

# **Отчёт по лабораторной работе №1**

## **Информационная безопасность**

**Настройка рабочего пространства и конфигурация операционной системы на виртуальную машину.**

Выполнила: Коняева Марина Александровна,  
НФИбд-01-21, 1032217044

# Содержание

<b>Цель работы</b>	<b>4</b>
<b>Теоретическое введение</b>	<b>5</b>
<b>Выполнение лабораторной работы</b>	<b>8</b>
Установка и конфигурация операционной системы на виртуальную машину .	8
Virtual Box . . . . .	8
Переход в ОС Linux . . . . .	13
Домашнее задание . . . . .	14
<b>Вывод</b>	<b>17</b>
<b>Список литературы. Библиография</b>	<b>18</b>

## Список иллюстраций

1	(рис. 1. Общие настройки) . . . . .	8
2	(рис. 2. Имя и путь ОС) . . . . .	9
3	(рис. 3. Виртуальный жесткий диск) . . . . .	9
4	(рис. 4. Итог настроек) . . . . .	9
5	(рис. 5. Носители) . . . . .	10
6	(рис. 6. Стартовое меню установки) . . . . .	10
7	(рис. 7. Server with GUI) . . . . .	10
8	(рис. 8. Enable KDUMP) . . . . .	11
9	(рис. 9. Root password) . . . . .	11
10	(рис. 10. Соглашение об именовании) . . . . .	11
11	(рис. 11. Итоговое меню установки) . . . . .	12
12	(рис. 12. Установка) . . . . .	12
13	(рис. 13. успешное создание пользователя) . . . . .	13
14	(рис. 14. Подключение гостевых настроек) . . . . .	13
15	(рис. 15. dmesg) . . . . .	14
16	(рис. 16. dmesg   less, версия ядра линукс, частота процессора, модель процессора) . . . . .	14
17	(рис. 17. Версия ядра Linux (Linux version) . . . . .	15
18	(рис. 18. Частота процессора (Detected Mhz processor) . . . . .	15
19	(рис. 19. Модель процессора (CPU0) . . . . .	15
20	(рис. 20. Объем доступной оперативной памяти) . . . . .	15
21	(рис. 21. Тип обнаруженного гипервизора) . . . . .	15
22	(рис. 22. Тип файловой системы корневого раздела) . . . . .	16
23	(рис. 23. Последовательность монтирования файловых систем) . . . . .	16

## **Цель работы**

Настроить рабочее пространство для лабораторных работ, приобрести практические навыки установки операционной системы на виртуальную машину и настройки минимально необходимых для дальнейшей работы сервисов.

# Теоретическое введение

**Oracle VM VirtualBox** — это мощная и бесплатная виртуализационная платформа, разработанная корпорацией Oracle, которая позволяет пользователям создавать и управлять виртуальными машинами на своих компьютерах. [1]

**Системы контроля версий (Version Control System, VCS)** применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется. [2]

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматиче-

ски или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.

Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

Примеры команд для Git:

Назва-	
ние	
команды	Описание команды
<code>git clone</code>	Клонирование репозитория на ПК
<code>git commit -m "Initial Commit"</code>	Оставление коммита
<code>git push</code>	Загрузка изменений на гит

Назва-	
ние	
команды	Описание команды
make	Конвертация файла .md

**Markdown** - это легковесный язык разметки, который широко используется для создания форматированного текста в веб-среде. Его простота и читаемость делают его популярным среди разработчиков, писателей и блогеров. Синтаксис Markdown состоит из простых символов и правил форматирования, которые позволяют создавать заголовки, списки, ссылки, изображения и другие элементы веб-страниц без необходимости использовать сложные HTML-теги. Он также легко читается в исходном виде и может быть конвертирован в различные форматы, такие как HTML, PDF или документы Microsoft Word, делая Markdown удобным инструментом для создания содержательного и красочного контента в интернете. [3]

