


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

Установка и настройка Virtual Box

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

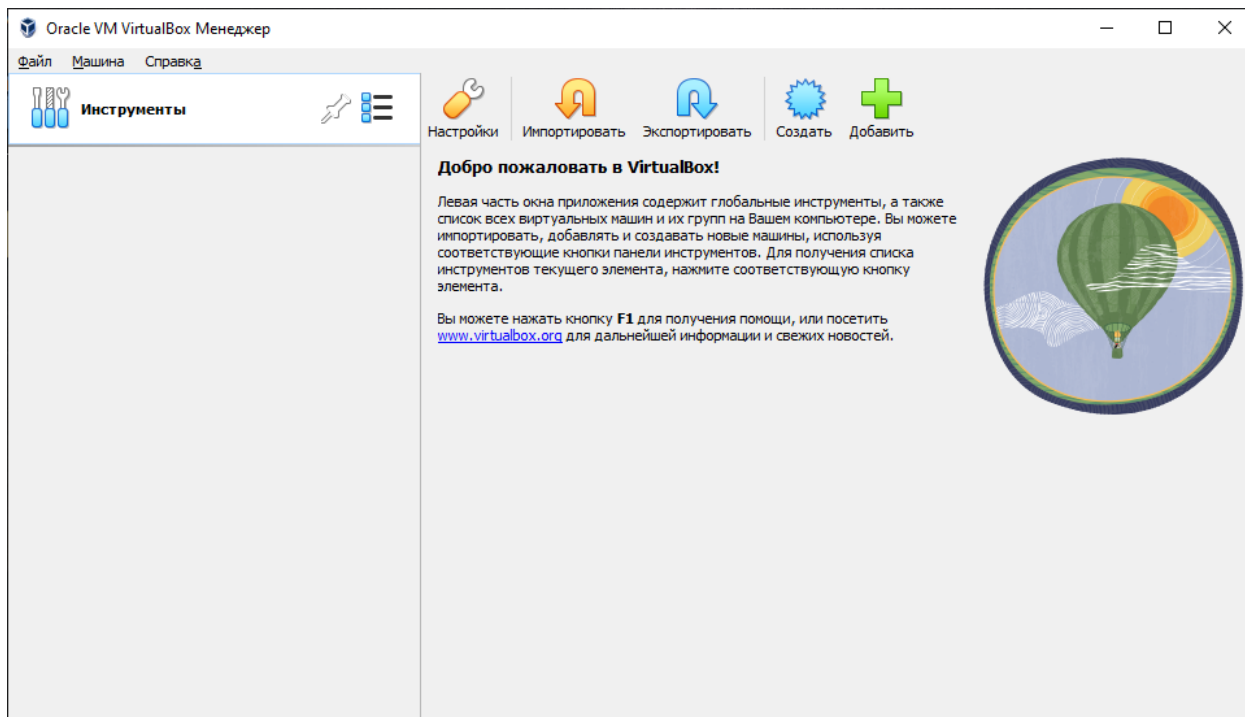
VirtualBox – это специальное средство для виртуализации, позволяющее запускать операционную систему внутри другой. Оно поставляется в двух версиях – с открытым и закрытым исходным кодом. С помощью VirtualBox мы можем не только запускать ОС, но и настраивать сеть, обмениваться файлами и делать многое 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): <https://www.virtualbox.org/wiki/Downloads>



The screenshot shows the VirtualBox website's download page. On the left is a navigation menu with links: About, Screenshots, Downloads, Documentation (with sub-links for End-user docs and Technical docs), Contribute, and Community. The main content area features the VirtualBox logo, a heading 'Download VirtualBox', and a sub-heading 'VirtualBox binaries'. Below this, there is a paragraph stating that by downloading, users agree to the terms and conditions of the respective license. A note mentions that for the latest VirtualBox 6.1 packages, users should see 'VirtualBox 6.1 builds', and that version 6.1 will remain supported until December 2023. A section titled 'VirtualBox 7.0.8 platform packages' lists several download links for different operating systems: Windows hosts, macOS / Intel hosts, Developer preview for macOS / Arm64 (M1/M2) hosts, Linux distributions, Solaris hosts, and Solaris 11 IPS hosts. At the bottom, it states that the binaries are released under the terms of the GPL version 3.

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

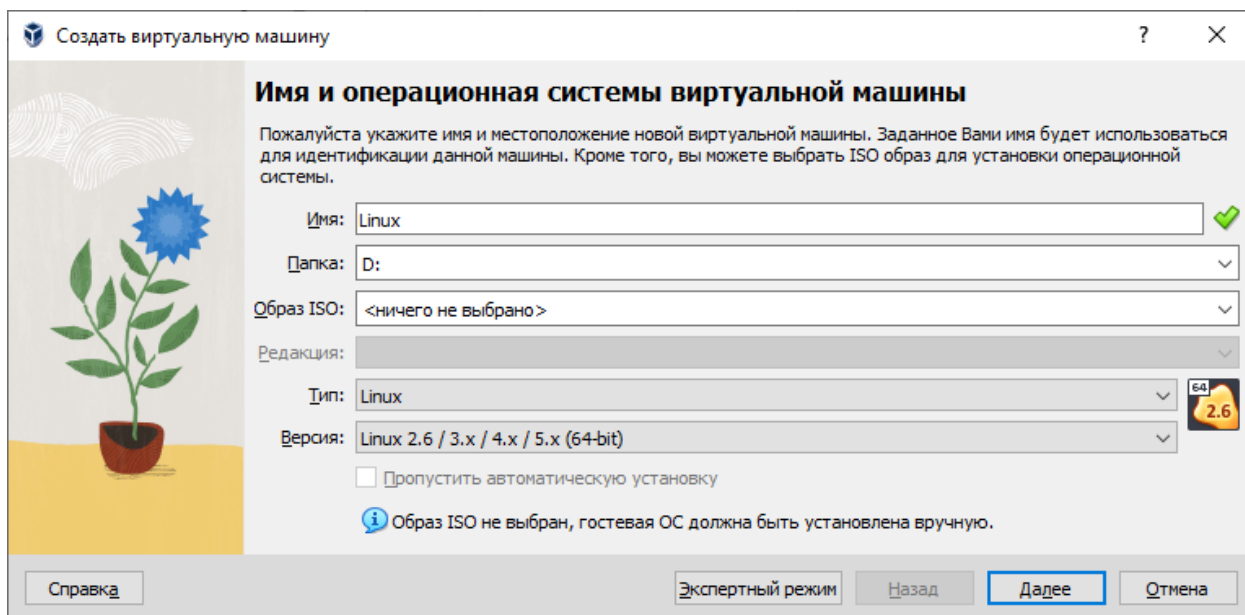


Рассмотрим, как создать виртуальную машину и провести дополнительные настройки.

Создаем виртуальную машину

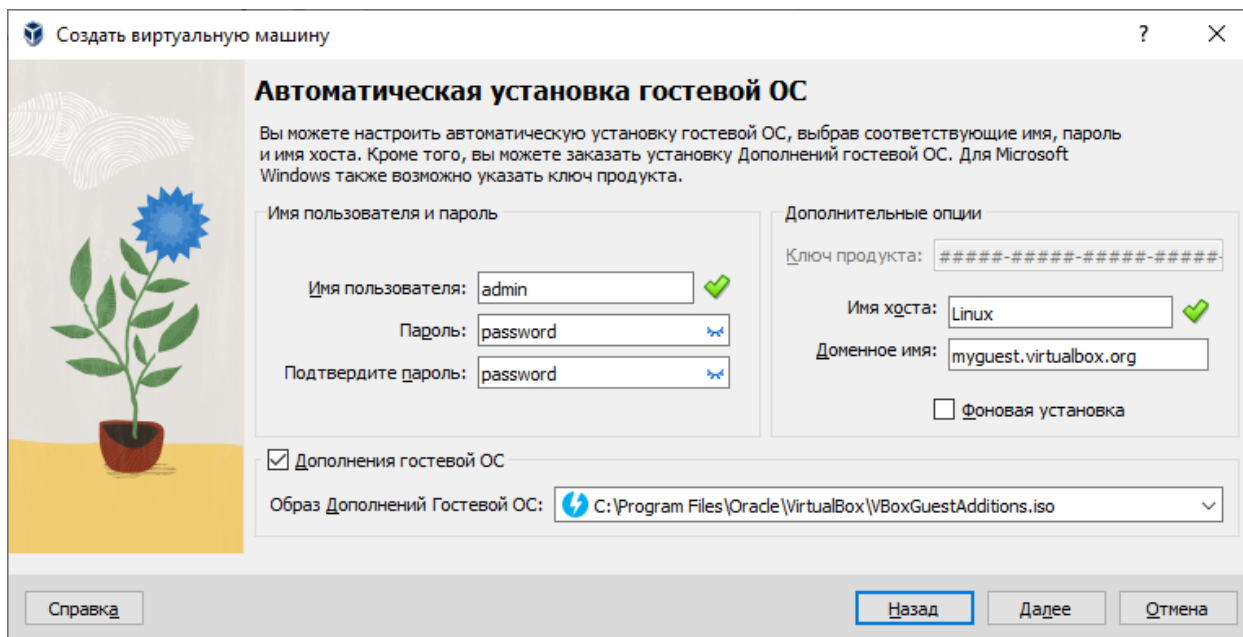
Основная функция VirtualBox – виртуализация. Чтобы запустить новую операционную систему, необходимо создать для нее виртуальную машину. Для этого необходимо выполнить следующее:

Запускаем VirtualBox и в правой части выбираем «Создать».

