



## 104.6 Criar e alterar links simbólicos e hardlinks

### Referência ao LPI objective

LPIC-1 version 5.0, Exam 101, Objective 104.6

### Peso

2

### Áreas chave de conhecimento

- Criar links.
- Identificar links simbólicos e/ou hardlinks.
- Copiar arquivos versus criar links de arquivos.
- Usar links para dar suporte a tarefas de administração do sistema.

### Segue uma lista parcial dos arquivos, termos e utilitários utilizados

- `ln`
- `ls`



# 104.6 Lição 1

|                      |  |
|----------------------|--|
| <b>Certificação:</b> | LPIC-1   |
| <b>Versão:</b>       | 5.0  |
| <b>Tópico:</b>       | 104 Dispositivos, sistemas de arquivos do Linux, hierarquia padrão de sistemas de arquivos |
| <b>Objetivo:</b>     | 104.6 Criar e alterar links físicos e simbólicos   |
| <b>Lição:</b>        | 1 de 1   |

## Introdução

No Linux, alguns arquivos recebem um tratamento especial, seja devido ao local em que estão armazenados, como os arquivos temporários, ou à maneira como interagem com o sistema de arquivos, como os links. Nesta lição, aprenderemos o que são os links e como gerenciá-los.

## Compreendendo os links

Já dissemos que, no Linux, tudo é tratado como um arquivo. Mas existe um tipo *especial* de arquivo chamado *link*. Há dois tipos de links em um sistema Linux:

### Links simbólicos

Também chamados de *soft links*, eles apontam para o caminho de outro arquivo. Se excluirmos o arquivo para o qual o link aponta (chamado *destino*), o link ainda existirá, mas “para de funcionar”, pois passará a apontar para “nada”.

### Links físicos

Um link físico é como um segundo nome para o arquivo original. Eles *não* são duplicatas,

mas sim uma entrada adicional no sistema de arquivos que aponta para o mesmo local (inode) no disco.

**TIP**

Um *inode* é uma estrutura de dados que armazena os atributos de um objeto (como um arquivo ou diretório) em um sistema de arquivos. Entre esses atributos estão as permissões, proprietário e os blocos do disco nos quais estão armazenados os dados referentes àquele objeto. São semelhantes a um verbete em um índice; por isso o nome, que vem de “index node”.

## Trabalhando com links físicos

### Criando links físicos

O comando para criar um link físico no Linux é `ln`. A sintaxe básica é:

```
$ ln TARGET LINK_NAME
```

O `TARGET` já deve existir (trata-se do arquivo para o qual o link apontará) e, se o destino não estiver no diretório atual ou se você deseja criar o link em outro lugar, é *obrigatório* especificar o caminho completo para ele. Por exemplo, o comando

```
$ ln target.txt /home/carol/Documents/hardlink
```

Cria um arquivo chamado `hardlink` no diretório `/home/carol/Documents/`, vinculado ao arquivo `target.txt` no diretório atual.

Se deixarmos de fora o último parâmetro (`LINK_NAME`), será criado um link com o mesmo nome do destino no diretório atual.

### Gerenciando os links físicos

Os links físicos são entradas no sistema de arquivos que têm nomes diferentes, mas que apontam para os mesmos dados no disco. Todos esses nomes são equivalentes e podem ser usados para se referir a um arquivo. Se você alterar o conteúdo de um dos nomes, o conteúdo de todos os outros nomes que apontam para aquele arquivo será alterado, já que todos apontam para os mesmos dados. Se você excluir um dos nomes, os outros nomes continuarão a funcionar.

Isso acontece porque quando você “exclui” um arquivo, os dados não são realmente apagados do disco. O sistema simplesmente exclui a entrada na tabela do sistema de arquivos apontando para o inode correspondente aos dados no disco. Mas se houver uma segunda entrada apontando para o

mesmo inode, ainda será possível acessar os dados. É como se fossem duas estradas convergindo no mesmo ponto. Mesmo se bloquearmos ou redirecionarmos uma das estradas, ainda será possível chegar ao destino usando a outra.

