



[Главная](#) >> [Терминал](#) >> Как убить процесс Linux

Как убить процесс Linux

Обновлено: 01 октября 2021 Опубликовано: 4 марта, 2017 от [admin](#), 4 комментариев, время чтения: 7 минут

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

Несмотря на то что Linux стабильнее чем Windows, в плане работы программы и различных сервисов, случается всякое и иногда возникает необходимость завершить процесс Linux. Это может понадобиться, если программа зависла, когда вы запустили системный сервис в фоне через терминал, а не систему инициализации, а также во многих других случаях, когда убить процесс Linux проще, чем перезагружать всю систему.

В этой статье мы рассмотрим несколько самых распространенных способов завершить процесс Linux. Опишем подробно как происходит остановка процесса и как все сделать правильно.

Содержание статьи

- [Как происходит завершение процесса?](#)
- [Как убить процесс Linux?](#)
- [Как завершить процесс с помощью pkill](#)
- [Как остановить процесс с помощью killall](#)
- [Выводы](#)

Конфиденциальность - Условия использования

Privacy

Как происходит завершение процесса?

Управление процессами в операционной системе Linux осуществляется с помощью сигналов. В том числе и завершение любого процесса. Сигналы передает система, но также их может передавать пользователь с помощью специальных команд или даже сочетаний клавиш в терминале. Когда процессу приходит сигнал о необходимости завершиться, он должен выполнить некоторые подготовительные действия.

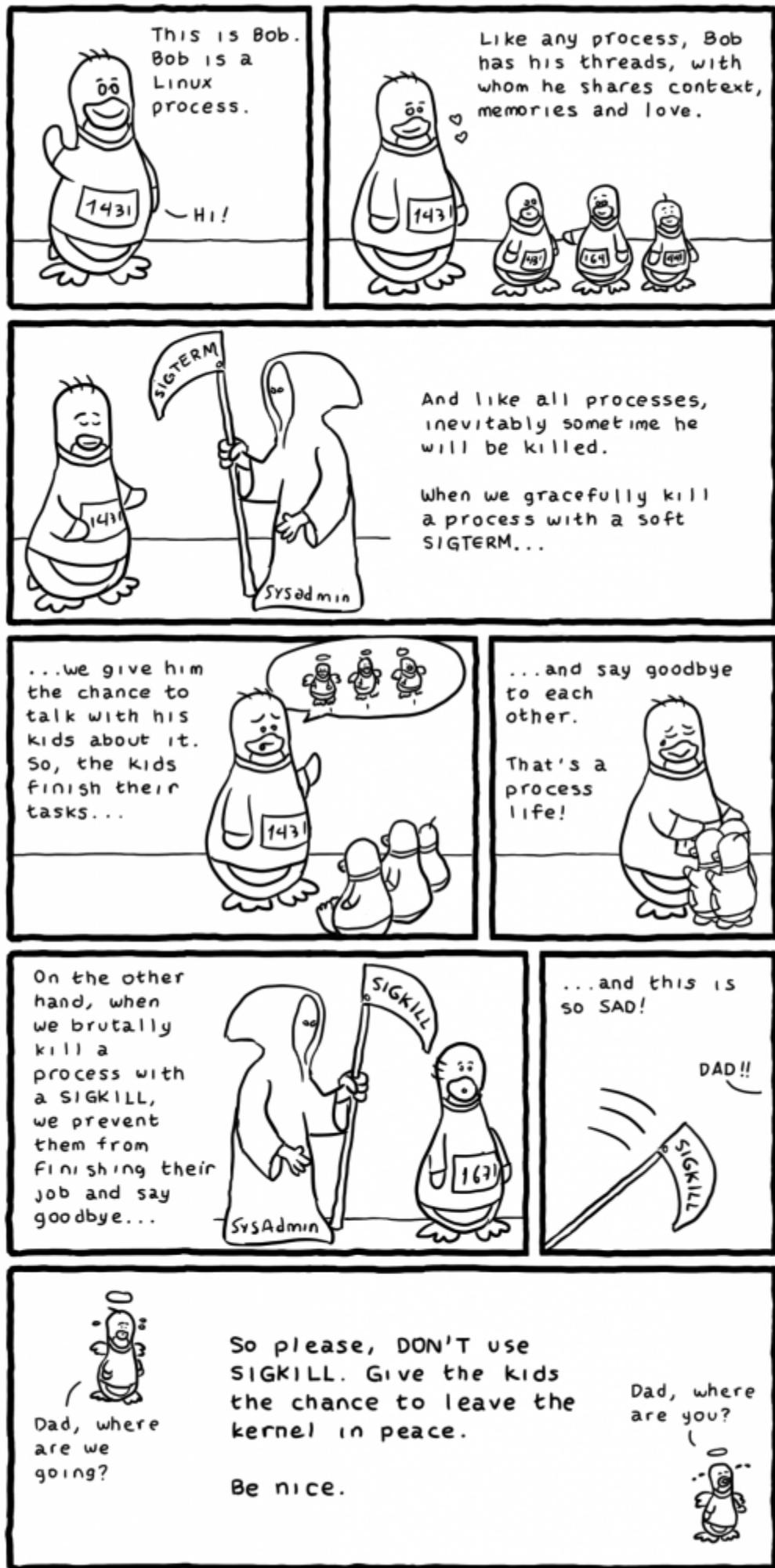
Необходимо завершить дочерние процессы, удалить временные файлы, сокеты и так далее. Но в зависимости от сложности ситуации процесс может реагировать не на все сигналы. Рассмотрим основные сигналы, которые используются для завершения процесса:

- **SIGINT** - самый безобидный сигнал завершения, означает Interrupt. Он отправляется процессу, запущенному из терминала с помощью сочетания клавиш **Ctrl+C**. Процесс правильно завершает все свои действия и возвращает управление;
- **SIGQUIT** - это еще один сигнал, который отправляется с помощью сочетания клавиш, программы, запущенной в терминале. Он сообщает ей что нужно завершиться и программа может выполнить корректное завершение или проигнорировать сигнал. В отличие от предыдущего, она генерирует дамп памяти. Сочетание клавиш **Ctrl+%**;
- **SIGHUP** - сообщает процессу, что соединение с управляющим терминалом разорвано, отправляется, в основном, системой при разрыве соединения с интернетом;
- **SIGTERM** - немедленно завершает процесс, но обрабатывается программой, поэтому позволяет ей завершить дочерние процессы и освободить все ресурсы;
- **SIGKILL** - тоже немедленно завершает процесс, но, в отличие от предыдущего варианта, он не передается самому процессу, а обрабатывается ядром. Поэтому ресурсы и дочерние процессы остаются запущенными.

