Лабораторная работа №5 (DevOps №1). Базовые инструменты администрирования и автоматизации работы Linux

Оглавление

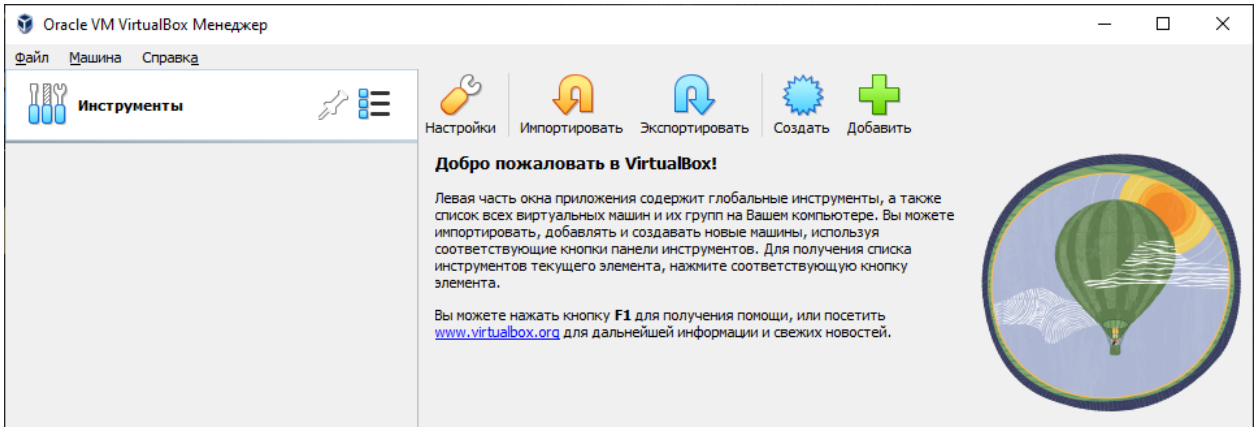
Установка и настройка Virtual Box

Пароль администратора на всех заранее созданных виртуальных машинах (Alt Linux, Linux Mint): password

VirtualBox – это специальное средство для виртуализации, позволяющее запускать операционную систему внутри другой. Оно поставляется в двух версиях – с открытым и закрытым исходным кодом. С помощью VirtualBox мы можем не только запускать ОС, но и настраивать сеть, обмениваться файлами и делать многое другое.

VirtualBox поставляется в бесплатном доступе для Linux, Solaris, macOS и Microsoft Windows. Скачать ее можно с официального сайта: https://www.virtualbox.org/wiki/Downloads

Как только установка будет завершена, перед нами отобразится главный экран программы.

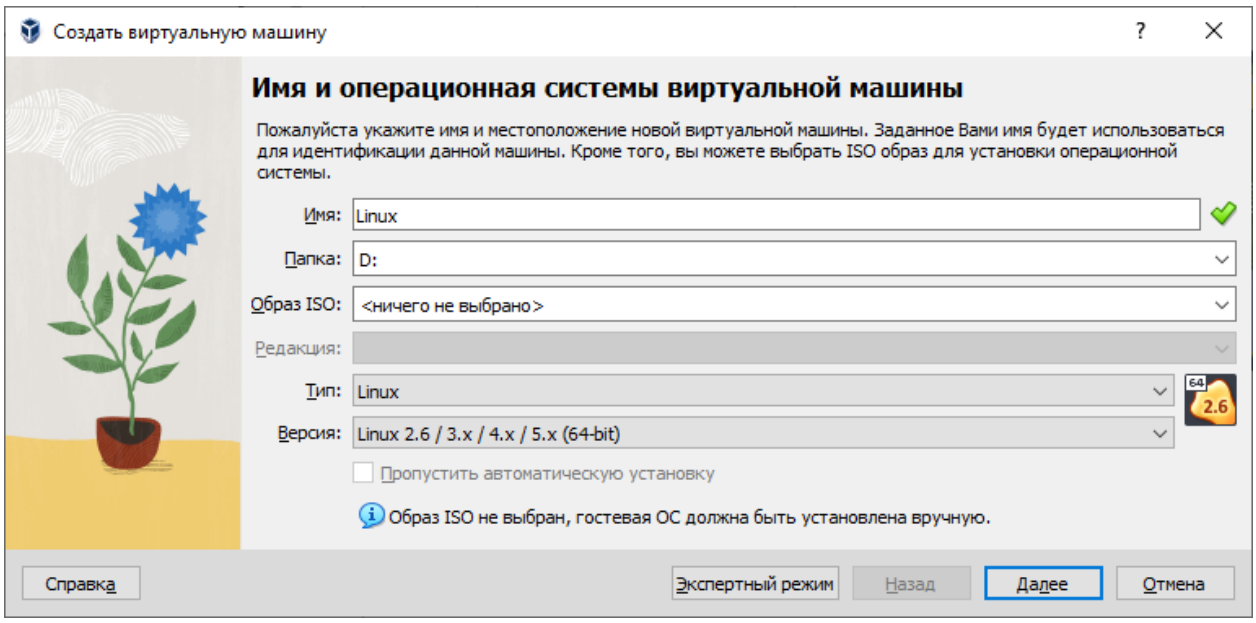


Рассмотрим, как создать виртуальную машину и провести дополнительные настройки.

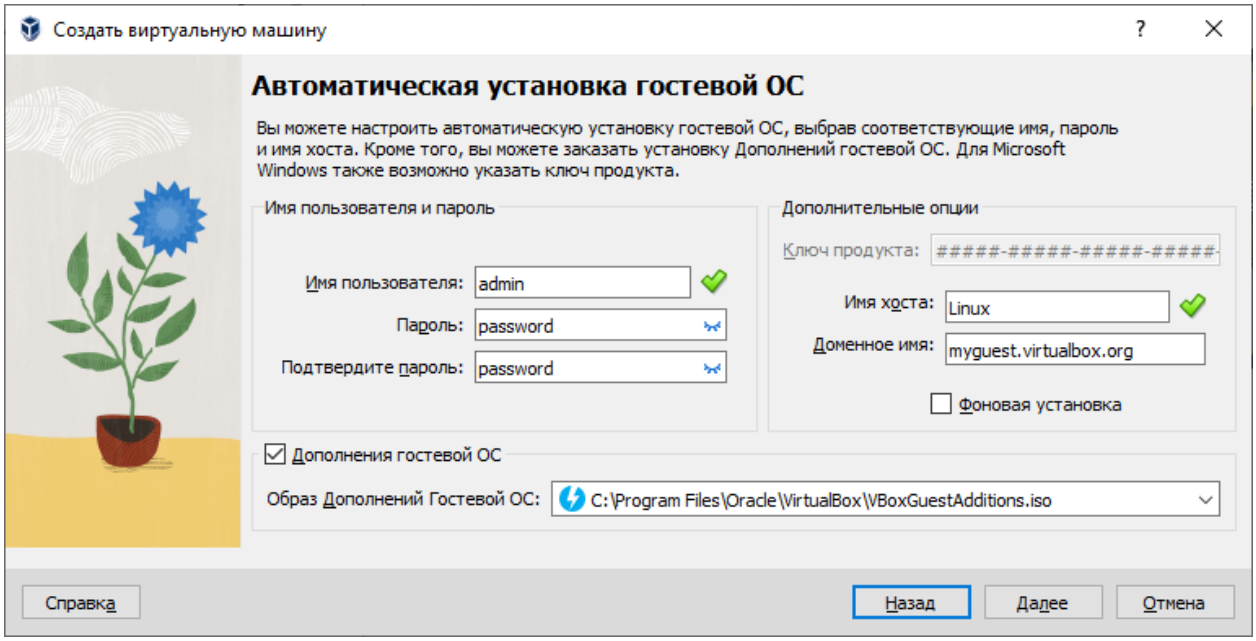
Создаем виртуальную машину

Основная функция VirtualBox – виртуализация. Чтобы запустить новую операционную систему, необходимо создать для нее виртуальную машину. Для этого необходимо выполнить следующее:

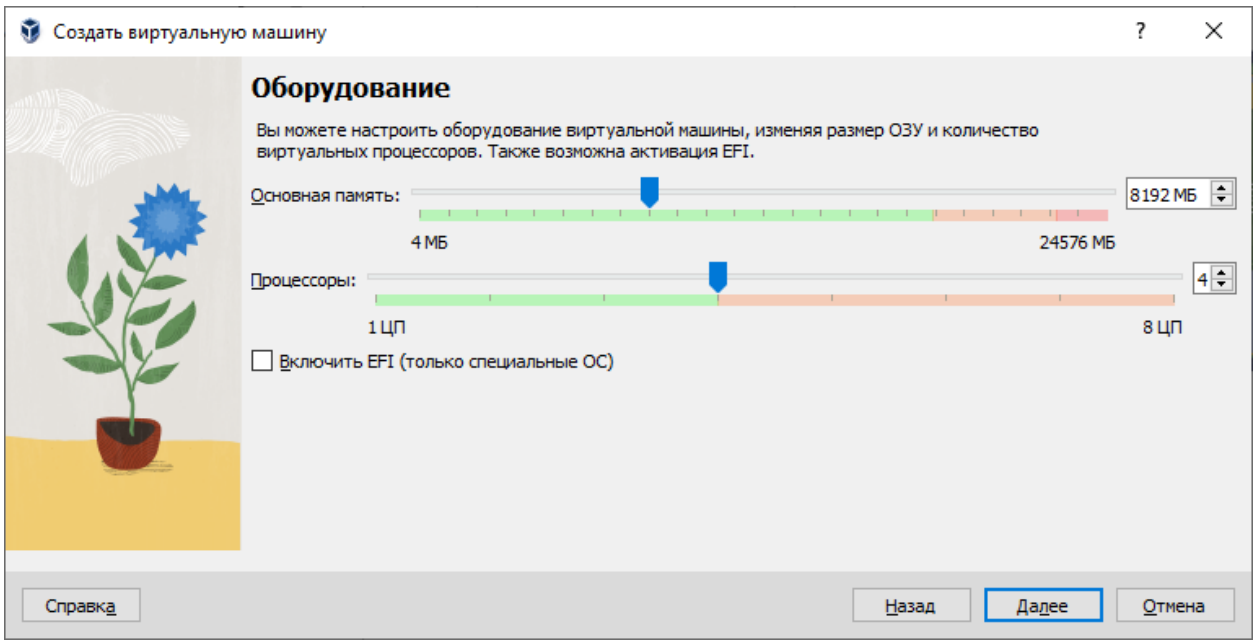
Запускаем VirtualBox и в правой части выбираем «Создать».



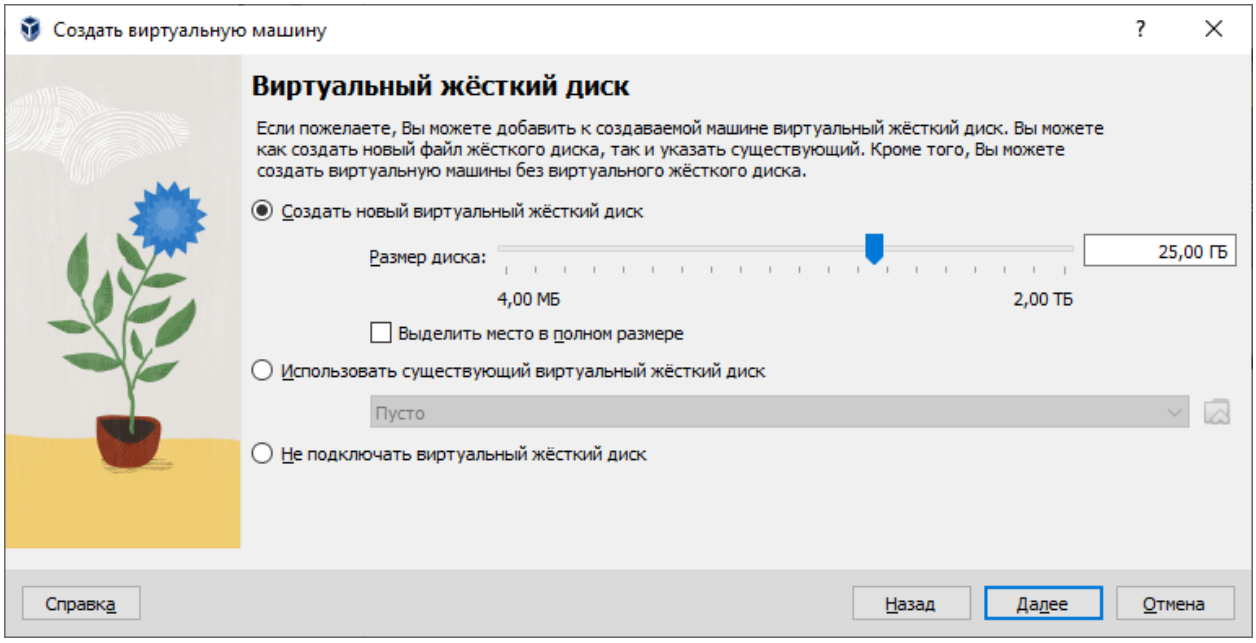
В отобразившемся окне прописываем имя операционной системы и указываем путь до машины. Обратите внимание на то, что тип ОС выбирается автоматически в зависимости от введенного названия. Также здесь можно сразу указать путь к образу установочного диска ОС, тогда следующим пунктом будут параметры автоматической установки.



Выбираем, сколько оперативной памяти и ядер процессора будет отведено под будущую ОС.



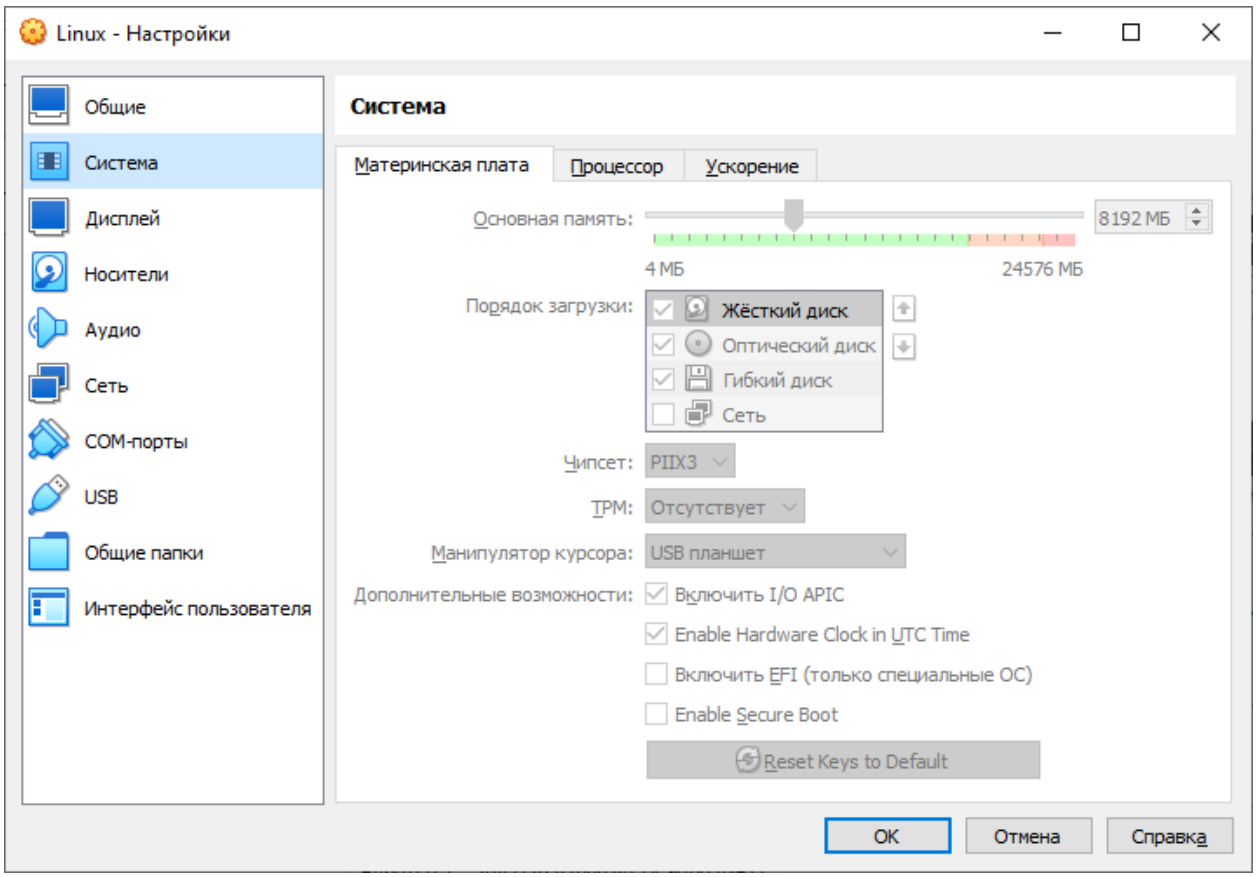
Далее выбираем объем и тип виртуального жесткого диска: динамический или фиксированный (выделить место в полном объеме). При динамическом виртуальном жестком диске размер файла диска будет увеличиваться в зависимости от его наполнения в виртуальной машине.



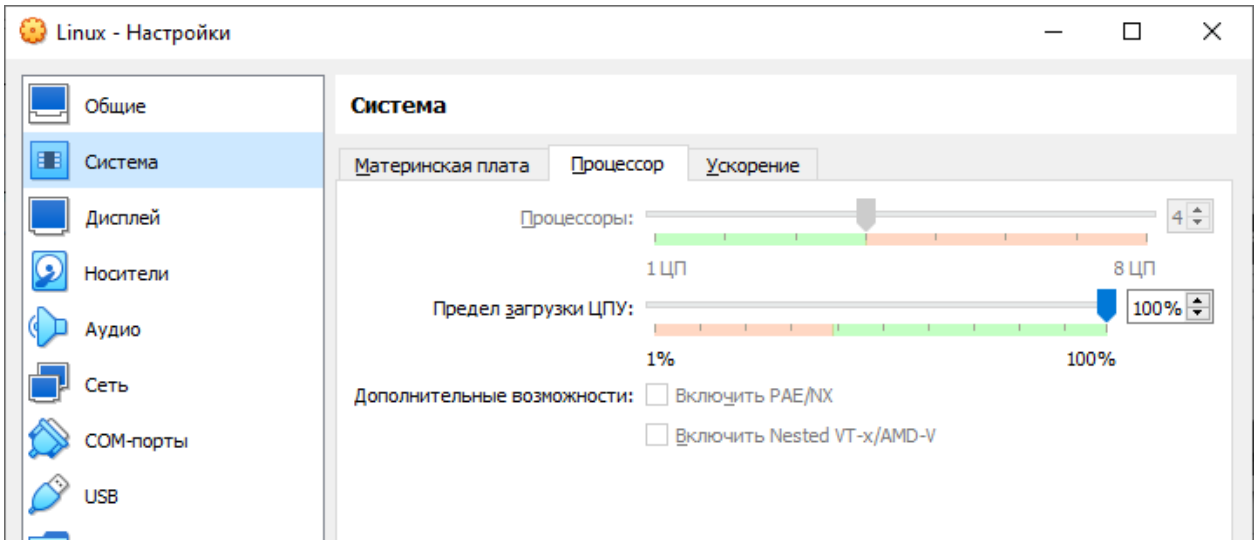
В результате будет создана новая виртуальная машина.

Обратите внимание на то, что если мы еще не устанавливали операционную систему, то теперь мы можем запустить виртуальную машину и поставить на нее нужную ОС, но перед этим пройдемся по некоторым параметрам.

Кликаем правой кнопкой мыши по виртуальной машине и выбираем «Настроить…» - «Система».

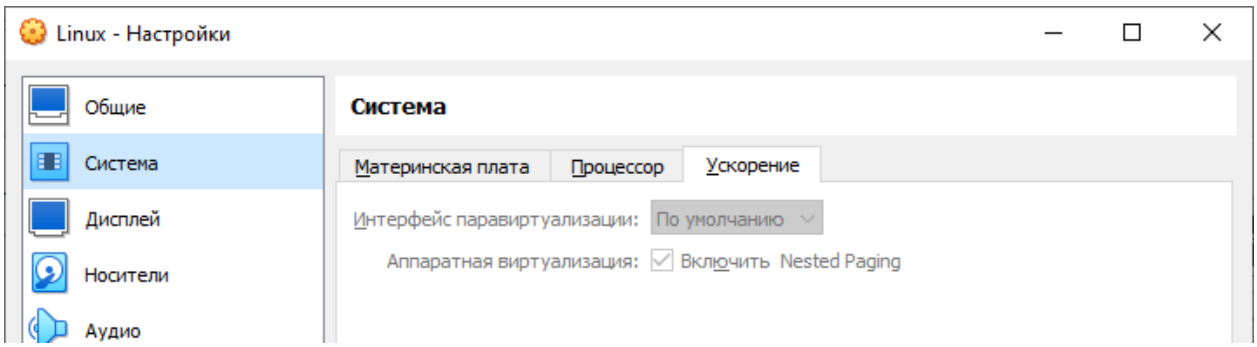


В отобразившемся окне в разделе «Процессор» можно изменить число процессоров и предел загрузки ЦПУ.

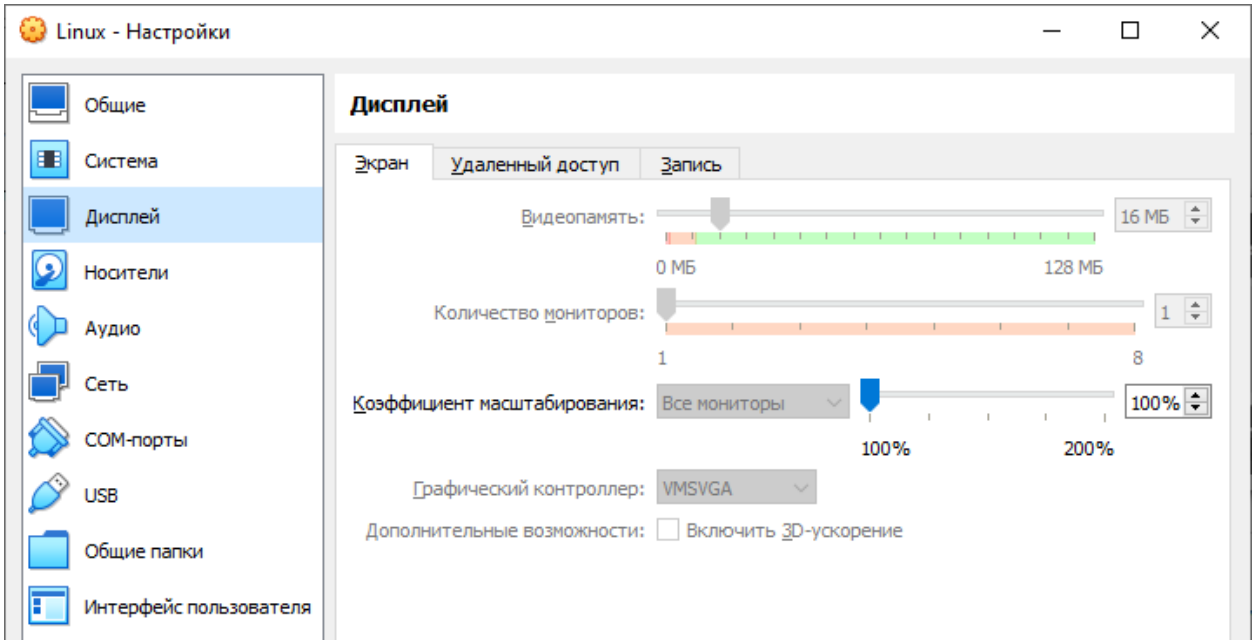


Функция «Включить PAE/NX» предназначена для поддержки 4 и более Гб ОЗУ в 32-битных системах.

Во вкладке «Ускорение» мы можем выбрать режим виртуализации, а также настроить дополнительные параметры для увеличения скорости работы.



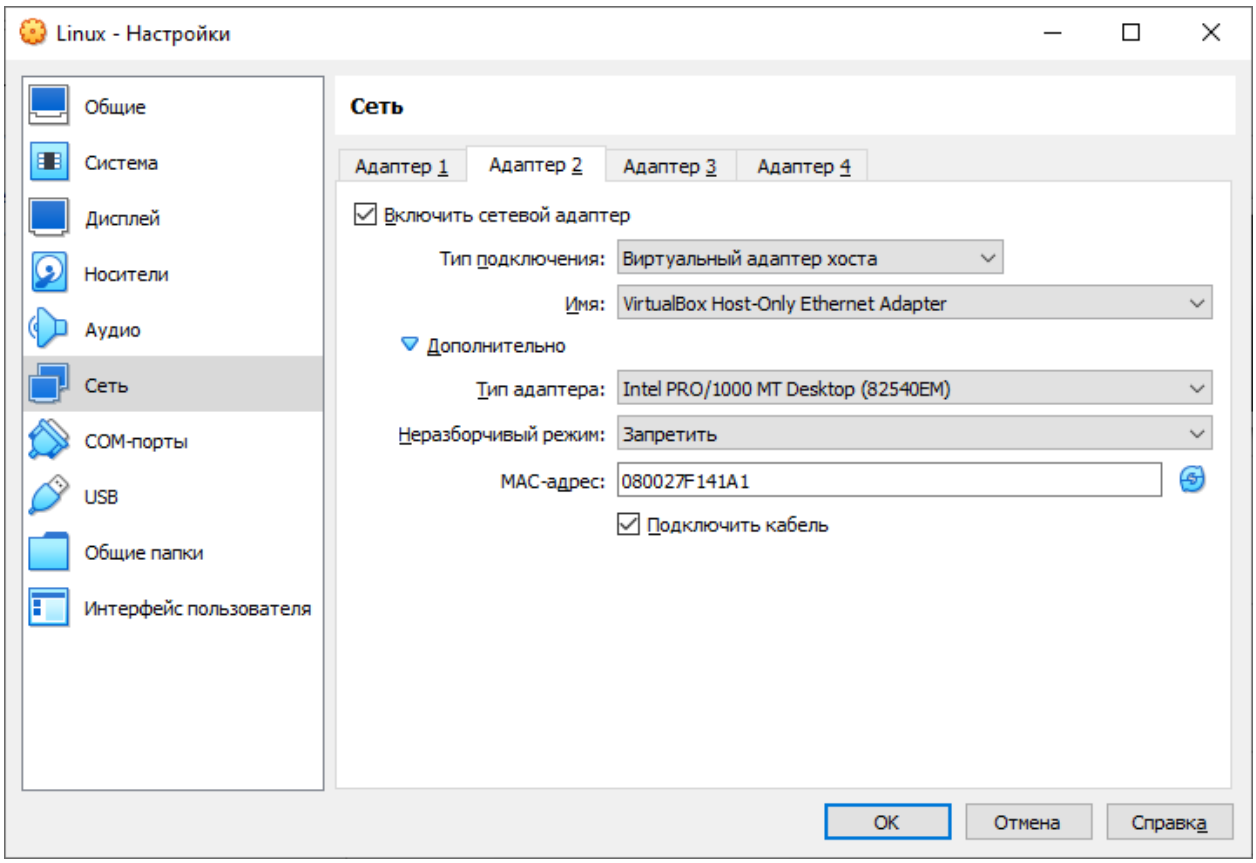
Иногда при установке новой виртуальной машины значение видеопамяти по умолчанию составляет 16 Мб, тогда как рекомендуется выделять не менее 128 Мб. Изменить это можно в настройках раздела «Дисплей».



В этом же разделе можно установить количество мониторов, изменить коэффициент масштабирования и многое другое.

Изначально виртуальная машина использует сеть NAT, что вполне удобно, если необходимо получить доступ к интернету. Если же вам нужно настроить взаимосвязь между разными ВМ, то потребуется выполнить дополнительные настройки.

В настройках переходим в раздел «Сеть» и заходим в подраздел «Адаптер 2». Там активируем пункт «Включить сетевой адаптер» и указываем тип подключения «Виртуальный адаптер хоста».



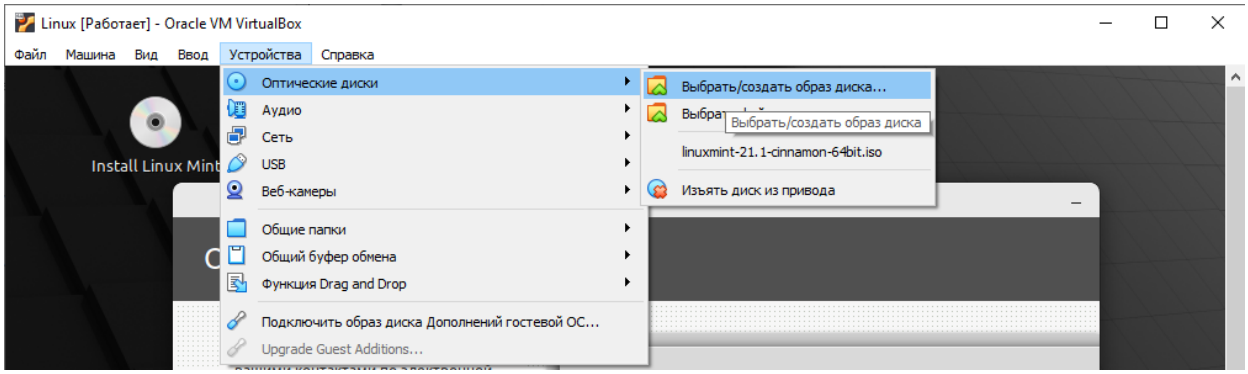
Обратите внимание на имя – теперь все, кто будет его использовать, автоматически подключатся к единой виртуальной сети.

Еще одна полезная функция – «Клонирование». С ее помощью мы можем сделать резервную копию виртуальной машины, чтобы в последующем обратиться к ней при возникновении различного рода проблем.

Для этого кликаем правой кнопкой мыши по виртуальной машине и выбираем «Клонировать...».

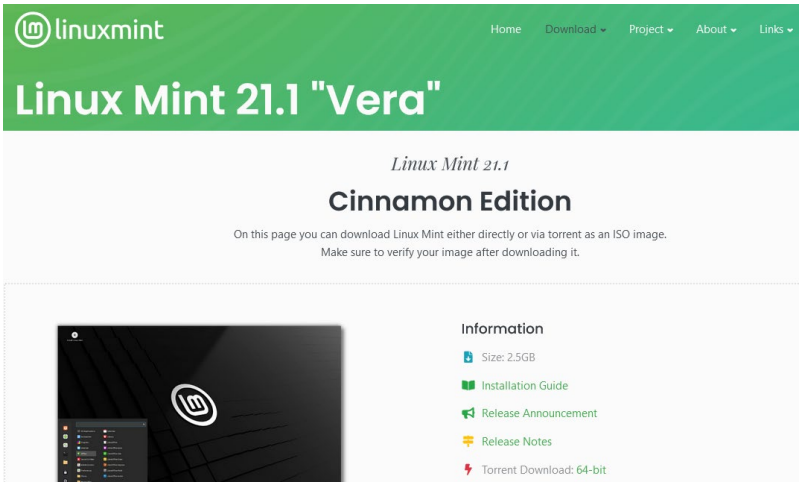
Для запуска созданной виртуальной машины в VirtualBox потребуется загрузочный диск необходимой операционной системы. Это обычный образ, который используется при установке ОС на ПК. Установить его в VirtualBox мы можем следующим образом:

1. Загрузить ISO-образ дистрибутива системы.
2. Выбираем созданную ранее виртуальную машину и в правой части нажимаем на кнопку «Запустить».
3. После запуска виртуальной машины в меню выбрать «Устройства» - «Выбрать файл диска».
4. Установить ОС, следуя инструкциям инсталлятора.

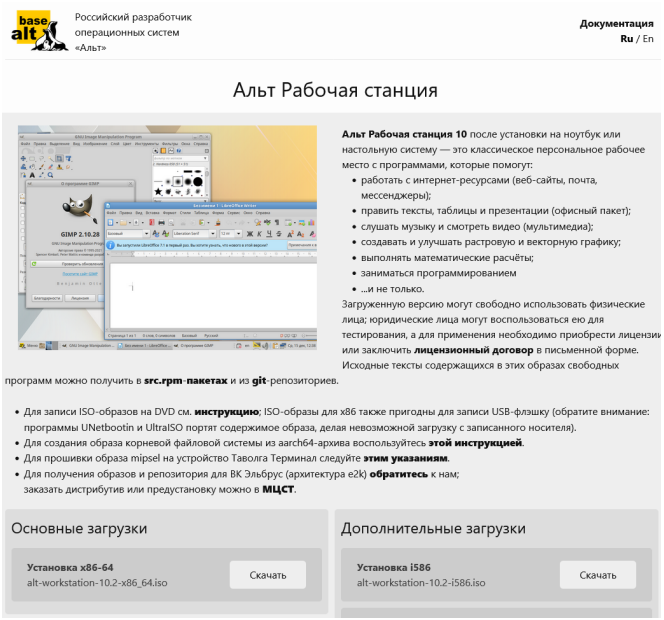


В качестве вариантов системы для выполнения лабораторных подойдут:

1. Linux Mint, дистрибутив доступен на официальном сайте по адресу https://www.linuxmint.com/



2. Alt Linux, дистрибутив доступен на официальном сайте по адресу https://getalt.org/ru/alt-workstation/

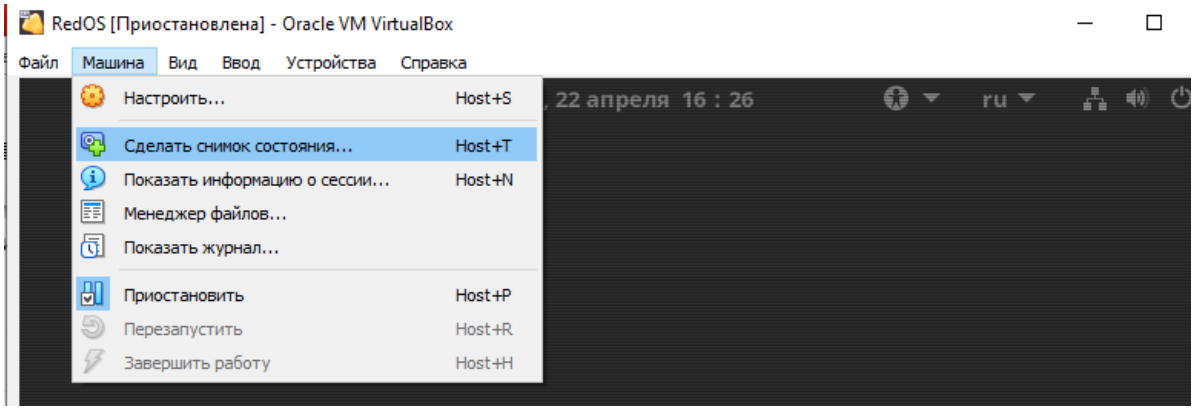


Когда операционная система будет установлена, вы получите к ней доступ через окно VirtualBox.

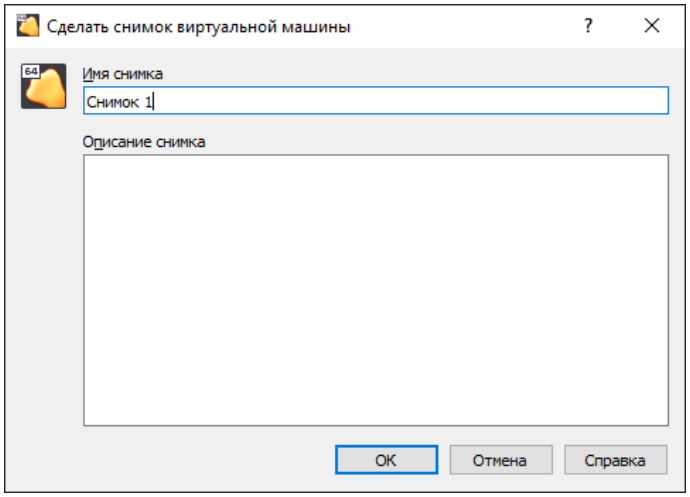
Ранее мы уже рассмотрели способ создания копии виртуальной машины, которой можно воспользоваться в случае непредвиденных проблем. Но это не единственный способ создания резервной копии – мы также можем использовать специальную функцию «Снимок состояния». Она позволяет возвращать систему к предыдущему состоянию.

Создать снимок можно следующим образом:

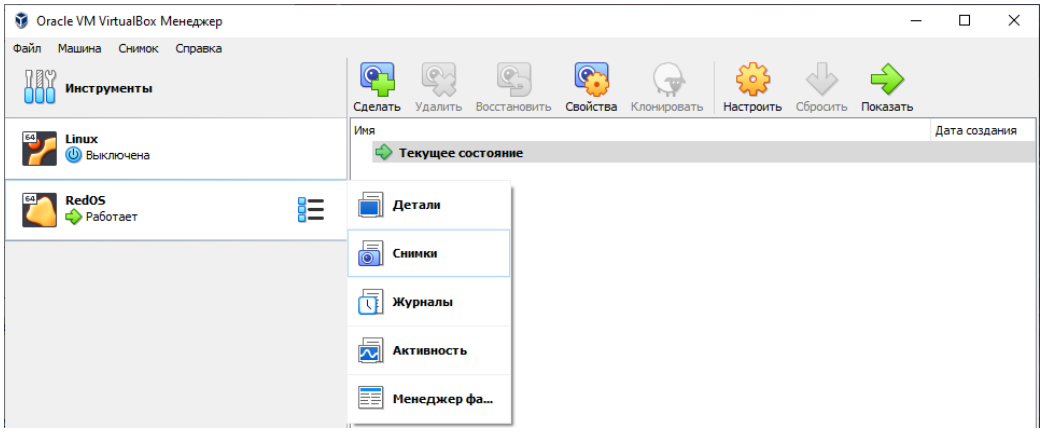
Запускаем виртуальную машину и в верхней части выбираем «Машина» -> «Сделать снимок состояния…».



Задаем ему имя и по желанию прописываем описание.



Вернуться к созданному снимку мы можем через меню «Машина» -> «Инструменты» -> «Снимки».



Пользователи и права. Работа с пользователями и структура прав

Разделение привилегий — одна из основных парадигм безопасности в операционных системах семейства Linux и Unix. Обычные пользователи работают с ограниченными привилегиями и могут влиять только на собственную рабочую среду, но не на операционную систему в целом.

Специальный пользователь с именем **root**, имеет привилегии суперпользователя. Это административная учетная запись без ограничений, действующих для обычных пользователей. Пользователи могут выполнять команды с привилегиями суперпользователя или root разными способами.

В Linux существуют служебные пользователи и пользователи с возможностью логина. Пример служебных пользователей: **daemon**, **apache**. Как правило, служебным пользователям запрещен логин в BASH.

Хорошей практикой считается запуск фоновых программ (демонов) под непривелигированным пользователем.

Суперпользователь (**root**) имеет все права на все файлы и процессы, за исключением правил, заданных расширенными атрибутами.

Пользователь может входить в несколько групп.

Основные служебные файлы: /etc/passwd, /etc/groups, /etc/shadow

Личные настройки программ и личные файлы пользователей хранятся в их домашней директории, над которой пользователь полностью властен (имеет все возможные права).

При необходимости повышения привилегий (для доступа к системным командам и файлам) необходимо использовать утилиты для повышения привилегий.

Самый простой и удобный способ получить привилегии **root** — просто войти на сервер как пользователь **root**.

Если вы входите на локальный компьютер (или используете консоль для внеполосного подключения к виртуальному серверу), введите **root** как имя пользователя в строке входа и пароль пользователя **root**, когда система его запросит.

Входить в систему как пользователь **root** обычно не рекомендуется, потому что при этом можно легко начать использовать систему не для административных задач, что довольно опасно.

Следующий способ получить привилегии суперпользователя позволяет становиться пользователем **root**, когда вам это потребуется.

Для этого можно использовать команду **su** (substitute user) для замены пользователя.

Чтобы получить привилегии **root**, введите:

```
su
```

Вам будет предложено ввести пароль для пользователя **root**, после чего будет создан сеанс оболочки **root**.

После завершения задач, для которых требуются привилегии **root**, вернитесь в обычную оболочку с помощью следующей команды:

```
exit
```

Последний способ получения привилегий **root** заключается в использовании команды **sudo**.

Команда **sudo** позволяет выполнять разовые команды с привилегиями **root** без необходимости создавать новую оболочку. Она выполняется следующим образом:

```
sudo command_to_execute
```

В отличие от **su**, для команды **sudo** требуется пароль *текущего* пользователя, а не пароль пользователя **root**.

В связи с вопросами безопасности доступ **sudo** не предоставляется пользователям по умолчанию, и его необходимо настроить перед использованием.

Основные команды для работы с пользователями

Название команды	**Описание**	**Пример**
chmod	Изменение прав на файл/директорию	chmod ug+rwx ./filename chmod o-rwx ./filename
chown	Смена владельца (юзер, группа)	chown -R username:group ./dirname/
adduser, useradd, userdel	Добавление, удаление пользователя	useradd -m username –s /bin/bash userdel username
usermod	Изменение параметров пользователя	usermod -a -G username group
sudo, su	Повышение привилегий	sudo -i su username

(!) если добавить нового пользователя в Linux Mint командой **useradd** без указания опции **-m** (создание домашнего каталога пользователя), авторизоваться по логину и паролю нового пользователя в графической оболочке будет невозможно.

(!) по умолчанию пользователь создается с командной оболочкой **sh**. Чтобы пользователь по умолчанию имел доступ к более продвинутую оболочку **bash**, нужно при добавлении пользователя к команде **useradd** добавить флаг **-s /bin/bash**.

Права доступа к файлам

Права доступа определяют, какие действия конкретный пользователь может или не может совершать с определенными файлами и каталогами.

С помощью разрешений можно создать надежную среду — такую, в которой никто не может поменять содержимое ваших документов или повредить системные файлы.

Как работают права доступа

Есть 3 вида разрешений. Они определяют права пользователя на 3 действия: чтение, запись и выполнение. В Linux эти действия обозначаются вот так:

r — read (чтение) — право просматривать содержимое файла;

w — write (запись) — право изменять содержимое файла;

x — execute (выполнение) — право запускать файл, если это программа или скрипт.

У каждого файла есть 3 группы пользователей, для которых можно устанавливать права доступа.

owner (владелец) — отдельный человек, который владеет файлом. Обычно это тот, кто создал файл, но владельцем можно сделать и кого-то другого.

group (группа) — пользователи с общими заданными правами.

others (другие) — все остальные пользователи, не относящиеся к группе и не являющиеся владельцами.

Как узнать разрешения файла

Чтобы посмотреть права доступа к файлу, нужно вызвать команду **ls** с опцией **-l**. Эта опция отвечает за вывод списка в длинном формате.

```
ls -l <путь>
```

Первый блок указывает тип файла: файл или директория (- или d)

Второй блок указывает права для текущего владельца файла/директории (user, u)

Третий блок – права для группы (group, g)

Четвертый блок – для всех остальных пользователей (others, o)

Список и описание основных цифровых и буквенных прав на файлы и директории.

Бинарная нотация	**Буквенная нотация**	**Описание**
777	rwx rwx rwx	Все могут всё. Обычно лучше не использовать.
755	rwx r-x r-x	Полные права у владельца, группа и остальные могут читать и исполнять. Подходит для общих программ.
700	rwx --- ---	Полные права у владельца, никаких прав у группы и остальных. Подходит для программ, с которыми может работать только владелец.
666	rw- rw- rw-	Все могут писать и читать.
644	rw- r-- r--	Владелец может читать и писать, группа и остальные могут только читать. Подходит для общих файлов данных/настроек.
600	rw- --- ---	Владелец может читать и писать, группа и остальные не могут ничего. Подходит для файлов данных, которые предназначены только для владельца.

Как изменить права доступа файлов

Для изменения прав доступа к файлу или каталога используется команда **chmod** (от англ. change mode). Эта команда меняет биты режима файла — если совсем просто, это индикатор разрешений.

```
chmod [разрешение] [путь]
```

Аргументы команды **chmod**, отвечающие за разрешение, состоят из 3 компонентов:

- Для кого мы меняем разрешение? Обозначается первыми буквами слов: [ugoa] — user (пользователь, он же владелец), group (группа), others (другие), all (все).
- Мы предоставляет или отзываем разрешения? Обозначается плюсом +, если предоставляем, минусом –, если отзываем.
- Какое разрешение мы хотим изменить? Чтение (r), запись (w), выполнение (x).

Прав доступа пользователей

Команда **sudo** настраивается с помощью файла, расположенного в каталоге /etc/sudoers.

Основные файлы: файл /etc/sudoers и директория /etc/sudoers.d/

В первом столбце задается группировка пользователей, которым нужно дать возможность повышения привиллегий.

Для указания группы нужно поставить знак % и указать название группы.

Для повышения полномочий без ввода пароля пользователя достаточно добавить фразу “NOPASSWD”.

Файл **sudoers** недопустимо редактировать напрямую текстовым редактором, необходимо использовать специальную программу **visudo**.

Неправильный синтаксис файла /etc/sudoers может нарушить работу системы и сделать невозможным получение повышенного уровня привилегий.

Команда **visudo** открывает текстовый редактор обычным образом, но проверяет синтаксис файла при его сохранении. Это не даст ошибкам конфигурации возможности блокировать операции **sudo**, что может быть единственным способом получить привилегии **root**.

Обычно **visudo** открывает файл /etc/sudoers в текстовом редакторе **vi**. Однако в Ubuntu команда **visudo** настроена на использование текстового редактора **nano**.

Первая строка **Defaults env_reset** сбрасывает среду терминала для удаления переменных пользователя. Эта мера безопасности используется для сброса потенциально опасных переменных среды в сеансе **sudo**.

Вторая строка, **Defaults mail_badpass**, предписывает системе отправлять уведомления о неудачных попытках ввода пароля **sudo** для настроенного пользователя **mailto**. По умолчанию это учетная запись **root**.

Третья строка, начинающаяся с “Defaults secure_path=…”, задает переменную **PATH** (места в файловой системе, где операционная система будет искать приложения), которая будет использоваться для операций **sudo**. Это предотвращает использование пользовательских путей, которые могут быть вредоносными.

Строки пользовательских привилегий

Четвертая строка, которая определяет для пользователя **root** привилегии **sudo**, отличается от предыдущих строк.

root ALL=(ALL:ALL) ALL

Первое поле показывает имя пользователя, которое правило будет применять к (**root**).

Первое “**ALL**” означает, что данное правило применяется ко всем хостам.

Второе “**ALL**” означает, что пользователь **root** может запускать команды от лица всех пользователей.

Третье “**ALL**” означает, что пользователь **root** может запускать команды от лица всех групп.

Последнее “**ALL**” означает, что данные правила применяются всем командам.

Это означает, что наш пользователь **root** сможет выполнять любые команды с помощью **sudo** после ввода пароля.

Строки групповых привилегий

Следующие две строки похожи на строки привилегий пользователя, но задают правила **sudo** для групп.

Имена, начинающиеся с **%**, означают названия групп.

Например, группа **admin** может выполнять любые команды от имени любого пользователя на любом хосте. Группа **sudo** имеет те же привилегии, но может выполнять команды от лица любой группы.

Последняя строка выглядит как комментарий:

. . .

#includedir /etc/sudoers.d

Она *действительно* начинается с символа #, который обычно обозначает комментарии. Однако данная строка означает, что файлы в каталоге **/etc/sudoers.d** также рассматриваются как источники и применяются.

Файлы в этом каталоге следуют тем же правилам, что и сам файл **/etc/sudoers**. Любой файл, который не заканчивается на ~ и не содержит символа ., также считывается и добавляется в конфигурацию **sudo**.

В основном это нужно, чтобы приложения могли изменять привилегии **sudo** после установки. Размещение всех правил в одном файле в каталоге **/etc/sudoers.d** позволяет видеть, какие привилегии присвоены определенным учетным записям, а также легко сменять учетные данные без прямого изменения файла **/etc/sudoers**.

Как и в случае с файлом **/etc/sudoers**, другие файлы в каталоге **/etc/sudoers.d** также следует редактировать с помощью команды **visudo**. Для редактирования этих файлов применяется следующий синтаксис:

```
sudo visudo -f /etc/sudoers.d/file_to_edit
```

Присвоение пользователю привилегий Sudo

Чаще всего при управлении разрешениями **sudo** используется операция предоставления новому пользователю общего доступа **sudo**. Это полезно, если вы хотите предоставить учетной записи полный административный доступ к системе.

В системе с группой администрирования общего назначения, такой как система Ubuntu, проще всего будет добавить данного пользователя в эту группу.

Например, в Ubuntu 20.04 группа **sudo** имеет полные привилегии администратора. Добавляя пользователя в эту группу, мы предоставляем ему такие же привилегии:

```
sudo usermod -aG sudo username
```

Также можно использовать команду **gpasswd**:

```
sudo gpasswd -a username sudo
```

Обе команды выполняют одно и то же.

Создание псевдонимов

Файл **sudoers** можно организовать более эффективно, группируя элементы с помощью разнообразных псевдонимов.

Например, мы можем создать три разных группы пользователей с некоторыми общими участниками:

```
. . .
User_Alias    GROUPONE = abby, brent, carl
User_Alias    GROUPTWO = brent, doris, eric,
User_Alias    GROUPTHREE = doris, felicia, grant
. . .
```

Имена групп должны начинаться с заглавной буквы. Затем мы можем дать участникам группы **GROUPTWO** разрешение на обновление базы данных **apt**, создав следующее правило:

```
. . .
GROUPTWO          ALL = /usr/bin/apt-get update
. . .
```

Если мы не укажем пользователя или группу для запуска, команда **sudo** по умолчанию использует пользователя **root**.

Мы можем дать членам группы **GROUPTHREE** разрешение на выключение и перезагрузку системы, создав псевдоним команды и используя его в правиле для **GROUPTHREE**:

```
. . .
Cmnd_Alias  POWER = /sbin/shutdown, /sbin/halt, /sbin/reboot, /sbin/restart
GROUPTHREE      ALL = POWER
. . .
```

Создадим псевдоним команды с именем **POWER**, который будет содержать команды выключения и перезагрузки системы. Затем мы дадим членам группы **GROUPTHREE** разрешение на выполнение этих команд.

Также мы можем создать псевдонимы запуска от имени, которые могут заменять часть правила, где указывается, от имени какого пользователя следует выполнить команду:

```
. . .
Runas_Alias          WEB = www-data, apache
GROUPONEALL = (WEB) ALL
. . .
```

Это позволит любому участнику группы **GROUPONE** выполнять команды от имени пользователя **www-data** или пользователя **apache**.

Необходимо помнить, что в случае конфликта правил более поздние правила имеют приоритет перед более ранними.

Фиксация правил

Есть ряд способов, которые позволяют более точно контролировать реакцию **sudo** на вызов.

Команда **updatedb**, связанная с пакетом **mlocate**, относительно безобидна при ее выполнении в системе с одним пользователем. Если мы хотим разрешить пользователям выполнять ее с привилегиями **root** *без* ввода пароля, мы можем создать правило следующего вида:

```
. . .
GROUPONEALL = NOPASSWD: /usr/bin/updatedb
. . .
```

NOPASSWD — это свойство, означающее, что пароль не запрашивается. У него есть сопутствующее свойство **PASSWD**, которое используется по умолчанию и требует ввода пароля. Данное свойство актуальной для остальной части строки, если его действие не переопределяется дублирующим тегом в этой же строке.

Например, мы можем использовать следующую строку:

. . .

```
GROUPTWO        ALL = NOPASSWD: /usr/bin/updatedb, PASSWD: /bin/kill
```

. . .

Также полезно свойство **NOEXEC**, которое можно использовать для предотвращения опасного поведения некоторых программ.

Например, некоторые программы, такие как **less**, могут активировать другие команды, вводя их через свой интерфейс:

```
!command_to_run
```

При этом все команды пользователя выполняются с теми же разрешениями, что и команда **less**, что может быть довольно опасно.

Чтобы ограничить такое поведение, мы можем использовать следующую строку:

. . .

```
username     ALL = NOEXEC: /usr/bin/less
```

. . .

Настройка портов виртуальной машины для SSH-подключения

Рассмотрим возможность программы VirtualBox пробрасывать порты в гостевую операционную систему, для того чтобы иметь доступ к ней с хоста, т.е. с реальной машины, например, в тех случаях, когда подключение (сеть) в гостевой ОС в VirtualBox работает в режиме «NAT».

Данная возможность будет полезна, например, когда Вам нужен одновременно и доступ к Интернету в гостевой операционной системе, и доступ к каким-нибудь сервисам в гостевой ОС с хоста.

Доступ к Интернету в гостевой ОС можно получить с помощью выбора типа подключения «NAT», но как Вы, наверное, знаете, при этом теряется доступ к сервисам гостевой ОС с реального Вашего компьютера. Данную проблему как раз и решает проброс портов.

Например, Вы не можете подключиться к виртуальной машине по SSH со своего компьютера, или обратиться к Web серверу, или даже просто скопировать команду и вставить в консоль. Поэтому пробросим порт в гостевую ОС, например, для того чтобы подключится к ней по SSH всем известной программой PuTTY. С помощью нее мы сможем без проблем управлять сервером и в случае необходимости копировать команды в нее.

PuTTY – это бесплатная клиентская программа для удаленного подключения, например к серверам по протоколам SSH, Telnet и другим. Она позволяет управлять

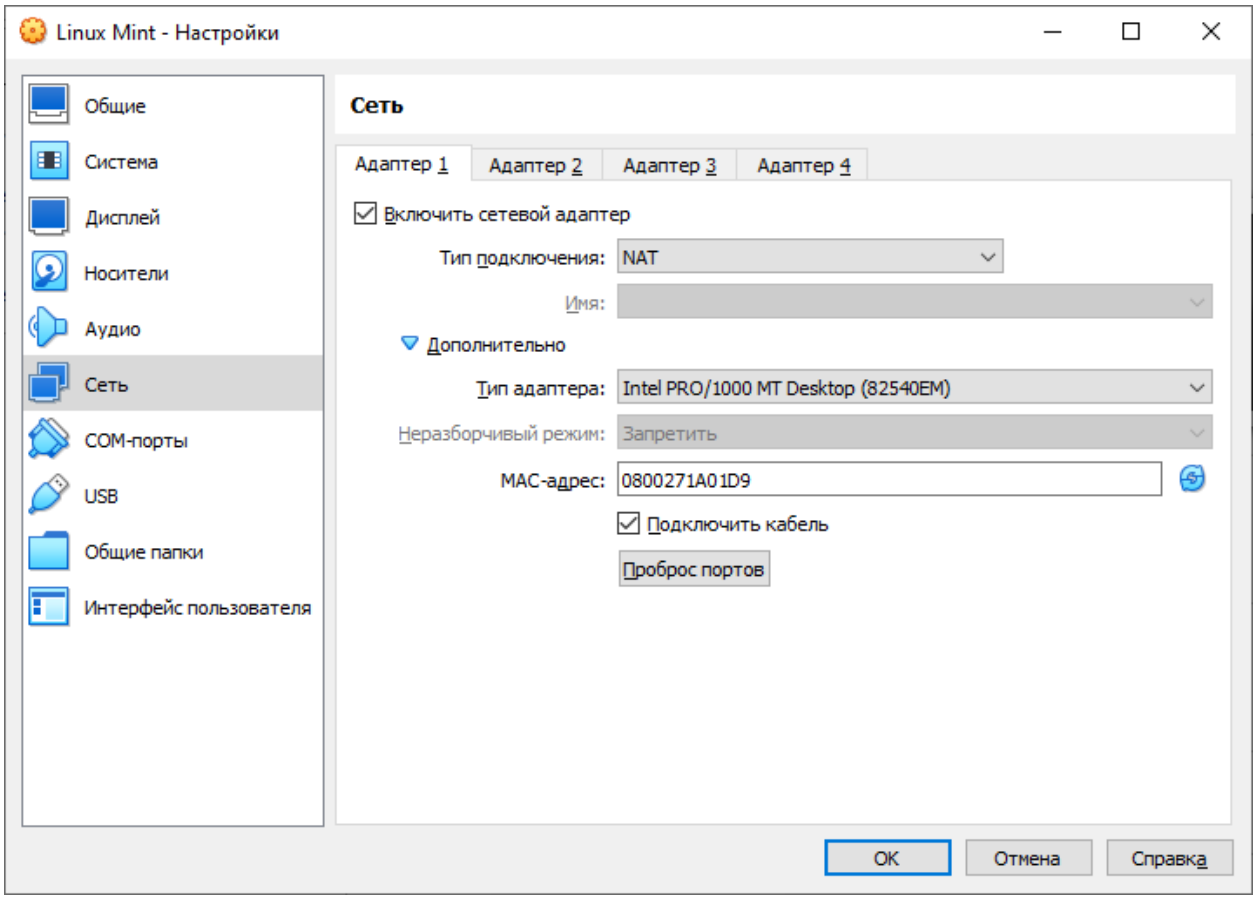
удаленным компьютером. PuTTY — это клиентская часть, серверная должна быть реализована на удаленной стороне, должен быть установлен SSH сервер. Скачать PuTTY можно с официального сайта — www.putty.org.

Настройка проброса портов в VirtualBox для SSH

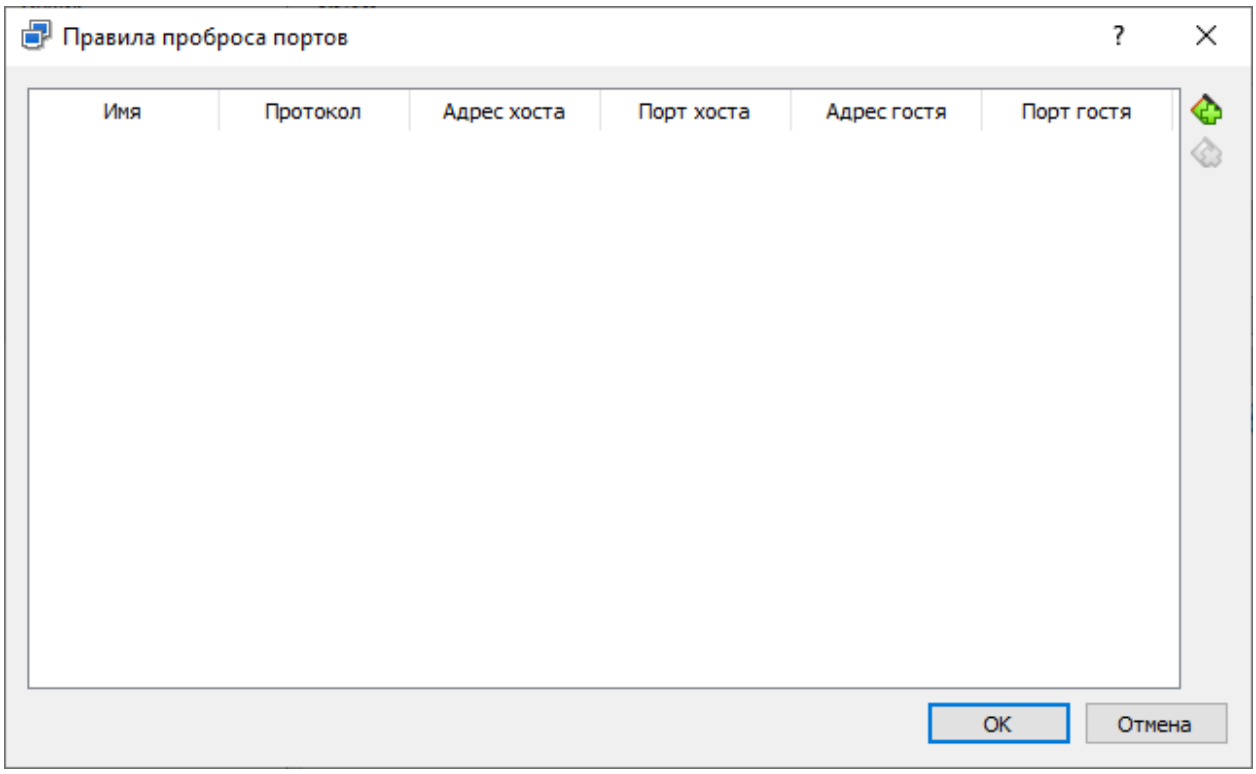
SSH – сервер обычно прослушивает 22 порт, поэтому нам необходимо пробросить подключение именно на 22 порт.

Для того чтобы пробросить порт, запускаем VirtualBox и заходим в настройки нашей виртуальной машины. Затем переходим в раздел настроек «Сеть», и открываем вкладку с включенным адаптером. Тип подключения - «NAT» - для доступа в Интернет.

Далее щелкаем на пункт «Дополнительно», чтобы отобразить дополнительные настройки данного адаптера, и после этого нажимаем на кнопку «Проброс портов».



В итоге у нас откроется окно «Правила проброса портов». Для добавления нового правила нажимаем на иконку с плюсиком.



На данном этапе Вы уже должны знать IP адрес гостевой операционной системы и порт, который прослушивает SSH сервер (по умолчанию это 22 порт). IP адрес можно узнать с помощью команды ifconfig (по умолчанию это 10.0.2.15).

Описание колонок таблицы с правилами:

Имя – название правила, в нашем случае разумно назвать SSH;

Протокол – протокол, по которому будет происходить взаимодействие. В обычных случаях это – TCP;

IP хоста – IP адрес Вашего реального компьютера, можно указать 127.0.0.1 или оставить данное поле пустым;

Порт хоста – любой свободный порт компьютера, который будет использоваться для перенаправления на нужный порт в гостевой ОС, например 2222;

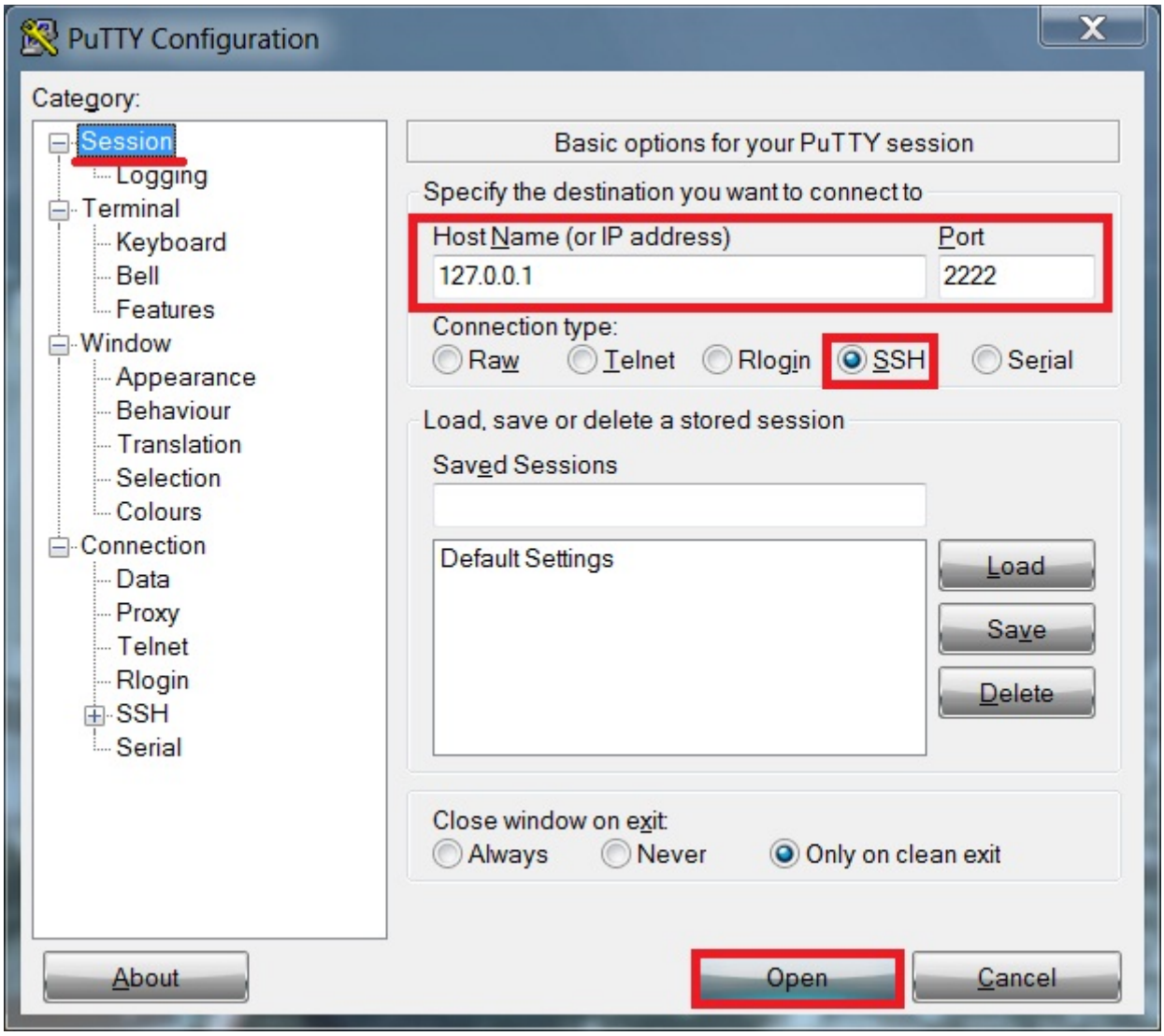
IP гостя – здесь указываем IP адрес гостевой операционной системы, на который будет происходить перенаправление, по умолчанию это 10.0.2.15;

Порт гостя – порт гостевой ОС, на который нам необходимо пробрасывать наши запросы. В нашем случае это 22 порт, который прослушивает SSH сервер.

После заполнения таблицы с правилом, нажимаем «ОК».

На этом настройка проброса портов закончена, теперь мы можем проверить работу данного правила (проброса). Для этого сначала запускаем виртуальную машину, проверяем наличие установленного пакета **openssh-server** (команда в терминале **apt install openssh-server**), а затем запускаем программу **PuTTY**. На вкладке Session выбираем тип соединения SSH и в поле Host Name пишем локальный адрес компьютера

(хоста), которой мы указывали в колонке «IP хоста» в правилах проброса портов, если ничего не указывали, то пишите 127.0.0.1. В поле порт указываем номер порта, который мы указывали в колонке «Порт хоста», т.е. 2222. Нажимаем кнопку «Open».



Если после этого Вы получили приглашение на ввод логина, то это значит, что подключение работает, и Вы, соответственно, можете вводить учетные данные от гостевой операционной системы.

Альтернативный вариант подключения: OpenSSH

В Windows 10 и Windows Server 2019 появился **встроенный SSH клиент**, который вы можете использовать для подключения к *Nix серверам, ESXi хостам и другим устройствам по защищенному протоколу, вместо Putty, MTPuTTY или других сторонних SSH клиентов. Встроенный SSH клиент Windows основан на порте **OpenSSH** и предустановлен в ОС, начиная с Windows 10 1809.

Установка клиента OpenSSH в Windows 10

Клиент OpenSSH входит в состав Features on Demand Windows 10 (как и RSAT). Клиент SSH установлен по умолчанию в Windows Server 2019 и Windows 10 1809 и более новых билдах.

Проверить, что SSH клиент установлен, можно следующей командой:

```
Get-WindowsCapability -Online | ? Name -like 'OpenSSH.Client*'
```

```
PS C:\WINDOWS\system32> Get-WindowsCapability -Online | ? Name -like 'OpenSSH.Client*'

Name  : OpenSSH.Client~~~~0.0.1.0
State : Installed
```

В нашем примере клиент OpenSSH установлен (статус: **State: Installed**).

Если SSH клиент отсутствует (**State: Not Present**), его можно установить:

С помощью команды PowerShell:

```
Add-WindowsCapability -Online -Name OpenSSH.Client*
```

С помощью DISM:

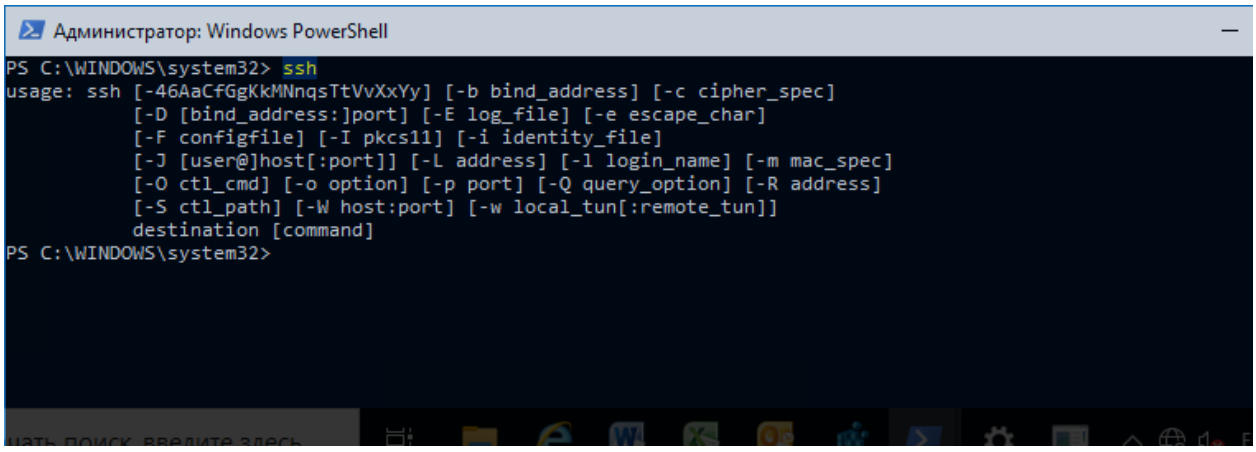
```
dism /Online /Add-Capability /CapabilityName:OpenSSH.Client~~~~0.0.1.0
```

Через «Параметры» - «Приложения» - «Дополнительные возможности» - «Добавить компонент». Найдите в списке **Клиент OpenSSH** и нажмите кнопку «Установить».

Как использовать SSH клиент в Windows 10?

Чтобы запустить SSH клиент, запустите командную строку PowerShell или cmd.exe. Выведите доступные параметры и синтаксис утилиты ssh.exe, набрав команду:

```
ssh
```



Для подключения к удаленному серверу по SSH используется команда:

```
ssh username@ip
```

Если SSH сервер запущен на нестандартном порту, отличном от TCP/22, можно указать номер порта:

```
ssh username@ip -p port
```

Например, чтобы подключиться к Linux хосту с IP адресом 192.168.1.202 под root, выполните:

```
ssh root@192.168.1.202
```

При первом подключении появится запрос на добавление ключа хоста в доверенные, наберите yes -> Enter (при этом отпечаток ключа хоста добавляется в файл **C:\Users\username\.ssh\known_hosts**).

Создание SSH-ключа

SSH-ключи используются для авторизации на сервере SSH без использования пароля в целях безопасности и удобства. Хорошим подходом является полный запрет авторизации на SSH-сервере по паролю и использование исключительно SSH-ключей, созданных для индивидуальных пользователей. У ключа есть открытая (хранится на сервере SSH) и закрытая (хранится на компьютере пользователя) части. Эти части называются «парой ключей».

1. **Сгенерировать ключ.** Выполнить команду «ssh-keygen -t rsa». При возникновении сообщения «Enter passphrase» допустимо не задавать никакую ключевую фразу.
2. **Указать место создания ключа.** При возникновении сообщения «Enter file in which to save the key» допустимо либо указать предпочтительный файл, в который сгенерируется ключ, либо оставить по умолчанию — «/home/username/.ssh/id_rsa»
3. **Ключ будет сгенерирован.** Закрытая часть будет расположена по умолчанию в файле id_rsa в указанной выше директории, а открытая — в файле id_rsa.pub
4. **Скопировать пару в нужные места.** Открытая часть копируется в файл /home/username/.ssh/authorized_keys, закрытая — на свой локальный компьютер (для копирования файла на локальный компьютер можно использовать SCP или PSCP, которые рассмотрены далее).

Указание закрытого SSH-ключа в клиенте (на локальном компьютере)

При использовании openssh команда выглядит следующим образом:

```
ssh –i /home/username/.ssh/id_rsa username@ip –p port,
```

где username – имя пользователя на сервере, ip – адрес подключения (127.0.0.1 в случае виртуальной машины, port – номер порта для подключения).

Или, как пример для пути в Windows:

```
ssh username@ip -i "C:\Users\username\.ssh\id_rsa"
```

Передача файлов между локальным компьютером и виртуальной машиной

SCP: копирование файлов из/в Windows через SSH

С помощью утилиты **scp.exe**, которая входит в состав пакета клиента SSH, вы можете скопировать файл с вашего компьютера на SSH сервер:

```
scp.exe "E:\docs\report.txt" username@ip:/home
```

Можно рекурсивно скопировать все содержимое каталога:

```
scp.exe -r E:\docs\ username@ip:/home
```

И наоборот, вы можете скопировать файл с удаленного сервера на ваш компьютер:

```
scp.exe username@ip:/home/report e:\tmp
```

Если вы настроите аутентификацию по RSA ключам, то при копировании файлов не будет появляться запрос на ввод пароля для подключения к SSH серверу. Это удобно, когда вам нужно настроить автоматическое копирование файлов по расписанию.

(!) В утилите **scp.exe** номер порта для подключения, в отличии от команды ssh, указывается через опцию -P, а не -p.

Альтернативный способ передачи файлов: команда pscp

pscp – это реализация протокола SCP, в которой мы можем безопасно передавать и копировать файлы и папки по сети с помощью соединения SSH.

Утилиту pscp можно скачать по следующей ссылке: https://putty.org.ru/download.html

pscp может быть установлен в автономном режиме или с пакетом установки putty.

Использование pscp.exe

1. Скачать утилиту pscp.exe в произвольную директорию на компьютере администратора, например, C:\Putty.

2. На компьютере администратора запустить консоль и перейти в директорию установки Putty командой:

```
cd C:\Putty
```

3. Запустить утилиту pscp.exe, например, со следующими параметрами:

```
pscp E:\test.cer username@ip:/home
```

где

E:\test.cer – путь до файла на компьютере администратора, который необходимо доставить на удаленный сервер

ip – адрес интерфейса удаленного сервера

/home – директория на сервере, в которую будет скопирован файл

4. Появится запрос на ввод пароля пользователя root. Ввести пароль и нажать Enter.

username@ip's password:

Скачивание файла с удаленного сервера и дополнительные параметры команды PSCP производятся аналогично SCP.

Работа с файловой системой Linux

BASH имеет встроенные команды, а также способен запускать внешние программы.

