

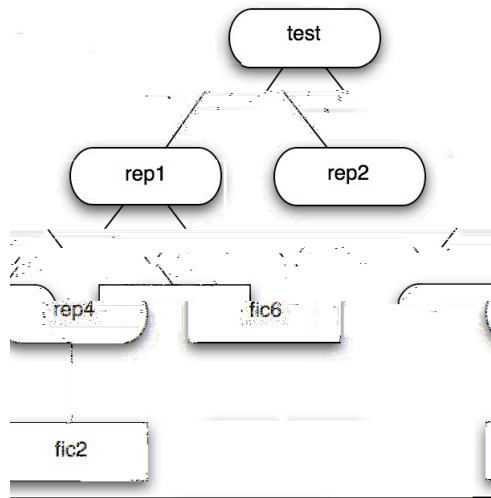
Introduction aux systèmes d'exploitation (IS1)

TP 3 – Liens, i-nœuds

Dans ce TP, nous allons explorer un peu plus en détail les possibilités du système de fichier UNIX à travers le concept de *lien* et la notion de numéro d'*i-nœud*.

Exercice 1 – Préparons le terrain.

Pour commencer, créez l'arborescence suivante dans votre dossier ~/IS1.



1. À l'aide de la commande `echo`, écrivez `Il fait beau aujourd'hui` dans le fichier `fic6`.
2. À l'aide de la commande `cat` affichez le contenu du fichier `fic6` depuis le répertoire `rep2`.
3. L'option `-l` de la commande `ls` permet entre autre d'observer les droits d'un fichier/répertoire. À quoi correspondent les autres informations que l'on obtient grâce à cette commande ?
4. À l'aide de la commande `ls`, vérifiez que l'arborescence que vous avez créée est bien conforme au schéma.

Correction.

```
1.~$ echo "Il fait beau aujourd'hui" > test/rep1/fic6
2.~$ cat ../rep1/rep4/fic6
3.~$ ls -l test/rep1/
total 0
-rw-r--r--  1 login  group    0 Sep 30 16:19  fic6
drwxr-xr-x  2 login  group 102 Sep 30 16:18  rep4
```

Les informations sont, de gauche à droite :

- les droits,
- le nombre de liaisons physique au fichier ou répertoire,
- le propriétaire,
- le groupe,
- la taille,
- la date de la dernière modification,
- le nom,
- en cas de lien symbolique "`->`" et la cible du lien.

Liens

Un système UNIX n'identifie pas un fichier par son nom, mais par une structure appelée *i-nœud* (*inode* en anglais), elle-même identifiée par un numéro. Un i-nœud permet, en plus de stocker le contenu d'un fichier, de se souvenir des permissions de ce fichier ou de la date de dernière modification, par exemple. Lorsqu'un fichier *fic* "se trouve" dans un répertoire, en réalité c'est que le répertoire contient le nom *fic* associé au numéro de l'i-nœud correspondant. On appelle cela un *lien* du nom *fic* vers l'i-nœud.

Il est tout à fait possible de créer plusieurs liens vers un même i-nœud, ce qui aura pour effet de créer plusieurs noms pour un même fichier, mais le contenu du fichier ne serait stocké qu'une fois (puisqu'il est stocké dans l'i-nœud).

La commande `ln` sert à créer des *liens*. Il existe deux types de liens, les liens dits *physiques* (*hard links* en anglais), et les *liens symboliques* (*soft links* en anglais). Un lien physique permet de partager l'i-nœud d'un fichier, tandis qu'un lien symbolique est un raccourci vers un chemin donné.

Elle s'utilise de façon similaire à la commande `cp` :

`ln [-s] source destination`

- *source* est le nom du fichier sur lequel on crée un lien.
- *destination* est le nom du lien.
- L'option `-s` permet la création d'un lien symbolique. Par défaut, le lien créé est physique.

Exercice 2 – Créons des liens sur des fichiers.

