

Лабораторная работа №1

Информационная безопасность

Феоктистов Владислав Сергеевич

Содержание

1	Цель работы	6
2	Задание	7
3	Теоретическое введение	8
4	Выполнение лабораторной работы	10
4.1	Установка программы VirtualBox	10
4.2	Создание виртуальной машины	15
4.3	Установка Rocky Linux	26
4.4	Подключение гостевой ОС	36
5	Выводы	41
6	Контрольные вопросы	42
7	Домашняя работа	47
	Список литературы	52

Список иллюстраций

4.1	Официальный сайт VirtualBox	10
4.2	Скачивание программы VirtualBox	11
4.3	Установка VirtualBox	11
4.4	Установка VirtualBox	12
4.5	Установка VirtualBox	12
4.6	Установка VirtualBox	13
4.7	Установка VirtualBox	13
4.8	Установка VirtualBox	14
4.9	Запуск VirtualBox	14
4.10	Настройки VirtualBox	15
4.11	Официальный сайт Rocky Linux	16
4.12	Скачивание образа ОС Rocky Linux	16
4.13	Создание виртуальной машины. Название, тип и версия.	17
4.14	Создание виртуальной машины. Выделение оперативной памяти.	18
4.15	Создание виртуальной машины. Жесткий диск.	18
4.16	Создание виртуальной машины. VDI	19
4.17	Создание виртуальной машины. Формат хранения.	20
4.18	Создание виртуальной машины. Имя и размер файла/виртуальной машины	20
4.19	Завершение первичной настройки виртуальной машины	21
4.20	Дополнительная настройка виртуальной машины. Процессор.	22
4.21	Дополнительная настройка виртуальной машины. Носители.	23
4.22	Дополнительная настройка виртуальной машины. Носители. Контроллер.	24
4.23	Дополнительная настройка виртуальной машины. Носители. Контроллер.	24
4.24	Дополнительная настройка виртуальной машины. Носители. Контроллер.	25
4.25	Запуск виртуальной машины	25
4.26	Установка Rocky Linux	26
4.27	Установка Rocky Linux. Выбор языка	27
4.28	Установка Rocky Linux. Конфигурационное окно	28
4.29	Установка Rocky Linux. Выбор программ	29
4.30	Установка Rocky Linux. Место установки ОС	29
4.31	Установка Rocky Linux. KDUMP	30
4.32	Установка Rocky Linux. Network & host name	31
4.33	Установка Rocky Linux. Пароль для root пользователя	32

4.34	Установка Rocky Linux. Создание пользователя с правами	32
4.35	Установка Rocky Linux. Завершение настройки	33
4.36	Установка Rocky Linux. Завершение загрузки	34
4.37	Лицензионное соглашение	34
4.38	Лицензионное соглашение	35
4.39	Вход в учетную запись	36
4.40	Подключение образа диска Дополнений гостевой ОС	37
4.41	Подключение образа диска Дополнений гостевой ОС. Диалоговое окно	38
4.42	Подключение образа диска Дополнений гостевой ОС. Завершение установки	39
4.43	Режим полного экрана	40
7.1	Утилита dmesg	47
7.2	Утилита dmesg с использованием less	48
7.3	Версия линукса	48
7.4	Частота процессора	49
7.5	Модель процессора	49
7.6	Объем оперативной памяти	50
7.7	Тип гипервизора	50
7.8	Тип файловой системы корневого раздела	51

Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . .	8
3.2	Описание некоторых, используемых в процессе выполнения лабораторной работы, команд	9

1 Цель работы

Целью данной работы является приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов, работы с терминалом: работа с утилитами `dmesg`, `less`, `grep`

2 Задание

Установить программу VirtualBox для запуска операционных систем внутри основной/текущей операционной системы, создать виртуальную машину в VirtualBox, скачать образ Rocky Linux с официального сайта и установить её на созданную виртуальную машину, установить Дополнения гостевой ОС, выполнить домашнее задание.

3 Теоретическое введение

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux

Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую систему
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Таблица 3.2: Описание некоторых, используемых в процессе выполнения лабораторной работы, команд

Команда	Описание
<code>dmesg</code>	Вывод буфера сообщений ядра в стандартный поток вывода (stdout) (по умолчанию на экран)
<code>less</code>	Отображает содержимое файла или вывода команды по одной странице за раз
<code>ls</code>	Выводит содержимое каталога. Опция <code>-l</code> выводит дополнительную информацию, <code>-a</code> отображает скрытые файлы, в названии которых в самом начале стоит символ <code>.'</code>
<code>grep</code>	Дает возможность вести поиск строк. Также можно передать вывод любой команды в <code>grep</code> , что сильно упрощает работу во время поиска

Более подробно об Unix см. в [1–6].

4 Выполнение лабораторной работы

4.1 Установка программы VirtualBox

В первую очередь, я скачал VirtualBox для запуска виртуальных машин с гостевыми операционными системами внутри основной системы. Обычно VM (Virtual Machine) будет отображаться как окно на рабочем столе компьютера, но в зависимости от того, какой из различных интерфейсов VirtualBox используется, она может отображаться в полноэкранном режиме или удаленно на другом компьютере.

Удобнее всего скачивать дистрибутив виртуальной машины с официального сайта «www.virtualbox.org», со странички <https://www.virtualbox.org/wiki/Downloads> (рис. 4.1 и 4.2).



Рис. 4.1: Официальный сайт VirtualBox

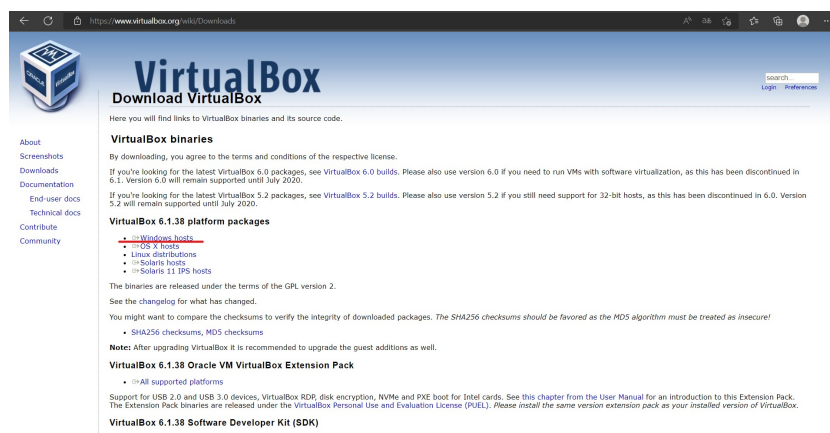


Рис. 4.2: Скачивание программы VirtualBox

Перейду к установке VirtualBox и его настройке. Нажимаю по установщику в формате .exe для систем на базе Windows и нажимаю каждый раз кнопку «Next» (рис. 4.3 - 4.5), а на рисунке 4.6 «Yes».



Рис. 4.3: Установка VirtualBox

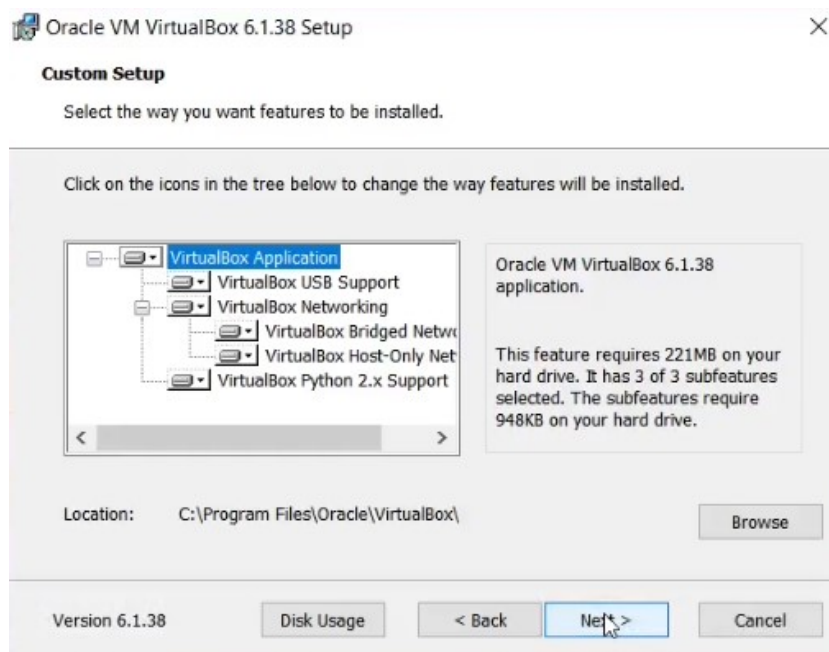


Рис. 4.4: Установка VirtualBox

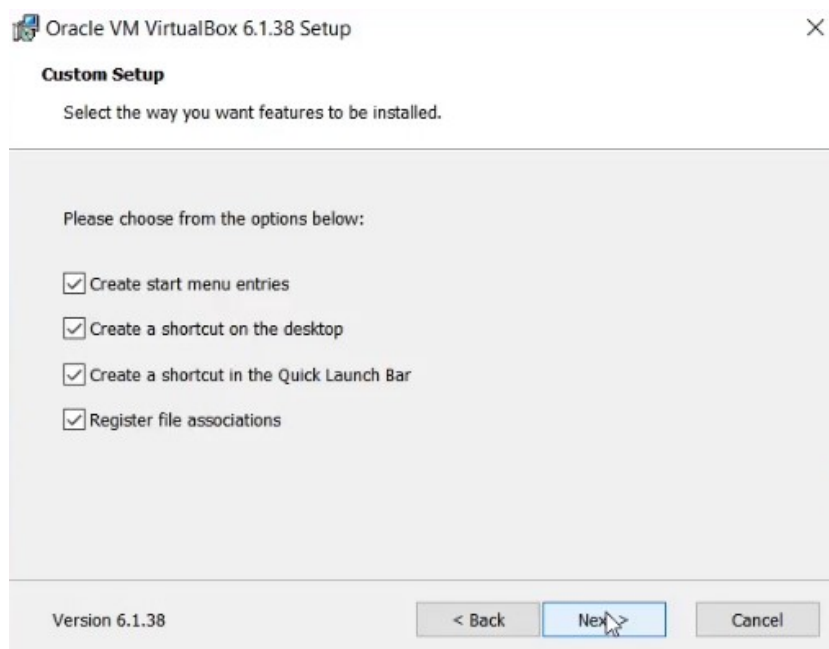


Рис. 4.5: Установка VirtualBox



Рис. 4.6: Установка VirtualBox

На рисунке 4.7 - кнопку «Install», а на рисунке 4.8 – «Finish».

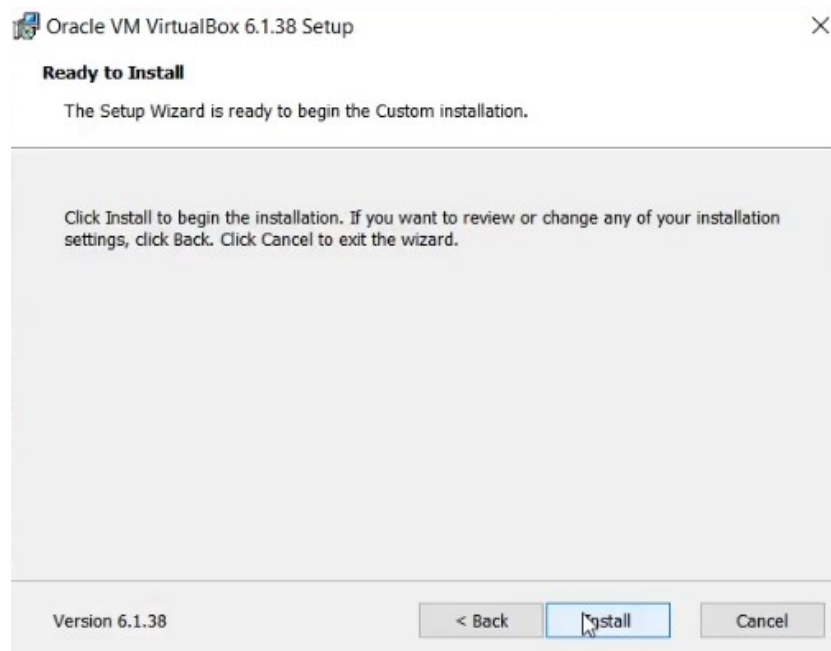


Рис. 4.7: Установка VirtualBox



Рис. 4.8: Установка VirtualBox

После установки, если была установлена галочка перед «Start Oracle VM VirtualBox», откроется окно программы VirtualBox, как на рисунке 4.9. Иначе нужно будет запустить программу из меню «Пуск».



Рис. 4.9: Запуск VirtualBox

После запуска я нажал на кнопку «Настройки». В разделе «Общие» стоит поменять путь для размещения виртуальных ОС, если не устраивает размещение или

недостаточно места на текущем диске (рис. 4.10). Я ничего не менял.

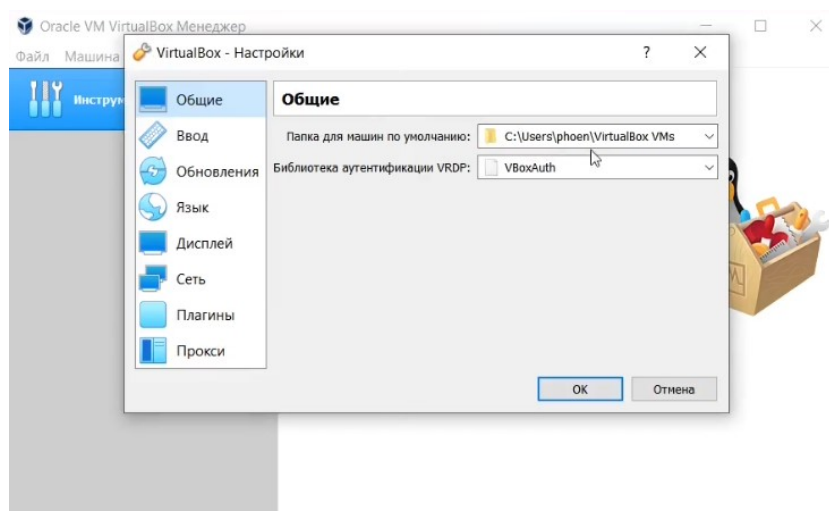


Рис. 4.10: Настройки VirtualBox

На этом установка VirtualBox завершена.