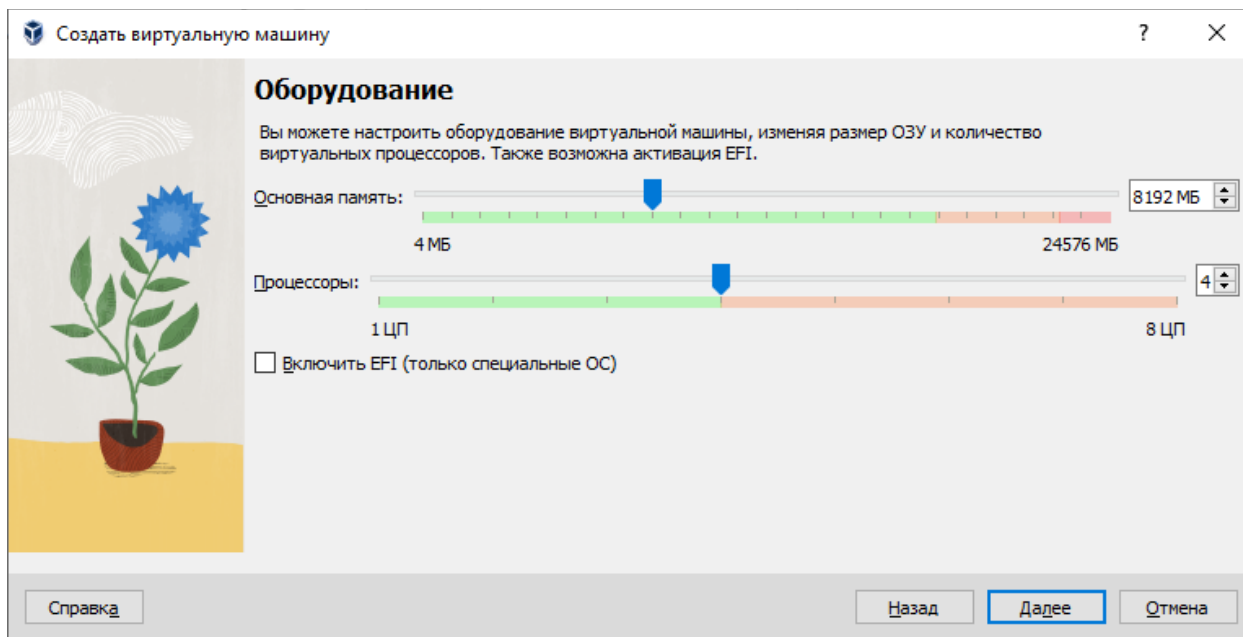


В отобразившемся окне прописываем имя операционной системы и указываем путь до машины. Обратите внимание на то, что тип ОС выбирается автоматически в зависимости от введенного названия. Также здесь можно сразу указать путь к образу

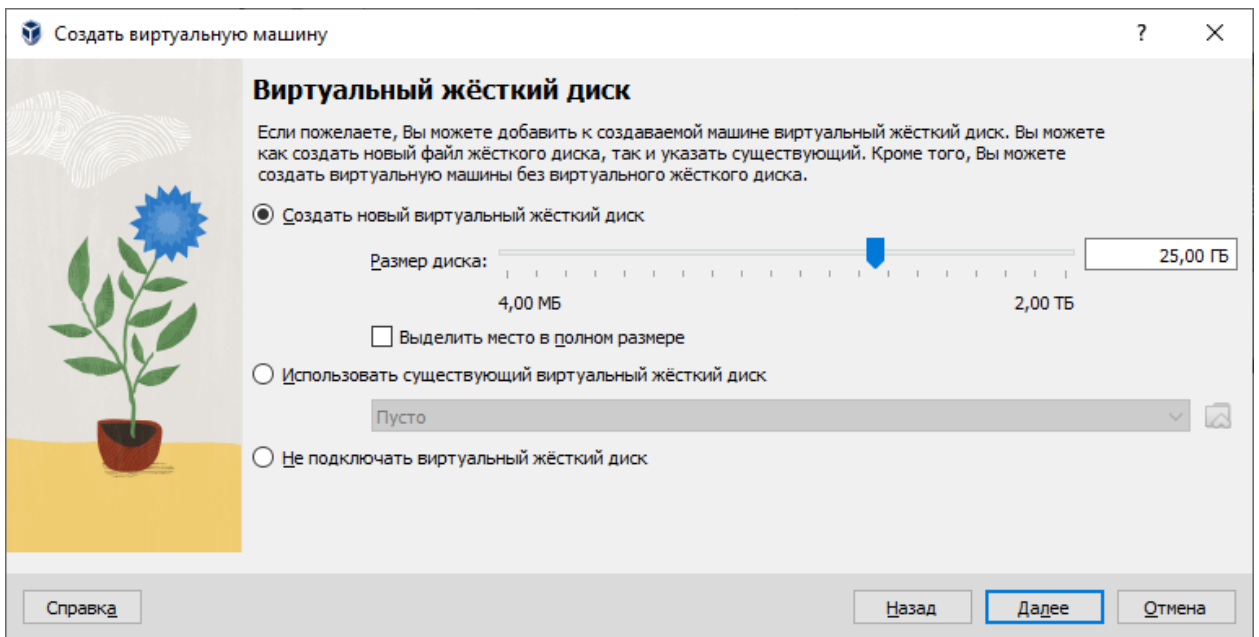
установочного диска ОС, тогда следующим пунктом будут параметры автоматической установки.



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



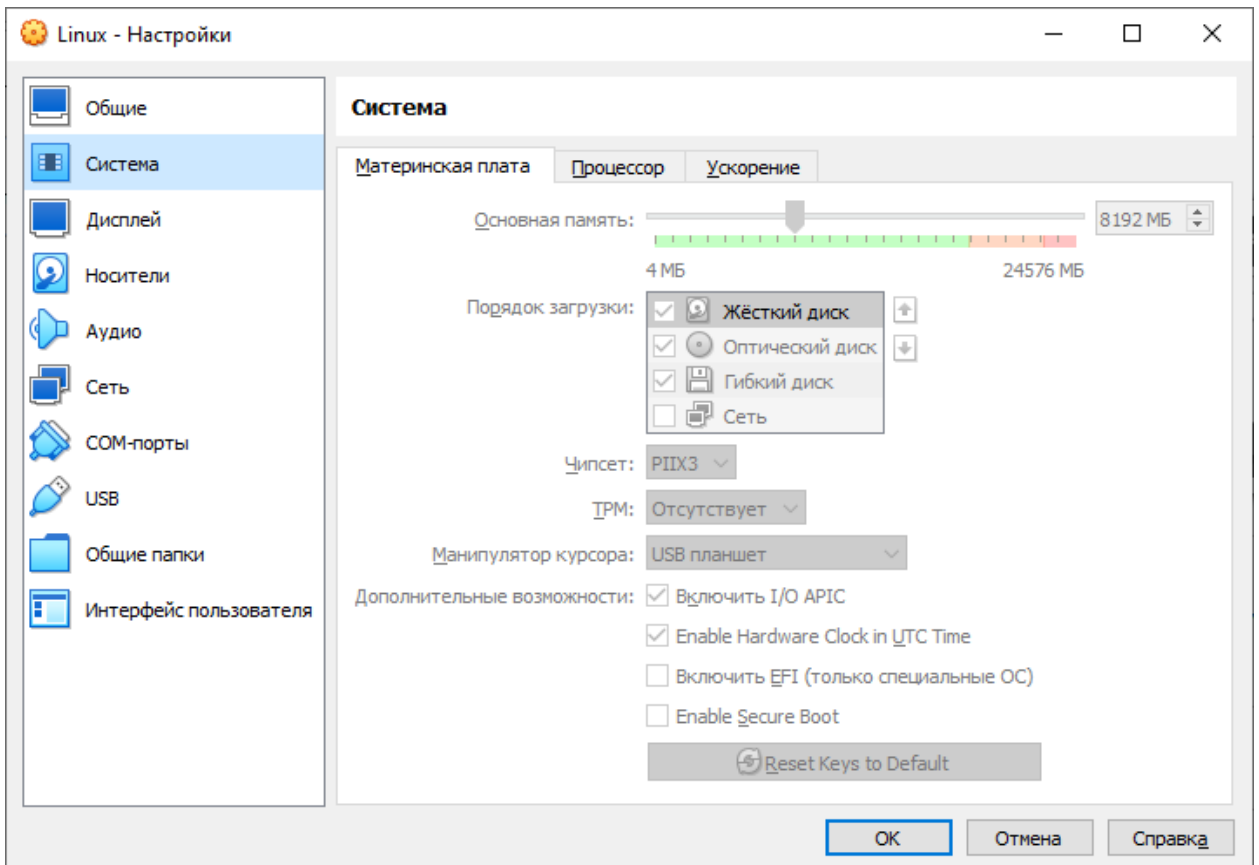
Далее выбираем объем и тип виртуального жесткого диска: динамический или фиксированный (выделить место в полном объеме). При динамическом виртуальном жестком диске размер файла диска будет увеличиваться в зависимости от его наполнения в виртуальной машине.