# Выполнение лабораторной работы

## Установка и конфигурация операционной системы на виртуальную машину

### Virtual Box

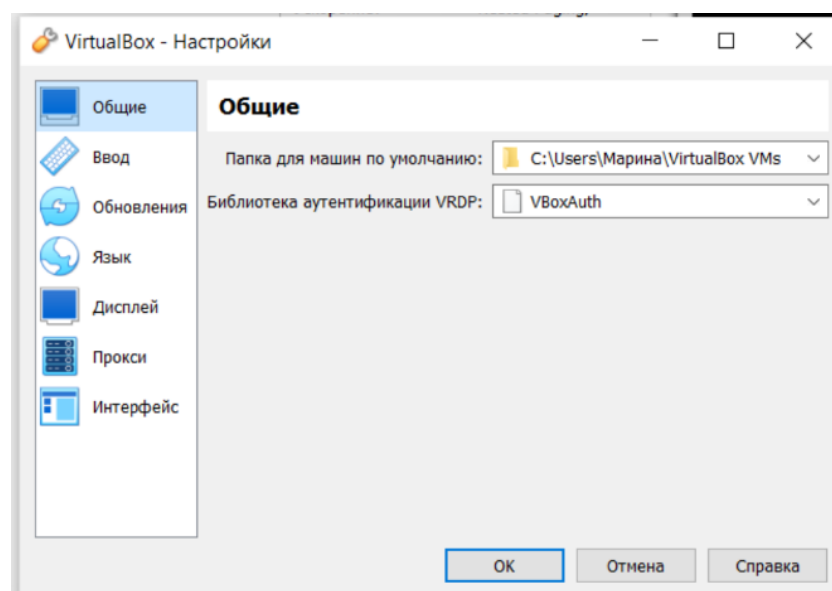


Рис. 1: (рис. 1. Общие настройки)




## Укажите имя и тип ОС

Пожалуйста укажите имя и местоположение новой виртуальной машины и выберите тип операционной системы, которую Вы собираетесь установить на данную машину. Заданное Вами имя будет использоваться для идентификации данной машины.

Имя:

Папка машины:

Тип:  

Версия:

Рис. 2: (рис. 2. Имя и путь ОС)

## Жесткий диск

При желании к новой виртуальной машине можно подключить виртуальный жёсткий диск. Вы можете создать новый или выбрать из уже имеющихся.

Если Вам необходима более сложная конфигурация Вы можете пропустить этот шаг и внести изменения в настройки машины после её создания.

Рекомендуемый объём нового виртуального жёсткого диска равен **8,00 Гб**.

- ☐ Не подключать виртуальный жёсткий диск
- ☒ Создать новый виртуальный жёсткий диск
- ☐ Использовать существующий виртуальный жёсткий диск

Рис. 3: (рис. 3. Виртуальный жесткий диск)

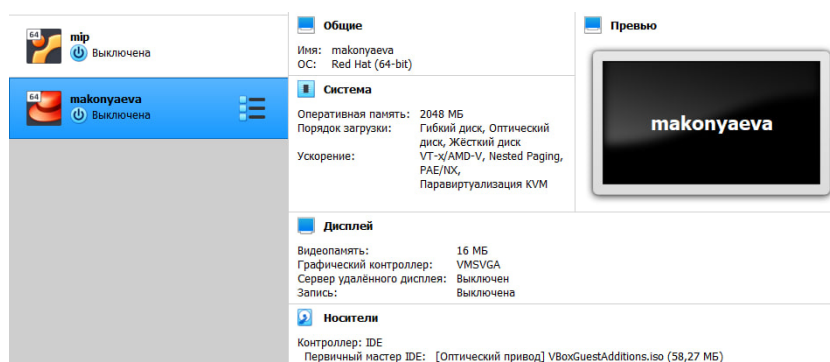


Рис. 4: (рис. 4. Итог настроек)