4.2 Создание виртуальной машины

В начале скачиваю образ ОС (операционной системы). В моем случае это будет Rocky Linux 8, поскольку на 9ой версии не получится установить pandoc-crossref (невозможно на момент создания отчета). Скачать образ можно на официальном сайте «www.rockylinux.org» со странички <https://www.rockylinux.org/download> (рис. 4.11).

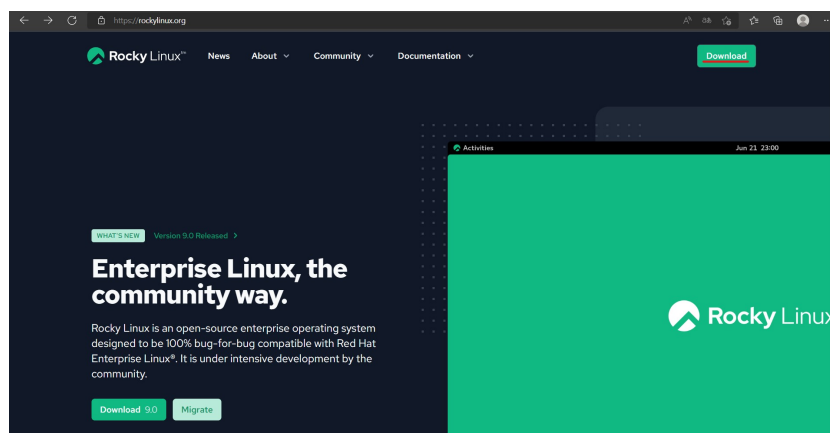


Рис. 4.11: Официальный сайт Rocky Linux

У меня на компьютере установлен 64 битный процессор Intel(R) Core(TM) i7-8550U с частотой 1.8ГГц, а также 16 ГБ ОЗУ, поэтому в качестве образа я выбрал 64 битную версию, которая указана в самом начале списка под названием x86_64. Для скачивания нажимаю на DVD в соответствующей с версией (рис. 4.12).

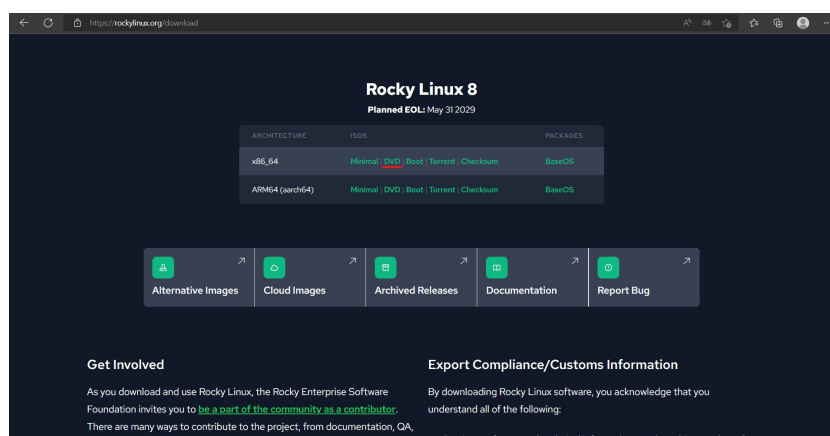


Рис. 4.12: Скачивание образа ОС Rocky Linux

Теперь перейду к созданию виртуальной машины и её настройке. Как на рисунке 4.13 выбираю тип Linux, а версию RedHat с соответствующей разрядностью, у меня это 64-bit. Разным дистрибутивам соответствуют разные версии, для Rocky – это RedHat. Имя для виртуальной машины можно задать любое, на ее работу это никак не повлияет. Я же назвал в соответствии с заданием - логином учетной записи от ДК кабинета: vsfeoktistov. Нажимаю на кнопку «Далее».

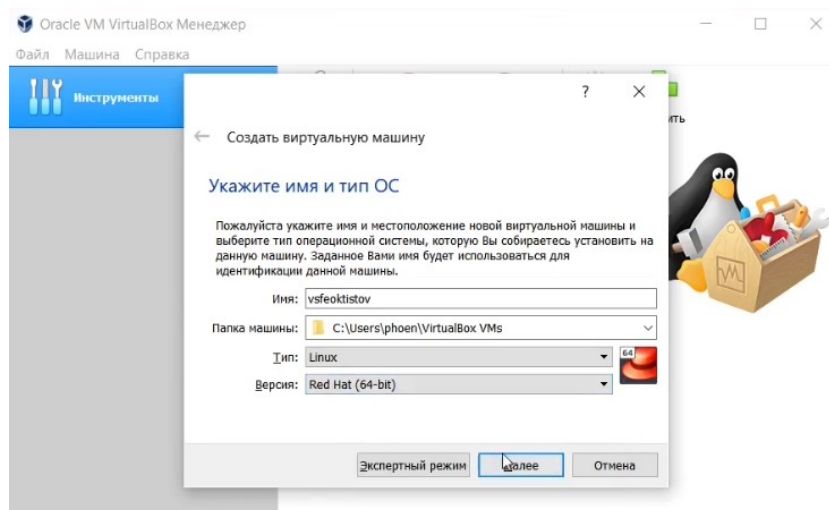


Рис. 4.13: Создание виртуальной машины. Название, тип и версия.

Выбираю необходимый объем памяти для виртуальной машины. Рекомендуется установить не менее 2024 МБ и не более 50% от общего объема памяти ОЗУ компьютера так, чтобы осталось и для работы основной системы, на которой будет запускаться виртуальная машина. Так как у меня 16 ГБ ОЗУ я выделил 6 ГБ (6144 МБ) (рис. 4.14). Можно выбрать другой размер объема оперативной памяти, передвигая курсор по шкале, или ввести размер памяти в цифровом значении в МБ. Выбирать стоит исходя из задач, для которых будет предназначена VM. В дальнейшем, при необходимости, можно изменить объем выделяемой оперативной памяти в настройках этой виртуальной машины.

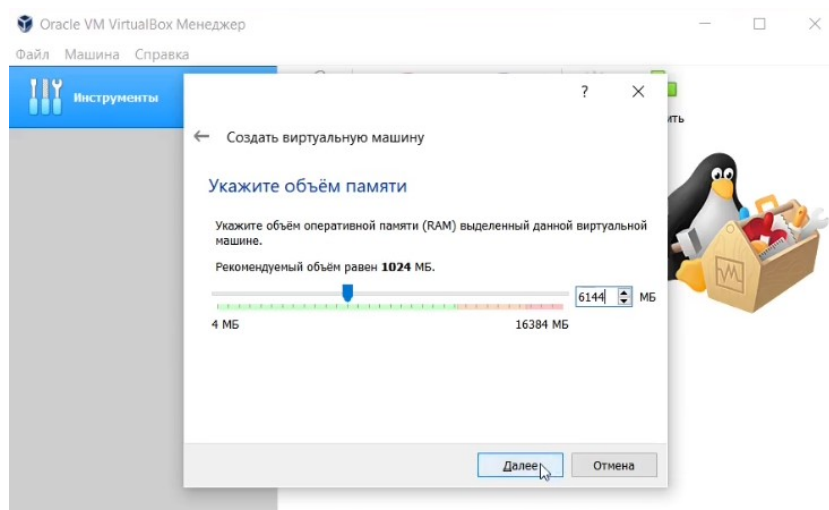


Рис. 4.14: Создание виртуальной машины. Выделение оперативной памяти.

Далее можно создать новый виртуальный диск или использовать существующий. Так как я только что установил VirtualBox, я выбрал «Создать новый виртуальный жёсткий диск». Есть ещё вариант «Не подключать виртуальный жесткий диск», чтобы подключить виртуальный жесткий диск после создания виртуальной машины (рис. 4.15).

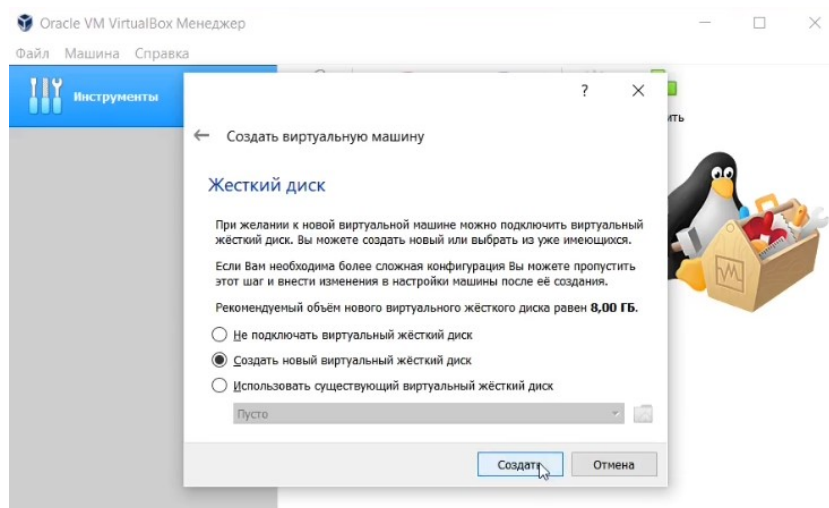


Рис. 4.15: Создание виртуальной машины. Жесткий диск.

В следующем окне необходимо будет выбрать тип виртуального жесткого диска. Вы можете выбрать следующие форматы дисков (виртуальных контейнеров):

- VDI (VirtualBox Disk Image) — формат диска VirtualBox. - VMDK (Virtual Machine Disk) — формат диска VMware. - VHD (Virtual Hard Disk) — формат диска Microsoft. Если не планируется использовать создаваемый виртуальный накопитель с другими инструментами программной виртуализации, стоит оставить предлагаемый по умолчанию тип жесткого диска «VDI (VirtualBox Disk Image)» 4.16. Диски этого формата будут открываться в программе Виртуал Бокс.

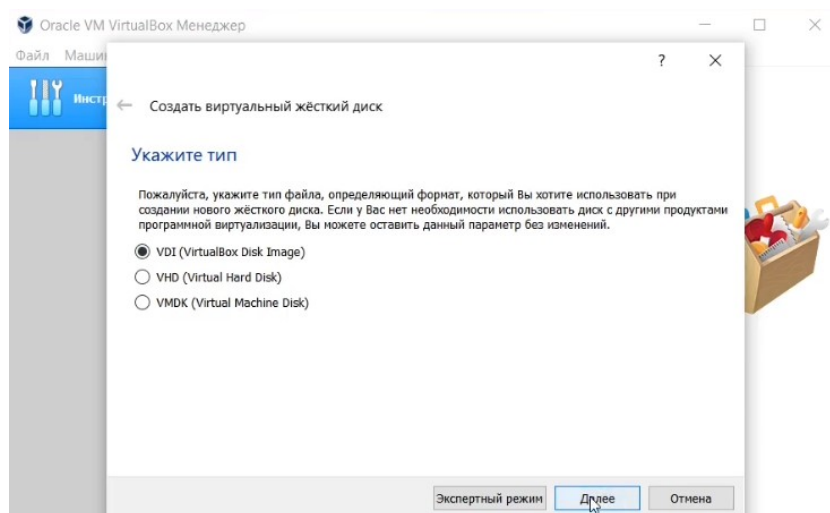


Рис. 4.16: Создание виртуальной машины. VDI

Далее нужно выбрать формат хранения: динамический или фиксированный.

Файл динамического жесткого диска будет занимать небольшое место на физическом жестком диске вашего компьютера. По мере заполнения данными он будет увеличиваться в размере до предельного объема.

Файл фиксированного жесткого диска сразу займет весь объем виртуального жесткого диска. Что касается формата хранения, лучше выбрать «Динамический», так это сэкономит время и место на физическом диске, потеряв, правда, немного в производительности (рис. 4.17).

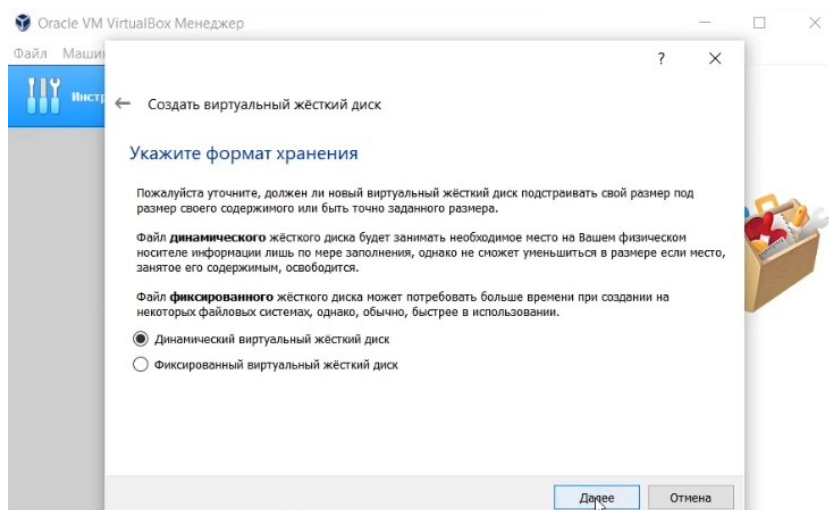


Рис. 4.17: Создание виртуальной машины. Формат хранения.

Размер виртуального диска устанавливаем по желанию, но не меньше, чем указано в требованиях операционной системы. Жмем «Создать». Программа VirtualBox сама рекомендует объем жесткого диска для конкретной операционной системы. Однако стоит выбирать больший объем памяти для установки утилит и дополнительных программ, а также для хранения документов и файлов (рис. 4.18). Для создания нового виртуального диска я нажал на кнопку «Создать».

