

Лабораторная работа №1

Установка и конфигурация операционной системы на виртуальную машину

Парфенова Елизавета Евгеньевна

Содержание

1 Цель работы	5
2 Задание	6
3 Теоретическое введение	7
4 Выполнение лабораторной работы	8
4.1 Установка Linux Rocky	8
4.2 Домашнее задание	17
4.3 Настройка git	20
4.4 Создание и настройка репозитория курса	21
4.5 Настройка конвертирования md-файлов	23
5 Выводы	24
6 Контрольные вопросы	25
Список литературы	27

Список иллюстраций

4.1 Начало создания виртуальной машины	8
4.2 Размер основной памяти виртуальной машины	9
4.3 Установка конфигурации жесткого диска: загрузочный	9
4.4 Установка конфигурации жесткого диска: VDI	10
4.5 Установка конфигурации жесткого диска: динамический	10
4.6 Размер диска	11
4.7 Добавление образа операционной системы	11
4.8 Запуск виртуальной машины	12
4.9 Установка основного языка	13
4.10 Отклбчение KDUMP	13
4.11 Изменения в разделе выбора программ	13
4.12 Имя пользователя и пароль	14
4.13 Пароль для root	14
4.14 Изменение имени узла	15
4.15 Окончание установки	15
4.16 Автоматическое отключение оптического диска	16
4.17 Установка Virtual Box Guest Additions	16
4.18 Подключение образа диска дополнительной гостевой ОС	17
4.19 Просмотра последовательности загрузки системы	18
4.20 Вывод команды dmesg	18
4.21 Внрсия ядра Linux	19
4.22 Частота процессора	19
4.23 Модель процессора	19
4.24 Объем доступной оперативной памяти	19
4.25 Тип обнаруженного гипервизора	20
4.26 Тип файловой системы корневого раздела	20
4.27 Последовательность монтирования файловых систем	20
4.28 Успешно подклбченные ключи SSH и GPG	21
4.29 Создание репозитория курса	21
4.30 Заполнение репозитория и отправка на сервер	22
4.31 Отправка на сервер	22
4.32 Репозиторий на сервере	22
4.33 Успешное конвертирование	23

Список таблиц

1 Цель работы

- приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.
- создание репозитория для дальнейшего выполнения лабораторных работ
- настройка git

2 Задание

- установить операционную систему Linux Rocky на виртуальную машину Oracle VirtualBox
- создать и настроить репозиторий курса
- настроить git

3 Теоретическое введение

Виртуальная машина (VM, от англ. virtual machine) – программная или аппаратная система, эмулирующая аппаратное обеспечение компьютера и исполняющая программы для guest-платформы (guest – гостевая платформа) на host-платформе (host – хост-платформа, платформа-хозяин) или виртуализирующая некоторую платформу и создающая на ней среды, изолирующие друг от друга программы и даже операционные системы [1]

VirtualBox (Oracle VM VirtualBox) – программный продукт виртуализации для операционных систем Windows, Linux, FreeBSD, macOS, Solaris/OpenSolaris, ReactOS, DOS и других [2].

Linux (в части случаев GNU/Linux) – семейство Unix-подобных операционных систем на базе ядра Linux, включающих тот или иной набор утилит и программ проекта GNU, и, возможно, другие компоненты [3].

Rocky Linux – дистрибутив Linux, разработанный Rocky Enterprise Software Foundation. Это полный бинарно-совместимый выпуск, использующий исходный код операционной системы Red Hat Enterprise Linux (RHEL)[4].

4 Выполнение лабораторной работы

4.1 Установка Linux Rocky

Выполнять лабораторные работы будем на операционной системе Linux Ubuntu. Начнем с установки Oracle VirtualBox. Я выполнила это с помощью терминала командой `sudo apt-get install virtualbox`. Установка произошла в автоматически определенную виртуальную машину директорию.

После успешной установки virtualbox я скачала образ диска с официального сайта Rocky Linux [5] и начала создание новой виртуальной машины. Для этого кликнула кнопку “Создать”. Затем заполнила название виртуальной машины в соответствии с соглашением об именовании и ее характеристики (Linux, Red Hat 64-bit) в открывшемся окне (рис. 4.1).