Важно понимать, что нужно дать процессу возможность завершиться корректно. Желательно, чтобы порты и сокеты были освобождены, закрыты и удалены временные файлы. Поэтому никогда не передавайте сразу SIGKILL. Передавайте сигналы завершения в последовательности, как они перечислены выше.

Сначала **Ctrl+C**, если это возможно, затем SIGTERM - он хоть и завершает процесс, но делает это культурно, и только в крайнем случае SIGKILL. А теперь рассмотрим как убить процесс по pid Linux на практике. Если вы всегда используете SIGKILL, тогда на ум приходит такая картинка:





Как убить процесс Linux?

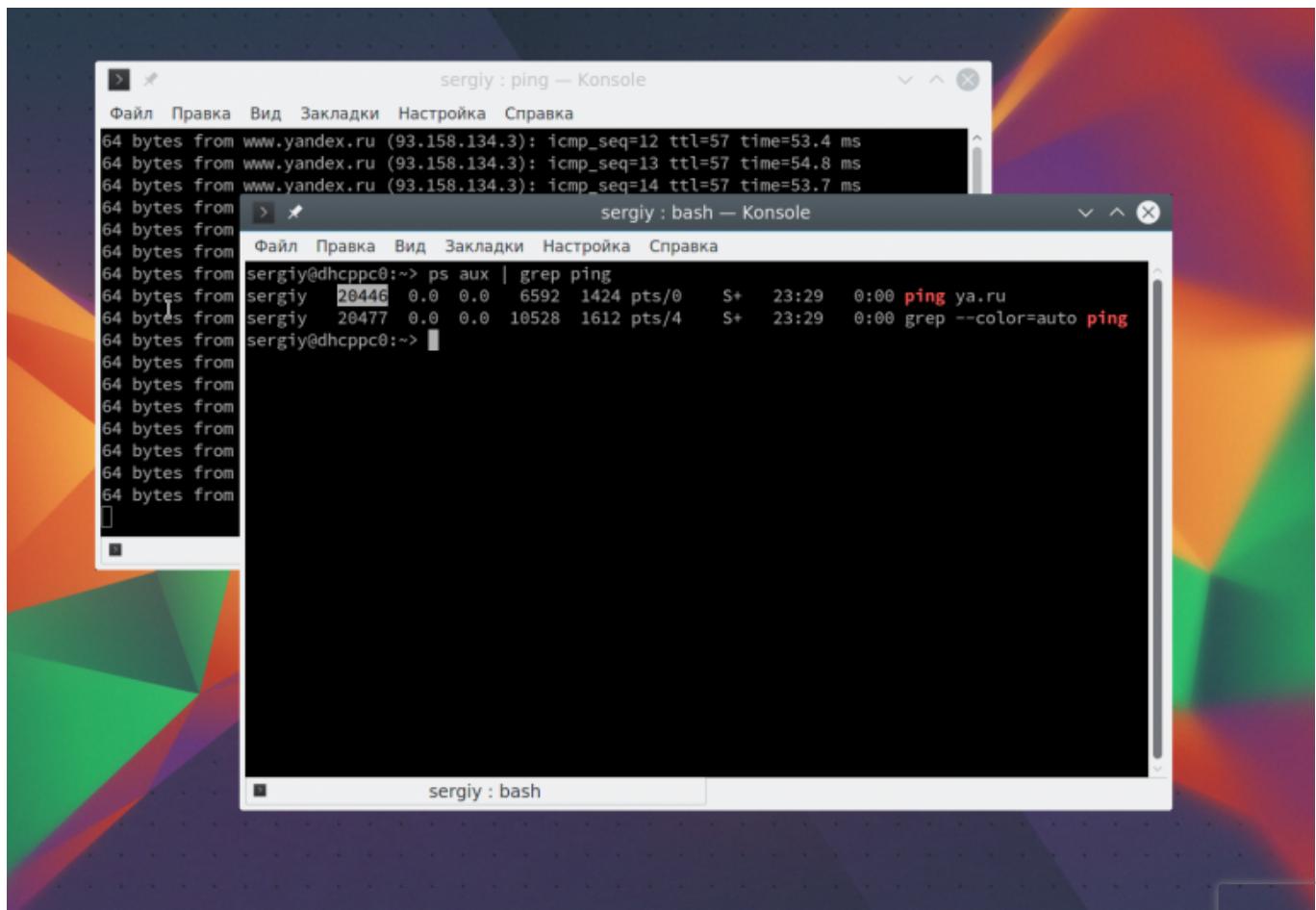
Для передачи сигналов процессам в Linux используется утилита `kill`. Ее синтаксис очень прост:

```
$ kill -сигнал pid_процесса
```

Сигнал представляет собой один из выше перечисленных сигналов для завершения процесса. По умолчанию, если этот параметр не указан, используется сигнал SIGTERM, что является очень правильно. Также нам нужно указать какой процесс нужно завершить. Для этого используется уникальный идентификатор процесса - PID.