Тип	**Название команды**	**Описание**
Операции с файлами	touch, cp, mv, rm	Создание, копирование, перемещение, удаление
Операции с директориями	mkdir, rmdir	Создание, удаление директории
	cd, pwd, ls	Смена директории, вывод текущего пути, листинг директории
Вывод файлов	cat, tail, less	Вывод, вывод последних строк, вывод первых строк
Операции с процессами	ps, kill	Вывод списка процессов, завершение процесса
Операции с правами	chown, chmod	Смена владельца, смена прав доступа файла/директории
Поиск	grep	Поиск строки в файлах
Информация	man	Вывод справочной страницы о конкретной программе
	echo	Вывод текстовой строки

Основные текстовые редакторы в Linux

Название ПО	**Тип**	**Особенности**
Vim	Консольный, графический	Огромные возможности, много готовых настроек, однако требует времени на освоение
Nano	Консольный	Простой и понятный, однако не хватает продвинутых возможностей
Emacs	Консольный, графический	Огромные возможности, много готовых настроек, однако требует времени на освоение
Mcedit	Консольный	Простой и понятный, однако продвинутым инженерам может не хватать возможностей
Atom	Графический	Быстро развивается, много плагинов, однако для простого редактирования не подходит

IM создан на основе классического UNIX-редактора Vi.

Сущестует много версий VIM, в т.ч. с графическим интерфейсом.

IM имеет 4 режима работы: нормальный, режим вставки, режим командной строки, визуальный режим.

Конвейер – механизм, позволяющий перенаправить вывод одной команды на вход другой и составлять цепочки выполнения команд.

Пример:

cat ./file | grep test

Циклы – последовательное выполнение команд с заданным количеством шагов.

Пример:

for i in {1..10}; do

echo $i

done

Переменные окружения – заданные переменные, к которым имеют доступ запускаемые программы.

Пример:

echo $PATH

Поиск по истории команд BASH можно осуществлять не только поиском по файлу ~/.bash_history, но и с помощью контекстного поиска, нажав на “ctrl+r” и начав вводить начало искомой строки

Циклы можно использовать не только в скриптах, но и непосредственно в командной строке: for i in {1..10}; do echo $i; done

В данном случае функцию переноса строки выполняет знак “;”

Строку с командой в командной строке можно закомментировать точно так же, как и в файле скрипта, поставив “#”. Далее к ней можно вернуться поиском.

Чтобы выполняемая команда не сохранилась в ~/.bash_history достаточно прямо перед ней поставить один знак пробела.

Для продуктивной работы с выводом на экран одновременно нескольких bash сессий принято использовать терминальные мультиплексоры: tmux, terminator и т.п.

Чтобы длительная команда не прервалась при отключении пользователя от bash сессии используют программу “screen”.

Выполняемую команду можно отправить в фоновый режим нажатием “ctrl+z”, а вернуть – набрав fg (перемещение фонового процесса в активный).

«Всё есть файл» - концепция построения иерархии ФС и работы в UNIX-подобных ОС, согласно которой работа с системой сводится к работе с файлами.

Иерархия файловой системы

FHS – Filesystem Hierarchy Standard, стандарт иерархии файловой системы, унифицирующий расположение и назначение файлов и структуры директорий в Linux.

/boot – файлы, необходимые для загрузки ОС: ядро ОС, конфигурационные файлы и модули

/dev – файлы всех устройств в ОС

/etc – конфигурационные файлы программ и системных модулей

/home – домашние директории пользователей (кроме суперпользователя)

/opt – директория для других программ. Например, не поставляемых в составе дистрибутива

/root – домашняя директория суперпользователя (root)

/mnt – директория для монтирования других ФС

/proc, /sys – “виртуальные” ФС для хранения переменных системы/ядра ОС

/usr/bin – двоичные файлы приложений

/usr/lib – библиотеки приложений

/var/log – логи приложений и системы

Основные конфигурационные и информационные файлы Linux:

/etc/fstab – настраивает список монтируемых ФС и опции монтирования

/ets/resolv.conf – список используемых DNS-серверов

/etc/hosts – список соответствий DNS-имен и IP-адресов

/var/log/messages, /var/log/syslog – основные системные журналы (syslog), расположение отличается в разных дистрибутивах

/etc/crontab, /etc/cron.d/ – файл и директория с заданиями планировщика cron

/etc/passwd, /etc/groups – список пользователей и групп в Linux

/etc/sudoers, /etc/sudoers.d/ – список правил повышения полномочий пользователями и группами

/etc/default/grub – настроечный файл для загрузчика GRUB

/home/username/.bashrc – конфигурационный файл BASH для конкретного пользователя

/home/username/.bash_history – список последних команд, выполненных интерпретатором BASH от конкретного пользователя

Примеры файловых систем, используемых в Linux:

Название ФС	**Где используется**	**Особенности**
ext2	Накопители с ограниченными циклами записи	Нет журнала операций
ext3, ext4	Системные разделы, общего назначения	Используются по умолчанию во множестве дистрибутивов
xfs	Файловые хранилища	Эффективная работа с большими файлами, не уменьшается
fat	Загрузочные разделы, флеш накопители	Поддерживается огромным количеством устройств
reiserfs	Системные разделы	Эффективная работа со множеством мелких файлов, слабо развивается

/boot содержит данные, необходимые для загрузки ОС

swap содержит раздел подкачки и используется при исчерпании оперативной памяти

/ содержит всю иерархию ФС

/home содержит домашние каталоги пользователей

/mnt/storage содержит подключаемую «внешнюю» ФС

Основные команды для работы с ФС

Название команды	**Описание**	**Пример**
df	Вывод информации об утилизации подключенных ФС	df -h
du	Подсчет размера файлов и директорий	du -ha /var/log/
resize2fs	Изменение размера ФС ext2/ext3/ext4	resize2fs /dev/sda
fdisk	Просмотр информации о разделах дисков	fdisk /dev/sda
fsck	Запуск проверки ФС на целостность	fsck /dev/sda
mkfs	Создание ФС на диске	mkfs.ext4 /dev/sda
mount, umount	Монтирование, отмонтирование ФС	mount /dev/sda /mnt/storage

Самые часто встречающиеся программы для работы с текстом и примеры их использования:

Название команды	**Описание**	**Пример**
grep	Pattern matching	grep -oE "UUID=(\w+-?\w+)+\s" /etc/fstab
sed	Преобразование текста	cat /etc/fstab | sed 's/ext4/ext3/g'
awk	Язык поиска и обработки текста	cat /etc/mtab | awk '{print $1" "$2}'
sort, uniq	Сортировки и проверки на уникальность	sudo fdisk -l | grep "Disk \/" | sort
cut	Разбивка по секциям текста	cat /etc/mtab | cut -d ' ' -f1,2

head, tail, less	Вывод текста	
wc	Подсчет строк	fdisk -l | wc -l

Группировка – с помощью () можно сгруппировать регулярные выражения в один логический блок и затем применять к нему общие правила.

Последовательность – с помощью [] можно задать последовательность символов. Например, [a-z] соответствует английскому алфавиту (в нижнем регистре!).

Нестрочные символы – указывают на нестрочные символы в строке.

Например, начало, конец строки.

Специальные правила – указывают на общие правила обработки строки. Например, \d соответствует правилу “любая цифра”.

Перечисления – с помощью знаков “*, +, ?, {}” можно указать количество

указанного регулярного выражения.

Список сокращений

ВМ – виртуальная машина.

ФС – файловая система.

ОС – операционная система.

LVM - logical volume manager.

Задание на лабораторную работу

Цель: научиться использовать инструменты для работы с пользователями и правами. Научиться подключаться к удаленному серверу по протоколу SSH. Научиться работать с базовыми командами и скриптами. Научиться использовать базовые инструменты обработки текста в Linux.

1. **Создать на виртуальной машины нового пользователя для себя.** С помощью стандартных методов создать пользователя (даже если устанавливали Linux на свою виртуальную машину и создавали пользователя при установке системы, показать добавление нового пользователя). Имя пользователя задать в виде – *ФамилияИО* (ваши фамилия и инициалы).

2. **Дать новому пользователю возможность повышать привилегии.** Через добавление в sudoers. Затем перезайти под новым пользователем, убедиться, что права администратора системы есть (например, через возможность устанавливать программные пакеты или просматривать содержимое системных каталогов). *Все дальнейшие шаги делать через нового пользователя.*

3. **Запустить SSH-клиент.** В случае использования графического клиента запустить программу. В случае использования терминальных клиентов — открыть терминал.

4. **Подключиться по SSH к своей виртуальной машине.** В графическом клиенте произвести необходимые настройки. В терминальном — выполнить команду подключения. Например, для openssh: ssh username@ip_address –p port_number

5. **Создать пару SSH-ключей с опциями по умолчанию.** Пользуясь разобранным в методических указаниях алгоритмом создать пару ключей, не изменяя значения по умолчанию.

6. **Разместить пару ключей в нужных местах.** Пользуясь разобранным в методических указаниях алгоритмом разместить пару ключей в нужных местах (*на виртуальной машине и вашей основной машине*) для использования в авторизации.

7. **Отключиться от учетной записи в ОС на виртуальной машине и авторизоваться по ключу из вашей основной ОС.**

8. **Создать файл script.sh и сделать его исполняемым.** С помощью команд «touch» и «chmod +x» создать в своей домашней директории файл скрипта и дать ему права на исполнение.

9. **Открыть файл скрипта любым редактором.** Выбор редактора индивидуален и зависит от преследуемых целей и личных предпочтений.

10. **Придумать и написать алгоритм создания дерева каталогов.** С помощью известных логических конструкций или команд написать в скрипте алгоритм создания дерева каталогов вида /home/username/1/2/3.

11. **Запустить скрипт.** Выполнить свой скрипт и получить нужный результат.

12. **Просмотреть список монтируемых ФС.** Посмотреть содержимое файла «/etc/fstab», сделать вывод о том, какие ФС монтируются при загрузке и какой тип они имеют, для чего используются.

13. **Просмотреть утилизацию примонтированных ФС.** С помощью команды «df» просмотреть общий и свободный объем ФС, примонтированных в даный момент к ОС.

14. **Посчитать объем файлов в домашней директории.** С помощью команды «du» посчитать объем всех файлов, расположенных в своей домашней директории.

15. **Поиск текста по регулярным выражениям.** Найти в главном файле Syslog все сообщения за (определенный период времени).

Результат: подключились к своей ВМ, создали нового пользователя, дали возможность авторизоваться по SSH. Создали ключ и авторизовались по нему. Написали скрипт, создающий дерево каталогов. Посмотрели, какие ФС используются в ОС и узнали их объем. Посмотрели, как работают системные программы, рассмотрели базовые случаи.

ТРЕБОВАНИЯ К ОТЧЕТУ

Отчет должен содержать следующие разделы:

1. Титульный лист, оформленный согласно утвержденному образцу.
2. Цели выполняемой лабораторной работы.
3. Задание на лабораторную работу.
4. Описание процесса выполнения работы: для каждого действия, производимого в командной строке, в отчет следует включить:
   - краткое описание действия;
   - вводимая команда или команды;
   - реакция системы на ввод команд (если объем выводимых данных превышает несколько строк, всю информацию включать в отчет не следует).
5. Выводы.