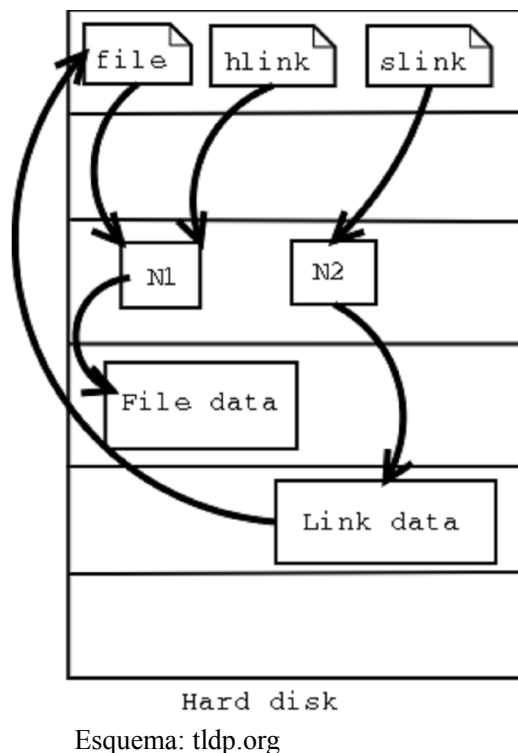


Diferencias entre soft (symbolic) y hard links

ABRIL 27, 2011

Hoy vamos a tratar de comprender las **diferencias que existen en Unix / Linux entre los enlaces simbólicos (soft o symbolic links) y los enlaces duros (hard links).**



Enlaces simbólicos (soft / symbolic links)

La manera más sencilla de comprender que es un enlace simbólico en Linux es compararlo con el “enlace directo” o “shortcut” en Windows. El fichero o directorio se encuentra en un único punto del disco y los enlaces son un *puntero* contra él. **Cada enlace simbólico tiene su propio número de inodo** lo que permite hacer enlaces simbólicos entre distintos sistemas de ficheros.

Para crear enlaces (tanto simbólicos como duros) usamos el **comando ln**. En este caso vamos a crear un enlace simbólico (parámetro **-s**) del fichero test:

```
$ ln -s test enlace-a-test
```

Si listamos ambos veremos que el enlace tiene el carácter **l** que lo identifica como enlace simbólico:

```
$ ls -l
lrwxrwxrwx 1 alex alex 4 2011-04-27 18:59 enlace-a-test -> test
-rw-r--r-- 1 alex alex 0 2011-04-27 18:58 test
```

Para confirmar que el enlace simbólico tiene un inodo distinto usamos el comando stat:

```
$ stat test
  File: «test»
  Size: 0          Blocks: 0          IO Block: 4096   archivo regular vacío
Device: 804h/2052d Inode: 73793         Links: 1
```

```

File: «test»
Size: 0          Blocks: 0          IO Block: 4096   archivo regular vacío
Device: 804h/2052d Inode: 73793      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/   alex)   Gid: ( 1000/   alex)
Access: 2011-04-27 18:58:53.124142406 +0200
Modify: 2011-04-27 18:58:53.124142406 +0200
Change: 2011-04-27 18:58:53.124142406 +0200

$ stat enlace-a-test
File: «enlace-a-test» -> «test»
Size: 4          Blocks: 0          IO Block: 4096   vínculo simbólico
Device: 804h/2052d Inode: 77212      Links: 1
Access: (0777/lrwxrwxrwx)  Uid: ( 1000/   alex)   Gid: ( 1000/   alex)
Access: 2011-04-27 18:59:07.812139890 +0200
Modify: 2011-04-27 18:59:06.460112888 +0200
Change: 2011-04-27 18:59:06.460112888 +0200

```

También lo podemos verificar sacando el inodo en el ls (-i):

```

$ ls -li
77212 lrwxrwxrwx 1 alex alex 4 2011-04-27 18:59 enlace-a-test -> test
73793 -rw-r--r-- 1 alex alex 0 2011-04-27 18:58 test

```

Hay que tener en cuenta, que en Linux / Unix (al igual que con los accesos directos de Windows), **si borramos el fichero o directorio origen, el enlace simbólico permanece pero los datos desaparecen para siempre.**

Enlaces duros (hard links)

Los enlaces duros lo que hacen es **asociar dos o más ficheros compartiendo el mismo inodo**. Esto hace que cada enlace duro es una copia exacta del resto de ficheros asociados, tanto de datos como de permisos, propietario, etc. Esto implica también que cuando se realicen cambios en uno de los enlaces o en el fichero este también se realizará en el resto de enlaces.

Los enlaces duros **no pueden hacerse contra directorios y tampoco fuera del propio sistema de ficheros.**

Vamos a crear un hard link contra el fichero “test” de antes y veremos que efectivamente comparten inodo y que los datos se sincronizan entre ambos:

```

$ ln test enlace-duro-test
$ ls -li
73793 -rw-r--r-- 2 alex alex 5 2011-04-27 19:09 enlace-duro-test
73793 -rw-r--r-- 2 alex alex 5 2011-04-27 19:09 test

```

En la primera columna verificamos que tienen el mismo número de inodo y en la tercera se especifica cuando enlaces duros tiene el fichero. Si hacéis cambios en uno de ellos veréis que también se hacen en el resto. Si por ejemplo cambiamos los permisos al fichero test:

```

$ chmod 0755 test
$ ls -li
73793 -rwxr-xr-x 2 alex alex 5 2011-04-27 19:09 enlace-duro-test
73793 -rwxr-xr-x 2 alex alex 5 2011-04-27 19:09 test

```

Y finalmente el stat de cada uno verifica todo lo que comentamos:

```
$ stat test
```

Y finalmente el stat de cada uno verifica todo lo que comentamos:

```
$ stat test
  File: «test»
  Size: 5          Blocks: 8          IO Block: 4096   archivo regular
Device: 804h/2052d Inode: 73793       Links: 2
Access: (0755/-rwxr-xr-x)  Uid: ( 1000/   alex)   Gid: ( 1000/   alex)
Access: 2011-04-27 19:09:51.528132995 +0200
Modify: 2011-04-27 19:09:53.640114896 +0200
Change: 2011-04-27 19:11:42.516138726 +0200

$ stat enlace-duro-test
  File: «enlace-duro-test»
  Size: 5          Blocks: 8          IO Block: 4096   archivo regular
Device: 804h/2052d Inode: 73793       Links: 2
Access: (0755/-rwxr-xr-x)  Uid: ( 1000/   alex)   Gid: ( 1000/   alex)
Access: 2011-04-27 19:09:51.528132995 +0200
Modify: 2011-04-27 19:09:53.640114896 +0200
Change: 2011-04-27 19:11:42.516138726 +0200
```

Diferencias entre soft y hard links

- Los enlaces simbólicos se pueden hacer con ficheros y directorios mientras que los duros solo entre ficheros.
- Los enlaces simbólicos se pueden hacer entre distintos sistemas de ficheros, los duros no.
- Los enlaces duros comparten el número de inodo, los simbólicos no.
- En los enlaces simbólicos si se borra el fichero o directorio original, la información se pierde, en los duros no.
- Los enlaces duros son copias exactas del fichero mientras que los simbólicos son meros punteros o “accesos directos”.