

# Лабораторная работа №1

## Установка и конфигурация операционной системы на виртуальную машину

Карпова Есения Алексеевна

### Содержание

1	Цель работы .....	1
2	Задания .....	1
3	Теоретическое введение .....	1
4	Выполнение лабораторной работы .....	2
4.1	Установка и настройка виртуальной машины.....	2
4.2	Установка и настройка виртуальной машины.....	7
4.3	Ответы на контрольные вопросы .....	10
5	Выводы.....	12

### 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 вкладку “Машина” и выбираю “Создать”. Указываю имя виртуальной машины, тип операционной системы — Linux, RedHat (64-bit) (рис. 1).

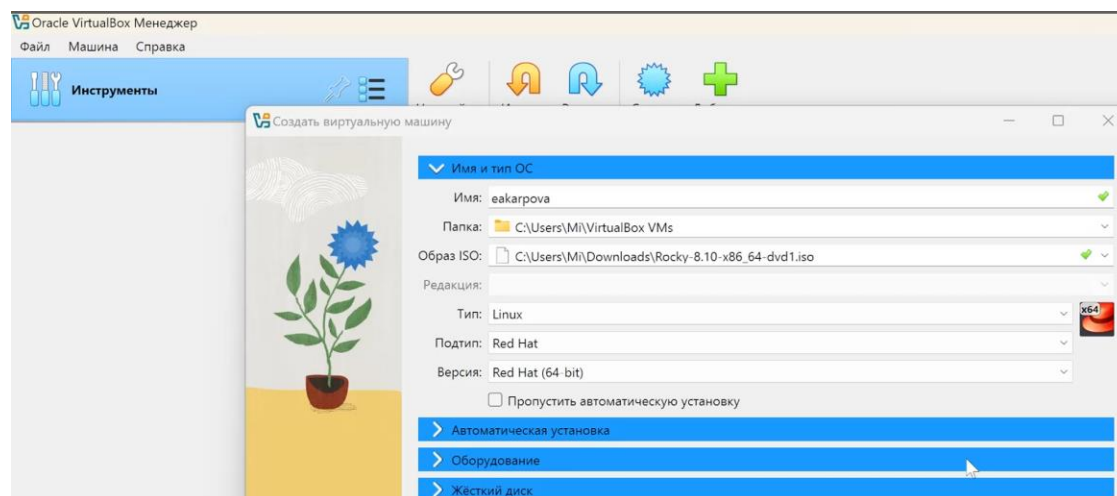