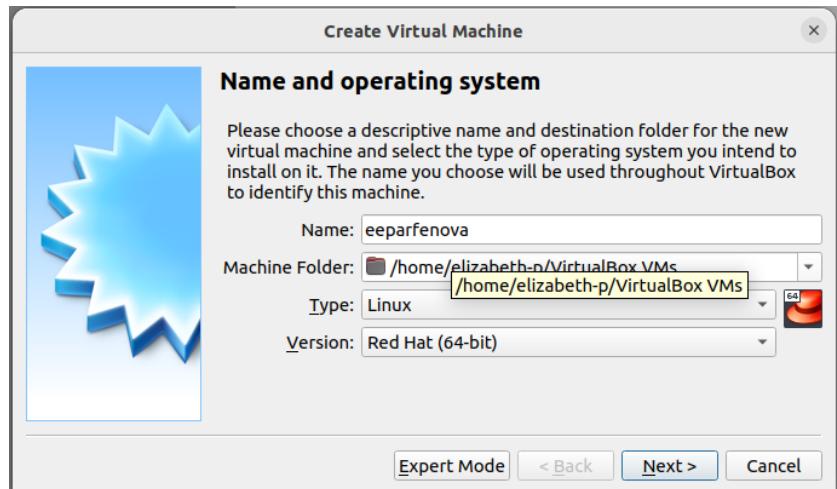


Рис. 4.1: Начало создания виртуальной машины

Далее я устанавливала размер основной памяти виртуальной машины в 2048

МБ (рис. 4.2). Задала конфигурацию жёсткого диска — загрузочный (рис. 4.3), VDI (VirtualBox Disk Image) (рис. 4.4), динамический виртуальный диск (рис. 4.5). Задала размер диска в 40 ГБ, его расположение я не изменяла (рис. 4.6).

