

# **Лабораторная работа №1**

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

Смородова Дарья Владимировна

2022 Sep 10th

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Задание</b>	<b>6</b>
<b>3 Теоретическое введение</b>	<b>7</b>
<b>4 Выполнение лабораторной работы</b>	<b>8</b>
<b>5 Контрольные вопросы</b>	<b>27</b>
<b>6 Выводы</b>	<b>30</b>
<b>7 Список литературы</b>	<b>31</b>

# **List of Tables**

# List of Figures

4.1	Имя машины и тип ОС . . . . .	8
4.2	Объем основной памяти . . . . .	9
4.3	Создание нового жесткого диска . . . . .	10
4.4	Указание типа жесткого диска . . . . .	10
4.5	Определение формата хранения жесткого диска . . . . .	11
4.6	Определение размера и расположения жесткого диска . . . . .	11
4.7	Подключение образа оптического диска . . . . .	12
4.8	Виртуальная машина . . . . .	12
4.9	Запуск виртуальной машины . . . . .	13
4.10	Выбор языка для процесса установки . . . . .	14
4.11	Выбор программ . . . . .	15
4.12	Отключение KDUMP . . . . .	16
4.13	Выбор места установки . . . . .	17
4.14	Настройка сети и имени узла . . . . .	18
4.15	Установка пароля для root . . . . .	19
4.16	Создание пользователя . . . . .	20
4.17	Завершение установки ОС . . . . .	21
4.18	Установка драйверов . . . . .	22
4.19	Завершение установки драйверов . . . . .	22
4.20	dmesg . . . . .	23
4.21	dmesg   less . . . . .	23
4.22	Linux version . . . . .	24
4.23	Detected MHz processor . . . . .	24
4.24	CPU0 . . . . .	24
4.25	Memory available . . . . .	25
4.26	Hypervisor detected . . . . .	25
4.27	Root . . . . .	26
4.28	Mounting . . . . .	26

# **1 Цель работы**

Приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

## **2 Задание**

1. Установить операционную систему Linux с дистрибутивом Rocky на виртуальную машину VirtualBox;
2. Настроить необходимые для работы сервисы.
3. Узнать информацию о системе при помощи команды dmesg.

### **3 Теоретическое введение**

Linux — это семейство операционных систем (ОС), работающих на основе одноименного ядра.

Линус Торвальдс — первый разработчик и создатель Linux. Именно в честь него и была названа ОС. В 1981 году Линус начал работу над собственной ОС семейства Unix. Через три года появилась первая версия, доступная для скачивания.<sup>1</sup>

Дистрибутив — форма распространения программного обеспечения.

Rocky Linux - это корпоративная операционная система с открытым исходным кодом, разработанная для 100% совместимости с Red Hat Enterprise Linux (RHEL). Rocky Linux собирает исходные тексты непосредственно из RHEL.<sup>2</sup>

Операционная система CentOS (сокращенно от английского «Community ENTerprise Operating System») — дистрибутив Linux, основанный на коммерческом дистрибутиве Red Hat Enterprise Linux компании Red Hat, который предназначен для корпоративного использования.<sup>3</sup>

---

<sup>1</sup>Linux

<sup>2</sup>Rocky Linux

<sup>3</sup>CentOS Linux

## 4 Выполнение лабораторной работы

1. Откроем VirtualBox и начнем создание виртуальной машины с операционной системой Linux.
2. Зададим имя машины и тип операционной системы (рис. 4.1):

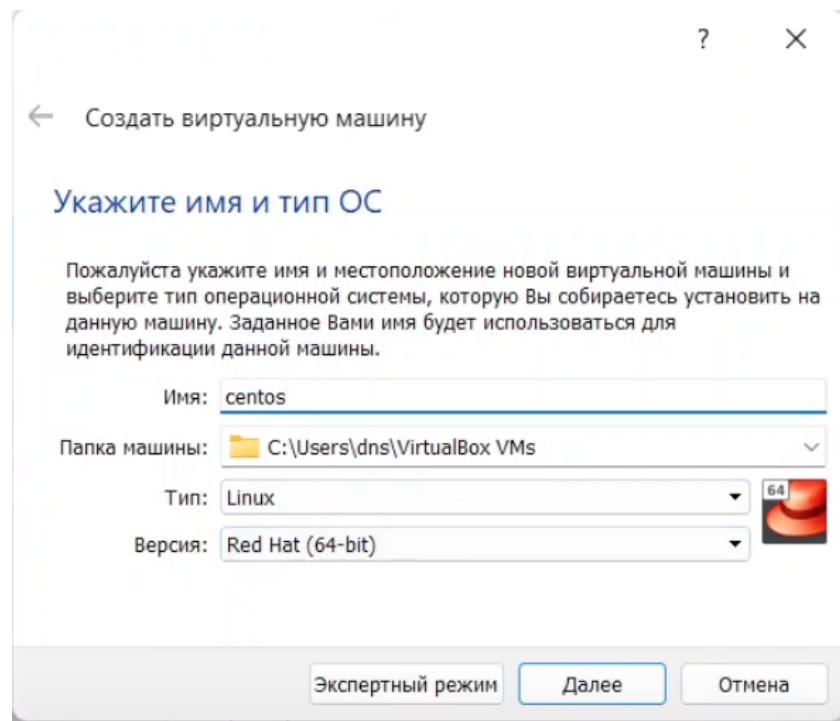


Figure 4.1: Имя машины и тип ОС

3. Укажем размер основной памяти виртуальной машины — 2048 МБ (рис. 4.2):

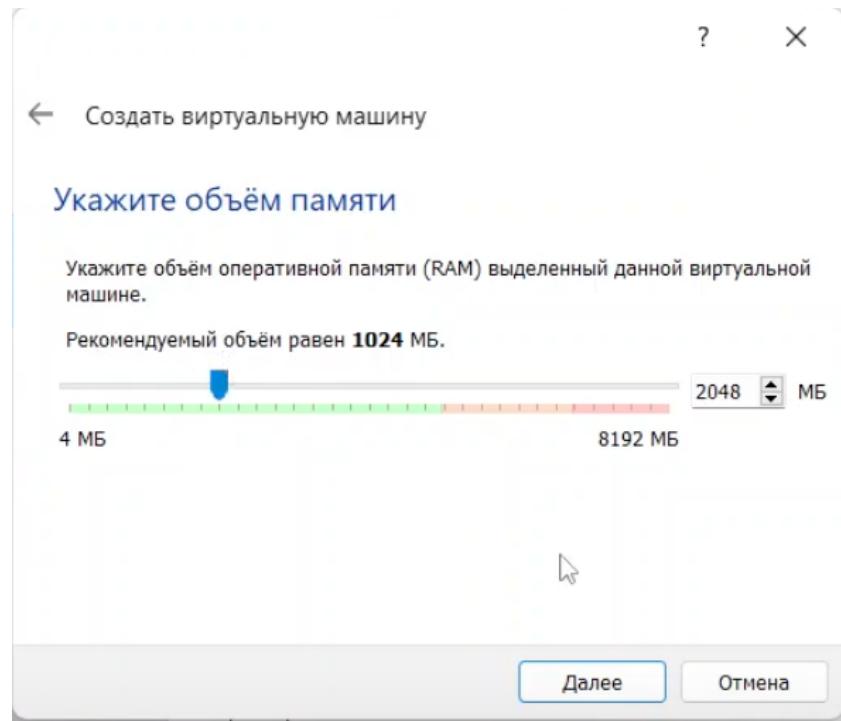


Figure 4.2: Объем основной памяти

4. Создадим новый виртуальный жесткий диск и зададим конфигурацию жесткого диска — загрузочный, VDI (VirtualBox Disk Image), динамический виртуальный диск. Зададим размер диска — 20 ГБ и его расположение (рис. 4.3 - 4.6):

