Лабораторная работа №1

Установка и конфигурация операционной системы на виртуальную машину

Карпова Есения Алексеевна

Содержание

# 1 Цель работы

Целью данной работы является приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

# 2 Задания

1. Установка и настройка виртуальной машины
2. Домашнее задание
3. Контрольные вопросы

# 3 Теоретическое введение

Виртуальная машина — это программная или аппаратная система, эмулирующая аппаратное обеспечение и позволяющая запускать операционные системы и приложения в изолированных средах. Один из популярных инструментов виртуализации — Oracle VM VirtualBox, поддерживающий различные операционные системы, включая Windows, Linux, macOS и другие. Виртуальные машины дают возможность работать с несколькими операционными системами на одном физическом устройстве, эффективно распределяя ресурсы.

Операционная система управляет ресурсами компьютера и позволяет запускать прикладные программы. На виртуальной машине она функционирует так же, как и на физическом компьютере, но с учетом особенностей виртуализации. Для Linux существует множество дистрибутивов, таких как Ubuntu, Fedora, Debian и Rocky Linux. Последний представляет собой дистрибутив, ориентированный на корпоративное использование и поддерживающий долгосрочную стабильность, что делает его популярным выбором для серверных решений и в облачных средах. Rocky Linux является продолжением CentOS, обеспечивая совместимость с Red Hat Enterprise Linux (RHEL) и предлагая открытый и свободно доступный вариант для бизнеса и разработки.

# 4 Выполнение лабораторной работы

## 4.1 Установка и настройка виртуальной машины

Я выполняю лабораторную работу на своей технике, поэтому располагаю файл с виртуальной машиной в удобной для меня директории, предварительно скачав образ операционной системы Rocky с официального сайта. Перейдя в программу, я перехожу в вкладу “Машина” и выбираю “Создать”. Указываю имя виртуальной машины, тип операционной системы — Linux, RedHat (64-bit) (рис. 1).

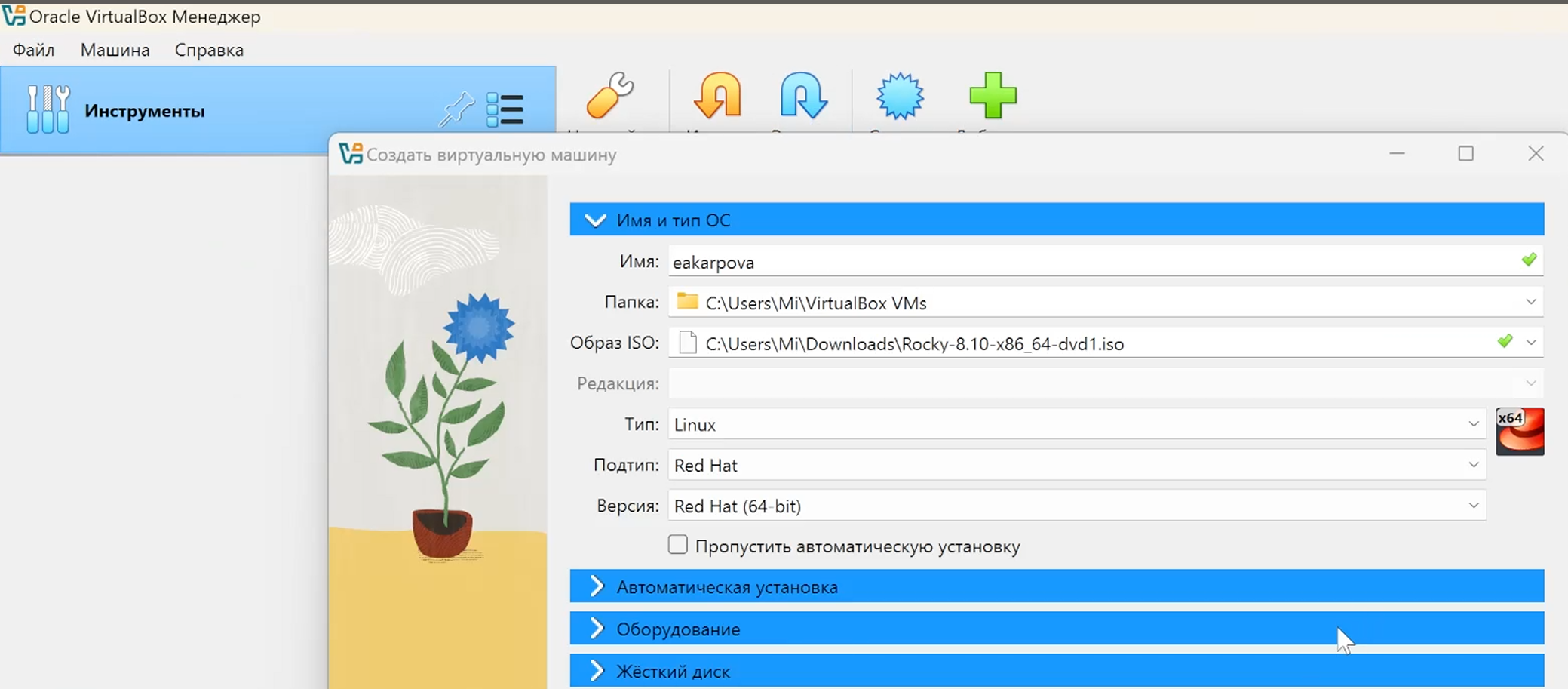


Рис. 1: Создание виртуальной машины

Указываю размер основной памяти виртуальной машины — 2048 МБ, задаю конфигурацию жёсткого диска — загрузочный (VDI - динамический виртуальный диск) и размер диска — 40 ГБ. (рис. 2).

