

Отчёт по лабораторной работе №1

**Установка и конфигурация операционной системы на виртуальную
машину**

Акунаева Антонина Эрдниевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Домашнее задание	14
4	Выводы	17
	Список литературы	18

Список иллюстраций

2.1	Создание ОС Rocky Linux	6
2.2	Виртуальное оборудование Rocky Linux	7
2.3	Виртуальный жёсткий диск Rocky Linux	7
2.4	Окно установки Rocky Linux	8
2.5	Настройка Rocky Linux: язык	8
2.6	Настройка Rocky Linux: оборудование	9
2.7	Настройка Rocky Linux: раскладка клавиатуры	9
2.8	Настройка Rocky Linux: поддержка языков	10
2.9	Настройка Rocky Linux: KDUMP	10
2.10	Настройка Rocky Linux: настройка сети	11
2.11	Настройка Rocky Linux: добавление пароля root	11
2.12	Настройка Rocky Linux: добавление администратора	12
2.13	Завершение настройки Rocky Linux	12
2.14	Завершение установки Rocky Linux	13
2.15	Подключение образа диска дополнений гостевой ОС	13
3.1	Использование команды dmesg less	14
3.2	Нахождение версии ядра Linux при помощи dmesg grep -i	15
3.3	Нахождение частоты процессора при помощи dmesg grep -i	15
3.4	Нахождение модели процессора при помощи dmesg grep -i	15
3.5	Нахождение доступной оперативной памяти при помощи dmesg grep -i	16
3.6	Нахождение типа обнаруженного гипервизора при помощи dmesg grep -i	16
3.7	Нахождение информации о файловых системах при помощи dmesg grep -i	16

Список таблиц

1 Цель работы

Целью данной работы является приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов. [1]

2 Выполнение лабораторной работы

Запускаем Oracle VirtualBox и создаём операционную систему виртуальной машины - Rocky (дистрибутив Linux). Задаём имя и остальные параметры, указываем образ диска ОС (рис. 2.1).

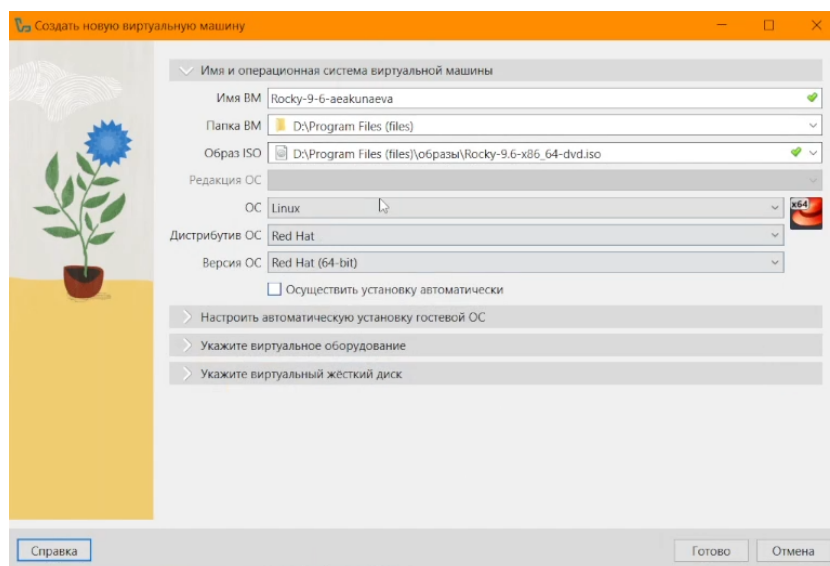


Рис. 2.1: Создание ОС Rocky Linux

Задаём значения для выделяемой основной памяти и ЦПУ как 4096 МБ и 2 ядра (рис. 2.2).