Допустим, у нас выполняется утилита `ping`. Мы хотим ее завершить с помощью `kill`. Тогда, сначала мы узнаем ее идентификатор с помощью команды `ps`:

```
$ ps aux | grep ping
```



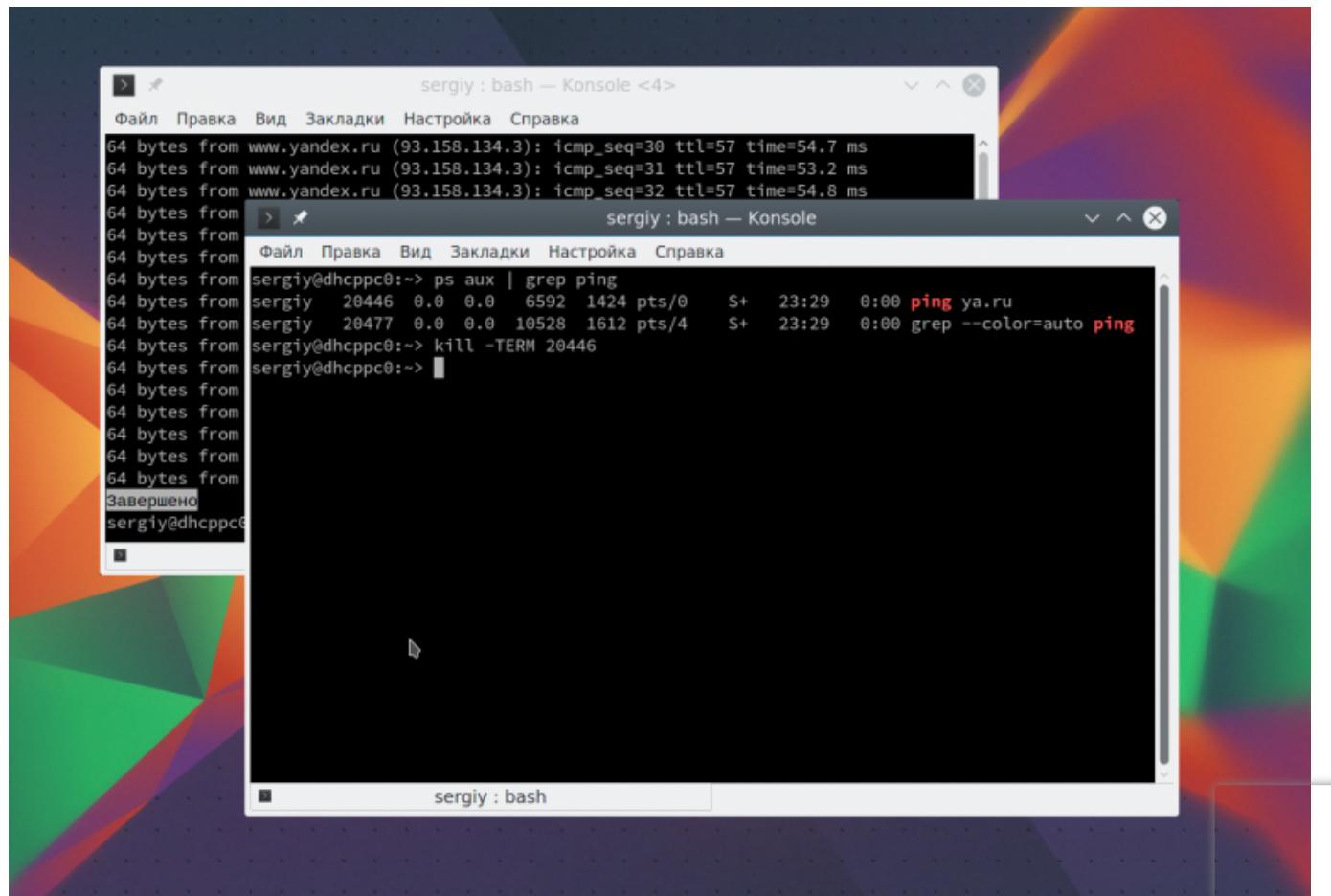
В первой строчке отобразится сама утилита `ping`, а во второй сама программа `ps`. Берем нужный PID и завершаем процесс с помощью SIGTERM:

```
$ kill 20446
```