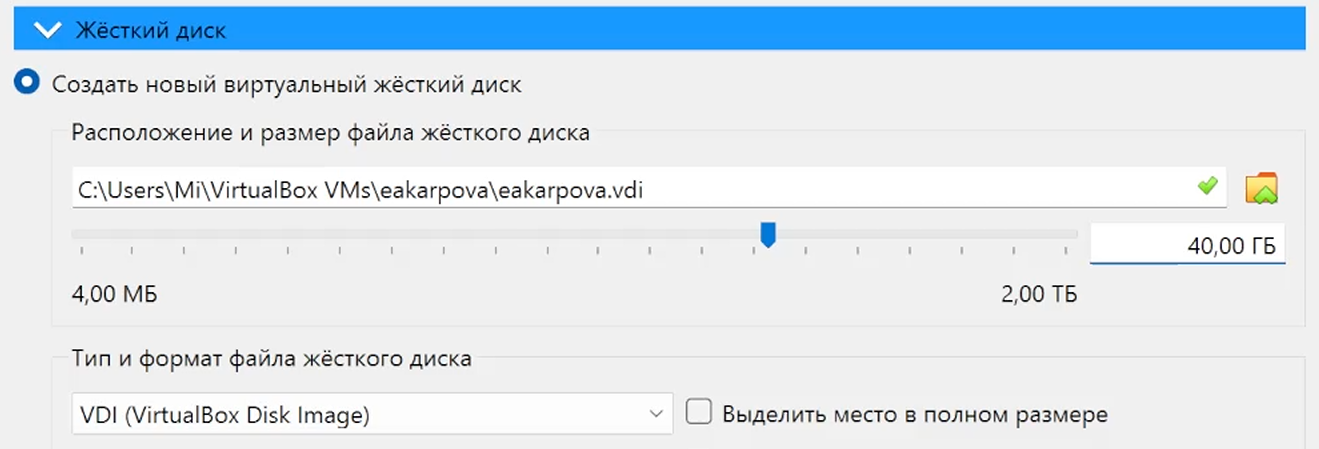


Рис. 2: Задание настроек ВМ

После того, как я задала основные настройки ВМ, у меня открылось окно, где справа расположена созданная ВМ, а также перечислены ее характеристики (рис. 3).

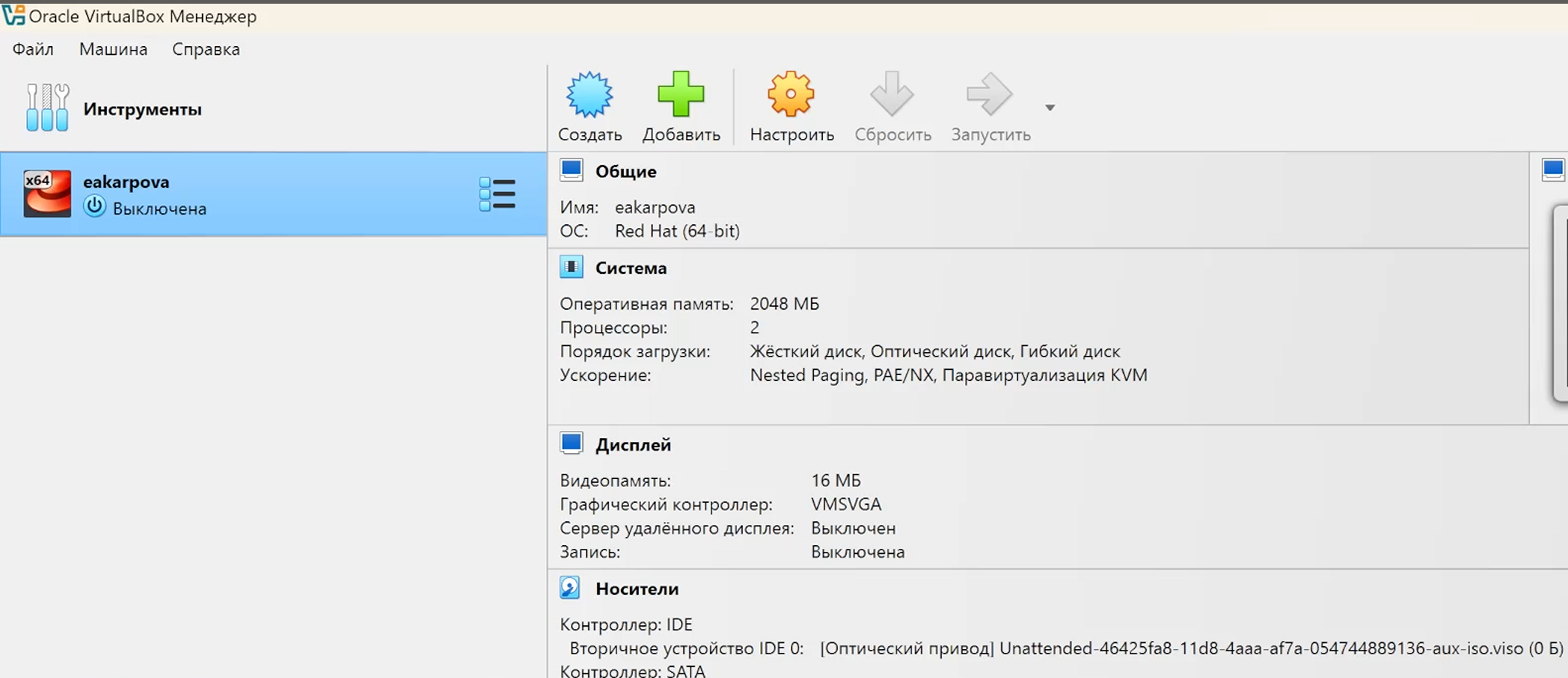


Рис. 3: Окно с созданной ВМ

Добавляю новый привод оптических дисков и выбераю образ операционной системы Rocky (рис. 4).

