



[Главная](#) >> [Терминал](#) >> Символические и жесткие ссылки Linux

Символические и жесткие ссылки Linux

Опубликовано: 27 декабря, 2016 от [admin](#) , 12 комменариев, время чтения: 6 минут

Символические и жесткие ссылки – это особенность файловой системы Linux, которая позволяет размещать один и тот же файл в нескольких директориях. Это очень похоже на ярлыки в Windows, так как файл на самом деле остается там же где и был, но вы можете на него сослаться из любого другого места.

В Linux существует два типа ссылок на файлы. Это символические и жесткие ссылки Linux. Они очень сильно отличаются и каждый тип имеет очень важное значение. В этой небольшой статье мы рассмотрим чем же отличаются эти ссылки, зачем они нужны, а также как создавать ссылки на файлы в Linux.

Содержание статьи:

- [Символические ссылки](#)
- [Жесткие ссылки](#)
- [Использование ссылок в Linux](#)
 - [Создание символических ссылок](#)
 - [Создание жестких ссылок](#)
- [Выводы](#)



Символические ссылки

Символические ссылки более всего похожи на обычные ярлыки. Они содержат адрес нужного файла в вашей файловой системе. Когда вы пытаетесь открыть такую ссылку, то открывается целевой файл или папка. Главное ее отличие от жестких ссылок в том, что при удалении целевого файла ссылка останется, но она будет указывать в никуда, поскольку файла на самом деле больше нет.

Вот основные особенности символических ссылок:

- Могут ссылаться на файлы и каталоги;
- После удаления, перемещения или переименования файла становятся недействительными;
- Права доступа и номер `inode` отличаются от исходного файла;
- При изменении прав доступа для исходного файла, права на ссылку останутся неизменными;
- Можно ссылаться на другие разделы диска;
- Содержат только имя файла, а не его содержимое.

Теперь давайте рассмотрим жесткие ссылки.

Жесткие ссылки

Этот тип ссылок реализован на более низком уровне файловой системы. Файл размещен только в определенном месте жесткого диска. Но на это место могут ссылаться несколько ссылок из файловой системы. Каждая из ссылок – это отдельный файл, но ведут они к одному участку жесткого диска. Файл можно перемещать между каталогами, и все ссылки останутся рабочими, поскольку для них неважно имя. Рассмотрим особенности:

- Работают только в пределах одной файловой системы;
- Нельзя ссылаться на каталоги;
- Имеют ту же информацию `inode` и набор разрешений что и у исходного файла;
- Разрешения на ссылку изменяться при изменении разрешений файла;
- Можно перемещать и переименовывать и даже удалять файл без вреда ссылке.

Использование ссылок в Linux

Теоретические отличия вы знаете, но осталось закрепить все это на практике, поэтому давайте приведем несколько примеров работы со ссылками в Linux. Для создания символических ссылок существует утилита `ln`. Ее синтаксис очень прост:

```
$ ln опции файл_источник файл_ссылки
```