Рис. 1: Создание виртуальной машины

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

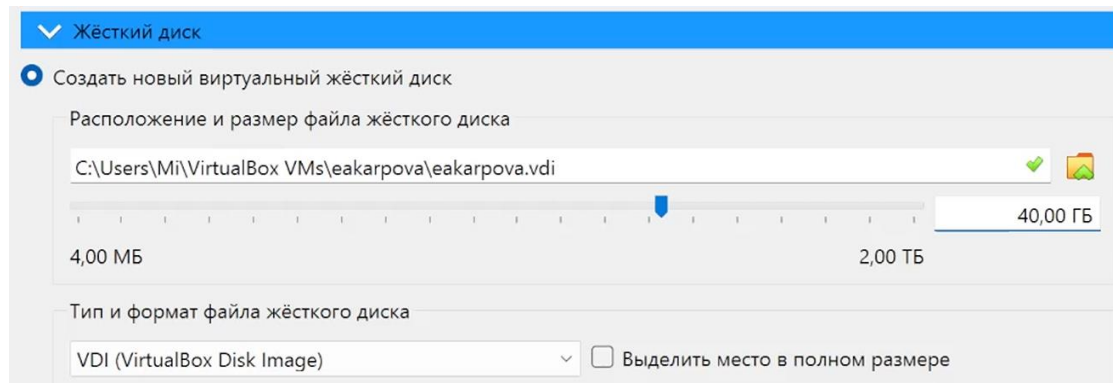


Рис. 2: Задание настроек VM

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

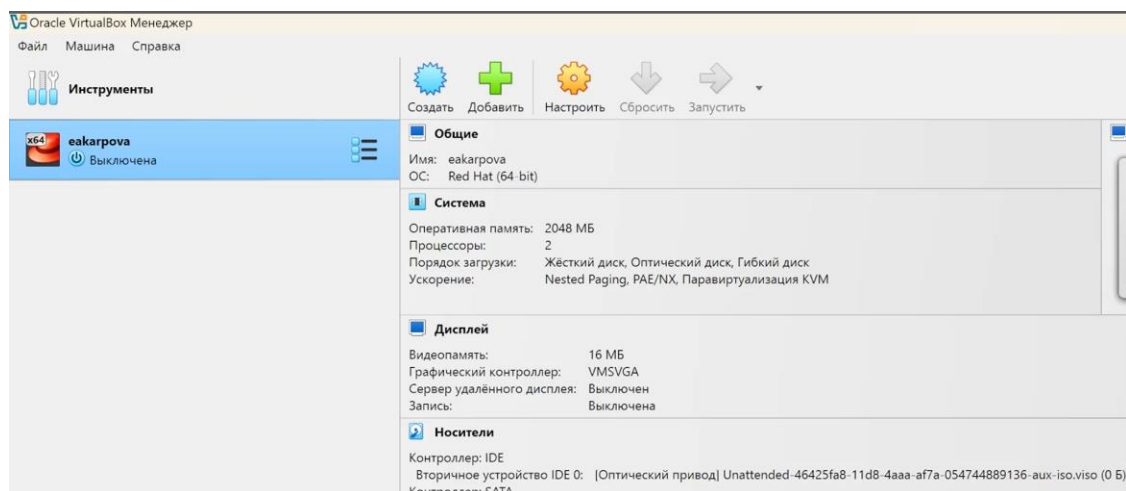


Рис. 3: Окно с созданной VM

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

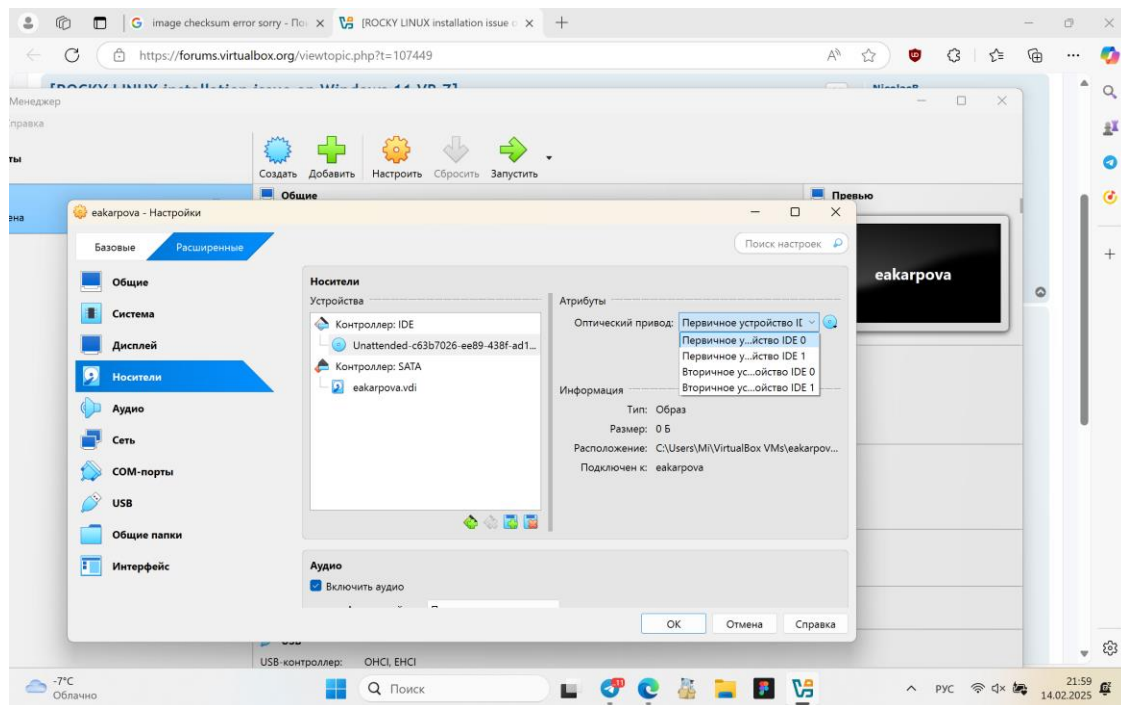


Рис. 4: Добавление образа ОС

Запускаю виртуальную машину, выбираю English в качестве языка интерфейса и перехожу к настройкам установки операционной системы (рис. 5).

