Как пользоваться SSH

Обновлено: 01 октября 2021 **Опубликовано:** 17 февраля, 2016 от admin , 16 комменариев, время чтения: 12 минут

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

SSH - (Secure Shell) - это протокол удаленного управления компьютером с операционной системой Linux. В основном ssh используется для удаленного управления серверами через терминал. Если вы администратор нескольких серверов или даже продвинутый веб-мастер, то наверное, вы часто сталкиваетесь с необходимостью работать с тем или иным компьютером по ssh. В Linux для этого используется сервер ssh на машине, к которой нужно подключиться и клиент, на той из которой подключаются.

В этой инструкции мы рассмотрим как пользоваться ssh, а также ее возможности, о которых вы даже не знали. Скорее всего, вы уже знаете как подключиться к серверу по ssh, но у этой утилиты есть еще много возможностей, таких как передача файлов ssh, подключение без пароля или выполнение скрипта на удаленном сервере. Все это мы и рассмотрим далее в статье. Но начнем с самых основ.

Содержание статьи

Базовый синтаксис

Синтаксис команды выглядит следующим образом:

```
$ ssh [опции] имя пользователя@сервер [команда]
```

Важно заметить что ssh может работать по двум версиям протокола. Версии 1 и 2. Понятное дело, что версия 2 лучше и поддерживает больше типов шифрования и аутентификации. Больше в этой статье об отличиях протоколов мы говорить не будем и я буду подразумевать что вы используете версию 2.

Опции команды SSH

Теперь давайте рассмотрим самые основные опции команды ssh:

- **f** - перевести ssh в фоновый режим;
- **g** - разрешить удаленным машинам обращаться к локальным портам;
- **l** - имя пользователя в системе;
- **n** - перенаправить стандартный вывод в /dev/null;
- **p** - порт ssh на удаленной машине;
- **q** - не показывать сообщения об ошибках;

- **v** - режим отладки;
- **x** - отключить перенаправление X11;
- **X** - включить перенаправление X11;
- **C** - включить сжатие.

Это далеко не все опции утилиты, остальные выходят за рамки данной статьи. Многие настройки работы ssh можно изменять через конфигурационный файл ~/.ssh/config но здесь мы это тоже подробно рассматривать не будем.

Настройка сервера SSH

Настройки сервера SSH находятся в файле /etc/ssh/sshd_config. Многие из них мы тоже трогать не будем. Рассмотрим только самые интересные. Сначала откройте файл /etc/ssh/sshd.conf

Порт ssh

По умолчанию ssh работает на порту 22. Но такое поведение небезопасно, поскольку злоумышленник знает этот порт и может попробовать выполнить Bruteforce атаку для перебора пароля. Порт задается строчкой:

```
Port 22
```

Поменяйте значение порта на нужное.

Протокол SSH

По умолчанию сервер ssh может работать по двум версиям протокола, для совместимости. Чтобы использовать только протокол версии два раскомментируйте строчку:

```
#  Protocol 2
```

И приведите ее к такому виду:

```
Protocol 2
```

Рут доступ

По умолчанию Root доступ по ssh разрешен, но такое поведение очень небезопасно, поэтому раскомментируйте строчку:

```
PermitRootLogin no
```

Доступ только определенного пользователя к SSH

Мы можем разрешить доступ к ssh только для определенного пользователя или группы. Для этого добавьте строчки:

```
AllowUsers User1, User2, User3
AllowGroups Group1, Group2, Group3
```

Здесь User1 и Group1 - пользователь и группа к которым нужно разрешить доступ.

Выполнение X11 приложений

Не все знают но есть возможность использовать ssh для запуска полноценных X11 приложений. Об этом мы поговорим ниже, но чтобы все заработало необходимо разрешить эту возможность на стороне сервера, добавьте такую строчку:

```
X11Forwarding yes
```

Основные опции рассмотрели, перед тем как переходить дальше, не забудьте перезагрузить ssh сервер чтобы сохранить изменения:

```
$ service sshd restart
```

Использование SSH

Основная цель этой статьи - показать интересные и полезные способы использования ssh, о которых, возможно, вы не знали. Переходим к самому вкусному - возможности ssh.

Подключение к серверу

Чтобы просто подключиться к серверу по SSH используйте такую команду:

```
$ ssh user@host
```

Выполнить команду

Мы привыкли подключаться к удаленному серверу, а уже потом выполнять нужные команды, но на самом деле утилита ssh позволяет сразу выполнить нужную команду без открытия терминала удаленной машины. Например:

```
$ ssh user@host ls
```

Выполнит команду ls на удаленном сервере и вернет ее вывод в текущий терминал.