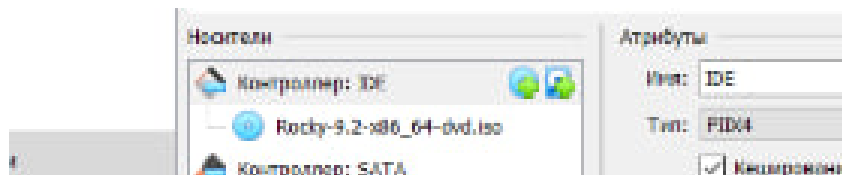


Рис. 5: (рис. 5. Носители)

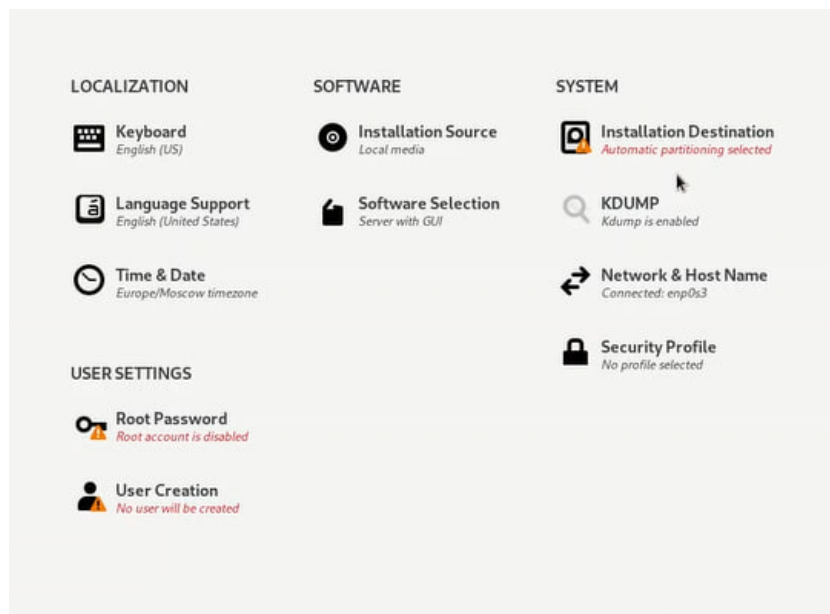


Рис. 6: (рис. 6. Стартовое меню установки)

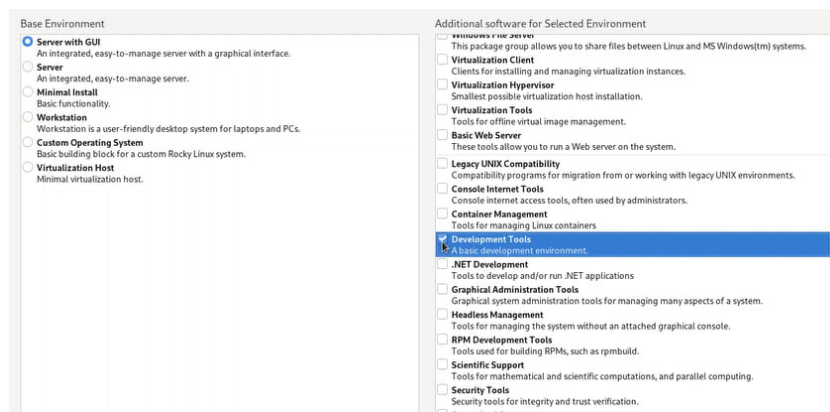


Рис. 7: (рис. 7. Server with GUI)

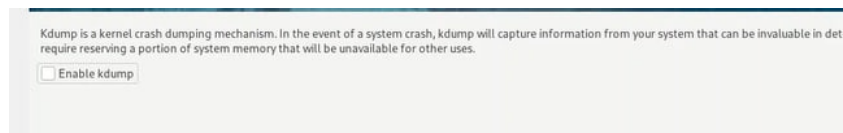


Рис. 8: (рис. 8. Enable KDUMP)