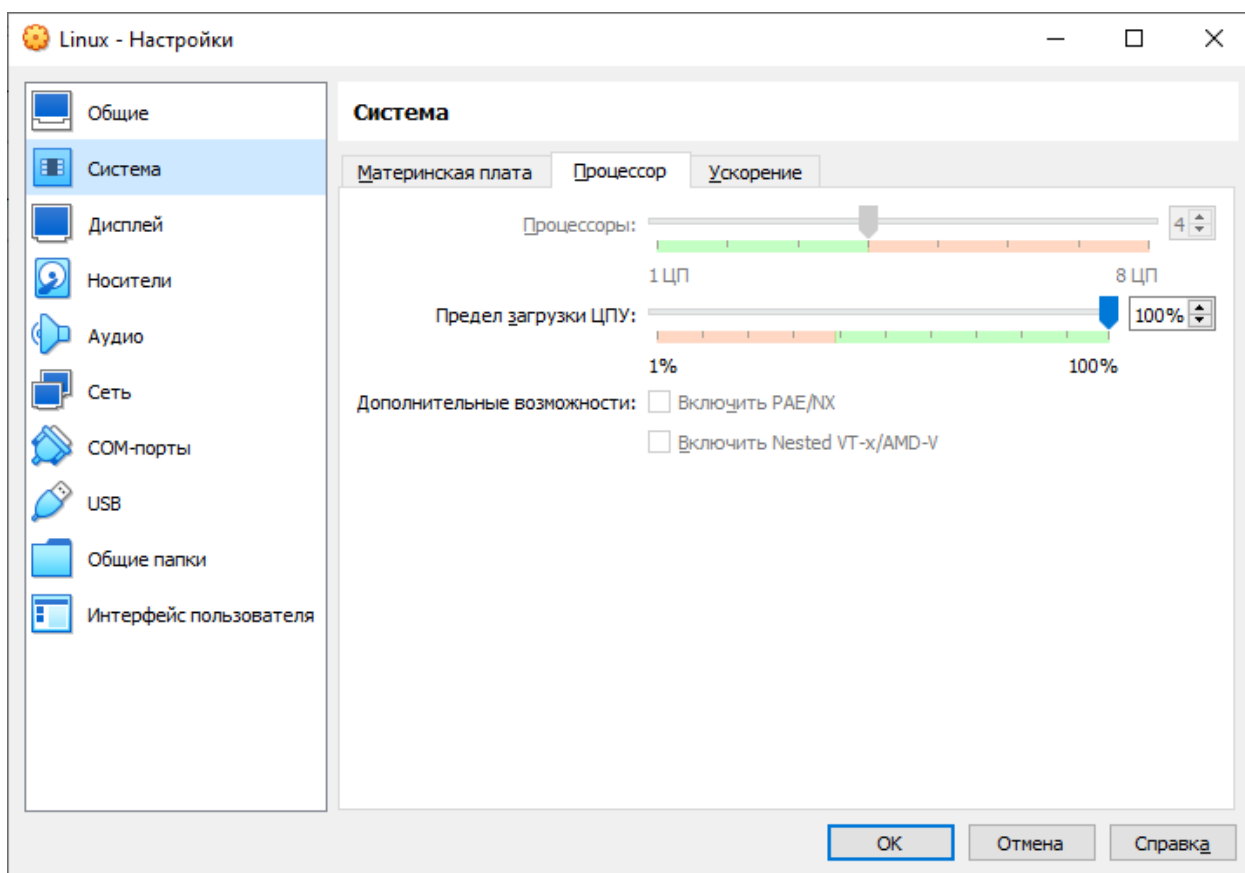
В результате будет создана новая виртуальная машина.

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».

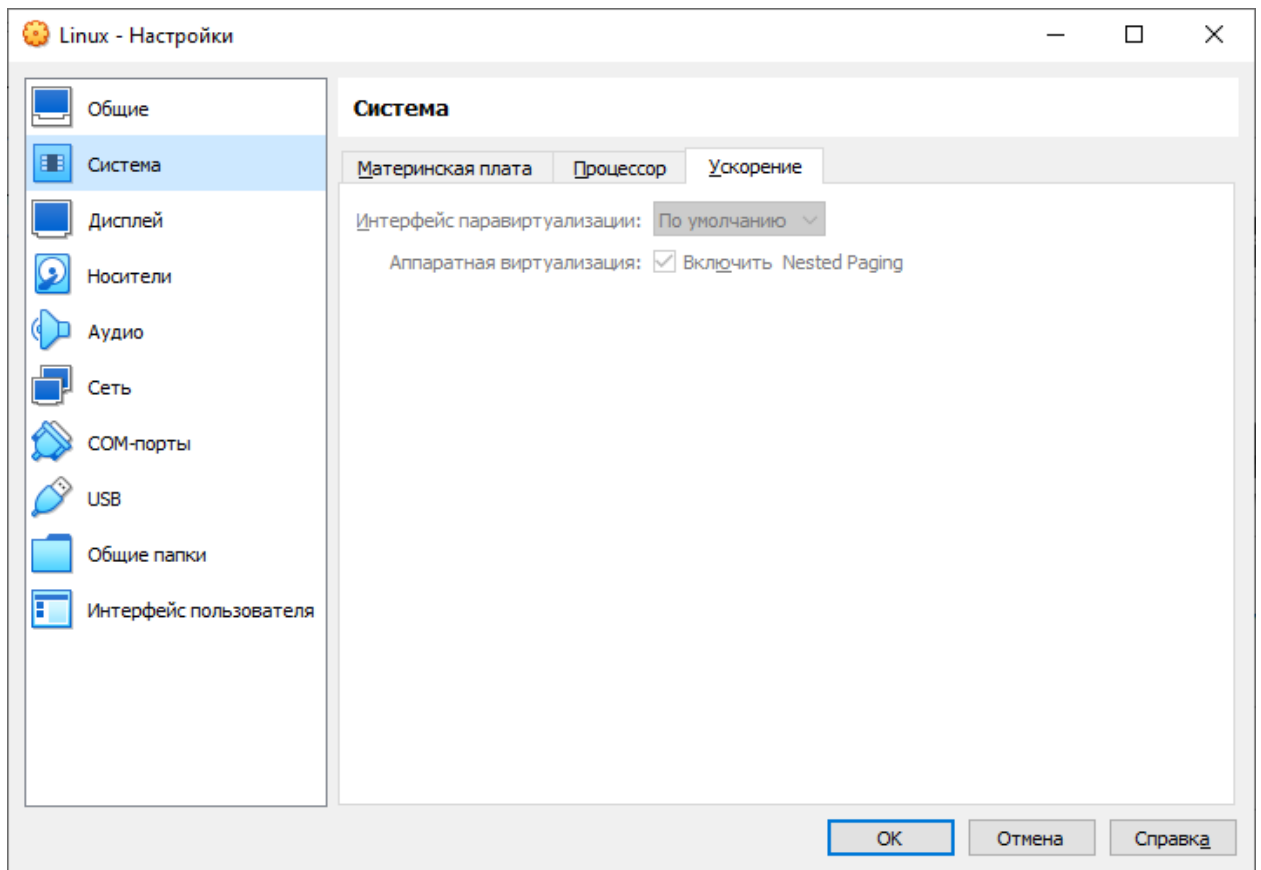


В отобразившемся окне в разделе «Процессор» можно изменить число процессоров и предел загрузки ЦПУ.

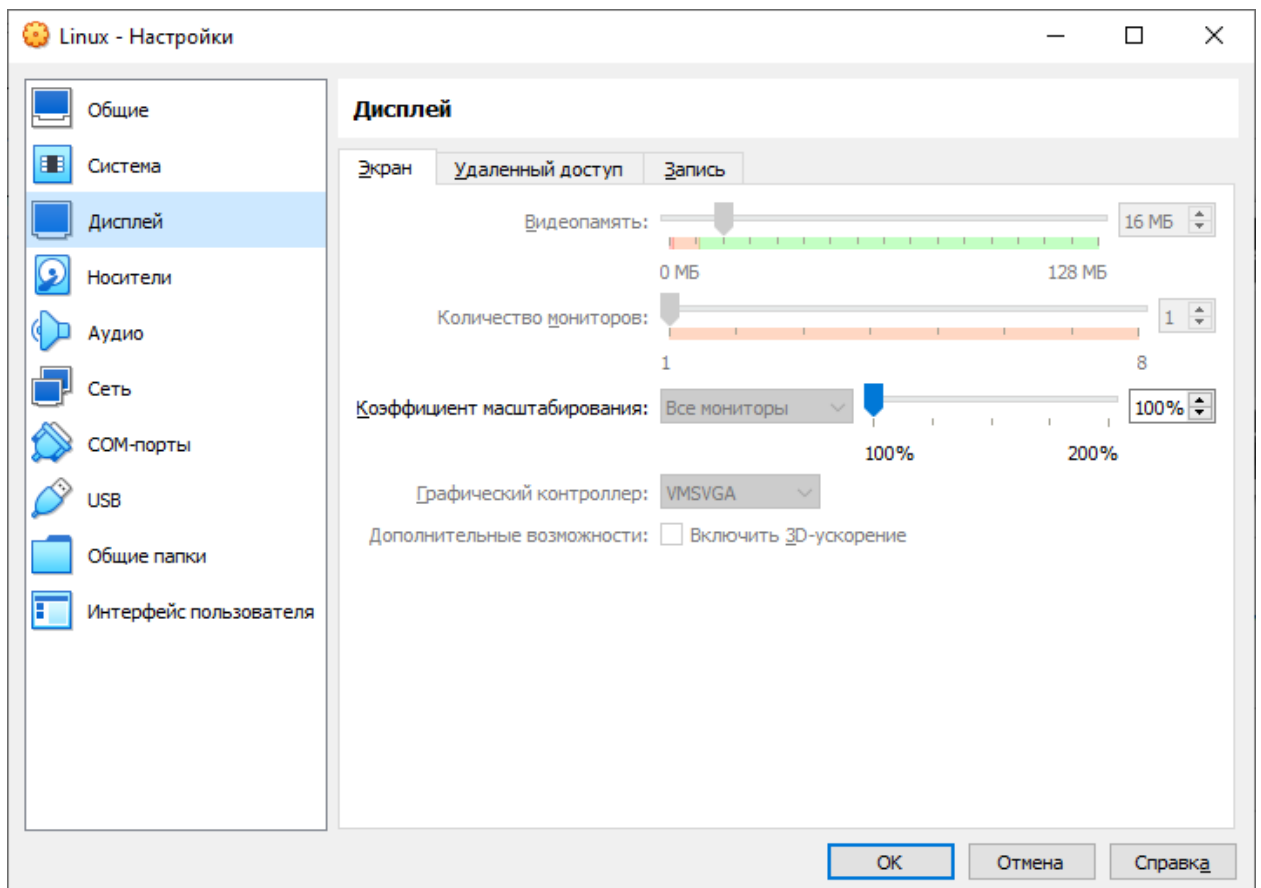


Функция «Включить PAE/NX» предназначена для поддержки 4 и более Гб ОЗУ в 32-битных системах.

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



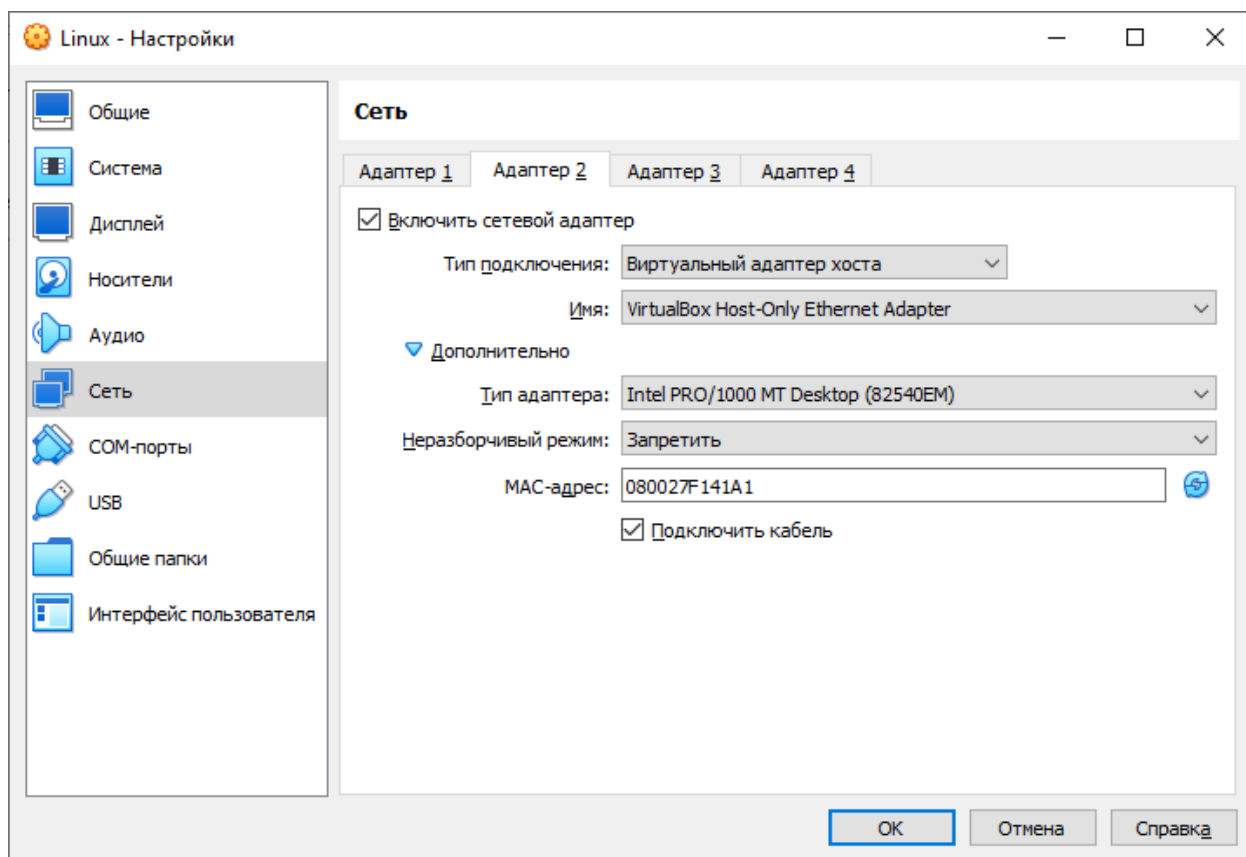
Иногда при установке новой виртуальной машины значение видеопамати по умолчанию составляет 16 Мб, тогда как рекомендуется выделять не менее 128 Мб. Изменить это можно в настройках раздела «Дисплей».



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.

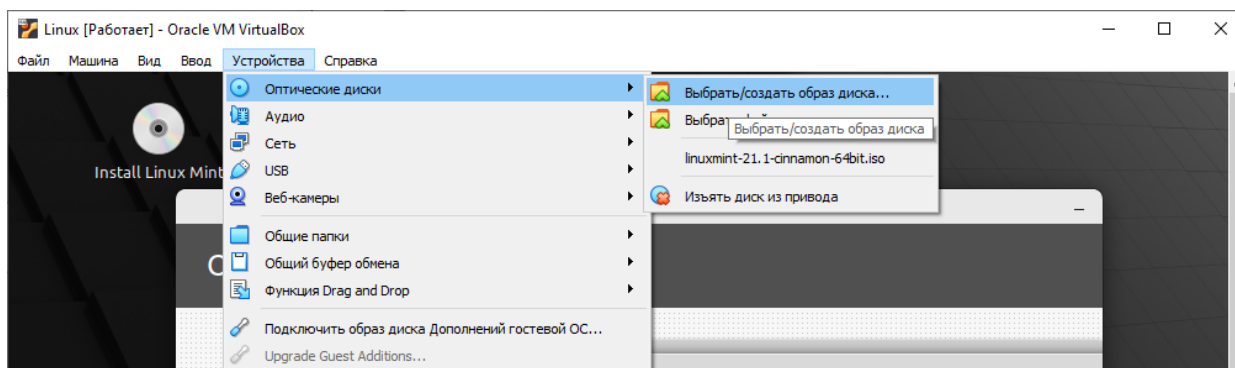
Еще одна полезная функция – «Клонирование». С ее помощью мы можем сделать резервную копию виртуальной машины, чтобы в последующем обратиться к ней при возникновении различного рода проблем.

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:

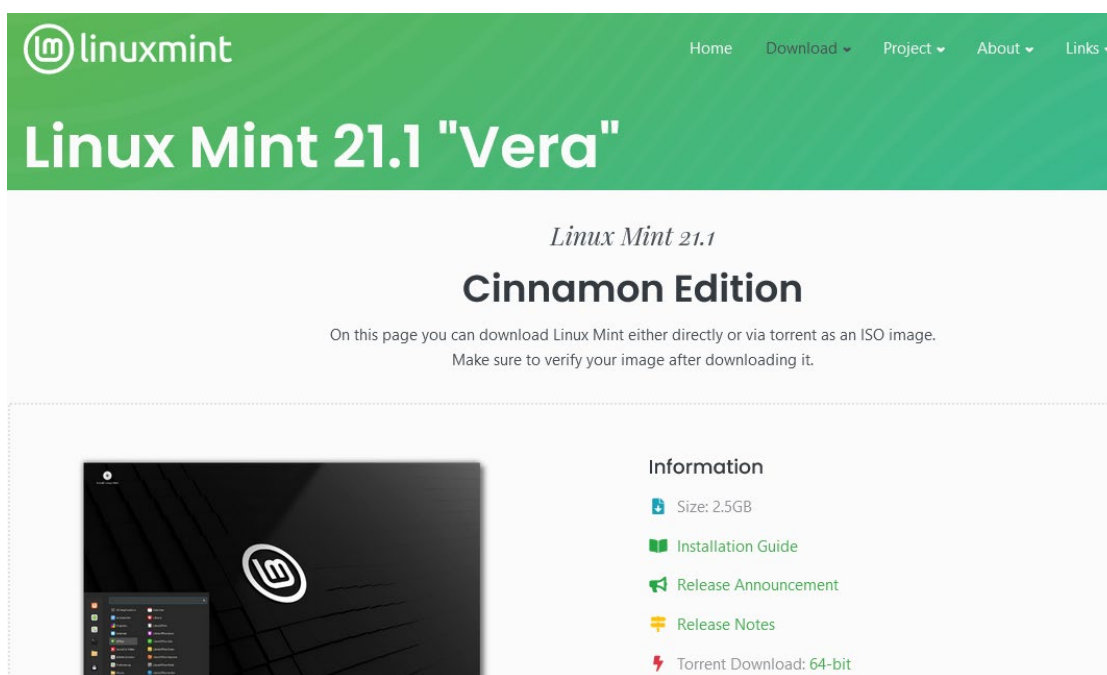
1. Загрузить ISO-образ дистрибутива системы.

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.

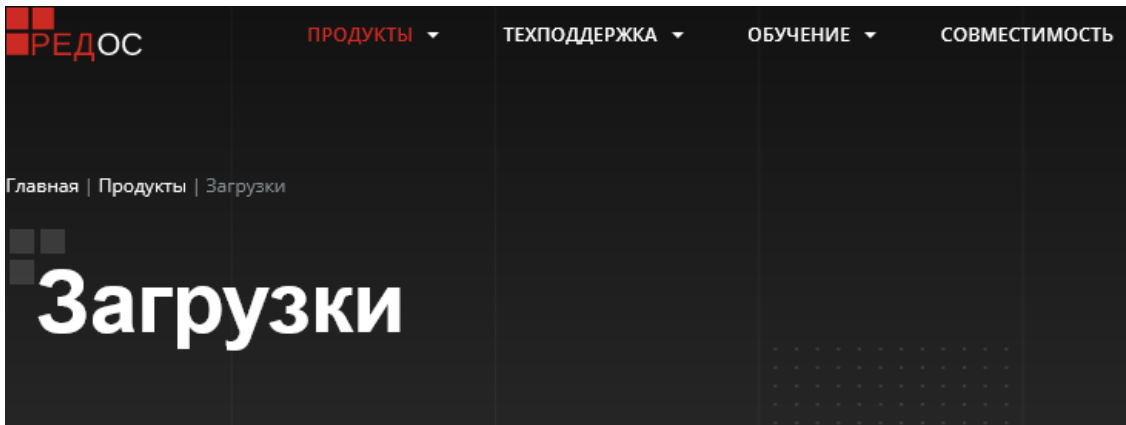


В качестве вариантов системы для выполнения лабораторных подойдут:

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



2. РедОС, дистрибутив доступен на официальном сайте по адресу <https://redos.red-soft.ru/product/downloads/>



Стандартная редакция

[СВЕРНУТЬ ↑](#)

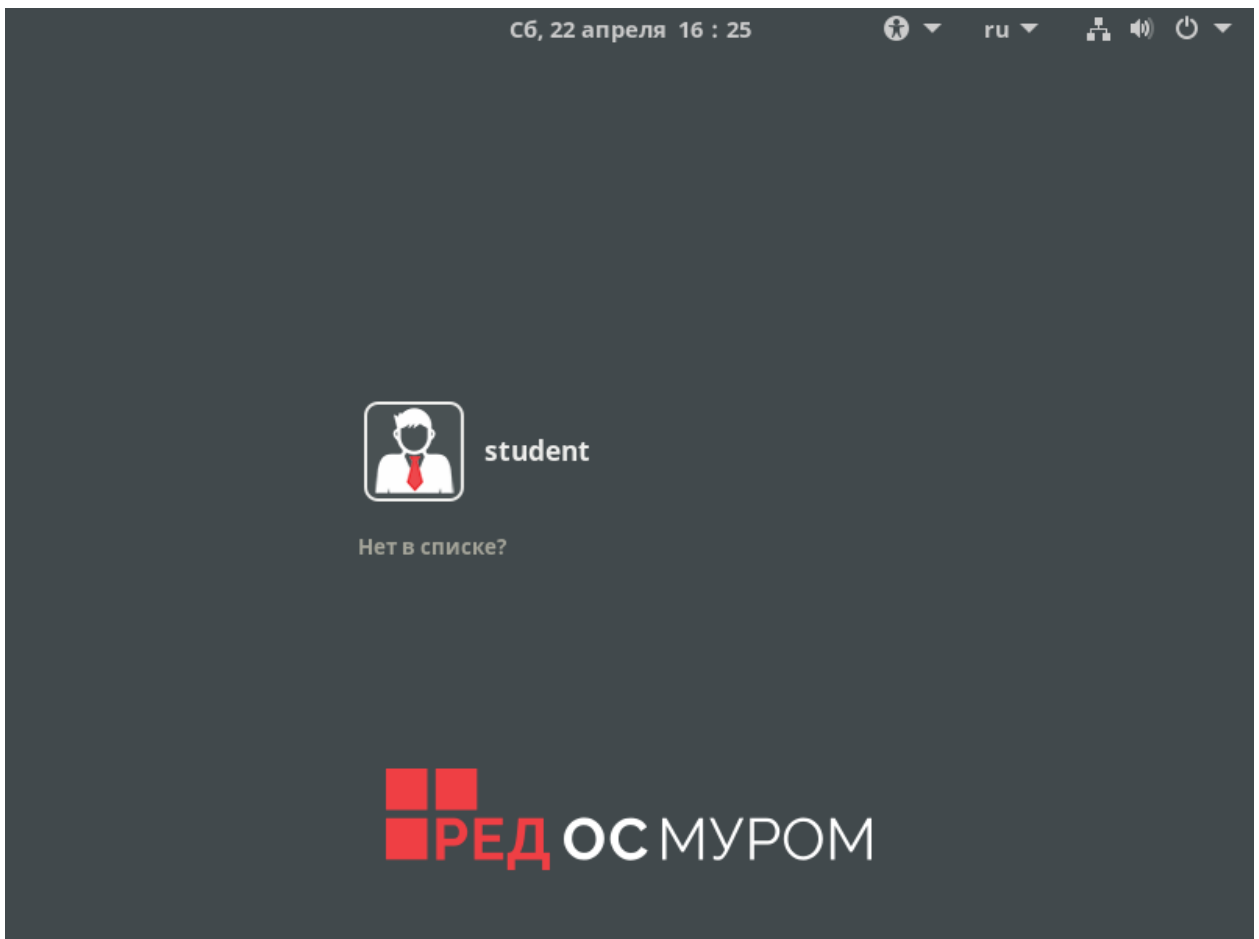
ISO-образ включает конфигурации «Рабочая станция» и «Сервер».

РЕД ОС 7.3 | Ядро Linux 5.15.72 (в репозитории 5.15.87)



7.3.2 x86_64 4 587 Мб | Хеш сумма **MD5SUM**

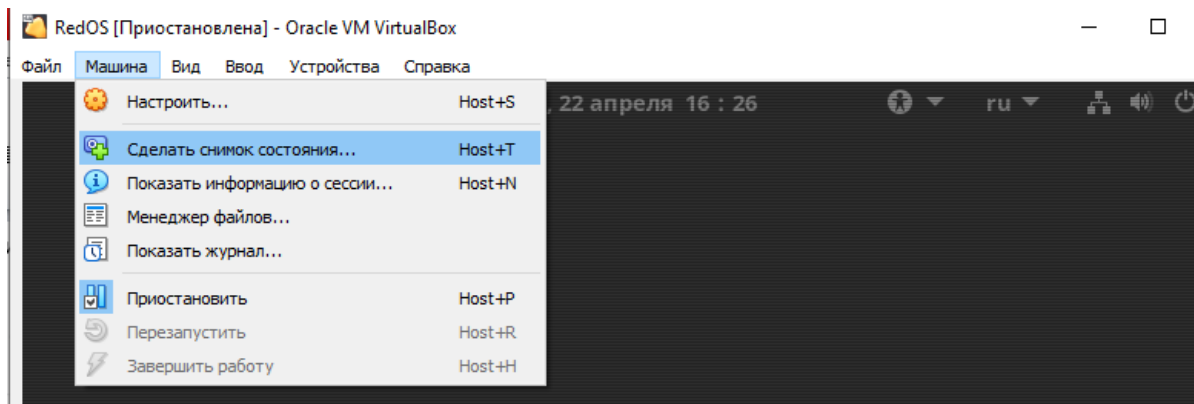
Когда операционная система будет установлена, вы получите к ней доступ через окно VirtualBox. Например, вот так будет выглядеть РедОС:



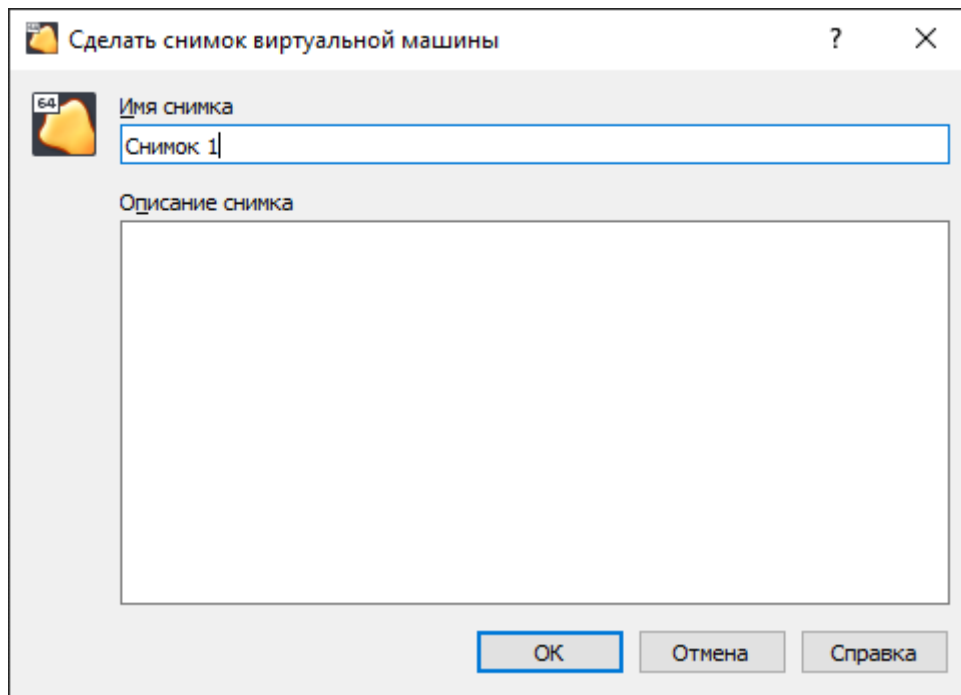
Ранее мы уже создавали копию виртуальной машины, которой можно воспользоваться в случае непредвиденных проблем. Но это не единственный способ создания резервной копии – мы также можем использовать специальную функцию «Снимок состояния». Она позволяет возвращать систему к предыдущему состоянию.

Создать снимок можно следующим образом:

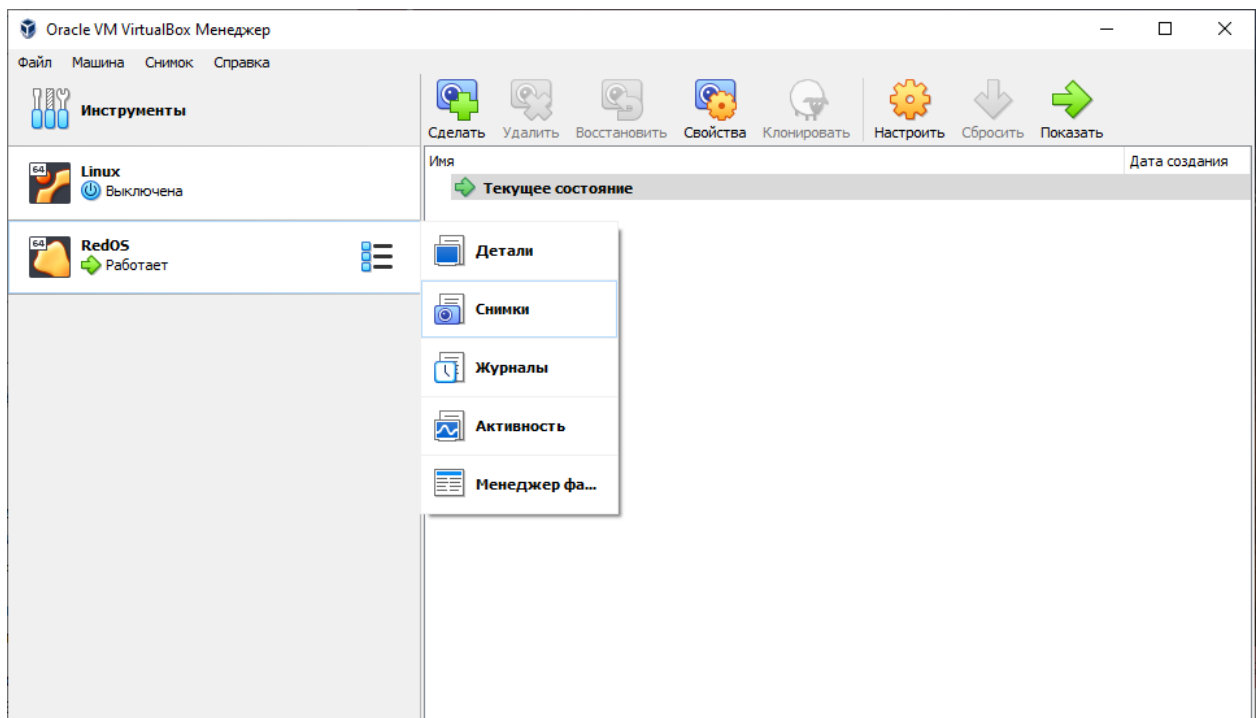
Запускаем виртуальную машину и в верхней части выбираем «Машина» -> «Сделать снимок состояния...».



Задаем ему имя и по желанию прописываем описание.



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



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 подключиться 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.

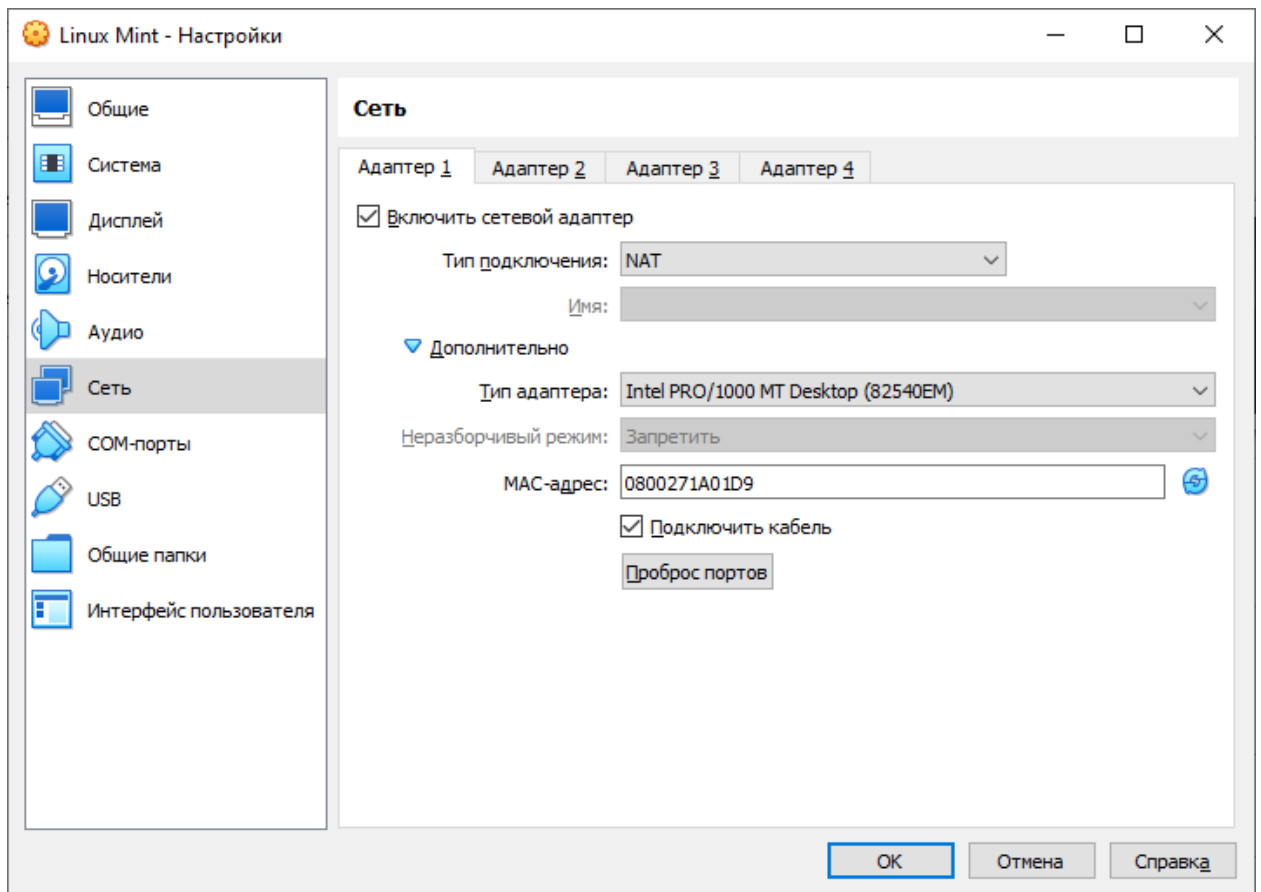
PuTTY – это бесплатная клиентская программа для удаленного подключения, например к серверам по протоколам SSH, Telnet и другим. Она позволяет управлять удаленным компьютером. PuTTY — это клиентская часть, серверная должна быть реализована на удаленной стороне, должен быть установлен SSH сервер. Скачать PuTTY можно с официального сайта — www.putty.org.

Настройка проброса портов в VirtualBox для SSH

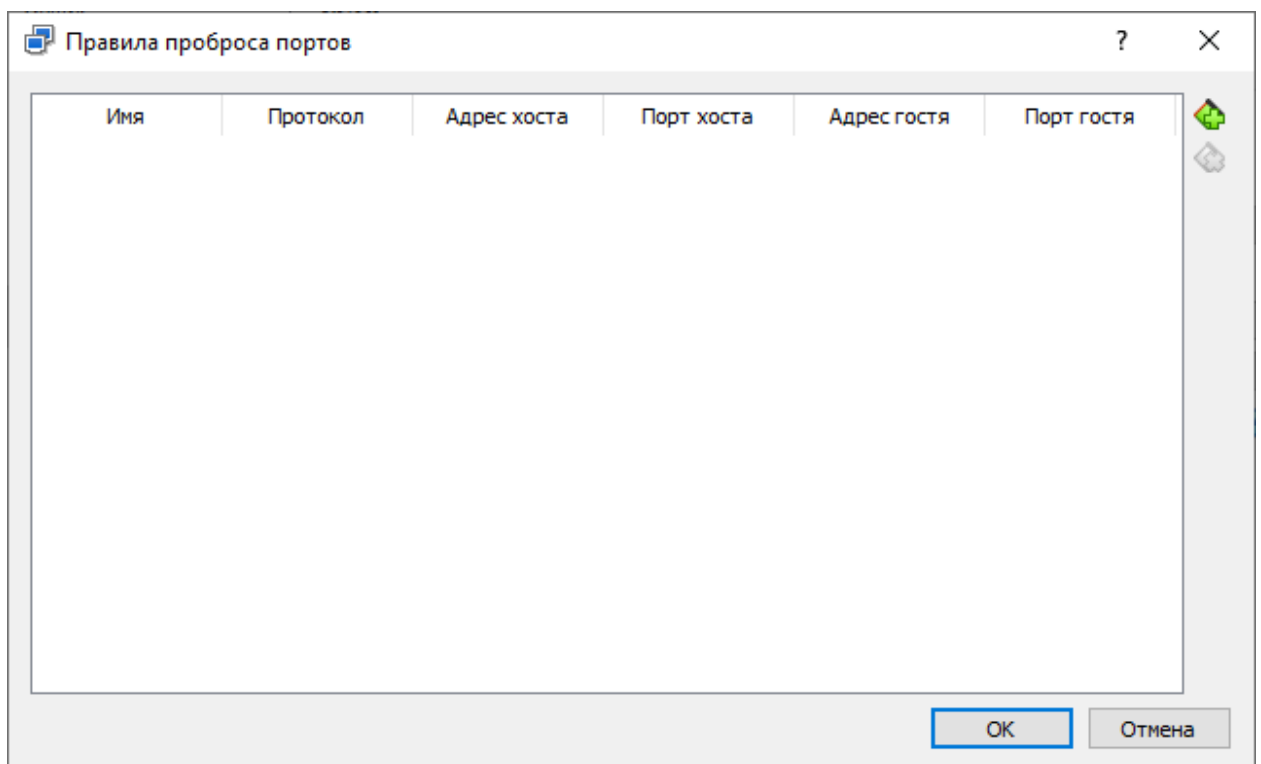
SSH – сервер обычно прослушивает 22 порт, поэтому нам необходимо пробросить подключение именно на 22 порт.

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 10.2.0.15).

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;

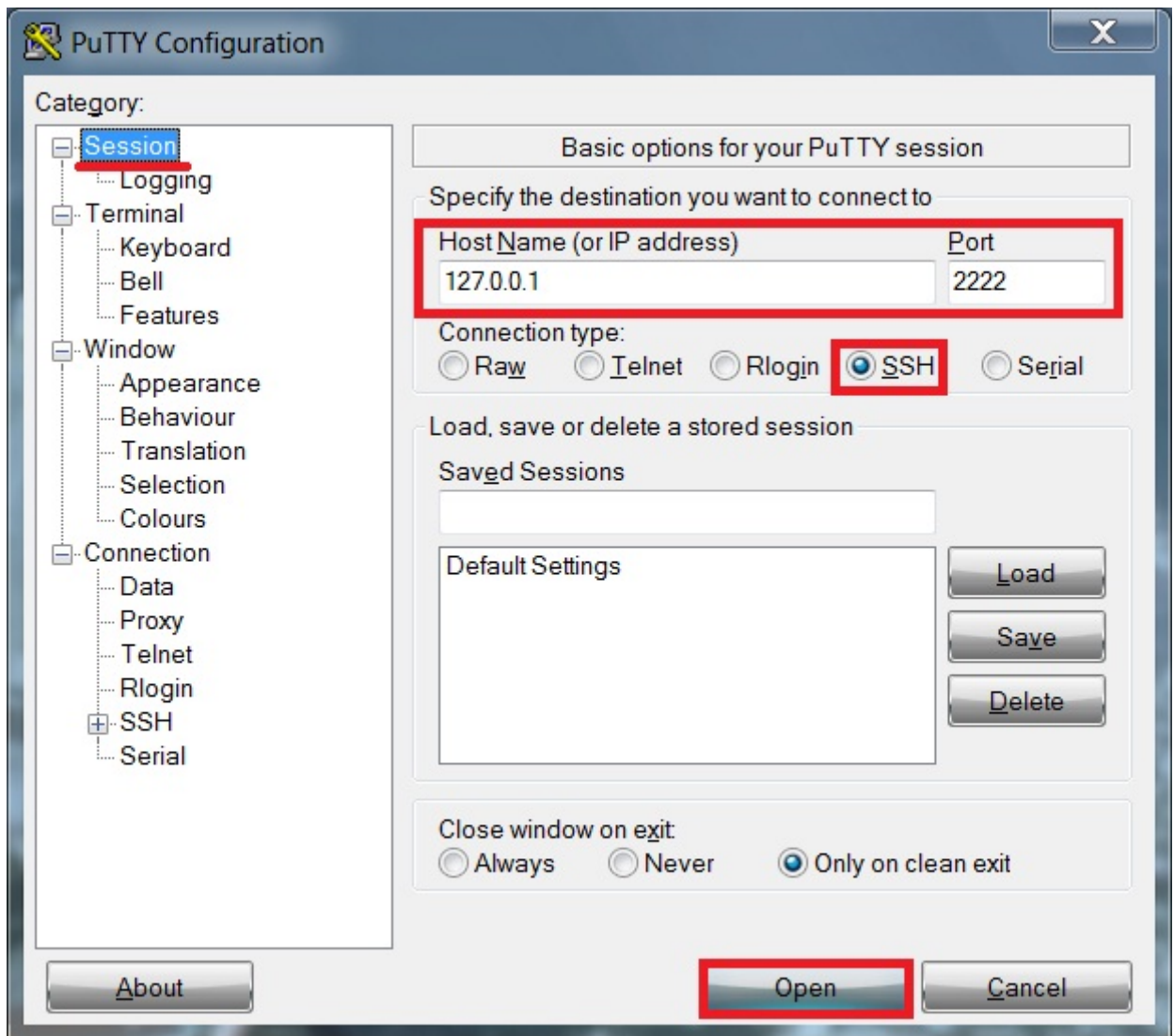
Порт хоста – любой свободный порт компьютера, который будет использоваться для перенаправления на нужный порт в гостевой ОС, например 2222;

IP гостя – здесь указываем IP адрес гостевой операционной системы, на который будет происходить перенаправление, в моем случае это 10.2.0.15;

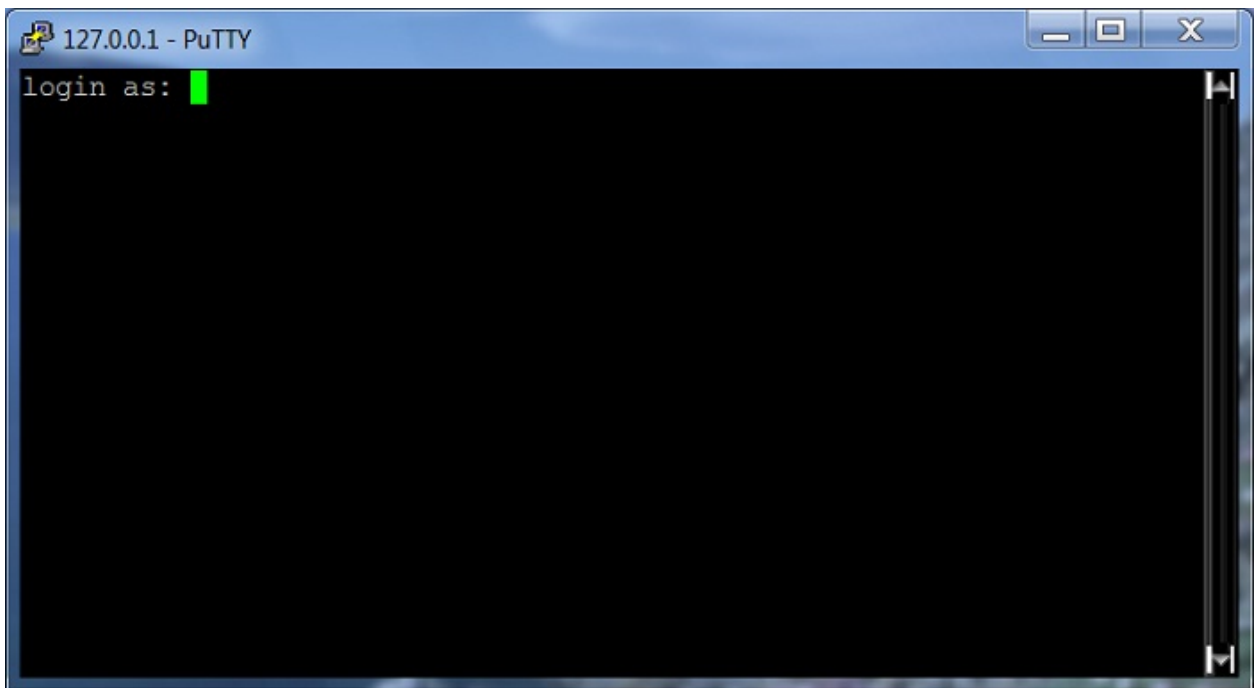
Порт гостя – порт гостевой ОС, на который нам необходимо пробрасывать наши запросы. В нашем случае это 22 порт, который прослушивает SSH сервер.

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.



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.

Указание закрытого SSH-ключа в клиенте.

При использовании openssh команда выглядит следующим образом:

```
«ssh -i /home/username/.ssh/id_rsa username@ip»
```

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

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

/etc/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
fscck	Запуск проверки ФС на целостность	fscck /dev/sda
mkfs	Создание ФС на диске	mkfs.ext4 /dev/sda
mount, umount	Монтирование, отмонтирование ФС	mount /dev/sda /mnt/storage

Менеджер логических томов (англ. logical volume manager) — подсистема операционных систем Linux и OS/2, позволяющая использовать разные области одного жёсткого диска и/или области с разных жёстких дисков как один логический том. Реализована с помощью подсистемы device mapper.

LVM добавляет уровень абстракции между физическими/логическими дисками (привычными разделами, с которыми работает fdisk и аналогичные программы) и файловой системой. Это достигается путём разбивки изначальных разделов на блоки либо использования отдельных разделов или блочных устройств (physical volume (pv)) и объединения их в единый виртуальный том, точнее, группу томов (volume group (vg)), которая далее разбивается на логические тома (logical volume (lv)). Для файловой системы логический том представлен как обычное блочное устройство, хотя отдельные pv-тома могут находиться на разных физических устройствах (и даже сам pv может быть распределён подобно RAID).

Термины:

Физический том (англ. physical volume, pv) — устройство, представляющееся системе как один диск (жёсткий диск или его раздел, RAID-массив).

Группа томов (англ. volume group, vg) — несколько физических томов pv (группа, набор).

Логический том (англ. logical volume, lv) — логический раздел; аналог разделов hda1, sdb3 и др.; виртуальное блочное устройство.

Физический диапазон (англ. physical extent, pe) — область на физическом томе pv размером в несколько мегабайт. pv разбивается на области pe равного размера.

Логический диапазон (англ. logical extent, le) — область на логическом томе lv. lv разбивается на области le равного размера.

Основные команды для работы с LVM:

Название команды	Описание	Пример
Pvs, vgs, lvs	Вывод информации об утилизации подключенных ФС	pvs
Pvcreate, pvremove	Подсчет размера файлов и директорий	pvcreate /dev/sda
vgcreate, vgremove	Изменение размера ФС ext2/ext3/ext4	vgcreate vg01 /dev/sda

lvcreate	Просмотр информации о разделах дисков	lvcreate -n lv01 -l 100%VG vg01
lvextend	Запуск проверки ФС на целостность	lvextend /dev/vg01/lv01 -L+1000M
pvmove	Создание ФС на диске	pvmove /dev/sda /dev/sdb

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.

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

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

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/mstab awk '{print \$1" "\$2}'
sort, uniq	Сортировки и проверки на уникальность	sudo fdisk -l grep "Disk \\" sort
cut	Разбивка по секциям текста	cat /etc/mstab 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.

Задание на лабораторную работу

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

1. **Запустить SSH-клиент.** В случае использования графического клиента запустить программу. В случае использования терминальных клиентов — открыть терминал.
2. **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`
3. **Создать пару SSH-ключей с опциями по умолчанию.** Пользуясь разобранным в методических указаниях алгоритмом создать пару ключей, не изменяя значения по умолчанию.
4. **Разместить пару ключей в нужных местах.** Пользуясь разобранным в методических указаниях алгоритмом разместить пару ключей в нужных местах для использования в авторизации.
5. **Отключиться от ОС сервера и попробовать авторизоваться по ключу.**
6. Повторить подключение по SSH, но уже с подключением по ключу.
7. **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.
8. **Открыть файл скрипта любым редактором.** Выбор редактора индивидуален и зависит от преследуемых целей и личных предпочтений.
9. **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.
10. **Запустить скрипт.** Выполнить свой скрипт и получить нужный результат.
11. **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.
12. **Просмотреть утилизацию примонтированных ФС.** С помощью команды «df» просмотреть общий и свободный объем ФС, примонтированных в данный момент к ОС.
13. **Посчитать объем файлов в домашней директории.** С помощью команды «du» посчитать объем всех файлов, расположенных в своей домашней директории.
14. **Создать LVM с новым диском, подключенным к VM.** Создать на нем ФС ext4, которая будет монтироваться в директорию /mnt/storage при загрузке VM.
15. Описать все эти действия в скрипте, снабженном комментариями.
16. **Поиск текста по регулярным выражениям.** Найти в главном файле Syslog все сообщения за (определенный период времени).

17. **Форматирование вывода.** Вывести на экран название PV и соответствующих им VG в формате “PV = VG”, используя команду вывода из состава LVM.

Результат: подключились к своей VM, создали ключ и авторизовались по нему. Написали скрипт, создающий дерево каталогов. Посмотрели, какие ФС используются в ОС и узнали их объем. Посмотрели, как работают 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.