goto-linux.com

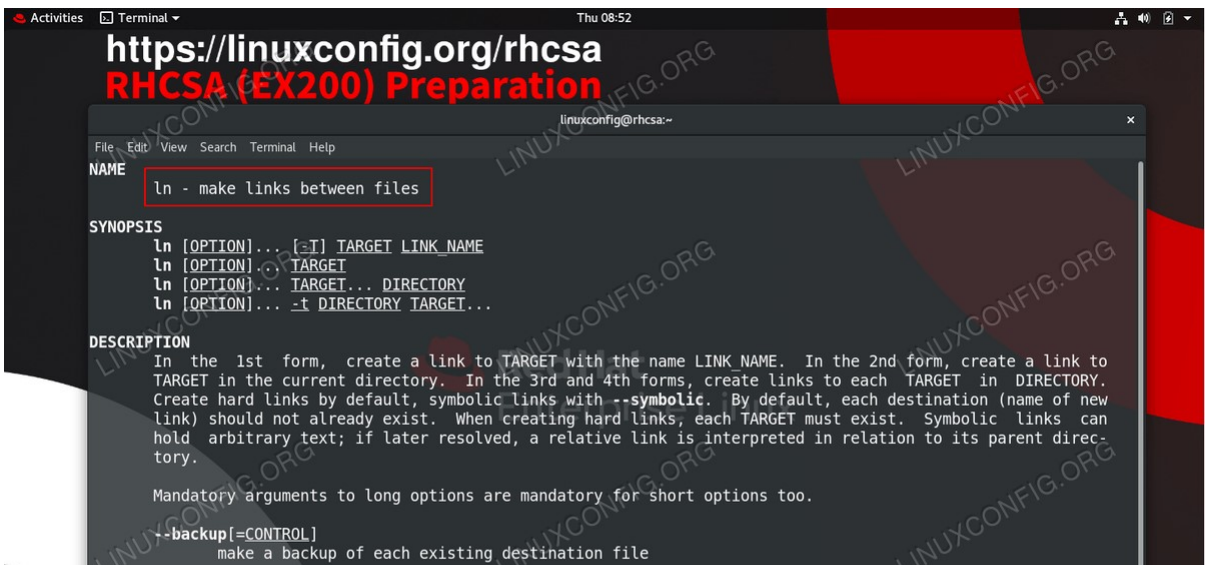
Crear enlaces duros y blandos - Preparación para el examen RHCSA

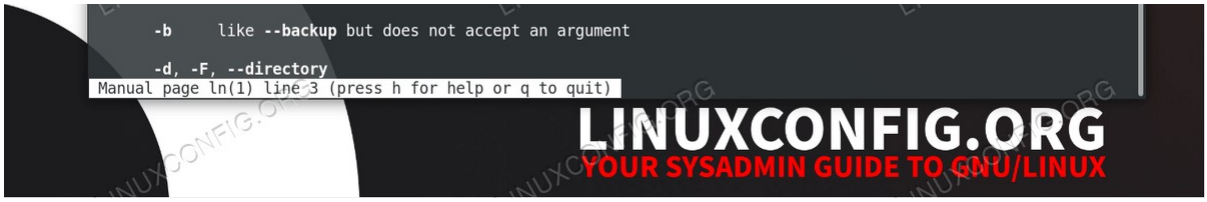
7-9 minutos

En esta parte de la preparación del examen RHCSA, centraremos nuestra atención en los enlaces. Hay dos tipos de enlaces, enlaces duros y enlaces blandos. En este artículo hablaremos sobre cómo crear y eliminar enlaces y también discutiremos algunos antecedentes básicos detrás de ambos, los enlaces duros y los enlaces blandos.