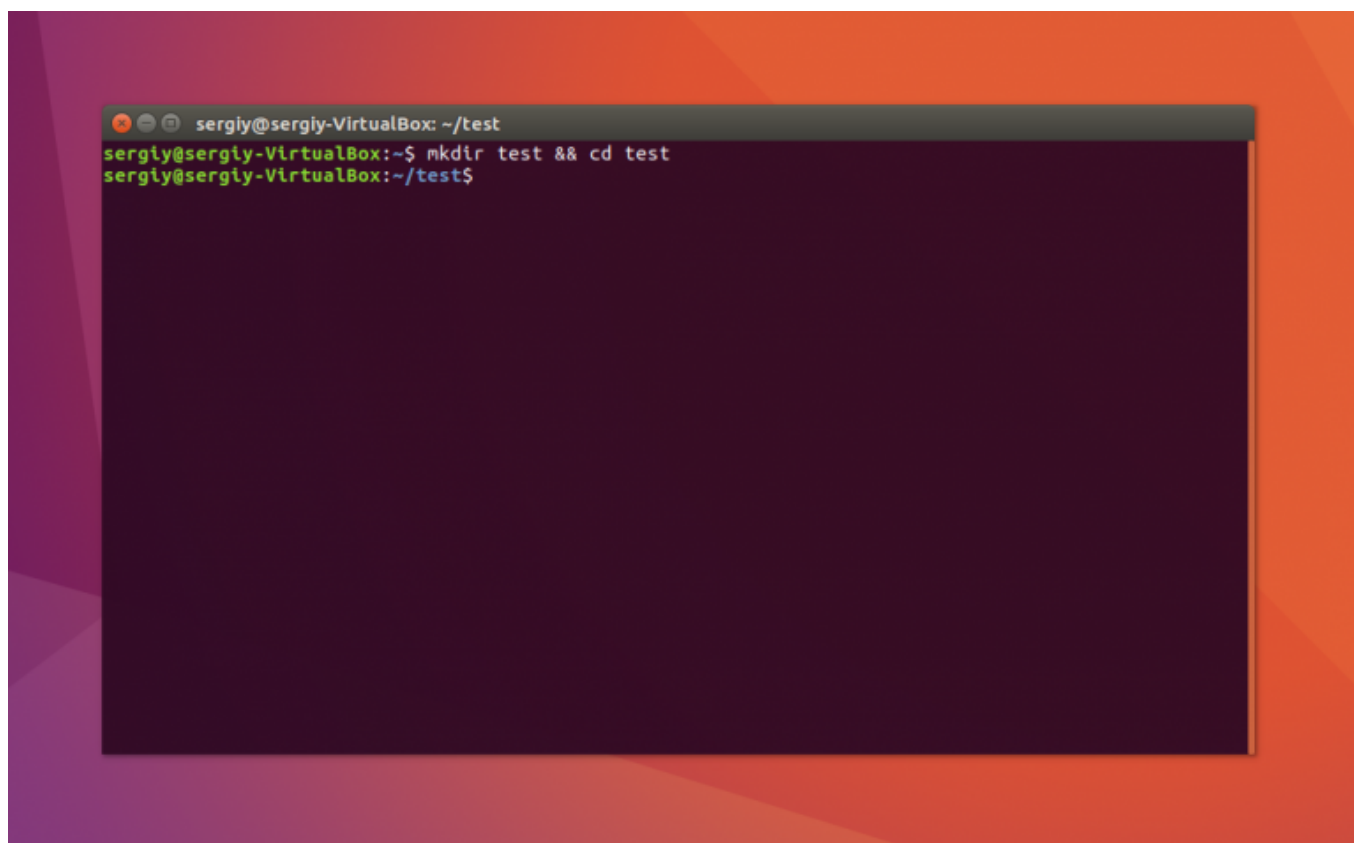
Рассмотрим опции утилиты:

- **-d** - разрешить создавать жесткие ссылки для директорий суперпользователю;
- **-f** - удалять существующие ссылки;
- **-i** - спрашивать нужно ли удалять существующие ссылки;
- **-P** - создать жесткую ссылку;
- **-r** - создать символическую ссылку с относительным путем к файлу;
- **-s** - создать символическую ссылку.

Создание символических ссылок

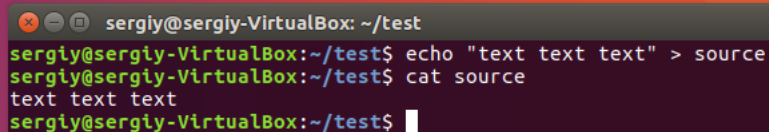
Сначала создайте папку test и перейдите в нее:

```
$ mkdir test && cd test
```



Затем создайте файл с именем source с каким-либо текстом:

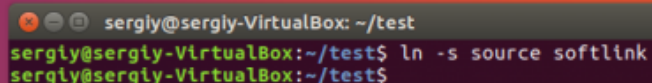
```
$ echo "текст текст текст текст" > source  
$ cat source
```

A terminal window titled 'sergiy@sergiy-VirtualBox: ~/test'. The user enters the command 'echo "text text text" > source', which creates a file named 'source' containing the text 'text text text'. Then, the user enters 'cat source', which displays the contents of the file: 'text text text'.

```
sergiy@sergiy-VirtualBox: ~/test
sergiy@sergiy-VirtualBox:~/test$ echo "text text text" > source
sergiy@sergiy-VirtualBox:~/test$ cat source
text text text
sergiy@sergiy-VirtualBox:~/test$
```

Файл готов, дальше создадим символическую ссылку Linux, для этого используется команда `ln` с опцией `-s`:

```
$ ln -s source softlink
```

A terminal window titled 'sergiy@sergiy-VirtualBox: ~/test'. The user enters the command 'ln -s source softlink', which creates a symbolic link named 'softlink' pointing to the file 'source'.

```
sergiy@sergiy-VirtualBox: ~/test
sergiy@sergiy-VirtualBox:~/test$ ln -s source softlink
sergiy@sergiy-VirtualBox:~/test$
```

Попробуем посмотреть содержимое файла по ссылке:

Privacy

```
$ cat softlink
```



```
sergiy@sergiy-VirtualBox: ~/test
sergiy@sergiy-VirtualBox:~/test$ cat softlink
text text text
sergiy@sergiy-VirtualBox:~/test$
```

