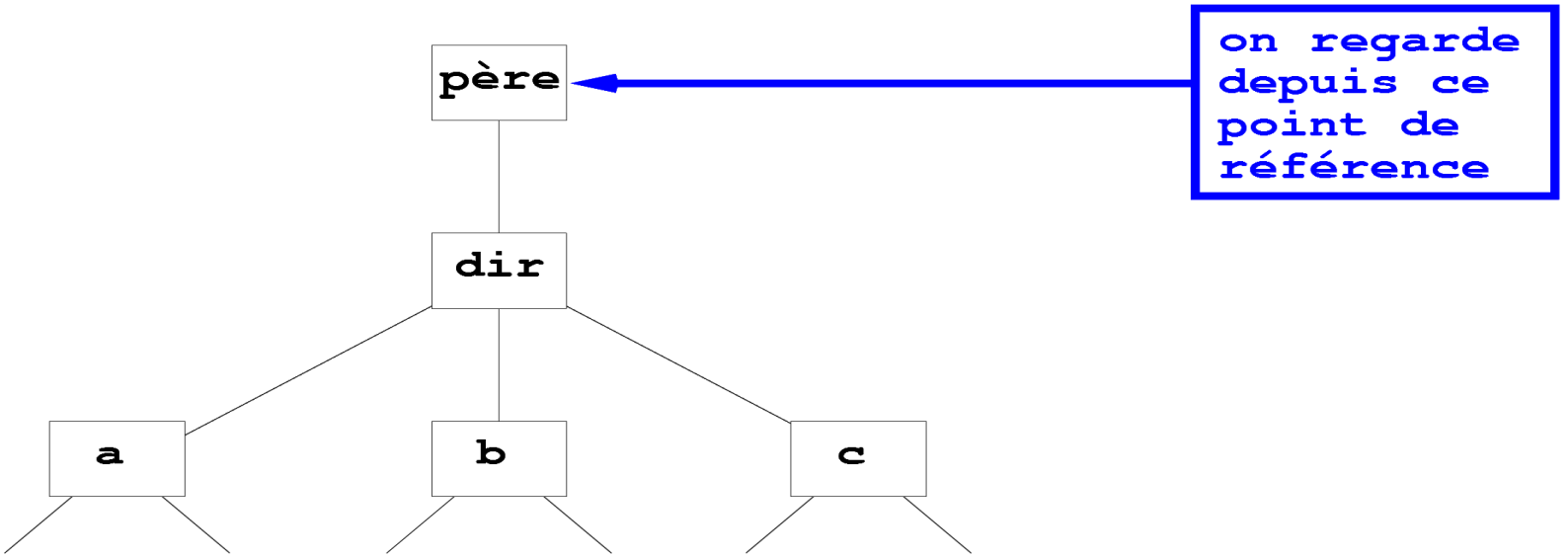
```
total 40
drwxrwxr-x  4 franck franck  4096 jan 28 09:58 ./
drwxrwxr-x 91 franck franck 12288 mar  9 23:53 ../
-rwxrwxr-x  1 franck franck   161 jui  3  2008 creuser.sh*
-rwxrwxr-x  1 franck franck    74 mar 28  2008 liste2.sh*
-rw-rw-r--  1 franck franck    60 jui  3  2008 liste.txt
drwx----wx  2 franck franck  4096 mai  8  2008 rep/
drwxrwxr-x  2 franck franck  4096 mai  8  2008 rep2/
unix (END)
```

Liens sur objets

- Dans l'inode d'un objet, il y a un compteur de liens :
  - compteur incrémenté lors de la création d'un nouveau lien
  - compteur décrémenté lors de la suppression d'un lien
  - l'objet est détruit lorsque le dernier lien sur l'objet est supprimé
  - « . » et « .. » sont des liens

```
[aoi@test]$ ls -l
total 5
-rw-r--r-- 3 franck franck  111 mar 13 19:34 c
-rw-r--r-- 3 franck franck  111 mar 13 19:34 c-1
-rw-r--r-- 3 franck franck  111 mar 13 19:34 c-2
-rw-r--r-- 1 franck franck   40 mar 13 19:26 d
drwxr-xr-x 2 franck franck 1024 mar 15 15:56 e/
```

