chuzhoy007.ru /komanda-cp-v-linux-kopirovanie-faylov-i-katalogov-polnoe-rukovodstvo

Команда cp в Linux, копирование файлов и каталогов полное руководство

Команда cp в Linux (сокращение от «copy») является мощным инструментом для копирования файлов и директорий. Это одна из наиболее часто используемых команд. Она позволяет пользователям создавать точные копии файлов и каталогов, а также выполнять ряд дополнительных операций, таких как сохранение атрибутов файлов, работа с разреженными файлами и многое другое.

Синтаксис и основные опции

Прежде чем начать, важно понимать синтаксис и доступные опции утилиты. В этом разделе мы подробно рассмотрим, как правильно формировать команду и какие основные опции используются для управления копированием. Это позволит максимально эффективно использовать «cp» и адаптировать ее под свои потребности при работе в командной строке Linux.

```
1 cp [опции] [источник] [назначение]
```

- **[опции]** — флаги, предоставляющие дополнительные настройки.
- **[источник]** — путь к файлу или директории, который вы хотите скопировать.

- **[назначение]** — путь, по которому будет создана копия источника.

Опции

- **-a, —archive** — копирование файлов и директорий в архивном режиме, сохраняя все атрибуты и рекурсивно обрабатывая поддиректории.
- **—attributes-only** — копирование только атрибутов файлов, без копирования данных файлов.
- **—backup[=КОГДА]** — создание резервной копии каждого существующего файла. Параметр «КОГДА» опционален и определяет, когда создавать резервные копии.
- **-b** — Аналогично опции «—backup», но не принимает аргумента. Создает резервные копии каждого существующего файла.
- **—copy-contents** — при рекурсивном копировании также копирует содержимое специальных файлов.
- **-d** — копирует каталоги, включая символические и жёсткие ссылки, сохраняя их как ссылки. Аналогично параметрам «—no-dereference» и «—preserve=links».
- **-f, —force** — принудительное копирование, даже если это перезапишет файлы.
- **-i, —interactive** — интерактивный режим, который спрашивает пользователя о перезаписи файлов.
- **-H** — следовать по символьным ссылкам в исходном файле и копировать файлы, на которые они указывают, вместо самих ссылок.
- **-l, —link** — создавать жёсткие ссылки на файлы вместо их копирования.
- **-L, —dereference** — всегда следовать символьным ссылкам в источнике, копировать содержимое, на которое они указывают, вместо самих символьных ссылок.
- **-n, —no-clobber** — не перезаписывать существующие файлы при копировании, иначе говоря, пропускать файлы, если файл с таким именем уже существует в целевом каталоге (отменяет ранее указанный ключ -i).
- **-P, —no-dereference** — не следовать по символьным ссылкам в источнике. Эта опция предотвращает разыменование символьных ссылок при копировании файлов и каталогов.
- **—preserve[=СПИС_АТТР]** — сохранять указанные атрибуты файлов при копировании. По умолчанию, это включает атрибуты `mode`, `ownership` и `timestamps`. Опция позволяет указать дополнительные атрибуты, такие как `context`, `links`, `xattr`, и `all`, для сохранения при копировании.
- **-p** — это сокращение от `--preserve=mode,ownership,timestamps`, что означает сохранение атрибутов файла, таких как права доступа (`mode`), владелец (`ownership`) и временные метки (`timestamps`), при копировании. По сути то же, что и предыдущий параметр.
- **—no-preserve=СПИС_АТТР** — это опция, которая используется для указания, какие атрибуты файла не должны сохраняться при копировании с помощью команды `cp`. Вы можете перечислить атрибуты, которые необходимо исключить из сохранения, вместо `СПИС_АТТР`. Например, если вы хотите исключить сохранение атрибута владельца (`ownership`) при копировании, вы можете использовать `--no-preserve=ownership`.