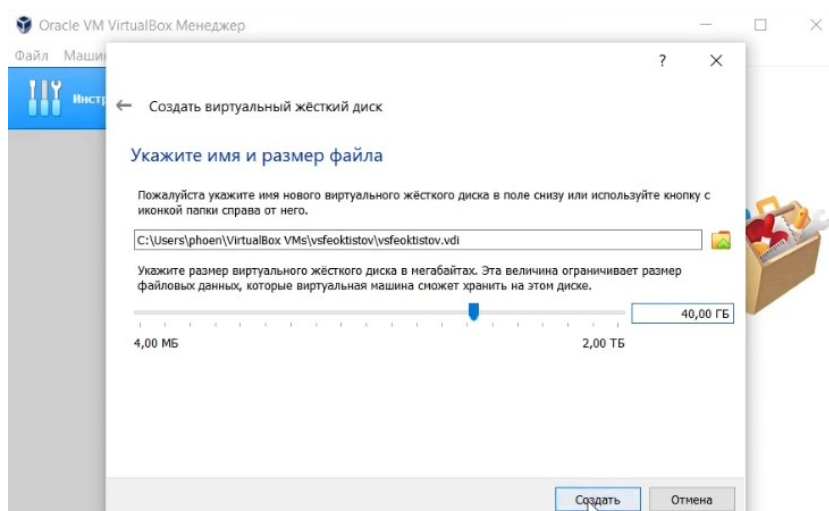


Рис. 4.18: Создание виртуальной машины. Имя и размер файла/виртуальной машины

Первичная настройка виртуальной машины окончена (рис. 4.19).

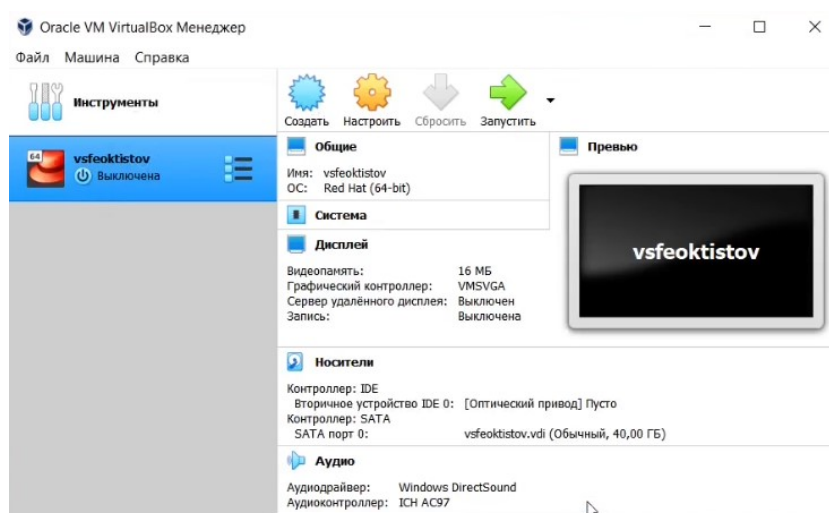


Рис. 4.19: Завершение первичной настройки виртуальной машины

Теперь можно перейти к дополнительной настройке VM. Для этого я перешел в настройки. В разделе «Система» во вкладке «Процессор» можно определить число ядер центрального процессора компьютера, которые могут использоваться виртуальной машиной, а также предельную допустимую нагрузку на процессор.

Рекомендуют выделять не менее двух ядер. Решение о том, позволять ли виртуальной машине загружать процессор на 100% принимается исходя из его производительности. На слабых и старых машинах лучше оставить предлагаемую по умолчанию настройку (рис. 4.20).

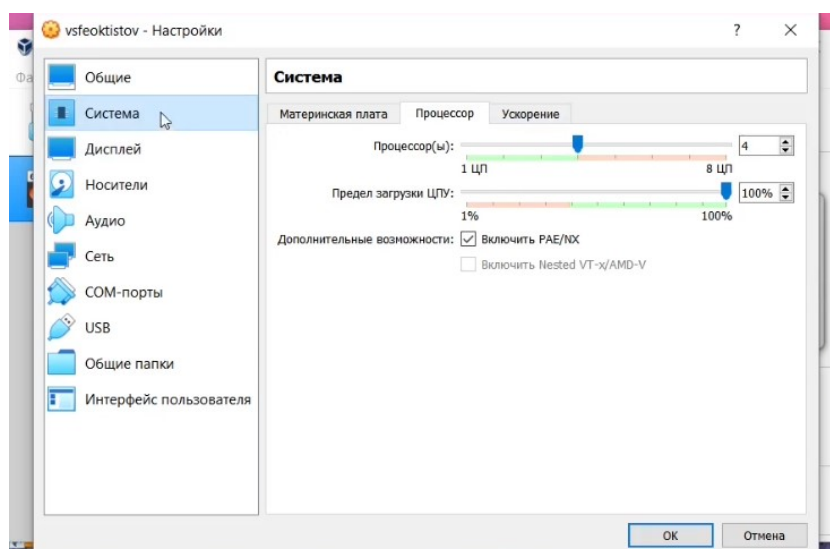


Рис. 4.20: Дополнительная настройка виртуальной машины. Процессор.

Категория «Носители» в настройках виртуальной машины позволяет подключать к виртуальной машине виртуальный жесткий диск, CD/DVD, дискету и диски.

На реальном ПК так называемые «контроллеры хранения» подключают физические диски к остальной части компьютера. Аналогично, VirtualBox предоставляет виртуальные контроллеры виртуальной машины на виртуальную машину. Под каждым контроллером отображаются виртуальные устройства (жесткие диски, CD/DVD или флоппи-дисководы), подключенные к контроллеру.

Если используется мастер «Создать виртуальную машину» для создания машины, обычно можно увидеть следующее:

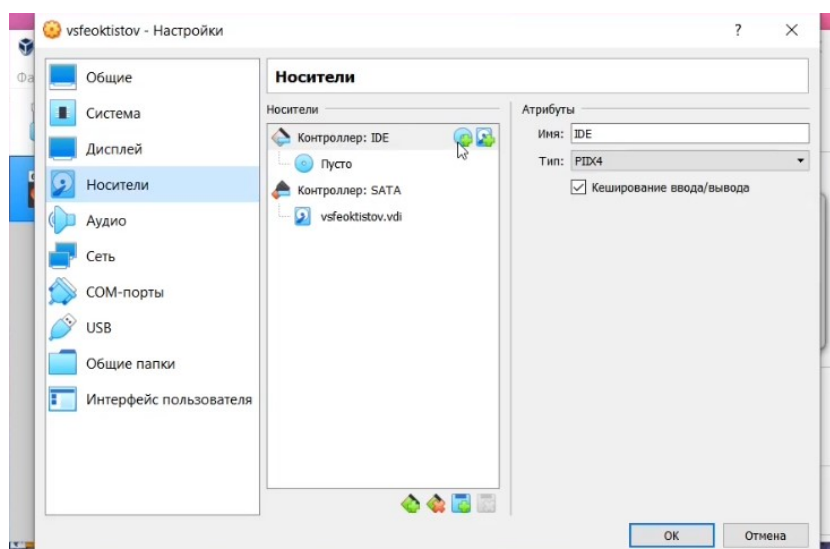


Рис. 4.21: Дополнительная настройка виртуальной машины. Носители.

В зависимости от типа гостевой операционной системы, который вы выбрали при создании виртуальной машины, типичная компоновка устройств хранения в новой виртуальной машине выглядит следующим образом: - Вы увидите контроллер IDE, к которому подключен виртуальный CD/DVD-привод (к порту «вторичный мастер» контроллера IDE). - Вы также увидите контроллер SATA, который является более современным типом контроллера хранения для увеличения пропускной способности жесткого диска, к которому прикреплены виртуальные жесткие диски. Первоначально у вас обычно будет один такой виртуальный диск, но, может быть более одного, каждый из которых представлен файлом образа диска (в этом случае файл VDI).

Я перехожу в контроллер и нажимаю на кнопку «Добавить жесткий диск». Появляется следующее окно:

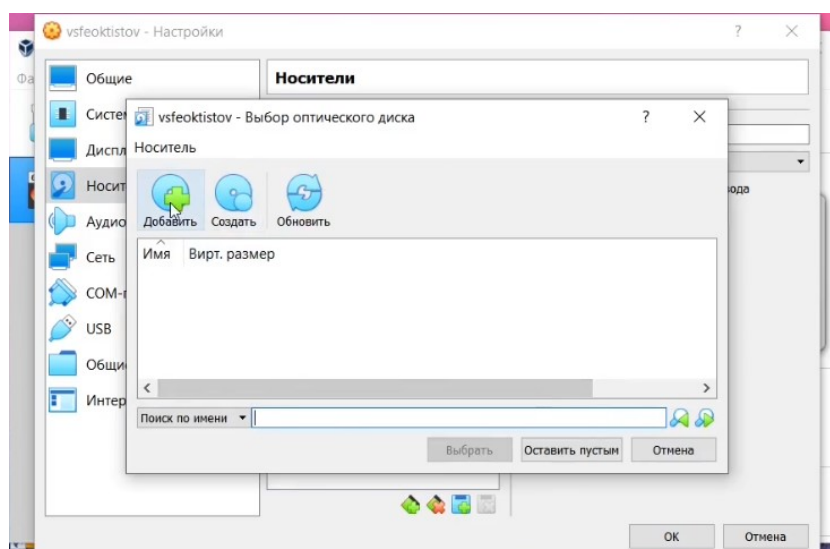


Рис. 4.22: Дополнительная настройка виртуальной машины. Носители. Контроллер.

Нажимаю «Добавить» и указываю путь на образ моей операционной системы (рис. 4.23 и 4.24).

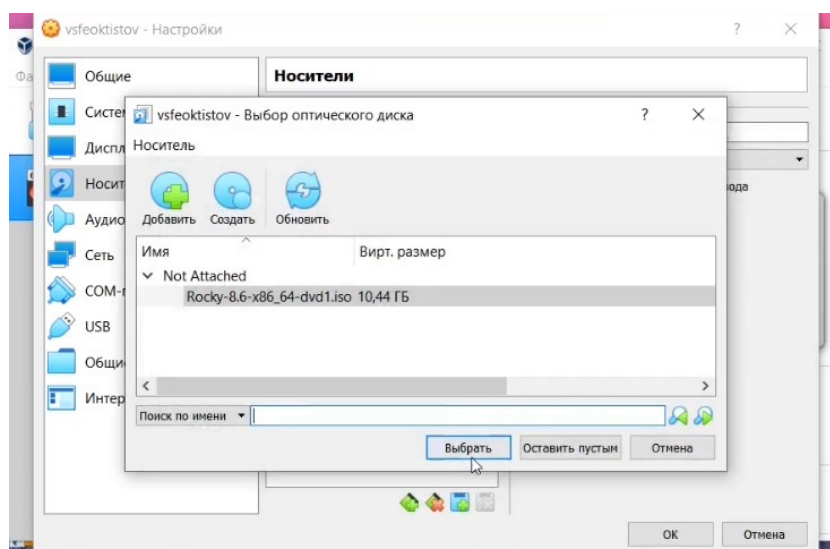


Рис. 4.23: Дополнительная настройка виртуальной машины. Носители. Контроллер.

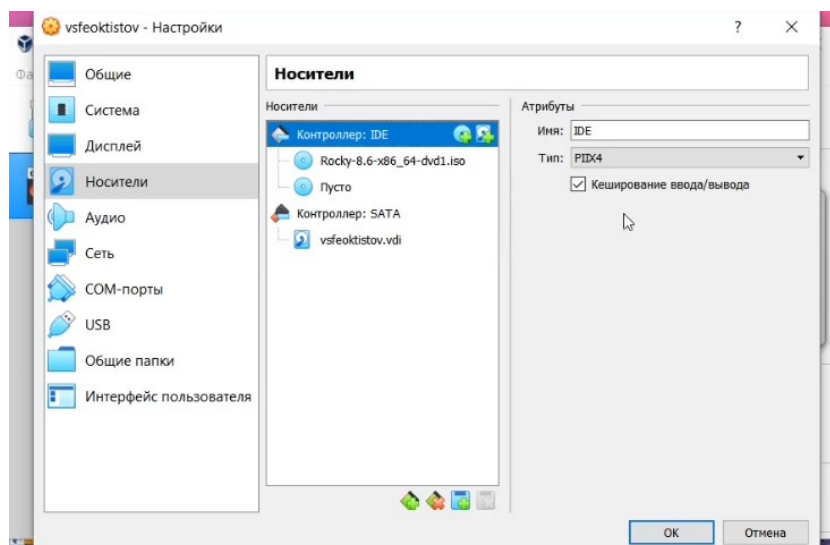


Рис. 4.24: Дополнительная настройка виртуальной машины. Носители. Контроллер.

После всех настроек можно перейти к запуску виртуальной машины. Для этого нужно нажать на кнопку «Запустить», находясь в нужном разделе в списке виртуальных машин. У меня она одна (рис. 4.25).