Выполнить локальный скрипт

Выполним интерпретатор bash на удаленном сервере и передадим ему наш локальный скрипт с помощью перенаправления ввода Bash:


```
$ ssh user@host 'bash -s' < script.sh
```

Бекап на удаленный сервер и восстановление

Мы можем сохранять бекэп диска сразу на удаленном сервере с помощью ssh. Перенаправим вывод dd с помощью оператора перенаправления |, затем сохраним его на той стороне в файл:

```
$ sudo dd if=/dev/sda | ssh user@host 'dd of=sda.img'
```

Теперь чтобы восстановить состояние диска из сделанной копии выполните:

```
$ ssh user@host 'dd if=sda.img' | dd of=/dev/sda
```

Здесь и выше /dev/sda имя файла вашего жесткого диска.

Аутентификация без пароля

Использование ssh пароля для входа на сервер не только неудобно но и небезопасно, потому что этот пароль в любой момент может быть подобран. Самый надежный и часто используемый способ аутентификации - с помощью пары ключей RSA. Секретный ключ хранится на компьютере, а публичный используется на сервере для удостоверения пользователя.

Настроить такое поведение очень легко. Сначала создайте ключ командой:

```
$ ssh-keygen -t rsa
```

Во время создания ключа нужно будет ответить на несколько вопросов, расположение оставляйте по умолчанию, если хотите подключаться без пароля - поле Passphare тоже оставьте пустым.

Затем отправляем ключ на сервер:

```
$ ssh-copy-id -i ~/.ssh/id_rsa.pub user@host
```

Вот и все. Теперь при попытке подключится к этому серверу пароль запрашиваться не будет, а сразу произойдет подключение. Смотрите подробнее создание открытого ключа для ssh.

Взять пароль из локального файла

Напомню, что хранить пароли в обычных текстовых файлах небезопасно, но если хотите, то да - возможно. Для этого используется оператор перенаправления ввода Bash:

```
$ ssh user@host < local_file.txt
```

Изменить приветствие SSH

При входе по ssh может выводиться приветствие, изменить его очень легко. За это отвечает файл /etc/issue. Просто откройте этот файл и введите нужный текст:

```
$ vi /etc/issue
```

```
Welcome!
```

Смотрим неудачные попытки входа SSH

Хотите посмотреть были ли попытки неудачного доступа по ssh к вашему серверу и с каких IP адресов? Запросто, все запросы логируются в файл /var/log/secure, отфильтруем только нужные данные командой:

```
$ cat /var/log/secure | grep "Failed password for"
```

Передача файлов по SSH

Кроме выполнения команд, можно копировать файлы по ssh. Для этого используется утилита scp. Просто укажите файл, который нужно передать, удаленный сервер и папку на сервере, вот:

```
$ scp /адрес/локального/файла пользователь@хост:адрес/папки
```

Например:

```
$ scp ~/test.txt user@host:documents
```

Кроме утилиты scp, передача файлов ssh может быть выполнена более хитрым способом. Прочитаем файл и с помощью cat, передадим, а там сохраним поток в файл:

```
$ cat localfile | ssh user@host "cat > remotefile"
```

Или так:

```
$ ssh user@host "cat > remotefile" < localfile
```

Пойдем еще дальше, вы можете сжимать файлы перед передачей с помощью tar, а потом их сразу же на лету распаковывать:

```
$ tar czf - /home/user/file | ssh user@host tar -xvzf -C /home/remoteuser/
```

Такое копирование файлов ssh позволяет отправлять сразу целые папки.

Запуск графических приложений по ssh

Если вам нужно запустить то или иное графическое приложение на удаленной машине необязательно для этого использовать VNC, вы можете обойтись возможностями ssh. Программа будет выполняться на стороне сервера, а вам будет лишь транслироваться окно, чтобы вы могли сделать все что нужно. Причем все данные шифруются. Чтобы эта функция работала, нужно включить ее поддержку на стороне сервера.

Затем просто выполняем команду запуска графического приложения на удаленном сервере вот таким образом:

```
$ ssh -XC user@remotehost "eclipse"
```

Как вы уже видели опция X разрешает перенаправление X11 на стороне клиента, а C - сжатие данных.