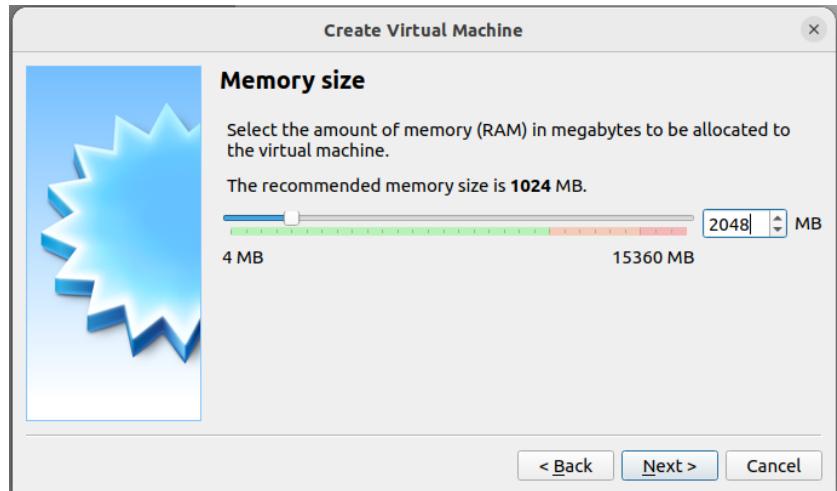


Рис. 4.2: Размер основной памяти виртуальной машины

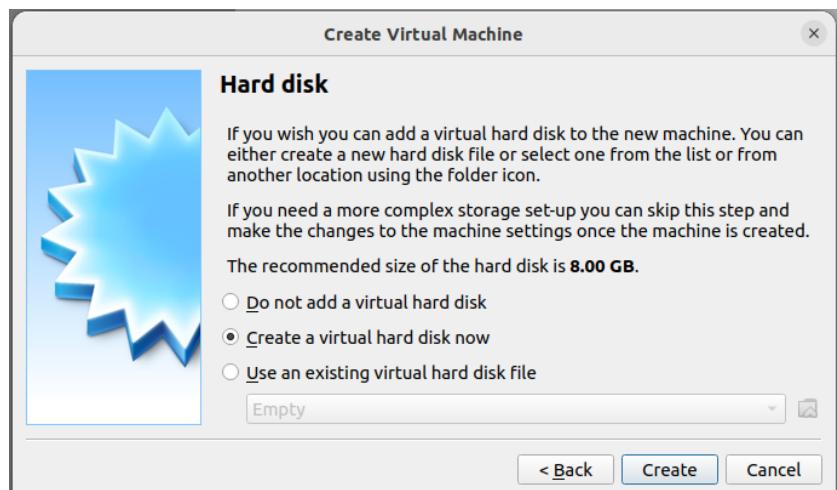


Рис. 4.3: Установка конфигурации жесткого диска: загрузочный

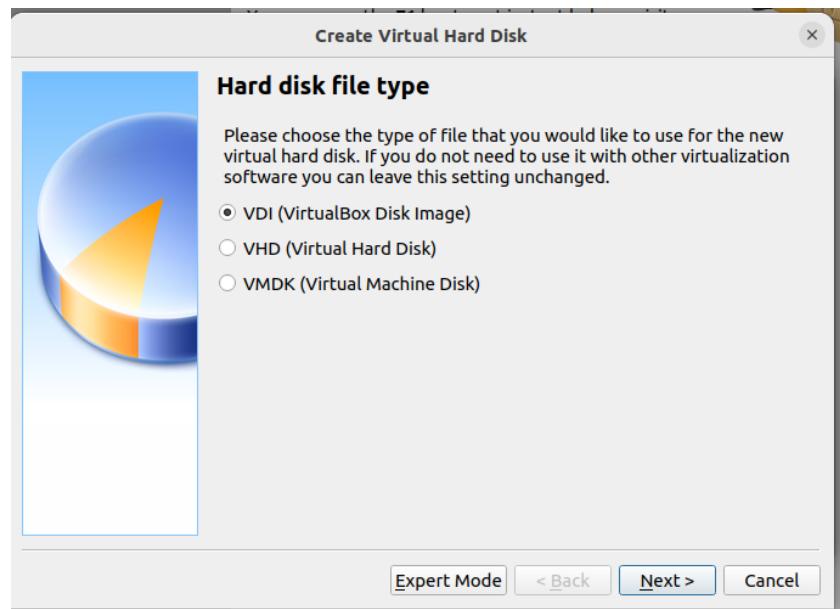


Рис. 4.4: Установка конфигурации жесткого диска: VDI

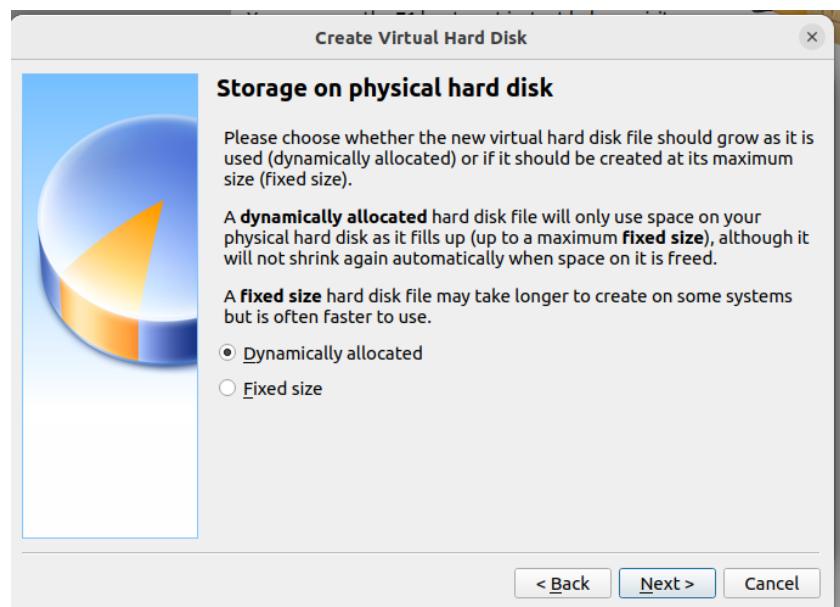


Рис. 4.5: Установка конфигурации жесткого диска: динамический

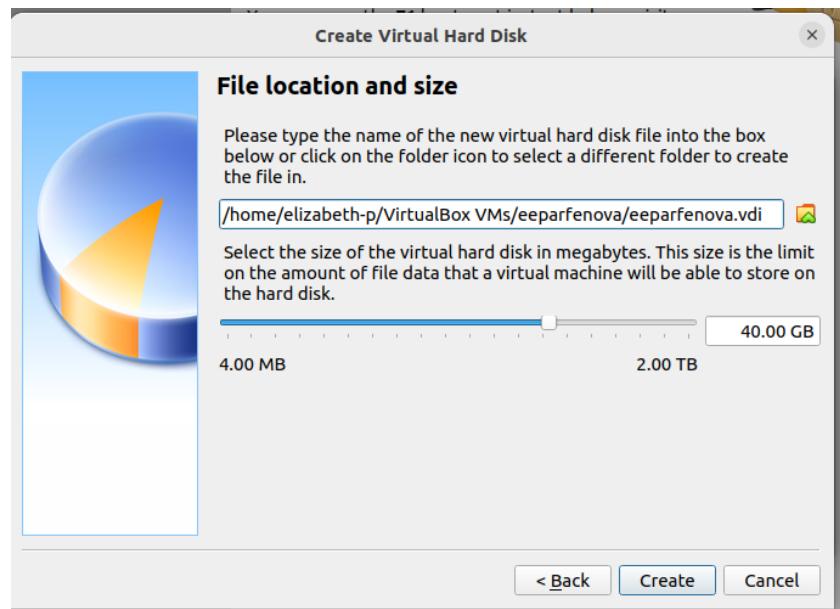


Рис. 4.6: Размер диска

В настройках VirtualBox я выбрала вкладку “Носители” и в ней добавила новый привод оптических дисков. Там я выбрала образ операционной системы, который скачала ранее (рис. 4.7).

