

Faire un lien symbolique immuable sur Linux

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 9 septembre 2026

<https://www.bortzmeyer.org/lien-immuable.html>

Hier, j'ai cherché à faire un lien symbolique « immuable » sur Linux, plus exactement un lien que son ou sa propriétaire ne risquerait pas de supprimer par accident, tout en lui permettant de créer et détruire d'autres objets dans ce répertoire. Si vous connaissez Linux, vous allez probablement crier « `chattr +i`, `chattr +i!` ». Mais, non, cela ne marche pas.

Essayons quand même, sur le compte d'un utilisateur nommé Bob. D'abord avec un fichier ordinaire :

```
$ echo "Contenu sans intérêt" > myfile
$ sudo chattr +i myfile
$ ls -l myfile
-rw-rw-r-- 1 bob bob 23 Sep  9 15:12 myfile
$ lsattr myfile
----i-----e----- myfile
$ rm myfile
rm: cannot remove 'myfile': Operation not permitted
```

Parfait, tout marche (une alternative à `sudo` aurait été la capacité `CAP_LINUX_IMMUTABLE` mais c'est une autre histoire). Essayons maintenant avec un lien symbolique :

```
$ ln -s /etc/motd mylink
$ sudo chattr +i mylink
chattr: Operation not supported while reading flags on mylink
```

Bon, c'est raté, `chattr` ne marche pas sur les liens symboliques. (Le message exact dépend de votre `libc`; vous aurez peut-être plutôt « *Symbolic link loop* ».) En tout cas, ça ne marche pas sur Linux. (Il paraît que cela fonctionne sur FreeBSD.)

Comment faire, alors? Changer le propriétaire du lien ne change rien car la suppression est une opération sur le répertoire. Le propriétaire du répertoire peut supprimer des fichiers qui ne lui appartiennent pas. Et changer le propriétaire du répertoire ne suffit pas, car on veut que Bob puisse continuer à lire et à écrire dans ce répertoire. Le *"sticky bit"* seul ne suffirait pas non plus, il empêche de supprimer les fichiers des autres, pas les siens.

ChatGPT a finalement trouvé une solution, mais à la troisième itération <<https://chatgpt.com/share/6aa15fb4-d53c-83ed-9f2e-e65e4cfcc2a7>>, après quelques idées qui ne marchaient pas (ChatGPT, comme tout utilisateur des réseaux sociaux, ne teste pas ses idées avant de les publier; « *You're right* », « *Yes. Your suspicion is exactly correct* »):

```
$ ln -s /etc/motd mylink
$ sudo chown root:bob .
$ sudo chmod +t .
$ sudo chmod g+w .
$ sudo chown --no-dereference root:root mylink

$ rm mylink
rm: cannot remove 'mylink': Operation not permitted

$ ls -alt
total 8
drwxrwxr-t 2 root bob  4096 Sep  9 15:26 .
lrwxrwxrwx 1 root root    9 Sep  9 15:26 mylink -> /etc/motd
drwxr-x--- 3 bob  bob  4096 Sep  9 15:25 ..
```

Détaillons cette solution. On met `root` comme propriétaire du répertoire (on peut aussi utiliser un autre utilisateur que `root`) mais le groupe est `bob`, un groupe qui inclut Bob. On met ensuite le *"sticky bit"*. On donne au groupe le droit d'écriture dans le répertoire. Et enfin on change le propriétaire du lien pour que cela ne soit pas Bob (notez le `--no-dereference` sans lequel `chown` essaierait de changer la cible du lien et pas le lien). Et Bob ne peut plus supprimer le lien, tout en pouvant lire et écrire dans le répertoire.