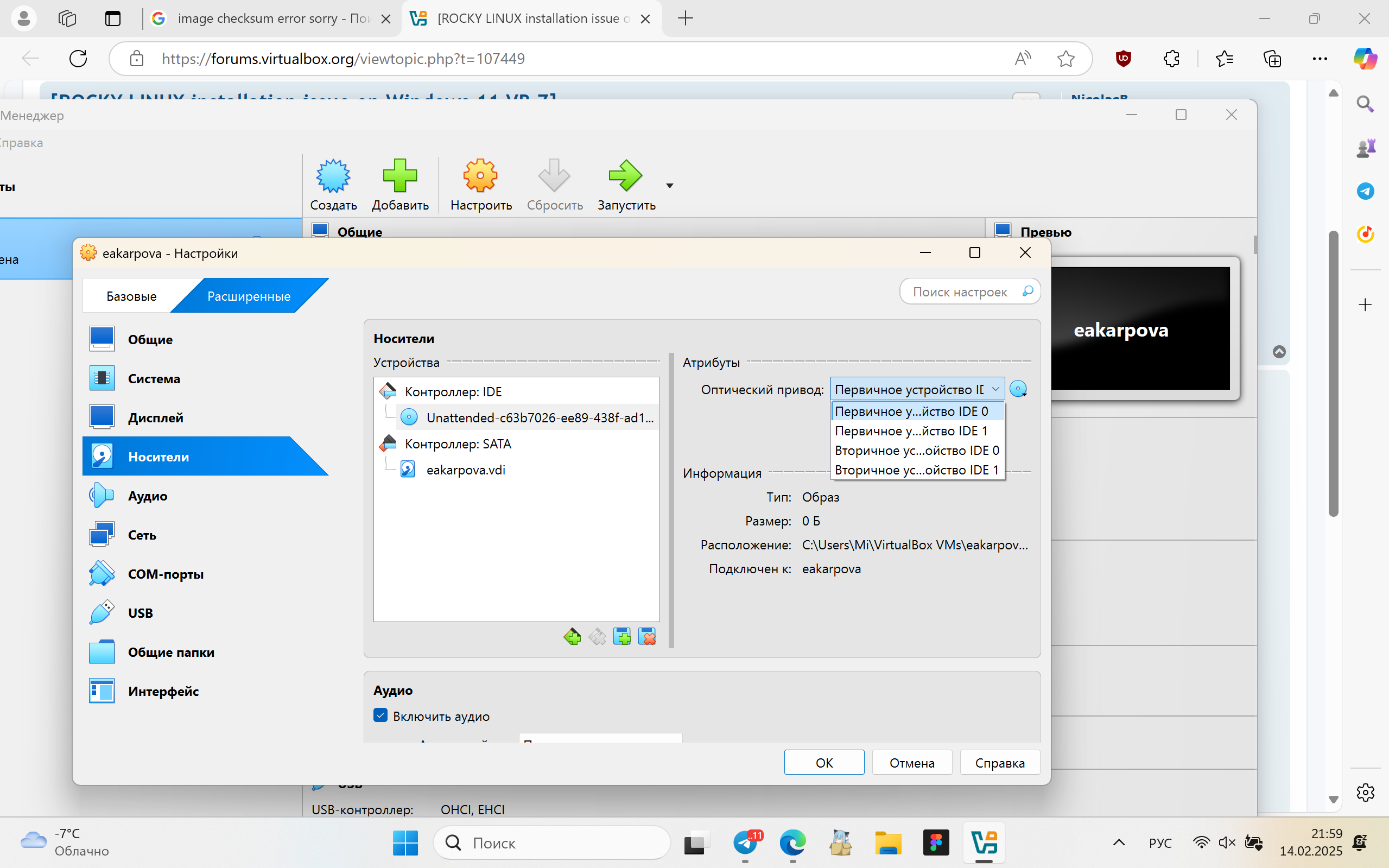


Рис. 4: Добавление образа ОС

Запускаю виртуальную машину, выбераю English в качестве языка интерфейса и перехожу к настройкам установки опера- ционной системы (рис. 5).