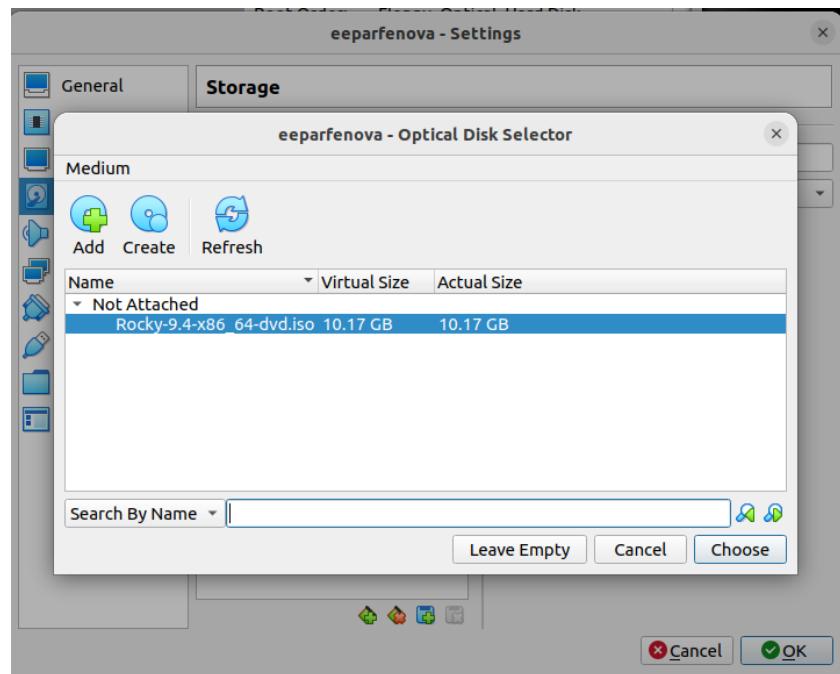


Рис. 4.7: Добавление образа операционной системы

После успешного создания виртуальной машины я приступила к установке операционной системы. В начале я запустила виртуальную машину (рис. 4.8).

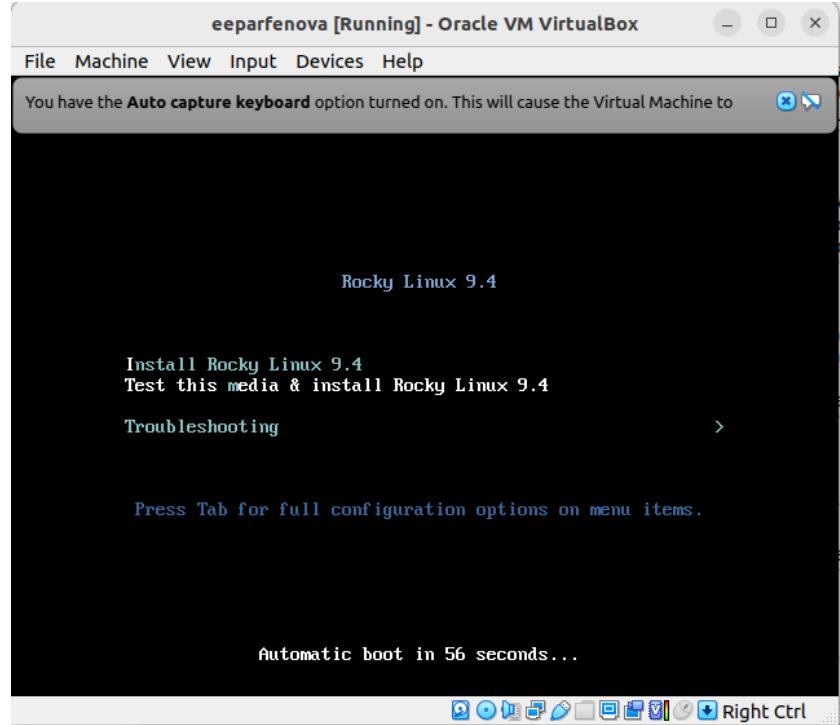


Рис. 4.8: Запуск виртуальной машины

Далее я начала настраивать установку ОС. Поставила английский язык (основной) (рис. 4.9), затем добавила русский в раскладку клавиатуры. Далее отключила KDUMP (рис. 4.10). Место установки ОС оставила без изменения, как и требовалось. В разделе выбора программ в качестве базового окружения указала Server with GUI , а в качестве дополнения – Development Tools (рис. 4.11). Затем я установила имя пользователя и пароль (рис. 4.12), а также пароль для root (рис. 4.13). Далее во вкладке Networks and Hostname в качестве имени узла указала eeparfeno.localdomain (рис. 4.14).