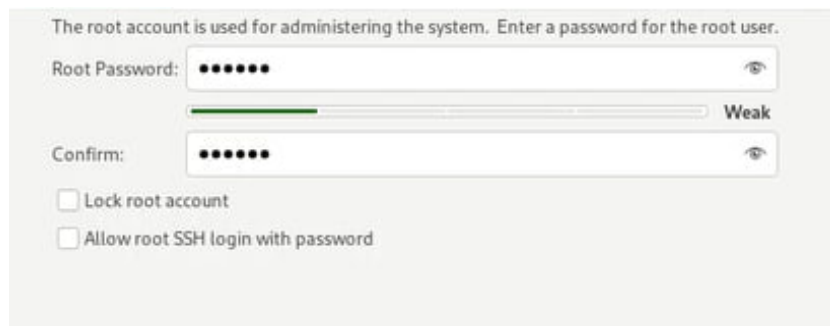


Рис. 9: (рис. 9. Root password)

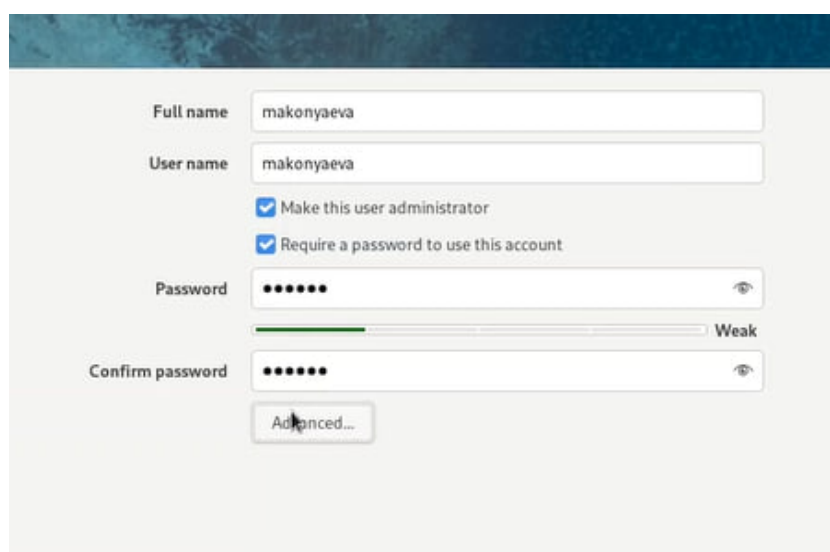


Рис. 10: (рис. 10. Соглашение об именовании)

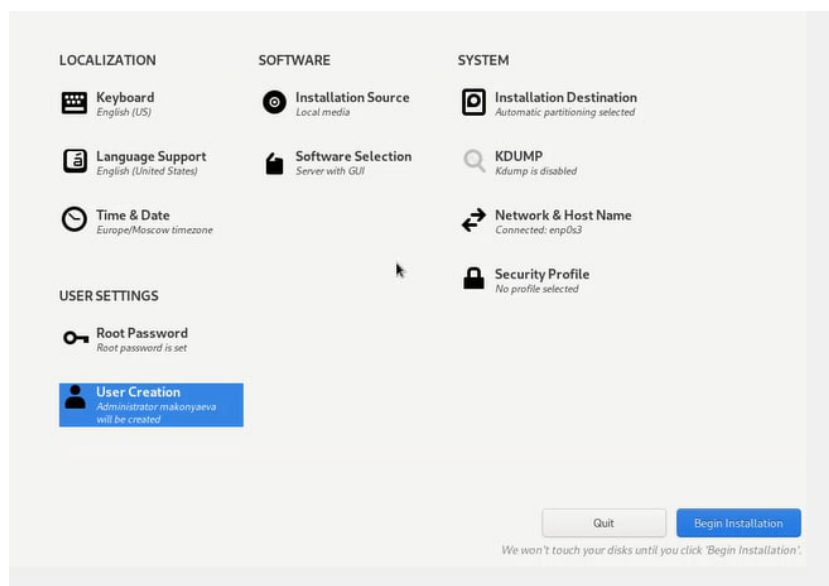


Рис. 11: (рис. 11. Итоговое меню установки)

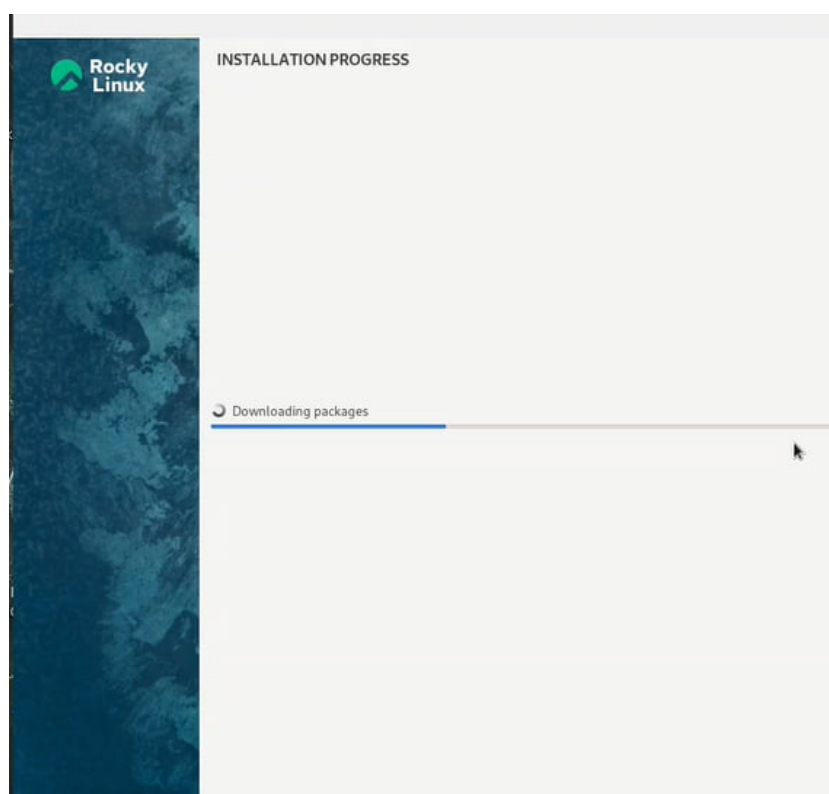


Рис. 12: (рис. 12. Установка)

## Переход в ОС Linux

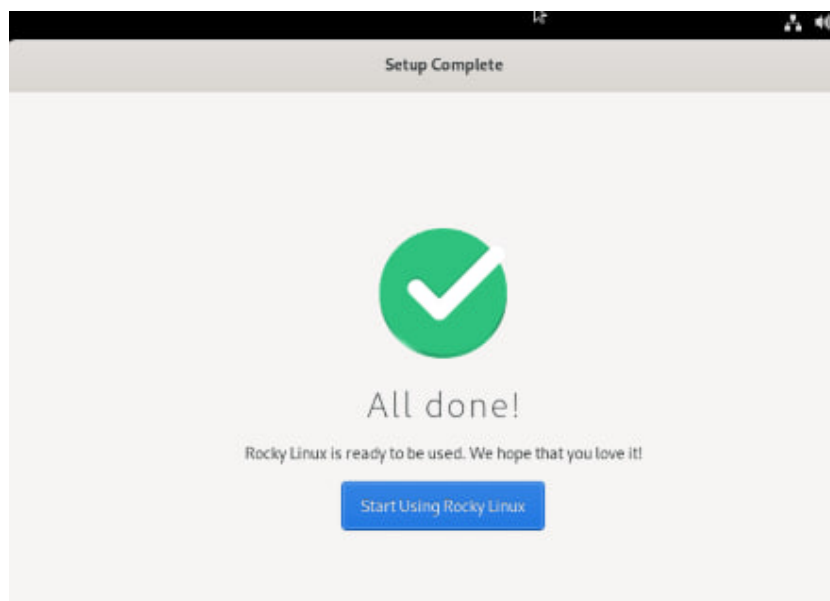


Рис. 13: (рис. 13. успешное создание пользователя)

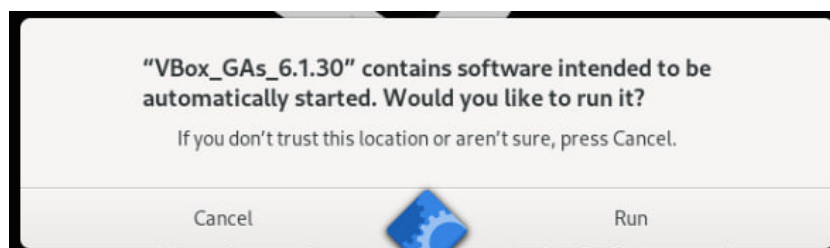


Рис. 14: (рис. 14. Подключение гостевых настроек)

## Домашнее задание

```
0.000000] Linux version 5.14.0-284.11.1.el9_2.x86_64 (mockbuild@iadi1-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 Enterpri
e 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.e
9_2.x86_64 root=/dev/mapper/rl_mvmlashenko-root ro resume=/dev/mapper/rl_mvml
ashenko-swap rd.lvm.lv=rl_mvmlashenko/root rd.lvm.lv=rl_mvmlashenko/swap r
ghb quiet
0.000000] [Firmware Bug]: TSC doesn't count with P0 frequency!
0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point re
gisters'
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 0x0000000000000000-0x0000000000009fbff] usable
0.000000] BIOS-e820: [mem 0x0000000000009fc00-0x0000000000009ffff] reserved
0.000000] BIOS-e820: [mem 0x000000000000f0000-0x000000000000ffffff] reserved
0.000000] BIOS-e820: [mem 0x00000000001000000-0x0000000000dffff] usable
0.000000] BIOS-e820: [mem 0x0000000000dffff0000-0x0000000000dffff] ACPI dat
```

Рис. 15: (рис. 15. dmesg)

```
[ 0.000000] Linux version 5.14.0-427.13.1.el9_4.x86_64 (mockbuild@iadi1-prod-b
uild001.bld.equ.rockylinux.org) (gcc (GCC) 11.4.1 20231218 (Red Hat 11.4.1-3), G
NU 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/ro
ot rd.lvm.lv=rl/swap rhgb quiet
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point regi
sters'
[ 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 0x0000000000000000-0x0000000000009fbff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000009fc00-0x0000000000009ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x000000000000f0000-0x000000000000ffffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000001000000-0x00000000007ffff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000007ffff0000-0x00000000007ffff] ACPI data
```

Рис. 16: (рис. 16. dmesg | less, версия ядра линукс, частота процессора, модель процессора)

```
[makonyaeva@user ~]$ dmesg | grep -i version
[ 0.000000] Linux version 5.14.0-427.13.1.el9_4.x86_64 (mockbuild@iad1-prod-b
uild001.bld.equ.rockylinux.org) (gcc (GCC) 11.4.1 20231218 (Red Hat 11.4.1-3), G
NU ld version 2.35.2-43.el9) #1 SMP PREEMPT_DYNAMIC Wed May 1 19:11:28 UTC 2024
[ 0.003930] IOAPIC[0]: apic_id 1, version 32, address 0xfec00000, GSI 0-23
[ 0.207272] acpihp: ACPI Hot Plug PCI Controller Driver version: 0.5
[ 0.457263] Block layer SCSI generic (bsg) driver version 0.4 loaded (major 2
46)
[ 0.465569] shpchp: Standard Hot Plug PCI Controller Driver version: 0.4
[ 1.139717] AVX2 version of gcm_enc/dec engaged.
[ 1.139859] registered taskstats version 1
[ 1.759902] fuse: init (API version 7.36)
[ 2.155579] device-mapper: uevent: version 1.0.3
[ 2.736722] libata version 3.00 loaded.
[ 2.740224] ata_piix 0000:00:01.1: version 2.13
[ 2.769635] ahci 0000:00:0d.0: version 3.0
[ 3.124509] vmwgfx 0000:00:02.0: [drm] Running on SVGA version 2.
[makonyaeva@user ~]$
```

Рис. 17: (рис. 17. Версия ядра Linux (Linux version))

```
[makonyaeva@user ~]$ dmesg | grep -i processor
[ 0.000006] tsc: Detected 1992.000 MHz processor
[ 0.197509] smpboot: Total of 1 processors activated (3984.00 BogoMIPS)
[ 0.235791] ACPI: Added _OSI(Processor Device)
[ 0.235793] ACPI: Added _OSI(Processor Aggregator Device)
```

Рис. 18: (рис. 18. Частота процессора (Detected Mhz processor))

```
[makonyaeva@user ~]$ dmesg | grep -i CPU0
[ 0.196685] smpboot: CPU0: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz (family:
0x6, model: 0x8e, stepping: 0xa)
```

Рис. 19: (рис. 19. Модель процессора (CPU0))

```
[ 0.024826] Memory: 211640K/2096696K available (16384K kernel code, 5626K r
ata, 11748K rodata, 3892K init, 5956K bss, 145392K reserved, 0K cma-reserved)
[ 0.094211] Freeing SMP alternatives memory: 36K
[ 0.206208] x86/mm: Memory block size: 128MB
```

Рис. 20: (рис. 20. Объем доступной оперативной памяти)

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

Рис. 21: (рис. 21. Тип обнаруженного гипервизора)



```
[makonyaeva@user ~]$ dmesg | grep -i "VFS: Mounted root"
[makonyaeva@user ~]$ df -T
```

Filesystem	Type	1K-blocks	Used	Available	Use%	Mounted on
devtmpfs	devtmpfs	4096	0	4096	0%	/dev
tmpfs	tmpfs	1007480	0	1007480	0%	/dev/shm
tmpfs	tmpfs	402992	6168	396824	2%	/run
/dev/mapper/rl-root	xfs	13979648	6001076	7978572	43%	/
/dev/sda1	xfs	983040	277632	705408	29%	/boot
tmpfs	tmpfs	201496	104	201392	1%	/run/user/1000
/dev/sr0	iso9660	59672	59672	0	100%	/run/media/makonyaeva/VBox_GAs_6.1.30

Рис. 22: (рис. 22. Тип файловой системы корневого раздела)

```
[makonyaeva@user ~]$ dmesg | grep -i "Mounted"
[ 5.884545] systemd[1]: Mounted Huge Pages File System.
[ 5.884934] systemd[1]: Mounted POSIX Message Queue File System.
[ 5.886125] systemd[1]: Mounted Kernel Debug File System.
[ 5.887169] systemd[1]: Mounted Kernel Trace File System.
```

Рис. 23: (рис. 23. Последовательность монтирования файловых систем)



## **Вывод**

Были настроено рабочее пространство для лабораторных работ, приобретены практические навыки установки операционной системы на виртуальную машину и настройки минимально необходимых для дальнейшей работы сервисов.

## **Список литературы. Библиография**

- [1] Документация по Virtual Box: <https://www.virtualbox.org/wiki/Documentation>
- [2] Документация по Git: <https://git-scm.com/book/ru/v2>
- [3] Документация по Markdown: <https://learn.microsoft.com/ru-ru/contribute/markdown-reference>