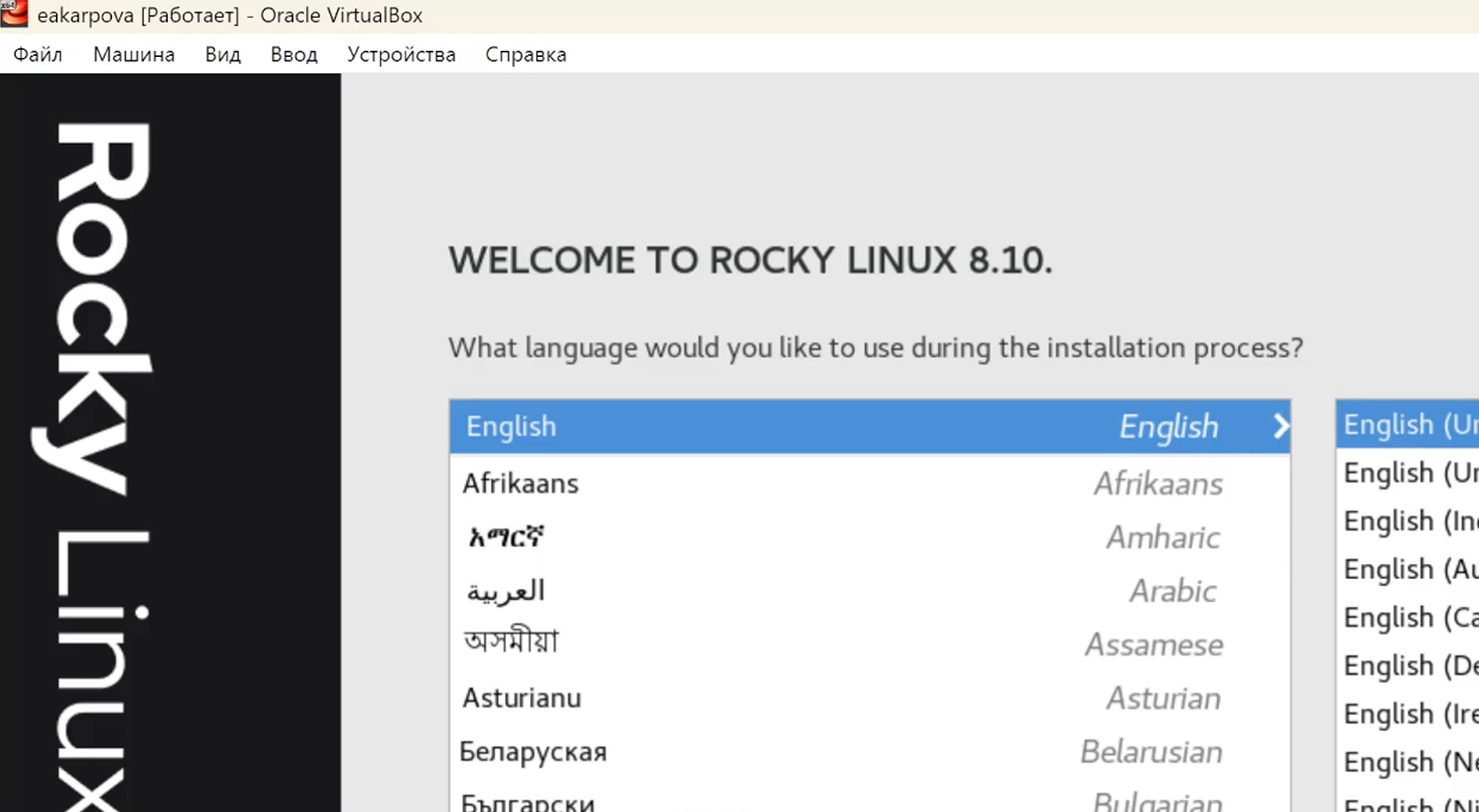


Рис. 5: Выбор языка интерфейса

В открывшемся окне приступаю к настройкам операционной системы. Включаю сетевое соединение и в качестве имени узла указываю eakarpova.localdomain (рис. 6).

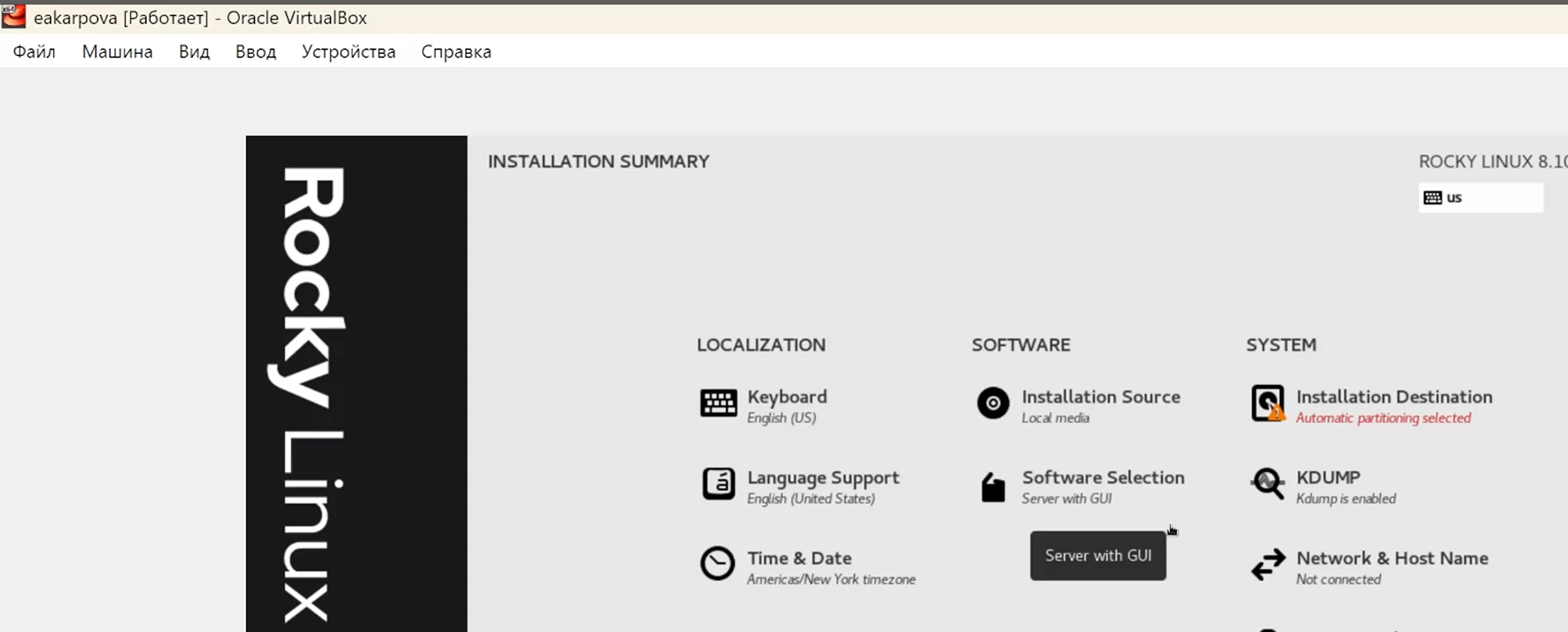


Рис. 6: Окно настроек ОС

Место установки ОС оставляю без изменения. Отключаю KDUMP. (рис. 7).