En este tutorial aprenderás:

- ¿Qué son los enlaces simbólicos (suaves)?
- ¿Qué son los enlaces duros?
- Cómo crear un enlace simbólico
- Cómo crear un enlace duro
- Cómo eliminar el enlace





Página manual del `ln`comando

Requisitos de software y convenciones utilizados

Requisitos de software y convenciones de línea de comandos de Linux

Categoría	Requisitos, convenciones o versión de software utilizada
Sistema	Red Hat Enterprise Linux 8 o cualquier otra distribución GNU / Linux
Software	N / A
Otro	Acceso privilegiado a su sistema Linux como root o mediante el `sudo`comando.
Convenciones	**#** : requiere que los comandos de Linux se ejecuten con privilegios de root, ya sea directamente como usuario root o mediante el uso del `sudo`comando **$** : requiere que los comandos de Linux se ejecuten como un usuario normal sin privilegios

¿Qué son los enlaces en los sistemas GNU / Linux?

Cada archivo tiene una información sobre su fecha de creación, modificación, acceso, así como la propiedad del archivo y sus permisos almacenados en un denominado **inodo** . Además de los metadatos almacenados ya mencionados, el **inodo** también almacena información sobre un bloque de datos donde el

contenido real del archivo se almacena en el sistema de archivos.

Por lo tanto, la función principal del **inodo** es describir un objeto del sistema de archivos, como un archivo o un directorio. Para acceder al objeto del sistema de archivos asociado con un **inodo** específico, debemos proporcionar al usuario un **enlace rígido** que sea el nombre real del archivo o directorio.

Esto explica el primer tipo de enlaces que son **enlaces duros**. El segundo tipo de enlaces en el sistema operativo GNU / Linux son **enlaces simbólicos,** también conocidos como enlaces blandos. La diferencia entre enlaces duros y simbólicos es que los enlaces simbólicos solo apuntan a enlaces duros, es decir, apuntan a los nombres de archivo o directorio existentes. En pocas palabras, los enlaces permiten al usuario acceder a archivos o directorios a través de múltiples nombres.

Enlaces duros

Dado que **los enlaces duros** están asociados con los **inodos,** que a su vez son una característica del sistema de archivos, los enlaces duros no pueden cruzar los sistemas de archivos, por lo tanto, solo son válidos dentro del mismo sistema de archivos. Además, no es posible crear enlaces duros en directorios. Cualquier intento de crear un enlace duro desde un directorio dará como resultado `hard link not allowed for directory`un mensaje de error.

Vamos a crear algunos enlaces duros. En este ejemplo, crearemos primero un archivo arbitrario llamado que `sandbox`contiene un texto `linuxconfig.org`. Una vez listo, crearemos múltiples **enlaces duros** a este archivo apuntando desde diferentes ubicaciones.

1. Cree un archivo llamado que `sandbox`contenga un texto

`RHCSA`dentro del directorio de inicio de un usuario `~/`.

```
$ echo "RHCSA" > ~/sandbox
```

Verifique el contenido del archivo con el `cat`comando:

```
$ cat ~/sandbox
RHCSA
```

2. Aún ubicado dentro del directorio de inicio de un usuario, cree un enlace duro al `sandbox`archivo desde el `/tmp/`directorio llamado `hardlink1`.

```
$ ln sandbox /tmp/hardlink1
```

Ahora verifique el contenido del enlace duro recién creado `/tmp/hardlink1`. El contenido del archivo `/tmp/hardlink1`original `sandbox`debe ser el mismo:

```
$ cat /tmp/hardlink1
RHCSA
```

3. Verifique la información de enlace asociada con ambos nombres de archivo `sandbox`y `/tmp/hardlink1`.

```
$ ls -l /tmp/hardlink1
-rw-rw-r--. 2 linuxconfig linuxconfig 6 Jul 25
10:20 /tmp/hardlink1
$ ls -l ~/sandbox
-rw-rw-r--. 2 linuxconfig linuxconfig 6 Jul 25
10:20 /home/linuxconfig/sandbox
```

Tenga en cuenta el número asociado `2`como se muestra en la salida anterior. Este número indica el número de enlaces duros asociados con un inodo específico.

NOTA

En esta etapa, es importante comprender que no existe una diferencia real entre el archivo original `sandbox`y el `/tmp/hardlink1`archivo recién creado . Ambos señalan el mismo **inodo** usando diferentes nombres de archivo.

4. Elimine el enlace duro usando `unlink`o `rm`comando.

```
$ unlink sandbox
$ ls -l /tmp/hardlink1
-rw-rw-r--. 1 linuxconfig linuxconfig 6 Jul 25
10:20 /tmp/hardlink1
```

En este caso, ambos comandos `rm`o `unlink`eliminarán un enlace rígido pero no los datos y el inodo asociados reales. Como el `sandbox`enlace duro se ha eliminado, solo queda `1`un enlace duro asociado con el inodo original. A continuación, eliminaremos el último enlace duro asociado con este archivo:

```
$ rm /tmp/hardlink1
```

En este punto, el enlace al inodo que apunta al contenido de nuestro archivo original se pierde, por lo tanto, consideramos que este archivo se ha eliminado. Si no hay enlaces duros que apuntan a un inodo, el sistema de archivos ahora puede sobrescribir la ubicación de este inodo con nuevos datos.

¿SABÍAS?
Puede eliminar cualquier archivo (dado que tiene los permisos adecuados) mediante el comando de desvinculación? Pruebalo ahora:

```
$ touch file
$ unlink file
```

Si entendió los comandos anteriores, entonces ha dominado los enlaces duros GNU / Linux como se explica en este tutorial.

Enlaces simbólicos

Además de los enlaces duros, también hay un tipo diferente de enlaces disponibles en el sistema operativo GNU / Linux. **Los enlaces simbólicos** pueden cruzar sistemas de archivos, y

también es posible crear un **enlace simbólico** de un directorio. Sin embargo, los enlaces simbólicos en lugar del **inodo** real , solo enlazan a los enlaces duros existentes (nombre de archivo o directorio). Por esta razón, si el enlace real al que apunta el enlace simbólico se elimina, el enlace simbólico se rompe.

1. Primero creemos algunos objetos de sandbox para jugar. En este caso `mydir`crearemos un directorio llamado y dentro de este directorio crearemos un archivo llamado `myfile`.

   ```
   $ mkdir mydir
   $ touch mydir/myfile
   ```

2. A continuación, crearemos un nuevo enlace simbólico del directorio existente mydir utilizando el `ln`comando con una combinación de `-s`opciones.

   ```
   $ ln -s ~/mydir /tmp/symdir
   ```

   Ahora, hemos creado un enlace simbólico llamado `symdir`ubicado dentro del `/tmp`directorio.

   ```
   $ cd /tmp/
   $ ls -l symdir
   lrwxrwxrwx. 1 linuxconfig linuxconfig 23 Jul 25
   14:05 symdir -> /home/linuxconfig/mydir
   ```

   Tenga en cuenta el primer carácter de la salida anterior. En este caso, el carácter `l`indica que estamos tratando con un enlace simbólico.

   NOTA

   Al crear enlaces simbólicos, tenga en cuenta que el `ln`comando almacena la ruta real proporcionada como cadena. Si no está dentro del mismo directorio, en muchos casos debe proporcionar la ruta completa de pedido para que funcione el enlace simbólico

   El recién creado `symdir`debe contener el archivo previamente creado `myfile`:

```
$ cd symdir
$ ls
myfile
$pwd
/tmp/symdir
```

Ejercicios

1. Juega con enlaces simbólicos. Cree un enlace simbólico a un archivo después de eso elimine el archivo original y vea qué sucedió con su enlace simbólico.
2. Qué sucede cuando ejecuta un `ls`comando con un solo argumento. Por ejemplo, ejecute el siguiente comando `ln -s /etc/services`. ¿Que pasó?
3. Determine si necesita ser el propietario del archivo para crear un enlace simbólico. ¿Se aplica la misma regla para los enlaces duros?