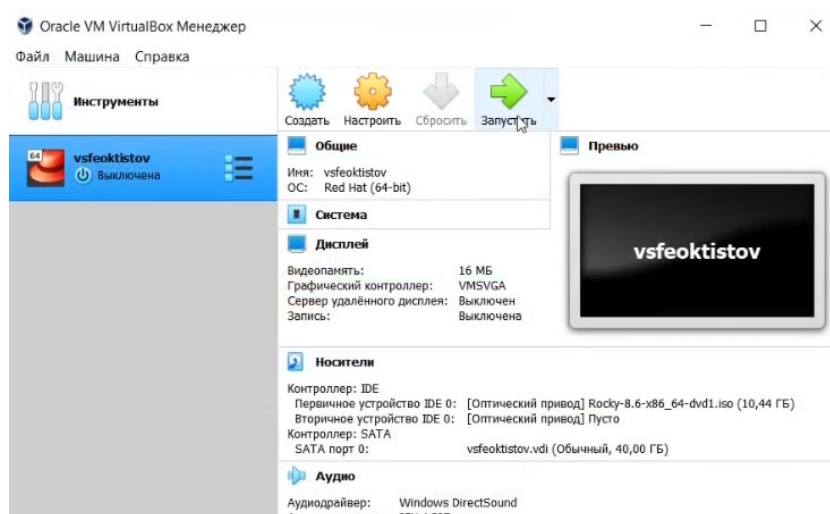


Рис. 4.25: Запуск виртуальной машины

4.3 Установка Rocky Linux

После запуска виртуальной машины отобразится меню носителя. Для перехода к установке нужно выбрать 'Install Rocky Linux 8' [Enter]. Это запустит установщик с графической оболочкой GUI (рис. 4.26)

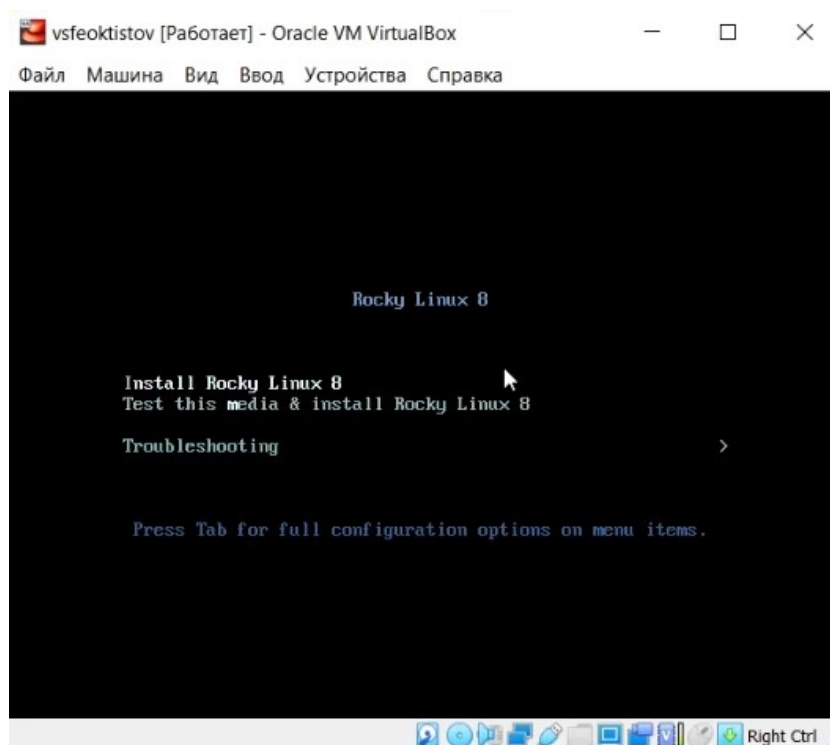


Рис. 4.26: Установка Rocky Linux

Далее необходимо выбрать язык системы. Выбирайте наиболее подходящий для вас настройки. Я оставил "English" (рис. 4.27)

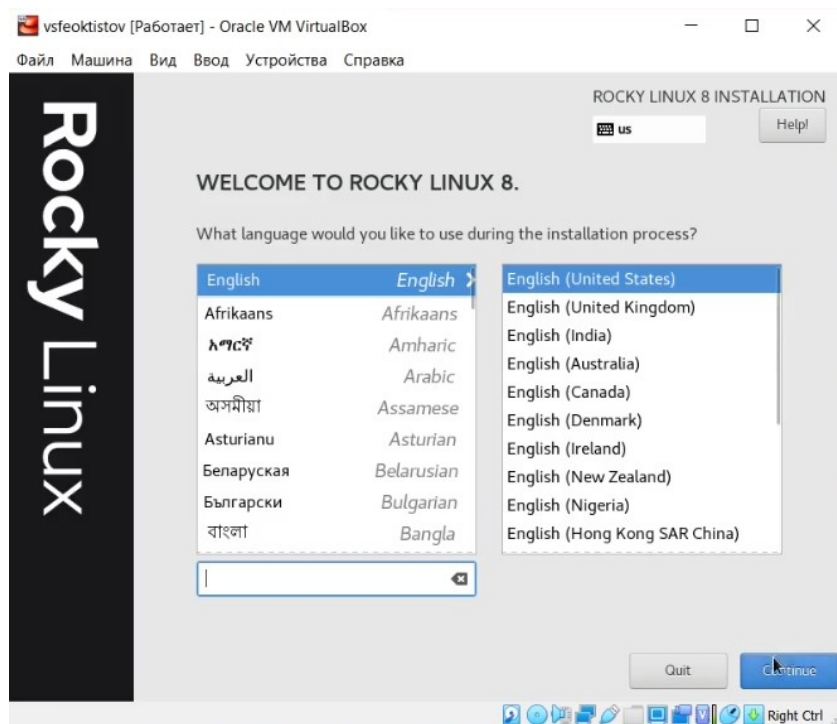


Рис. 4.27: Установка Rocky Linux. Выбор языка

После появится графическое конфигурационное окно с настройками для будущей системы (рис. 4.28).

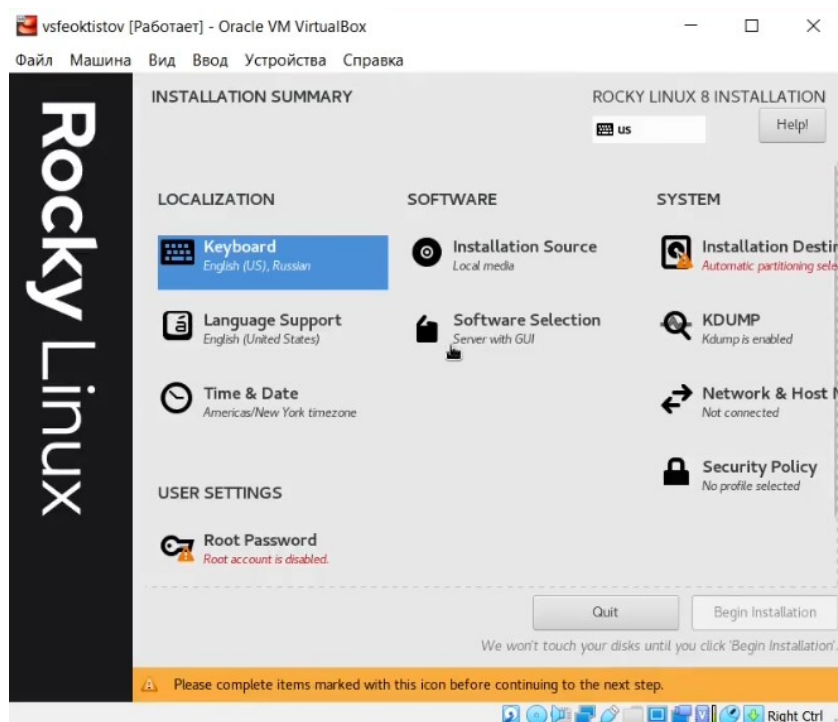


Рис. 4.28: Установка Rocky Linux. Конфигурационное окно

При необходимости можно скорректировать часовой пояс, раскладку клавиатуры (я добавил русский язык, но в качестве языка по умолчанию оставил английский).

В разделе выбора программ в качестве базового окружения оставил Server with GUI, а в качестве дополнения - Development Tools (рис. 4.29)

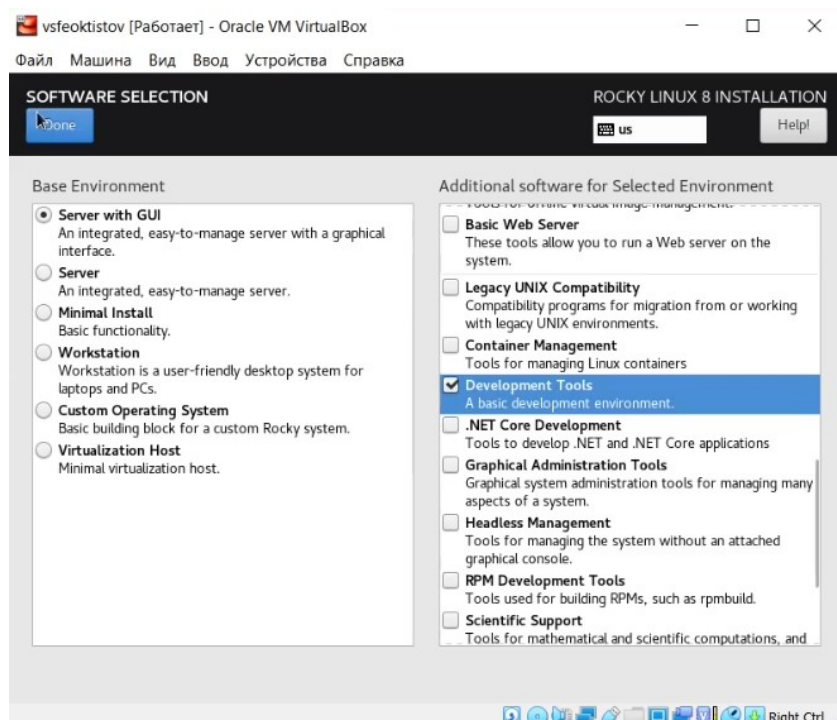


Рис. 4.29: Установка Rocky Linux. Выбор программ

Место установки ОС оставляю без изменения (рис. 4.30)

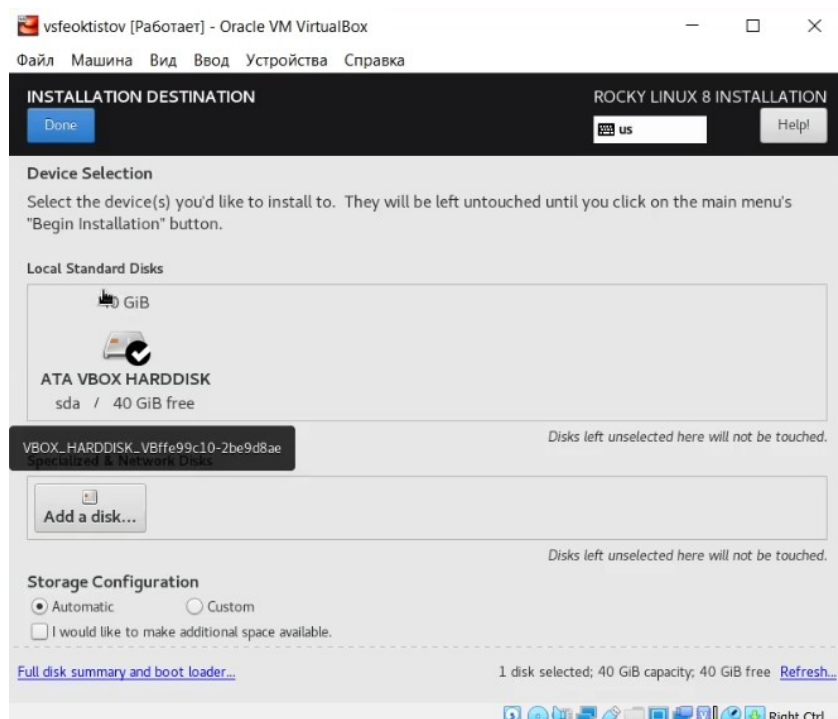


Рис. 4.30: Установка Rocky Linux. Место установки ОС

Далее отключаю KDUMP (рис. 4.31)

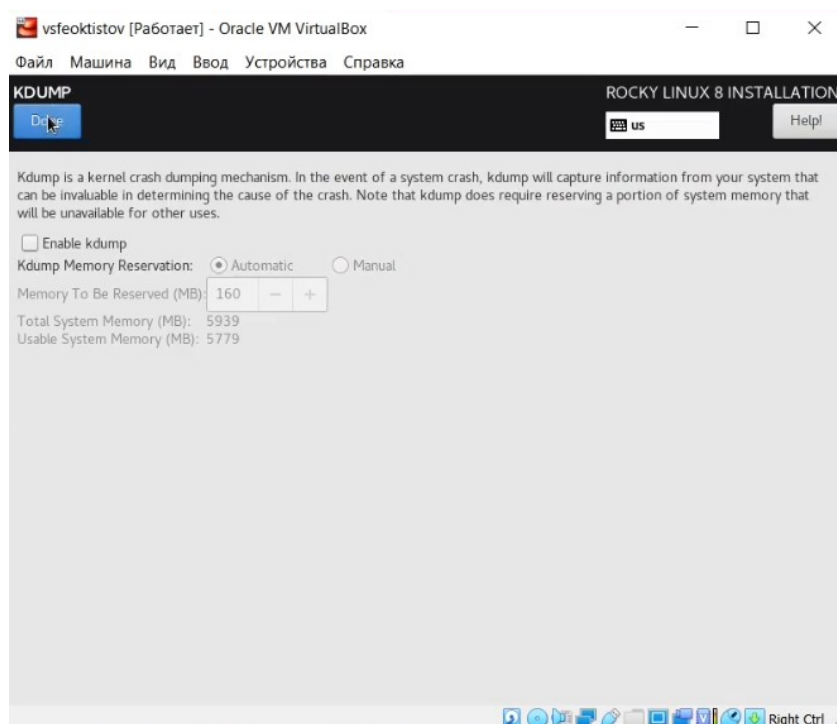


Рис. 4.31: Установка Rocky Linux. KDUMP

Включаю сетевое соединение и в качестве имени узла указываю vsfeoktistov.localdomain (рис. 4.32).

