

Korte uitleg: Harde en zachte links

Deze *Korte uitleg* gaat er vanuit dat zeker de *Korte uitleg: Directories, meta-data van bestanden, inodes* is begrepen.

Harde links

Even terug naar de directory. Die bestaat uit een tabel van twee kolommen: bestandsnaam en inode nummer. Wat nu als er in die tabel twee bestandsnamen voorkomen met hetzelfde inode nummer? En kan/mag dat? Ja, met enige beperkingen mag en kan dat. Die bestandsnaam met dezelfde inode mag zelfs in een andere directory voorkomen. Tenslotte gaat het om een hard adres in het file system en je komt dus altijd op dezelfde inode uit vanuit welke directory je daar ook naar verwijst.

Echter, het moet wel in hetzelfde file system zijn. Er is geen manier om direct naar een inode in een ander file system te verwijzen.

De meeste lezers hebben het al begrepen. We noemen dit een "harde link/hard link".

Een andere beperking is dat je harde links niet voor directories kunt maken (maar het systeem doet dat wel!).

```
henk@boven:~/test/lnt> touch aap
henk@boven:~/test/lnt> ln aap noot
henk@boven:~/test/lnt> ls -li
totaal 0
6030300 -rw-r--r-- 2 henk wij 0 5 jul 14:47 aap
6030300 -rw-r--r-- 2 henk wij 0 5 jul 14:47 noot
henk@boven:~/test/lnt>
```

Hierboven wordt het bestand **aap** gemaakt, daarna maken we een harde link met de naam **noot**. We zien dat beide namen naar dezelfde inode verwijzen. Als we dus iets wijzigen via de ene bestandsnaam gebeurt dat ook voor de andere naam. Een wijziging in de inode van **aap**:

```
henk@boven:~/test/lnt> chmod go-r aap
henk@boven:~/test/lnt> ls -li
totaal 0
6030300 -rw----- 2 henk wij 0 5 jul 14:47 aap
6030300 -rw----- 2 henk wij 0 5 jul 14:47 noot
henk@boven:~/test/lnt>
```

En een wijziging in de inhoud van **noot**:

```
henk@boven:~/test/lnt> echo "hoera" >>noot
henk@boven:~/test/lnt> ls -li
totaal 8
6030300 -rw----- 2 henk wij 6 5 jul 14:50 aap
6030300 -rw----- 2 henk wij 6 5 jul 14:50 noot
henk@boven:~/test/lnt> cat aap
```

```
hoera
henk@boven:~/test/lnt>
```

Let nu eens op de derde kolom waar telkens **2** staat. Dat betekent dat er twee harde links zijn. je bent dus gewaarschuwd.
Wat gebeurt er als je één van de twee weggooit?

```
henk@boven:~/test/lnt> rm aap
henk@boven:~/test/lnt> ls -li
totaal 4
6030300 -rw----- 1 henk wij 6 5 jul 14:50 noot
henk@boven:~/test/lnt>
```

Alleen de bestandsnaam/inode verwijzing uit de directory wordt verwijderd. En de inode teller staat nu op **1**. De inode (en dus het bestand) wordt pas verwijderd als alle verwijzingen verwijderd zijn en de inode teller dus op **0** komt.

We hebben boven gezegd dat je geen hard links kunt maken van een directory, maar dat het systeem dat wel kan:

```
henk@boven:~/test/lnt> ls -lai
totaal 8
6029480 drwxr-xr-x 2 henk wij 4096 5 jul 14:59 .
6029799 drwxr-xr-x 5 henk wij 4096 5 jul 14:32 ..
henk@boven:~/test/lnt>
```

We zien hier de inode teller van **.** op **2** staan en die van **..** zelfs op **5**. Dat komt omdat **.** in deze directory verwijst naar inode **6029480**, maar **lnt** in directory **test** doet dat ook:

```
henk@boven:~/test/lnt> ls -ldi ~/test/lnt/
6029480 drwxr-xr-x 2 henk wij 4096 5 jul 14:59 /home/henk/test/lnt/
henk@boven:~/test/lnt>
```