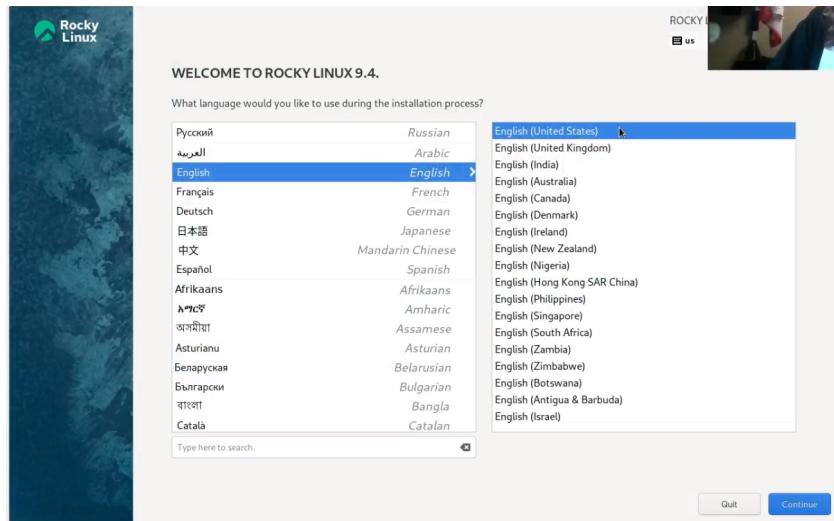


Рис. 4.9: Установка основного языка

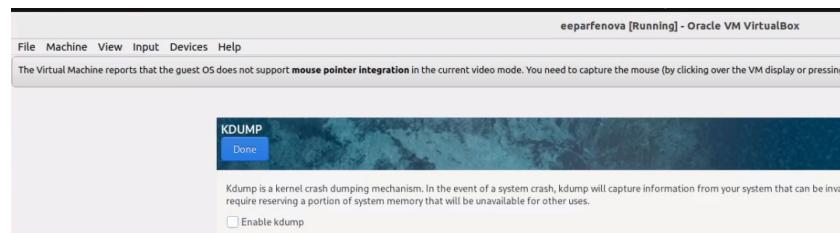


Рис. 4.10: Отклчение KDUMP

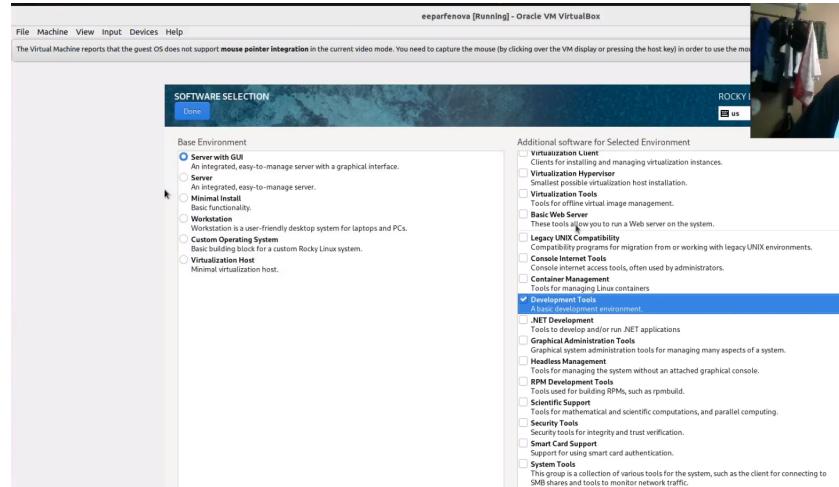


Рис. 4.11: Изменения в разделе выбора программ

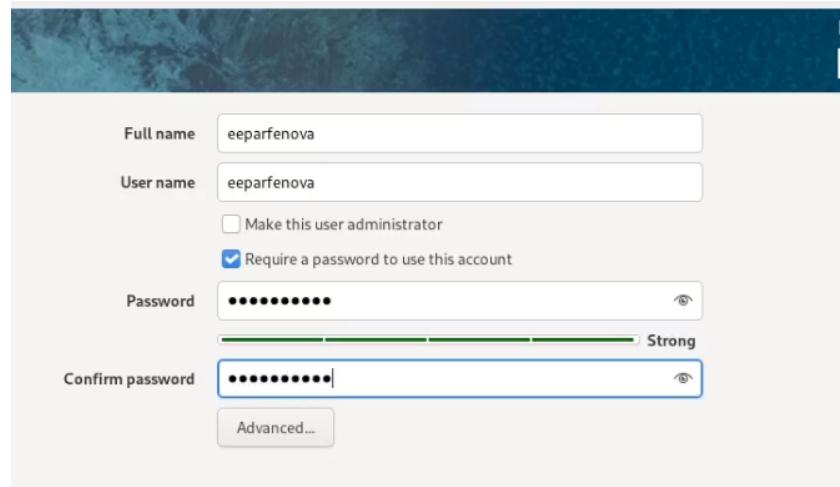


Рис. 4.12: Имя пользователя и пароль