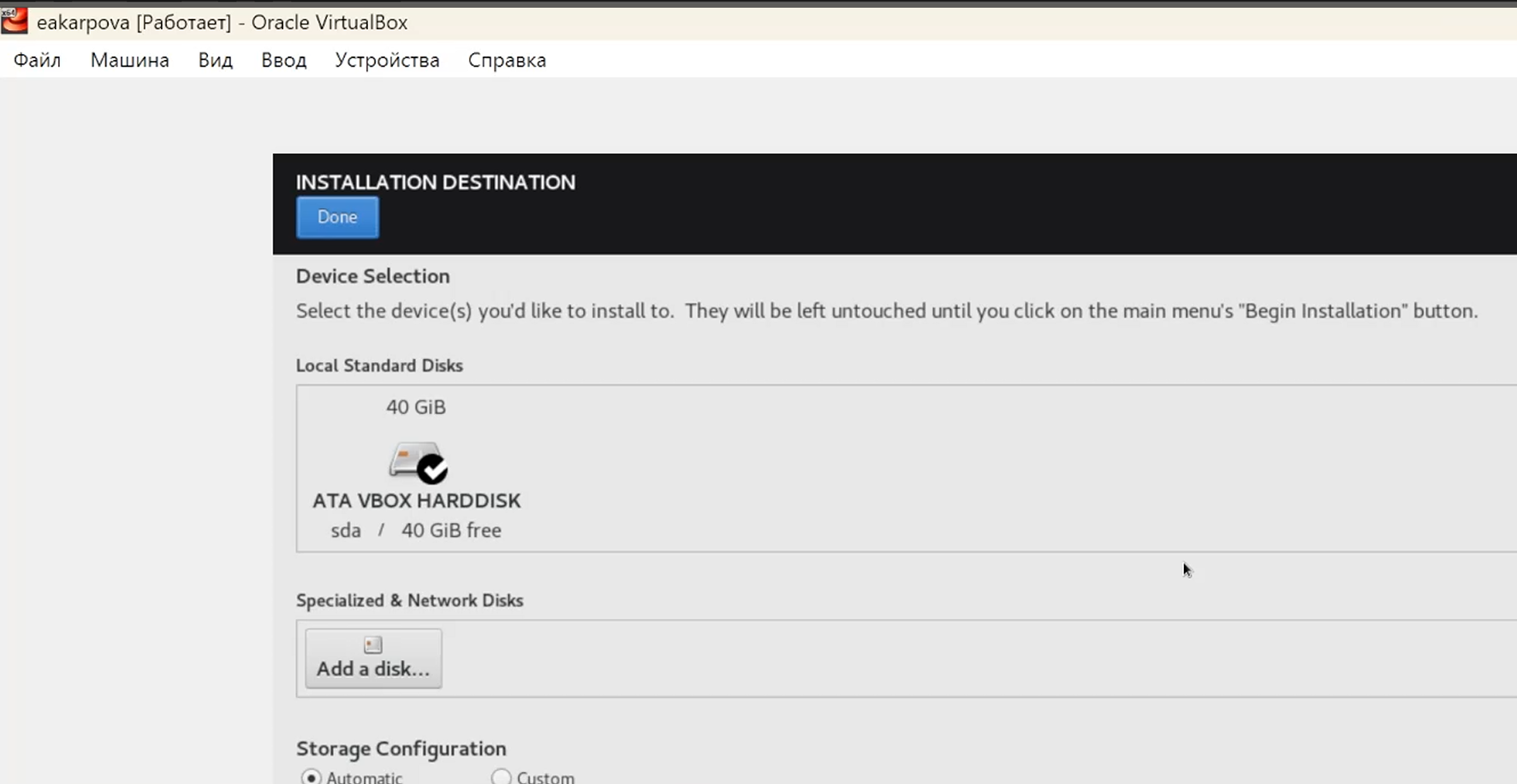


Рис. 7: Место установки ОС

В разделе выбора программ укажите в качестве базового окружения Server with GUI , а в качестве дополнения — Development Tools (рис. 8).

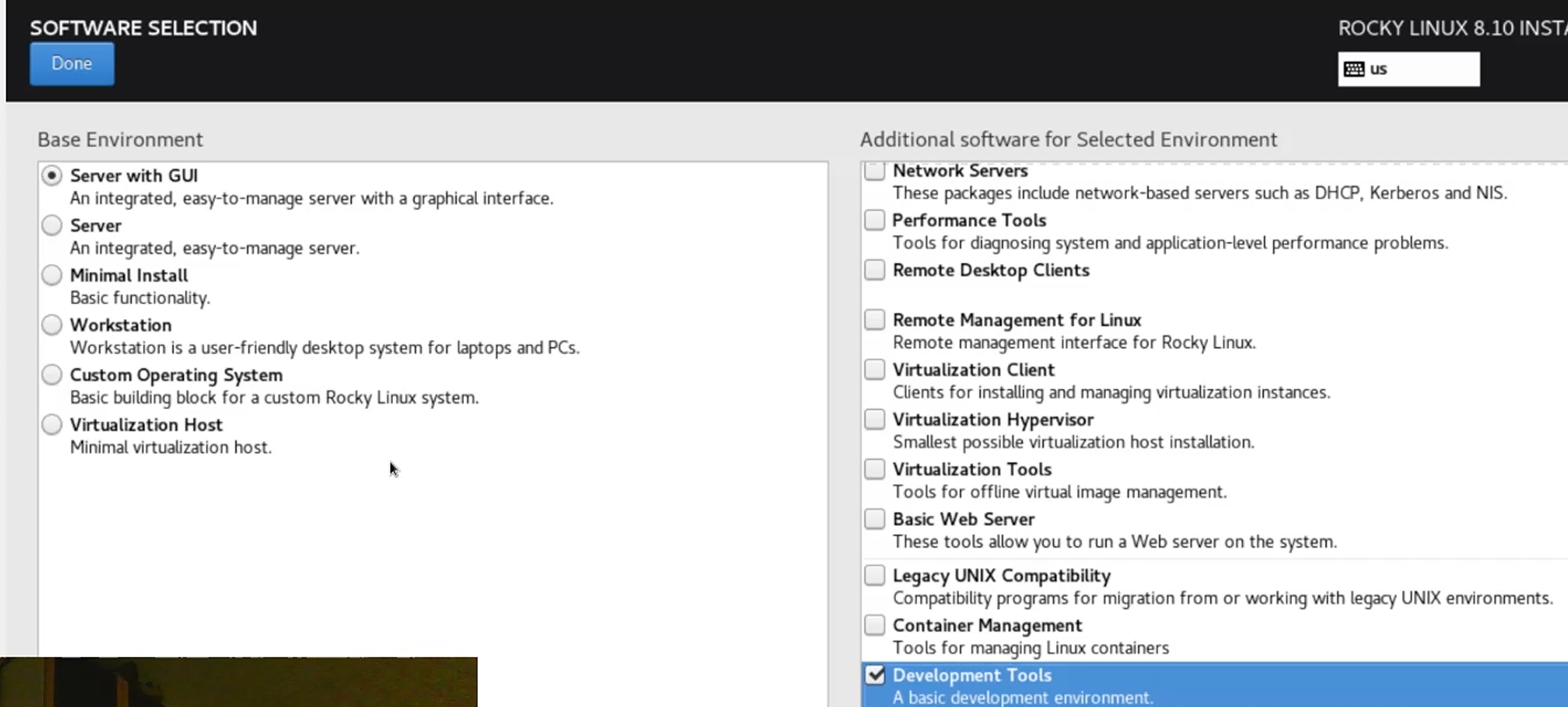


Рис. 8: Выбор программ

Корректирую часовой пояс, раскладку клавиатуры (рис. 9).