En **..** is in werkelijkheid **~/test**

```
henk@boven:~/test/lnt> ls -lai ~/test/
totaal 32
6029799 drwxr-xr-x 5 henk wij 4096 5 jul 14:32 .
6029313 drwxr-xr-x 87 henk wij 4096 5 jul 11:35 ..
1441878 drwxr-xr-x 2 henk wij 4096 17 mei 15:48 bestanden
1441888 drwxr-xr-x 2 henk wij 4096 18 mei 12:37 fff
6029554 -rw-r--r-- 1 henk wij 0 26 mrt 10:31 file
6029495 -rw-r--r-- 1 henk wij 566 13 mrt 11:08 hhh
6029480 drwxr-xr-x 2 henk wij 4096 5 jul 14:59 lnt
6029668 -rwxr--r-- 1 henk wij 49 13 mrt 10:39 script
6030011 -rw-r--r-- 1 henk wij 3 31 mrt 19:29 spsp
henk@boven:~/test/lnt>
```

Er is een verwijzing naar dezelfde inode in **/home/henk** (onder de naam **test**), in

`/home/henk/test` (onder de naam `.`) en in `/home/henk/test/bestanden` (onder de naam `..`) en dat laatste ook voor `fff` en `lnt`. Totaal vijf (5).
Hoeveel directories staan er in de directory `/home/henk`?

Zachte links

Maar wat als je graag een handige naam wilt hangen aan een directory of aan een bestand dat op een ander file system staat?
Daarvoor hebben ze de symbolische link (ook wel soft link) uitgevonden:

```
henk@boven:~/test/lnt> ln -s /mnt/A/home/henk oude-home
henk@boven:~/test/lnt> l
totaal 8
drwxr-xr-x 2 henk wij 4096  5 jul 17:25 ./
drwxr-xr-x 5 henk wij 4096  5 jul 14:32 ../
lrwxrwxrwx 1 henk wij  16  5 jul 17:25 oude-home -> /mnt/A/home/henk/
henk@boven:~/test/lnt>
```

Wat we hier zien is een nieuw type bestand (aangegeven met `l`) en dat bestand bevat de naam van een pad naar een ander bestand (kijk maar, het bestand is 17 bytes groot en de string `/mnt/A/home/henk/` is 17 tekens lang). Wat ook opvalt zijn de permissies. In feite gelden niet de permissies die je hier ziet, maar de permissies van het uiteindelijke echte bestand. Logisch hè? Anders zou je die echte permissies zo maar kunnen omzeilen.

In tegenstelling tot het maken van een harde link hoeft het bestand waarnaar je verwijst niet te bestaan.

- Voordeel: je kunt een link leggen naar een bestand op een file system dat niet altijd is gemount of een bestand dat je nog gaat aanmaken.
- Nadeel: een tikfout, of later het weggooien van het gelinkte bestand levert verrassingen op.

Omdat zo'n link alleen maar een stukje tekst is zonder veel controle kun je nog gekkere dingen maken. Zo kun je een link naar een link naar een link maken en zelfs uiteindelijk in een kringetje ronddraaien. Dat kringetje is niet aanbevolen:

```
henk@boven:~/test/lnt> ln -s wim zus
henk@boven:~/test/lnt> ln -s zus wim
henk@boven:~/test/lnt> l
totaal 8
drwxr-xr-x 2 henk wij 4096  5 jul 17:42 ./
drwxr-xr-x 5 henk wij 4096  5 jul 14:32 ../
lrwxrwxrwx 1 henk wij  16  5 jul 17:25 oude-home -> /mnt/A/home/henk/
lrwxrwxrwx 1 henk wij   3  5 jul 17:42 wim -> zus
lrwxrwxrwx 1 henk wij   3  5 jul 17:42 zus -> wim
henk@boven:~/test/lnt> cat wim
cat: wim: Te veel niveaus van symbolische koppelingen -- vermoedelijk een lus
henk@boven:~/test/lnt>
```