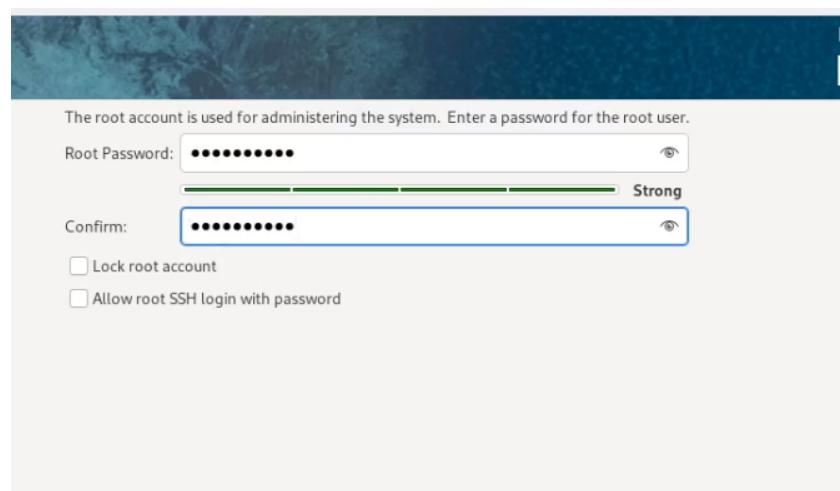


Рис. 4.13: Пароль для root

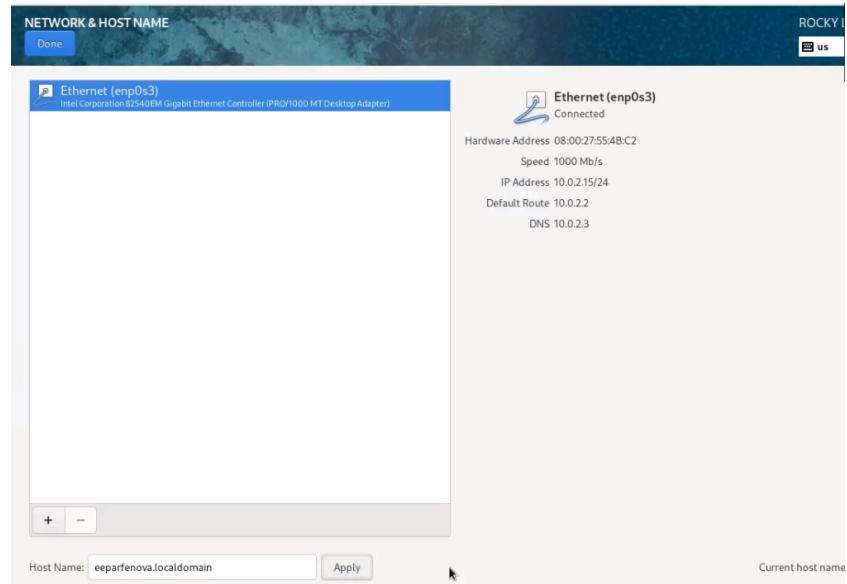


Рис. 4.14: Изменение имени узла

После успешной установки ОС (рис. 4.15), я корректно перезапустила виртуальную машину.



Рис. 4.15: Окончание установки

В VirtualBox оптический диск у меня отключился автоматически (рис. 4.16).

