

## Lab– Créer des liens durs et des liens symboliques

### Introduction

GNU/Linux fournit deux types différents de liens pour les fichiers au sein du système de fichiers: les liens durs et les liens logiciels (symboliques). Les deux offrent la possibilité d'avoir plus d'un nom de chemin qui fait référence au même fichier, mais chacun a ses propres avantages et inconvénients.

### Objectifs

**Partie 1:Créer des liens durs**

**Partie 2:Créer des liens symboliques**

### Contexte/scénario

En tant qu'administrateur, vous serez responsable de la maintenance des liaisons matérielles et logicielles du système. Vous devez savoir ce que sont ces fichiers, la différence entre les liens physiques et les liens logiciels et comment les créer.

### Ressources requises

Ordinateur équipé d'Ubuntu dans une machine virtuelle VirtualBox.

**Exécutez un terminal shell :**

### Partie 1 : Créer des liens durs :

Les fichiers de lien dur sont comme les fichiers normaux, sauf qu'ils partagent le même numéro d' inode avec un autre fichier. Cela signifie que bien que les noms de chemin des fichiers liés en dur puissent être différents, tout le reste à leur sujet est identique.

Pour comprendre les liens durs, il est utile de comprendre un peu comment le système de fichiers garde la trace des fichiers. Pour chaque fichier créé, il existe un bloc de données sur le système de fichiers qui stocke les métadonnées du fichier.

Les métadonnées sont des informations sur le fichier telles que les autorisations, la propriété et les horodatages.

Ces métadonnées sont appelées la table d'inode du fichier. La table d'inode comprend également des pointeurs vers les autres blocs du système de fichiers appelés blocs de données où les données sont stockées.

Chaque fichier sur une partition a un numéro d'identification unique appelé un numéro **d'inode**. La commande `ls -li` affiche le numéro d'inode d'un fichier.

**Exécutez les deux commandes suivantes :**

```
user@localhost:~$ touch gnu
```

```
user@localhost:~$ ls -li gnu
73798439 gnu
```

Ce qui définit un fichier n'est pas son nom, mais plutôt le numéro qui lui a été attribué par le système de fichier appelé **numéro d'inode** (index-node).

## GNU/Linux

Pour créer des liens durs, la commande `ln` est utilisée avec deux arguments (et sans l'option `-s`). Le premier argument est un nom de fichier existant à lier, appelé une cible, et le deuxième argument est le nouveau nom de fichier à lier à la cible.

La syntaxe de la commande `ln` :

```
ln TARGET hard_LINK_NAME
```

Lorsqu'un lien dur est créé, la valeur du nombre de liens est augmentée d'une valeur de un, l'exemple suivant illustre l'utilisation de la commande `ln` pour créer un fichier lien dur vers un autre fichier. Notez le nombre de liens (le nombre après les autorisations) avant et après la création du lien dur:

```
user@localhost:~$ touch profile.txt
user@localhost:~$ ls -l profile.txt
-rw-r--r-- 1 user user 0 Mai 24 16:24 profile.txt
user@localhost:~$ ln profile.txt myprofile.txt
user@localhost:~$ ls -il profile.txt myprofile.txt
123456 -rw-r--r-- 2 user user 0 Mai 24 16:24 profile.txt

123456 -rw-r--r-- 2 user user 0 Apr Mai 24 16:24 myprofile.txt
```

Lorsque vous affichez des fichiers de lien dur, la commande `ls -i` peut être utile pour valider que les fichiers partagent en fait un inode, l'option `-i` entraîne l'affichage du numéro d'inode devant les droits d'accès au fichier.

Exécutez la commande suivante avec les droits root pour créer un lien dur pour un répertoire :

```
[root@localhost ~]# ln /etc /myetc
```

```
ln: '/etc': hard link not allowed for directory
```

Aucun lien dur ne peut être établi vers des répertoires .

### Résumé :

#### Lien physique(dur)

Syntaxe de la commande

```
ln <nom_fic> <nouveau_nom_fic>
```

#### Caracteristiques :

Permet de donner plusieurs noms à un fichier ordinaire

Permet Pas pour les répertoires

Ne traverse pas les partitions

Un fichier est supprimé (détruit) quand TOUS ses liens dur (physiques) sont supprimés.

## Partie 2: Créer des liens symboliques :

Les liens logiciels, également appelés liens symboliques, sont des fichiers conçus pour pointer vers d'autres fichiers via un autre chemin. Par exemple, le fichier `/bin/systemd` est un lien symbolique vers le fichier `/lib/systemd/systemd`.

Prenez l'exemple des «**raccourcis**», il est courant pour les utilisateurs de placer des icônes sur le bureau qui, lorsque vous cliquez dessus, exécutent un programme ou ouvrent un fichier. Ces icônes sont en fait des fichiers qui pointent vers d'autres fichiers; en d'autres termes, ces icônes sont des liens logiciels.

À mesure que le système GNU/Linux évolue, l'organisation des programmes change également. Pour conserver une certaine compatibilité descendante pour les programmes et les processus, et pour faciliter la tâche des utilisateurs et des administrateurs, des liens symboliques sont créés.

Les liens symboliques sont très visuels par rapport aux liens dur. En effet, les liens logiciels ou symboliques se distinguent par leur type de fichier.

Pour créer un lien symbolique vers un fichier, utilisez la commande `ln` avec l'option `-s`. Le premier argument est le nom du fichier d'origine et le deuxième argument est le nom du lien symbolique à créer.

La syntaxe de la commande `ln -s` :

```
ln -s TARGET soft_LINK_NAME
```

Par exemple, le fichier `/bin/systemd` est un lien symbolique pointant vers le fichier `/lib/systemd/systemd`:

```
user@localhost:~$ ls -l /bin/systemd
lrwxrwxrwx 1 root root 20 Feb 28 21:03 /bin/systemd -> /lib/systemd/systemd
```

Notez que le premier caractère précédant les droits d'accès est la lettre **l** pour link:

Dans l'exemple suivant, nous allons créer un fichier appelé `file1.txt` et créer un fichier lien symbolique appelé `file2.txt`:

```
user@localhost:~$ touch file1.txt

user@localhost:~$ ln -s file1.txt file2.txt
user@localhost:~$ ls -l file*
-rw-rw-r-- 1 user user 9 May 9 02:48 file1.txt
lrwxrwxrwx 1 user user 9 May 9 02:49 file2.txt -> file1.tx
```

Contrairement aux liens dur, les fichiers de liens logiciels n'augmentent pas le nombre de liens dur associés à un fichier standard. Dans l'exemple ci-dessus, le nombre de liens pour le fichier `file1.txt` reste à 1 quel que soit le nombre de fichiers de liens symboliques créés pour faire référence à ce fichier. Rappelez-vous que le nombre de liens est le nombre suivant immédiatement le champ des droits d'accès.

Exécutez la commande suivante avec des droits élevés(root).

```
user@localhost:~$ sudo su
```

## GNU/Linux

indiquez votre mot de passe:

Password: (le mot de passe ne s'affichera pas à l'écran, c'est normal !!)

```
root@localhost/home/user:~#cd
```

Exécutez les commandes suivantes pour copier un fichier dans le répertoire / tmp, puis créez un lien symboliques vers ce fichier.

```
[root@localhost ~]# cp /etc/hosts /tmp

[root@localhost ~]# ln -s /tmp/hosts /var/hosts
[root@localhost ~]# ls -l /var/hosts
lrwxrwxrwx 1 root root 10 Apr 19 16:01 /var/hosts -> /tmp/hosts
[root@localhost ~]# ls -l /tmp/hosts
-rw-r--r-- 1 root root 172 Apr 19 16:01 /tmp/hosts
```

Affichez le contenu du fichier /tmp/hosts via le fichier /var/hosts en exécutant la commande suivante:

```
[root@localhost ~]# more /var/hosts
127.0.0.1          localhost                ::1
localhost ip6-localhost ip6-loopback      fe00::0
ip6-localnet                               ff00::0
ip6-mcastprefix
```

Supprimez le fichier /tmp/hosts et essayez de visualiser à nouveau le fichier /var/hosts en exécutant les commandes suivantes:

```
[root@localhost ~]# rm /tmp/hosts
rm: remove regular file '/tmp/hosts'? y
[root@localhost ~]# more /var/hosts
/var/hosts: No such file or directory
```

Notez l'étrange message d'erreur Aucun fichier ou répertoire de ce type. Le fichier /var/hosts existe, mais lorsqu'il suit le lien, le système ne peut pas trouver le fichier / tmp / hosts, ce qui entraîne le message d'erreur. C'est la principale faiblesse des liens logiciels.

Affichez le fichier /var/hosts en exécutant la commande suivante:

```
[root@localhost ~]# ls -l /var/hosts
lrwxrwxrwx 1 root root 10 Apr 19 16:01 /var/hosts -> /tmp/hosts
```

La couleur rouge dans le nom de fichier indique que quelque chose ne va pas; dans ce cas, on dit que le lien est cassé.

Des liens symboliques peuvent également être créés vers des répertoires. Par exemple, exécutez les commandes suivantes (remplacez user par votre login):

```
[root@localhost ~]# ln -s /home/user /tmp/user
[root@localhost ~]# ls -l /tmp/user
lrwxrwxrwx 1 root root 8 Mai 19 17:20 /tmp/user -> /home/user
```

### Résumé :

Liens logiciel (symbolique)

Syntaxe de la commande

```
ln -s <nom_fic> <nouveau_nom_fic>
```

### Caracteristiques :

Crée un raccourci

Traverse les partitions

Fonctionne aussi pour les répertoires

## *A vous de jouer...*

### Ouvrez un nouveau terminal shell

**Vous allez réaliser les tâches suivantes :**

**Tâche 1 :** Créez un répertoire dans `/home/user/Bureau` (remplacez user par votre login). Vérifiez que le répertoire a été correctement créé en le listant avec une commande appropriée.

**Tâche 2 :** Créez un fichier nommé **origin**.

- Ce fichier doit être placé dans le répertoire créé à l'étape précédente. Il peut être vide ou contenir du contenu.

**Tâche 3 :** Éditez le fichier **origin** via **nano** et ajoutez le texte suivant : "ceci est un test".

- Sauvegardez le fichier et quittez l'éditeur.
- Affichez ensuite le contenu du fichier **origin-dup**.

Que constatez-vous ? ( Vous devriez remarquer que le contenu de **origin-dup** est identique à celui de **origin**, car ils pointent vers le même fichier physique ).

**Tâche 4 :** Créez un lien symbolique nommé **lien-origin**.

- Le lien symbolique doit pointer vers le fichier **origin**.

**Tâche 5 :** Créez un lien dur nommé **origin-dup**.

- Ce lien dur doit pointer directement vers le fichier origin.
- Confirmez sa création en listant les détails du répertoire.

#### **Tâche 6 : Identifiez les numéros d'inode.**

- Affichez le numéro d'inode du fichier origin.
- Affichez également celui du lien dur origin-dup.
- Comparez les numéros d'inode des deux fichiers ( Vous devriez constater qu'ils partagent le même numéro d'inode, car ils sont des noms différents pour le même fichier physique).

#### **Tâche 7 : Modifiez les droits d'accès de origin-dup.**

- Appliquez les droits (rwxr-xr-x) au fichier aorigin-dup puis comparez les droits d'accès entre origin et origin-dup.
  - Que constatez-vous ? ( Les modifications appliquées aux droits d'accès de origin-dup affectent également origin, car ils sont liés au même fichier physique).

### **Conclusion**

**Les liens durs** partagent le même numéro d'inode et font référence au même fichier physique. Toute modification sur l'un affecte automatiquement l'autre.

**Les liens symboliques**, en revanche, agissent comme des raccourcis et ne partagent pas le numéro d'inode avec le fichier d'origine.

