

(/linux/)

(<https://www.baeldung.com/linux/>)

Абсолютные и относительные символические ссылки и перемещение относительной символической ссылки

Последнее обновление: 28 февраля 2023 г.



Автор: Хикс Герганов

(<https://www.baeldung.com/linux/author/hiksgerganov>)

Просмотр файлов

(<https://www.baeldung.com/linux/category/files/viewing>)

ln (<https://www.baeldung.com/linux/tag/ln>)

readlink (<https://www.baeldung.com/linux/tag/readlink>)

1. Введение

Связывание файлов ([/linux/absolute-and-relative-symlinks](https://www.baeldung.com/linux/absolute-and-relative-symlinks)) позволяет получить доступ к файлам с помощью нескольких имен и путей. Существует два типа ссылок:

- **символическая ссылка (symlink)** ([/linux/symbolic-and-hard-links#symbolic-links](https://www.baeldung.com/linux/symbolic-and-hard-links#symbolic-links)), указывающая только на исходный файл (target)
- **жесткая ссылка** ([/linux/symbolic-and-hard-links#hard-links](https://www.baeldung.com/linux/symbolic-and-hard-links#hard-links)), представляющая другой путь к **индексу** ([/linux/inodes#3-link](https://www.baeldung.com/linux/inodes#3-link)) целевого объекта

Файлы исчезают, как только удаляются все их жесткие ссылки. С другой стороны, символические ссылки - это в основном доступность и удобство организации. Таким образом, мы можем даже создавать их с относительными, а не абсолютными (полными) путями ([/linux/bash-expand-relative-path#differences-in-absolute-and-relative-paths](https://www.baeldung.com/linux/bash-expand-relative-path#differences-in-absolute-and-relative-paths)) к исходной цели. В этом случае они называются относительными символическими ссылками.

В этом руководстве мы рассмотрим стандартный способ перемещения или копирования относительной символической ссылки (symlink) без разрыва ее связи с источником. Сначала мы определяем относительные и абсолютные символические ссылки. Далее мы произведем преобразование между двумя типами ссылок. Наконец, мы покажем, как безопасно переместить относительную символическую ссылку.

Мы протестировали код из этого руководства на Debian (<https://www.debian.org/>) 11 (Bullseye) с GNU (<https://www.gnu.org/>) Bash (<https://www.gnu.org/software/bash/manual/bash.html>) 5.1.4. Он должен работать в большинстве POSIX ([/linux/posix](https://www.baeldung.com/linux/posix))-совместимых сред.

2. Относительные и абсолютные символические ссылки

Как правило, относительные пути имеют несколько недостатков ([/linux/vim-find-full-path-current-file#why-use-absolute-paths](https://www.baeldung.com/linux/vim-find-full-path-current-file#why-use-absolute-paths)) по сравнению с абсолютными путями.

(/linux/)

(<https://www.baeldung.com/linux/>)

(/bael-search)

Тем не менее, относительный путь в символической ссылке полезен:

- отсутствует жестко заданная структура внешней **файловой системы** (/linux/filesystems)
- **одновременное перемещение относительной ссылки и целевого объекта сохраняет их взаимосвязь**

Таким образом, относительные символические ссылки могут быть очень удобны для таких действий, как установка, развертывание и перенос, при условии сохранения их взаимосвязи.

Естественно, мы создаем ссылки с помощью стандартной команды `ln` (<https://pubs.opengroup.org/onlinepubs/9699919799/utilities/ln.html>) POSIX. Итак, **давайте воспользуемся `ln` (/linux/linking-to-files) для создания относительных и абсолютных символьных ссылок на файл:**

```
$ ln --symbolic --relative /file /filesymrel  
$ ln --symbolic /file /filesymabs
```



Теперь мы можем проверить, на что указывает каждая из них с помощью `ls` (<https://man7.org/linux/man-pages/man1/ls.1.html>):

```
$ ls -l /file*  
-rw-r--r-- 1 baeldung baeldung 0 Jan 10 01:00 /file  
lrwxrwxrwx 1 baeldung baeldung 5 Jan 10 01:01 /filesymabs -> /file  
lrwxrwxrwx 1 baeldung baeldung 4 Jan 10 01:01 /filesymrel -> file
```



Важно отметить, что **путь, на который указывает `/filesymrel`, не начинается с косой черты**. Действительно, это означает, что это относительная символическая ссылка.