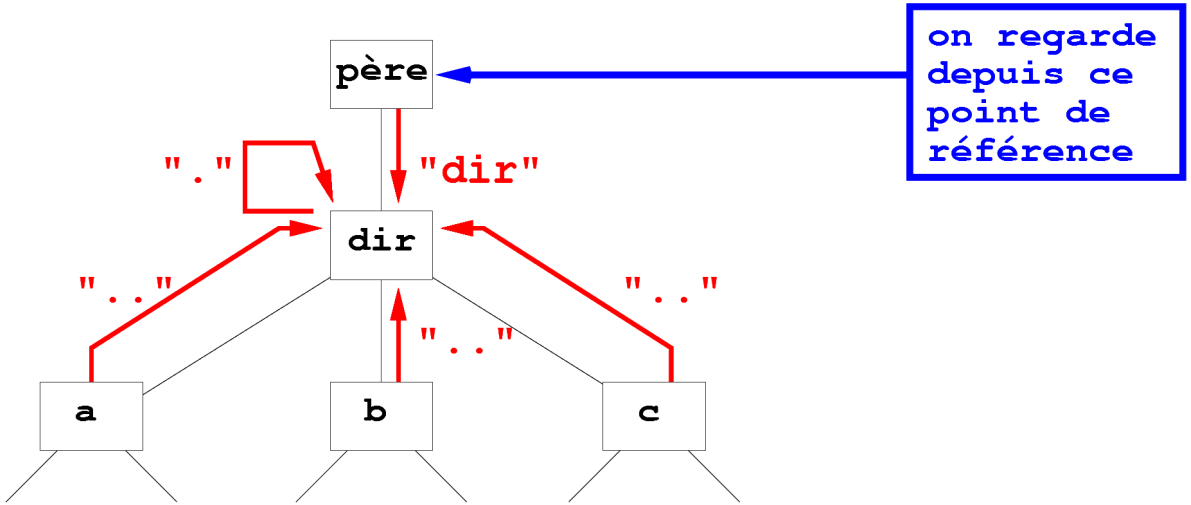
Liens sur objets




```
[aoi@test]$ ls -l dir
total 3
drwxr-xr-x 2 franck franck 1024 mar 16 23:23 a/
drwxr-xr-x 2 franck franck 1024 mar 16 23:23 b/
drwxr-xr-x 2 franck franck 1024 mar 16 23:23 c/
[aoi@test]$ ls -ld dir
drwxr-xr-x 5 franck franck 1024 mar 16 23:23 dir/
```

Liens sur objets

- Il y a en effet 5 liens sur l'objet « `dir` » :



- `lien « /chemin/dir »`
- `lien « /chemin/dir/. »`
- `lien « /chemin/dir/a/.. »`
- `lien « /chemin/dir/b/.. »`
- `lien « /chemin/dir/c/.. »`

Liens sur objets

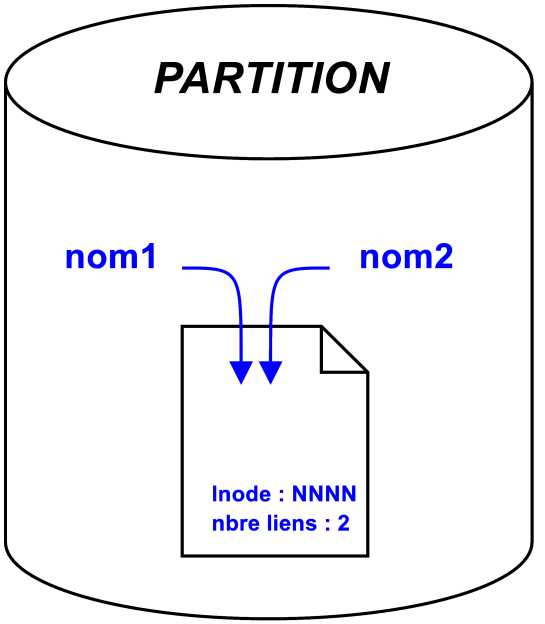
- la commande « `ls -li` » permet de vérifier que l'inode correspondant à chaque nom relève bien du même fichier

```
[aoi@test]$ ls -ldi dir dir/. dir/a/.. dir/b/.. dir/c/..
2165772 drwxr-xr-x 5 franck franck 1024 mar 16 23:23 dir/
2165772 drwxr-xr-x 5 franck franck 1024 mar 16 23:23 dir/./
2165772 drwxr-xr-x 5 franck franck 1024 mar 16 23:23 dir/a/../
2165772 drwxr-xr-x 5 franck franck 1024 mar 16 23:23 dir/b/../
2165772 drwxr-xr-x 5 franck franck 1024 mar 16 23:23 dir/c/../
```

- Il existe deux types de liens
  - lien **hard**
  - lien **symbolique**

Liens hard : ln

- Le lien **hard** utilise le numéro d'inode pour identifier l'objet
  - un numéro unique par partition
  - confiné à l'intérieur d'une même partition (unicité de l'inode)
- Contraintes
  - hard link impossible vers une autre partition (risque de perte de l'unicité de l'inode)
  - hard link impossible vers un répertoire (risque de boucles invisibles dans l'arborescence)
- Utilisation courante : liens hard dans un environnement « chrooté »