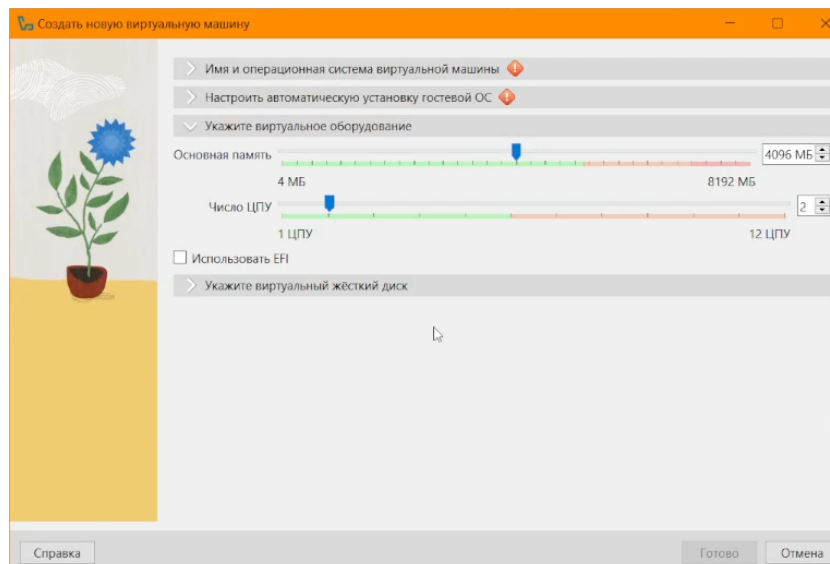


Рис. 2.2: Виртуальное оборудование Rocky Linux

Выделяем для жёсткого диска 40 ГБ памяти (рис. 2.3). Затем запускаем виртуальную машину.

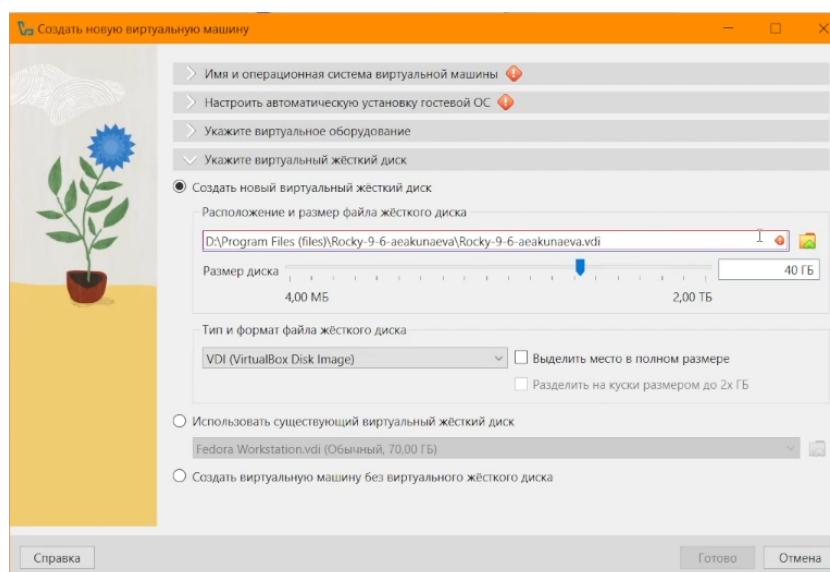


Рис. 2.3: Виртуальный жёсткий диск Rocky Linux

В окне запущенной ВМ выбираем *Install Rocky Linux 9.6* (рис. 2.4).

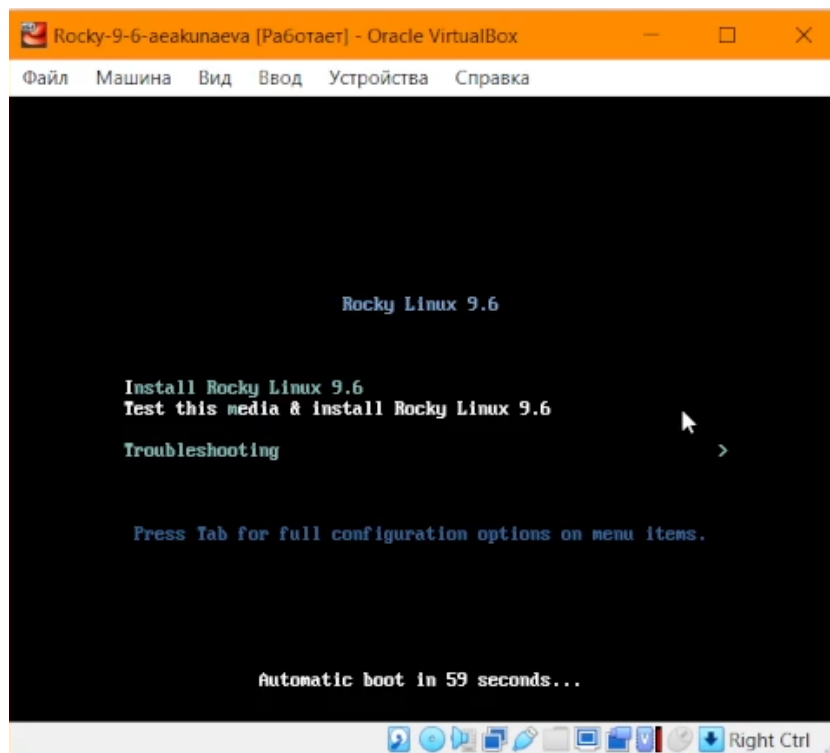


Рис. 2.4: Окно установки Rocky Linux

Выбираем основным языком ОС - английский (США) (рис. 2.5).