(/linux/)

(https://www.baeldung.com/linux/)

(/bael-search)

Давайте попробуем прочитать из нее до и после перемещения:

```
$ cat /filesymrel
Content.
$ mv /filesymrel /dir/
$ cat /dir/filesymrel
cat: /dir/filesymrel: No such file or directory
```

Как и ожидалось, в отличие от абсолютных ссылок, мы не можем прочитать символическую ссылку, как только ее местоположение относительно целевого объекта изменяется. Что мы можем с этим поделать?

3. Преобразование относительной и абсолютной символической ссылки

Аналогично путям, мы всегда можем переключаться между относительными и абсолютными символическими ссылками. Для этого мы используем две команды:

- ***readlink*** (/linux/absolute-directory-of-file#1-readlink) с его флагом ***-f*** или ***-canonicalize***, который извлекает абсолютный целевой путь из символьной ссылки
- ***ln*** для воссоздания и *принудительной* перезаписи *символьной* ссылки

Фактически, мы можем применить обе команды одновременно. Давайте возьмем нашу предыдущую символическую ссылку */ filesymrel*:

```
$ readlink /filesymrel
file
```

Сначала, мы преобразуем относительную символическую ссылку в абсолютную символическую ссылку:

`(/linux/)`

`(https://www.baeldung.com/linux/)`

`(/bael-search)`

```
$ ln --force --symbolic "$(readlink --canonicalize /filesymrel)" /filesymrel
```



Теперь мы можем перепроверить целевую ссылку:

```
$ readlink /filesymrel  
/file
```



Как и ожидалось, теперь это абсолютный путь.

Далее, мы преобразуем абсолютную символическую ссылку в *относительную* символическую ссылку:

```
$ ln --force --symbolic --relative "$(readlink --canonicalize /filesymrel)"  
/filesymrel
```



На этом этапе у нас снова есть относительная ссылка:

```
$ readlink /filesymrel  
file
```



Теперь пришло время применить эту концепцию к перемещению или копированию.

4. Сохраняйте относительную символическую ссылку при изменении местоположения (/linux/)

(<https://www.baeldung.com/linux/>)

После ознакомления с относительными символическими ссылками давайте попробуем переместить их, не нарушая их функции. (/bael-search)

Для начала мы создаем относительную символическую ссылку на */file* и подтверждаем ее путь, а также содержимое целевого файла:

```
$ ln --symbolic --relative /file /filesymrel
$ readlink /filesymrel
file
$ cat /filesymrel
Content.
```

Пришло время повторить проблему с перемещением.

4.1. Проблема с перемещением относительной символической ссылки

Теперь мы попробуем скопировать символическую ссылку напрямую в другое место и убедимся, что она не работает:

```
$ mv /filesymrel /dir/
$ readlink /dir/filesymrel
file
$ cat /dir/filesymrel
cat: /dir/filesymrel: No such file or directory
```

Важно отметить, что если бы файл находился в `/dir/file`, мы бы теперь видели его содержимое вместо содержимого `/file`, исходного целевого файла. Поскольку это тоже не то, чего мы хотим, давайте посмотрим, как мы можем уменьшить путаницу. (`/linux/`)
(<https://www.baeldung.com/linux/>)

4.2. Преобразование и перемещение символической ссылки (/bael-search)