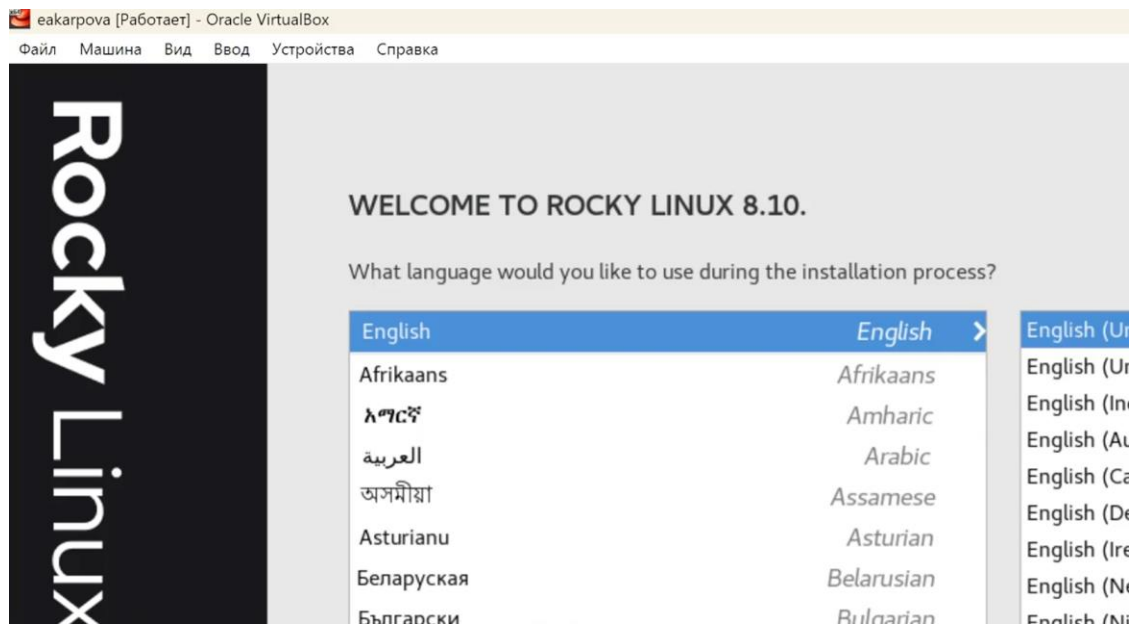


Рис. 5: Выбор языка интерфейса

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

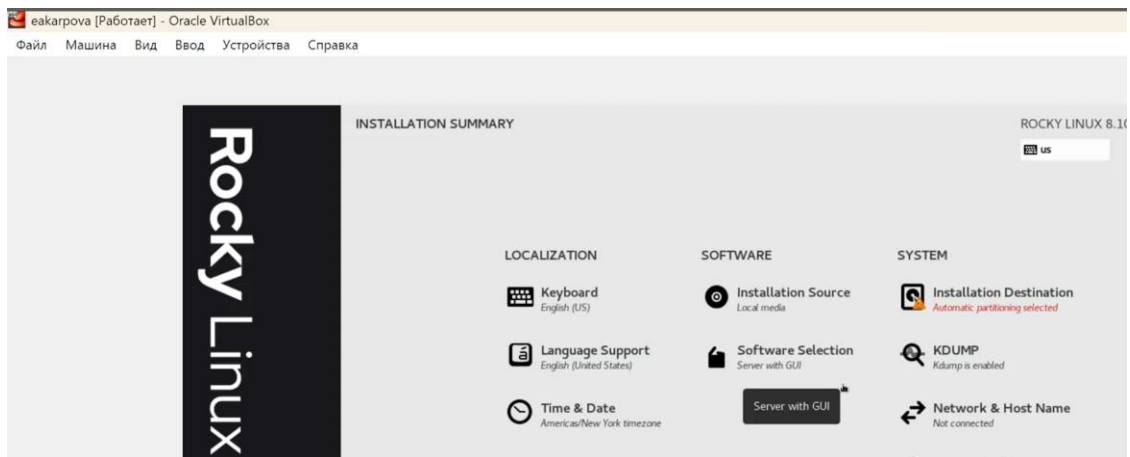


Рис. 6: Окно настроек ОС

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

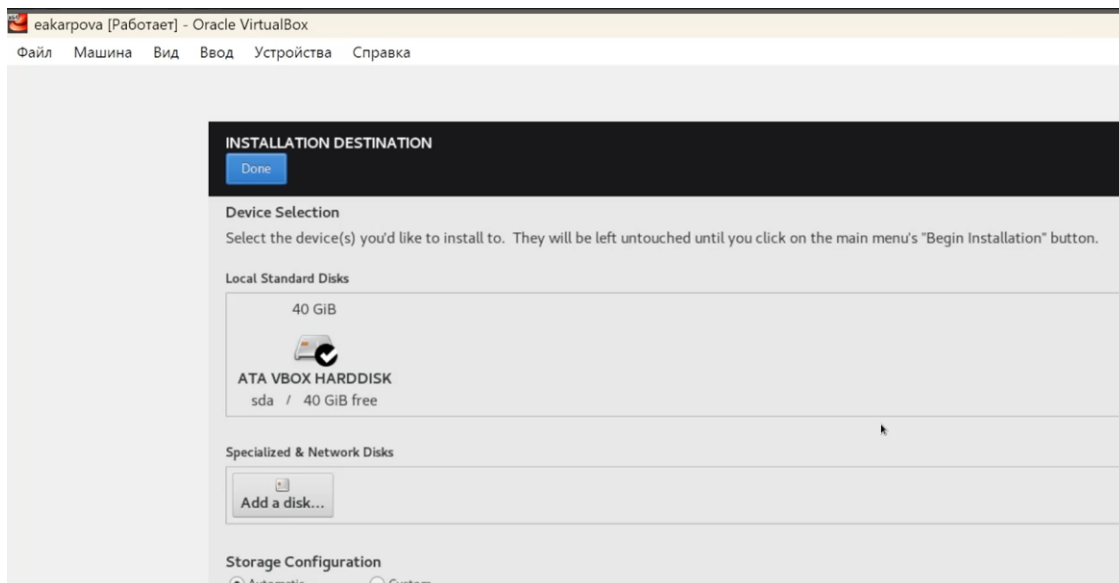


Рис. 7: Место установки ОС

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

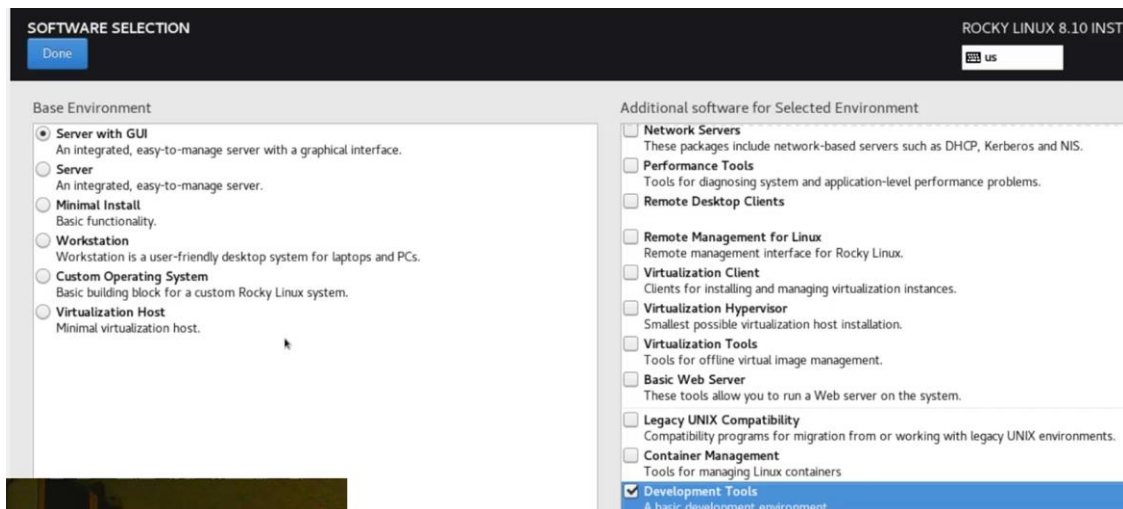


Рис. 8: Выбор программ

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

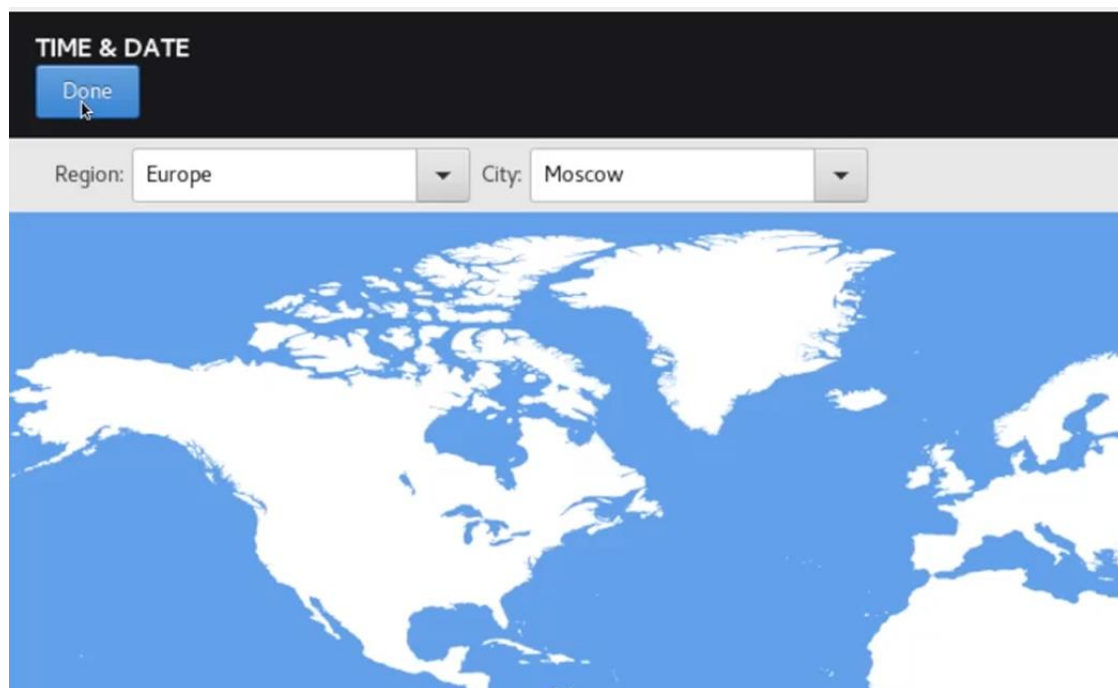


Рис. 9: Настройка часового пояса

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

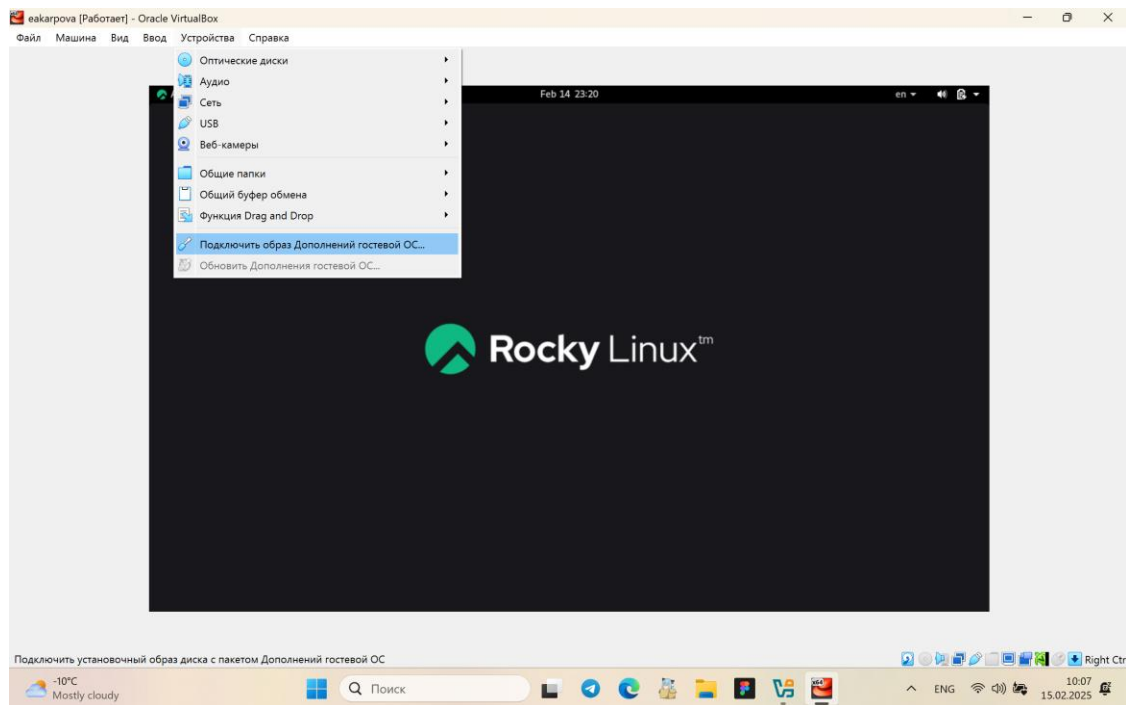


Рис. 10: Образ диска дополнений

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

```

VirtualBox Guest Additions installation
File Edit View Search Terminal Help