Или:

```
$ kill -TERM 20446
```

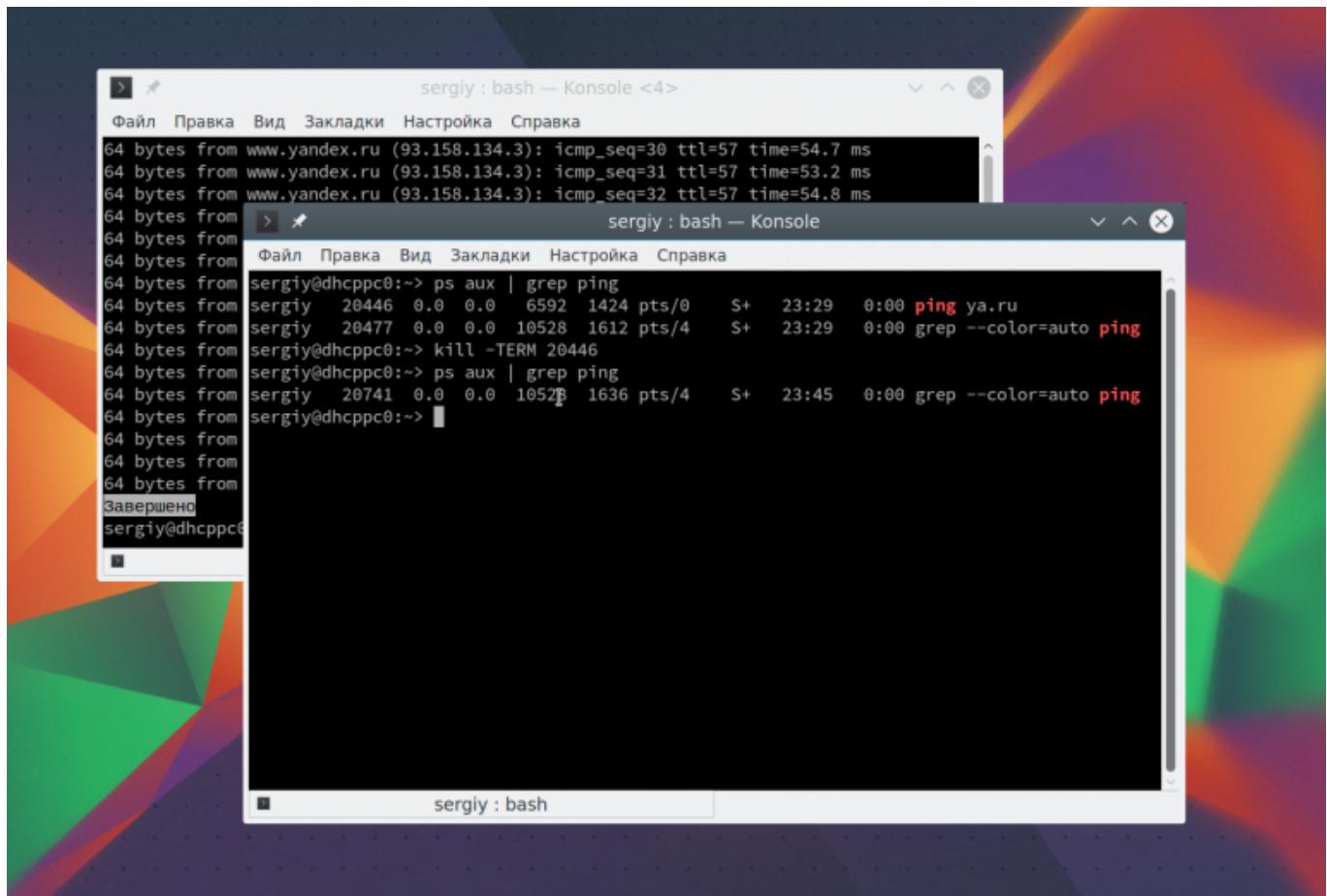


И только если после этой команды процесс продолжил висеть, а это вы можете проверить, выполнив ps. Только теперь можно выполнить SIGKILL:

Privacy

```
$ kill -KILL 20446
```

Теперь снова проверяем:

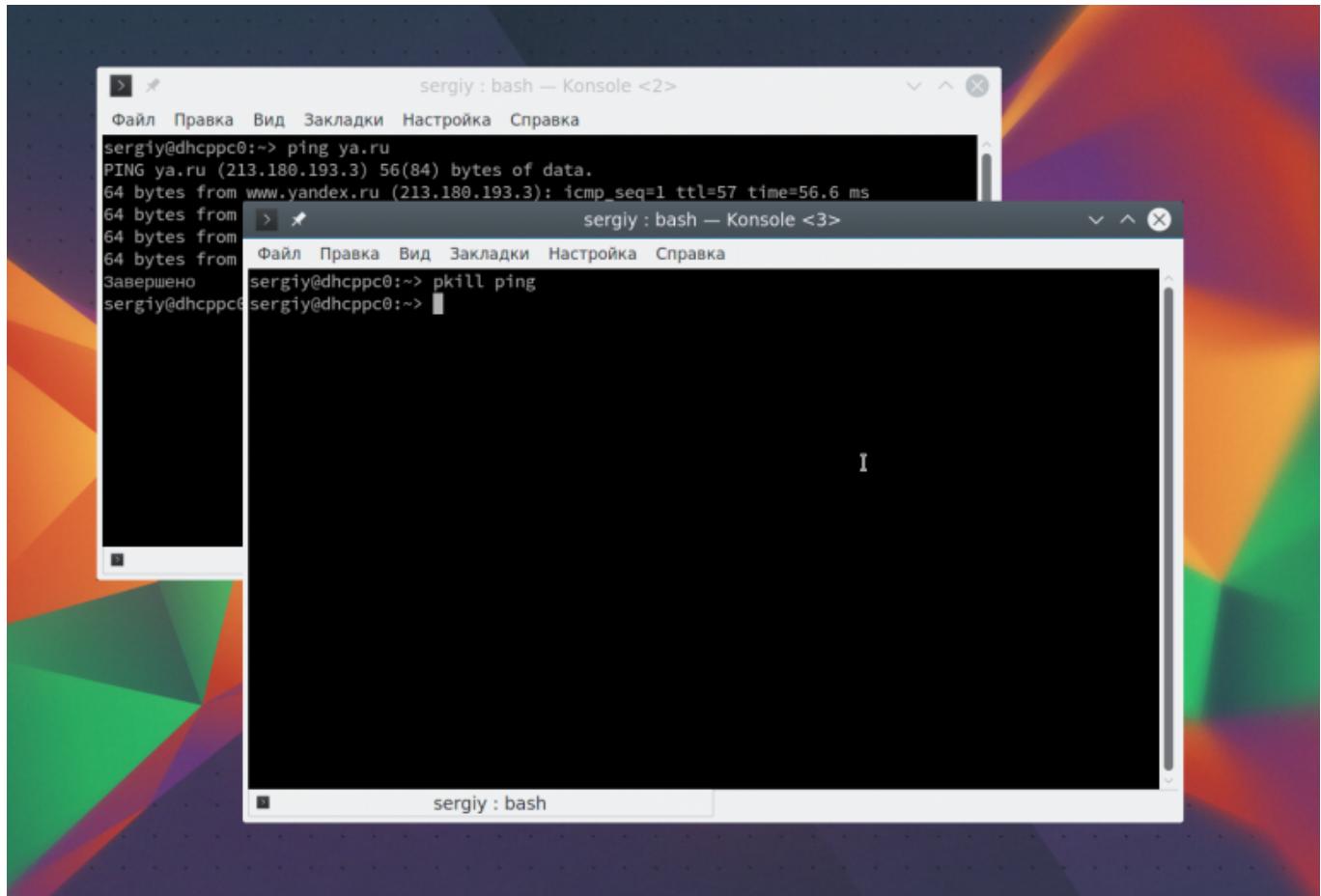


Если процесс запущен от суперпользователя, то, естественно, вам нужно использовать sudo. Не всегда удобно уничтожать процесс по его PID, как минимум, потому, что вам этот PID нужно еще узнать. Мы могли бы нагородить сложных конструкций с использованием args, чтобы вычислять автоматически pid по имени процесса и сразу же его завершать, но в этом нет необходимости. Уже существуют специальные утилиты.