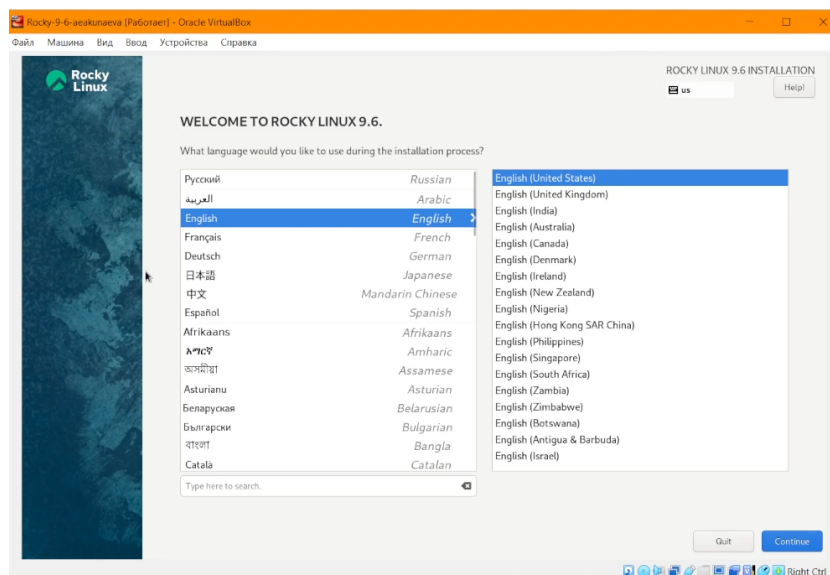


Рис. 2.5: Настройка Rocky Linux: язык

Во вкладке *Software selection* выбираем *Server with GUI*, а справа находим и от-

мечаем *Development Tools*, как указано в требованиях лабораторной работы (рис. 2.6).

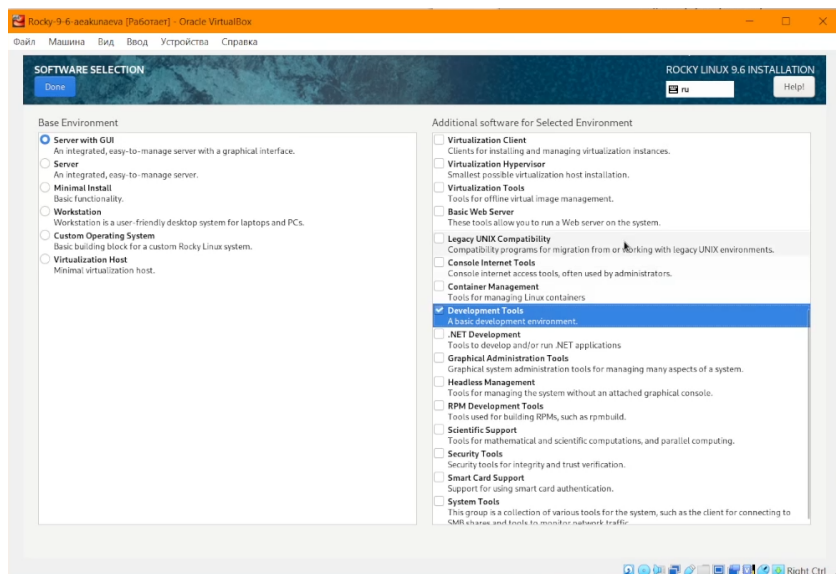


Рис. 2.6: Настройка Rocky Linux: оборудование

Добавляем во вкладке раскладки клавиатуры русский язык (Россия) и настраиваем удобный шорткат для смены языков (в нашем случае *Alt+Shift*) (рис. 2.7).

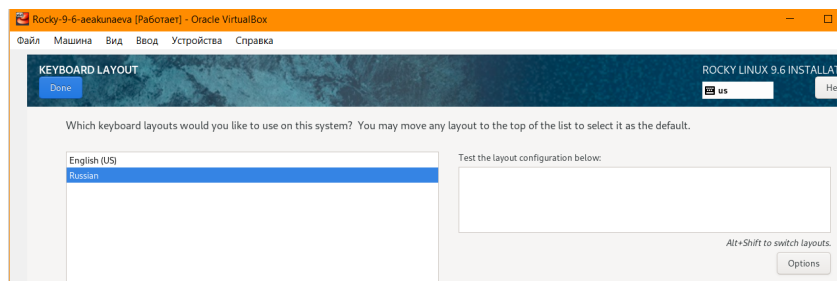


Рис. 2.7: Настройка Rocky Linux: раскладка клавиатуры

Добавляем поддержку русского языка (рис. 2.8).