Завершение сессии SSH

Если вы использовали SSH с нестабильным интернетом, когда соединение время от времени рвется, то вам уже, наверное, надоело закрывать терминал, потому что иначе, на первый взгляд, сеанс никак не прекратить. Когда соединение с удаленным сервером разорвано вы не можете ввести никакую команду и сочетания клавиш **Ctrl+C**, **Ctrl+Z**, **Ctrl+D** не работают. И не будут работать поскольку клиент пытается отправить эти команды на сервер. Но есть решение - Escape последовательности. Чтобы активировать их поддержку добавьте строку:

```
EscapeChar ~
```

В файл **/etc/ssh/ssh_config**. Теперь, чтобы разорвать SSH соединение достаточно нажать **Enter** и набрать:

```
~.
```

Другие управляющие символы можно узнать нажав:

```
~?
```

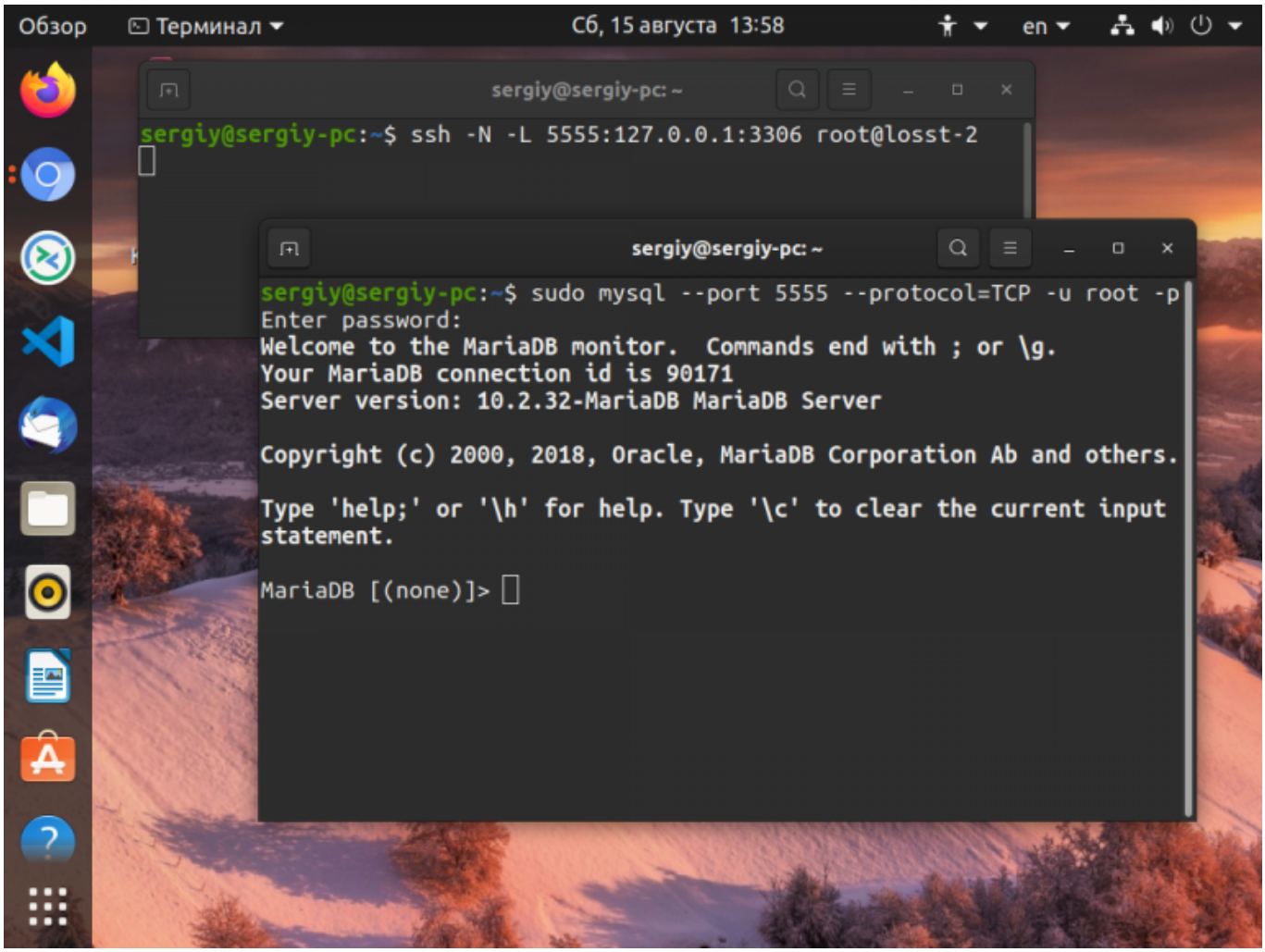
Туннели SSH

С помощью SSH туннелей вы можете пробросить порт с удалённого сервера на локальную машину. Это очень полезно, в первую очередь, для разработчиков. Для того чтобы пробросить порт с удалённой машины локальной используйте опцию **-L** и такой синтаксис:

```
$ ssh -L локальный_порт:удаленный_адрес:удаленный_порт пользователь@сервер
```

