















4.8.11. Файли-посилання

|  |  |
| --- | --- |
|  | ***Файл-посилання****- це особливий тип файла, який, по суті, вказує на інший файл або місце розташування у файловій системі комп’ютера. Такі файли часто використовують для більш легкого доступу до файлів.* |

У Linux існує два види файлів-посилань: **жорсткі** та **символьні**.

**Символьні посилання** - це файли особливого типу, єдиним вмістом яких є довільний рядок, який може вказувати на існуючий файл. Тобто, символьні посилання містять адресу потрібного файла у файловій системі. Коли ви звертаєтеся до символьного посилання, насправді, ви звертаєтеся до файла, на який посилання вказує.

|  |  |
| --- | --- |
|  | *При видаленні цільового файла символьне посилання залишиться, але воно буде вказувати в нікуди, оскільки файла, насправді, більше немає. Тобто, якщо перейменувати початковий файл або перемістити його у інший каталог, символьне посилання залишається, але втрачає працездатність.* |

Символьні посилання можна помітити у файловому менеджері - їх значки схожі на звичайні файли, але зі стрілкою внизу значка. Тому вони найбільше схожі на звичайні ярлики, як у Windows, так як файл, насправді, залишається там, де і був, але на нього можна посилатися з будь-якого іншого місця.

Основні властивості символьних посилань:

* можуть посилатися на файли і каталоги;
* після видалення, переміщення або перейменування файла стають недійсними;
* права доступу і номер [inode](https://linuxguide.rozh2sch.org.ua/" \l "inode) відрізняються від початкового файла;
* права доступу для символьних посилань значення не мають (права доступу завжди rwxrwxrwx);
* при зміні прав доступу для початкового файла, права на посилання залишаться незмінними;
* можна посилатися на інші розділи диска;
* містять тільки ім’я файла, а не його вміст.

**Жорстке посилання** також дозволяє швидко отримати доступ до ресурсу з іншого місця або під іншим ім’ям. Жорсткі посилання - це інша назва того самого файла. Це дозволяє мати один і той же файл у декількох місцях.

Особливість жорстких посилань полягає у тому, що вони прямо вказують на номер індексного дескриптора [inode](https://linuxguide.rozh2sch.org.ua/" \l "inode) і, відповідно, такі імена можуть вказувати тільки на файли всередині однієї й тієї ж самої файлової системи, тобто на тому ж самому носієві даних, на якому знаходиться файл, що містить це ім’я.

Це може здатися трохи дивним, але жорстке посилання означає, що у нас є доступ до одного файла з двох (або більше) місць. Жорстке посилання не є окремою копією файла, це той же файл, який відображається у двох різних місцях.

|  |  |
| --- | --- |
|  | *Переміщення або зміна імені початкового файла не впливає на працездатність жорсткого посилання. Якщо видаляють один з цих файлів (точніше, одне з цих імен), то файл ще зберігається на диску (поки в нього є хоча б одне ім’я-посилання).* |

Розглянемо деякі властивості жорстких посилань:

* працюють тільки в межах однієї файлової системи (причина проста: лічильник посилання зберігається у самому [inode](https://linuxguide.rozh2sch.org.ua/" \l "inode), а останній не може спільно використовуватися у різних файлових системах);
* не можна посилатися на каталоги, лише на файли;
* мають ту ж інформацію [inode](https://linuxguide.rozh2sch.org.ua/" \l "inode) і набір дозволів, що і у початкового файла;
* дозволи на посилання змінюються при зміні дозволів файла;
* можна переміщувати та перейменовувати і, навіть, видаляти файл без шкоди посиланням.

Відмінності між символьними і жорсткими посиланнями

Відмінність жорстких посилань від символьних полягає в тому, що у випадку символьного посилання, якщо початковий файл видалений, то символьне посилання перестає працювати. Символьне посилання буде як і раніше існувати, але воно буде порожнім, і спроба доступу до нього призведе до помилки, оскільки початковий файл або каталог був видалений.

Якщо створити жорстке посилання на файл, а потім видалити оригінал, файл все ще буде існувати у місці розташування посилання.

Для створення символьних посилань існує утиліта [ln](https://linuxguide.rozh2sch.org.ua/" \l "ln), яка має наступний синтаксис

ln опції файл\_початковий файл\_посилання

Опції можуть мати такі значення:

-s

створення символьного посилання;

-r

створення символьного посилання з відносним шляхом до файла;

-P

створення жорсткого посилання;

-f

видалення існуючих посилань;

-i

запитати чи видаляти вже існуючі посилання з таким ім’ям.

Для початку, створимо каталог testlinks і перейдемо у нього

teacher@teacher-VirtualBox:~/terminal\_work$ mkdir testlinks && cd testlinks

Тепер створимо початковий файл з ім’ям source, який буде містити текст symbolic links, і переглянемо вміст цього файла, використовуючи команду cat

teacher@teacher-VirtualBox:~/terminal\_work/testlinks$ echo "symbolic links" > source

teacher@teacher-VirtualBox:~/terminal\_work/testlinks$ cat source

symbolic links

Початковий файл готовий. Далі створимо символьне посилання на файл source за допомогою команди ln з опцією -s і переглянемо вміст файла за цим символьним посиланням командою cat

teacher@teacher-VirtualBox:~/terminal\_work/testlinks$ ln -s source softlink

teacher@teacher-VirtualBox:~/terminal\_work/testlinks$ cat softlink

symbolic links

Як бачимо, вміст у файлі-посиланні і початковому (оригінальному) файлі однаковий.

|  |  |
| --- | --- |
|  | *При створенні символьних посилань, імена файлів можуть бути як абсолютними, так і відносними.* |

Але використовуючи утиліту ls, можна побачити, що вказані файли мають відмінності

teacher@teacher-VirtualBox:~/terminal\_work/testlinks$ ls -li

загалом 4

1700690 lrwxrwxrwx 1 teacher teacher 6 тра 27 15:44 softlink -> source

1700689 -rw-r--r-- 1 teacher teacher 15 тра 27 15:40 source

Незважаючи на те, що вміст однаковий, тут ми бачимо, що адреса inode (у першій колонці) і права доступу до файлів (у другій колонці) відрізняються, крім того, явно показано, що softlink - це символьне посилання Linux на початковий файл source (можна легко ідентифікувати символьне посилання по маленькій літері l на початку рядка з правами доступу).

Тепер видалимо початковий файл source командою rm і переглянемо вміст файла-посилання softlink

teacher@teacher-VirtualBox:~/terminal\_work/testlinks$ rm source

teacher@teacher-VirtualBox:~/terminal\_work/testlinks$ cat softlink

cat: softlink: Немає такого файла або каталогу # символьне посилання не працює, оскільки початковий файл був видалений

teacher@teacher-VirtualBox:~/terminal\_work/testlinks$ ls

softlink # символьне посилання залишилося після видалення оригінального файла

Отримана помилка, що такого файла не існує, тому що, дійсно, початковий файл був видалений. Якщо ж видалити файл-посилання, то початковий файл залишиться на місці.

Розглянемо особливості процесу створення жорстких посилань. Знову створимо початковий файл source, але вже з текстом hard links

teacher@teacher-VirtualBox:~/terminal\_work/testlinks$ echo "hard links" > source

teacher@teacher-VirtualBox:~/terminal\_work/testlinks$ cat source

hard links

Створимо жорстке посилання hardlink на файл source, викликавши утиліту ln без параметрів, і переглянемо вміст файла-посилання hardlink

teacher@teacher-VirtualBox:~/terminal\_work/testlinks$ ln source hardlink

teacher@teacher-VirtualBox:~/terminal\_work/testlinks$ cat hardlink

hard links

Вміст файлів (початкового і жорсткого посилання) однаковий. Якщо переглянути інформацію про ці файли за допомогою утиліти ls, то побачимо, що inode і права доступу також співпадають

teacher@teacher-VirtualBox:~/terminal\_work/testlinks$ ls -li

загалом 8

1700689 -rw-r--r-- 2 teacher teacher 11 тра 27 16:04 hardlink

1700689 -rw-r--r-- 2 teacher teacher 11 тра 27 16:04 source

Зверніть увагу на номер в третій колонці - цифру 2, яка означає, що є **два екземпляри початкового файла**. Тобто, початковий файл source, так і жорстке посилання на нього hardlink, мають номер 2 у списку, тому що кожен з них є посиланням чи екземпляром того ж файла. Якщо додати нове жорстке посилання на оригінальний файл, тоді цифра збільшиться ще на 1 пункт, і буде показувати цифру 3.

|  |  |
| --- | --- |
|  | *Файл залишиться на диску, поки є жорстке посилання на нього. Коли лічильник екземплярів скидається до 0, файл видаляється. Тобто, якщо створити, наприклад, 5 жорстких посилань, то поки не буде видалено усі ці посилання, файл буде існувати.* |

Видалимо початковий файл source і переглянемо вміст файла-посилання hardlink

teacher@teacher-VirtualBox:~/terminal\_work/testlinks$ rm source

teacher@teacher-VirtualBox:~/terminal\_work/testlinks$ cat hardlink

hard links

Як бачимо, нічого не сталося і жорстке посилання, як і раніше, вказує на потрібну ділянку диска. Це головна відмінність жорсткого посилання від символьного.