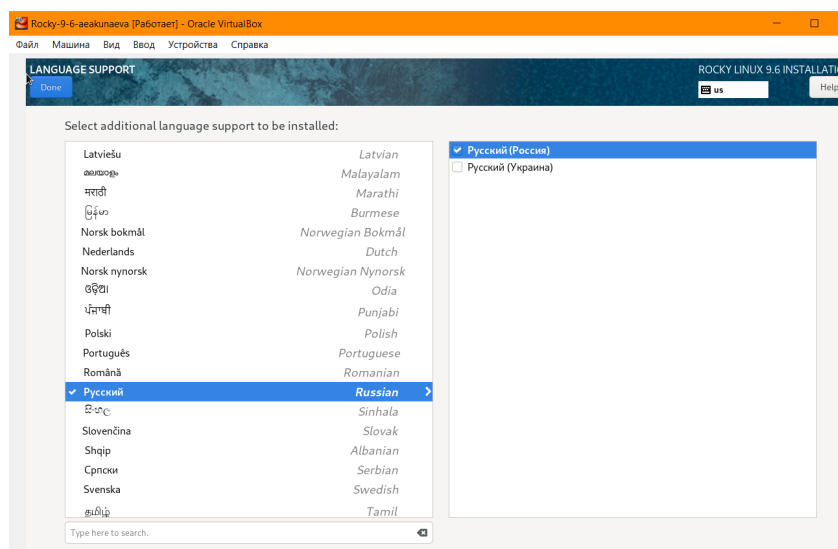


Рис. 2.8: Настройка Rocky Linux: поддержка языков

Отключаем KDUMP (рис. 2.9).

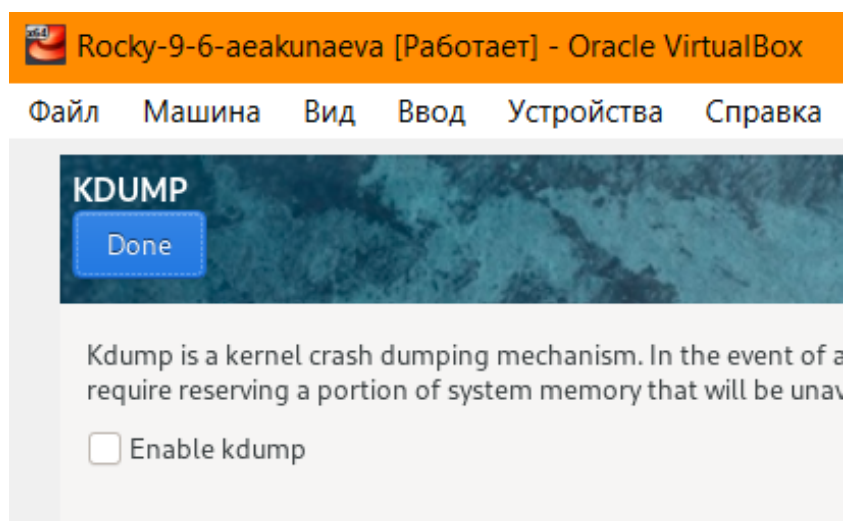


Рис. 2.9: Настройка Rocky Linux: KDUMP

Во вкладке настройки сети подключаем её и изменяем имя хоста на собственное (*aeakunaeva.localmain*) (рис. 2.10).

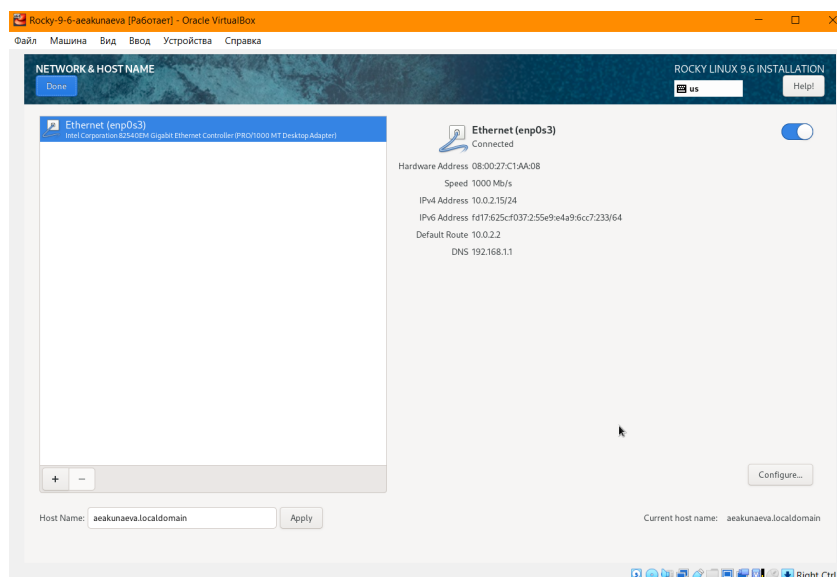


Рис. 2.10: Настройка Rocky Linux: настройка сети

Устанавливаем пароль для администрирования и добавляем возможность логина SSH с этим паролем (рис. 2.11). Также добавляем аккаунт администратора с паролем (рис. 2.12).