Например, сделаем удалённую базу данных доступной локально на порту 5555. Для этого выполните подставив свои значения:

```
$ $ ssh -N -L 5555:127.0.0.1:3306 root@losst-1
```



Опция **-N** сообщает, что команду на удалённой машине выполнять не нужно. Локальный порт - 5555, поскольку сервер баз данных слушает на локальном интерфейсе удалённой машины, то и здесь надо указывать адрес 127.0.0.1. А порт MySQL по умолчанию 3306. Если же вы хотите чтобы локальный сервис был доступен на удалённой машине, то следует использовать опцию -R:

```
$ $ ssh -N -R 5555:127.0.0.1:3306 root@losst-1
```

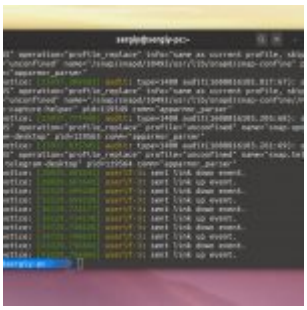
Теперь локальная база данных на порту 3306 будет доступна на удалённом сервере при обращении к порту 5555.

Выводы

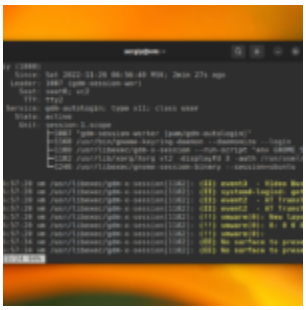
Теперь вы знаете как пользоваться SSH. Как видите, технология SSH позволяет сделать намного больше чем можно предположить с первого взгляда, и это еще далеко не все. Какие интересные возможности SSH используете вы при повседневной работе? Поделитесь в комментариях!

Похожие записи

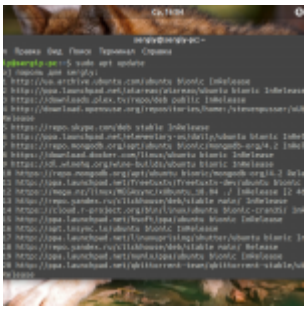
Как пользоваться ffmpeg



Как пользоваться dmesg



Как пользоваться loginctl



Как пользоваться apt

Оцените статью

(18 оценок, среднее: **4,78** из 5)

Команды

сеть

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.

16 комментариев к “Как пользоваться SSH”

Игорь Романов
18 февраля, 2016 в 10:29 пп

В эту статью добавить бы еще про sftp и sshfs.

Ответить

admin
19 февраля, 2016 в 8:45 дп

Да! Отличная идея, думаю так и сделаю.

Ответить

answIT
21 февраля, 2016 в 6:10 дп

Настоящая магия ssh в туннелировании. С его помощью можно проделывать удивительные штуки. Недаром говорят - что ssh-туннели - это VPN для бедных. Дело в том, что через них действительно можно открыть доступ к чему угодно, поскольку в туннель можно направить любой трафик, пробросить любые порты. Но я скажу, не для бедных - а там где нужно быстро, прямо здесь и сейчас получить доступ туда, куда его нет. ВПН - настраивать гораздо сложнее и дольше, чем пробросить туннель за несколько секунд. А ещё, ssh сервер можно запустить на windows. Как и зачем это нужно - об этом тоже есть статья у нас на сайте.

Ответить

stc

30 августа, 2016 в 10:36 пп

"По умолчанию Root доступ по ssh разрешен," - вранье - в большинстве дистрибутивов логин рута на ssh по умолчанию запрещен.

Ответить

admin

31 августа, 2016 в 5:07 дп

Во всех, которые я тестировал до сих пор - был разрешен.