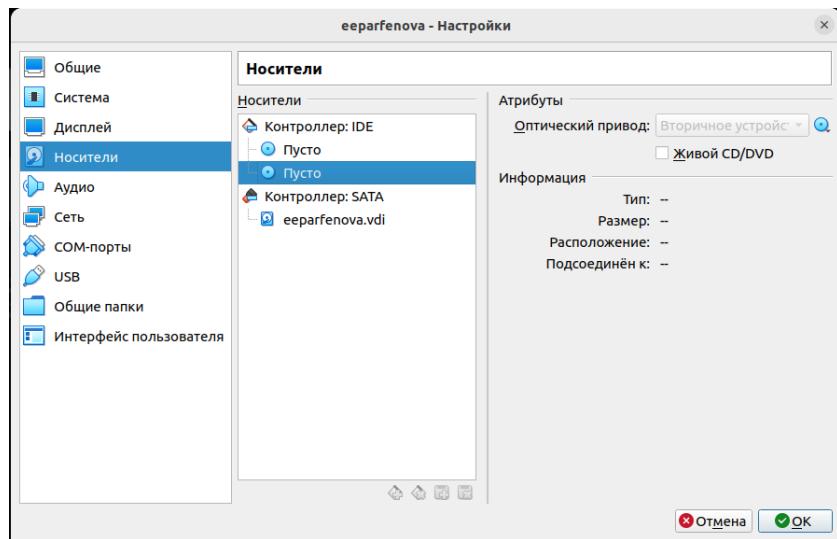


Рис. 4.16: Автоматическое отключение оптического диска

Затем в меню Устройства виртуальной машины я подключила образ диска дополнений гостевой ОС. В начале от меня потребовалось установить Virtual Box Guest Additions (рис. 4.17), а затем произошло подключение (рис. 4.18). После всего я корректно перезаустила виртуальную машину.

Имя пользователя и название хоста при установке сделаны в соответствии с соглашением об именовании.