Liens hard : ln

- Commande « ln » (link)
  - ln original synonyme

```
[aoi@test]$ ls -l fichier1
-rw-r--r-- 1 franck franck 0 mar 17 08:31 fichier1

[aoi@test]$ ln fichier1 fichier2

[aoi@test]$ ls -l fichier1 fichier2
-rw-r--r-- 2 franck franck 0 mar 17 08:31 fichier1
-rw-r--r-- 2 franck franck 0 mar 17 08:31 fichier2

[aoi@test]$ ls -li fichier1 fichier2
1666053 -rw-r--r-- 2 franck franck 0 mar 17 08:31
fichier1
1666053 -rw-r--r-- 2 franck franck 0 mar 17 08:31
fichier2
```

Liens hard : ln

- Copies et déplacement vers un lien

```
[aoi@test]$ cp c fichier1
cp: écraser `fichier1'?y

[aoi@test]$ ls -li fichier1 fichier2
1666053 -rw-r--r-- 2 franck franck 111 mar 17 08:34 fichier1
1666053 -rw-r--r-- 2 franck franck 111 mar 17 08:34 fichier2

[aoi@test]$ ls -li fichier3
1666055 -rw-r--r-- 1 franck franck 136 mar 17 08:43 fichier3

[aoi@test]$ mv fichier3 fichier2
mv: écraser `fichier2'?y

[aoi@test]$ ls -li fichier1 fichier2
1666053 -rw-r--r-- 1 franck franck 111 mar 17 08:34 fichier1
1666055 -rw-r--r-- 1 franck franck 136 mar 17 08:43 fichier2
```

Liens hard : ln

- Différence entre copie et lien

```
$ ls -i toto
291303 toto

$ cp toto titi

$ ll -i toto titi
291301 -rw-r--r-- 1 franck franck 24 2010-03-24 15:20 titi
291303 -rw-r--r-- 1 franck franck 24 2010-03-24 15:20 toto


$ ln toto lien-toto

$ ll -i toto lien-toto
291303 -rw-r--r-- 2 franck franck 24 2010-03-24 15:20 lien-
toto
291303 -rw-r--r-- 2 franck franck 24 2010-03-24 15:20 toto
```

Liens hard : ln

- Suppression d'un lien par rm

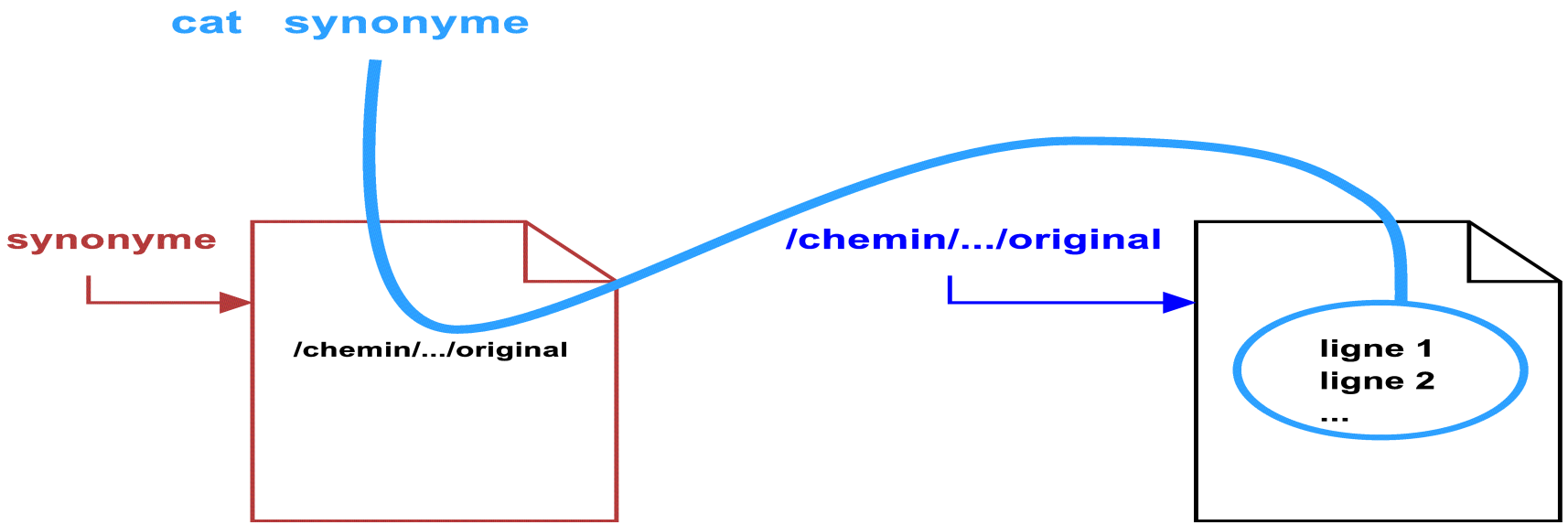
```
[aoi@test]$ ls -li fichier1 fichier2
1666053 -rw-r--r-- 2 franck franck 111 mar 17 08:34 fichier1
1666053 -rw-r--r-- 2 franck franck 111 mar 17 08:34 fichier2

[aoi@test]$ rm fichier1

[aoi@test]$ ls -li fichier1 fichier2
ls: fichier1: Aucun fichier ou répertoire de ce type
1666053 -rw-r--r-- 1 franck franck 111 mar 17 08:34 fichier2
```