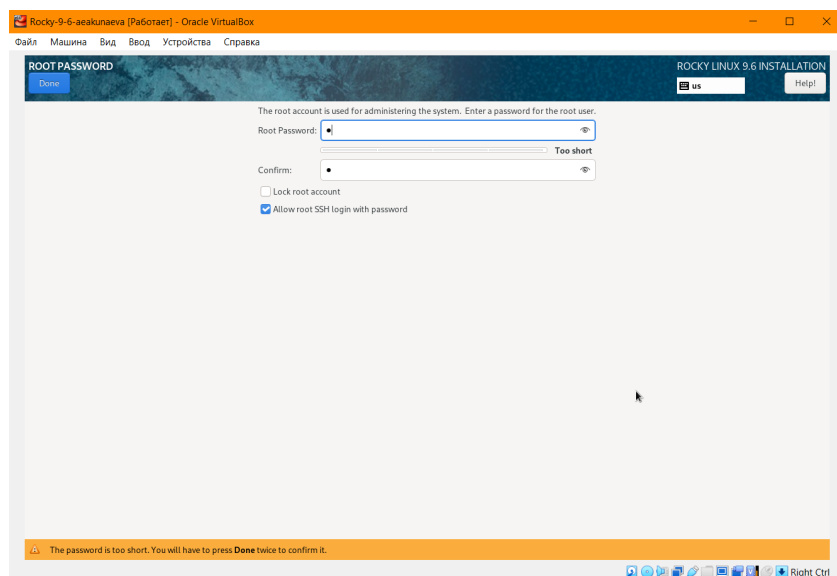


Рис. 2.11: Настройка Rocky Linux: добавление пароля root

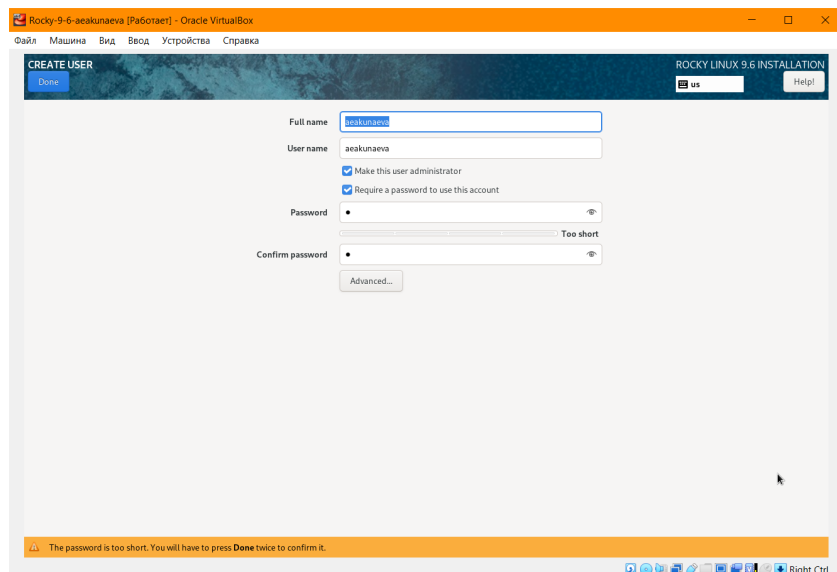


Рис. 2.12: Настройка Rocky Linux: добавление администратора

После завершения настройки устанавливаем ОС (рис. 2.13-рис. 2.14).