Para conferir, podemos usar o parâmetro `-li` de `ls`. Considere o seguinte conteúdo de um diretório:

```
$ ls -li
total 224
3806696 -r--r--r-- 2 carol carol 111702 Jun  7 10:13 hardlink
3806696 -r--r--r-- 2 carol carol 111702 Jun  7 10:13 target.txt
```

O número anterior às permissões é o número inode. Percebeu que tanto o arquivo `hardlink` quanto o arquivo `target.txt` têm o mesmo número (3806696)? Isso ocorre porque um é o link físico do outro.

Mas qual é o original e qual é o link? Não dá pra dizer, já que, na prática, eles são a mesma coisa.

Observe que todo link físico que aponta para um arquivo aumenta a *contagem de links* do arquivo. Este é o número logo após as permissões na saída de `ls -li`. Por padrão, todo arquivo tem uma contagem de links de 1 (os diretórios têm uma contagem de 2), e cada link físico apontando para ele aumenta a contagem em um. Por isso a contagem de links é 2 nos arquivos da lista acima.

Ao contrário dos links simbólicos, só é possível criar links físicos para arquivos, e tanto o link quanto o destino devem residir no mesmo sistema de arquivos.

## Movendo e removendo links físicos

Como os links físicos são tratados como arquivos regulares, eles podem ser excluídos com `rm` e renomeados ou movidos no sistema de arquivos com `mv`. E como um link rígido aponta para o mesmo inode do destino, ele pode ser movido livremente, sem medo de “quebrar” o link.

## Links simbólicos

### Criando links simbólicos

O comando usado para criar um link simbólico também é `ln`, mas com o parâmetro `-s` adicionado. Assim:

```
$ ln -s target.txt /home/carol/Documents/softlink
```

Criamos assim um arquivo chamado `softlink` no diretório `/home/carol/Documents/`, apontando

para o arquivo `target.txt` no diretório atual.

Como no caso dos links físicos, é possível omitir o nome do link para criar um link com o mesmo nome do destino no diretório atual.

## Gerenciando links simbólicos

Os links simbólicos apontam para outro caminho no sistema de arquivos. Podemos criar links simbólicos para arquivos e diretórios, mesmo em diferentes partições. É muito fácil identificar um link simbólico na saída de `ls`:

```
$ ls -lh
total 112K
-rw-r--r-- 1 carol carol 110K Jun  7 10:13 target.txt
lrwxrwxrwx 1 carol carol   12 Jun  7 10:14 softlink -> target.txt
```

No exemplo acima, o primeiro caractere nas permissões para o arquivo `softlink` é `l`, indicando um link simbólico. Além disso, logo após o nome do arquivo, vemos o nome do destino para o qual o link aponta, o arquivo `target.txt`.

Note que nas listagens de arquivos e diretórios os links simbólicos sempre mostram as permissões `rwx` para o usuário, o grupo e outros, mas, na prática, as permissões de acesso para eles são as mesmas do destino.

## Movendo e removendo links simbólicos

Como no caso dos links físicos, os links simbólicos podem ser removidos usando `rm` e movidos ou renomeados com `mv`. No entanto, deve-se tomar cuidado especial ao criá-los, para evitar “quebrar” o link se ele for movido de seu local original.

Ao criar links simbólicos, é preciso estar ciente de que, a menos que um caminho seja totalmente especificado, o local do destino será interpretado como sendo *relativo* ao local do link. Isso pode criar problemas caso o link ou o arquivo para o qual ele aponta seja movido.

É mais fácil entender isso com um exemplo. Digamos que temos um arquivo chamado `original.txt` no diretório atual e desejamos criar um link simbólico para ele chamado `softlink`. Poderíamos usar:

```
$ ln -s original.txt softlink
```

E aparentemente tudo correria bem. Vamos verificar com `ls`:

```
$ ls -lh
total 112K
-r--r--r-- 1 carol carol 110K Jun  7 10:13 original.txt
lrwxrwxrwx 1 carol carol   12 Jun  7 19:23 softlink -> original.txt
```

Observe como o link é elaborado: `softlink` aponta para (`->`) `original.txt`. Entretanto, vamos ver o que acontece se movermos o link para o diretório anterior e tentarmos exibir seu conteúdo com o comando `less`:

```
$ mv softlink ../
$ less ../softlink
../softlink: No such file or directory
```

Como o caminho para `original.txt` não foi especificado, o sistema pressupõe que ele está no mesmo diretório que o link. Quando isso deixa de ser verdade, o link para de funcionar.

Para evitar isso, devemos sempre especificar o caminho completo para o destino ao criar o link:

```
$ ln -s /home/carol/Documents/original.txt softlink
```

Dessa forma, o link continuará a funcionar mesmo se for movido, porque aponta para a localização absoluta do destino. Verifique com `ls`:

```
$ ls -lh
total 112K
lrwxrwxrwx 1 carol carol   40 Jun  7 19:34 softlink ->
/home/carol/Documents/original.txt
```

## Exercícios Guiados

1. Qual é o parâmetro de `chmod` no modo *simbólico* para ativar o sticky bit em um diretório?

2. Imagine que existe um arquivo chamado `document.txt` no diretório `/home/carol/Documents`. Com qual comando criaríamos um link simbólico para ele chamado `text.txt` no diretório atual?

3. Explique a diferença entre um link físico para um arquivo e uma cópia desse arquivo.

## Exercícios Exploratórios

- Imagine que dentro de um diretório você crie um arquivo chamado `recipes.txt`. Dentro deste diretório, você também criará um link físico para este arquivo, chamado `receitas.txt`, e um link simbólico (ou *soft*) para este chamado `rezepte.txt`.

```
$ touch recipes.txt
$ ln recipes.txt receitas.txt
$ ln -s recipes.txt rezepte.txt
```

O conteúdo do diretório deve aparecer assim:

```
$ ls -lhi
total 160K
5388833 -rw-r--r-- 4 carol carol 77K jun 17 17:25 receitas.txt
5388833 -rw-r--r-- 4 carol carol 77K jun 17 17:25 recipes.txt
5388837 lrwxrwxrwx 1 carol carol 12 jun 24 10:12 rezepte.txt -> receitas.txt
```

Lembre-se que, por ser um link físico, `receitas.txt` aponta para o mesmo inode que `recipes.txt`. O que aconteceria com o link `rezepte.txt` se o nome `receitas.txt` fosse excluído? Por quê?

- Imagine que você tem um pendrive conectado ao sistema e montado em `/media/youruser/FlashA`. Você deseja criar em seu diretório pessoal um link chamado `schematics.pdf` apontando para o arquivo `esquema.pdf` no diretório raiz do pendrive. Então, você digita o comando:

```
$ ln /media/youruser/FlashA/esquema.pdf ~/schematics.pdf
```

O que aconteceria? Por quê?

- Considere a seguinte saída de `ls -lah`:

```
$ ls -lah
total 3,1M
drwxr-xr-x 2 carol carol 4,0K jun 17 17:27 .
drwxr-xr-x 5 carol carol 4,0K jun 17 17:29 ..
-rw-rw-r-- 1 carol carol 2,8M jun 17 15:45 compressed.zip
-rw-r--r-- 4 carol carol 77K jun 17 17:25 document.txt
```



```
-rw-rw-r-- 1 carol carol 216K jun 17 17:25 image.png
-rw-r--r-- 4 carol carol  77K jun 17 17:25 text.txt
```

- Quantos links apontam para o arquivo `document.txt`?

- Trata-se de links físicos ou simbólicos?

- Qual parâmetro você deveria passar para `ls` para ver qual inode é ocupado por cada arquivo?

4. Imagine que você tem, em seu diretório `~/Documents`, um arquivo chamado `clients.txt` contendo alguns nomes de clientes e um diretório chamado `somedir`. Dentro dele, existe um arquivo *diferente, também* chamado `clients.txt`, contendo nomes diferentes. Para replicar essa estrutura, use os seguintes comandos:

```
$ cd ~/Documents
$ echo "John, Michael, Bob" > clients.txt
$ mkdir somedir
$ echo "Bill, Luke, Karl" > somedir/clients.txt
```

Em seguida você cria um link dentro de `somedir` chamado `partners.txt` apontando para esse arquivo, com os comandos:

```
$ cd somedir/
$ ln -s clients.txt partners.txt
```

Assim, a estrutura do diretório é:

```
Documents
|-- clients.txt
'-- somedir
    |-- clients.txt
    '-- partners.txt -> clients.txt
```

Agora, você move `partners.txt` de `somedir` para `~/Documents`, e lista seu conteúdo.

```
$ cd ~/Documents/  
$ mv somedir/partners.txt .  
$ less partners.txt
```

O link ainda funciona? Se sim, qual arquivo terá seu conteúdo listado? Por quê?

5. Considere os arquivos a seguir:

```
-rw-r--r-- 1 carol carol 19 Jun 24 11:12 clients.txt  
lrwxrwxrwx 1 carol carol 11 Jun 24 11:13 partners.txt -> clients.txt
```

Quais são as permissões de acesso de `partners.txt`? Por quê?

## Resumo

Nesta lição, você aprendeu:

- O que são links.
- A diferença entre os links *simbólicos* e *físicos*.
- Como criar links.
- Como movê-los, renomeá-los ou excluí-los.

Os seguintes comandos foram abordados nesta lição:

- `ln`: O comando “link”. Sozinho, este comando cria um link físico. Com a opção `-s`, podemos criar um link *simbólico* ou *soft link*. Lembre-se de que os links físicos só podem residir na mesma partição e sistema de arquivos, e que os links simbólicos podem atravessar partições e sistemas de arquivos (inclusive armazenamento conectado em rede)
- O parâmetro `-i` do `ls`, que permite visualizar o número inode de um arquivo.

## Respostas aos Exercícios Guiados

1. Qual é o parâmetro de `chmod` no modo *simbólico* para ativar o sticky bit em um diretório?

O símbolo do sticky bit no modo simbólico é `t`. Como queremos habilitar (adicionar) essa permissão no diretório, o parâmetro deve ser `+t`.

2. Imagine que existe um arquivo chamado `document.txt` no diretório `/home/carol/Documents`. Com qual comando criaríamos um link simbólico para ele chamado `text.txt` no diretório atual?

`ln -s` é o comando para criar um link simbólico. Como é necessário especificar o caminho completo do arquivo para o qual estamos criando o link, o comando seria:

```
$ ln -s /home/carol/Documents/document.txt text.txt
```

3. Explique a diferença entre um link físico para um arquivo e uma cópia desse arquivo.

Um link físico é apenas outro nome para um arquivo. Mesmo que pareça uma duplicata do arquivo original, para todos os efeitos, o link e o original são iguais, pois apontam para os mesmos dados no disco. As alterações feitas no conteúdo do link serão refletidas no original e vice-versa. Uma cópia é uma entidade completamente independente, ocupando um lugar diferente no disco. As alterações na cópia não serão refletidas no original e vice-versa.

## Respostas aos Exercícios Exploratórios

1. Imagine que dentro de um diretório você crie um arquivo chamado `recipes.txt`. Dentro deste diretório, você também criará um link físico para este arquivo, chamado `receitas.txt`, e um link simbólico (ou *soft*) para este chamado `rezepte.txt`.

```
$ touch recipes.txt
$ ln recipes.txt receitas.txt
$ ln -s recipes.txt rezepte.txt
```

O conteúdo do diretório deve aparecer assim:

```
$ ls -lhi
total 160K
5388833 -rw-r--r-- 4 carol carol 77K jun 17 17:25 receitas.txt
5388833 -rw-r--r-- 4 carol carol 77K jun 17 17:25 recipes.txt
5388837 lrwxrwxrwx 1 carol carol 12 jun 24 10:12 rezepte.txt -> recipes.txt
```

Lembre-se que, por ser um link físico, `receitas.txt` aponta para o mesmo inode que `recipes.txt`. O que aconteceria com o link `rezepte.txt` se o nome `receitas.txt` fosse excluído? Por quê?

O link simbólico `rezepte.txt` deixaria de funcionar, porque os links simbólicos apontam para nomes, e não inodes, e o nome `receitas.txt` não existe mais, mesmo que os dados ainda estejam no disco com o nome `recipes.txt`.

2. Imagine que você tem um pendrive conectado ao sistema e montado em `/media/youruser/FlashA`. Você deseja criar em seu diretório pessoal um link chamado `schematics.pdf` apontando para o arquivo `esquema.pdf` no diretório raiz do pendrive. Então, você digita o comando:

```
$ ln /media/youruser/FlashA/esquema.pdf ~/schematics.pdf
```

O que aconteceria? Por quê?

O comando falharia. A mensagem de erro seria `Invalid cross-device link` (Link inválido entre dispositivos), esclarecendo o motivo: os links físicos não podem apontar para um destino em uma partição ou dispositivo diferente. A única maneira de criar um link como esse é usar um link *simbólico* ou *soft*, adicionando o parâmetro `-s` a `ln`.

3. Considere a seguinte saída de `ls -lah`:

```
$ ls -lah
total 3,1M
drwxr-xr-x 2 carol carol 4,0K jun 17 17:27 .
drwxr-xr-x 5 carol carol 4,0K jun 17 17:29 ..
-rw-rw-r-- 1 carol carol 2,8M jun 17 15:45 compressed.zip
-rw-r--r-- 4 carol carol 77K jun 17 17:25 document.txt
-rw-rw-r-- 1 carol carol 216K jun 17 17:25 image.png
-rw-r--r-- 4 carol carol 77K jun 17 17:25 text.txt
```

- Quantos links apontam para o arquivo `document.txt`?

Todos os arquivos começam com `1` na contagem de links. Como a contagem de links desse arquivo é `4`, existem três links apontando para ele.

- Trata-se de links físicos ou simbólicos?

São links físicos, já que os links simbólicos não aumentam a contagem de links de um arquivo.

- Qual parâmetro você deveria passar para `ls` para ver qual inode é ocupado por cada arquivo?

O parâmetro é `-li`. O inode será mostrado na primeira coluna da saída de `ls`, como abaixo:

```
$ ls -lahi
total 3,1M
5388773 drwxr-xr-x 2 carol carol 4,0K jun 17 17:27 .
5245554 drwxr-xr-x 5 carol carol 4,0K jun 17 17:29 ..
5388840 -rw-rw-r-- 1 carol carol 2,8M jun 17 15:45 compressed.zip
5388833 -rw-r--r-- 4 carol carol 77K jun 17 17:25 document.txt
5388837 -rw-rw-r-- 1 carol carol 216K jun 17 17:25 image.png
5388833 -rw-r--r-- 4 carol carol 77K jun 17 17:25 text.txt
```

4. Imagine que você tem, em seu diretório `~/Documents`, um arquivo chamado `clients.txt` contendo alguns nomes de clientes e um diretório chamado `somedir`. Dentro dele, existe um arquivo *diferente, também* chamado `clients.txt`, contendo nomes diferentes. Para replicar essa estrutura, use os seguintes comandos:

```
$ cd ~/Documents
```

```
$ echo "John, Michael, Bob" > clients.txt
$ mkdir somedir
$ echo "Bill, Luke, Karl" > somedir/clients.txt
```

Em seguida você cria um link dentro de `somedir` chamado `partners.txt` apontando para esse arquivo, com os comandos:

```
$ cd somedir/
$ ln -s clients.txt partners.txt
```

Assim, a estrutura do diretório é:

```
Documents
|-- clients.txt
'-- somedir
    |-- clients.txt
    '-- partners.txt -> clients.txt
```

Agora, você move `partners.txt` de `somedir` para `~/Documents`, e lista seu conteúdo.

```
$ cd ~/Documents/
$ mv somedir/partners.txt .
$ less partners.txt
```

O link ainda funciona? Se sim, qual arquivo terá seu conteúdo listado? Por quê?

Isto é quase uma “pegadinha”, mas o link funcionará e o arquivo listado será o que está em `~/Documents`, contendo os nomes `John, Michael, Bob`.

Lembre-se de que, como não especificamos o caminho completo para o alvo `clients.txt` ao criar o link simbólico `partners.txt`, o local do destino será interpretado como sendo relativo ao local do link, que neste caso é o diretório atual.

Quando o link é movido de `~/Documents/somedir` para `~/Documents`, ele deveria parar de funcionar, já que o destino não estava mais no mesmo diretório do link. Porém, coincidentemente existe um arquivo chamado `clients.txt` em `~/Documents`, de forma que o link apontará para esse arquivo em vez do destino original dentro de `~/somedir`.

Para evitar isso, sempre especifique o caminho completo para o destino ao criar um link

simbólico.

5. Considere os arquivos a seguir:

```
-rw-r--r-- 1 carol carol 19 Jun 24 11:12 clients.txt  
lrwxrwxrwx 1 carol carol 11 Jun 24 11:13 partners.txt -> clients.txt
```

Quais são as permissões de acesso de `partners.txt`? Por quê?

As permissões de acesso de `partners.txt` são `rw-r--r--`, já que os links sempre herdam as mesmas permissões de acesso do alvo.