- Place occupée (les liens ne prennent pas de place)

```
[aoi@test]$ ls -la
total 4
drwxr-xr-x 2 franck franck 1024 mar 17 08:56 ./
drwxr-xr-x 5 franck franck 1024 mar 17 08:56 ../
-rw-r--r-- 2 franck franck  111 mar 17 08:34 fichier1
-rw-r--r-- 2 franck franck  111 mar 17 08:34 fichier2
[aoi@test]$ du -b
1135    .
[aoi@test]$ rm fichier2
[aoi@test]$ du -b
1135    .
```

Liens symboliques : ln -s

- Le lien **symbolique** est un fichier spécial contenant le chemin d'accès à un autre objet
  - un numéro unique par partition
- Utilisation courante :
  - fichier commun à plusieurs utilisateurs sur une partition différente
  - liens sur des répertoires (cf. pb de filesystem saturé)



Liens symboliques : ln -s

- Commande « ln » (link)
  - ln **-s** original synonyme

```
[aoi@test]$ ls -l
total 1
-rw-r--r-- 1 franck franck 294 mar 18 22:43 fichier1

[aoi@test]$ ln -s fichier1 fichier2

[aoi@test]$ ls -li fichier1 fichier2
1674243 -rw-r--r-- 1 franck franck 294 mar 18 22:43 fichier1
1674242 lrwxrwxrwx 1 franck franck   8 mar 18 22:45 fichier2 ->
fichier1

[aoi@test]$ ls -lL fichier1 fichier2
-rw-r--r-- 1 franck franck 294 mar 18 22:43 fichier1
-rw-r--r-- 1 franck franck 294 mar 18 22:43 fichier2
```

Liens symboliques : ln -s

- Suppression d'un lien symbolique « rm »

```
[aoi@test]$ ls -li fichier1 fichier2
1674243 -rw-r--r-- 1 franck franck 294 mar 18 22:43 fichier1
1674242 lrwxrwxrwx 1 franck franck   8 mar 18 22:45 fichier2 ->
fichier1

[aoi@test]$ rm fichier1
rm: détruire fichier régulier `fichier1'? y

[aoi@test]$ ls -li fichier2
1674242 lrwxrwxrwx 1 franck franck 8 mar 18 22:45 fichier2 ->
fichier1

[aoi@test]$ ls -liL fichier2
ls: fichier2: Aucun fichier ou répertoire de ce type
[aoi@test]$ cat fichier2
cat: fichier2: Aucun fichier ou répertoire de ce type
```

Liens symboliques : ln -s

- Les systèmes Unix imposent les droits « `lrwxr-xr-x` » ou « `lrwxrwxrwx` » sur le lien
  - ils ne peuvent être modifés
  - on peut seulement modifier les droits du fichier pointé par un lien symbolique

```
[aoi@test]$ ls -l fichier1 fichier2
-rw-r--r-- 1 franck franck 307 mar 18 23:03 fichier1
lrwxrwxrwx 1 franck franck   8 mar 18 22:45 fichier2 ->
fichier1

[aoi@test]$ chmod 600 fichier2

[aoi@test]$ ls -l fichier1 fichier2
-rw------- 1 franck franck 307 mar 18 23:03 fichier1
lrwxrwxrwx 1 franck franck   8 mar 18 22:45 fichier2 ->
fichier1
```

Utilisation des liens

- En pratique les liens symboliques sont plus faciles d'utilisation que les liens hard
  - Scripts System V
  - Permet de déplacer le contenu d'un répertoire de l'arborescence vers une autre partition tout en conservant les chemins

Quelques conseils LPI...

- Connaître l'utilisation des commandes décrites
- Connaître l'utilisation de l'option -g pour que ces commandes agissent sur les quotas de groupe