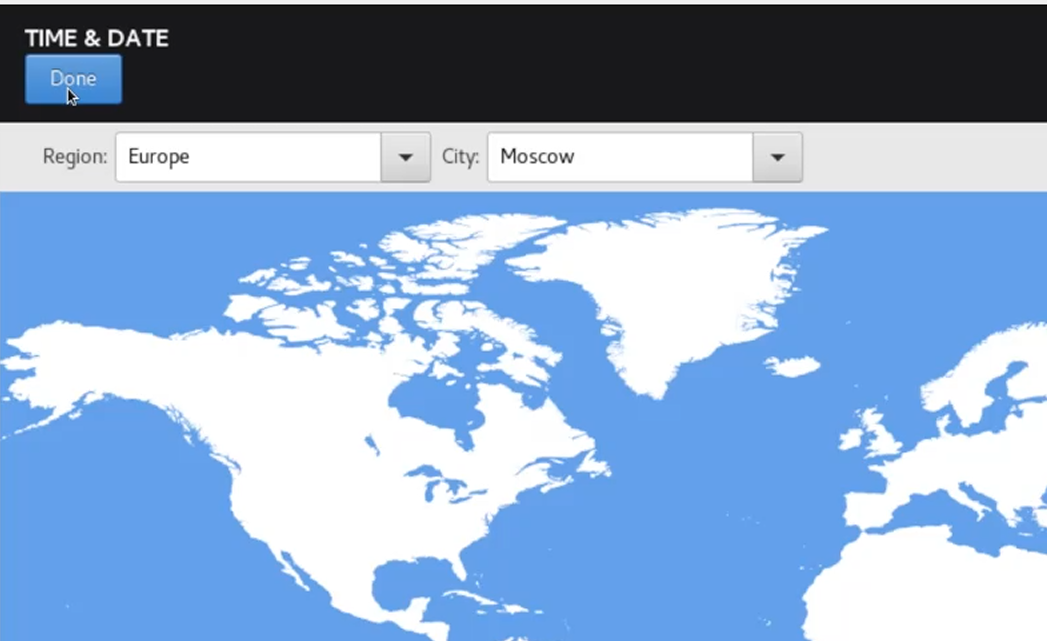


Рис. 9: Настройка часового пояса

Установливаю пароль для root и пользователя с правами администратора. После завершения установки операционной системы корректно перезапускаю виртуальную машину и при запросе примаю условия лицензии. Затем вхожу в ОС под заданной учётной записью. В меню “Устройства виртуальной машины” подключаю образ диска дополнений гостевой ОС (рис. 10).

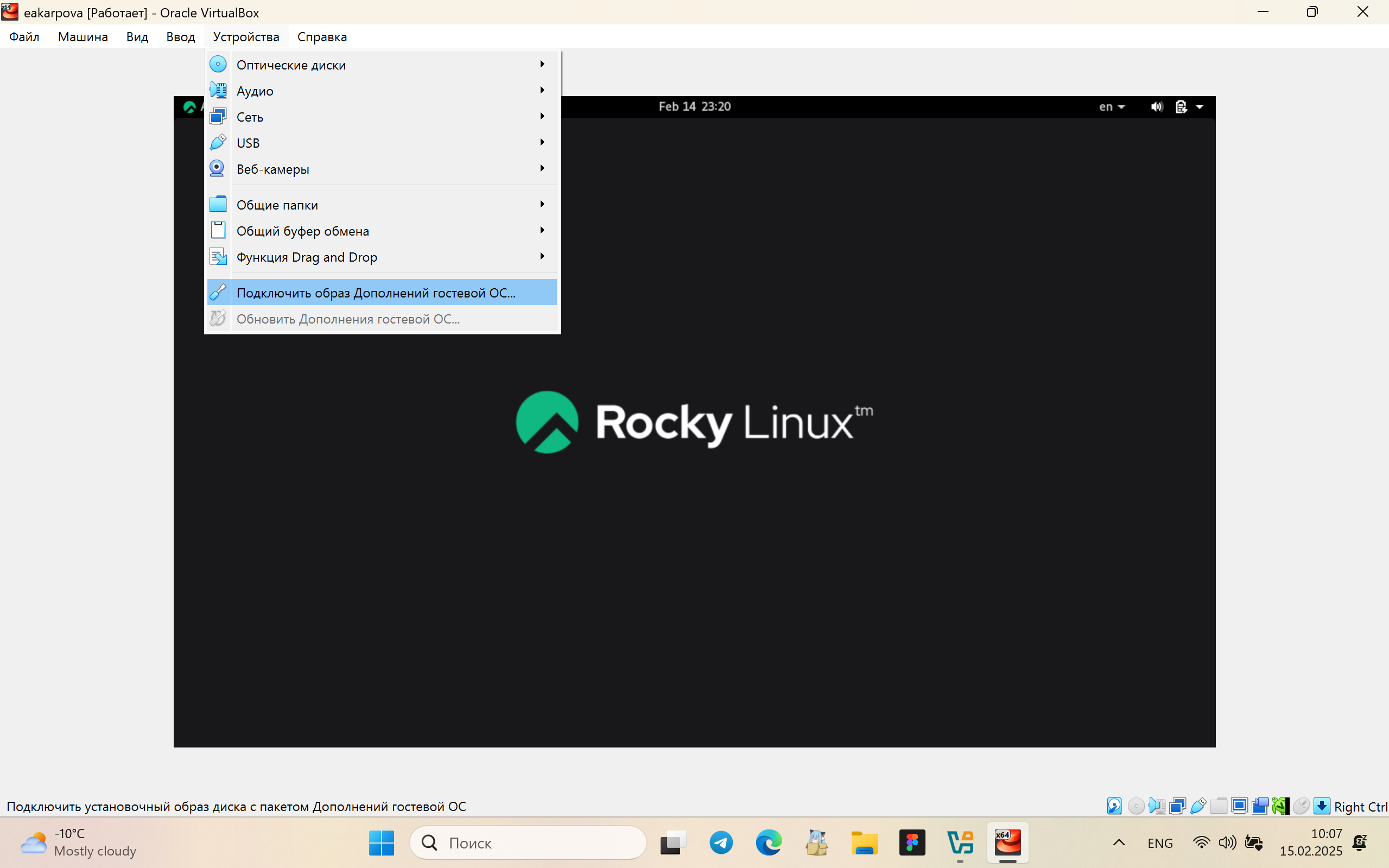


Рис. 10: Образ диска дополнений

После загрузки дополнений нажмаю Enter и корректно перезагружаю виртуальную машину. (рис. 11).