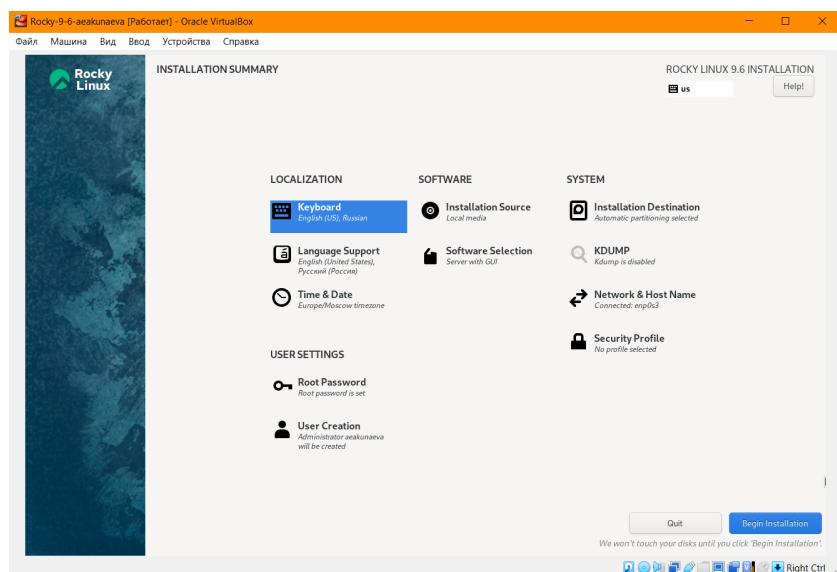


Рис. 2.13: Завершение настройки Rocky Linux

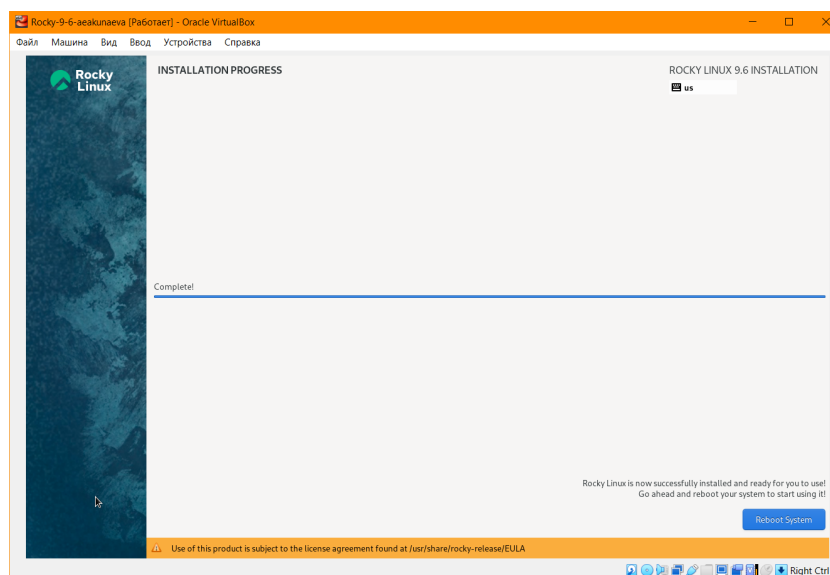


Рис. 2.14: Завершение установки Rocky Linux

Войдя под своим аккаунтом администратора, выбираем у ВМ *Устройства* -> *Подключить образ диска дополнений гостевой ОС*, дожидаемся установки и перезапускаем виртуальную машину (рис. 2.15).

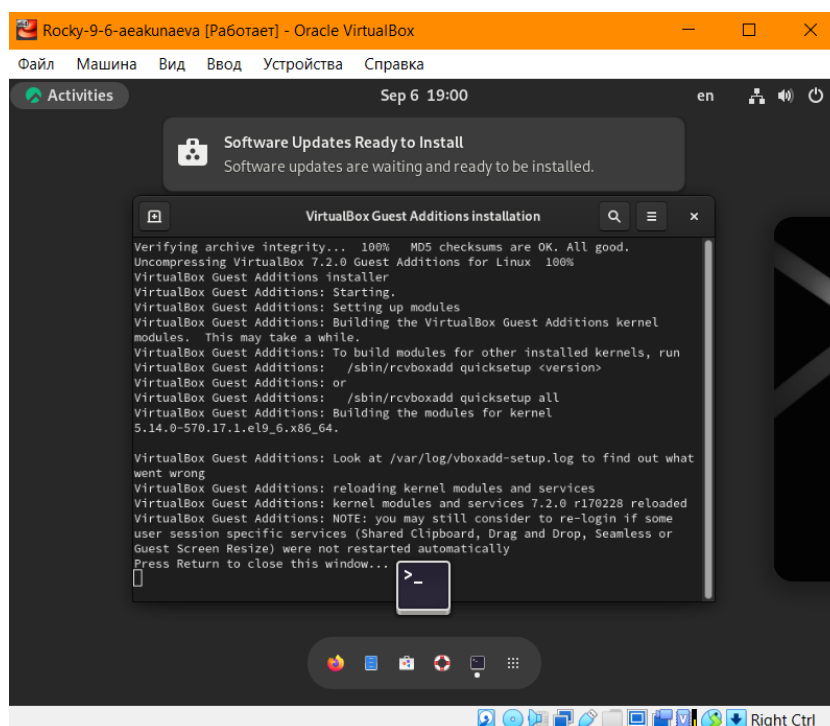
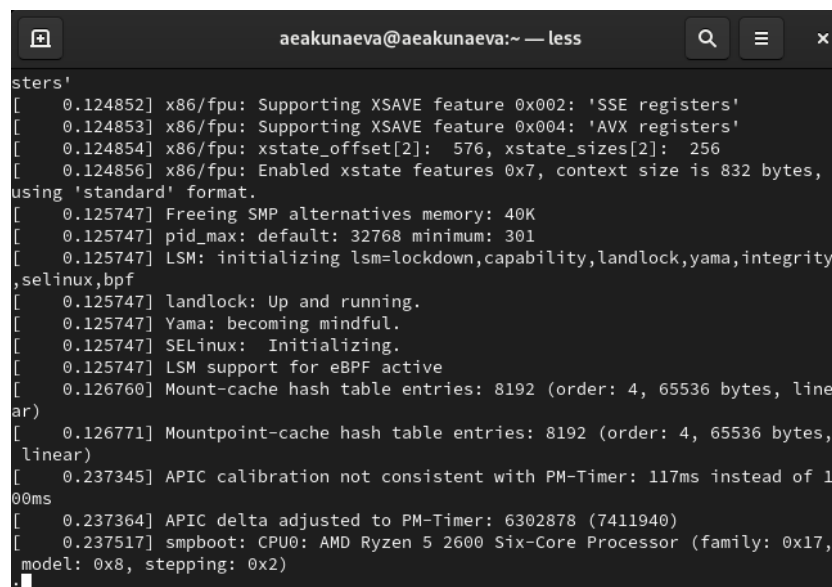