- **—parents** — включает исходный путь в каталог назначения, позволяя сохранить структуру каталогов исходного файла при копировании. Это полезно при необходимости создания копии файла внутри определенного каталога, сохранив при этом его исходное расположение.
- **-R, -r, —recursive** — используется для рекурсивного копирования каталогов, включая все их содержимое и подкаталоги.
- —reflink[=КОГДА] — Определяет, когда использовать облегченное копирование (clone/CoW). Когда указан, блоки данных копируются только при изменении. Аргумент КОГДА определяет, в каких случаях применять облегченное копирование (всегда, автоматически или никогда).
- **—remove-destination** — даляет каждый файл назначения перед попыткой его открыть. Эта опция позволяет избежать ошибок при копировании файлов, которые могут находиться в папке-назначении до выполнения операции копирования.
- **—sparse=КОГДА** — Контролирует создание разреженных файлов во время копирования. Параметр «КОГДА» определяет, в каких случаях создаются разреженные файлы; `auto` (по умолчанию): Разреженные файлы создаются при обнаружении длинных последовательностей нулевых байтов в исходном файле. `always`: Всегда создаются разреженные файлы, независимо от содержания исходного файла. `never`: Запрещает создание разреженных файлов при копировании. Это позволяет управлять использованием разреженных файлов и экономить место на диске в зависимости от потребностей.
- **—strip-trailing-slashes** — удалить все конечные косые черты (слэши) из каждого аргумента-источника. Это полезно, например, когда вы копируете содержимое директории и не хотите включать конечные слэши в имена файлов или папок назначения.
- **-s, —symbolic-link** — предписывает создавать символические ссылки вместо копирования файлов или директорий.
- **-S, —suffix=СУФФИКС** — Эта опция позволяет вам задать суффикс для резервных копий файлов при использовании опции `--backup`. Например, если вы установите суффикс как `.bak`, резервные копии будут иметь имена вида `filename.bak`.
- **—target-directory=КАТ** — С помощью этой опции вы можете указать каталог, в который нужно скопировать все исходные файлы и директории.
- **-T, —no-target-directory** — Эта опция указывает, что целью для копирования считать обычным файлом, а не каталогом.
- **-u, —update** — При использовании этой опции, файл будет скопирован только в том случае, если исходный файл новее, чем файл назначения, или если файл назначения отсутствует.
- **-v, —verbose** — При использовании этой опции будет выводиться подробная информация о процессе копирования, поясняя, что именно копируется.
- **-x, —one-file-system** — Оставаться в пределах одной файловой системы при копировании, игнорируя монтированные файловые системы, если они присутствуют в исходном пути.
- **-Z** — Установить контекст безопасности SELinux файла назначения равным типу по умолчанию. Эта опция используется для управления контекстом безопасности SELinux в целевых файлах.
- **—context[=CTX]** — Подобно опции `-Z`, позволяет установить контекст безопасности SELinux для файла назначения. Если указан аргумент CTX, то контекст будет установлен равным

этому значению; в противном случае, контекст будет установлен равным типу по умолчанию.

- **—version** — Выводит информацию о версии cp и завершает выполнение.

Основное использование

В этом разделе мы рассмотрим основные сценарии использования команды «cp». Здесь вы найдете обзор основных операций, которые можно выполнять с помощью этой команды, включая копирование файлов и каталогов, а также передачу данных между разными местами.

Как сделать копию файла в текущем каталоге

Для создания копии файла в текущем каталоге используйте команду «cp» с указанием имени исходного файла и желаемого имени для копии. Вот синтаксис этой операции:

```
1 cp имя_исходного_файла имя_копии
```

Где:

- имя_исходного_файла — это имя файла, который вы хотите скопировать и который находится в текущем каталоге.
- имя_копии — это имя, которое вы хотите присвоить создаваемой копии файла.

  Чтобы узнать имя текущей папки используйте команду pwd.