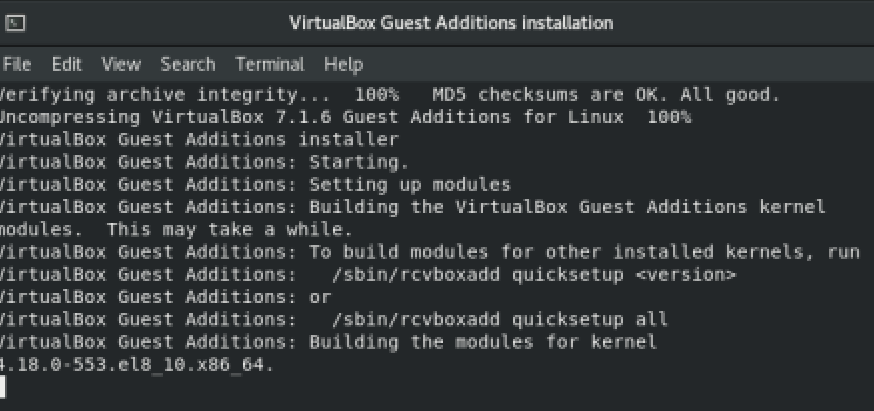


Рис. 11: Загрузка дополнений

## 4.2 Установка и настройка виртуальной машины

С помощью конструкции с grep:

dmesg | grep -i “то, что ищем”

Получаю следующую информацию:

1. Версия ядра Linux (Linux version).

Команда: dsmeg (рис. 12).

Информация выводится в первой строке, поэтому я не использовала дополнительные утилиты.

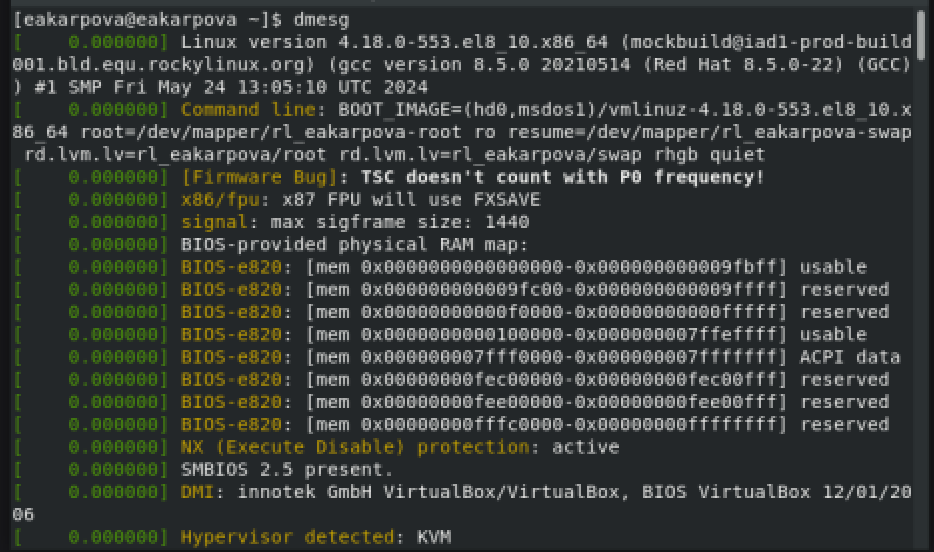


Рис. 12: Версия ядра Linux

1. Частота процессора (Detected Mhz processor)

Команда: dmesg | grep -i “mhz processor” (рис. 13).

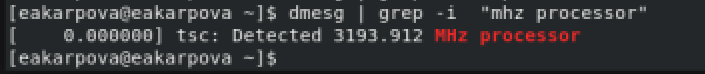


Рис. 13: Частота процессора

1. Модель процессора (CPU0)

Команда: dmesg | grep -i “cpu” (рис. 14).

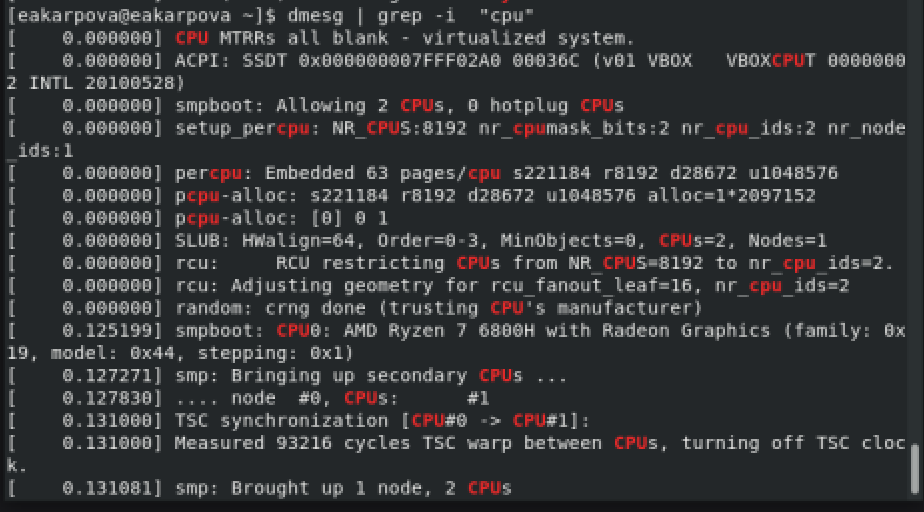


Рис. 14: Модель процессора

1. Объем доступной оперативной памяти (Memory available)

Команда: dmesg | grep -i “memory” (рис. 15).