Как видите, нет никакой разницы между ней и исходным файлом. Но утилита `ls` покажет что это действительно ссылка:

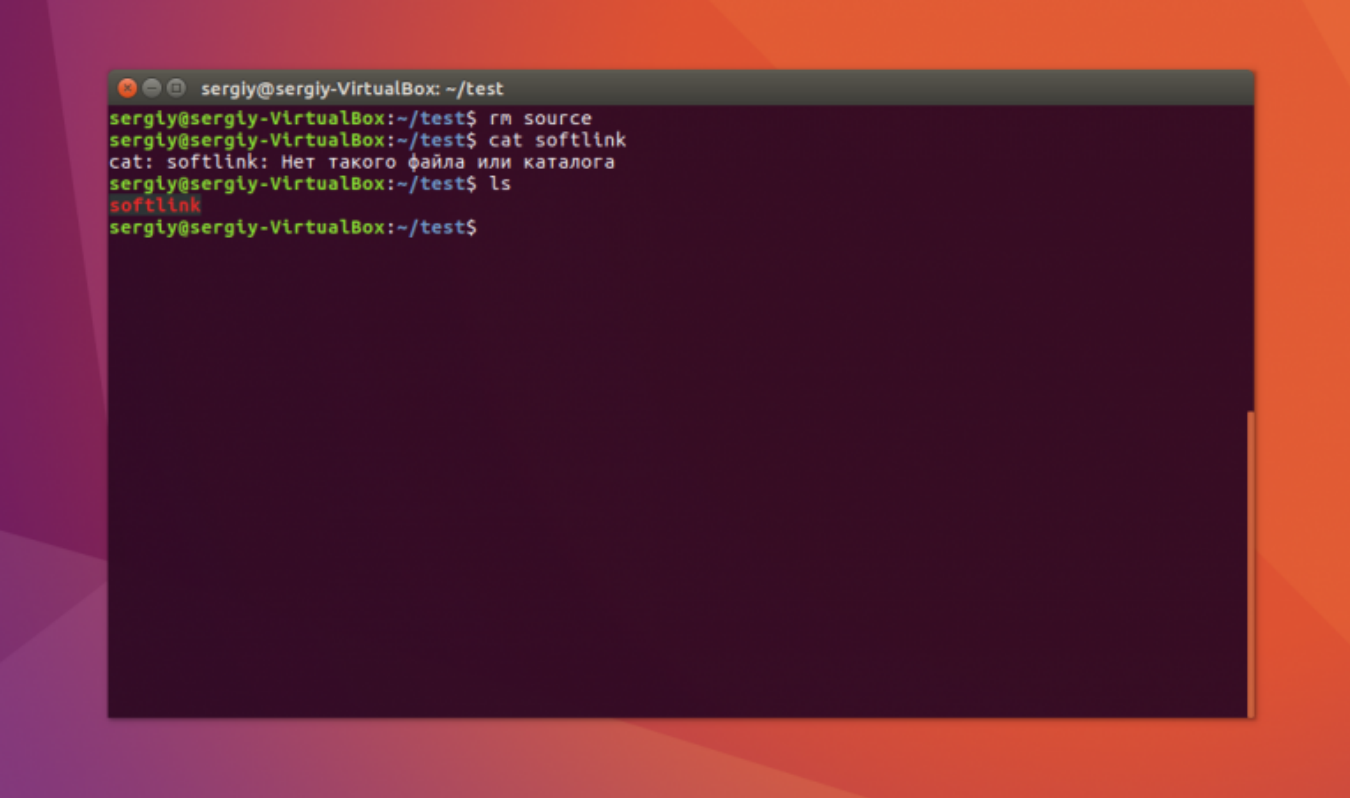
```
$ ls -li
```

```
sergiy@sergiy-VirtualBox: ~/test
sergiy@sergiy-VirtualBox:~/test$ cat softlink
text text text
sergiy@sergiy-VirtualBox:~/test$ ls -li
итого 4
673865 lrwxrwxrwx 1 sergiy sergiy 6 Дек 26 13:16 softlink -> source
673864 -rw-r--r-- 1 sergiy sergiy 15 Дек 26 13:16 source
sergiy@sergiy-VirtualBox:~/test$
```

Несмотря на то, что содержимое одинаковое, здесь мы видим, что адрес иномы и права доступа к файлам отличаются, кроме того, явно показано что это символическая ссылка Linux.

