

GNU - Linux // Commande : ln

Lien Symbolique et Lien Physique.....	2
Création d'un fichier.....	2
Création d'un lien symbolique (Le Raccourci).....	2
Création d'un lien physique (Le Clone).....	3
Comprendre les inodes.....	3
Que ce passe t'il si je modifie le fichier aliasFichier ?.....	4
Que ce passe t'il si je supprime le fichier initial ?.....	4
Si je copie aliasFichier.txt dans dossierTest avec le nom fichier.txt.....	5

Lien Symbolique et Lien Physique

Création d'un dossier dossierTest.

```
user@lab-debian-27:~$ ls -l
total 0
```

```
user@lab-debian-27:~$ mkdir dossierTest
user@lab-debian-27:~$ ls -l
total 4
drwxrwxr-x 2 user user 4096 Jan 12 10:36 dossierTest
```

Création d'un fichier

Création d'un fichier fichier.txt contenant "Hello World"

```
user@lab-debian-27:~$ echo Hello World > dossierTest/fichier.txt
user@lab-debian-27:~$ ls -l dossierTest/
total 4
-rw-rw-r-- 1 user user 12 Jan 12 10:40 fichier.txt
user@lab-debian-27:~$ cat dossierTest/fichier.txt
Hello World
```

Note : "Nous partons d'un fichier standard. La commande `ls -l` nous montre que le compteur de liens (le chiffre après les permissions) est à **1**. Cela signifie que ce fichier n'a pour l'instant qu'un seul nom sur le disque."

Création d'un lien symbolique (Le Raccourci)

Ensuite, tu utilises la commande `ln -s`.

L'action : On crée `symbolicFichier.txt` qui pointe vers le fichier dans le dossier.

```
user@lab-debian-27:~$ ln -s dossierTest/fichier.txt symbolicFichier.txt
user@lab-debian-27:~$ ls -l
total 4
drwxrwxr-x 2 user user 4096 Jan 12 10:40 dossierTest
lrwxrwxrwx 1 user user   23 Jan 12 10:41 symbolicFichier.txt -> dossierTest/fichier.txt
```

L'analyse du résultat (`ls -l`) :

- On voit la lettre **l** au début de la ligne de permissions (`lrwxrwxrwx`), ce qui confirme que c'est un lien symbolique.
- On voit une flèche `->` qui indique explicitement le chemin vers le fichier cible (`dossierTest/fichier.txt`).

Note : "Le lien symbolique fonctionne comme un raccourci Windows. C'est un petit fichier qui contient simplement l'adresse de l'original. Si on regarde ses permissions, il a souvent tous les droits (`rw`), car les vrais droits qui comptent sont ceux du fichier cible."

Création d'un lien physique (Le Clone)

C'est l'étape cruciale pour comprendre la différence. On utilise `ln` sans option.

L'action : On crée `aliasFichier.txt` lié au même fichier source.

```
user@lab-debian-27:~$ ln dossierTest/fichier.txt aliasFichier.txt
user@lab-debian-27:~$ ls -l
total 8
-rw-rw-r-- 2 user user  12 Jan 12 10:40 aliasFichier.txt
drwxrwxr-x 2 user user 4096 Jan 12 10:40 dossierTest
lrwxrwxrwx 1 user user  23 Jan 12 10:41 symbolicFichier.txt -> dossierTest/fichier.txt
user@lab-debian-27:~$ cat aliasFichier.txt
Hello World
```

L'analyse du résultat (`ls -l`) :

- Le type de fichier est `-` (fichier ordinaire), tout comme l'original. Visuellement, rien ne le distingue d'un fichier classique.
- Le point clé :** Le compteur de liens est passé à **2** pour `aliasFichier.txt` ET pour le fichier original dans `dossierTest` (bien que ce ne soit pas affiché sur la capture, ils partagent ce compteur).

Note : "Ici, nous n'avons pas créé un raccourci, mais une seconde 'étiquette' sur les mêmes données. Le système voit maintenant deux noms valides pour accéder au même contenu physique sur le disque dur. C'est pour cela que le compteur de liens indique **2**."

