SISTEMAS INFORMÁTICOS

LINUX

DIFERENCIAS ENTRE ENLACES BLANDOS Y DUROS

Alberto Martínez Pérez

1º CFGS DESARROLLO DE APLICACIONES WEB (daw)

**ÍNDICE**

[DIFERENCIAS ENTRE ENLACES BLANDOS Y DUROS 3](#_Toc124978155)

[INODOS 4](#_Toc124978156)

# DIFERENCIAS ENTRE ENLACES BLANDOS Y DUROS

|  |  |  |
| --- | --- | --- |
|  | ENLACE BLANDO/SIMBÓLICO/*SOFT LINK* | ENLACE DURO/FUERTE/*HARD LINK* |
| ¿Qué son? | Son archivos que referencian a otro archivo, el cual referencia a un contenido almacenado en el disco duro (es decir, un enlace duro). | Son archivos que apuntan a un contenido almacenado en el disco duro. |
| ¿Cuál es su inodo? | Los enlaces blandos tienen un inodo diferente al del archivo al que referencian. Cada enlace blando tendrá su propio inodo. | Los enlaces duros que apunten al mismo contenido en disco compartirán a su vez el mismo inodo. |
| ¿Cuáles son sus permisos? | Los permisos (propietario y grupo) pueden ser diferentes al archivo original. | Todos los permisos serán iguales que en el archivo original. |
| ¿Dónde se pueden crear? | Se pueden crear en cualquier partición del disco o incluso en otro disco duro diferente al del original. | Únicamente se deben crear en la misma partición que el archivo original. |
| ¿Dónde se pueden usar? | Se pueden usar en cualquier ubicación, partición y sistema de archivos del disco. | Sólo se pueden usar en la partición en la que han sido creados. |
| ¿Qué ocurre cuando los borramos? | Si borramos un enlace blando, perdemos el “acceso” directo al enlace duro que referencia, pero se puede seguir accediendo a la información. | Si borramos un enlace duro (y es el único con acceso a la información) se pierde el acceso a la información del disco.  Si borramos un enlace duro que es referenciado por un blando, este último pierde su utilidad. |
| ¿Qué ocurre cuando modificamos algo en el enlace o en el archivo original? | Cualquier cambio que se produzca en el archivo original o en el enlace blando afectará por igual a ambos. | Cualquier cambio que se produzca en el archivo original o en el enlace duro afectará por igual a ambos. |
| ¿Pueden referenciar a carpetas? | Se pueden crear enlaces simbólicos a carpetas. | No podemos crear enlaces duros de carpetas. |
| ¿Cuánto ocupan? | Ocupan más tamaño en disco. | Ocupan menos tamaño en disco. |
| ¿Cuál es más rápido? | El acceso al contenido es más lento. | El acceso al contenido es más rápido. |
| ¿Qué ocurre cuando se cambia la ubicación del archivo original? | Si se cambia la ubicación del archivo original, se romperá el enlace simbólico. | Si cambiamos de ubicación el archivo original, el enlace duro no se rompe. |

# INODOS

Son estructuras de datos que almacenan información sobre un archivo de nuestro sistema de archivos (es decir, metadatos de un archivo). No tienen nombre y se identifican mediante un número entero único y cada uno de ellos sólo puede contener los datos de un único archivo del sistema de archivos.

Los metadatos almacenados son:

* **Número de inodo**. Sirve para identificar al inodo.
* **Tamaño del archivo** y número de bloques que ocupa.
* **Dispositivo** **de almacenamiento** del archivo.
* **Número de enlaces**.
* **Identificador de usuario** (UID o UserID).
* **Identificador de grupo** (GID o GroupID).
* **Marcas de tiempo** (por ejemplo, la fecha de creación, la fecha de último acceso…).

Para acceder a estos datos utilizamos el comando stat seguido del nombre del archivo, por ejemplo:

Texto

Descripción generada automáticamente

Su función principal es el indicar las posiciones del disco duro en las que están almacenados los datos del propio archivo.

El tamaño por defecto en ext4 de los inodos es de 256 bytes y cada bloque del disco tiene una capacidad de 4096 bytes por lo que cada bloque puede guardar hasta 16 inodos.

Los inodos se guardan al inicio de cada partición del disco en una tabla de inodos. Los inodos de cada zona redireccionan a bloques del disco duro lo más cercanos a la ubicación del inodo para, de esta manera, minimizar el desplazamiento del cabezal de lectura.

Cada tabla de direccionamiento tiene 15 entradas, 12 de las cuales permiten un direccionamiento directo a un bloque de datos del disco duro. A partir de la 13 habrá que realizar un direccionamiento indirecto, de forma que, por ejemplo, la posición 14 de la tabla nos dirigirá a un bloque de datos con otra tabla de direcciones la cual sí nos podrá dirigir hacia los bloques donde se encuentra el contenido al que queremos acceder.