Ссылки

Linux предоставляет два разных типа ссылок для файлов в файловой системе: жесткие и мягкие. Обе предоставляют возможность иметь более одного пути, который ссылается на один и тот же файл, но у каждой есть свои преимущества и недостатки.

Мягкие ссылки

Мягкие ссылки, также известные как символические ссылки, представляют собой файлы, предназначенные для указания на другие файлы с помощью путей. Например, файл **/etc/grub.conf** является символической ссылкой, указывающей на файл **/boot/grub/grub.conf**.

Обычно пользователи размещают значки на рабочем столе, при нажатии на которые запускается какая-либо программа или открывается файл. Эти значки на самом деле являются файлами, которые указывают на другие файлы; другими словами, эти значки являются мягкими ссылками.

В случае файла **grub.conf** этот файл конфигурации изначально был сохранен в каталоге **/etc.** Когда разработчики впоследствии переместили его в каталог **/boot/grub**, была создана мягкая ссылка, чтобы администраторам было легче находить новое местоположение (или продолжать использовать старое имя файла, которое теперь указывает на новое имя файла).).

Мягкие ссылки очень наглядны по сравнению с жесткими. Это связано с тем, что мягкие ссылки различаются по типу файла. Например, подробный список файла /etc/grub.conf показывает, что это символическая ссылка:

```
lrwxrwxrwx. 1 root root 22 Nov  6  2012 /etc/grub.conf -> ../boot/grub/grub.conf
```

Хотя это полезно для обычных пользователей, иногда это может быть недостатком, поскольку некоторые программы могут отказываться работать с файлом типа символьной ссылки и могут потребовать обычный файл. В подробном списке символической ссылки обратите внимание, что первым символом, предшествующим разрешениям, является буква l.

```
lrwxrwxrwx. 1 root root 22 Nov  6  2012 /etc/grub.conf -> ../boot/grub/grub.conf
```

Еще одна вещь, которую стоит заметить при перечислении мягкой ссылки, это то, что за именем файла ссылки следует стрелка, которая указывает на путь. При доступе к файлу **/etc/grub.conf** следует ссылка, а вместо этого - файл **/boot/grub/grub.conf**. Например, команда **more /etc/grub.conf** действительно отобразит содержимое файла **/boot/grub/grub.conf.**

```
lrwxrwxrwx. 1 root root 22 Nov  6  2012 /etc/grub.conf -> ../boot/grub/grub.conf
```

Разрешения, отображаемые в мягкой ссылке, определяют, кто может пытаться перейти по этой ссылке. Как правило, разрешения для софт-ссылки являются **rwx** для всех пользователей. Разрешения для файла, с которым была связана ссылка, будут определять фактические или эффективные разрешения, которые будут иметь пользователи, если они попытаются перейти по ссылке.

Например, разрешения для файла **/etc/grub.conf** имеют вид **rwxrwxrwx**, что указывает на то, что у всех будет полный доступ для перехода по ссылке. Однако проверка разрешений для файла **/boot/grub/grub.conf** показывает, что только пользователь **root** будет иметь доступ к файлу:

```
-rw-------. 1 root root 2279 Apr 29  2013 /boot/grub/grub.conf
```

Чтобы создать файл мягкой ссылки, используйте команду **ln** с опцией **-s**. Первый аргумент - это исходное имя файла, а второй аргумент - это имя создаваемой ссылки.

Обязательно поместите аргументы в правильном порядке, так как невозможно создать ссылку из имени файла, которое уже существует:

```
touch file1.txt
ln -s ./file2.txt ./file1.txt
ln: failed to create symbolic link `./file1.txt': File exists
```

Поменяйте местами аргументы и символическая ссылка будет успешно создана.

```
ln -s ./file1.txt ./file2.txt
ls -l ./file*
-rw-rw-r-- 1 sysadmin sysadmin 0 Jul 27 04:12 ./file1.txt
lrwxrwxrwx 1 sysadmin sysadmin 11 Jul 27 04:13 ./file2.txt ->
./file1.txt
```

В отличие от жестких ссылок, файлы мягких ссылок не увеличивают количество ссылок, связанных с обычным файлом. В вышеприведенном примере номер счетчика ссылок для **./file1.txt** останется равным единице, независимо от того, сколько файлов мягкой или символической ссылки было создано для ссылки на этот файл. Количество ссылок - это число, следующее сразу за полем разрешений.

Жесткие ссылки

Файлы жестких ссылок похожи на обычные файлы, за исключением того, что они совместно используют индекс с другим файлом. Это означает, что хотя имена жестко связанных файлов могут быть разными, все остальное в них идентично. Это иногда дает жестким ссылкам преимущество перед программными ссылками в том, что программы не могут различить обычный файл и файл с жесткой ссылкой.