1. Comme expliqué ci-dessus, la commande « `ln` » sert à créer des *liens*. Utilisez la pour créer un lien **physique** du fichier *fic6* dans *test* sous le nom de *lpfic6*.
2. Modifiez avec un éditeur de texte (ou avec `echo`) le fichier *lpfic6*. Que constatez-vous pour le fichier *fic6* ? Réciproquement, modifiez *fic6*, lisez *lpfic6*. Concluez.
3. Modifiez les droits d'accès au fichier *fic6* pour les membres du groupe. Que constatez-vous pour *lpfic6* ? Pouvez-vous avancer une explication ?
4. La commande « `ln` » peut aussi créer des liens *symboliques* avec l'option `-s`. Créez un lien symbolique du fichier *fic6* dans *test* que vous appellerez *lsfic6*.
5. Regardez toutes les informations concernant les fichiers *lpfic6* et *lsfic6*. Quelles différences notez-vous ?
6. Essayez de modifier les droits d'accès à *lsfic6*. Que constatez-vous ?
7. Modifiez les droits d'accès au répertoire *rep1* pour ne plus y avoir accès. Essayez d'afficher le contenu de *lpfic6* et *lsfic6*. Que constatez-vous ? Pouvez-vous avancer une explication ?
8. Modifiez de nouveau les droits d'accès au répertoire *rep1* pour y avoir de nouveau accès. Supprimez le fichier *fic6* dans le répertoire *test*. Essayez d'afficher le contenu de *lpfic6* et *lsfic6*. Que constatez-vous ?
9. Créez un nouveau fichier ayant nom *fic6* dans le répertoire *rep1* et écrivez la phrase nouveau fichier dans ce fichier. Tapez `cat lsfic6` depuis le répertoire *test*. Que constatez-vous ?

Correction.

```
1.~$ ln test/rep1/fic6 test/lpfic6
2.~$ echo "ecrit dans lpfic6" > test/lpfic6
~$ cat test/rep1/fic6
ecrit dans lpfic6
~$ echo "ecrit dans fic6" >> test/rep1/fic6
~$ cat test/lpfic6
ecrit dans lpfic6
ecrit dans fic6
On conclut que test/rep1/fic6 et test/lpfic6 sont deux noms qui se réfèrent au même espace dans la mémoire.
```

```

3.~$ chmod g-r test/rep1/fic6
~$ ls -l test/rep1/
total 8
-rw---r--  2 login  group   34 Sep 30 17:16 fic6
drwxr-xr-x  3 login  group  102 Sep 30 16:18 rep4
~$ ls -l test/
total 8
-rw---r--  2 login  group   34 Sep 30 17:16 lpfic6
drwxr-xr-x  4 login  group  136 Sep 30 16:19 rep1
drwxr-xr-x  2 login  group   68 Sep 30 16:17 rep2

```

Les fichiers test/rep1/fic6 et test/lpfic6 partagent le même i-nœud et donc ont les mêmes droits d'accès.

```

4.~$ cd test
~/test$ ln -s rep1/fic6 lsfic6

```

```

5.~$ ls -l test/
total 16
-rw---r--  2 login  group   34 Sep 30 17:16 lpfic6
lrwxr-xr-x  1 login  group   14 Sep 30 17:32 lsfic6@ -> test/rep1/fic6
drwxr-xr-x  4 login  group  136 Sep 30 16:19 rep1/
drwxr-xr-x  2 login  group   68 Sep 30 16:17 rep2/

```

Les permissions indiquées pour le lien symbolique ne sont pas les mêmes que sa cible. Les permissions n'ont en fait aucune signification pour un lien symbolique : ce sont celles de la cible qui sont prises en compte. Remarquons que la première lettre des dix caractères concernant les permissions est un "l", comme "lien".

```

6.~$ chmod g+rw test/lsfic6
~$ ls -l test/
total 16
-rw-rw-r--  2 login  group   34 Sep 25 17:16 lpfic6
lrwxr-xr-x  1 login  group   14 Sep 25 17:32 lsfic6@ -> test/rep1/fic6
drwxr-xr-x  4 login  group  136 Sep 25 16:19 rep1/
drwxr-xr-x  2 login  group   68 Sep 25 16:17 rep2/