Рис. 2.15: Подключение образа диска дополнений гостевой ОС

3 Домашнее задание

Войдём в систему под своим аккаунтом, в терминале пропишем команду и ознакомимся с её выводом (рис. 3.1):

`dmesg | less`



```
aeakunaeva@aeakunaeva:~ — less
sters'
[ 0.124852] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[ 0.124853] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[ 0.124854] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[ 0.124856] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes,
using 'standard' format.
[ 0.125747] Freeing SMP alternatives memory: 40K
[ 0.125747] pid_max: default: 32768 minimum: 301
[ 0.125747] LSM: initializing lsm=lockdown,capability,landlock,yama,integrity
,selinux,bpf
[ 0.125747] landlock: Up and running.
[ 0.125747] Yama: becoming mindful.
[ 0.125747] SELinux: Initializing.
[ 0.125747] LSM support for eBPF active
[ 0.126760] Mount-cache hash table entries: 8192 (order: 4, 65536 bytes, line
ar)
[ 0.126771] Mountpoint-cache hash table entries: 8192 (order: 4, 65536 bytes,
linear)
[ 0.237345] APIC calibration not consistent with PM-Timer: 117ms instead of 1
00ms
[ 0.237364] APIC delta adjusted to PM-Timer: 6302878 (7411940)
[ 0.237517] smpboot: CPU0: AMD Ryzen 5 2600 Six-Core Processor (family: 0x17,
model: 0x8, stepping: 0x2)
```

Рис. 3.1: Использование команды `dmesg | less`

Далее используем команду `dmesg`, чтобы определить некоторые данные:

1. Версия ядра Linux - 5.14.0-570.17.1.el9_6.x86_64 (рис. 3.2):

`dmesg | grep -i "version"`

```
aeakunaeva@aeakunaeva:~$ dmesg |grep -i "version"
[ 0.000000] Linux version 5.14.0-570.17.1.el9_6.x86_64 (mockbuild@iad1-prod-b
uild001.bld.equ.rockylinux.org) (gcc (GCC) 11.5.0 20240719 (Red Hat 11.5.0-5), G
NU ld version 2.35.2-63.el9) #1 SMP PREEMPT_DYNAMIC Fri May 23 22:47:01 UTC 2025
[ 0.032353] IOAPIC[0]: apic_id 2, version 32, address 0xfec00000, GSI 0-23
[ 0.251960] acpiphp: ACPI Hot Plug PCI Controller Driver version: 0.5
[ 0.354010] Block layer SCSI generic (bsg) driver version 0.4 loaded (major 2
46)
[ 0.356840] shpchp: Standard Hot Plug PCI Controller Driver version: 0.4
[ 1.050081] AVX2 version of gcm_enc/dec engaged.
[ 1.050516] registered taskstats version 1
[ 1.484785] fuse: init (API version 7.37)
[ 1.742153] device-mapper: uevent: version 1.0.3
[ 2.091075] vboxguest: Successfully loaded version 7.2.0 r170228
[ 2.091116] vboxguest: Successfully loaded version 7.2.0 r170228 (interface 0
x00010004)
[ 2.093352] libata version 3.00 loaded.
[ 2.099853] ata_piix 0000:00:01.1: version 2.13
[ 2.110974] ahci 0000:00:0d.0: version 3.0
[ 2.533144] vmwgfx 0000:00:02.0: [drm] Running on SVGA version 2.
[ 21.494603] 16:02:15.576918 main OS Version: #1 SMP PREEMPT_DYNAMIC Fri M
ay 23 22:47:01 UTC 2025
```