Как завершить процесс с помощью pkill

Утилита `pkill` - это оболочка для `kill`, она ведет себя точно так же, и имеет тот же синтаксис, только в качестве идентификатора процесса ей нужно передать его имя. Утилита сканирует директорию `/proc` и находит PID первого процесса с таким именем, затем отправляет ему `SIGTERM`. Таким образом, вы можете убить процесс по имени `Linux`. Например, если мы хотим завершить тот же `ping`:

```
$ pkill ping
```

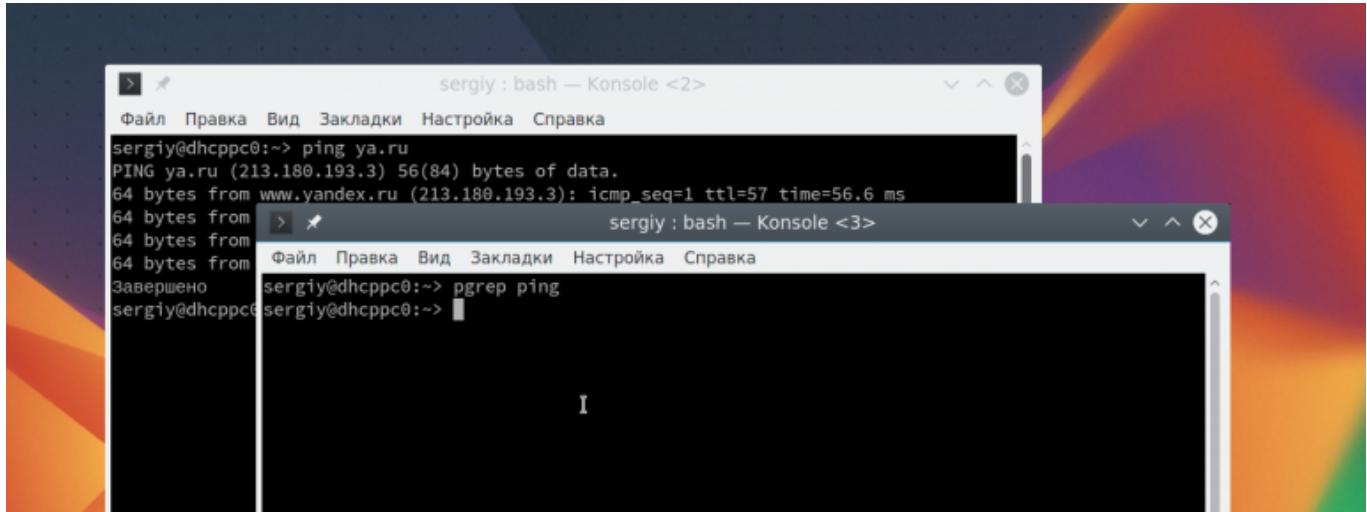


Также можно вручную задать тип сигнала:

```
$ pkill -TERM ping
```

Вместо `ps`, вы можете использовать утилиту `pgrep` для поиска `pid` процесса, убедимся что наша программа завершена:

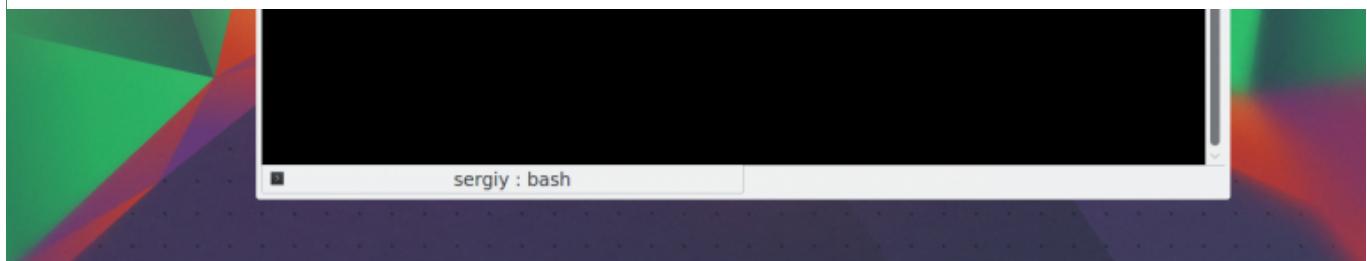
```
$ pgrep ping
```



Была ли эта информация полезной для вас?

