

Examen 104 - Dispositivos, Sistemas de Archivos Linux y Jerarquía Estándar

104.6 Crear y cambiar enlaces duros y simbólicos

Teoría

Los enlaces permiten que un mismo archivo o directorio sea referenciado desde múltiples ubicaciones dentro del sistema de archivos, o que un nombre apunte a otro nombre. Existen dos tipos principales de enlaces en Linux: duros y simbólicos.

Conceptos Clave:

1. **Inodos:** Cada archivo y directorio en un sistema de archivos Linux (excepto los enlaces simbólicos) está asociado con un **inodo**. Un inodo contiene toda la información sobre el archivo o directorio, excepto su nombre y su contenido (los datos). Esto incluye los permisos, la propiedad, las marcas de tiempo, y lo más importante, los punteros a los bloques de disco donde se almacenan los datos reales del archivo.
2. **Entradas de Directorio:** Un directorio es esencialmente una tabla que mapea nombres de archivo a números de inodo. Cuando buscas un archivo por su nombre (ej: `cat /ruta/archivo.txt`), el sistema busca el nombre en la tabla de entradas del directorio padre para encontrar el número de inodo asociado. Luego usa el inodo para encontrar los datos del archivo.

Tipos de Enlaces:

1. Enlaces Duros (Hard Links):

- Un enlace duro es simplemente otra **entrada de directorio** que apunta al **mismo inodo** que el archivo original.
- Piensa en ello como tener múltiples nombres (entradas de directorio) para el mismo conjunto de datos en el disco.
- Características:
 - **Comparten el mismo inodo:** Tienen los mismos permisos, propietario, grupo y datos de archivo que el original.
 - **No distinguen entre "original" y "enlace":** Todos los nombres que apuntan al mismo inodo son igualmente válidos. Eliminar el "original" o cualquiera de los enlaces duros simplemente elimina una entrada de directorio.
 - **El archivo real (los datos en disco) solo se elimina cuando el número de enlaces duros que apuntan a su inodo llega a CERO** (y ningún proceso tiene el archivo abierto).
 - **Limitación 1: No pueden enlazar directorios.** Los enlaces duros solo funcionan para archivos regulares. Esto evita bucles infinitos en la estructura del sistema de archivos.
 - **Limitación 2: Solo funcionan dentro del mismo sistema de archivos.** Un inodo es único solo dentro de su sistema de archivos. Un enlace duro no puede apuntar a un inodo en otra partición o dispositivo.

- En la salida de `ls -l`, la segunda columna muestra el "contador de enlaces" (link count) para el inodo. Este número indica cuántos enlaces duros (entradas de directorio) apuntan a ese inodo.

2. Enlaces Simbólicos (Symbolic Links o Soft Links):

- Un enlace simbólico es un **archivo especial** que contiene la **ruta (path)** al archivo o directorio al que apunta. No apunta directamente al inodo de destino.
- Piensa en ello como un "acceso directo" o un "puntero".
- Características:
 - **Tienen su propio inodo:** Es un inodo de tipo "enlace simbólico". Sus permisos y propiedad son los del propio archivo de enlace simbólico (generalmente `rw-rw-rw-` por defecto, aunque los permisos del *destino* son los que realmente importan al acceder).
 - **Contienen una ruta:** Cuando intentas acceder al contenido de un enlace simbólico, el sistema lee la ruta almacenada en el enlace y te redirige al archivo o directorio de destino.
 - **Pueden enlazar directorios.**
 - **Pueden enlazar a través de sistemas de archivos.** La ruta almacenada puede apuntar a cualquier parte accesible en el sistema de archivos montado.
 - **Pueden apuntar a destinos que no existen.** Si el archivo o directorio al que apunta un enlace simbólico se elimina o se renombra, el enlace simbólico se vuelve un **enlace "colgante" (dangling link)** o "roto" (broken link). Seguirá existiendo como archivo, pero apuntará a un lugar inválido. `ls -l` a menudo los muestra en rojo o con una indicación de error.
- En la salida de `ls -l`:
 - El primer carácter es `l`.
 - Se muestra la ruta a la que apunta (ej: `nombre_enlace -> /ruta/a/destino`).
 - La segunda columna muestra 1 (el inodo del enlace simbólico solo tiene una entrada de directorio, la del propio enlace).

El Comando `ln`:

El comando `ln` (link) se utiliza para crear enlaces.

- `ln <origen> <enlace>`: Crea un **enlace duro** llamado `<enlace>` que apunta a `<origen>`. `<origen>` debe existir. Si `<enlace>` es un directorio, el enlace se crea dentro de ese directorio con el mismo nombre base que `<origen>`.
- `ln -s <origen> <enlace>`: Crea un **enlace simbólico** llamado `<enlace>` que apunta a `<origen>`. `<origen>` puede ser un archivo o un directorio, y no necesita existir cuando se crea el enlace (aunque el enlace estará roto hasta que el destino exista). Si `<enlace>` es un directorio, el enlace simbólico se crea dentro de ese directorio.

- `ln <origen1> <origen2> ... <directorio_destino>`: Crea enlaces duros en <directorio_destino> para cada <origen>.
- `ln -s <origen1> <origen2> ... <directorio_destino>`: Crea enlaces simbólicos en <directorio_destino> para cada <origen>.
- `ln -f`: Fuerza la eliminación de enlaces de destino existentes si ya existen.

Cuándo usar uno u otro:

- **Enlaces duros:** Útiles cuando quieres tener múltiples nombres para el mismo archivo *dentro del mismo sistema de archivos* y no quieres que la eliminación de uno afecte a los otros (hasta que sean el último enlace). Son más eficientes ya que el acceso es directo al inodo.
- **Enlaces simbólicos:** Más comunes y flexibles. Útiles para enlazar directorios, enlazar a través de sistemas de archivos, o cuando quieres que el enlace "se rompa" si el destino original se elimina. Son esenciales para la organización del sistema (ej: `/usr/bin` a menudo contiene enlaces simbólicos a binarios en otros lugares) y para software que espera encontrar archivos en ubicaciones específicas.