Ответить

ANTON

23 февраля, 2020 в 1:43 пп

Зачем про sftp, если есть scp?)
Можно добавить еще про ~/.ssh/config

Ответить

Den

15 ноября, 2017 в 6:45 пп

ЗАВЕРШЕНИЕ СЕССИИ SSH - А не подскажите какой сигнал бросает ubuntu при обрыве соединения интернет

Ответить

Den

15 ноября, 2017 в 6:54 пп

Как обрыв соединения пытаюсь перехватить событие SINGHUP, но соединение просто виснет

Ответить

Sairan

25 ноября, 2018 в 7:18 пп

Здравствуйте!
С vi /etc/issue не получилось, добавить в новой строчке текст, авторизовался локально ssh root@localhost и ничего не увидел

Ответить

Keks

19 апреля, 2019 в 3:34 пп

Спасибо за статью!
Еще постоянно пользуюсь временным пробросом порта (портов). Например, когда нужно посмотреть экран, подключаюсь прокидывая VNC-порт 5900. Преимущество в том, что я запускаю со своего компьютера соединение на localhost, и удаленная машина видит, будто к ней на порт подключаются с localhost.
ssh -L 5900:localhost:5900 user@host
Еще можно выполнять двойной проброс порта. Подключиться на удаленный сервер №1, а с него "перепробросить" порт на другой удаленный сервер №2, доступный для подключения только с сервера №1:
My_comp: Keks$ ssh -L 5900:localhost:12399 user@Server1
Server1: user$ ssh -L 12399:localhost:5900 user@Server2

Ответить

Гость

11 октября, 2021 в 8:21 пп

А как запускать графические приложения на сервере от рута?

Ответить

Гость

13 октября, 2021 в 9:00 дп

У меня при вводе команды ssh -XC имя_пользователя@ip_адрес "sudo имя_программы" выводиться сообщение "нет tty и не указана программа askpass". В то время как без sudo приложения удаленного сервера запускаются нормально. Сам терминал от рута запускается тоже а вот графические приложения нет.

Ответить

Валентин

18 февраля, 2022 в 3:45 пп

Народ, а где можно попрактиковаться с SSH& А то у меня удаленного сервера нет

Ответить

admin

18 февраля, 2022 в 8:39 пп

В VirtualBox можно поставить.

Ответить

Павел

7 июня, 2022 в 12:17 дп

Здравствуйте. написал конфиг для соединения приватного компьютера с публичн компьютером. ~/.ssh/config

```
Host monero
Hostname "IP адрес публичного сервера"
User monero
RemoteForward 0.0.0.0:2222 127.0.0.1:22
ServerAliveInterval 30
ServerAliveCountMax 5
ExitOnForwardFailure yes
```

запускаю с приватного сервера "ssh monero" и все запускается и работает. Решил написать демона для того что бы соединение устанавливалось после перезагрузки приватного сервера. и не запускается. Подскажите что я не так написал в демоне ? По моему что то в строке запуска? не могу понять -что конкретно.

```
[Unit]
Description = Tunnel
After = network-online.target
Wants = network-online.target
#
[Service]
User = paul
Type = simple
ExecStart = /home/paul/ssh monero
Restart = always
RestartSec = 30s
#
[Install]
WantedBy = multi-user.target
```

Ответить

Максим

21 августа, 2022 в 5:06 пп

Здравствуйте!
Никак не могу подключиться по ssh.
Ввожу верный пароль, пишет "Permission denied, please try again.".
Пожалуйста подскажите как исправить!

Ответить

Оставьте комментарий

Имя *

Email

Я прочитал и принимаю политику конфиденциальнсти. Подробнее Политика конфиденциальности *