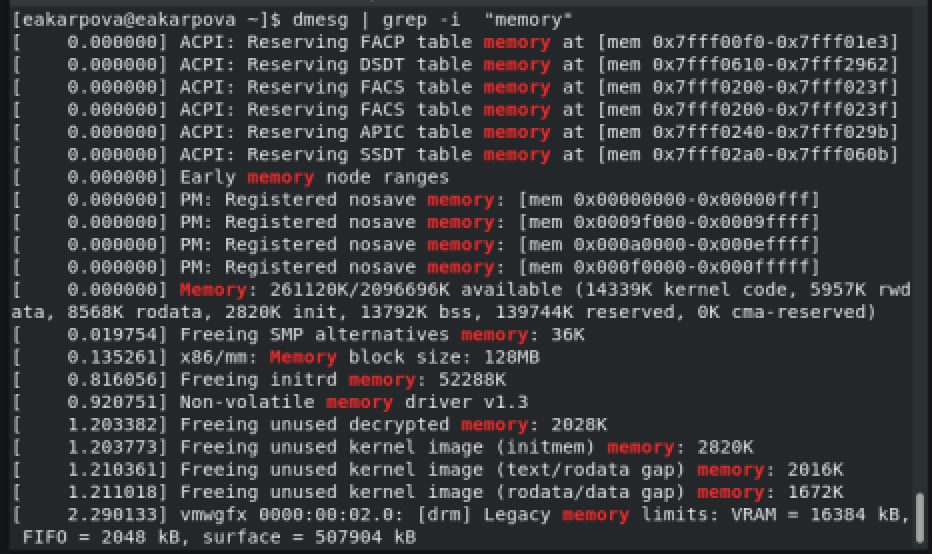


Рис. 15: Объем ОП

1. Тип обнаруженного гипервизора (Hypervisor detected). Команда: dmesg | grep -i “hypervisor” (рис. 16).
2. Тип файловой системы корневого раздела Команда: dmesg | grep -i “filesystem” (рис. 16).

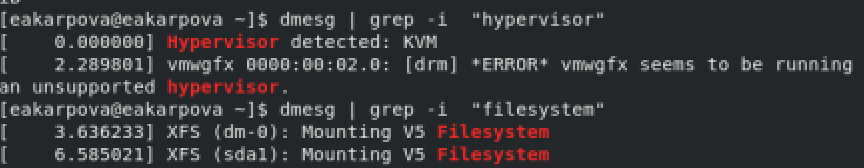


Рис. 16: Тип файловой системы и гипервизора

## 4.3 Ответы на контрольные вопросы

1. Учётная запись пользователя содержит:
2. Имя пользователя (логин) – идентификатор для входа в систему
3. Идентификатор пользователя (UID) – уникальный числовой идентификатор пользователя
4. Идентификатор группы (GID) – идентификатор основной группы пользователя
5. Домашний каталог – путь к каталогу, предназначенному для хранения личных файлов пользователя
6. Используемая командная оболочка (shell) – интерпретатор команд, используемый пользователем
7. Зашифрованный пароль – для аутентификации пользователя
8. Команды терминала и примеры их использования:

* Получение справки по команде:
* \* `man <команда>` – отображает руководство пользователя для указанной команды (пример: `man ls`).  
   \* `<команда> --help` - отображает краткую информацию об использовании команды.
* Перемещение по файловой системе:
* \* `cd <каталог>` – переход в указанный каталог (пример: `cd /home/user/documents`).  
   \* `cd ..` – переход в родительский каталог.  
   \* `cd` – переход в домашний каталог пользователя.
* Просмотр содержимого каталога:
* \* `ls` – вывод списка файлов и подкаталогов в текущем каталоге.  
   \* `ls -l` – вывод подробной информации о файлах и подкаталогах (права, размер, дата изменения).  
   \* `ls -a` - отображает все файлы, включая скрытые.
* Определение объёма каталога:
* \* `du -sh <каталог>` – вывод объёма указанного каталога в человеко-читаемом формате (пример: `du -sh /var/log`).
* Создание/удаление каталогов/файлов:
* \* `mkdir <каталог>` – создание нового каталога (пример: `mkdir new\_directory`).  
   \* `rmdir <каталог>` – удаление пустого каталога (пример: `rmdir empty\_directory`).  
   \* `touch <файл>` – создание нового файла (пример: `touch new\_file.txt`).  
   \* `rm <файл>` – удаление файла (пример: `rm my\_file.txt`).  
   \* `rm -r <каталог>` – удаление каталога и всего его содержимого (пример: `rm -r old\_directory`).
* Задание прав доступа к файлам/каталогам:
* \* `chmod <права> <файл>` – изменение прав доступа к файлу или каталогу (пример: `chmod 755 script.sh`).  
   \* `chmod +x <файл>` - сделать файл исполняемым.
* Просмотр истории команд:
* \* `history` – вывод списка ранее выполненных команд.

1. Файловая система – это метод организации хранения данных на диске, определяющий структуру и атрибуты файлов и каталогов.

* Примеры файловых систем:
  + ext4: Наиболее распространённая файловая система в Linux, характеризуется надёжностью и производительностью.
  + NTFS: Стандартная файловая система для операционных систем Windows, поддерживает расширенные атрибуты и права доступа.
  + Btrfs: Современная файловая система с поддержкой снапшотов, сжатия данных и других продвинутых функций.

1. Список смонтированных файловых систем можно получить с помощью команды mount. Эта команда отображает информацию о каждой примонтированной файловой системе, включая её тип и точку монтирования. Также, информация о монтировании файловых систем хранится в файле /etc/fstab.
2. Для удаления зависшего процесса необходимо сначала определить его PID (идентификатор процесса). Это можно сделать с помощью команд ps aux | grep <имя\_процесса> или top. Затем можно использовать команду kill:
   * kill <PID> – отправка сигнала SIGTERM для запроса корректного завершения процесса.
   * kill -9 <PID> – отправка сигнала SIGKILL для принудительного завершения процесса (следует использовать только в крайних случаях).

# 5 Выводы

В ходе лабораторной работы я приобрела практические навыки установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов