

Лабораторная работа №1

Основы администрирования операционных систем

Верниковская Е. А., НПИбд-01-23

5 сентября 2024

Российский университет дружбы народов, Москва, Россия

Вводная часть

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

1. Скачать Rocky Linux.
2. Создать виртуальную машину.
3. Настроить виртуальную машину.

Выполнение лабораторной работы

Скачиваем Rocky Linux (рис. 1), (рис. 2)

Создание виртуальной машины

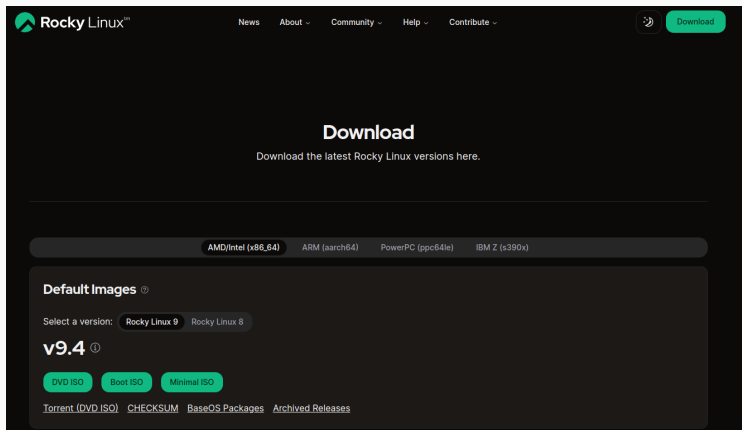


Рис. 1: Сайта Rocky Linux

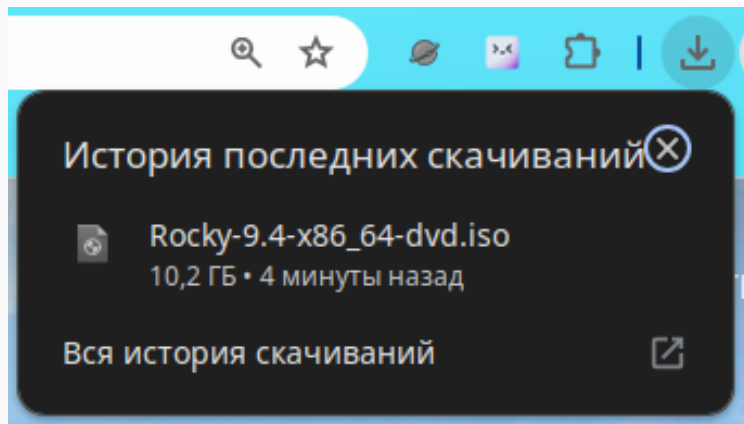


Рис. 2: Установка Rocky Linux с сайта

Открываем VirtualBox и создаём новую виртуальную машину

Указываем имя виртуальной машины, определяем тип операционной системы и указываем путь к iso-образу (рис. 3)

Создание виртуальной машины

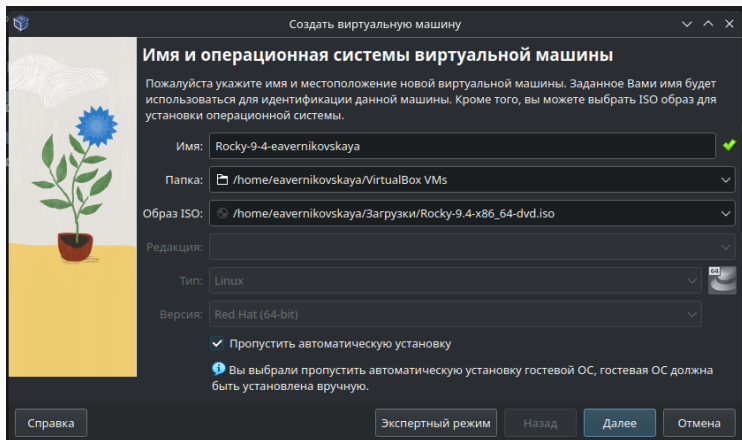


Рис. 3: Создание виртуальной машины (1)

Далее указываем размер оперативной памяти виртуальной машины - 2048 МБ
и число процессоров - 2 (рис. 4)

Создание виртуальной машины

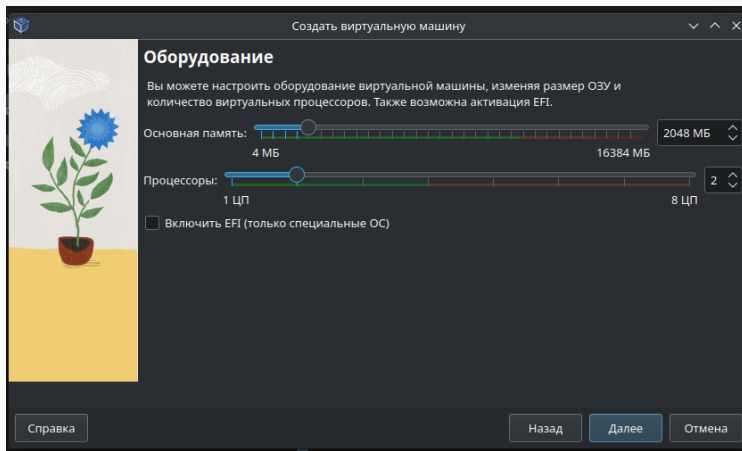


Рис. 4: Создание виртуальной машины (2)

Задаём размер виртуального жёсткого диска - 40 ГБ (рис. 5)

Создание виртуальной машины

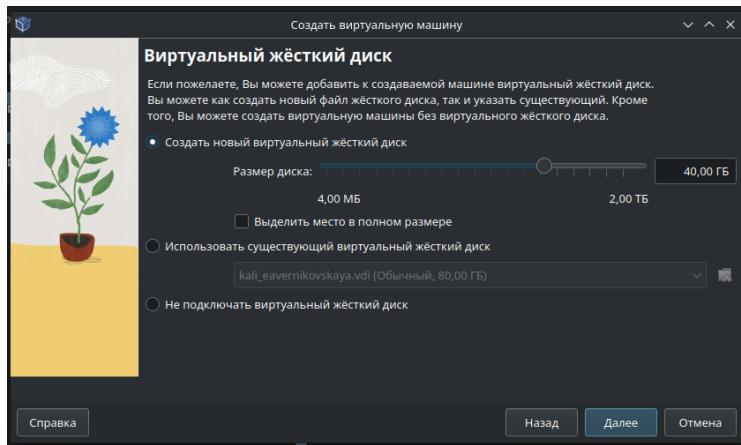


Рис. 5: Создание виртуальной машины (3)

Далее запускаем виртуальную машину (рис. 6)

Создание виртуальной машины

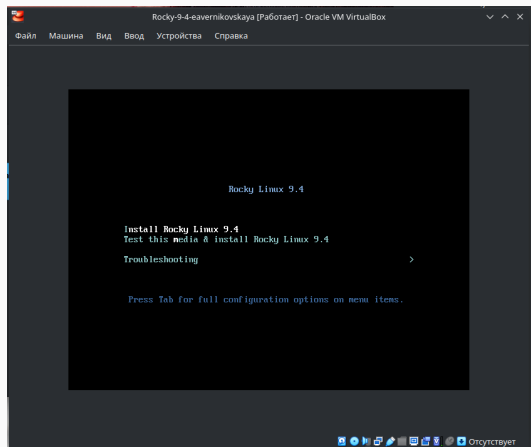


Рис. 6: Запуск виртуальной машины

После запуска устанавливаем английский язык интерфейса (рис. 7)

Установка операционной системы

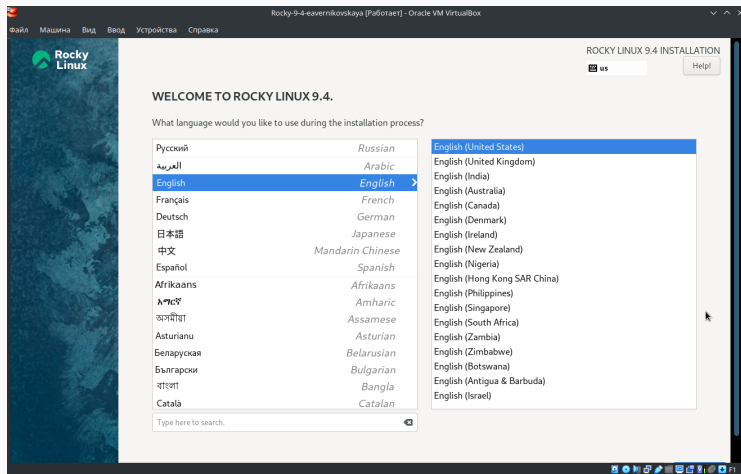


Рис. 7: Запуск виртуальной машины

Далее отключаем KDUMP, а место установки ОС оставляем без изменения (рис. 8), (рис. 9), (рис. 10)

Установка операционной системы

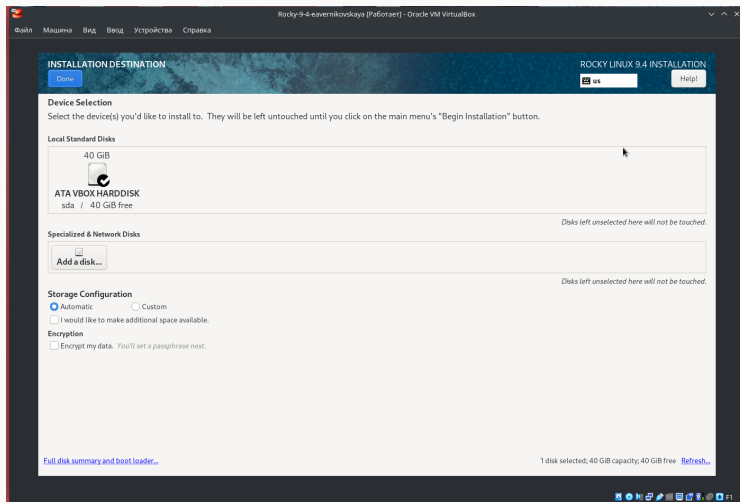


Рис. 8: Место установки ОС

Установка операционной системы

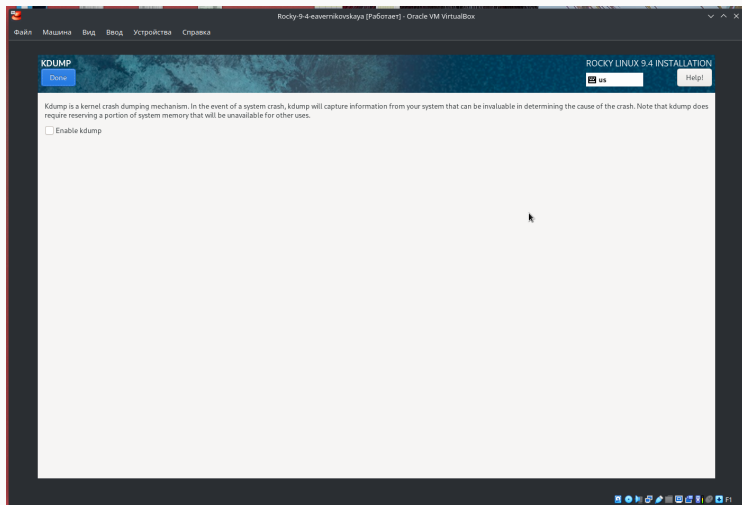


Рис. 9: Отключение KDUMP

Установка операционной системы

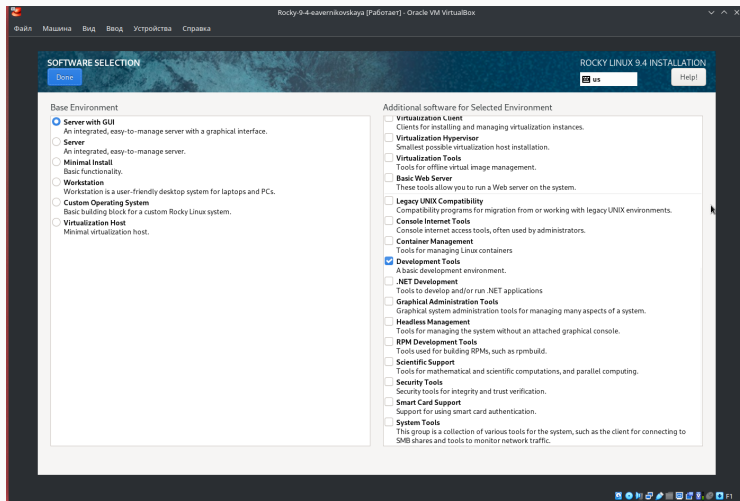


Рис. 10: Настройка ВМ

Включаем сетевое соединение и в качестве имени узла указываем `user.localdomain`, где вместо `user` имя нашего пользователя в соответствии с соглашением об именовании (рис. 11)

Установка операционной системы

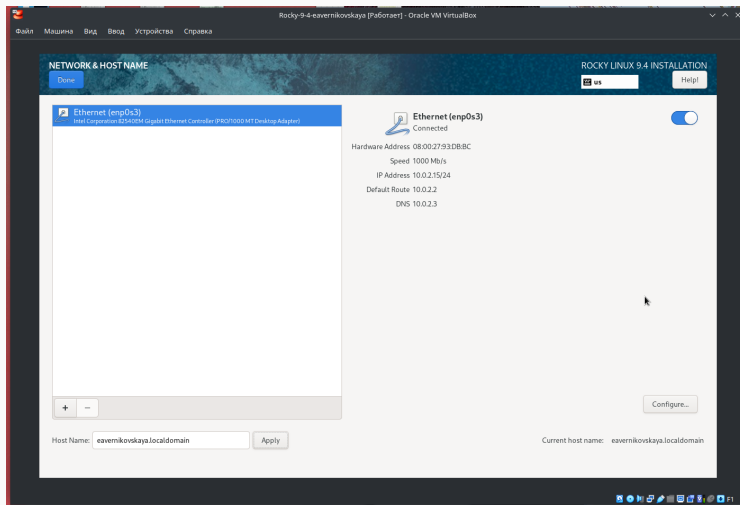


Рис. 11: Сетевое соединение

Устанавливаем пароль для root, разрешение на ввод пароля для root при использовании SSH (рис. 12)

Установка операционной системы

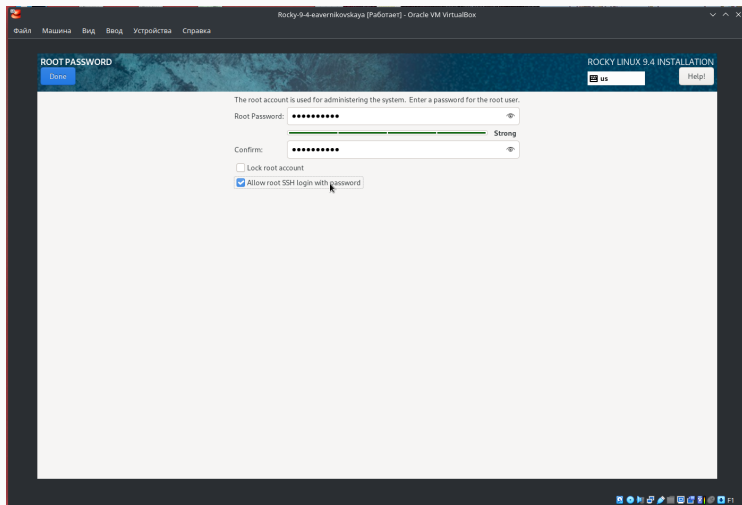


Рис. 12: Пароль для root

Затем задаём локального пользователя с правами администратора и пароль для него (рис. 13)

Установка операционной системы

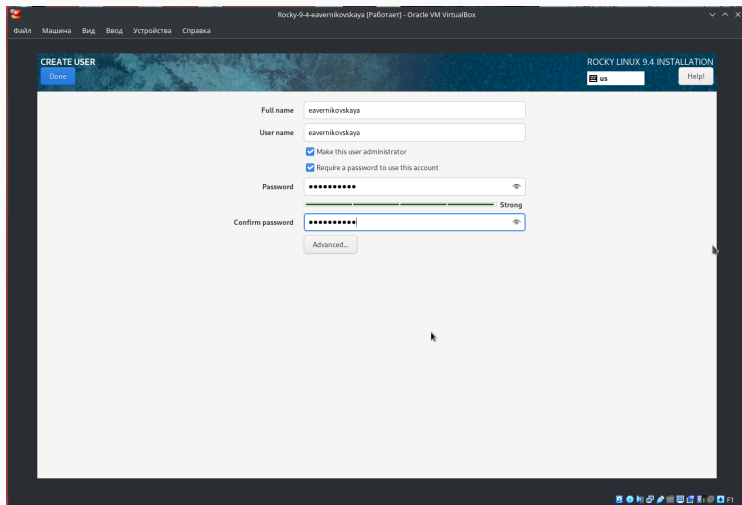


Рис. 13: Пароль для root

Начинаем установку операционной системы (рис. 14), (рис. 15)

Установка операционной системы

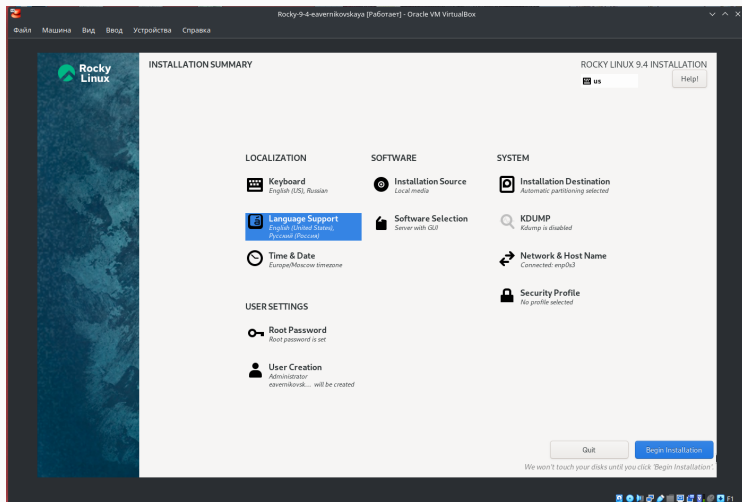


Рис. 14: Выставленные настройки

Установка операционной системы

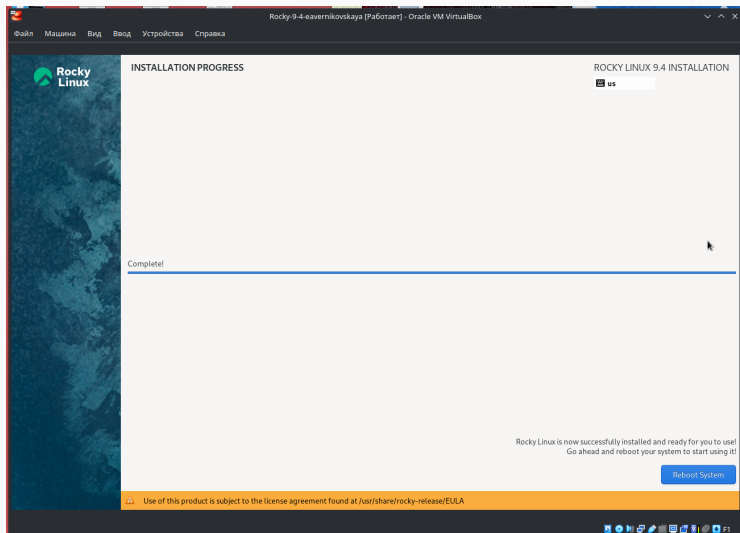


Рис. 15: Установка ОС

После установки ОС и перезапуска ВМ входим в ОС под заданной нами при установке учётной записью (рис. 16), (рис. 17)

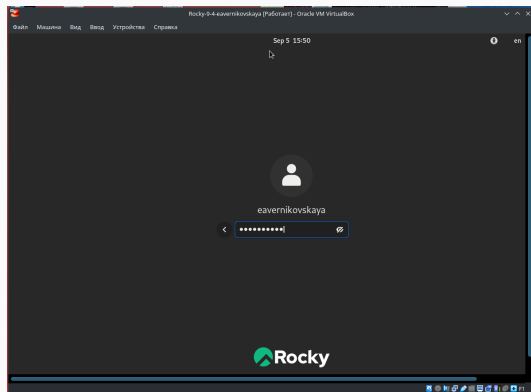


Рис. 16: Вход в учётную запись

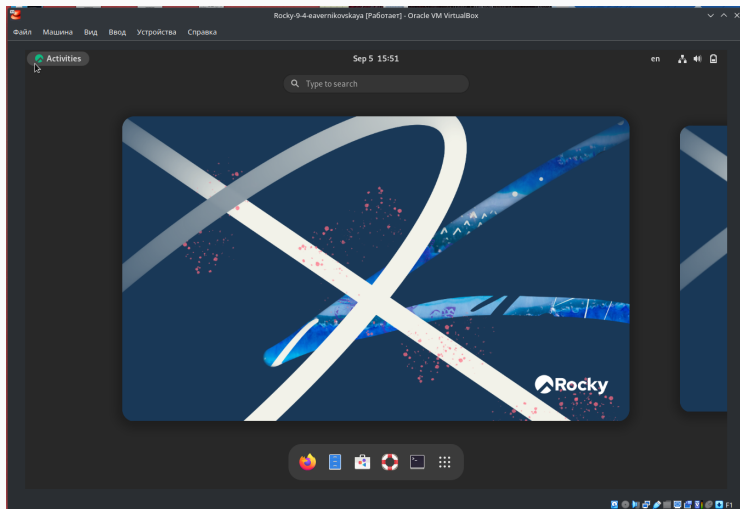
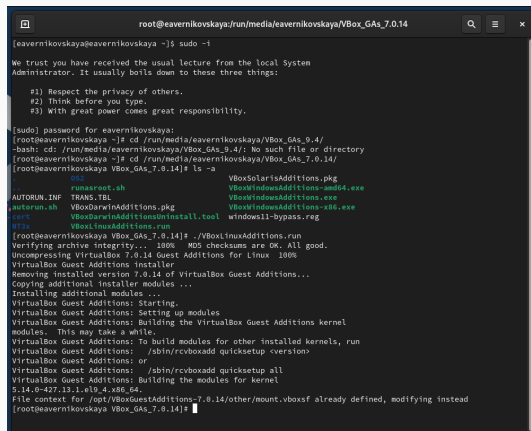


Рис. 17: Rocky Linux

Далее через терминал подключаем образ диска дополнений гостевой ОС: (рис. 18)

- заходим в пользователя root, с помощью *sudo -i*
- переходим в каталог `/run/media/имя_пользователя/VBox_GAs_версия/`
- запускаем `VBoxLinuxAdditions.run`



```
root@eavernikovskaya:/run/media/eavernikovskaya/VBox_GAs_7.0.14

[eavernikovskaya@eavernikovskaya ~]$ sudo -i

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

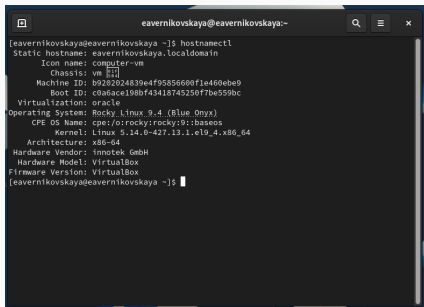
[sudo] password for eavernikovskaya:
[root@eavernikovskaya ~]# cd /run/media/eavernikovskaya/VBox_GAs_9.4/
-bash: cd: /run/media/eavernikovskaya/VBox_GAs_9.4/: No such file or directory
[root@eavernikovskaya ~]# cd /run/media/eavernikovskaya/VBox_GAs_7.0.14/
[root@eavernikovskaya VBox_GAs_7.0.14]# ls -la
total 108
drwxr-xr-x 2 root root 4096 Aug 14 12:02
-rw-r--r-- 1 root root  112 Aug 14 12:02 .DS_Store
-rw-r--r-- 1 root root  112 Aug 14 12:02 .runasroot.sh
-rw-r--r-- 1 root root  112 Aug 14 12:02 AUTORUN.INF
-rw-r--r-- 1 root root  112 Aug 14 12:02 TRANS.TBL
-rw-r--r-- 1 root root  112 Aug 14 12:02 autorun.sh
-rw-r--r-- 1 root root  112 Aug 14 12:02 VBoxDarwinAdditions.pkg
-rw-r--r-- 1 root root  112 Aug 14 12:02 VBoxDarwinAdditionsUninstall.tool
-rw-r--r-- 1 root root  112 Aug 14 12:02 VBoxLinuxAdditions.run
-rw-r--r-- 1 root root  112 Aug 14 12:02 VBoxSolarisAdditions.pkg
-rw-r--r-- 1 root root  112 Aug 14 12:02 VBoxWindowsAdditions-amd64.exe
-rw-r--r-- 1 root root  112 Aug 14 12:02 VBoxWindowsAdditions.exe
-rw-r--r-- 1 root root  112 Aug 14 12:02 VBoxWindowsAdditions-x86.exe
-rw-r--r-- 1 root root  112 Aug 14 12:02 windows11-bypass.reg

[root@eavernikovskaya VBox_GAs_7.0.14]# ./VBoxLinuxAdditions.run
Verifying archive integrity... 100% MD5 checksums are OK. All good.
Uncompressing VirtualBox 7.0.14 Guest Additions for Linux 100%
VirtualBox Guest Additions installer
Removing installed version 7.0.14 of VirtualBox Guest Additions...
Copying additional installer modules ...
Installing additional modules ...
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
5.14.0-427.13.1.el9_4.x86_64.
File context for /opt/VBoxGuestAdditions-7.0.14/other/mount.vboxsf already defined, modifying instead
[root@eavernikovskaya VBox_GAs_7.0.14]#
```

Рис. 18: Подключение образ диска дополнений гостевой ОС

Установка имени пользователя и названия хоста

При установке виртуальной машины мы задали имя пользователя и имя хоста, удовлетворяющее соглашению об именовании. Поэтому нам не надо ничего исправлять. Я просто посмотрю имя хоста с помощью *hostnamectl* (рис. 19)

A terminal window titled 'eavernikovskaya@eavernikovskaya:~' showing the output of the 'hostnamectl' command. The output lists system information including static hostname, icon name, chassis, machine ID, boot ID, virtualization type, operating system, CPE OS name, kernel version, architecture, hardware vendor, hardware model, and firmware version.

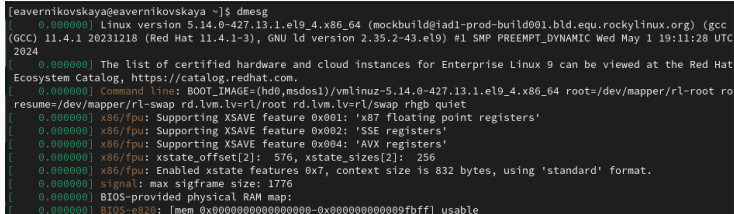
```
eavernikovskaya@eavernikovskaya:~$ hostnamectl
Static hostname: eavernikovskaya.localdomain
Icon name: computer-vm
Chassis: vm
Machine ID: b9282024839e4f95856600f1e460ebe9
Boot ID: c9a6ace198bf43418745250f7be559bc
Virtualization: oracle
Operating System: Rocky Linux 9.4 (Blue Onyx)
CPE OS Name: cpe:/o:rocky:rocky:9:baseos
Kernel: Linux 5.14.0-427.13.1.el9_4.x86_64
Architecture: x86_64
Hardware Vendor: innotek GmbH
Hardware Model: VirtualBox
Firmware Version: VirtualBox
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 19: Имя хоста

Домашнее задание

Выполнение домашнего задания

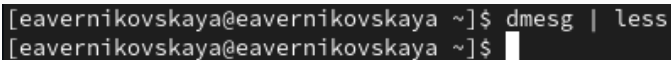
В окне терминала проанализируем последовательность загрузки системы, выполнив команду *dmesg* (рис. 20)



```
[eavernikovskaya@eavernikovskaya ~]$ dmesg
[ 0.000000] Linux version 5.14.0-427.13.1.el9_4.x86_64 (mockbuild@iad1-prod-build001.bld.equ.rockylinux.org) (gcc
(GCC) 11.4.1 20231218 (Red Hat 11.4.1-3), GNU ld version 2.35.2-43.el9) #1 SMP PREEMPT_DYNAMIC Wed May 1 19:11:28 UTC
2024
[ 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-5.14.0-427.13.1.el9_4.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. 20: Команда *dmesg*

Далее посмотрим вывод этой команды с помощью *dmesg | less* (рис. 21), (рис. 22)

A terminal window with a dark background. The prompt is [eavernikovskaya@eavernikovskaya ~]\$. The command dmesg | less has been entered. A white cursor is visible at the end of the second line of the prompt.

```
[eavernikovskaya@eavernikovskaya ~]$ dmesg | less  
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 21: Команда *dmesg | less* (1)

Выполнение домашнего задания

```
eavernikovskaya@eavernikovskaya:~$ less
[ 0.000000] Linux version 5.14.0-427.13.1.el9_4.x86_64 (mockbuild@adi-prod-build001.bld.equ.rockylinux.org) (gcc
(GCC) 11.4.1 20231218 (Red Hat 11.4.1-3), GNU ld version 2.35.2-43.el9) #1 SMP PREEMPT_DYNAMIC Wed May 1 10:11:28 UTC
2024
[ 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-5.14.0-427.13.1.el9_4.x86_64 root=/dev/mapper/rl-root ro
resumes=/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 0x0000000000000000-0x0000000000000bfff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000000fc00-0x0000000000000ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000000f0000-0x000000000000ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000001000000-0x0000000007ffff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000007fff0000-0x0000000007ffffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000ffff0000-0x00000000ffffffff] reserved
[ 0.000000] NX (Execute Disable) protection: active
[ 0.000000] SMBIOS 2.5 present.
[ 0.000000] DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[ 0.000000] Hypervisor detected: KVM
[ 0.000000] kvm-clock: Using msrs 4b564d01 and 4b564d00
[ 0.000000] kvm-clock: using sched offset of 6585529695 cycles
[ 0.000000] clocksource: kvm-clock: mask: 0xfffffffffff max_cycles: 0x1cd42e4dffb, max_idle_ns: 881598591483
ns
[ 0.000012] tsc: Detected 1991.997 MHz processor
[ 0.001748] e820: update [mem 0x00000000-0x00000fff] usable => reserved
[ 0.001754] e820: remove [mem 0x00000000-0x00000fff] usable
[ 0.001762] last_pfn = 0xffff max_arch_pfn = 0x400000000
[ 0.001772] MTRRs disabled by BIOS
[ 0.001776] x86/PAT: Configuration [0-7]: WB WC UC- UC WB WP UC- WT
[ 0.001864] found SMP MP-table at [mem 0x00099fff-0x0009ffff]
[ 0.002264] RAMDISK: [mem 0x313fe000-0x349f0fff]
[
:
```

Рис. 22: Команда dmesg | less (2)

Далее получаем следующую информацию:

1. Версия ядра Linux (Linux version) (рис. 23)
2. Частота процессора (Detected Mhz processor) (рис. 24)
3. Модель процессора (CPU0) (рис. 25)
4. Объем доступной оперативной памяти (Memory available) (рис. 26)
5. Тип обнаруженного гипервизора (Hypervisor detected) (рис. 27)
6. Тип файловой системы корневого раздела (рис. 28)
7. Последовательность монтирования файловых систем (рис. 29]

Выполнение домашнего задания

