

Лабораторная работа № 1

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

Шулуужук Айраана Вячеславовна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Создание виртуальной машины	7
3.2	Запуск виртуальной машины и образа диска домашней гостевой ОС	9
3.3	Домашнее задание	11
4	Ответы на контрольные вопросы	14
5	Выводы	16

Список иллюстраций

3.1	Создание виртуальной машины, путь к iso-образу	7
3.2	Установка размера памяти и числа процессоров	8
3.3	Установка размера жесткого диска	8
3.4	Созданная виртуальная машина	9
3.5	Выбор языка интерфейса	9
3.6	Установка сети и имени узла	10
3.7	Установка пароля для root	10
3.8	Подключение образа диска дополнений гостевой ОС	11
3.9	Выполнение команды dmegs less	11
3.10	Выполнение команды dmegs less	12
3.11	Команда dmesg grep -i "linux version"	12
3.12	Команда dmesg grep -i "MHz"	12
3.13	Команда dmesg grep -i "CPU0"	12
3.14	Команда dmesg grep -i "memory"	13
3.15	Команда dmesg grep -i "hypervisor detected"	13
3.16	Команда df -Th	13
3.17	Команда dmesg grep -i "mount"	13

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.

2 Задание

1. Создать новую виртуальную машину
2. Установить на виртуальную машину VirtualBox операционную систему Linux (дистрибутив Rocky)
3. Установка первоначальных настроек виртуальной машины
4. Запуск виртуальной машины и образа диска домашней гостевой ОС
5. Выполнение домашнего задания

3 Выполнение лабораторной работы

3.1 Создание виртуальной машины

Создаем виртуальную машину (рис. 3.1):

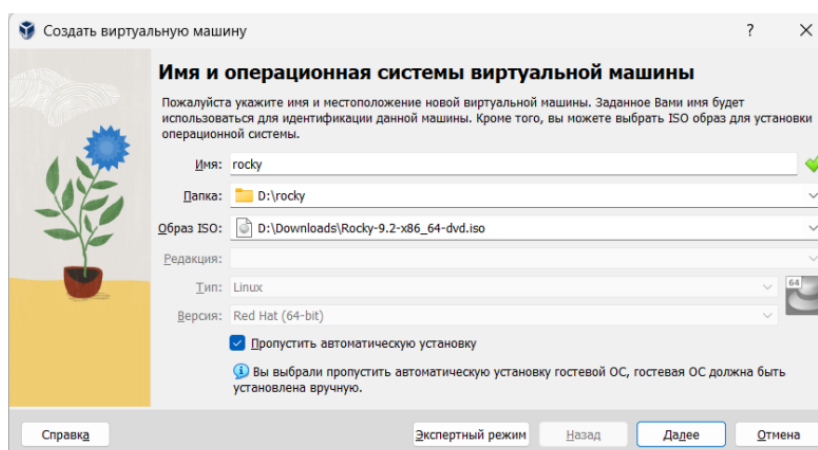


Рис. 3.1: Создание виртуальной машины, путь к iso-образу

Указываем размер основной памяти виртуальной машины – 4096 МБ и число процессоров – 2 (рис. 3.2)

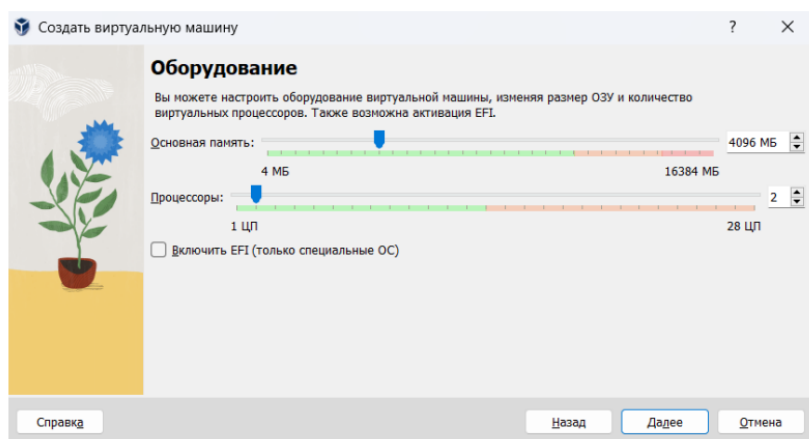


Рис. 3.2: Установка размера памяти и числа процессоров

Задаем размер виртуального жесткого диска – 30 ГБ (рис. 3.3)