```

Ce sont les permissions du fichier ciblé qui ont été modifiées.

```

7.~$ chmod u-rx test/rep1
~/test$ cat lsfic6
cat: lsfic6: Permission denied

```

Un lien symbolique ne fonctionne pas si sa cible se trouve dans un répertoire non accessible à l'utilisateur.

```

8.~$ chmod u+rx test/rep1
~/test$ rm rep1/fic6
~/test$ cat lsfic6
cat: lsfic6: No such file or directory
~/test$ cat lpfic6
ecrit dans lpfic6
ecrit dans fic6

```

Le lien lsfic6 est maintenant interrompu (*broken link*). Le fichier lpfic6 est inchangé : le bloc mémoire qui contient le texte écrit dans lpfic6 est toujours referé par l'i-nœud du fichier lpfic6.

```

9.~/test$ touch rep1/fic6
~/test$ echo "nouveau fichier" > rep1/fic6
~/test$ cat lsfic6
nouveau fichier

```

Le lien symbolique lsfic6 peut de nouveau être utilisé, pour accéder au nouveau fichier rep1/fic6.

Exercice 3 – Créons des liens sur des répertoires

1. Observons maintenant plus attentivement le répertoire rep2. Combien y a-t-il de liens sur ce répertoire ? À quoi correspondent ils ?
2. Dans le répertoire rep2, créez un sous-répertoire rep3. Combien y a-t-il maintenant de liens sur le répertoire rep2 ? Expliquez.
3. Créez un lien physique lprep4 du répertoire rep4 dans le répertoire test. Que remarquez-vous ?
4. Créez un lien symbolique lsrep4 du répertoire rep4 dans le répertoire test.
5. Créez un lien symbolique ls1srep4 du lien lsrep4 dans le répertoire rep2.
6. Donnez trois manières différentes de se déplacer dans le répertoire rep4 à partir du répertoire test. En utilisant successivement ces trois méthodes, déplacez vous dans le répertoire rep4 puis remontez dans le répertoire parent à l'aide de la commande `cd ..`. Que remarquez-vous ?
7. Que se passe-t-il si on utilise la commande `ls -R` sur lsrep4 ? et sur ls1srep4 ?

Correction.

```
1.~$ ls -l test/
total 16
-rw---r--  1 login  group   34 Sep 30 17:16 lpfic6
lrwxr-xr-x  1 login  group   14 Sep 30 17:32 lsfic6@ -> test/rep1/fic6
drwxr-xr-x  3 login  group  102 Sep 30 17:52 rep1/
drwxr-xr-x  2 login  group   68 Sep 30 16:17 rep2/
Il y a deux liens. Ils correspondent à celui qui se trouve dans test et à . qui se trouve dans rep2 lui-même.
```

```
2.~$ mkdir test/rep2/rep3
~$ ls -l test/
total 16
-rw---r--  1 login  group   34 Sep 25 17:16 lpfic6
lrwxr-xr-x  1 login  group   14 Sep 25 17:32 lsfic6@ -> test/rep1/fic6
drwxr-xr-x  3 login  group  102 Sep 25 17:52 rep1/
drwxr-xr-x  3 login  group  102 Sep 25 18:05 rep2/
Il y a un nouveau lien vers rep2, qui est dû au . qui se trouve dans rep3.
```