Комментировать

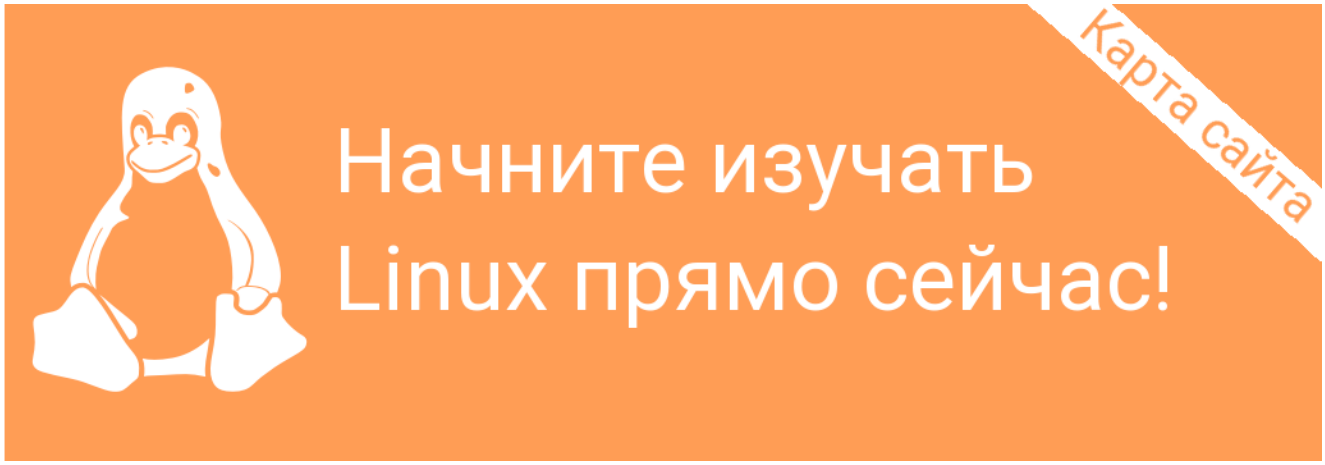
Русский

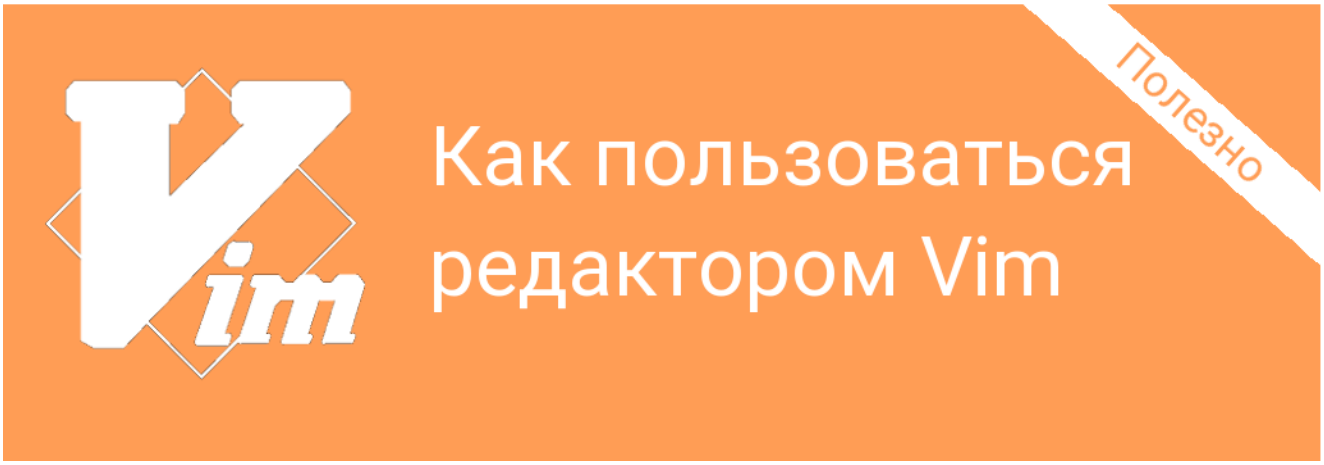
Поиск

ПОИСК ПО КОМАНДАМ

Начните вводить команду

Поиск





Лучшие Свежие Теги

Команда chmod Linux

2020-04-13

Команда find в Linux

2021-10-17

Как узнать IP-адрес Linux

Privacy

2023-04-14

Настройка Cron

2021-10-01

Права доступа к файлам в Linux

2020-10-09

РАССЫЛКА

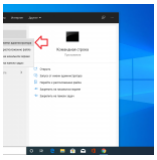
Ваш E-Mail адрес

☐ Я прочитал(а) и принимаю политику конфиденциальности

Sign up

Privacy

Windows Списки



Восстановление Grub после установки Windows 10

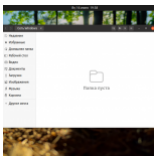
2020-08-15

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

МЕТА

Регистрация

Войти

Лента записей

Лента комментариев

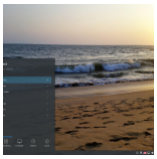
СЛЕДИТЕ ЗА НАМИ В СОЦИАЛЬНЫХ СЕТЯХ

Privacy

Интересное

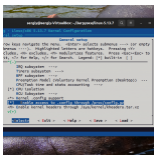
Прикольные команды Linux

2022-09-13



Самые маленькие дистрибутивы Linux

2020-12-15



Сборка ядра Linux

2021-08-14

Использование Docker для чайников

2021-04-08