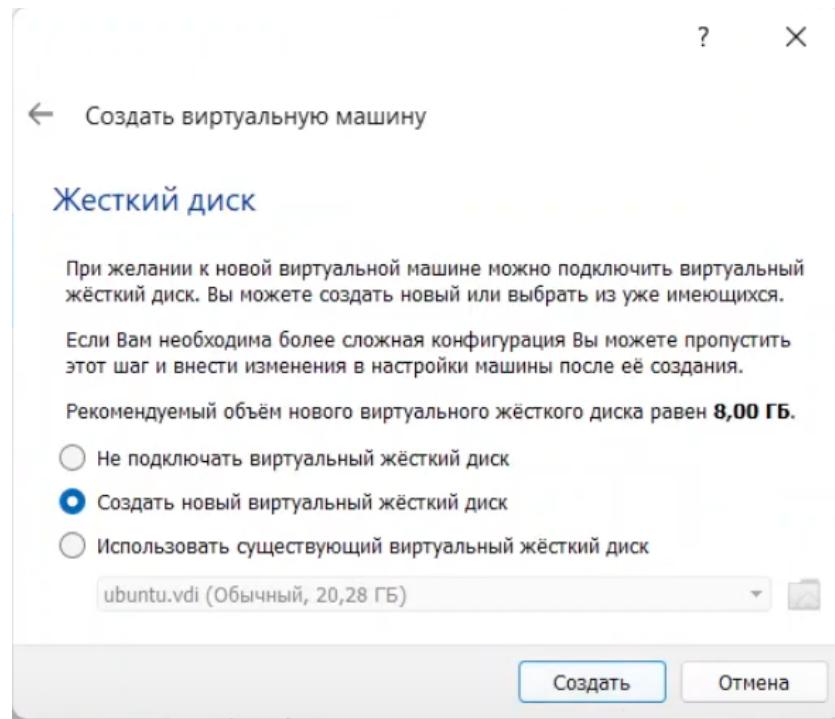


Figure 4.3: Создание нового жесткого диска

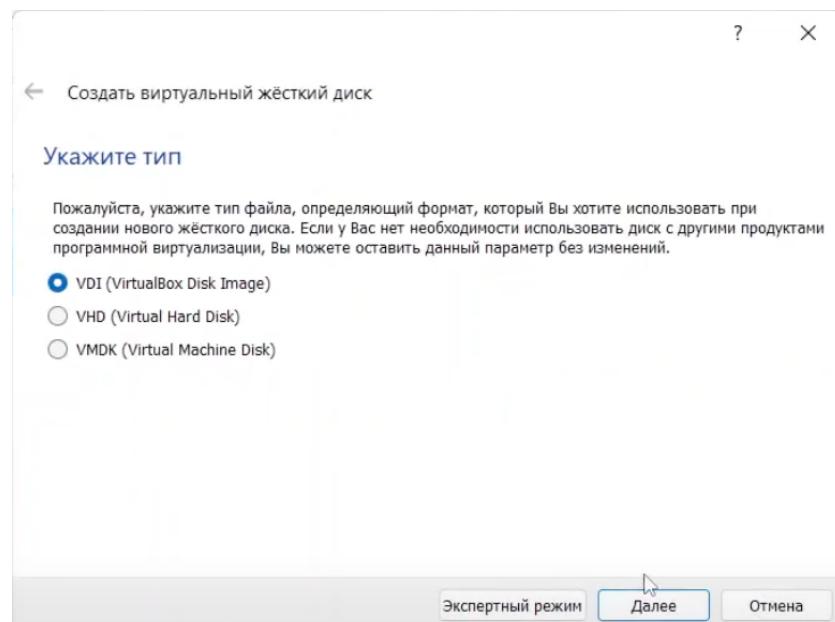


Figure 4.4: Указание типа жесткого диска

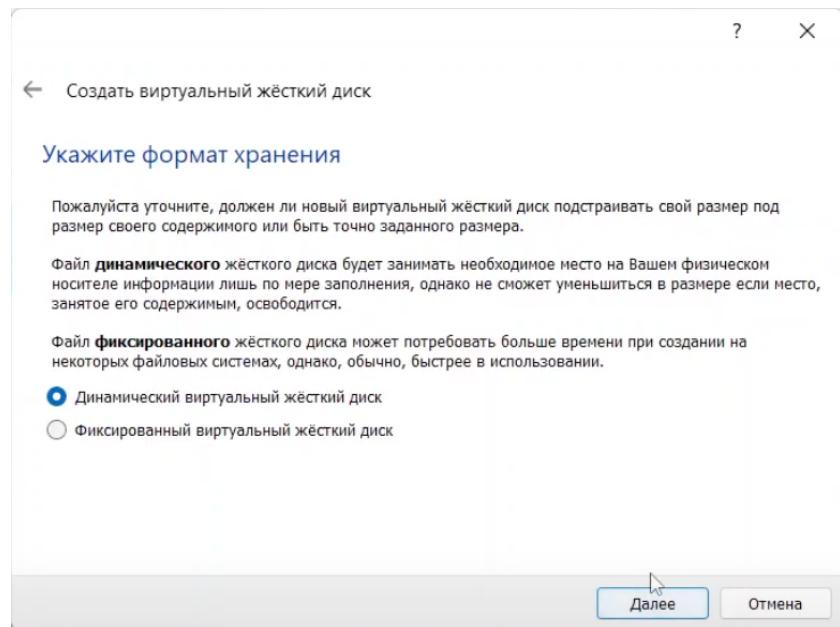


Figure 4.5: Определение формата хранения жесткого диска

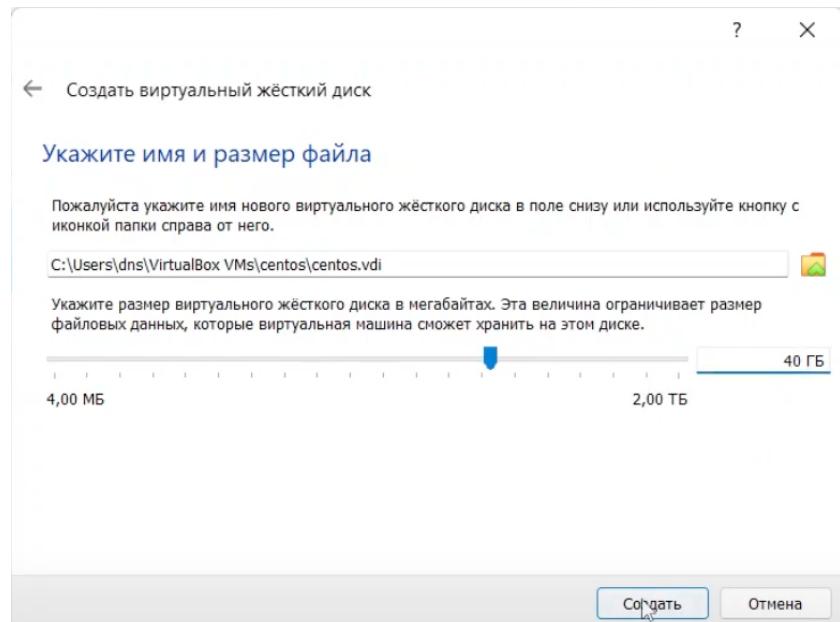


Figure 4.6: Определение размера и расположения жесткого диска

5. Подключим образ оптического диска (рис. 4.7):

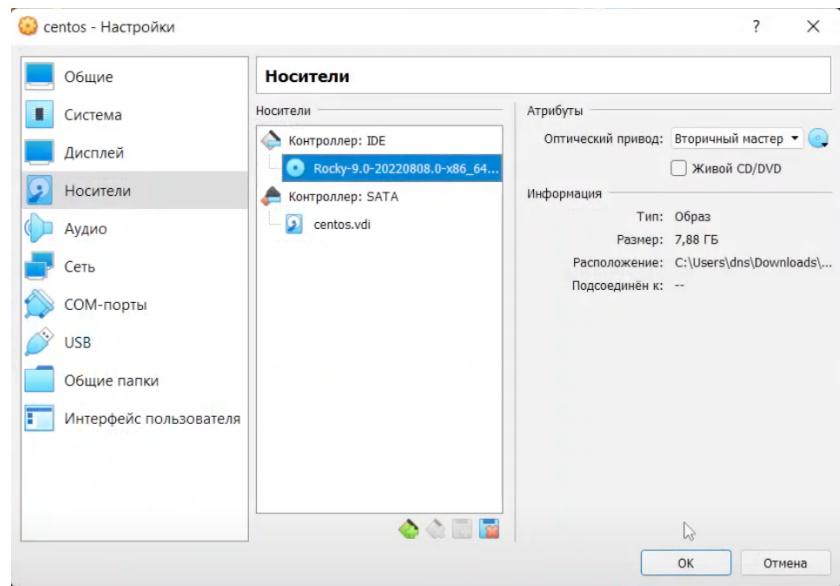


Figure 4.7: Подключение образа оптического диска

6. Посмотрим данные виртуальной машины (рис. 4.8):

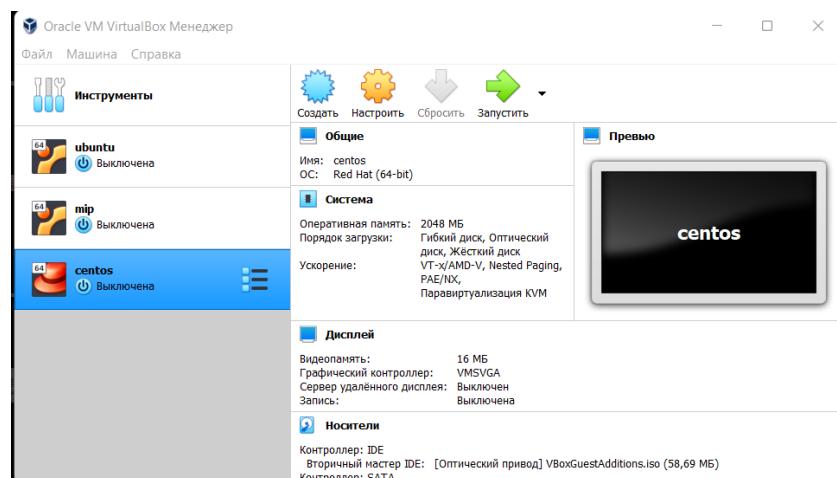


Figure 4.8: Виртуальная машина

7. Запустим виртуальную машину (рис. 4.9):

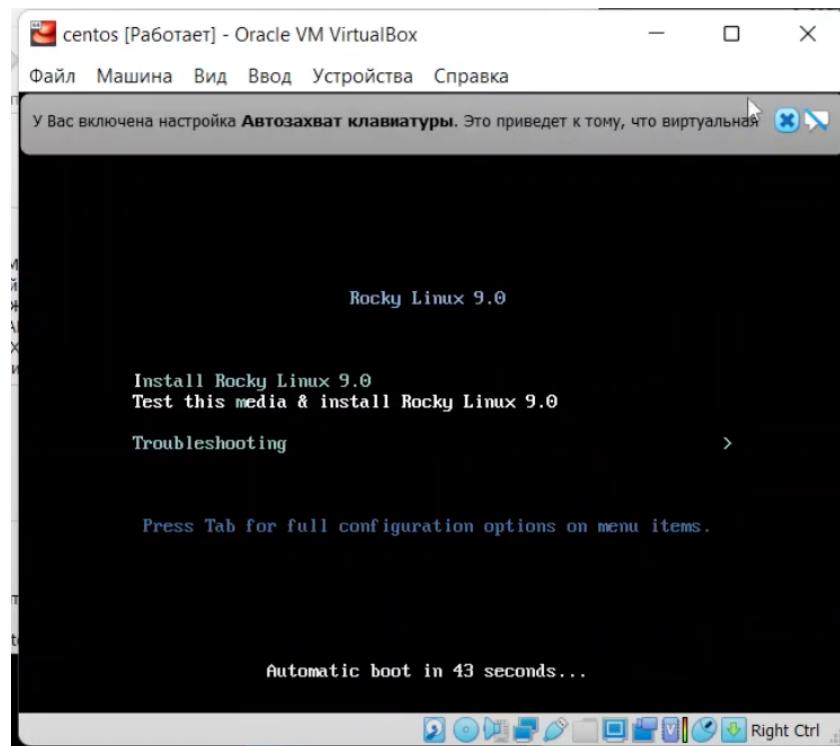


Figure 4.9: Запуск виртуальной машины

8. Установим русскоязычный интерфейс (рис. 4.10):

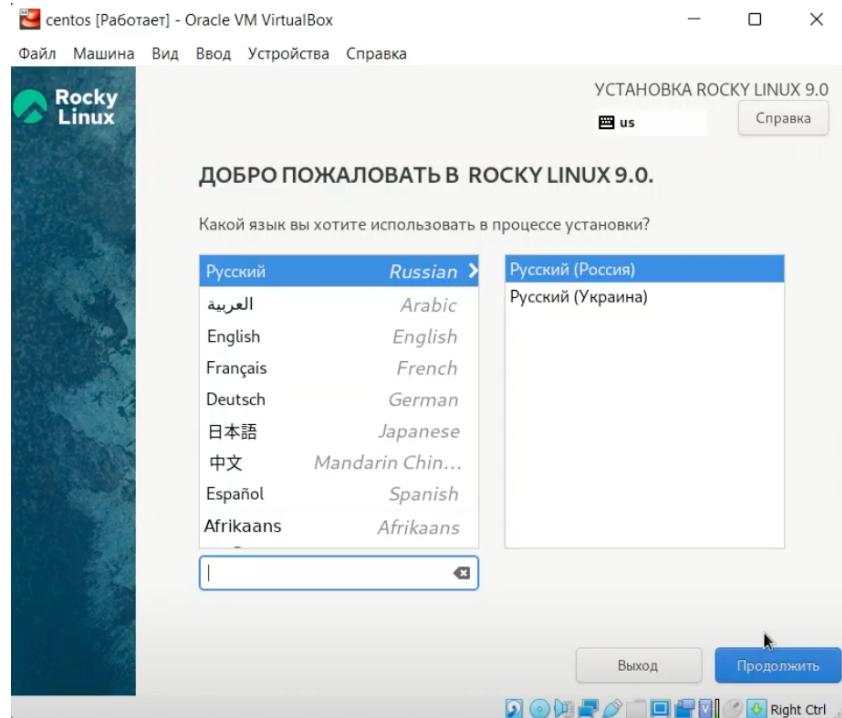


Figure 4.10: Выбор языка для процесса установки

9. Установим сервер с GUI и средства разработки (рис. 4.11):

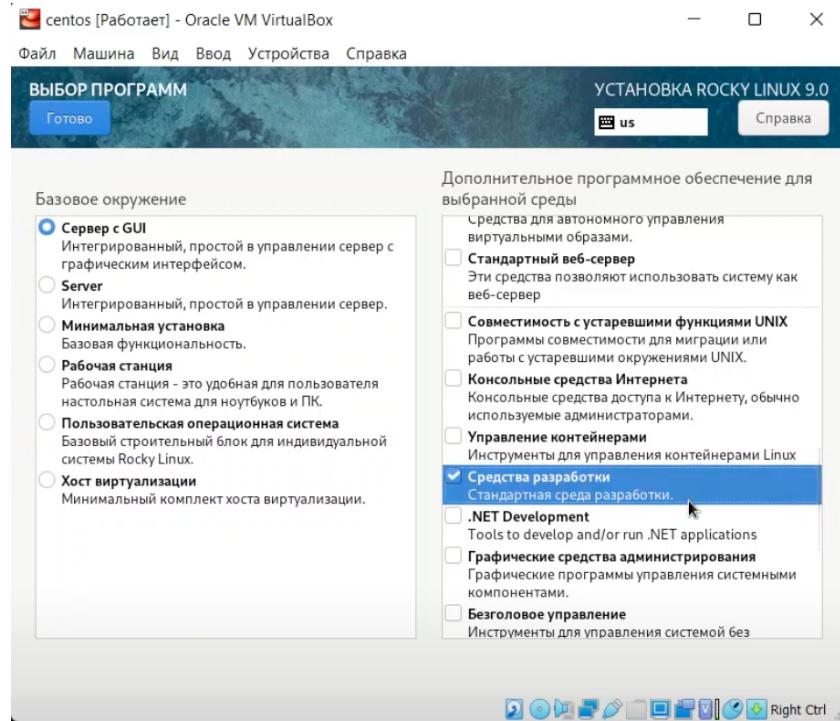


Figure 4.11: Выбор программ

10. Отключим KDUMP (рис. 4.12):

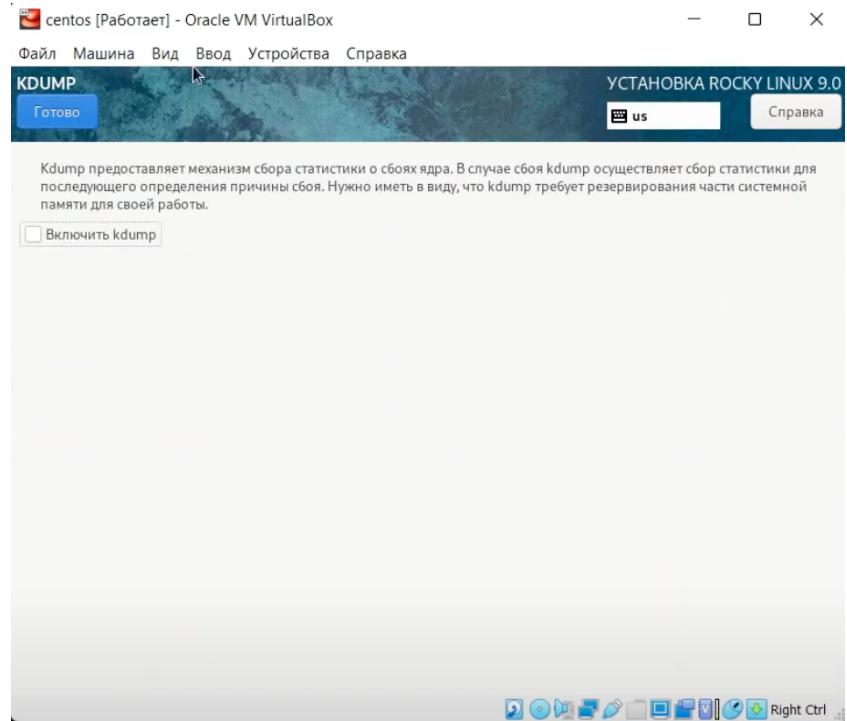


Figure 4.12: Отключение KDUMP

11. Выберем место установки (рис. 4.13):

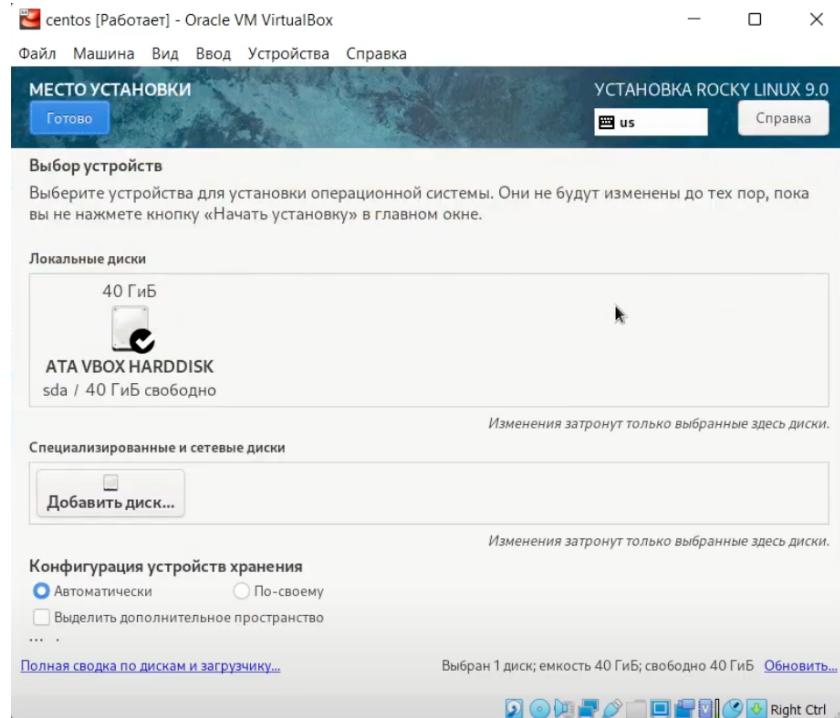


Figure 4.13: Выбор места установки

12. Настроим сеть и укажем «smorodovadv.localdomain» в качестве имени узла (рис. 4.14):

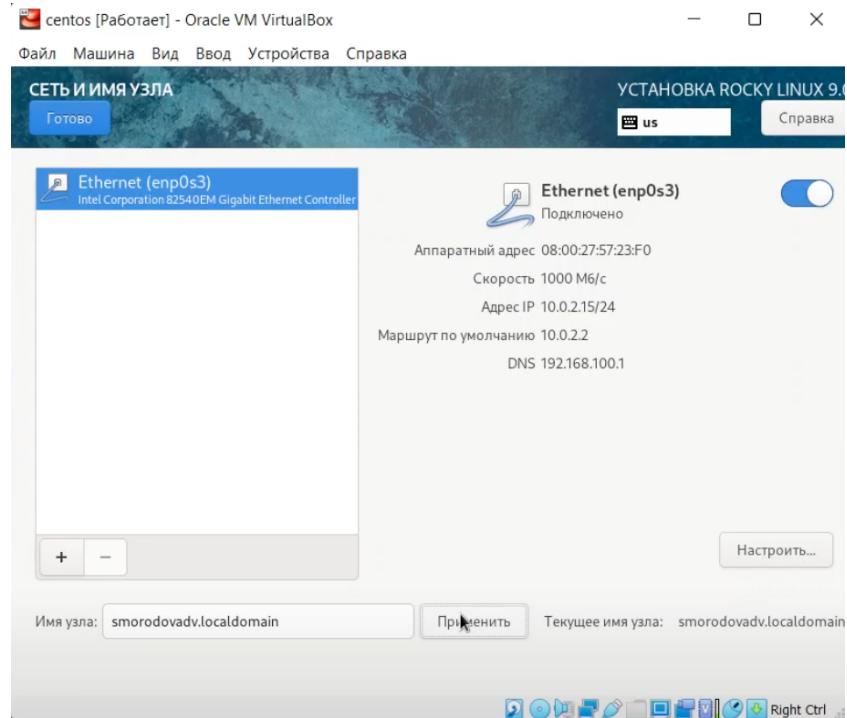


Figure 4.14: Настройка сети и имени узла

13. Установим пароль для root (рис. 4.15):

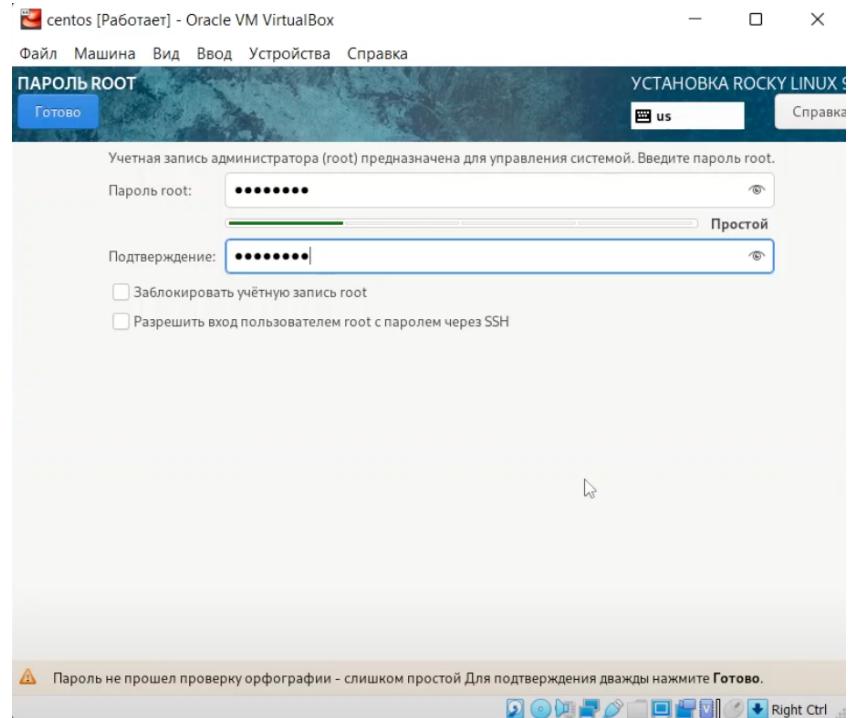


Figure 4.15: Установка пароля для root

14. Создадим пользователя с правами администратора (рис. 4.16):

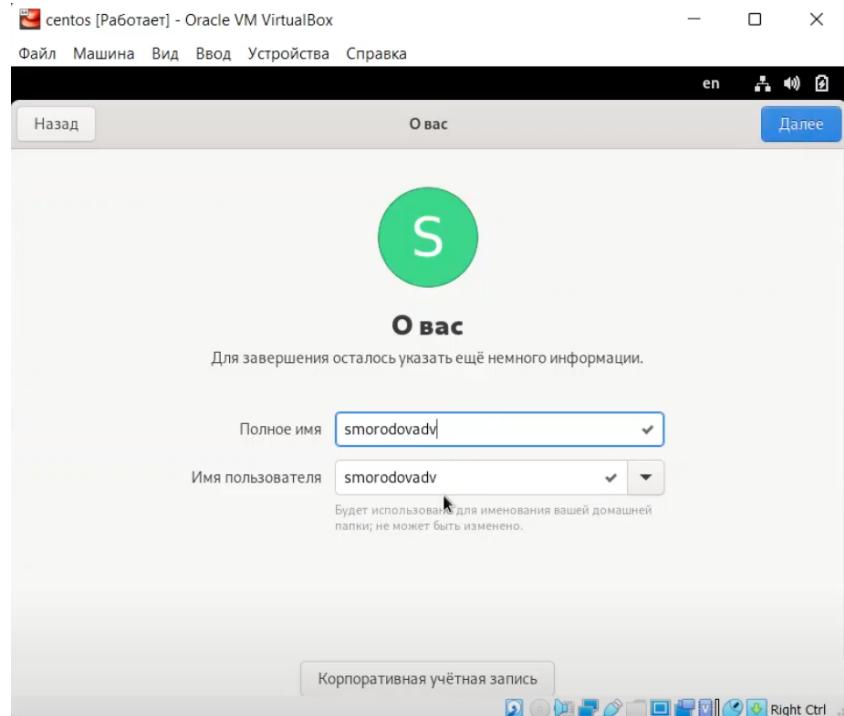


Figure 4.16: Создание пользователя

15. Завершим установку ОС (рис. 4.17):

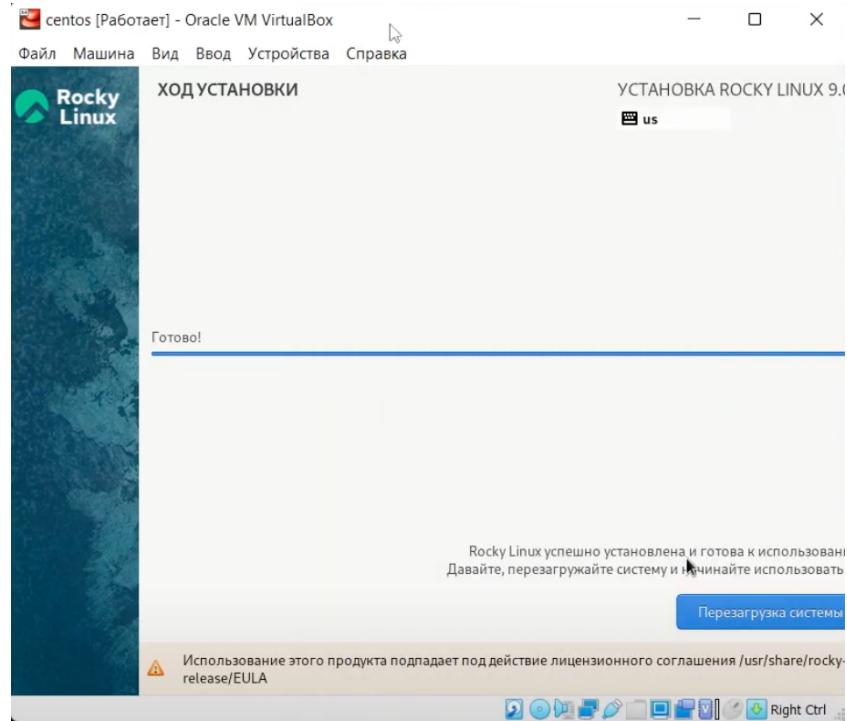


Figure 4.17: Завершение установки ОС

16. Установим необходимые драйвера (рис. 4.18 - 4.19):

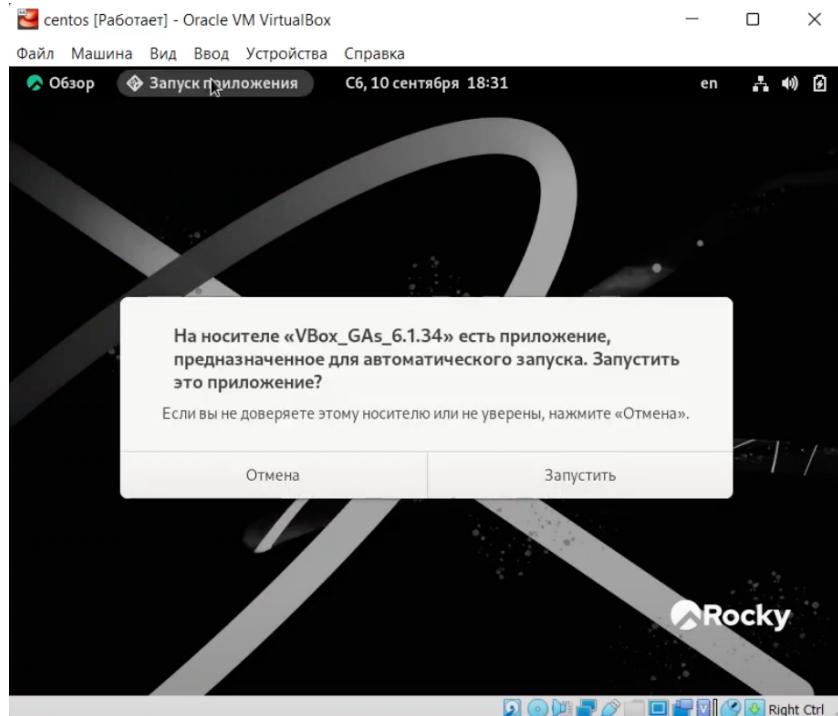


Figure 4.18: Установка драйверов

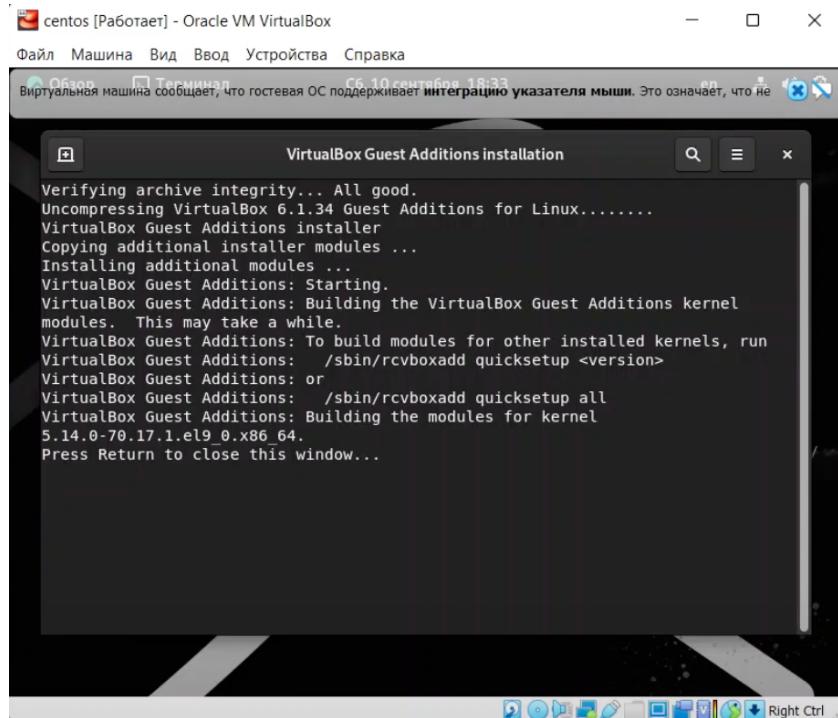
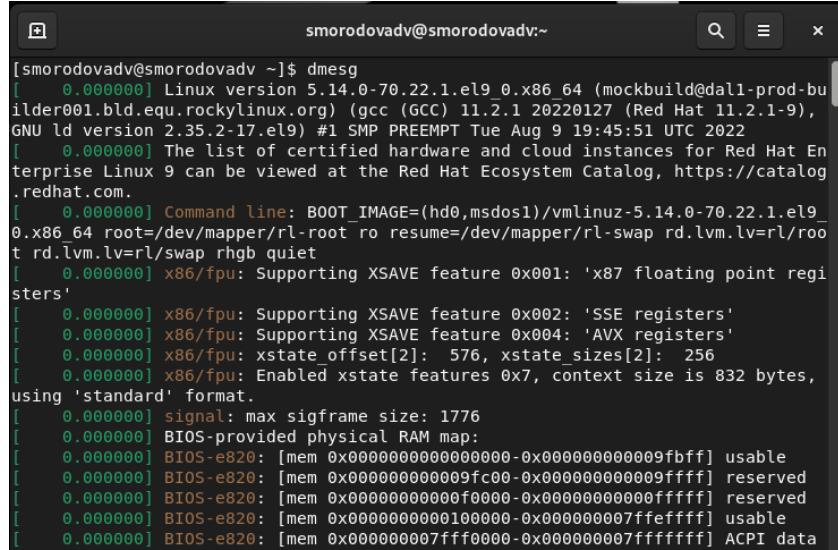


Figure 4.19: Завершение установки драйверов

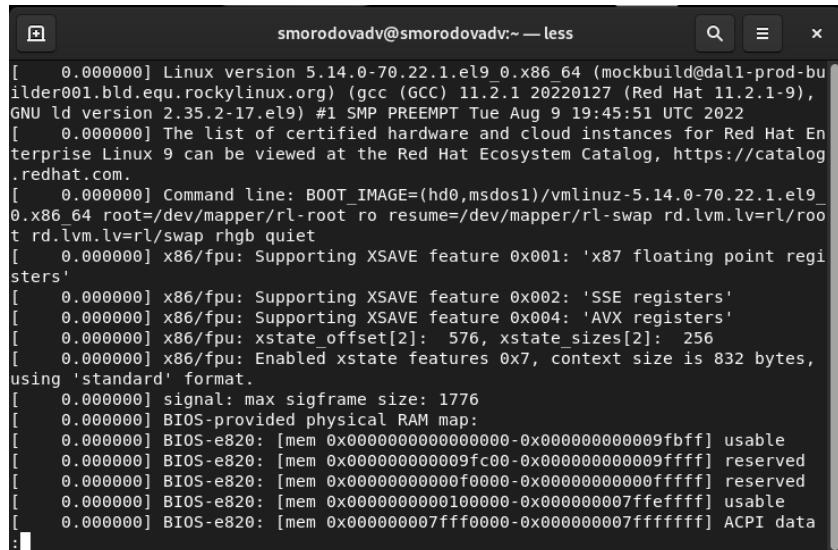
17. Посмотрим всю информацию при помощи команды dmesg (рис. 4.20):



```
[smorodovadv@smorodovadv ~]$ dmesg
[    0.000000] Linux version 5.14.0-70.22.1.el9_0.x86_64 (mockbuild@dal1-prod-builder001.bld.equ.rockylinux.org) (gcc (GCC) 11.2.1 20220127 (Red Hat 11.2.1-9), GNU ld version 2.35.2-17.el9) #1 SMP PREEMPT Tue Aug 9 19:45:51 UTC 2022
[    0.000000] The list of certified hardware and cloud instances for Red Hat Enterprise Linux 9 can be viewed at the Red Hat Ecosystem Catalog, https://catalog.redhat.com.
[    0.000000] Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-5.14.0-70.22.1.el9_0.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 registers'
[    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 0x0000000000000000-0x00000000000fbfff] usable
[    0.000000] BIOS-e820: [mem 0x00000000000fc00-0x00000000000fffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
[    0.000000] BIOS-e820: [mem 0x000000000100000-0x0000000007fffff] usable
[    0.000000] BIOS-e820: [mem 0x0000000007fff0000-0x0000000007ffffff] ACPI data
[    0.000000] :[
```

Figure 4.20: dmesg

18. Посмотрим всю информацию при помощи команды dmesg | less (рис. 4.21):



```
[smorodovadv@smorodovadv:~ — less
[    0.000000] Linux version 5.14.0-70.22.1.el9_0.x86_64 (mockbuild@dal1-prod-builder001.bld.equ.rockylinux.org) (gcc (GCC) 11.2.1 20220127 (Red Hat 11.2.1-9), GNU ld version 2.35.2-17.el9) #1 SMP PREEMPT Tue Aug 9 19:45:51 UTC 2022
[    0.000000] The list of certified hardware and cloud instances for Red Hat Enterprise Linux 9 can be viewed at the Red Hat Ecosystem Catalog, https://catalog.redhat.com.
[    0.000000] Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-5.14.0-70.22.1.el9_0.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 registers'
[    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 0x0000000000000000-0x00000000000fbfff] usable
[    0.000000] BIOS-e820: [mem 0x00000000000fc00-0x00000000000fffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
[    0.000000] BIOS-e820: [mem 0x000000000100000-0x0000000007fffff] usable
[    0.000000] BIOS-e820: [mem 0x0000000007fff0000-0x0000000007ffffff] ACPI data
[    0.000000] :[
```

Figure 4.21: dmesg | less

19. С помощью команды grep -i " " получим:

- Версию ядра Linux (Linux version)(рис. 4.22):

```
[smorodovadv@smorodovadv ~]$ dmesg |grep -i "Linux version"
[    0.000000] Linux version 5.14.0-70.22.1.el9_0.x86_64 (mockbuild@dall-prod-builder001.bld.equ.rockylinux.org) (gcc (GCC) 11.2.1 20220127 (Red Hat 11.2.1-9),
GNU ld version 2.35.2-17.el9) #1 SMP PREEMPT Tue Aug 9 19:45:51 UTC 2022
```

Figure 4.22: Linux version

- Частоту процессора (Detected Mhz processor) (рис. 4.23):

```
[smorodovadv@smorodovadv ~]$ dmesg |grep -i "Detected Mhz processor"
[smorodovadv@smorodovadv ~]$ dmesg |grep -i "Detected"
[    0.000000] Hypervisor detected: KVM
[    0.000007] tsc: Detected 1996.801 MHz processor
[    0.382311] hub 1-0:1.0: 12 ports detected
[    0.435368] hub 2-0:1.0: 12 ports detected
[    1.234682] systemd[1]: Detected virtualization oracle.
[    1.234684] systemd[1]: Detected architecture x86-64.
[    1.909560] Warning: Unmaintained hardware is detected: e1000:100E:8086 @ 00
00:00:03.0
[    3.999573] systemd[1]: Detected virtualization oracle.
[    3.999577] systemd[1]: Detected architecture x86-64.
[    5.409316] intel_rapl_msr: PL4 support detected.
[smorodovadv@smorodovadv ~]$ dmesg |grep -i "Detected MHz processor"
```

Figure 4.23: Detected MHz processor

- Модель процессора (CPU0)(рис. 4.24):

```
[smorodovadv@smorodovadv ~]$ dmesg |grep -i CPU0
[    0.155875] smpboot: CPU0: 11th Gen Intel(R) Core(TM) i3-1125G4 @ 2.00GHz (fa
mily: 0x6, model: 0x8c, stepping: 0x1)
```

Figure 4.24: CPU0

- Объем доступной оперативной памяти (Memory available)(рис. 4.25):

```
[smorodovadv@smorodovadv ~]$ dmesg |grep -i "Memory available"
[smorodovadv@smorodovadv ~]$ dmesg |grep -i "Memory"
[ 0.000685] ACPI: Reserving FACP table memory at [mem 0x7fff00f0-0x7fff01e3]
[ 0.000686] ACPI: Reserving DSDT table memory at [mem 0x7fff0470-0x7fff2794]
[ 0.000686] ACPI: Reserving FACS table memory at [mem 0x7fff0200-0x7fff023f]
[ 0.000687] ACPI: Reserving APIC table memory at [mem 0x7fff0200-0x7fff023f]
[ 0.000688] ACPI: Reserving SSDT table memory at [mem 0x7fff0240-0x7fff0293]
[ 0.000688] ACPI: Reserving Early memory node ranges
[ 0.001084] Early memory node ranges
[ 0.002006] PM: hibernation: Registered nosave memory: [mem 0x00000000-0x0000
0fff]
[ 0.002007] PM: hibernation: Registered nosave memory: [mem 0x0009f000-0x0009
ffff]
[ 0.002008] PM: hibernation: Registered nosave memory: [mem 0x000a0000-0x000e
ffff]
[ 0.002008] PM: hibernation: Registered nosave memory: [mem 0x000f0000-0x000f
ffff]
[ 0.017740] Memory: 260860K/2096696K available (14345K kernel code, 5949K rwd
ata, 9056K rodata, 2548K init, 5452K bss, 143008K reserved, 0K cma-reserved)
[ 0.053264] Freeing SMP alternatives memory: 36K
[ 0.167203] x86/mm: Memory block size: 128MB
[ 0.369374] Non-volatile memory driver v1.3
[ 1.006794] Freeing initrd memory: 52768K
[ 1.158105] Freeing unused decrypted memory: 2036K
[ 1.158432] Freeing unused kernel image (initmem) memory: 2548K
[ 1.159196] Freeing unused kernel image (text/rodata gap) memory: 2036K
[ 1.159415] Freeing unused kernel image (rodata/data gap) memory: 1184K
[ 2.318097] [TTM] Zone kernel: Available graphics memory: 1007148 KiB
[ 2.318241] [drm] Max dedicated hypervisor surface memory is 507904 kiB
[ 2.318242] [drm] Maximum display memory size is 16384 kiB
```

Figure 4.25: Memory available

- Тип обнаруженного гипервизора (Hypervisor detected) (рис. 4.26):

```
[smorodovadv@smorodovadv ~]$ dmesg |grep -i "Hypervisor"
[ 0.000000] Hypervisor detected: KVM
[ 2.318241] [drm] Max dedicated hypervisor surface memory is 507904 kiB
```

Figure 4.26: Hypervisor detected

- Тип файловой системы корневого раздела (рис. 4.27):

```
[smorodovadv@smorodovadv ~]$ dmesg |grep -i "Root"
[    0.000000] Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-5.14.0-70.22.1.el9_
0.x86_64 root=/dev/mapper/rl-root ro resume=/dev/mapper/rl-swap rd.lvm.lv=rl/roo
t rd.lvm.lv=rl/swap rhgb quiet
[    0.008167] Kernel command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-5.14.0-70.22
.1.el9 0.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
[    0.175600] ACPI: PCI Root Bridge [PCIO] (domain 0000 [bus 00-ff])
[    0.176321] pci_bus 0000:00: root bus resource [io 0x0000-0x0cf7 window]
[    0.176324] pci_bus 0000:00: root bus resource [io 0xd00-0xffff window]
[    0.176325] pci_bus 0000:00: root bus resource [mem 0x000a0000-0x000bffff win
dow]
[    0.176326] pci_bus 0000:00: root bus resource [mem 0x80000000-0xfdffff window]
[    0.176327] pci_bus 0000:00: root bus resource [bus 00-ff]
[    0.333743] Trying to unpack rootfs image as initramfs...
[    4.349437] systemd[1]: initrd-switch-root.service: Deactivated successfully.
[    4.349614] systemd[1]: Stopped Switch Root.
[    4.353368] systemd[1]: Stopped target Switch Root.
[    4.353423] systemd[1]: Stopped target Initrd Root File System.
[    4.421641] systemd[1]: plymouth-switch-root.service: Deactivated successfull
y.
[    4.421873] systemd[1]: Stopped Plymouth switch root service.
[    4.422194] systemd[1]: systemd-fsck-root.service: Deactivated successfully.
[    4.422236] systemd[1]: Stopped File System Check on Root Device.
[    4.473315] systemd[1]: Starting Remount Root and Kernel File Systems...
[    4.474067] systemd[1]: Repartition Root Disk was skipped because all trigger
condition checks failed.
[smorodovadv@smorodovadv ~]$ dmesg |grep -i "Root file system"
[    4.353423] systemd[1]: Stopped target Initrd Root File System.
```

Figure 4.27: Root

- Последовательность монтирования файловых систем (рис. 4.28):

```
[smorodovadv@smorodovadv ~]$ dmesg |grep -i "mounting"
[    3.301174] XFS (dm-0): Mounting V5 Filesystem
[    4.363226] systemd[1]: Mounting Huge Pages File System...
[    4.371267] systemd[1]: Mounting POSIX Message Queue File System...
[    4.376559] systemd[1]: Mounting Kernel Debug File System...
[    4.377949] systemd[1]: Mounting Kernel Trace File System...
[    5.814251] XFS (sda1): Mounting V5 Filesystem
```

Figure 4.28: Mounting

## 5 Контрольные вопросы<sup>1</sup>

1. Какую информацию содержит учётная запись пользователя? Учётная запись, как правило, содержит сведения, необходимые для опознания пользователя при подключении к системе, сведения для авторизации и учёта. Это идентификатор пользователя (login) и его пароль. Пароль или его аналог, как правило, хранится в зашифрованном или хэшированном виде для обеспечения его безопасности.
2. Укажите команды терминала и приведите примеры:

- для получения справки по команде:

Чтобы получить справку по команде, введите `man` перед ней, например, `man bash` выдаст руководство по терминалу.

- для перемещения по файловой системе:

Чтобы перемещаться по файловой системе используют команду `cd`, например `cd /user/Загрузки`.

- для просмотра содержимого каталога:

Команда `ls` позволяет просмотреть содержимое каталога.

- для определения объёма каталога:

Чтобы посмотреть объем каталога используют команду `du -s`, например `du -s /home/user/Загрузки`.

- для создания / удаления каталогов / файлов:

Чтобы создать каталог используется команда `rmdir`, а для создания

---

<sup>1</sup>Методические материалы к лабораторной работе

`mkdir`, для удаления файла `rm`, а для создания файла `touch`.

- для задания определённых прав на файл / каталог:

Для задания определенных прав на файл/каталог используют команду `chmod`.

- для просмотра истории команд:

Чтобы посмотреть историю команд используют команду `history`, например `history -c` очищает историю команд.

3. Что такое файловая система? Приведите примеры с краткой характеристикой. Файловая система – это инструмент, позволяющий операционной системе и программам обращаться к нужным файлам и работать с ними. При этом программы оперируют только названием файла, его размером и датой создания. Все остальные функции поиску необходимого файла в хранилище и работе с ним берет на себя файловая система накопителя.

Файловых систем довольно много. Но рядовым пользователям с операционной системой Windows на десктопном ПК знакомы только две.

- FAT – одна из старейших файловых систем, которая была разработана еще в 1977 году программистами компании Microsoft для гибких дисков. Современная версия FAT32 вышла в 1995 году. Она может работать с томами размером до 32 ГБ и файлами размером до 4 ГБ. При этом система не работает с накопителями объемом более 8 Тб. Поэтому сегодня FAT32 используется в основном только на флешках, картах памяти фотоаппаратов и музыкальных плееров.
- NTFS, или новая технология файловой системы была создана, чтобы устранить недостатки FAT32. Структура системы хранения данных имеет вид бинарного дерева. В отличие от иерархической, как у FAT32, доступ к информации осуществляется по запросу, а поиск ведется по названию файла. При этом система имеет каталог, отсортированный по названиям. Массив делится на 2 части и отсекается та, в которой дан-

ного файла не будет, оставшаяся часть также делиться на 2, и так далее до тех пор, пока не будет найден нужный файл. В отличие от предыдущей файловой системы, NTFS может работать с томами объемом 8 ПБ (1 петабайт – 1015 байт), и оперировать более чем 4 миллиардами файлов.

- На мобильных устройствах с ОС Android используется файловая система ext2/ext4, но только в установленном накопителе. Съемные карты памяти работают на системе FAT32.
4. Как посмотреть, какие файловые системы подмонтированы в ОС? Чтобы посмотреть какие файловые системы уже смонтированы в системе можно выполнить команду `mount` без параметров или выполнить команду `df -a`. Также можно посмотреть содержимое файла `/etc/mtab`. Команда `mount` при монтировании новой файловой системы добавляет в этот файл строку с информацией о добавляемой системе. А команда `umount` соответственно удаляет строку касающуюся отмонтированного раздела.
5. Как удалить зависший процесс? Чтобы убить зависший процесс нужно использовать команду `kill`.

## **6 Выводы**

В ходе данной лабораторной работы, мы приобрели практические навыки установки операционной системы на виртуальную машину и настройки минимально необходимых для дальнейшей работы сервисов.

## **7 Список литературы**

1. Методические материалы к лабораторной работе, представленные на сайте  
“ТУИС РУДН”
2. Rocky Linux
3. Linux
4. CentOS Linux