3.C'est impossible. Avec les implémentations existant actuellement, seul le super-utilisateur peut créer un lien matériel vers un répertoire, et encore, ce n'est pas toujours possible. Par contre, on peut tout à fait créer un lien symbolique sur un répertoire.

```
4.~/test$ ln -s rep1/rep4 lsrep4
```

```
5.~/test$ cd rep2
~/test/rep2$ ln -s ../lsrep4 ls1srep4
```

```
6. (a)~/test$ cd rep1/rep4
~/test/rep1/rep4$ cd ..
~/test/rep1$

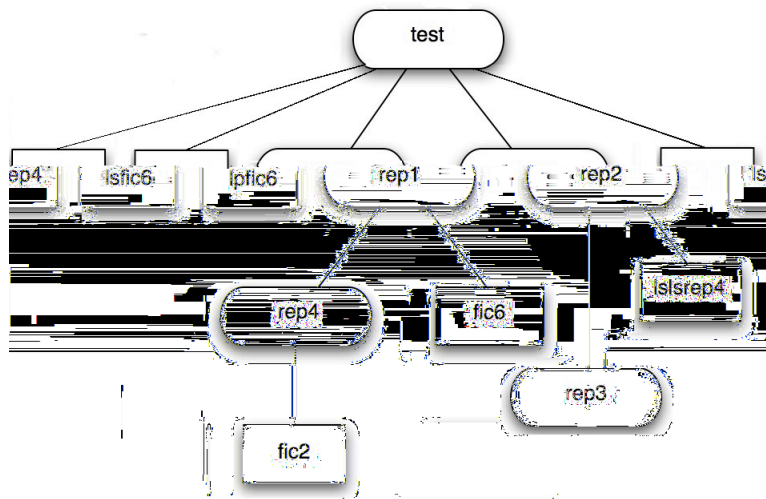
(b)~/test$ cd lsrep4
~/test/lsrep4$ cd ..
~/test$

(c)~/test$ cd rep2/ls1srep4
~/test/rep2/ls1srep4$ cd ..
~/test/rep2
```

On remarque que dans chaque cas, on se retrouve dans un répertoire différent après avoir tapé `cd ..`.

7.Seul le contenu du répertoire ciblé est affiché, et non pas les contenus de rep4, de lsrep4 et de ls1srep4 tous à la fois. (Ils ne sont pas considérés comme des répertoires imbriqués par `ls -R`.)

À la fin de cet exercice, nous avons maintenant l'arborescence suivante :



I-nœuds

Exercice 4 – I-nœuds

1. À l'aide de la commande `ls` munie de l'option `-i`, observez le numéro d'i-nœud du fichier `fic2`. Copiez le fichier `fic2` dans le répertoire `rep3`. Quel est son numéro d'i-nœud ?
2. Changez le nom de ce dernier fichier, pour l'appeler `fic6`. Le numéro d'i-nœud change-t-il ?
3. Créez dans `test` un lien physique et un lien symbolique de `rep1/rep4/fic2` sous les noms `lpfic2` et `lsfic2`. Comparez les numéros d'i-nœud de `rep1/rep4/fic2`, `lpfic2` et `lsfic2`. Que remarquez-vous ?
4. Observez maintenant le numéro d'i-nœud de votre répertoire personnel. Comparez-le avec le nombres d'i-nœud des derniers fichiers créés. Que remarquez vous ?

Correction.

```

1.~/test$ ls -i rep1/rep4/
11927984 fic2
timo@ukko:~/IS1/test$ cp rep1/rep4/fic2 rep2/rep3/
~/test$ ls -i rep2/rep3/
11927993 fic2
  
```

Le numéro d'i-nœud n'est pas le même car un nouveau fichier a été créé par la commande `cp`.

```

2.~/test$ mv rep2/rep3/fic2 rep2/rep3/fic6
~/test$ ls -i rep2/rep3/
11927993 fic6
  
```

Le numéro d'i-nœud n'a pas changé après renommage.

```

3.~/test$ ln rep1/rep4/fic2 lpfic2
~/test$ ln -s rep1/rep4/fic2 lsfic2
~/test$ ls -i rep1/rep4/fic2
11927984 rep1/rep4/fic2
~/test$ ls -i lpfic2
11927984 lpfic2
~/test$ ls -i lsfic2
11927996 lsfic2
  
```

Un fichier a le même numéro d'i-nœud qu'un lien physique vers lui même. En revanche, un lien symbolique n'a pas le même numéro d'i-nœud que sa cible.

4. Les numéros d'i-nœuds des derniers fichiers créés sont proches les uns des autres, et le numéro d'i-nœud du répertoire maison est beaucoup plus éloigné de ces derniers.