

Enlaces duros (hard links) en Linux explicados

Edison Achalma

Escuela Profesional de Economía, Universidad Nacional de San Cristóbal de Huamanga

Este abstract será actualizado una vez que se complete el contenido final del artículo.

Palabras Claves: keyword1, keyword2

Tabla de contenidos

Introduction

0.1	Cómo Crear un Enlace Duro (Hard Link) en Linux	1
0.2	Introducción	1
0.3	Características principales de los enlaces duros	1
0.4	Creando un enlace duro (hard link)	2

1 Publicaciones Similares

Enlaces duros (hard links) en Linux explicados

0.1 Cómo Crear un Enlace Duro (Hard Link) en Linux

0.2 Introducción

Los **enlaces duros o hard link** asocian dos o más ficheros compartiendo el mismo **inodo**, esto hace que cada **enlace duro** sea una copia exacta del resto de los ficheros enlazados, tanto en los datos como en los permisos, propietario, grupo, etc. Cuando se modifica uno de los enlaces o el fichero original, los cambios afectan al resto de los enlaces.

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

En sistemas linux también existen los enlaces simbolicos, también conocidos como **enlaces blandos o Symlinks**.

0.3 Características principales de los enlaces duros

- Solo se pueden hacer entre ficheros. No se pueden hacer entre directorios.
- No se pueden hacer entre distintos sistemas de ficheros.
- Comparten el número de inodo
- Si se borra el fichero original la información no se pierde.

Edison Achalma

El autor no tiene conflictos de interés que revelar. Los roles de autor se clasificaron utilizando la taxonomía de roles de colaborador (CRediT; <https://credit.niso.org/>) de la siguiente manera: Edison Achalma: conceptualización, redacción

La correspondencia relativa a este artículo debe dirigirse a Edison Achalma, Email: elmer.achalma.09@unsch.edu.pe

- Son copias exactas del fichero original. Los cambios aplicados a uno de ellos o al fichero original, afectan a todos.

0.4 Creando un enlace duro (hard link)

La sintaxis genérica para crear un **enlace duro** es la siguiente:

```
ln TARGET LINK_NAME
```

- **TARGET**: Nombre del archivo existente al que le crearemos el **enlace duro**.
- **LINK_NAME**: Nombre del **enlace duro**.

Veamos un ejemplo:

```
ln test.txt enlace-duro-a-test.txt
```

Si listamos ambos archivos con el comando `ls -li`,

```
ls -li
```

Observamos que ambos comparten el mismo inodo

```
786433 -rw-r--r-- 2 achalma achalma 0 jun 21 2023 test.txt
786433 -rw-r--r-- 2 achalma achalma 0 jun 21 2023 enlace-duro-a-test.txt
```

Se observa en la primera columna que ambos, archivo y enlace, comparten el mismo número de inodo (**786433**). La tercera columna indica cuantos **enlaces duros** tiene el fichero, en este caso **2**, el archivo original más el enlace.

Si modificamos uno de ellos, los cambios afectan a todos. Por ejemplo, vamos a conceder permiso de ejecución al propietario en el archivo `test.txt` y veamos que pasa con el enlace:

```
chmod u+x test.txt
```

Si volvemos a listar ambos archivos vemos que el cambio ha afectado a ambos, al fichero original y al enlace:

```
$ ls -li
786433 -rwxr--r-- 2 achalma achalma 0 jun 21 21:27 enlace-duro-a-test.txt
786433 -rwxr--r-- 2 achalma achalma 0 jun 21 21:27 test.txt
```

Si editásemos el archivo o el enlace, los cambios realizados en el contenido afectarían a ambos. ## Generar varios

Para crear 35 enlaces duros de `_metadata.yml` puedes usar un simple bucle `for` en la terminal de Linux:

```
for i in {1..35}; do ln _metadata.yml "_metadata${i}.yml"; done
```

Este comando hace lo siguiente:

1. `for i in {1..35};`: Esto establece un bucle que itera desde 1 hasta 35.
2. `ln _metadata.yml "_metadata${i}.yml";`: Dentro del bucle, se ejecuta el comando `ln` para crear un enlace duro de `_metadata.yml` con el nombre `_metadataX.yml`, donde X es el valor actual de `i` en el bucle.

Finalmente, se generarán los 35 enlaces duros.

1 Publicaciones Similares

Si te interesó este artículo, te recomendamos que explores otros blogs y recursos relacionados que

pueden ampliar tus conocimientos. Aquí te dejo algunas sugerencias:

1. [Comandos De Informacion Windows](#)
2. [Adb](#)
3. [Limpieza Y Optimizacion De Pc](#)
4. [Usando Apk En Window 11](#)
5. [Gestionar Versiones De Jdk En Kubuntu](#)
6. [Instalar Tor Browser](#)
7. [Crear Enlaces Duros O Hard Link En Linux](#)
8. [Comandos Vim](#)
9. [Guia De Git Y Github](#)
10. [00 Primeros Pasos En Linux](#)
11. [01 Introduccion Linux](#)
12. [02 Distribuciones Linux](#)
13. [03 Instalacion Linux](#)
14. [04 Administracion Particiones Volumenes](#)
15. [Atajos De Teclado Y Comandos Para Usar Vim](#)
16. [Instalando Specitify](#)
17. [Gestiona Tus Dotfiles Con Gnu Stow](#)

Esperamos que encuentres estas publicaciones igualmente interesantes y útiles. ¡Disfruta de la lectura!