Рис. 3.2: Нахождение версии ядра Linux при помощи dmesg | grep -i

2. Частота процессора - 3393.628 MHz (рис. 3.3):

dmesg | grep -i "processor"

```
[aeakunaeva@aeakunaeva ~]$ dmesg |grep -i "processor"
[ 0.000032] tsc: Detected 3393.628 MHz processor
[ 0.237517] smpboot: CPU0: AMD Ryzen 5 2600 Six-Core Processor (family: 0x17,
model: 0x8, stepping: 0x2)
[ 0.245808] smpboot: Total of 2 processors activated (13574.51 BogoMIPS)
[ 0.261893] ACPI: Added _OSI(Processor Device)
[ 0.261897] ACPI: Added _OSI(Processor Aggregator Device)
[aeakunaeva@aeakunaeva ~]$
```

Рис. 3.3: Нахождение частоты процессора при помощи dmesg | grep -i

3. Модель процессора - AMD Ryzen 5 2600 Six-Core Processor (рис. 3.4):

dmesg | grep -i "CPU0"

```
[aeakunaeva@aeakunaeva ~]$ dmesg |grep -i "CPU0"
[ 0.237517] smpboot: CPU0: AMD Ryzen 5 2600 Six-Core Processor (family: 0x17,
model: 0x8, stepping: 0x2)
```

Рис. 3.4: Нахождение модели процессора при помощи dmesg | grep -i

4. Объем доступной оперативной памяти - 3675284K/4193848K (рис. 3.5):

dmesg | grep -i "memory"

```
aeakunaeva@aeakunaeva:~  
[ 0.032408] PM: hibernation: Registered nosave memory: [mem 0xfec00000-0xfec0  
0fff]  
[ 0.032408] PM: hibernation: Registered nosave memory: [mem 0xfec01000-0xfedf  
ffff]  
[ 0.032409] PM: hibernation: Registered nosave memory: [mem 0xfec00000-0xfec0  
0fff]  
[ 0.032409] PM: hibernation: Registered nosave memory: [mem 0xfec01000-0xffffb  
ffff]  
[ 0.032410] PM: hibernation: Registered nosave memory: [mem 0xffffc0000-0xfffff  
ffff]  
[ 0.100129] Memory: 3675284K/4193848K available (16384K kernel code, 5766K rw  
data, 13624K rodata, 4048K init, 7384K bss, 253956K reserved, 0K cma-reserved)  
[ 0.125747] Freeing SMP alternatives memory: 40K  
[ 0.248816] x86/mm: Memory block size: 128MB  
[ 0.359377] Non-volatile memory driver v1.3  
[ 1.041026] Freeing initrd memory: 58920K  
[ 1.276279] Freeing unused decrypted memory: 2028K  
[ 1.277620] Freeing unused kernel image (initmem) memory: 4048K  
[ 1.278255] Freeing unused kernel image (rodata/data gap) memory: 712K  
[ 2.533312] vmwgfx 0000:00:02.0: [drm] Legacy memory limits: VRAM = 16384 KiB  
FIFO = 2048 KiB, surface = 507904 KiB  
[ 2.533323] vmwgfx 0000:00:02.0: [drm] Maximum display memory size is 16384 K  
iB
```

Рис. 3.5: Нахождение доступной оперативной памяти при помощи `dmesg | grep -i`

5. Тип обнаруженного гипервизора - KVM (рис. 3.6):

`dmesg | grep -i "hypervisor"`

```
[aeakunaeva@aeakunaeva ~]$ dmesg |grep -i "hypervisor"  
[ 0.000000] Hypervisor detected: KVM  
[ 2.533155] vmwgfx 0000:00:02.0: [drm] *ERROR* vmwgfx seems to be running on  
an unsupported hypervisor.
```

Рис. 3.6: Нахождение типа обнаруженного гипервизора при помощи `dmesg | grep -i`

6-7. Тип файловой системы корневого раздела - XFS (dm-0). Здесь же видна последовательность монтирования файловых систем (рис. 3.7):

`dmesg | grep -i "filesystem"`

```
[aeakunaeva@aeakunaeva ~]$ dmesg |grep -i "filesystem"  
[ 3.346618] XFS (dm-0): Mounting V5 Filesystem 41c3393a-58bf-4486-a6eb-2155f7  
10615a  
[ 16.216333] XFS (sda1): Mounting V5 Filesystem 25a77cb0-b697-4d7d-8545-944e2e  
52929c
```

Рис. 3.7: Нахождение информации о файловых системах при помощи `dmesg | grep -i`

4 Выводы

Я приобрела практические навыки установки операционной системы на виртуальную машину и настройки минимально необходимых для дальнейшей работы сервисов.

Список литературы

1. Кулябов Д.С. ТУИС РУДН.