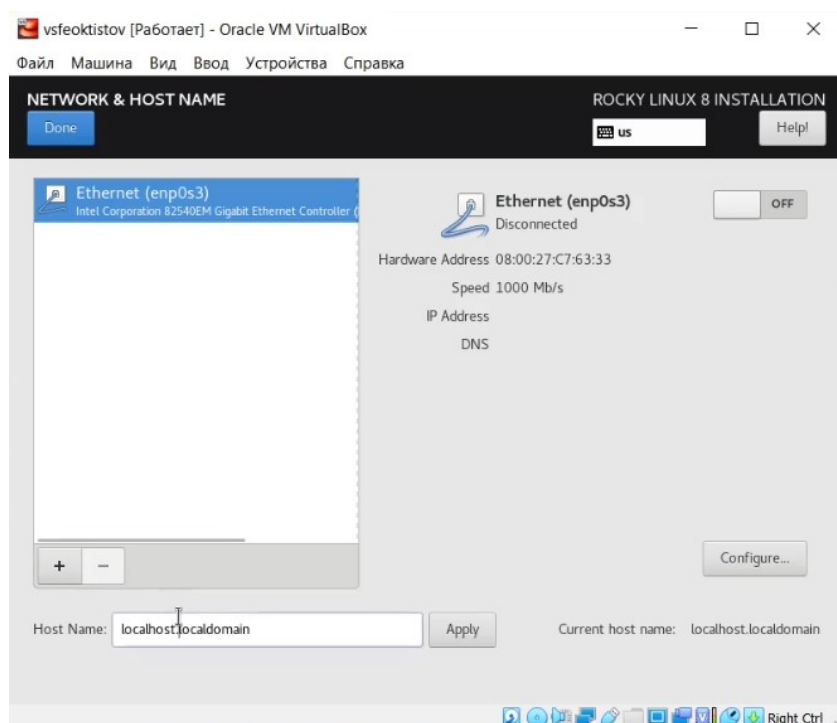


Рис. 4.32: Установка Rocky Linux. Network & host name

Устанавливаю пароль для root и пользователя с правами администратора (рис. 4.33 и 4.34)

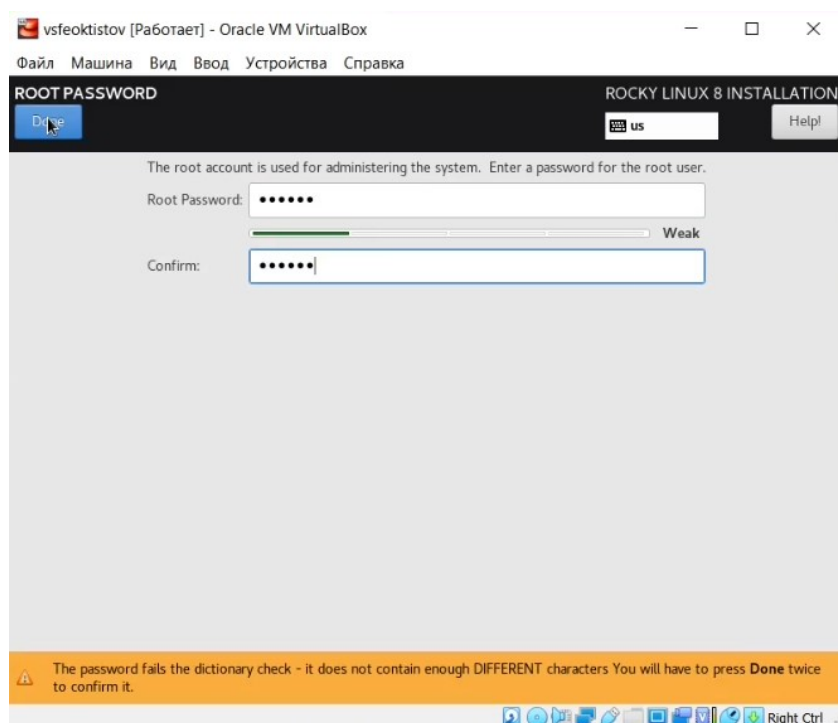


Рис. 4.33: Установка Rocky Linux. Пароль для root пользователя

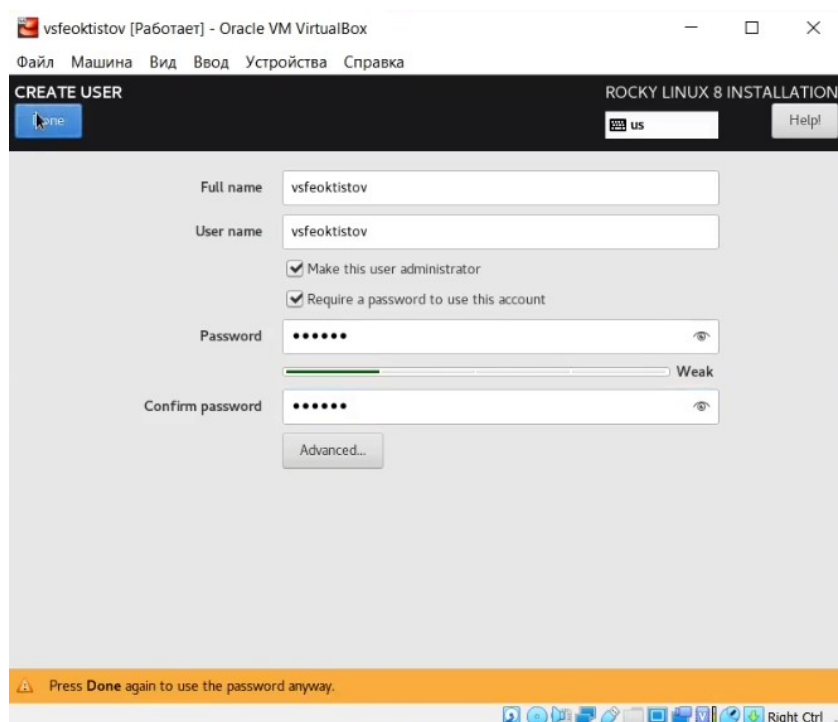


Рис. 4.34: Установка Rocky Linux. Создание пользователя с правами

Далее нажимаю на кнопку «Begin installation». Установка займет некоторое время (рис. 4.35).

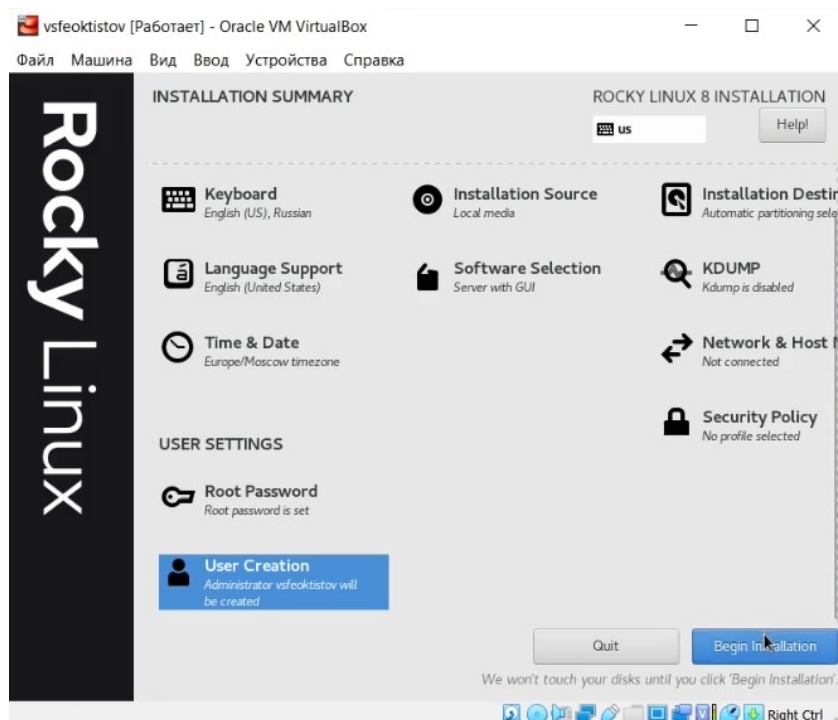


Рис. 4.35: Установка Rocky Linux. Завершение настройки

После завершения установки операционной системы нужно будет нажать на кнопку «Reboot System» для корректного перезапуска виртуальной машины (рис. 4.36).

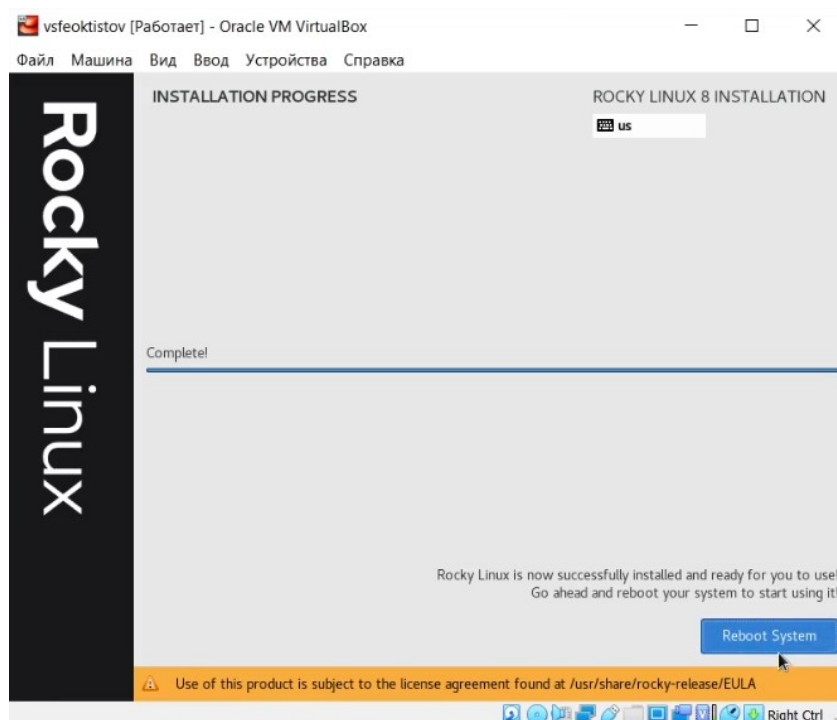


Рис. 4.36: Установка Rocky Linux. Завершение загрузки

После перезапуска нужно принять лицензионное соглашение (рис. 4.37 и 4.38).

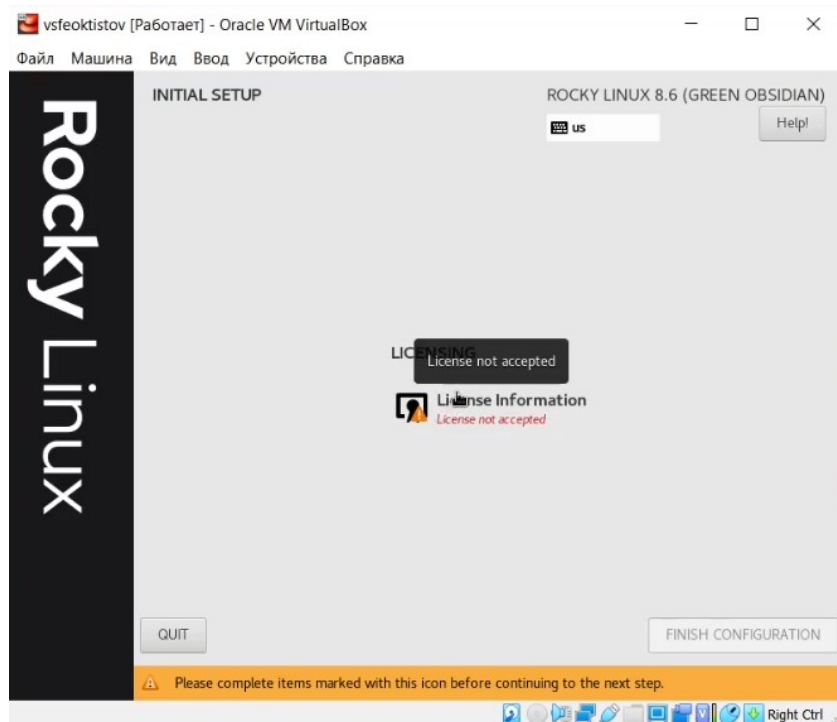


Рис. 4.37: Лицензионное соглашение

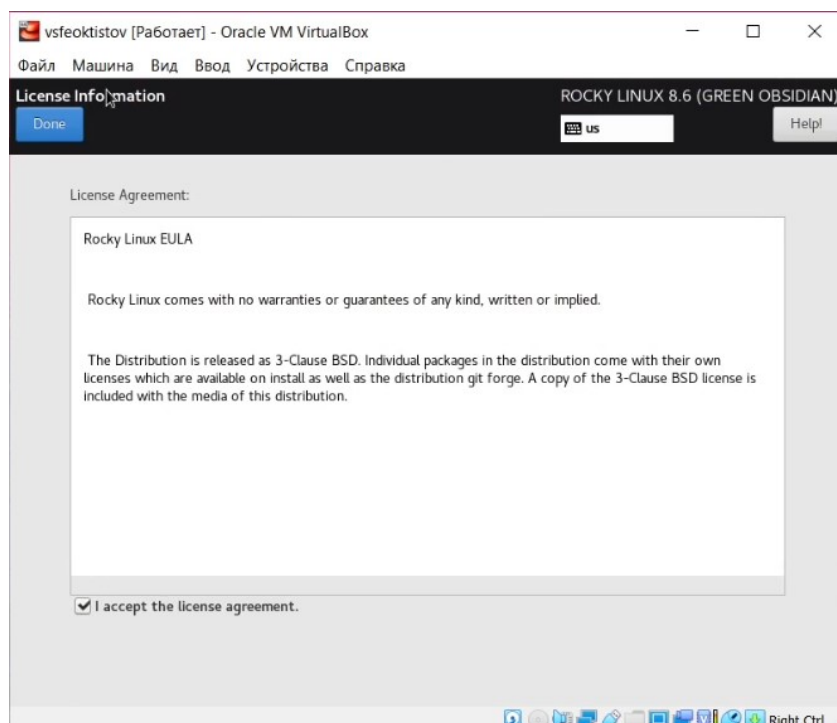


Рис. 4.38: Лицензионное соглашение

Теперь можно будет войти в ОС под заданной при установке учетной записью (рис. 4.39).

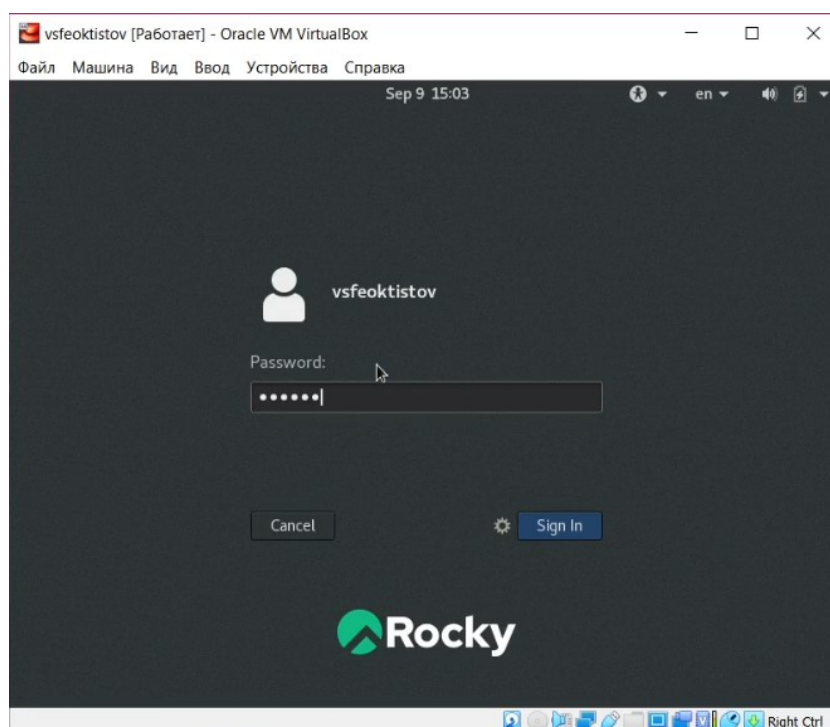


Рис. 4.39: Вход в учетную запись

На этом установка Rocky Linux 8 версии завершена.