Теперь удалите исходный файл и посмотрите что будет:

```
$ cat softlink
```



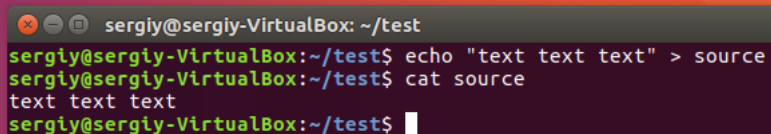
```
sergiy@sergiy-VirtualBox: ~/test
sergiy@sergiy-VirtualBox:~/test$ rm source
sergiy@sergiy-VirtualBox:~/test$ cat softlink
cat: softlink: Нет такого файла или каталога
sergiy@sergiy-VirtualBox:~/test$ ls
softlink
sergiy@sergiy-VirtualBox:~/test$
```

Вы получите ошибку, что такого файла не существует, потому что мы действительно удалили исходный файл. Если вы удалите ссылку, то исходный файл останется на месте.

Создание жестких ссылок

Снова создайте файл source с произвольным текстом:

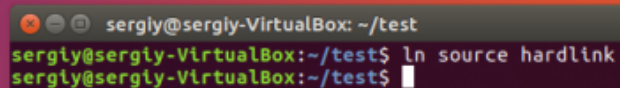
```
$ echo "текст текст текст текст" > source
$ cat source
```



```
sergiy@sergiy-VirtualBox: ~/test
sergiy@sergiy-VirtualBox:~/test$ echo "text text text" > source
sergiy@sergiy-VirtualBox:~/test$ cat source
text text text
sergiy@sergiy-VirtualBox:~/test$
```

Теперь создадим жесткую ссылку Linux. Для этого достаточно вызвать утилиту без параметров:

```
$ ln source hardlink
```

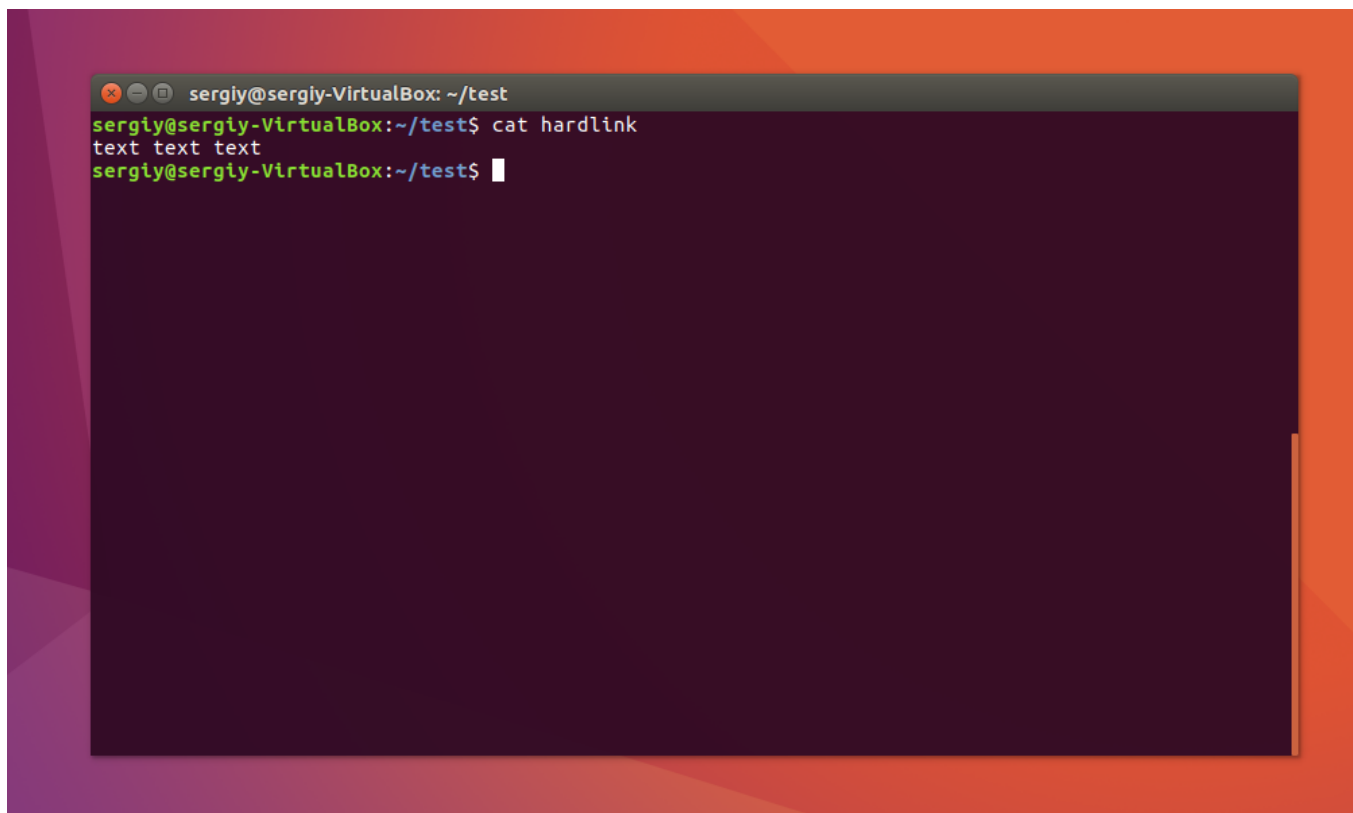


```
sergiy@sergiy-VirtualBox: ~/test
sergiy@sergiy-VirtualBox:~/test$ ln source hardlink
sergiy@sergiy-VirtualBox:~/test$
```