Например, если у вас есть файл с именем «document.txt» в текущем каталоге, и вы хотите создать его копию с именем «document_copy.txt», выполните следующую команду:

```
1 $ cp document.txt document_copy.txt
```

Читайте также: Настройка Debian после установки

После выполнения этой команды в текущем каталоге будет создана копия файла «document.txt» с именем «document_copy.txt». Для того, чтобы увидеть содержимое папки и убедиться, что новый файл появился используйте команду ls.

```
chuzhoy007_ru@linux:~$ cp document.txt document_copy.txt
chuzhoy007_ru@linux:~$ ls
 document_copy.txt   file2.txt   ventoy   Документы   Музыка          Шаблоны
 document.txt        Folder      Видео    Загрузки    Общедоступные
 file1.txt           Proekti     Год      Изображения 'Рабочий стол'
chuzhoy007_ru@linux:~$
```

Копирование одного файла в другой

Для копирования одного файла в другой, вы должны указать путь к исходному файлу и путь к файлу назначения. Узнать Вот синтаксис для этой операции:

```
1 cp исходный_файл файл_назначения
```

Здесь;

- **исходный_файл** — это путь к файлу, который вы хотите скопировать.
- **файл_назначения** — это путь и имя файла, в который вы хотите скопировать исходный файл.

Например, если у вас есть файл с именем «file1.txt», и вы хотите создать копию этого файла с именем «file2.txt», вы можете сделать следующее:

```
1 cp file1.txt file2.txt
```

После выполнения этой команды содержимое «file1.txt» будет скопировано в «file2.txt». Его содержимое будет перезаписано копией из «file1.txt».

> Учтите, что при таком копировании информация из file2.txt будет утеряна так как её заменит информация из file1.txt.

Копирование файла с изменением имени

Для копирования файла с изменением его имени укажите путь к исходному файлу и новому имени файла. Синтаксис операции выглядит следующим образом:

```
1 cp исходный_файл новое_имя_файла
```

Где:

- **исходный_файл** — это путь к файлу, который вы хотите скопировать.
- **новое_имя_файла** — это имя, которое вы хотите присвоить скопированному файлу.

Пример:

Если у вас есть файл «file1.txt», и вы хотите создать копию этого файла с новым именем «file1_copy.txt», выполните команду:

```
1 $ cp file1.txt file1_copy.txt
```

После выполнения этой команды будет создан файл «file1_copy.txt», который является копией «file1.txt», но имеет новое имя.

```
chuzhoy007_ru@linux:~$ cp file1.txt file1_copy.txt
chuzhoy007_ru@linux:~$ ls
 file1_copy.txt   file3.txt   ventoy   Документы     Музыка           Шаблоны
 file1.txt        Folder      Видео    Загрузки     Общедоступные
 file2.txt        Proekti     Год      Изображения  'Рабочий стол'
chuzhoy007_ru@linux:~$
```

Как копировать файл в другую папку

Для выполнения копирования файла в другую папку, укажите исходный файл, который нужно скопировать, и путь к папке назначения. Синтаксис этой операции выглядит следующим образом:

```
1 cp исходный_файл путь/к/папке/назначения/
```

- **исходный_файл** — это путь к файлу, который вы хотите скопировать.
- **путь/к/папке/назначения/** — это путь к папке, в которую вы хотите скопировать файл. Завершающий слэш (/) после имени папки указывает, что это каталог назначения.

Например, если у вас есть файл с именем «file1.txt», и вы хотите скопировать его в папку «Folder/test», выполните следующую команду:

```
1 $ cp file1.txt Folder/test/
```

Эта команда скопирует файл «file1.txt» в папку «Folder/test».

```
chuzhoy007_ru@linux:~$ cp file1.txt Folder/test/
chuzhoy007_ru@linux:~$ ls Folder/test/
file1.txt
chuzhoy007_ru@linux:~$
```

Копирование нескольких файлов

Для копирования нескольких файлов в другую папку, укажите пути к исходным файлам, которые вы хотите скопировать, и путь к папке назначения. Вы можете перечислить несколько исходных файлов, разделяя их пробелами. Синтаксис этой операции выглядит следующим образом:

```
1 cp исходный_файл1 исходный_файл2 ... исходный_файлN путь_к_папке_назначения/
```

Где:

- **исходный_файл1, исходный_файл2, …, исходный_файлN** — это пути к файлам либо файлы в текущей папке, которые вы хотите скопировать. Вы можете указать сколько угодно файлов.
- **путь_к_папке_назначения/** — это путь к папке, в которую вы хотите скопировать файлы. Завершающий слэш (/) после имени папки указывает, что это каталог назначения.

Пример:
Если у вас есть файлы «file1.txt», «file2.txt» и «file3.txt», и вы хотите скопировать их в папку «Folder», выполните следующую команду:

```
1 $ cp file1.txt file2.txt file3.txt Folder/
```

Эта команда скопирует файлы «file1.txt», «file2.txt» и «file3.txt» в папку «Folder».

```
chuzhoy007_ru@linux:~$ cp file1.txt file2.txt file3.txt Folder/
chuzhoy007_ru@linux:~$ ls Folder/
file1.txt  file2.txt  file3.txt  folder1  test  Vasiliy
chuzhoy007_ru@linux:~$
```

Копирование каталога и его содержимого. Опция -r

Для копирования каталога и всех файлов внутри него в системе Linux, используйте команду `cp` с опцией `-r` (или `-R`), чтобы скопировать содержимое каталога рекурсивно. Синтаксис операции выглядит следующим образом:

```
1 cp -r [исходный_каталог] [каталог_назначения]
```

Где:

- **[исходный_каталог]** — это путь к каталогу, который вы хотите скопировать.
- **[каталог_назначения]** — это путь к каталогу, в который вы хотите скопировать содержимое исходного каталога.

Пример:

Допустим, у вас есть каталог с именем «my_directory», и вы хотите создать копию этого каталога с именем «my_directory_copy». Выполните команду:

```
1 $ cp -r my_directory my_directory_copy
```

После выполнения этой команды будет создан каталог «my_directory_copy», включая все файлы и подкаталоги, содержащиеся в «my_directory».

```
chuzhoy007_ru@linux:~$ ls -R my_directory/
my_directory/:
folder

my_directory/folder:
test.txt

my_directory/folder/test.txt:
chuzhoy007_ru@linux:~$ cp -r my_directory my_directory_copy
chuzhoy007_ru@linux:~$ ls
 file1.txt   my_directory        Видео       Изображения     Шаблоны
 file2.txt   my_directory_copy   Год         Музыка
 file3.txt   Proekti             Документы   Общедоступные
 Folder      ventoy              Загрузки    'Рабочий стол'
chuzhoy007_ru@linux:~$ ls -R my_directory_copy/
my_directory_copy/:
folder

my_directory_copy/folder:
test.txt

my_directory_copy/folder/test.txt:
chuzhoy007_ru@linux:~$
```

> Обратите внимание, что опция `-r` означает «рекурсивно», что позволяет скопировать все файлы и подкаталоги внутри исходного каталога.

Сохранение метаданных (например, прав доступа и времени) при копировании файла. Опция -p

Для сохранения метаданных, таких как прав доступа и времена, при копировании файлов в Linux с помощью команды cp, вы можете использовать опцию «-p» или «—preserve=mode,ownership,timestamps.» Эта опция учитывает следующие метаданные:

- **mode** — права доступа к файлу.
- **ownership** — владелец и группа файла.
- **timestamps** — времена доступа, модификации и изменения файла.

Пример использования опции «-p»:

```
1 $ cp -p file1.txt file2.txt
```

Это скопирует файл «file1.txt» в «file2.txt», сохраняя все метаданные.

Как копировать каталог с сохранением метаданных. Опция -a

Если вам нужно скопировать каталог с его содержимым и структурой, также сохраняя метаданные, используйте опцию «-a» или «—archive». Эта опция эквивалентна «-dPR —preserve=all» и обеспечивает сохранение всех метаданных при копировании каталога:

Пример:

```
1 $ cp -a my_directory my_directory_copy/
```

Таким образом, вы можете копировать файлы и каталоги, сохраняя их метаданные.

Вывод информации о процессе копирования. Опция -v

Для вывода информации о процессе копирования при использовании команды cp, вы можете указать опцию «-v» или «—verbose». Это позволит команде cp выводить подробную информацию о каждом скопированном файле.

Пример:

```
1 $ cp -v file1.txt file2.txt
```

После выполнения этой команды, вы увидите информацию о процессе копирования, включая имена скопированных файлов. Это помогает отслеживать, какие файлы были скопированы, и убедиться, что операция прошла успешно.

```
chuzhoy007_ru@linux:~$ cp -v file1.txt file2.txt
'file1.txt' -> 'file2.txt'
chuzhoy007_ru@linux:~$
```

Интерактивное копирование с запросом подтверждения при замене файлов. Опция -i

Для выполнения интерактивного копирования с запросом подтверждения при замене файлов, вы можете использовать опцию «-i» или «—interactive». Эта опция заставляет команду cp спрашивать у вас подтверждение, прежде чем заменить существующие файлы в процессе копирования.

Пример:

```
1 cp -i file1.txt file2.txt
```

При выполнении этой команды, если файл file2.txt уже существует, команда cp запросит у вас подтверждение, прежде чем заменить его. Вы можете ввести y для подтверждения или n для отмены операции.

```
chuzhoy007_ru@linux:~$ cp -i file1.txt file2.txt
cp: переписать 'file2.txt'? y
chuzhoy007_ru@linux:~$
```

Это полезная опция, которая позволяет избежать случайного замещения файлов и предоставляет

вам контроль над процессом копирования.

Копирование файлов с сохранением символических ссылок. Опция -d

Для копирования файлов с сохранением символических ссылок при использовании «cp», можно применить опцию «-d» или «—no-dereference». Эта опция гарантирует, что символические ссылки будут скопированы как сами ссылки, а не их целевые файлы.

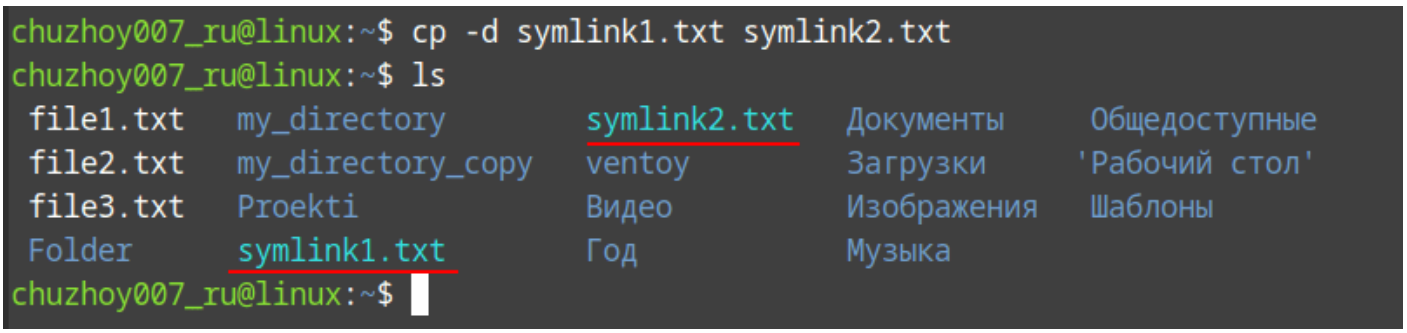
Читайте также: Команда split в Linux: как разбить файл на части и объединить их обратно

Пример использования опции «-d»:

```
1 $ cp -d symlink1.txt symlink2.txt
```

В этом примере «symlink1.txt» является символической ссылкой на файл, и с помощью опции «-d», команда cp скопирует символическую ссылку «symlink1.txt» как ссылку «symlink2.txt».

Это полезно, если вы хотите сохранить структуру символических ссылок в процессе копирования файлов и директорий.



Пропускать существующие файлы при копировании. Опция -n

Для пропуска существующих файлов при копировании, применяется опция «-n» или «—no-clobber». Она предотвращает перезапись существующих файлов в целевой директории.

Пример использования опции «-n»:

```
1 $ cp -n file1.txt file2.txt
```

Если file2.txt уже существует, то команда cp не будет перезаписывать его и оставит без изменений.

Это полезно, если вы хотите избежать случайной перезаписи существующих файлов и сохранить их целостность при копировании.

Копирование только обновленных файлов. Опция -u

Для копирования только обновленных файлов, используйте опцию «-u» или «—update». С её помощью файл копируется только в том случае, если исходный файл новее, чем файл назначения, или если файл назначения отсутствует.

Пример:

```
1 $ cp -u file1.txt file2.txt
```

Если «file2.txt» уже существует и он старше, чем «file1.txt», то копирование выполняться не будет, оставив файл «file2.txt» без изменений. Только в случае, если «file1.txt» новее или «file2.txt» отсутствует, будет выполнено копирование.

Это удобно, если вы хотите скопировать только те файлы, которые были изменены или добавлены с момента последней операции копирования.

Создание резервной копии существующего целевого файла при копировании. Опция -b

Когда вы копируете файл с помощью команды cp, можно создать резервную копию существующего целевого файла. Это может быть полезно, чтобы сохранить предыдущую версию перед его заменой. Для этого используется опция «—backup» или «-b».

Предположим, есть файл «existing_file.txt», и вы хотите создать его копию с именем «backup_existing_file.txt», сохраняя при этом резервную копию существующего файла, если он уже есть в целевой директории.

```
1 cp --backup=numbered existing_file.txt Proekti/backup_existing_file.txt
```

Теперь, если файл «backup_existing_file.txt» уже существует, он будет переименован, добавив суффикс, например, «.~1~», и затем создан новый файл «backup_existing_file.txt», копируя содержимое «existing_file.txt». Таким образом, предыдущая версия файла сохраняется в виде резервной копии.

```
chuzhoy007_ru@linux:~$ ls
 existing_file.txt   my_directory        ventoy      Изображения
 file1.txt           my_directory_copy   Видео       Музыка
 file2.txt           Proekti             Год         Общедоступные
 file3.txt           symlink1.txt        Документы  'Рабочий стол'
 Folder              symlink2.txt        Загрузки    Шаблоны
chuzhoy007_ru@linux:~$ cp --backup=numbered existing_file.txt Proekti/backup_exi
sting_file.txt
chuzhoy007_ru@linux:~$ ls Proekti/
backup_existing_file.txt  backup_existing_file.txt.~1~  Vasiliy
chuzhoy007_ru@linux:~$
```

Вы можете настроить формат суффикса для резервных копий с помощью переменной окружения `VERSION_CONTROL`. Возможные значения включают «numbered», «existing», «simple» и «none».

- **«numbered»** — Создаются нумерованные резервные копии (например, `backup_existing_file.txt.~1~`, `backup_existing_file.txt.~2~`, и так далее).
- **«existing»** — Если существуют нумерованные резервные копии, они будут переиспользованы, иначе создаются простые копии.

- **«simple»** — Создаются простые резервные копии (например, `backup_existing_file.txt~`).
- **«none»** — Резервные копии не создаются.

Вы можете установить значение `VERSION_CONTROL` так:

```
1 $ export VERSION_CONTROL=numbered
```

Как копировать файлы принудительно. Опция -f

Параметр `-f` используется для обозначения «принудительного» копирования. Его основное назначение заключается в том, чтобы перезаписать целевой файл (файл назначения), даже если он уже существует, без запроса на подтверждение или предупреждения.