На этом этапе мы можем попробовать наше решение:

```
$ ln --force --symbolic "$(readlink --canonicalize /filesymrel)" /filesymrel
$ mv /filesymrel /dir/filesymrel
$ ln --force --symbolic --relative "$(readlink --canonicalize /dir/filesymrel)"
/dir/filesymrel
```

Здесь есть три шага:

1. Преобразуйте относительную символическую ссылку в абсолютную символическую ссылку
2. Переместите символическую ссылку по мере необходимости
3. Преобразуйте перемещенную символическую ссылку обратно в относительную

Далее мы проверяем путь и целевое содержимое перемещенного `/dir / файла-символа`:

```
$ readlink /dir/filesymrel
../file
$ cat /dir/filesymrel
Content.
```

Как и ожидалось, теперь мы можем прочитать содержимое целевого объекта по символической ссылке, потому что ее путь правильный.

На самом деле, это решение можно оптимизировать.

4.3. Прямое создание символической ссылки в пункте назначения

Мы можем создать новую символическую ссылку непосредственно на основе исходной в пути назначения:

(/linux/)

(https://www.baeldung.com/linux/)

(/bael-search)

```
$ ln --symbolic --relative "$(readlink --canonicalize /filesymrel)"  
/dir/filesymrel  
$ readlink /dir/filesymrel  
../file  
$ cat /dir/filesymrel  
Content.
```

После этого мы можем удалить оригинал, при желании.

5. Краткое описание

В этой статье мы обсудили стандартный способ перемещения относительных символических ссылок без разрыва соединения с их целью. По сути, это сводится к одной команде:

```
$ ln --symbolic --relative "$(readlink --canonicalize SOURCE_SYMBOLIC_LINK)"  
DESTINATION_SYMBOLIC_LINK
```

В заключение, стандартный инструмент *ln* Linux позволяет нам перемещать и копировать относительные символические ссылки, не нарушая их.

Комментарии к этой статье закрыты!

(/linux/)

(https://www.baeldung.com/linux/)

(/bael-search)

КАТЕГОРИИ

АДМИНИСТРИРОВАНИЕ (/LINUX/CATEGORY/ADMINISTRATION)

FILES (/LINUX/CATEGORY/FILES)

FILESYSTEMS (/LINUX/CATEGORY/FILESYSTEMS)

INSTALLATION (/LINUX/CATEGORY/INSTALLATION)

NETWORKING (/LINUX/CATEGORY/NETWORKING)

PROCESSES (/LINUX/CATEGORY/PROCESSES)

SCRIPTING (/LINUX/CATEGORY/SCRIPTING)

SEARCH (/LINUX/CATEGORY/SEARCH)

SECURITY (/LINUX/CATEGORY/SECURITY)

WEB (/LINUX/CATEGORY/WEB)

СЕРИИ

АДМИНИСТРИРОВАНИЕ LINUX (/LINUX/LINUX-ADMINISTRATION-SERIES)

LINUX FILES (/LINUX/LINUX-FILES-SERIES)

LINUX PROCESSES (/LINUX/LINUX-PROCESSES-GUIDE)

О НАС

О BAELDUNG (/ABOUT)

THE FULL ARCHIVE (HTTPS://WWW.BAELDUNG.COM/LINUX/FULL_ARCHIVE)

EDITORS ([HTTPS://WWW.BAELDUNG.COM/EDITORS](https://www.baeldung.com/editors))

TERMS OF SERVICE ([HTTPS://WWW.BAELDUNG.COM/TERMS-OF-SERVICE](https://www.baeldung.com/terms-of-service))

PRIVACY POLICY ([HTTPS://WWW.BAELDUNG.COM/PRIVACY-POLICY](https://www.baeldung.com/privacy-policy))

COMPANY INFO ([HTTPS://WWW.BAELDUNG.COM/BAELDUNG-COMPANY-INFO](https://www.baeldung.com/baeldung-company-info))

CONTACT ([/CONTACT](/contact))

(</bael-search>)