4.4 Подключение гостевой ОС

В меню «Устройства» виртуальной машины подключаю образ диска дополнительный гостевой ОС (рис. 4.40).

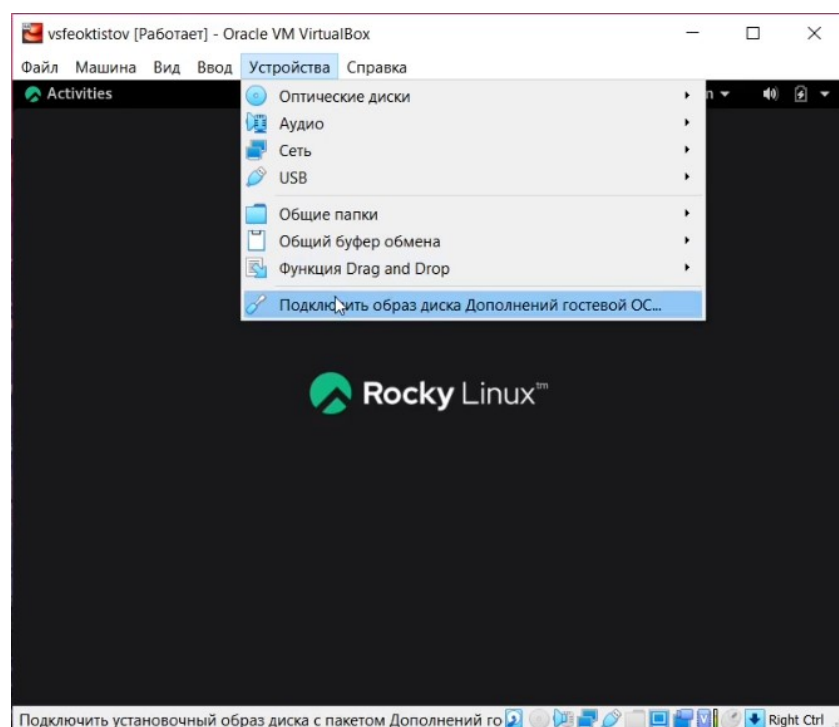


Рис. 4.40: Подключение образа диска Дополнений гостевой ОС

После чего появится диалоговое окно, в котором нужно нажать на кнопку «Run» (рис. рис. 4.41). Нужно будет также ввести пароль для подтверждения своих действий.

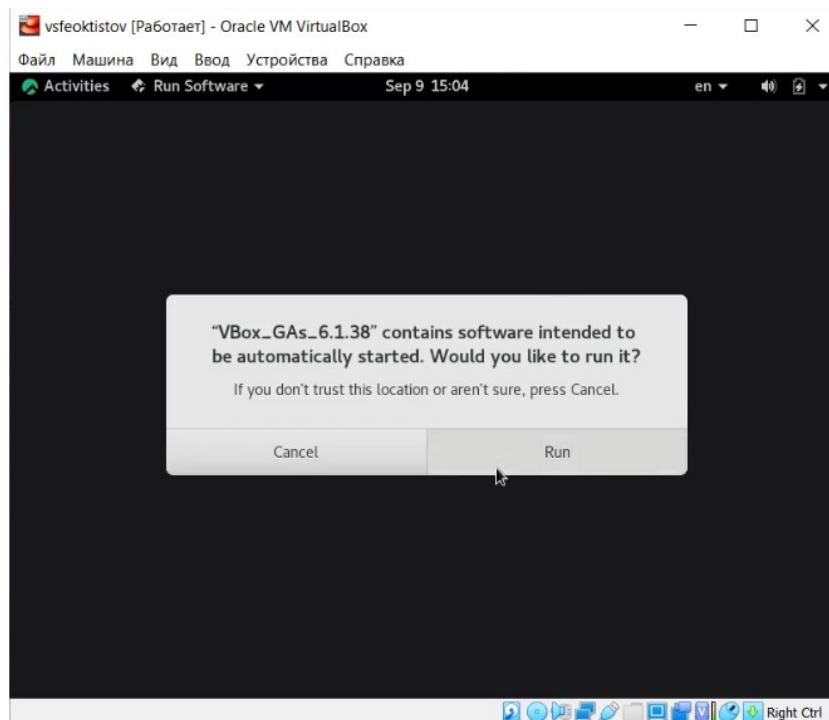


Рис. 4.41: Подключение образа диска Дополнений гостевой ОС. Диалоговое окно

Ожидаем завершение установки, для этого нужно подождать пока появится сообщение «Press Return to close this window...» (рис. 4.42). Нажимаю [Enter] и перезапускаю виртуальную машину.

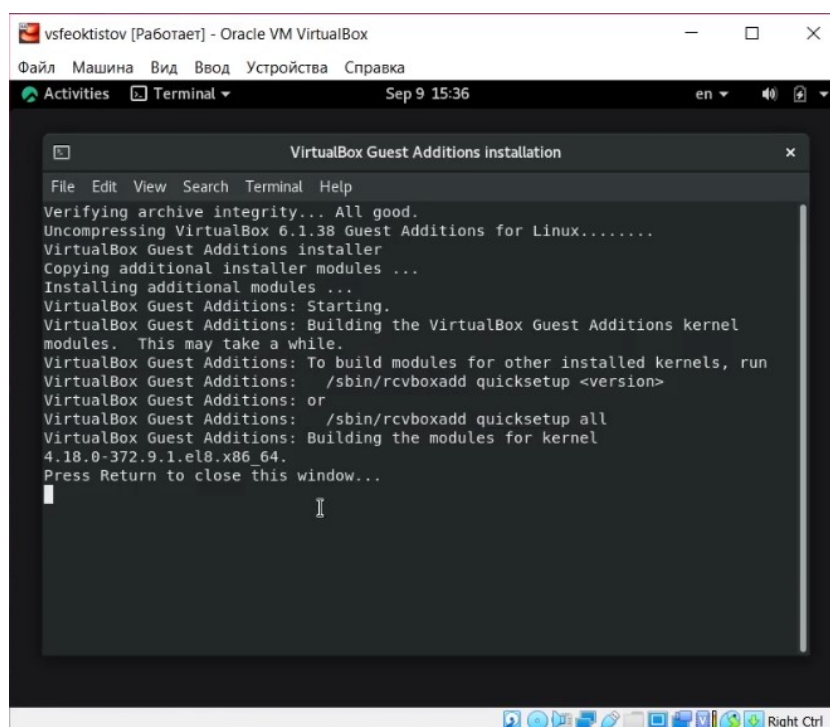


Рис. 4.42: Подключение образа диска Дополнений гостевой ОС. Завершение установки

После перезапуска и повторного входа в систему можно будет расширить визуализированный экран ОС до размера экрана. Для этого нажимаю на вкладку «Вид» в виртуальной машине и выбираю опцию «Режим полного экрана» (рис. 4.43).

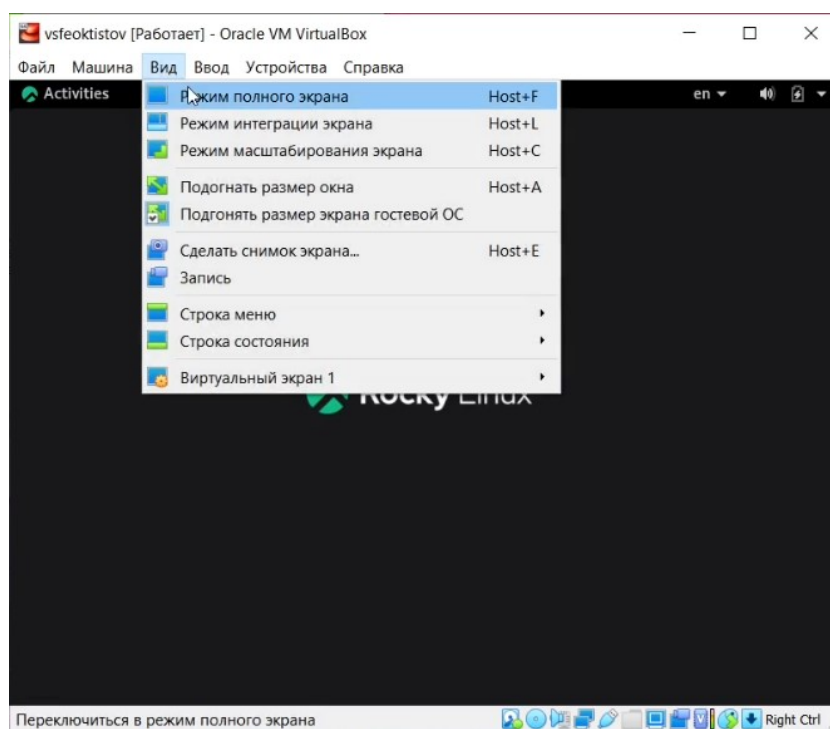


Рис. 4.43: Режим полного экрана

На этом подключение образа диска дополнений гостевой ОС завершено.

5 Выводы

Я приобрел практические навыки установки операционной системы на виртуальную машину и настроил минимально необходимые для дальнейшей работы сервисы. Также я узнал необходимую мне информацию о моей виртуальной системе через терминал.

6 Контрольные вопросы

1. Когда пользователь регистрируется в системе (проходит процедуру авторизации, например, вводя системное имя и пароль), он идентифицируется с учётной записью, в которой система хранит информацию о каждом пользователе: его системное имя и некоторые другие сведения, необходимые для работы с ним. Именно с учётными записями, а не с самими пользователями, и работает система. Таким образом, учетная запись пользователя содержит:

- 1) Системное имя (user name)
- 2) Идентификатор пользователя (UID)
- 3) Идентификатор группы (GID)
- 4) Полное имя (full name)
- 5) Домашний каталог (home directory)
- 6) Начальная оболочка (login shell)

2. Команды терминала:

- 1) `man [аргумент]` – для получения справки по команде. Пример: `man ls` – выведет информацию о команде `ls`;
- 2) `cd [путь]` – для перемещения по файловой системе. Пример: `cd /` – для перехода в корневой раздел;
- 3) `ls [опции]` – для просмотра содержимого каталога. Пример: `ls -alS /` – выведет подробной информации о файлах в корневом каталоге с сортировкой по алфавиту;
- 4) `du [опции] [путь]` – для определения объёма каталога. Пример: `du -h`

~/”Изображения” – выведет размер каталога «Изображения» с указанием единицы измерения;

- 5) `mkdir [опции] [путь] / rmdir [опции] [путь] / rm [опции] [путь]` – для создания / удаления каталогов / файлов. Примеры: `mkdir -pv ~/MyDir ~/MyDir2` – создаст каталоги MyDir и MyDir2 в домашней директории; `rmdir -v ~/MyDir` – удалит каталог MyDir из домашней директории; `rm -rv ~/MyDir2` – удалит каталог MyDir2 из домашней директории;
- 6) `chmod [аргументы] [путь]` - для задания определённых прав на файл / каталог. Пример: `chmod o-w ~/Readme.txt` – отнимет право изменения текстового файла Readme.txt остальным пользователям, кроме владельца и группы владельца.
- 7) `history [опции]` – для просмотра истории команд. Пример: `history -c` – очистит историю команд Linux.

3. Файловая система - порядок, определяющий способ организации, хранения и именования данных на носителях информации в компьютерах, а также в другом электронном оборудовании. Файловые системы в Linux используются не только для работы с файлами на диске, но и для хранения данных в оперативной памяти или доступа к конфигурации ядра во время работы системы. Каждый дистрибутив Linux позволяет использовать одну из этих файловых систем, каждая из них имеет свои преимущества и недостатки:

- Ext2;
- Ext3;
- Ext4;
- JFS;
- ReiserFS;
- XFS;
- Btrfs;
- ZFS; Все они включены в ядро и могут использоваться в качестве корневой файловой системы.

Ext2, Ext3, Ext4 или Extended Filesystem – это стандартная файловая система для Linux. Она была разработана еще для Minix. Она самая стабильная из всех существующих. Её кодовая база изменяется очень редко. Кроме того, данная файловая система содержит больше всего функций. Версия ext2 была разработана уже именно для Linux и получила много улучшений.

ReiserFS – была разработана намного позже, в качестве альтернативы ext3 с улучшенной производительностью и расширенными возможностями. Она была разработана под руководством Ганса Райзера и поддерживает только Linux. Из особенностей можно отметить динамический размер блока, что позволяет упаковывать несколько небольших файлов в один блок, что предотвращает фрагментацию и улучшает работу с небольшими файлами.

JFS или Journaled File System была разработана в IBM для AIX UNIX и использовалась в качестве альтернативы для файловых систем ext. Сейчас она используется там, где необходима высокая стабильность и минимальное потребление ресурсов. При разработке файловой системы ставилась цель создать максимально эффективную файловую систему для многопроцессорных компьютеров. Также как и ext, это журналируемая файловая система, но в журнале хранятся только метаданные, что может привести к использованию старых версий файлов после сбоев.

XFS – это высокопроизводительная файловая система, разработанная в Silicon Graphics для собственной операционной системы еще в 2001 году. Она изначально была рассчитана на файлы большого размера, и поддерживала диски до 2 Терабайт. Из преимуществ файловой системы можно отметить высокую скорость работы с большими файлами, отложенное выделение места, увеличение разделов на лету и незначительный размер служебной информации.

ZFS – журналируемая файловая система, однако в отличие от ext, в журнал записываются только изменения метаданных. Она используется по

умолчанию в дистрибутивах на основе Red Hat. Из недостатков – это невозможность уменьшения размера, сложность восстановления данных и риск потери файлов при записи, если будет неожиданное отключение питания, поскольку большинство данных находится в памяти BTRFS или B-Tree File System - это совершенно новая файловая система, которая сосредоточена на отказоустойчивости, легкости администрирования и восстановления данных. Файловая система объединяет в себе очень много новых интересных возможностей, таких как размещение на нескольких разделах, поддержка подтомов, изменение размера на лету, создание мгновенных снимков, а также высокая производительность. Но многими пользователями файловая система Btrfs считается нестабильной. Тем не менее, она уже используется как файловая система по умолчанию в OpenSUSE и SUSE Linux.

4. Команда `findmnt` используется для поиска примонтированных файловых систем. Она используется для поиска монтированных устройств, а также может монтировать или размонтировать их при необходимости. Для просмотра всех примонтированных файловых систем использовать команду:
`findmnt -all`
5. Каждый процесс в Linux имеет свой идентификатор, называемый PID. Перед тем, как выполнить остановку процесса, нужно определить его PID. Для этого воспользуемся командами `ps` и `grep`. Команда `ps` предназначена для вывода списка активных процессов в системе и информации о них. Команда `grep` запускается одновременно с `ps` (в канале) и будет выполнять поиск по результатам команды `ps`.

Есть еще один более простой способ узнать PID процесса — это команда `pidof`, которая принимает в качестве параметра название процесса и выводит его PID.

Когда известен PID процесса, мы можем убить его командой `kill`. Команда `kill` принимает в качестве параметра PID процесса.

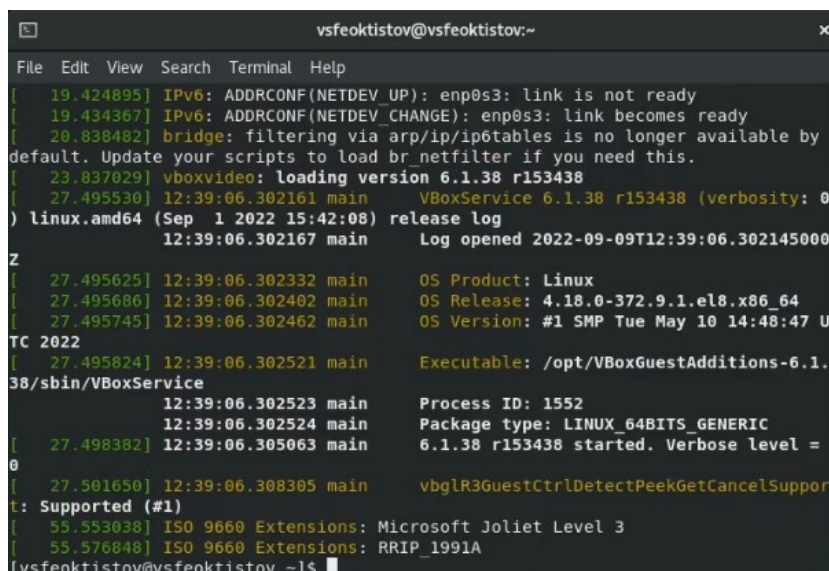
Вообще команда `kill` предназначена для послыки сигнала процессу. По умолчанию, если мы не указываем какой сигнал посылать, посылается сигнал `SIGTERM` (от слова *termination* — завершение). `SIGTERM` указывает процессу на то, что необходимо завершиться. Каждый сигнал имеет свой номер. `SIGTERM` имеет номер 15. Список всех сигналов (и их номеров), которые может послать команда `kill`, можно вывести, выполнив `kill -l`.

Сигнал `SIGTERM` может и не остановить процесс (например, при перехвате или блокировке сигнала), `SIGKILL` же выполняет уничтожение процесса всегда, так как его нельзя перехватить или проигнорировать.

Команда `killall` в Linux предназначена для «убийства» всех процессов, имеющих одно и то же имя. Это удобно, так как нам не нужно знать PID процесса.

7 Домашняя работа

После всех проделанных действий я проанализировал последовательность загрузки системы с помощью команды `dmesg` (рис. 7.1). При использовании этой команды выводится огромный текст с системной информацией.



```
vsfeoktistov@vsfeoktistov:~
File Edit View Search Terminal Help
[ 19.424895] IPv6: ADDRCONF(NETDEV_UP): enp0s3: link is not ready
[ 19.434367] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
[ 20.838482] bridge: filtering via arp/ip/ip6tables is no longer available by
default. Update your scripts to load br_netfilter if you need this.
[ 23.837029] vboxvideo: loading version 6.1.38 r153438
[ 27.495530] 12:39:06.302161 main      VBoxService 6.1.38 r153438 (verbosity: 0
) linux.amd64 (Sep  1 2022 15:42:08) release log
12:39:06.302167 main      Log opened 2022-09-09T12:39:06.302145000
Z
[ 27.495625] 12:39:06.302332 main      OS Product: Linux
[ 27.495686] 12:39:06.302402 main      OS Release: 4.18.0-372.9.1.el8.x86_64
[ 27.495745] 12:39:06.302462 main      OS Version: #1 SMP Tue May 10 14:48:47 U
TC 2022
[ 27.495824] 12:39:06.302521 main      Executable: /opt/VBoxGuestAdditions-6.1.
38/sbin/VBoxService
12:39:06.302523 main      Process ID: 1552
12:39:06.302524 main      Package type: LINUX_64BITS_GENERIC
[ 27.498382] 12:39:06.305063 main      6.1.38 r153438 started. Verbose level =
0
[ 27.501650] 12:39:06.308305 main      vbglR3GuestCtrlDetectPeekGetCancelSuppor
t: Supported (#1)
[ 55.553038] ISO 9660 Extensions: Microsoft Joliet Level 3
[ 55.576848] ISO 9660 Extensions: RRIP_1991A
[vsfeoktistov@vsfeoktistov ~]$
```

Рис. 7.1: Утилита `dmesg`

Для построчного просмотра последовательность загрузки системы можно использовать команду `dmesg | less` (рис. 7.2).

```
vsfeoktistov@vsfeoktistov:~  
File Edit View Search Terminal Help  
[ 0.000000] Linux version 4.18.0-372.9.1.el8.x86_64 (mockbuild@da11-prod-build-  
der001.bld.equ.rockylinux.org) (gcc version 8.5.0 20210514 (Red Hat 8.5.0-10) (G  
CC)) #1 SMP Tue May 10 14:48:47 UTC 2022  
[ 0.000000] Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-4.18.0-372.9.1.el8.  
x86_64 root=/dev/mapper/rl-root ro resume=/dev/mapper/rl-swap rd.lvm.lv=rl/root  
rd.lvm.lv=rl/swap rhgb quiet  
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point regi  
sters'  
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'  
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'  
[ 0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256  
[ 0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes,  
using 'standard' format.  
[ 0.000000] signal: max sigframe size: 1776  
[ 0.000000] BIOS-provided physical RAM map:  
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable  
[ 0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved  
[ 0.000000] BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved  
[ 0.000000] BIOS-e820: [mem 0x0000000000100000-0x0000000000dfffff] usable  
[ 0.000000] BIOS-e820: [mem 0x0000000000dfff0000-0x0000000000dfffffff] ACPI data  
[ 0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec0ffff] reserved  
[ 0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee0ffff] reserved  
[ 0.000000] BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved  
:
```

Рис. 7.2: Утилита dmesg с использованием less

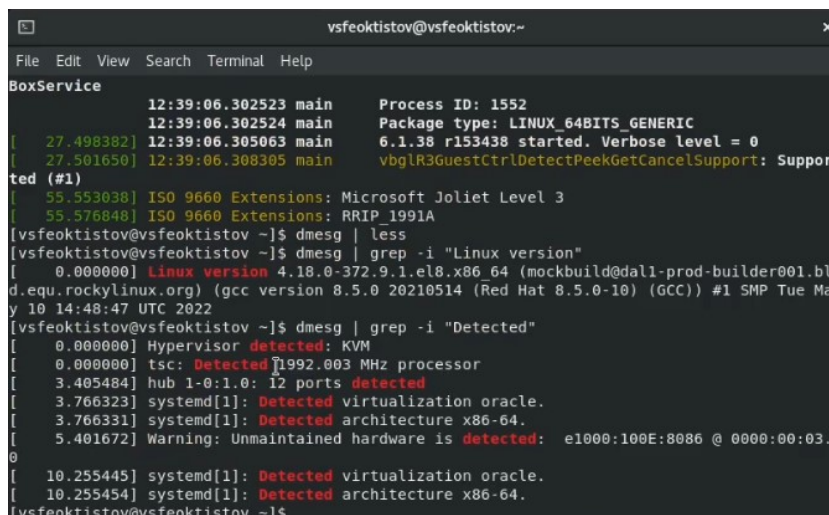
Производить поиск по такому тексту неудобно, поэтому я использовал команду dmesg с дополнительными аргументами: dmesg | grep -i “ключевые слова”.

Для того, чтобы найти информацию о версии ОС, я использовал команду dmesg | grep -i “Linux version”. Из рисунка 7.3 видно, что версия моей ОС – RedHat 8.5.0-10 (в начале видно, что стоит 64 разрядная система, а в скобках rockylinux).

```
vsfeoktistov@vsfeoktistov:~  
File Edit View Search Terminal Help  
[ 27.495530] 12:39:06.302161 main VBoxService 6.1.38 r153438 (verbosity: 0  
) linux.amd64 (Sep 1 2022 15:42:08) release log  
[ 27.495625] 12:39:06.302332 main OS Product: Linux  
[ 27.495686] 12:39:06.302402 main OS Release: 4.18.0-372.9.1.el8.x86_64  
[ 27.495745] 12:39:06.302462 main OS Version: #1 SMP Tue May 10 14:48:47 U  
TC 2022  
[ 27.495824] 12:39:06.302521 main Executable: /opt/VBoxGuestAdditions-6.1.  
38/sbin/VBoxService  
[ 27.495824] 12:39:06.302523 main Process ID: 1552  
[ 27.495824] 12:39:06.302524 main Package type: LINUX_64BITS_GENERIC  
[ 27.495824] 12:39:06.305063 main 6.1.38 r153438 started. Verbose level =  
0  
[ 27.501650] 12:39:06.308305 main vbglR3GuestCtrlDetectPeekGetCancelSuppor  
t: Supported (#1)  
[ 55.553038] ISO 9660 Extensions: Microsoft Joliet Level 3  
[ 55.576848] ISO 9660 Extensions: RRIP_1991A  
[vsfeoktistov@vsfeoktistov ~]$ dmesg | less  
[vsfeoktistov@vsfeoktistov ~]$ dmesg | grep -i "Linux version"  
[ 0.000000] Linux version 4.18.0-372.9.1.el8.x86_64 (mockbuild@da11-prod-build-  
der001.bld.equ.rockylinux.org) (gcc version 8.5.0 20210514 (Red Hat 8.5.0-10) (G  
CC)) #1 SMP Tue May 10 14:48:47 UTC 2022  
[vsfeoktistov@vsfeoktistov ~]$
```

Рис. 7.3: Версия линукса

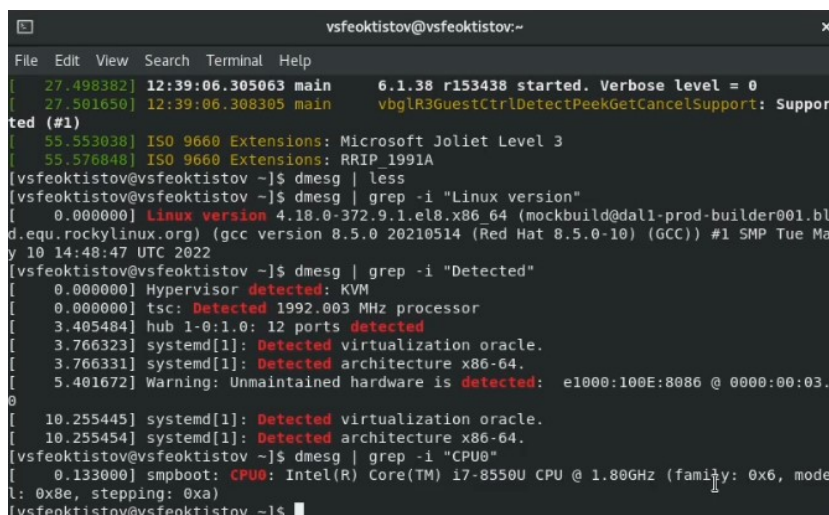
Для поиска частоты процессора при запуске использовал команду `dmesg | grep -i "Detected"`. Из рисунка 7.4 видно, что частота моего процессора на момент запуска составляла 1992.003 МГц.



```
vsfeoktistov@vsfeoktistov:~  
File Edit View Search Terminal Help  
BoxService  
[ 27.498382] 12:39:06.302523 main Process ID: 1552  
[ 27.501650] 12:39:06.302524 main Package type: LINUX_64BITS_GENERIC  
[ 27.501650] 12:39:06.305063 main 6.1.38 r153438 started. Verbose level = 0  
[ 27.501650] 12:39:06.308305 main vbgLR3GuestCtrlDetectPeekGetCancelSupport: Support  
ted (#1)  
[ 55.553038] ISO 9660 Extensions: Microsoft Joliet Level 3  
[ 55.576848] ISO 9660 Extensions: RRIP_1991A  
[vsfeoktistov@vsfeoktistov ~]$ dmesg | less  
[vsfeoktistov@vsfeoktistov ~]$ dmesg | grep -i "Linux version"  
[ 0.000000] Linux version 4.18.0-372.9.1.el8.x86_64 (mockbuild@dal1-prod-builder001.bl  
d.equ.rockylinux.org) (gcc version 8.5.0 20210514 (Red Hat 8.5.0-10) (GCC)) #1 SMP Tue Ma  
y 10 14:48:47 UTC 2022  
[vsfeoktistov@vsfeoktistov ~]$ dmesg | grep -i "Detected"  
[ 0.000000] Hypervisor detected: KVM  
[ 0.000000] tsc: Detected 1992.003 MHz processor  
[ 3.405484] hub 1-0:1.0: 12 ports detected  
[ 3.766323] systemd[1]: Detected virtualization oracle.  
[ 3.766331] systemd[1]: Detected architecture x86-64.  
[ 5.401672] Warning: Unmaintained hardware is detected: e1000:100E:8086 @ 0000:00:03.  
0  
[ 10.255445] systemd[1]: Detected virtualization oracle.  
[ 10.255454] systemd[1]: Detected architecture x86-64.  
[vsfeoktistov@vsfeoktistov ~]$
```

Рис. 7.4: Частота процессора

Команда `dmesg | grep -i "CPU0"` вывела информацию о модели процессора. У меня это Intel(R) Core(TM) i7-8550U @ 1.80GHz (рис. 7.5).

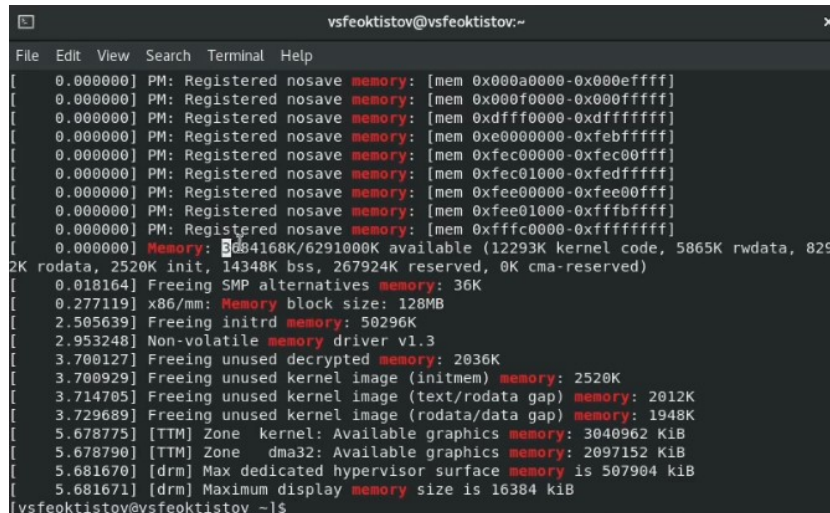


```
vsfeoktistov@vsfeoktistov:~  
File Edit View Search Terminal Help  
[ 27.498382] 12:39:06.305063 main 6.1.38 r153438 started. Verbose level = 0  
[ 27.501650] 12:39:06.308305 main vbgLR3GuestCtrlDetectPeekGetCancelSupport: Support  
ted (#1)  
[ 55.553038] ISO 9660 Extensions: Microsoft Joliet Level 3  
[ 55.576848] ISO 9660 Extensions: RRIP_1991A  
[vsfeoktistov@vsfeoktistov ~]$ dmesg | less  
[vsfeoktistov@vsfeoktistov ~]$ dmesg | grep -i "Linux version"  
[ 0.000000] Linux version 4.18.0-372.9.1.el8.x86_64 (mockbuild@dal1-prod-builder001.bl  
d.equ.rockylinux.org) (gcc version 8.5.0 20210514 (Red Hat 8.5.0-10) (GCC)) #1 SMP Tue Ma  
y 10 14:48:47 UTC 2022  
[vsfeoktistov@vsfeoktistov ~]$ dmesg | grep -i "Detected"  
[ 0.000000] Hypervisor detected: KVM  
[ 0.000000] tsc: Detected 1992.003 MHz processor  
[ 3.405484] hub 1-0:1.0: 12 ports detected  
[ 3.766323] systemd[1]: Detected virtualization oracle.  
[ 3.766331] systemd[1]: Detected architecture x86-64.  
[ 5.401672] Warning: Unmaintained hardware is detected: e1000:100E:8086 @ 0000:00:03.  
0  
[ 10.255445] systemd[1]: Detected virtualization oracle.  
[ 10.255454] systemd[1]: Detected architecture x86-64.  
[vsfeoktistov@vsfeoktistov ~]$ dmesg | grep -i "CPU0"  
[ 0.133000] smpboot: CPU0: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz (family: 0x6, mode  
l: 0x8e, stepping: 0xa)  
[vsfeoktistov@vsfeoktistov ~]$
```

Рис. 7.5: Модель процессора

Для того, чтобы посмотреть объем доступной оперативной памяти, я использо-
вал команду `dmesg | grep -i "Memory"`. Из рисунка 7.6 видно, что виртуальной

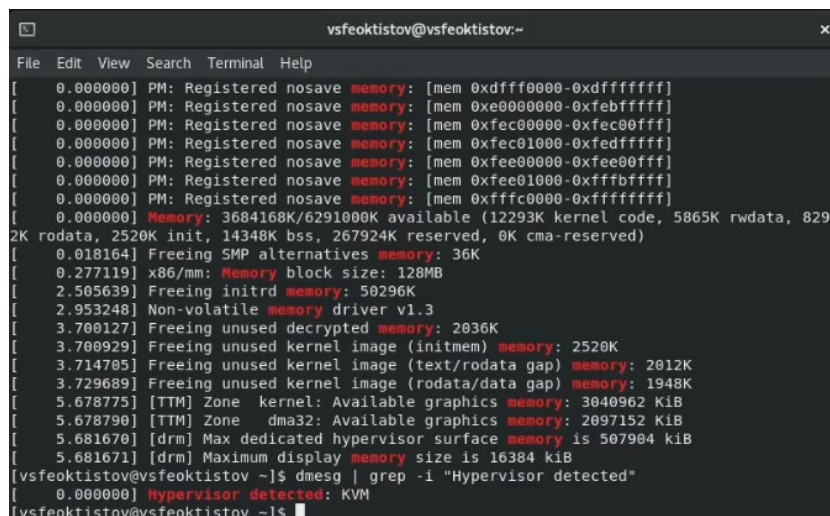
машине доступно 6291000 Кбайт ОЗУ или 6 ГБ ОЗУ. Столько я выделял при создании виртуальной машины в VirtualBox.



```
vsfeoktistov@vsfeoktistov:~  
File Edit View Search Terminal Help  
[ 0.000000] PM: Registered nosave memory: [mem 0x000a0000-0x000effff]  
[ 0.000000] PM: Registered nosave memory: [mem 0x000f0000-0x000fffff]  
[ 0.000000] PM: Registered nosave memory: [mem 0xdfff0000-0xdfffffff]  
[ 0.000000] PM: Registered nosave memory: [mem 0xe0000000-0xfefbffff]  
[ 0.000000] PM: Registered nosave memory: [mem 0xfec00000-0xfec0ffff]  
[ 0.000000] PM: Registered nosave memory: [mem 0xfec01000-0xfedffffff]  
[ 0.000000] PM: Registered nosave memory: [mem 0xfec00000-0xfec0ffff]  
[ 0.000000] PM: Registered nosave memory: [mem 0xfec01000-0xfefbffff]  
[ 0.000000] PM: Registered nosave memory: [mem 0xfec00000-0xfefbffff]  
[ 0.000000] Memory: 3684168K/6291000K available (12293K kernel code, 5865K rwd, 829  
2K rodata, 2520K init, 14348K bss, 267924K reserved, 0K cma-reserved)  
[ 0.018164] Freeing SMP alternatives memory: 36K  
[ 0.277119] x86/mm: Memory block size: 128MB  
[ 2.505639] Freeing initrd memory: 50296K  
[ 2.953248] Non-volatile memory driver v1.3  
[ 3.700127] Freeing unused decrypted memory: 2036K  
[ 3.700929] Freeing unused kernel image (initmem) memory: 2520K  
[ 3.714705] Freeing unused kernel image (text/rodata gap) memory: 2012K  
[ 3.729689] Freeing unused kernel image (rodata/data gap) memory: 1948K  
[ 5.678775] [TTM] Zone kernel: Available graphics memory: 3040962 KiB  
[ 5.678790] [TTM] Zone dma32: Available graphics memory: 2097152 KiB  
[ 5.681670] [drm] Max dedicated hypervisor surface memory is 507904 KiB  
[ 5.681671] [drm] Maximum display memory size is 16384 KiB  
vsfeoktistov@vsfeoktistov ~]$
```

Рис. 7.6: Объем оперативной памяти

Для обнаружения типа гипервизора использовал команду `dmesg | grep -i "Hypervisor detected"`. У меня это KVM (Kernel-based Virtual Machine) (рис. 7.7).



```
vsfeoktistov@vsfeoktistov:~  
File Edit View Search Terminal Help  
[ 0.000000] PM: Registered nosave memory: [mem 0xdfff0000-0xdfffffff]  
[ 0.000000] PM: Registered nosave memory: [mem 0xe0000000-0xfefbffff]  
[ 0.000000] PM: Registered nosave memory: [mem 0xfec00000-0xfec0ffff]  
[ 0.000000] PM: Registered nosave memory: [mem 0xfec01000-0xfedffffff]  
[ 0.000000] PM: Registered nosave memory: [mem 0xfec00000-0xfec0ffff]  
[ 0.000000] PM: Registered nosave memory: [mem 0xfec01000-0xfefbffff]  
[ 0.000000] PM: Registered nosave memory: [mem 0xfec00000-0xfefbffff]  
[ 0.000000] Memory: 3684168K/6291000K available (12293K kernel code, 5865K rwd, 829  
2K rodata, 2520K init, 14348K bss, 267924K reserved, 0K cma-reserved)  
[ 0.018164] Freeing SMP alternatives memory: 36K  
[ 0.277119] x86/mm: Memory block size: 128MB  
[ 2.505639] Freeing initrd memory: 50296K  
[ 2.953248] Non-volatile memory driver v1.3  
[ 3.700127] Freeing unused decrypted memory: 2036K  
[ 3.700929] Freeing unused kernel image (initmem) memory: 2520K  
[ 3.714705] Freeing unused kernel image (text/rodata gap) memory: 2012K  
[ 3.729689] Freeing unused kernel image (rodata/data gap) memory: 1948K  
[ 5.678775] [TTM] Zone kernel: Available graphics memory: 3040962 KiB  
[ 5.678790] [TTM] Zone dma32: Available graphics memory: 2097152 KiB  
[ 5.681670] [drm] Max dedicated hypervisor surface memory is 507904 KiB  
[ 5.681671] [drm] Maximum display memory size is 16384 KiB  
vsfeoktistov@vsfeoktistov ~]$ dmesg | grep -i "Hypervisor detected"  
[ 0.000000] Hypervisor detected: KVM  
vsfeoktistov@vsfeoktistov ~]$
```

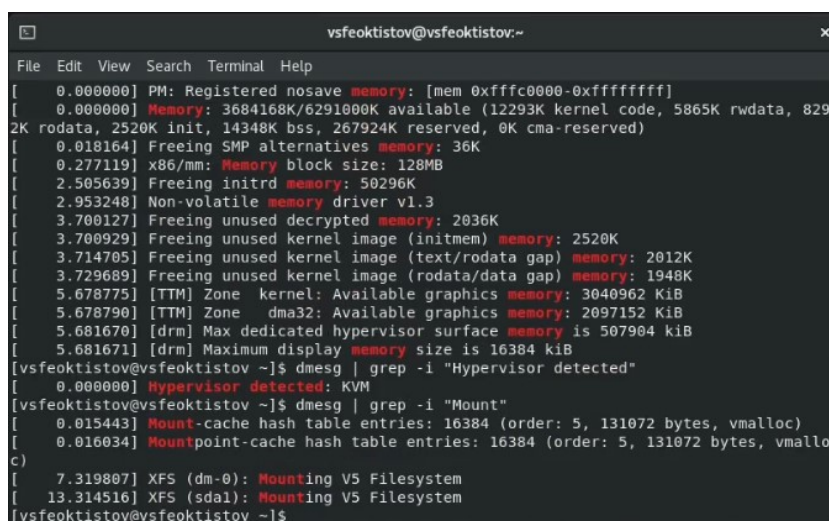
Рис. 7.7: Тип гипервизора

Гипервизор - программа или аппаратная схема, обеспечивающая или позволяющая одновременное, параллельное выполнение нескольких операционных

систем на одном и том же хост-компьютере. Гипервизор также обеспечивает изоляцию операционных систем друг от друга, защиту и безопасность, разделение ресурсов между различными запущенными ОС и управление ресурсами.

KVM – гипервизор, созданный в октябре 2006 года, был почти сразу интегрирован с основной веткой ядра Linux версии 2.6.20., выпущенной в начале 2007 года. Позже KVM был адаптирован как модуль ядра в FreeBSD. В KVM включены загружаемый модуль ядра `kvm.ko`, отвечающий за виртуализацию, процессорно-специфический загружаемый модуль для AMD или Intel `kvm-amd.ko` либо `kvm-intel.ko`, и компоненты пользовательского режима QEMU. KVM – полностью открытое ПО по лицензии GNU GPL и GNU LGPL.

Для получения информации о типе файловой системы корневого раздела и последовательности монтирования файловых систем я использовал команду `dmesg | grep -i "Mount"`. Из рисунка 7.8 видно, что у меня тип файловой системы XFS.



```
vsfeoktistov@vsfeoktistov:~  
File Edit View Search Terminal Help  
[ 0.000000] PM: Registered nosave memory: [mem 0xffff0000-0xffffffff]  
[ 0.000000] Memory: 3684168K/6291000K available (12293K kernel code, 5865K rwddata, 829  
2K rodata, 2520K init, 14348K bss, 267924K reserved, 0K cma-reserved)  
[ 0.018164] Freeing SMP alternatives memory: 36K  
[ 0.277119] x86/mm: Memory block size: 128MB  
[ 2.505639] Freeing initrd memory: 50296K  
[ 2.953248] Non-volatile memory driver v1.3  
[ 3.700127] Freeing unused decrypted memory: 2036K  
[ 3.700929] Freeing unused kernel image (initmem) memory: 2520K  
[ 3.714705] Freeing unused kernel image (text/rodata gap) memory: 2012K  
[ 3.729689] Freeing unused kernel image (rodata/data gap) memory: 1948K  
[ 5.678775] [TTM] Zone kernel: Available graphics memory: 3040962 KiB  
[ 5.678790] [TTM] Zone dma32: Available graphics memory: 2097152 KiB  
[ 5.681670] [drm] Max dedicated hypervisor surface memory is 507904 kiB  
[ 5.681671] [drm] Maximum display memory size is 16384 kiB  
[vsfeoktistov@vsfeoktistov ~]$ dmesg | grep -i "Hypervisor detected"  
[ 0.000000] Hypervisor detected: KVM  
[vsfeoktistov@vsfeoktistov ~]$ dmesg | grep -i "Mount"  
[ 0.015443] Mount-cache hash table entries: 16384 (order: 5, 131072 bytes, vmalloc)  
[ 0.016034] Mountpoint-cache hash table entries: 16384 (order: 5, 131072 bytes, vmallo  
c)  
[ 7.319807] XFS (dm-0): Mounting V5 Filesystem  
[ 13.314516] XFS (sda1): Mounting V5 Filesystem  
[vsfeoktistov@vsfeoktistov ~]$
```

Рис. 7.8: Тип файловой системы корневого раздела

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.