Comprendre les inodes

```
user@lab-debian-27:~$ ls -li
total 8
1569799 -rw-rw-r-- 2 user user  12 Jan 12 11:29 aliasFichier.txt
1569798 drwxrwxr-x 2 user user 4096 Jan 12 11:29 dossierTest
1569800 lrwxrwxrwx 1 user user  23 Jan 12 11:30 symbolicFichier.txt -> dossierTest/fichier.txt
user@lab-debian-27:~$ ls -li dossierTest/
total 4
1569799 -rw-rw-r-- 2 user user 12 Jan 12 11:29 fichier.txt
```

« La commande `ls -li` (l pour long, i pour inode) nous révèle la carte d'identité réelle des fichiers via la première colonne de chiffres :

- Les Liens Physiques :** Regardez `aliasFichier.txt` et `dossierTest/fichier.txt`. Ils possèdent **exactement le même numéro d'inode (1569799)**. Pour le système Linux, cela prouve de manière irréfutable qu'ils sont **le même fichier**. Ils partagent le même emplacement sur le disque dur.
- Le Lien Symbolique :** Regardez `symbolicFichier.txt`. Il possède un numéro d'inode **différent (1569800)**. C'est la preuve que c'est un fichier distinct, indépendant. Il a sa propre existence sur le disque (c'est un petit fichier qui contient juste le chemin de la cible). »

Que ce passe t'il si je modifie le fichier aliasFichier ?

On modifie le lien physique pour voir l'impact.

L'action : Tu ajoutes une ligne de texte dans aliasFichier.txt.

```
user@lab-debian-27:~$ echo Welcome To My World >> aliasFichier.txt
user@lab-debian-27:~$ cat aliasFichier.txt
Hello World
Welcome To My World
user@lab-debian-27:~$ cat dossierTest/fichier.txt
Hello World
Welcome To My World
user@lab-debian-27:~$ cat symbolicFichier.txt
Hello World
Welcome To My World
```

Le résultat : Le texte apparaît aussi bien dans l'original (dossierTest/fichier.txt) que dans le lien symbolique.

Note : "Comme le lien physique et le fichier original pointent vers la même zone du disque dur, modifier l'un modifie instantanément l'autre. Le lien symbolique, lui, ne fait que lire le contenu de l'original, il voit donc aussi la modification."

Que ce passe t'il si je supprime le fichier initial ?

C'est ici que la différence saute aux yeux. On supprime l'original.

L'action : rm dossierTest/fichier.txt.

```
user@lab-debian-27:~$ rm dossierTest/fichier.txt
user@lab-debian-27:~$ ls -l dossierTest/
total 0
user@lab-debian-27:~$ ls -l
total 8
-rw-rw-r-- 1 user user  32 Jan 12 10:47 aliasFichier.txt
drwxrwxr-x 2 user user 4096 Jan 12 10:48 dossierTest
lrwxrwxrwx 1 user user  23 Jan 12 10:41 symbolicFichier.txt -> dossierTest/fichier.txt
user@lab-debian-27:~$ cat symbolicFichier.txt
cat: symbolicFichier.txt: No such file or directory
user@lab-debian-27:~$ cat aliasFichier.txt
Hello World
Welcome To My World
```

Conséquence pour le Lien Symbolique :

- Il devient "cassé". La commande cat retourne une erreur "No such file or directory".
- **Pourquoi ?** "L'adresse indiquée par le raccourci ne mène plus nulle part."

Conséquence pour le Lien Physique :

- Il fonctionne toujours parfaitement ! Le contenu est toujours là.
- Le compteur de liens est redescendu à 1.
- **Pourquoi ?** "Tant qu'il reste au moins un lien physique (une étiquette) pointant vers les données, Linux ne supprime pas le contenu du disque. Le fichier aliasFichier.txt est devenu l'unique propriétaire des données."

Si je copie aliasFichier.txt dans dossierTest avec le nom fichier.txt

```
user@lab-debian-27:~$ cp aliasFichier.txt dossierTest/fichier.txt
user@lab-debian-27:~$ ls -l
total 8
-rw-rw-r-- 1 user user 32 Jan 12 10:47 aliasFichier.txt
drwxrwxr-x 2 user user 4096 Jan 12 11:22 dossierTest
lrwxrwxrwx 1 user user 23 Jan 12 10:41 symbolicFichier.txt -> dossierTest/fichier.txt
user@lab-debian-27:~$ ls -l dossierTest/
total 4
-rw-rw-r-- 1 user user 32 Jan 12 11:22 fichier.txt
user@lab-debian-27:~$ cat dossierTest/fichier.txt
Hello World
Welcome To My World
user@lab-debian-27:~$ cat symbolicFichier.txt
Hello World
Welcome To My World
```

Pourquoi le liens symbolique est de nouveaux fonctionnel ?

« Le lien symbolique ne pointe pas vers le contenu physique du fichier (l'inode), mais uniquement vers son **chemin** (son adresse sous forme de texte : dossierTest/fichier.txt).

Lorsque nous avons supprimé le fichier original, le chemin ne menait plus nulle part : le lien était "cassé". En remettant un fichier (même une copie) exactement au même endroit et avec le même nom, le chemin redevient valide. Le lien symbolique fonctionne donc à nouveau immédiatement. Il ne "sait" pas que le fichier a été remplacé, il constate juste que l'adresse répond. »