Да

Нет

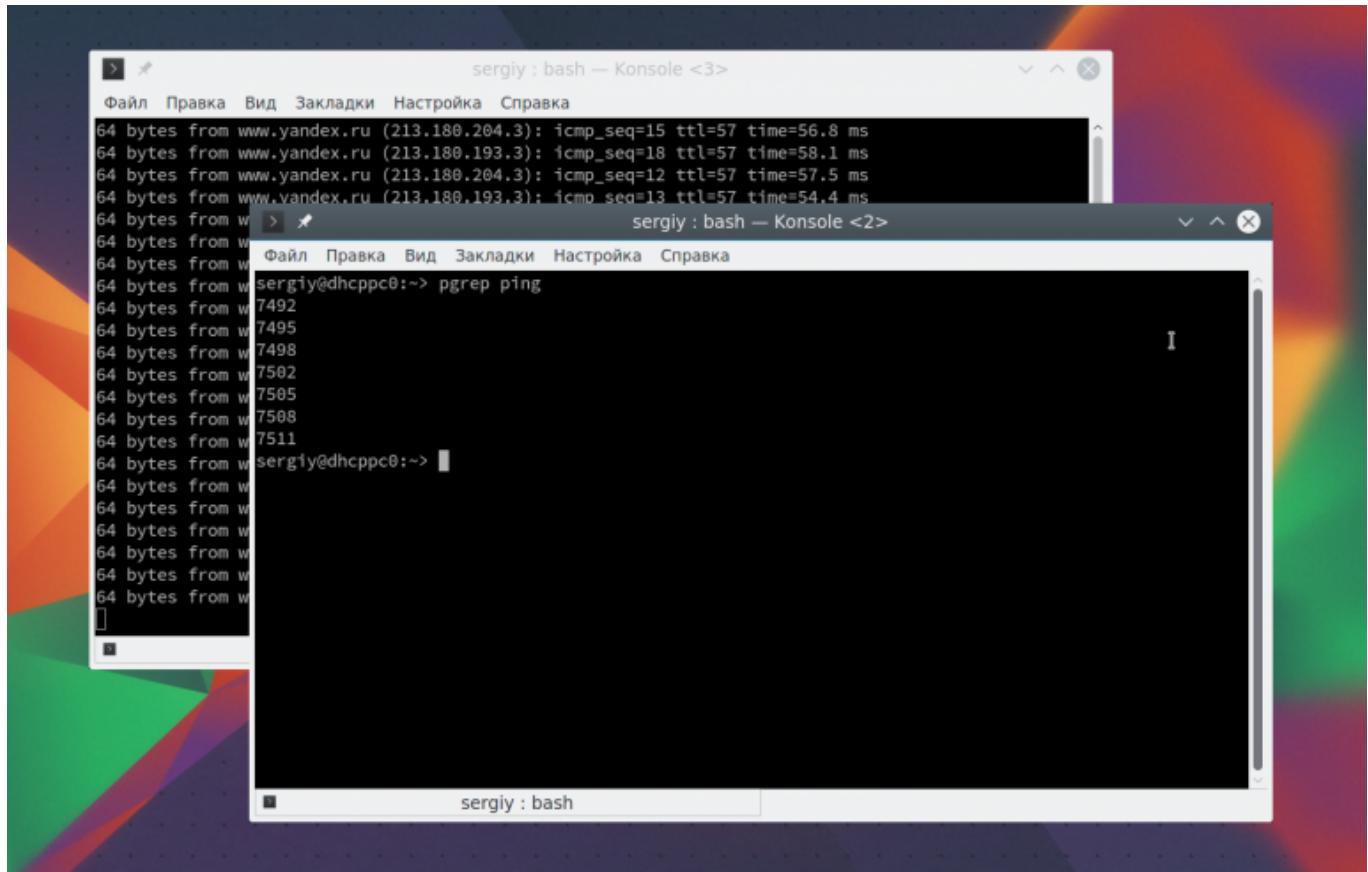


Но если вам программа создала несколько процессов, например, браузер chromium или firefox создают отдельный процесс для каждой из вкладок, то эта утилита мало чем поможет. Тут нужен следующий вариант.

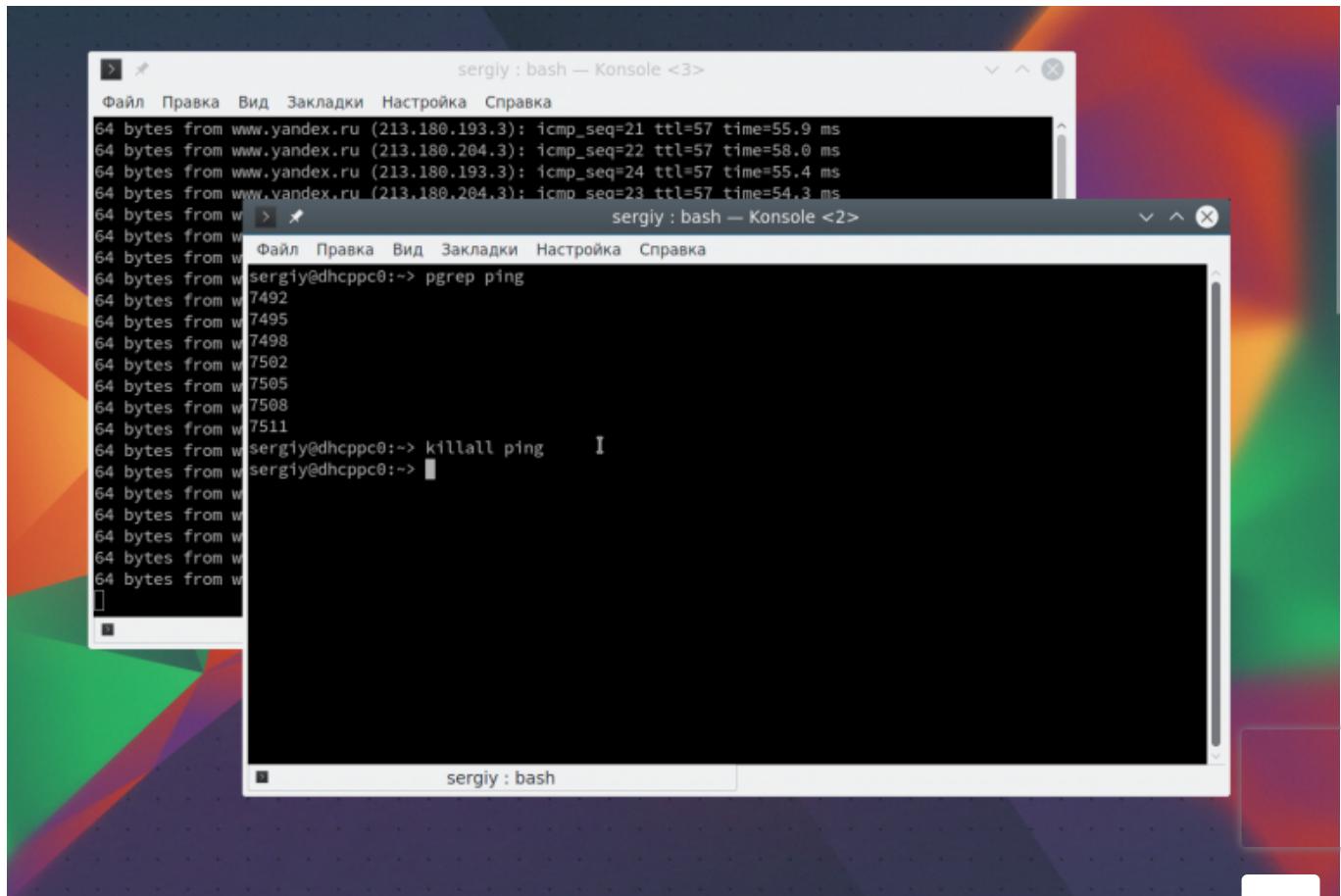
Как остановить процесс с помощью killall

killall работает аналогично двум предыдущим утилитам. Она тоже приминает имя процесса в качестве параметра и ищет его PID в директории /proc. Но эта утилита обнаружит все процессы, с таким именем и завершит их. Например:

```
$ killall ping
```

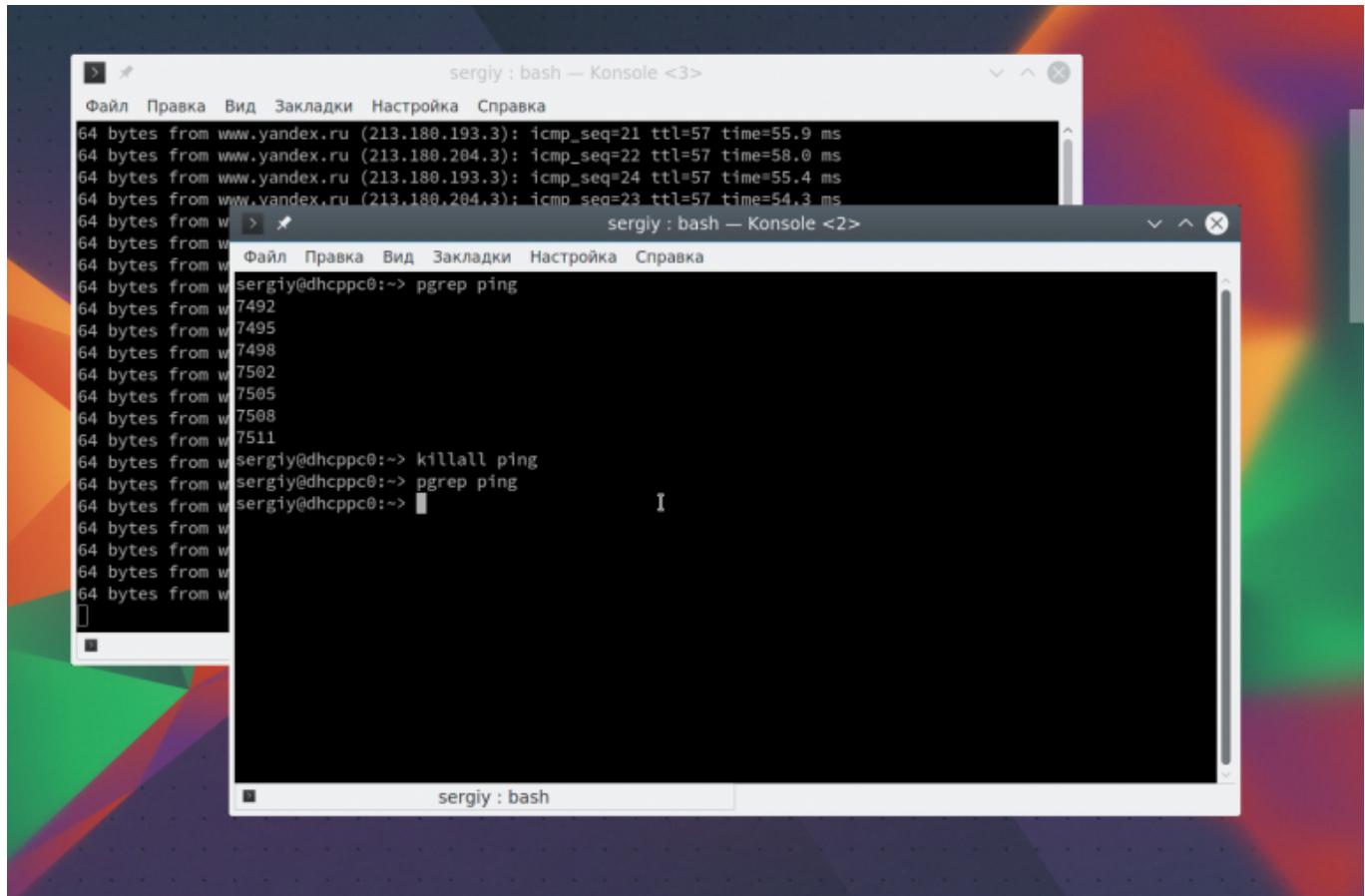


Как видите, запущено несколько процессов, осталось остановить процесс Linux с помощью `killall`:



Команда завершит все запущенные утилиты `ping`, вы можете убедиться в этом еще раз выполнив `pgrep`:

```
$ pgrep ping
```



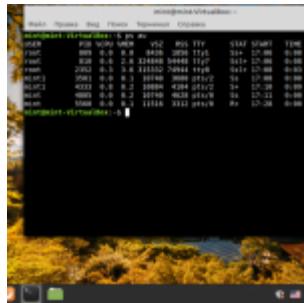
Выводы

В этой статье мы рассмотрели как убить процесс Linux. Иногда эта задача может быть очень полезной, но важно понимать, что ее нужно выполнять правильно. Нельзя сказать, что передача SIGKILL вместо SIGTERM очень опасна, но так делать не стоит. Надеюсь, эта информация была полезна для вас.

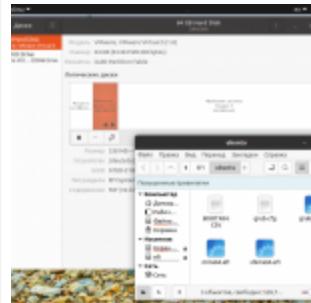
Похожие записи



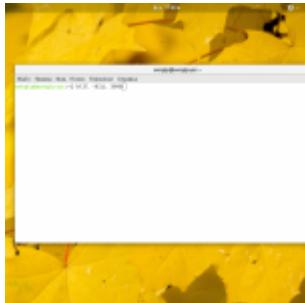
20 способов убить Linux



Как убить все процессы



Процесс загрузки Linux



Не убивается процесс Linux

Оцените статью

(22 оценок, среднее: 4,68 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.

4 комментария к “Как убить процесс Linux”

[Privacy](#)



[Павел Скуратович](#)
[4 марта, 2017 в 7:22пп](#)

У `killall` есть ещё замечательный ключик `-w` – не просто послать сигнал, но и подождать, пока все заданные процессы завершатся

[Ответить](#)



[Shhh](#)
[5 марта, 2017 в 12:43пп](#)

так `kill...` Ни слова о правах, масках, тгар-ах.. и о том как действительно завершить процесс, а не послать сигнал.

[Ответить](#)



[Денис](#)
[23 февраля, 2018 в 9:14дп](#)

На `sublime text 2` такая магия не действует, он неубиваем.

```
$ sudo killall -s 9 sublime_text  
$ pidof sublime_text  
3903
```

Естественно, `kill` с разными ключиками также бесполезен. Помогает только завершение сеанса. Ubuntu 16.04

[Ответить](#)



[Nicholas](#)
[29 августа, 2020 в 3:58пп](#)

Не помогла ни одна из перечисленных команд. При попытке убить указанный процесс выдает - Процесс не существует.

[Privacy](#)

[Ответить](#)

Оставьте комментарий

 Имя * Email

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

[Комментировать](#)

[Русский](#)

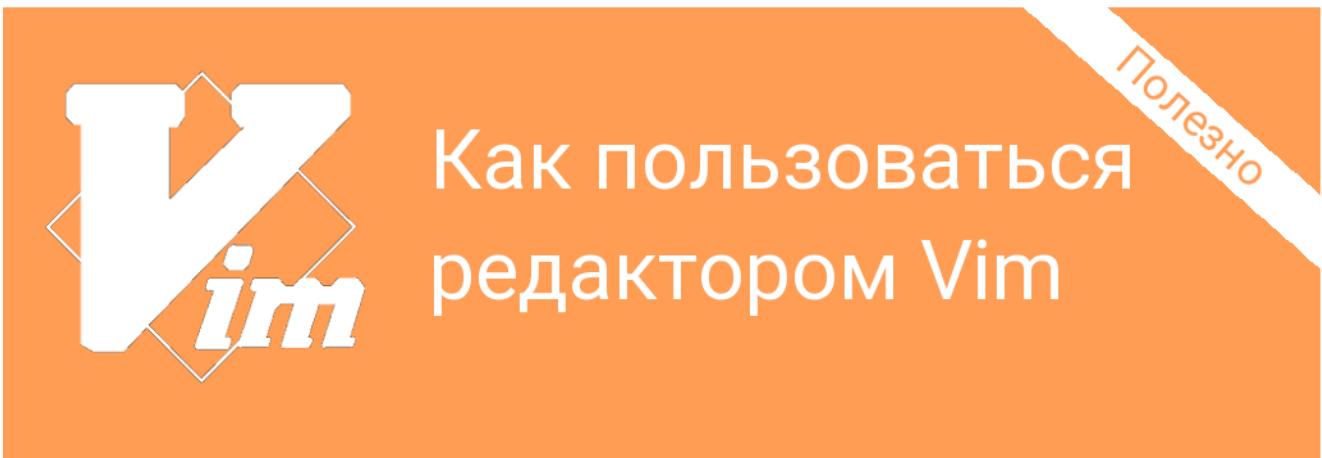
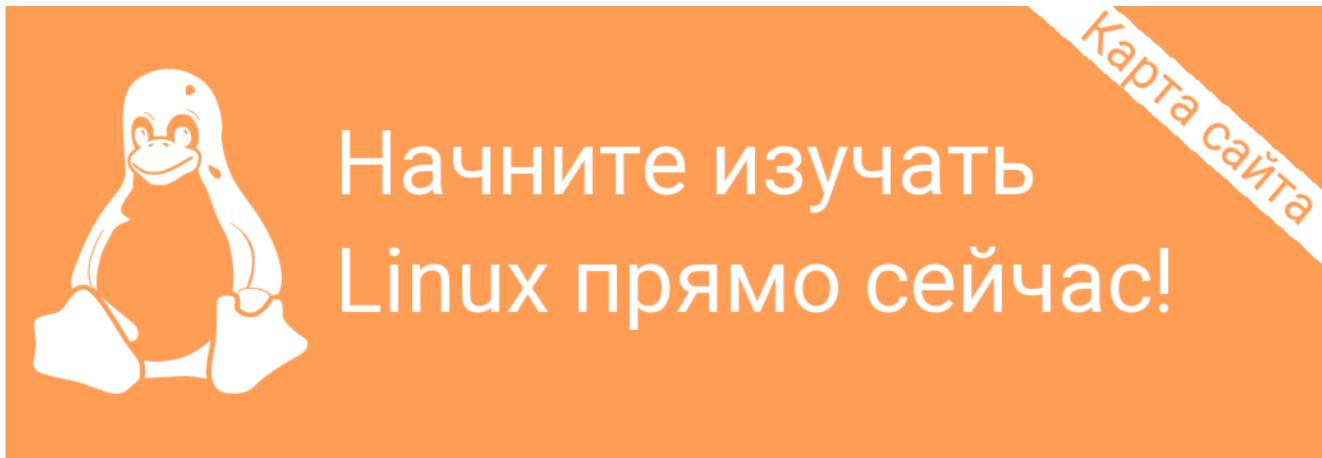
Поиск

[Privacy](#)

ПОИСК ПО КОМАНДАМ

Начните вводить команду

Поиск



Лучшие

Свежие

Теги



Команда chmod Linux

2020-04-13

Команда find в Linux

Privacy



2021-10-17

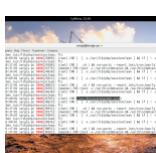
Как убрать IP-адрес Linux

2023-04-14



Настройка Сети

2021-10-01



Права доступа к файлам в Linux

2020-10-09



РАССЫЛКА

Ваш E-Mail адрес

Я прочитал(а) и принимаю политику конфиденциальности

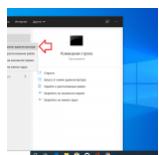
Sign up

Privacy



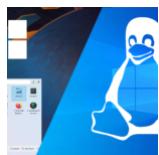
Windows

Списки



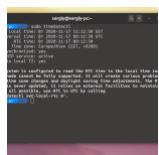
Восстановление Grub после установки Windows 10

2020-08-15



Установка Linux рядом с Windows 10 или 11

2023-02-08



Сбивается время в Ubuntu и Windows

2023-02-18



Ошибка Ubuntu не видит сеть Windows

2023-02-18

Смотреть ещё



META

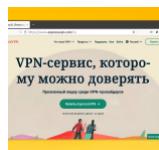
Privacy

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

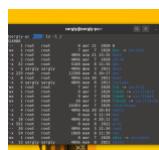
СЛЕДИТЕ ЗА НАМИ В СОЦИАЛЬНЫХ СЕТЯХ



Интересное

[Лучшие VPN сервисы для Linux](#)

2022-10-10

[Как правильно: папка или каталог в Linux](#)

2022-05-10

[Лучшие панели управления для Linux](#)

2020-12-08

[Что такое Inode](#)

2023-02-01

[Privacy](#)

©Losst 2024 CC-BY-SA [Политика конфиденциальности](#)



Privacy