Посмотрите содержимое файла:

Privacy

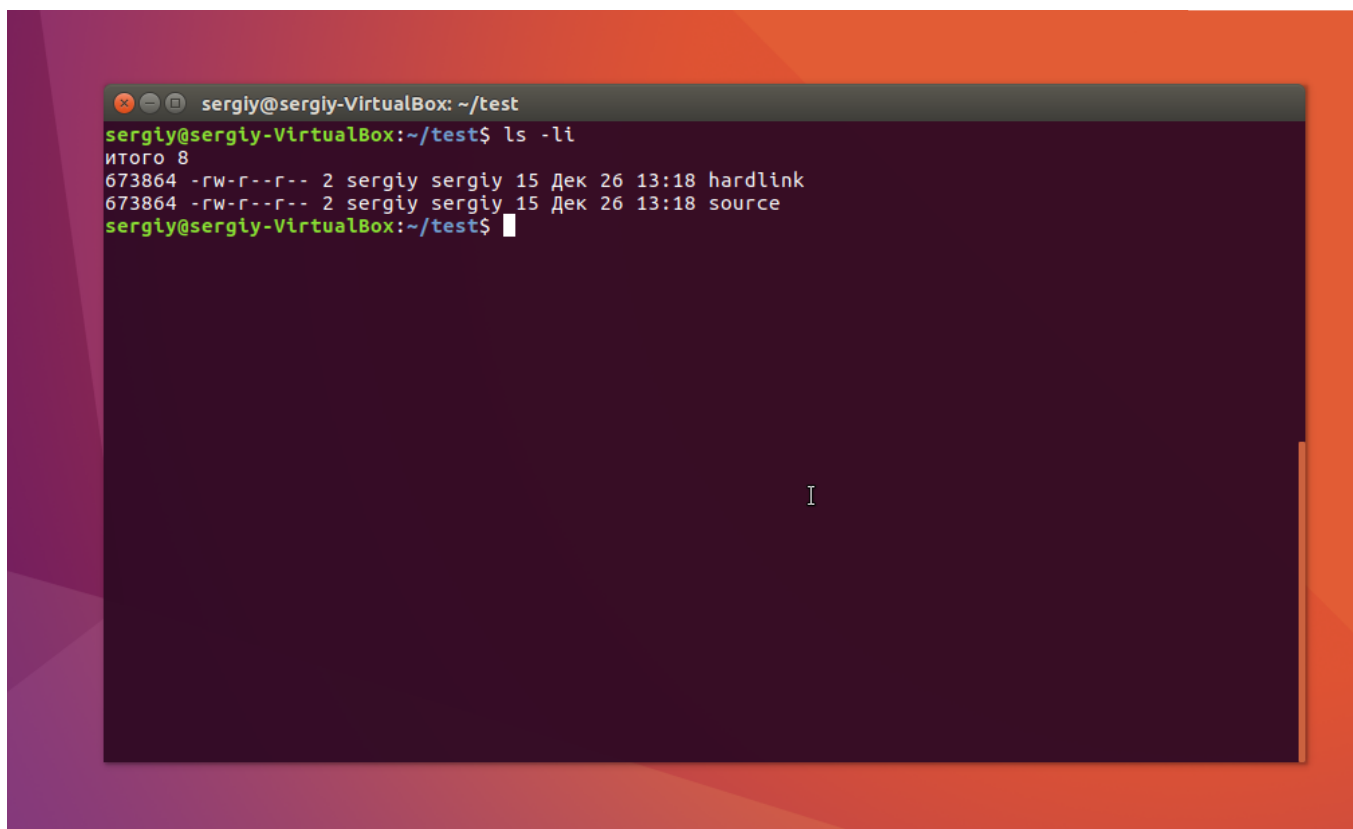

```
$ cat hardlink
```



```
sergiy@sergiy-VirtualBox: ~/test
sergiy@sergiy-VirtualBox:~/test$ cat hardlink
text text text
sergiy@sergiy-VirtualBox:~/test$
```

Данные те же самые, а если мы посмотрим вывод утилиты `ls`, то увидим что `inode` и права доступа тоже совпадают:

```
$ ls -li
```

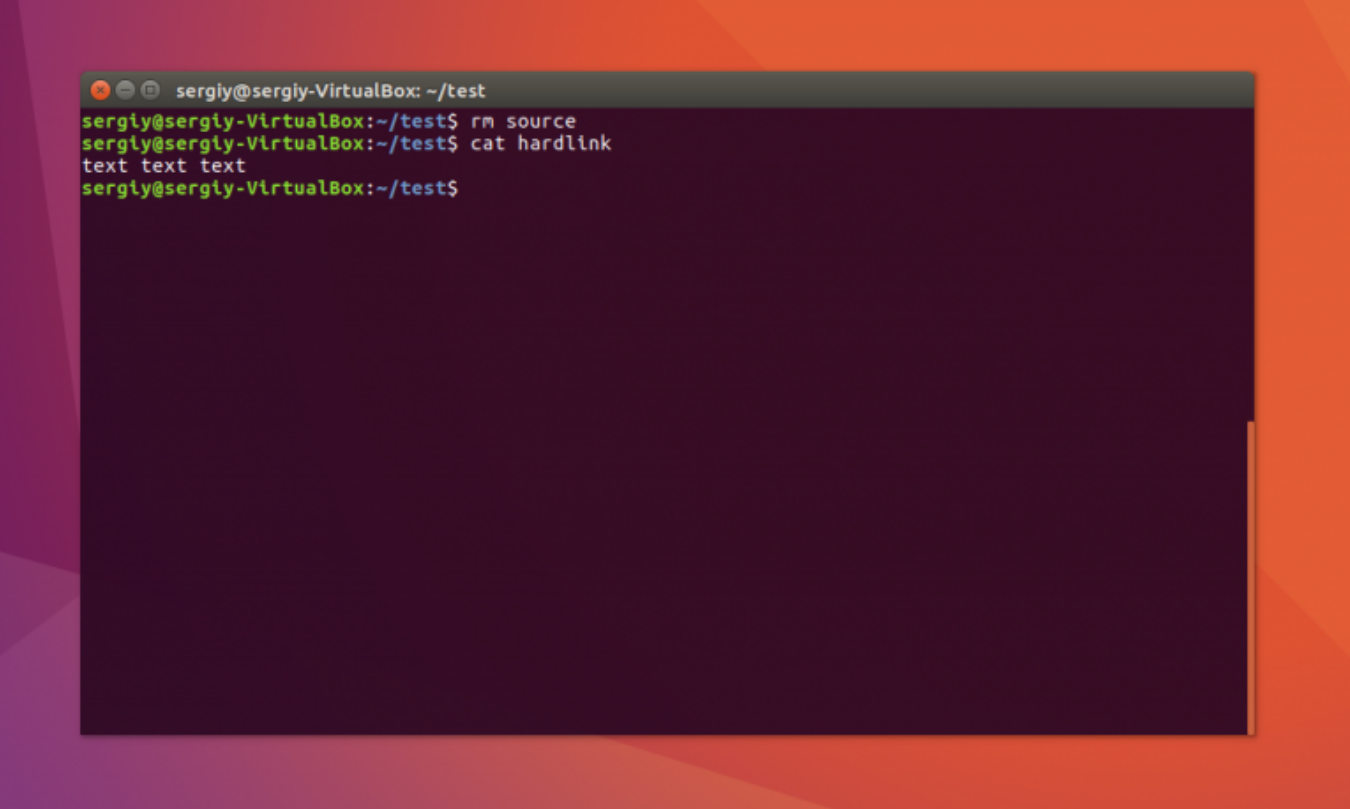
A terminal window titled 'sergiy@sergiy-VirtualBox: ~/test' is shown. The user has entered the command 'ls -li'. The output shows two files: 'hardlink' and 'source', both with permissions '-rw-r--r--', owned by 'sergiy', and located in the '/test' directory. The file 'hardlink' has a size of 673864 bytes and was created on December 26, 2013, at 13:18. The file 'source' also has a size of 673864 bytes and was created on the same date and time. The terminal prompt is 'sergiy@sergiy-VirtualBox:~/test\$'.

Если для одного из файлов поменять разрешения, то они изменяться и у другого. Теперь удалите исходный файл:

```
$ rm source
```

Затем посмотрите содержимое:

```
$ cat hardlink
```



```
sergiy@sergiy-VirtualBox: ~/test
sergiy@sergiy-VirtualBox:~/test$ rm source
sergiy@sergiy-VirtualBox:~/test$ cat hardlink
text text text
sergiy@sergiy-VirtualBox:~/test$
```

Как видите, ничего не произошло и ссылка по-прежнему указывает на нужный участок диска, это главное отличие жесткой ссылки от символической. Мы можем сделать вывод, что жесткая ссылка `linux` это обычный файл. Каждый файл имеет как минимум одну ссылку, но для некоторых мы можем создать несколько ссылок.

Выводы

Это все, что вам было необходимо знать про символические и жесткие ссылки `linux`. Надеюсь, вы получили общее представление об этих возможностях файловой системы и сможете использовать их для решения своих задач.

На завершение видео про ссылки в `Linux`:



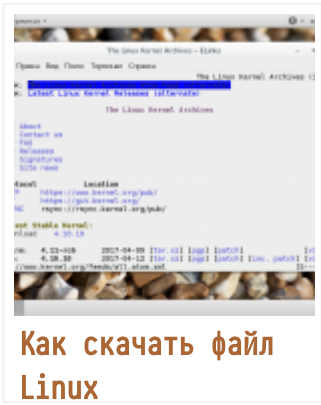
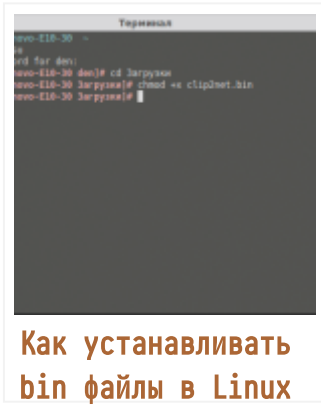
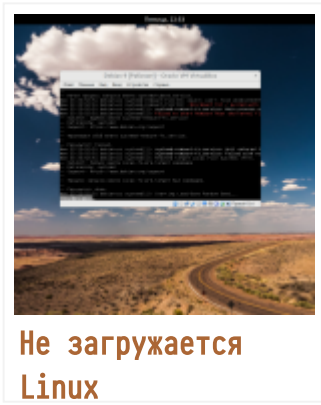
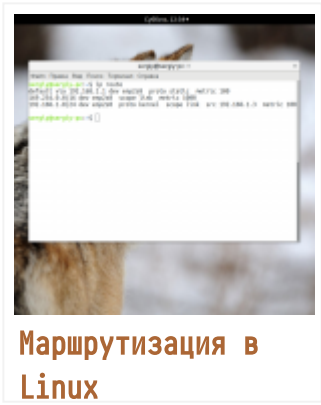
символические ссылки в linux



Обнаружили ошибку в тексте? Сообщите мне об этом. Выделите текст с ошибкой и нажмите Ctrl+Enter.

Похожие записи





Оцените статью

★★★★★ (39 оценок, среднее: 4,79 из 5)

📁 [Терминал](#)

Об авторе



ADMIN

Основатель и администратор сайта losst.ru, увлекаюсь открытым программным обеспечением и операционной системой Linux. В качестве основной ОС сейчас использую Ubuntu. Кроме Linux, интересуюсь всем, что связано с информационными технологиями и современной наукой.



Privacy

12 комментариев к “Символические и жесткие ссылки Linux”

**eugene**13 апреля, 2017 в 9:28 дп

Наглядно. Спасибо

[Ответить](#)**артем**26 ноября, 2017 в 4:03 дп

Почему не создаются ссылки в директориях сайта на сервере
символическая ссылка красная и не работает
жесткая ссылка вообще не создается пишет: ln: failed to access
права на папку такие же как везде
в остальных директориях все создается и работает

[Ответить](#)**Павел**3 декабря, 2018 в 1:43 пп

Потому, что жёсткие ссылки могут ссылаться лишь на файл, расположенный на том же логическом разделе жёсткого диска. Для мягких ссылок подобных ограничений нет,

[Ответить](#)**vlad**15 января, 2019 в 8:55 дп

Не до конца раскрыты понятия жесткая и символическая ссылка. В Линукс вся информация о файле привязана не к его имени, а к так называемому числовому индексному дескриптору, в котором указаны сведения об этом файле: в каких блоках диска хранится его содержимое, размер, время создания, модификации и др.

Индексные дескрипторы файлов хранятся в специальной таблице и имеют числовое значение, что не очень удобно для запоминания и использования, поэтому им удобнее дать имя.

Имя файла (можно создать несколько), ссылающееся на его индексный дескриптор, называется жесткой ссылкой. Но есть несколько ограничений:

- их можно создавать только на файлы, но не на каталоги;
- жесткую ссылку нельзя создать на файл, находящийся на другом диске.

А символическая ссылка (или мягкая) указывает на жесткие ссылки, не на индексный дескриптор.

[Ответить](#)



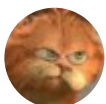
vlad

[18 января, 2019 в 8:01 дп](#)

Дополнение:

Жесткая ссылка (регулярная) это для обычных типов файлов, не специальных (каталог, сокет, блочное устройство и т.д.), но у специальных типов файлов имеются свои дескрипторы (inode).

[Ответить](#)



Stepan

[19 февраля, 2019 в 10:04 пп](#)

Какое же убогое описание...

[Ответить](#)

**Practic75**10 июля, 2019 в 9:20 дп

Не понятно в каких случаях, что использовать

[Ответить](#)**Виктор**15 января, 2020 в 1:27 пп

Статья очень вовремя: я уже пробовал создавать ссылки на рабочем столе, но они не работали, оказывается их надо создавать специальной программой. И нельзя мягкие ссылки копировать, а жёсткие?

[Ответить](#)**N**27 января, 2022 в 7:47 пп

В статье есть ошибка.

Невозможно создать жёсткую ссылку на директорию, соответственно, ключ `-d` является ошибочным.

[Ответить](#)**Алексей**16 июня, 2022 в 11:52 дп

Не то что бы ключ `-d` в статье ошибочно указан. Данный ключ в справке команды `ln` указан, и описано это так:

" `-d`, `-F`, `--directory` позволить суперпользователю создавать жесткие ссылки на каталоги (замечание: вероятно, это не получится из-за системных ограничений, даже для

Privacy

суперпользователя)"

Т.е. как будто есть редкие случаи, когда это возможно.

[Ответить](#)



анон

[25 октября, 2022 в 4:03 пп](#)

не поняла какой смысл создавать жесткие ссылки когда можно скопировать файл

[Ответить](#)



Андрей

[28 мая, 2023 в 7:43 дп](#)

чтобы место на диске не занимать в два раза больше

[Ответить](#)

Оставьте комментарий

Имя *

Email

☐ Я прочитал и принимаю политику конфиденциальности. Подробнее [Политика конфиденциальности](#) *

Privacy

Комментировать


Русский

Поиск

ПОИСК ПО КОМАНДАМ

Начните вводить команду

Поиск



Начните изучать
Linux прямо сейчас!

Карта сайта



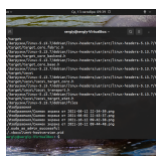
Как пользоваться редактором Vim

Полезно

Лучшие

Свежие

Теги



Команда find в Linux

2021-10-17



Команда chmod Linux

2020-04-13



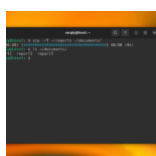
Права доступа к файлам в Linux

2020-10-09



Настройка Cron

2021-10-01



Копирование файлов в Linux

2023-03-03



Privacy

РАССЫЛКА

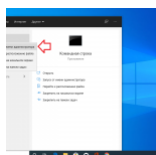
Ваш E-Mail адрес

☐ Я прочитал(а) и принимаю политику конфиденциальности

Sign up

Windows

Списки



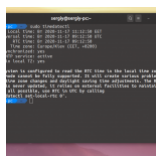
Восстановление Grub после установки Windows 10

2020-08-15



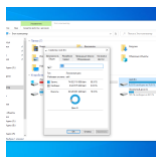
Ошибка Ubuntu не видит сеть Windows

2023-02-18



Сбивается время в Ubuntu и Windows

2023-02-18



Подключение ext4 в Windows

2023-02-18

Смотреть ещё

ОБНАРУЖИЛИ ОШИБКУ?



Privacy

Сообщите мне об этом. Выделите текст с ошибкой и нажмите Ctrl+Enter.

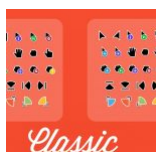
МЕТА

[Регистрация](#)[Войти](#)[Лента записей](#)[Лента комментариев](#)

СЛЕДИТЕ ЗА НАМИ В СОЦИАЛЬНЫХ СЕТЯХ



ИНТЕРЕСНОЕ



Лучшие темы курсоров Linux

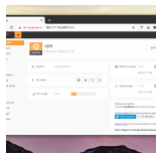
2020-12-18



Команды Linux для работы с файлами

2020-04-14

Privacy



Лучшие панели управления для Linux

2020-12-08



Команды терминала Linux

2020-12-18

©Losst 2023 CC-BY-SA [Политика конфиденциальности](#)