```
[eavernikovskaya@eavernikovskaya ~]$ dmesg | grep "Linux version"
[    0.000000] Linux version 5.14.0-427.13.1.el9_4.x86_64 (mockbuild@iad1-prod-build001.bld.equ.rockylinux.org) (gcc
(GCC) 11.4.1 20231218 (Red Hat 11.4.1-3), GNU ld version 2.35.2-43.el9) #1 SMP PREEMPT_DYNAMIC Wed May 1 19:11:28 UTC
2024
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 23: Версия ядра Linux

```
[eavernikovskaya@eavernikovskaya ~]$ dmesg | grep -i "Mhz"  
[    0.000012] tsc: Detected 1991.997 MHz processor  
[    3.931684] e1000 0000:00:03:0 eth0: (PCI:33MHz:32-bit) 08:00:27:93:db:bc  
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 24: Частота процессора

```
[eavernikovskaya@eavernikovskaya ~]$ dmesg | grep "CPU0"  
[    0.204654] smpboot: CPU0: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz (family: 0x6, model: 0x8e, stepping: 0xa)  
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 25: Модель процессора

```
[eavernikovskaya@eavernikovskaya ~]$ free -m
```

	total	used	free	shared	buff/cache	available
Mem:	1967	1060	450	19	623	906
Swap:	2095	0	2095			

```
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 26: Объем доступной оперативной памяти

```
[eavernikovskaya@eavernikovskaya ~]$ dmesg | grep "Hypervisor detected"
[    0.000000] Hypervisor detected: KVM
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 27: Тип обнаруженного гипервизора

Выполнение домашнего задания

```
[eavernikovskaya@eavernikovskaya ~]$ dmesg | grep -i "filesystem"
[ 5.826385] XFS (dm-0): Mounting V5 Filesystem 760e43c0-13b0-41b3-9e49-ad0b46fa0d30
[ 9.506433] XFS (sda1): Mounting V5 Filesystem 9a39db74-1505-4a34-92d7-aba95e8d7183
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 28: Тип файловой системы корневого раздела

Выполнение домашнего задания

```
[eavernikovskaya@eavernikovskaya ~]$ dmesg | grep -i "mount"
[ 0.086182] Mount-cache hash table entries: 4096 (order: 3, 32768 bytes, linear)
[ 0.086193] Mountpoint-cache hash table entries: 4096 (order: 3, 32768 bytes, linear)
[ 5.826385] XFS (dm-0): Mounting V5 Filesystem 760e43c0-13b0-41b3-9e49-ad0b46fa0d30
[ 5.842245] XFS (dm-0): Ending clean mount
[ 7.400515] systemd[1]: Set up automount Arbitrary Executable File Formats File System Automount Point.
[ 7.442881] systemd[1]: Mounting Huge Pages File System...
[ 7.446055] systemd[1]: Mounting POSIX Message Queue File System...
[ 7.449874] systemd[1]: Mounting Kernel Debug File System...
[ 7.453981] systemd[1]: Mounting Kernel Trace File System...
[ 7.534946] systemd[1]: Starting Remount Root and Kernel File Systems...
[ 9.506433] XFS (sda1): Mounting V5 Filesystem 9a39db74-1505-4a34-92d7-aba95e8d7183
[ 9.888368] XFS (sda1): Ending clean mount
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 29: Последовательность монтирования файловых систем

Подведение итогов

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

1. Лабораторная работа №1 [Электронный ресурс] URL:
https://esystem.rudn.ru/pluginfile.php/2400671/mod_resource/content/7/002-os_install-Rocky9.pdf
2. VirtualBox [Электронный ресурс] URL:
https://www.virtualbox.org/wiki/Linux_Downloads
3. Rocky Linux [Электронный ресурс] URL: <https://rockylinux.org/ru/download>