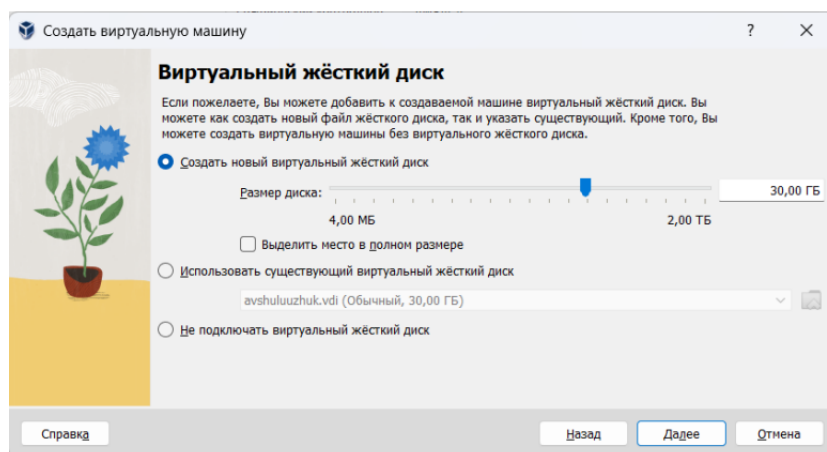


Рис. 3.3: Установка размера жесткого диска

Создали виртуальную машину и запускаем образ ОС (рис. 3.4)

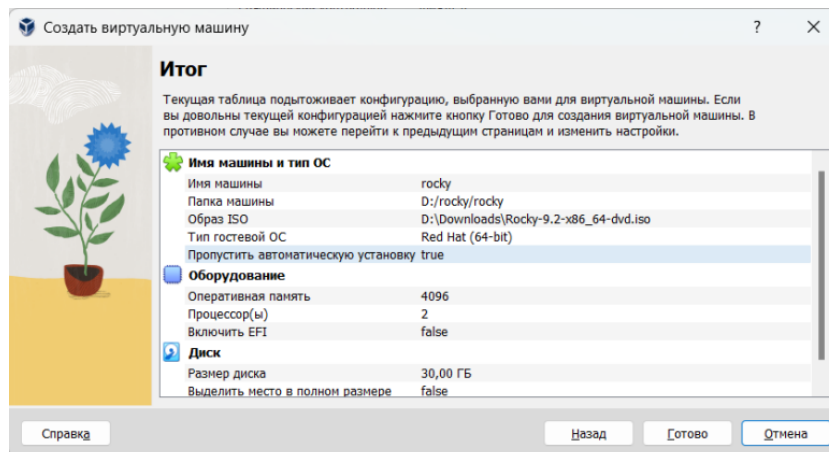


Рис. 3.4: Созданная виртуальная машина

3.2 Запуск виртуальной машины и образа диска домашней гостевой ОС

В качестве языка интерфейса выбираем английский язык (рис. 3.5)

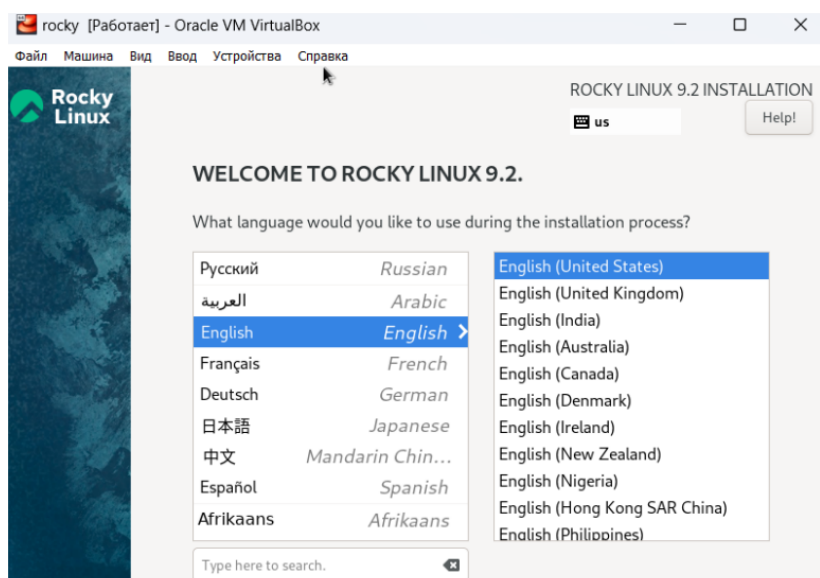


Рис. 3.5: Выбор языка интерфейса

Включим сетевое соединение и в качестве имени узла укажем user.localdomain,

где вместо user указываем имя пользователя(рис. 3.6)

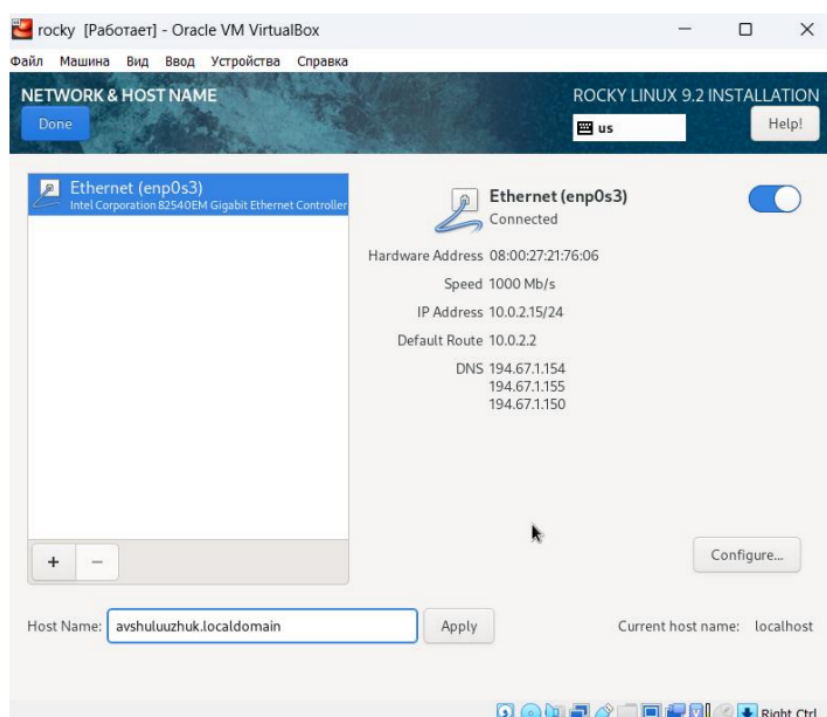


Рис. 3.6: Установка сети и имени узла

Установим пароль для root, задаем локального пользователя с правами администратора и пароль для него (рис. 3.7)

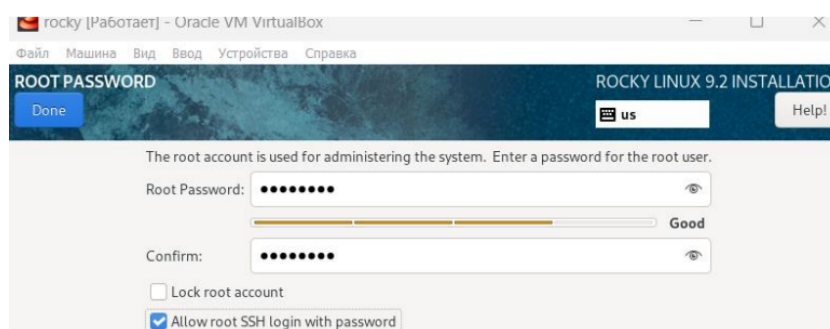


Рис. 3.7: Установка пароля для root

Начинаем установку ОС. Перезагружаем виртуальную машину. Подключаем образ диска дополнений гостевой ОС (рис. ??)

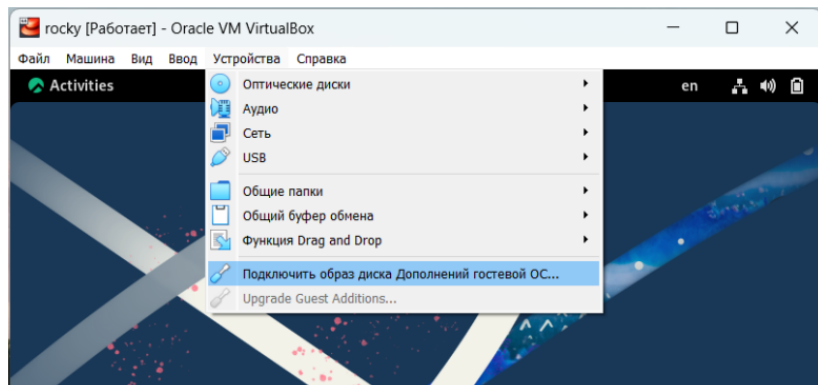


Рис. 3.8: Подключение образа диска дополнений гостевой ОС

3.3 Домашнее задание

В терминале выполняем команду `dmesg` и используем его для поиска `dmesg | less` (рис. 3.9) (рис. 3.10)

```

avshuluuzhuk@avshuluuzhuk:~ — less
0.000000] Linux version 5.14.0-284.11.1.el9_2.x86_64 (mockbuild@iad1-prod-build
001.bld.equ.rockylinux.org) (gcc (GCC) 11.3.1 20221121 (Red Hat 11.3.1-4), GNU ld ve
0.000000] The list of certified hardware and cloud instances for Enterprise Lin
0.000000] Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-5.14.0-284.11.1.el9_2.x
0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point register
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, usin
'standard' format.
0.000000] signal: max sigframe size: 1776
0.000000] BIOS-provided physical RAM map:
0.000000] BIOS-e820: [mem 0x0000000000000000-0x0000000000009fbfff] usable
0.000000] BIOS-e820: [mem 0x0000000000009fc00-0x0000000000009ffff] reserved
0.000000] BIOS-e820: [mem 0x000000000000f0000-0x000000000000ffffff] reserved
0.000000] BIOS-e820: [mem 0x00000000000100000-0x000000000000dffff] usable
0.000000] BIOS-e820: [mem 0x00000000000dffff000-0x00000000000dffffff] ACPI data
0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved

```

Рис. 3.9: Выполнение команды `dmesg | less`

```
avshuluuzhuk@avshuluuzhuk:~ — less
[ 0.000000] Linux version 5.14.0-284.11.1.el9_2.x86_64 (mockbuild@iad1-prod-build001.bld.equ.rockylinux.org) (gcc (GCC) 11.3.1 20221121 (Red Hat 11.3.1-4), GNU ld version 2.35.2-37.el9) #1 SMP PREEMPT_DYNAMIC Tue May 9 17:09:15 UTC 2023
[ 0.000000] The list of certified hardware and cloud instances for 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-5.14.0-284.11.1.el9_2.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 0x0000000000000000-0x000000000009fbff] usable
:
```

Рис. 3.10: Выполнение команды dmesg | less

Узнаем версию ядра Linux. Команда dmesg | grep -i “linux version” (рис. 3.11)

```
[avshuluuzhuk@avshuluuzhuk ~]$ dmesg | grep -i "linux version"
[ 0.000000] Linux version 5.14.0-284.11.1.el9_2.x86_64 (mockbuild@iad1-prod-build001.bld.equ.rockylinux.org) (gcc (GCC) 11.3.1 20221121 (Red Hat 11.3.1-4), GNU ld version 2.35.2-37.el9) #1 SMP PREEMPT_DYNAMIC Tue May 9 17:09:15 UTC 2023
[avshuluuzhuk@avshuluuzhuk ~]$
```

Рис. 3.11: Команда dmesg | grep -i “linux version”

Частота процессора. Команда dmesg | grep -i “MHz” (рис. 3.12)

```
[avshuluuzhuk@avshuluuzhuk ~]$ dmesg | grep -i "MHz"
[ 0.000010] tsc: Detected 2687.996 MHz processor
[ 2.472106] e1000 0000:00:03:00 eth0: (PCI:33MHz:32-bit) 08:00:27:21:76:06
[avshuluuzhuk@avshuluuzhuk ~]$
```

Рис. 3.12: Команда dmesg | grep -i “MHz”

Модель процессора. Команда dmesg | grep -i “CPU0” (рис. 3.13)

```
[avshuluuzhuk@avshuluuzhuk ~]$ dmesg | grep -i "CPU0"
[ 0.225638] smpboot: CPU0: 12th Gen Intel(R) Core(TM) i7-12700H (family: 0x6, model: 0x9a, stepping: 0x3)
[avshuluuzhuk@avshuluuzhuk ~]$
```

Рис. 3.13: Команда dmesg | grep -i “CPU0”

Объем доступной оперативной памяти – 3,8 ГБ. Команда `dmesg | grep -i “memory”` (рис. 3.14)

```
[ 0.013565] PM: hibernation: Registered nosave memory: [mem 0xffff0000-0xffffffff]
[ 0.043164] Memory: 3631796K/4193848K available (14342K kernel code, 5536K rwdata,
10180K rodata, 2792K init, 7524K bss, 243772K reserved, 0K cma-reserved)
[ 0.095844] Freeing SMP alternatives memory: 36K
```

Рис. 3.14: Команда `dmesg | grep -i “memory”`

Тип обнаруженного гипервизора - KVM. Команда `dmesg | grep -i “hypervisor detected”` (рис. 3.15)

```
[avshuluuzhuk@avshuluuzhuk ~]$ dmesg | grep -i "hypervisor detected"
[ 0.000000] Hypervisor detected: KVM
[avshuluuzhuk@avshuluuzhuk ~]$
```

Рис. 3.15: Команда `dmesg | grep -i “hypervisor detected”`

Тип файловой системы корневого раздела – xfs. Используем команду `df -Th` (рис. 3.17)

```
[avshuluuzhuk@avshuluuzhuk ~]$ df -Th
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  4.0M   0    4.0M  0%  /dev
tmpfs           tmpfs     2.0G   0    2.0G  0%  /dev/shm
tmpfs           tmpfs     784M  12M   773M  2%  /run
/dev/mapper/rl-root xfs       26G   5.2G   21G  20%  /
```

Рис. 3.16: Команда `df -Th`

Последовательность монтирования файловых систем. Для этого используем команду `dmesg | grep -i “mount”` (рис. 3.17)

```
[avshuluuzhuk@avshuluuzhuk ~]$ df -Th
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  4.0M   0    4.0M  0%  /dev
tmpfs           tmpfs     2.0G   0    2.0G  0%  /dev/shm
tmpfs           tmpfs     784M  12M   773M  2%  /run
/dev/mapper/rl-root xfs       26G   5.2G   21G  20%  /
```

Рис. 3.17: Команда `dmesg | grep -i “mount”`

4 Ответы на контрольные вопросы

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 учета. Это идентификатор пользователя (login) и его пароль. Пароль или его аналог, как правило, хранится в зашифрованном или хешированном виде для обеспечения его безопасности

2. Укажите команды терминала:

– для получения справки по команде – команда `help` – для перемещения по файловой системе – команда `cd` – для просмотра содержимого каталога – команда `ls` – для создания каталога – команда `mkdir` – для удаления каталогов – команда `rmdir` – для создания файлов – команда `touch` – для удаления файлов – команда `rm` – для задания определённых прав на файл / каталог – команда `chmod` – для просмотра истории команд – команда `history`

3. Что такое файловая система?

Файловая система Linux обычно представляет собой встроенный уровень операционной системы Linux, используемый для управления данными хранилища. Он контролирует, как данные хранятся и извлекаются. Он управляет именем файла, размером файла, датой создания и другой информацией о файле.

4. Как посмотреть, какие файловые системы смонтированы в ОС?

Команда `findmnt` — это простая утилита командной строки, используемая для отображения списка смонтированных файловых систем или поиска файловой системы в `/etc/fstab`, `/etc/mtab` и `/proc/self/mountinfo`

5. Как удалить зависший процесс?

Использование команды `xkill` в терминале. Это инструмент принудительного уничтожения процесса, который предустановлен в Ubuntu, но его можно также установить через терминал в других дистрибутивах. Команда `kill` может быть использована только с указанием идентификатора процесса.

5 Выводы

В ходе выполнения лабораторной работы были приобретены практические навыки установки операционной системы на виртуальную машину и настройка минимально необходимых для дальнейшей работы сервисов. Было выполнено дополнительное задание, где в процессе мы узнавали требуемую информацию