/erifying archive integrity... 100% MD5 checksums are OK. All good.
Uncompressing VirtualBox 7.1.6 Guest Additions for Linux 100%
VirtualBox Guest Additions installer
VirtualBox Guest Additions: Starting.
VirtualBox Guest Additions: Setting up modules
VirtualBox Guest Additions: Building the VirtualBox Guest Additions kernel
modules. This may take a while.
VirtualBox Guest Additions: To build modules for other installed kernels, run
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup <version>
VirtualBox Guest Additions: or
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup all
VirtualBox Guest Additions: Building the modules for kernel
4.18.0-553.el8_10.x86_64.

```

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: dmesg (рис. 12).

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

```
[eakarpova@eakarpova ~]$ dmesg
[ 0.000000] Linux version 4.18.0-553.el8_10.x86_64 (mockbuild@iad1-prod-build
001.bld.equ.rockylinux.org) (gcc version 8.5.0 20210514 (Red Hat 8.5.0-22) (GCC)
) #1 SMP Fri May 24 13:05:10 UTC 2024
[ 0.000000] Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-4.18.0-553.el8_10.x
86_64 root=/dev/mapper/rl_eakarpova-root ro resume=/dev/mapper/rl_eakarpova-swap
rd.lvm.lv=rl_eakarpova/root rd.lvm.lv=rl_eakarpova/swap rhgb quiet
[ 0.000000] [Firmware Bug]: TSC doesn't count with P0 frequency!
[ 0.000000] x86/fpu: x87 FPU will use FXSAVE
[ 0.000000] signal: max sigframe size: 1440
[ 0.000000] BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x0000000000009fbff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000009fc00-0x0000000000009ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x000000000000f0000-0x000000000000fffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000001000000-0x00000000007ffefffff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000007fff00000-0x00000000007ffffffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x00000000fec000000-0x00000000fec00ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fee000000-0x00000000fee00ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fffc00000-0x00000000ffffffffff] reserved
[ 0.000000] NX (Execute Disable) protection: active
[ 0.000000] SMBIOS 2.5 present.
[ 0.000000] DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/20
06
[ 0.000000] Hypervisor detected: KVM
```

Рис. 12: Версия ядра Linux

2. Частота процессора (Detected Mhz processor)

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

```
[eakarpova@eakarpova ~]$ dmesg | grep -i "mhz processor"
[ 0.000000] tsc: Detected 3193.912 MHz processor
[eakarpova@eakarpova ~]$
```

Рис. 13: Частота процессора

3. Модель процессора (CPU0)

