

Travaux pratiques – Créer des liens durs et des liens symboliques

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.

Il est important de comprendre les liens du système de fichiers, car un système de fichiers Linux typique contient par défaut bon nombre de ces fichiers. En tant qu'utilisateur régulier, vous pouvez régulièrement accéder à des fichiers qui sont en fait des liens symboliques. Comprendre le fonctionnement de ces fichiers vous permettra de les utiliser plus facilement.

Objectifs

Partie 1:Créer des liens symboliques

Partie 2:Créer des liens durs

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 14.04 Desktop LTS dans une machine virtuelle VirtualBox.

Ouvrez une fenêtre de terminal dans Ubuntu.

Connectez-vous à Ubuntu à l'aide des informations d'identification suivantes :

Utilisateur : admin

Mot de passe : admin



Cliquez sur l'icône du terminal pour ouvrir une fenêtre de terminal.



Partie 1: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 Linux grandit et se transforme, 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 logiciels sont très visuels par rapport aux liens dur. En effet, les liens logiciels ou symboliques se distinguent par leur type de fichier.

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

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

Dans une liste détaillée d'un lien symbolique, notez que le premier caractère précédant les autorisations est la lettre `l`:

Pour créer un fichier de lien logiciel, 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 à créer. La syntaxe de la commande `ln` :

```
ln -s TARGET LINK_NAME
```

les arguments doivent être placés dans le bon ordre, car il n'est pas possible de créer un lien à partir d'un nom de fichier qui existe déjà. Dans l'exemple suivant, nous allons créer un fichier appelé `file1.txt` et créer un fichier de lien logiciel appelé `file2.txt`:

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

admin@localhost:~$ ln -s file2.txt file1.txt
ln: failed to create symbolic link `file1.txt': File exists
```

Changez les arguments de l'exemple ci-dessus et un lien symbolique est créé avec succès:

```
admin@localhost:~$ ln -s file1.txt file2.txt
admin@localhost:~$ ls -l file*
-rw-rw-r-- 1 sysadmin sysadmin 0 May  9 02:48 file1.txt
lrwxrwxrwx 1 sysadmin sysadmin 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 associés à un fichier standard. Dans l'exemple ci-dessus, le nombre de liens pour le fichier `file1.txt` resterait à un, 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 autorisations:

```
sysadmin@localhost:~$ ls -l file*
-rw-r--r-- 1 admin admin May  9 14:39 file1.txt
lrwxrwxrwx 1 admin admin May  9 14:39 file2.txt -> file1.txt
```

Exécutez la commande suivante pour basculer vers le compte root. Lorsque vous y êtes invité, indiquez le mot de passe admin:

GBU/Linux

```
[sysadmin@localhost ~]$ sudo su
Password: admin (le mot de passe ne s'affichera pas à l'écran)
[root@localhost ~]#
```

Exécutez les commandes suivantes pour copier un fichier dans le répertoire / tmp, puis créez un lien logiciel 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
```

Notez que même s'il est facile de voir que /var/hosts pointe vers le fichier / tmp / hosts, il est impossible de déterminer un lien logiciel à partir du fichier pointé. En d'autres termes, le fichier / tmp / hosts ne vous indique pas que le fichier / var / hosts pointe vers lui. Affichez le 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 la liste du 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 logiciels peuvent également être créés vers des répertoires. Par exemple, exécutez les commandes suivantes:

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

Résumé :

Liens logiciel (symbolique)

Syntaxe de la commande

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

Crée un **raccourci**

Traverse les partitions

Fonctionne aussi pour les répertoires

Partie 2: 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 incluent 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 -i` affiche le numéro d'inode d'un fichier.

Exécutez les deux commandes suivantes :

```
admin@localhost:~$ touch linuxSyntaxe de la commande
```

```
admin@localhost:~$ ls -i linux
73798439 linux
```

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

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 LINK_NAME
```

GBU/Linux

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:

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

123456 -rw-r--r--      2      admin   admin 0 Apr 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 autorisations.

Exécutez la commande suivante pour essayer de créer un lien dur pour un répertoire :

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

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

Malheureusement, aucun lien dur ne peut être établi vers des répertoires.

Résumé :

Lien physique(dur)

Syntaxe de la commande

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

Permet de donner plusieurs noms à un fichier

Pas pour les répertoires

Ne traverse pas les partitions

un fichier est détruit quand TOUS ses liens physiques sont supprimés (≠ raccourcis)