Для копирования файлов и каталогов принудительно с использованием параметра «-f, вы можете выполнить команду:

```
1 cp -f исходный_файл/каталог файл_назначения
```

Здесь:

- **-f** — Этот параметр обозначает «принудительное» копирование. Если файл назначения с таким именем уже существует, он будет перезаписан без предупреждения.
- **исходный_файл/каталог** — Это путь к исходному файлу или каталогу, который вы хотите скопировать.
- **файл_назначения** — Это путь и имя файла или каталога, в который вы хотите скопировать исходный файл или каталог.

Например, чтобы скопировать файл «file1.txt» в каталог «destination» и принудительно перезаписать его, если файл назначения уже существует, выполните команду:

```
1 $ cp -f file1.txt destination/
```

Эта команда скопирует «file1.txt» в каталог «destination», и, если в «destination» уже существует файл с таким именем, он будет перезаписан без запроса подтверждения.

Вот некоторые основные случаи, когда параметр «-f» полезен:

1. **Принудительное перезаписывание**: Когда файл с таким же именем уже существует в целевой директории, без параметра `-f` команда `cp` обычно запрашивает подтверждение от пользователя, прежде чем перезаписать файл. Параметр `-f` позволяет выполнить перезапись без запроса.
2. **Автоматизированные задачи**: В автоматизированных сценариях, где вам не нужно вмешательство пользователя, параметр `-f` позволяет предотвратить появление запросов на подтверждение и гарантировать, что операция копирования будет завершена.
3. **Безопасное копирование при сценариях перезаписи**: Если вам нужно автоматически заменить целевой файл без возможности его случайного сохранения или копии, параметр `-f`

гарантирует, что старый файл будет заменен новым без сохранения.

Применение параметра «-f» требует внимательности, так как он может привести к потере данных, если неосторожно перезапишет файлы.

Работа с разреженными файлами. Опция —spare

Опциия «—sparse» используется для создания разреженных (sparse) копий файлов. Разреженные файлы — это файлы, которые содержат большое количество нулевых байтов, но они хранятся таким образом, что не занимают реального дискового пространства для каждого из нулевых байтов.

Это полезно использовать для экономии дискового пространства при копировании файлов, содержащих множество нулевых байтов, например, образов виртуальных машин.

Для примера создаем файл «virtual_disk.vdi» размером 5 Гб.

```
1 $ dd if=/dev/zero of=virtual_disk.vdi bs=1M count=5000
```

Теперь создаем разреженную копию этого файла:

```
1 $ cp --sparse=always virtual_disk.vdi virtual_disk_copy.vdi
```

Эта команда создаст разреженную копию файла «virtual_disk.vdi» с именем «virtual_disk_copy.vdi».

Чтобы проверить, что разреженная копия занимает меньше места на диске, используйте команду du для измерения размера файлов:

```
1 $ du -h virtual_disk.vdi
2 $ du -h virtual_disk_copy.vdi
```

Вы увидите, что размер «virtual_disk_copy.vdi» меньше, чем «virtual_disk.vdi» в моем примере это 0.

Когда вы создаете разреженную копию файла, она сохраняет нулевые байты как ссылки на одну и ту же область в памяти, в то время как обычная копия создает реальные нулевые байты, что может занять больше места на диске.

```
chuzhoy007_ru@linux:~$ cp --sparse=always virtual_disk.vdi virtual_disk_copy.vdi
chuzhoy007_ru@linux:~$ du -h virtual_disk.vdi
4,9G    virtual_disk.vdi
chuzhoy007_ru@linux:~$ du -h virtual_disk_copy.vdi
0       virtual_disk_copy.vdi
chuzhoy007_ru@linux:~$
```

Получение дополнительной информации

Чтобы получить дополнительную информацию изучите официальные короткую справку:

```
1 $ cp --help
```

и man страницу:

```
1 $ man cp
```