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



```

[eakarpova@eakarpova ~]$ dmesg | grep -i "cpu"
[ 0.000000] CPU MTRRs all blank - virtualized system.
[ 0.000000] ACPI: SSDT 0x000000007FFF02A0 00036C (v01 VBOX VBOXCPU 00000000
2 INTL 20100528)
[ 0.000000] smpboot: Allowing 2 CPUs, 0 hotplug CPUs
[ 0.000000] setup_percpu: NR_CPUS:8192 nr_cpumask_bits:2 nr_cpu_ids:2 nr_node
_ids:1
[ 0.000000] percpu: Embedded 63 pages/cpu s221184 r8192 d28672 u1048576
[ 0.000000] pcpu-alloc: s221184 r8192 d28672 u1048576 alloc=1*2097152
[ 0.000000] pcpu-alloc: [0] 0 1
[ 0.000000] SLUB: HWalign=64, Order=0-3, MinObjects=0, CPUs=2, Nodes=1
[ 0.000000] rcu: RCU restricting CPUs from NR_CPUS=8192 to nr_cpu_ids=2.
[ 0.000000] rcu: Adjusting geometry for rcu_fanout_leaf=16, nr_cpu_ids=2
[ 0.000000] random: crng done (trusting CPU's manufacturer)
[ 0.125199] smpboot: CPU0: AMD Ryzen 7 6800H with Radeon Graphics (family: 0x
19, model: 0x44, stepping: 0x1)
[ 0.127271] smp: Bringing up secondary CPUs ...
[ 0.127830] .... node #0, CPUs: #1
[ 0.131000] TSC synchronization [CPU#0 -> CPU#1]:
[ 0.131000] Measured 93216 cycles TSC warp between CPUs, turning off TSC cloc
k.
[ 0.131081] smp: Brought up 1 node, 2 CPUs

```

Рис. 14: Модель процессора

#### 4. Объем доступной оперативной памяти (Memory available)

Команда: `dmesg | grep -i "memory"` (рис. 15).

```

[eakarpova@eakarpova ~]$ dmesg | grep -i "memory"
[ 0.000000] ACPI: Reserving FACP table memory at [mem 0x7fff00f0-0x7fff01e3]
[ 0.000000] ACPI: Reserving DSDT table memory at [mem 0x7fff0610-0x7fff2962]
[ 0.000000] ACPI: Reserving FACS table memory at [mem 0x7fff0200-0x7fff023f]
[ 0.000000] ACPI: Reserving FACS table memory at [mem 0x7fff0200-0x7fff023f]
[ 0.000000] ACPI: Reserving APIC table memory at [mem 0x7fff0240-0x7fff029b]
[ 0.000000] ACPI: Reserving SSDT table memory at [mem 0x7fff02a0-0x7fff060b]
[ 0.000000] Early memory node ranges
[ 0.000000] PM: Registered nosave memory: [mem 0x00000000-0x00000fff]
[ 0.000000] PM: Registered nosave memory: [mem 0x0009f000-0x0009ffff]
[ 0.000000] PM: Registered nosave memory: [mem 0x000a0000-0x000effff]
[ 0.000000] PM: Registered nosave memory: [mem 0x000f0000-0x000fffff]
[ 0.000000] Memory: 261120K/2096696K available (14339K kernel code, 5957K rwd
ata, 8568K rodata, 2820K init, 13792K bss, 139744K reserved, 0K cma-reserved)
[ 0.019754] Freeing SMP alternatives memory: 36K
[ 0.135261] x86/mm: Memory block size: 128MB
[ 0.816056] Freeing initrd memory: 52280K
[ 0.920751] Non-volatile memory driver v1.3
[ 1.203382] Freeing unused decrypted memory: 2028K
[ 1.203773] Freeing unused kernel image (initmem) memory: 2820K
[ 1.210361] Freeing unused kernel image (text/rodata gap) memory: 2016K
[ 1.211018] Freeing unused kernel image (rodata/data gap) memory: 1672K
[ 2.290133] vmwgfx 0000:00:02.0: [drm] Legacy memory limits: VRAM = 16384 kB,
FIFO = 2048 kB, surface = 507904 kB

```

Рис. 15: Объем ОП

#### 5. Тип обнаруженного гипервизора (Hypervisor detected). Команда: `dmesg | grep -i "hypervisor"` (рис. 16).

6. Тип файловой системы корневого раздела Команда: `dmesg | grep -i "filesystem"` (рис. 16).

```
[eakarpova@eakarpova ~]$ dmesg | grep -i "hypervisor"
[ 0.000000] Hypervisor detected: KVM
[ 2.289801] vmwgfx 0000:00:02.0: [drm] *ERROR* vmwgfx seems to be running
an unsupported hypervisor.
[eakarpova@eakarpova ~]$ dmesg | grep -i "filesystem"
[ 3.636233] XFS (dm-0): Mounting V5 Filesystem
[ 6.585021] XFS (sda1): Mounting V5 Filesystem
```

Рис. 16: Тип файловой системы и гипервизора

#### 4.3 Ответы на контрольные вопросы

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

3. 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.
4. 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`.
  5. 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