Когда создается жесткая ссылка, значение счетчика ссылок увеличивается на единицу. Например, в следующем примере демонстрируется использование команды ln (без опции -s) для создания файла жесткой ссылки. Обратите внимание на количество ссылок (число после разрешений) до и после создания жесткой ссылки:

```
cd /tmp
cp /etc/hosts hosts
ls -l hosts
```

```
sysadmin@L-FW:~$ cp /etc/hosts /tmp
sysadmin@L-FW:~$ cd /tmp
sysadmin@L-FW:/tmp$ ls -l hosts
-rw-r--r-- 1 sysadmin sysadmin 186 May 10 21:33 hosts
```

```
ln hosts myhosts
ls -l hosts myhosts
```

```
sysadmin@L-FW:/tmp$ ln hosts myhosts
sysadmin@L-FW:/tmp$ ls -l hosts myhosts
-rw-r--r-- 2 sysadmin sysadmin 186 May 10 21:33 hosts
-rw-r--r-- 2 sysadmin sysadmin 186 May 10 21:33 myhosts
```

При просмотре файлов жестких ссылок параметр **ls -i** может быть полезен для проверки того, что файлы на самом деле совместно используют индекс, поскольку этот параметр приводит к отображению номера индекса перед разрешениями. Например, обратите внимание, что следующие два файла имеют **одинаковый номер индекса**:

```
sysadmin@L-FW:/tmp$ ls -li hosts myhosts
93 -rw-r--r-- 2 sysadmin sysadmin 186 May 10 21:33 hosts
93 -rw-r--r-- 2 sysadmin sysadmin 186 May 10 21:33 myhosts
```

Одним из недостатков мягких ссылок является то, что нелегко определить, есть ли у файла мягкая ссылка, указывающая на него. Например, посмотрите на вывод следующей команды ls:

```
ls -l dash
```

```
sysadmin@L-FW:/bin$ ls -l  dash
-rwxr-xr-x 1 root root 117208 Jan 24  2017 dash
sysadmin@L-FW:/bin$ ls -l  sh
lrwxrwxrwx 1 root root 4 Jan 24  2017 sh -> dash
```

Файл **/bin/sh** является символической ссылкой на файл /bin/dash, но это не очевидно, если посмотреть на исходный файл.

С другой стороны, обычный файл со значением счетчика жестких ссылок больше единицы - это файл с жесткой ссылкой. Используя номер индекса файла, можно найти все связанные файлы, используя команду **find** с опцией **-inum.** Например:

```
ls -i hosts
find –inum 93
```

```
sysadmin@L-FW:/bin$ cd /tmp
sysadmin@L-FW:/tmp$ ls -i hosts
93 hosts
sysadmin@L-FW:/tmp$ find -inum 93
find: ‘./systemd-private-6e192c8e3c574ec397c055f468eb4a0f-systemd-ti
ion denied
./myhosts
./hosts
find: ‘./data’: Permission denied
```

Примечание.

Если файлы жестко связаны друг с другом, то каждая жесткая ссылка является оригиналом. Если одна ссылка удалена, остальные все еще работают, то есть файл не удален. Пока остается одна жесткая ссылка, данные файла могут быть доступны.

Это не похоже на мягкие ссылки, в которых данные хранятся в указанном файле, а это означает, что если исходный файл удален, все мягкие ссылки теперь ни на что не указывают.

Мягкие ссылки против жестких ссылок

Что лучше: мягкие или жесткие ссылки? Ответ не так прост, поскольку он действительно зависит от нескольких критериев:

Преимущества жестких ссылок перед мягкими ссылками:

- Жестко связанные файлы неотличимы от программ из обычных файлов.
- Если файлы жестко связаны, то они всегда содержатся в одной файловой системе.
- Удаление жестких ссылок не удаляет фактические данные, если вы не удалите все жесткие ссылки.

Преимущества мягких ссылок перед жесткими:

- Мягкие ссылки могут быть сделаны на файл каталога; жесткие ссылки не могут.
- Мягкие ссылки могут быть сделаны из файла в одной файловой системе на файл в другой файловой системе; жесткие ссылки не могут.
- Мягкие ссылки очень наглядны, потому что вывод команды ls -l показывает, на какой файл указывает мягкая ссылка.

В общем случае, если вам нужно сослаться на файл в другой файловой системе или на каталог, то мягкие ссылки - это правильный тип для использования. В противном случае вы должны использовать жесткие ссылки.