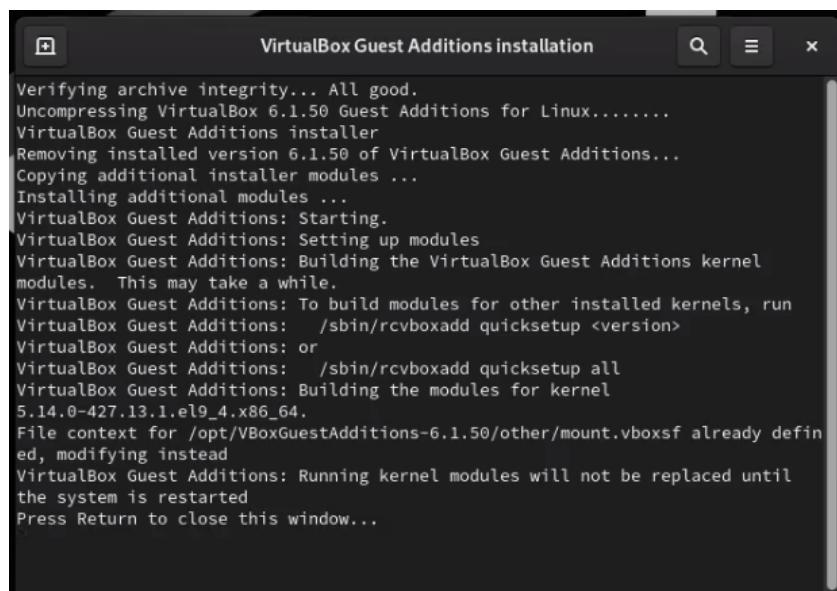


Рис. 4.17: Установка Virtual Box Guest Additions

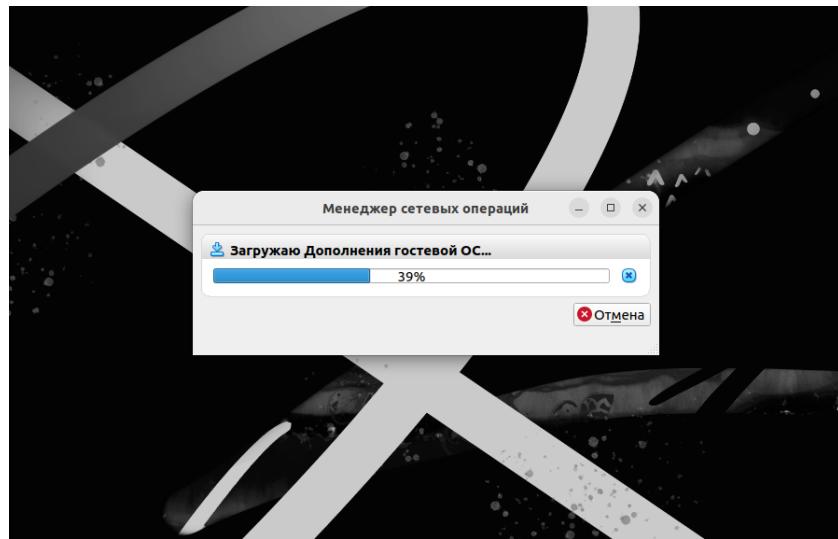


Рис. 4.18: Подключение образа диска дополнительной гостевой ОС

4.2 Домашнее задание

Выполним команду `dmesg |less` для просмотра последовательности загрузки системы (рис. 4.19). Также просто посмотрим вывод этой команды с помощью `dmesg | less` (рис. 4.20). Видим, что при загрузке системы последовательно завершилось очень много процессов.

```
[eeparfenova@eeparfenova ~]$ dmesg
[    0.000000] Linux version 5.14.0-427.13.1.el9_4.x86_64 (mockbuild@iadi1-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] [Firmware Bug]: TSC doesn't count with P0 frequency!
[    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
[    0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000100000-0x0000000007ffffff] usable
[    0.000000] BIOS-e820: [mem 0x000000000007ffff00-0x0000000007ffffffff] ACPI data
[    0.000000] BIOS-e820: [mem 0x000000000fec0000-0x000000000fec0fff] reserved
[    0.000000] BIOS-e820: [mem 0x000000000fee00000-0x000000000fee0fff] reserved
[    0.000000] BIOS-e820: [mem 0x000000000fffc0000-0x000000000fffffff] 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 cached offset of 5256266171 msrs...
```

Рис. 4.19: Просмотр последовательности загрузки системы

```
[eeparfenova@eeparfenova ~]$ dmesg
[    0.000000] Linux version 5.14.0-427.13.1.el9_4.x86_64 (mockbuild@iadi1-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] [Firmware Bug]: TSC doesn't count with P0 frequency!
[    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
[    0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000100000-0x0000000007ffffff] usable
[    0.000000] BIOS-e820: [mem 0x000000000007ffff00-0x0000000007ffffffff] ACPI data
[    0.000000] BIOS-e820: [mem 0x000000000fec0000-0x000000000fec0fff] reserved
[    0.000000] BIOS-e820: [mem 0x000000000fee00000-0x000000000fee0fff] reserved
[    0.000000] BIOS-e820: [mem 0x000000000fffc0000-0x000000000fffffff] 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 cached offset of 5256266171 msrs...
```

Рис. 4.20: Вывод команды dmesg

Получим следующую информацию с помощью команды *dmesg/grep* с различными параметрами. 1. Версия ядра Linux (Linux version) с помощью команды “*dmesg|grep ‘Linux Version’*”. Представлена на (рис. 4.21).

```
[eeparfeno[eeparfeno ~]$ dmesg | grep -i '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
```

Рис. 4.21: Внрсия ядра Linux

2. Частота процессора (Detected Mhz processor) с помощью команды “dmesg|grep ‘Mhz”’. Представлена на (рис. 4.22).

```
[ 0.000000] Hyper-V is not present. KVM
[eeparfeno[eeparfeno ~]$ dmesg | grep -i 'Mhz'
[ 0.000008] tsc: Detected 3293.806 MHz processor
[ 4.009235] e1000 0000:00:03.0 eth0: (PCI:33MHz:32-bit) 08:00:27:55:4b:c2
[eeparfeno[eeparfeno ~]$
```

Рис. 4.22: Частота процессора

3. Модель процессора (CPU0) с помощью команды “dmesg|grep ‘CPU0’ ”(рис. 4.23). Модель процессора моего девайса AMD Ryzen 5.

```
[eeparfeno[eeparfeno ~]$ dmesg | grep -i 'CPU0'
[ 0.052474] CPU0: Hyper-Threading is disabled
[ 0.176328] smpboot: CPU0: AMD Ryzen 5 5600H with Radeon Graphics (family: 0x19, model: 0x50, stepping: 0x0)
```

Рис. 4.23: Модель процессора

4. Объем доступной оперативной памяти (Memory available) с помощью команды “dmesg|grep ‘Memory’ ”. Все данные представлены на (рис. 4.24).

```
[eeparfeno[eeparfeno ~]$ dmesg | grep -i 'Memory'
[ 0.001290] ACPI: Reserving FACP table memory at [mem 0x7ffff00f0-0x7ffff01e3]
[ 0.001291] ACPI: Reserving DSDT table memory at [mem 0x7ffff0470-0x7ffff2794]
[ 0.001292] ACPI: Reserving FACS table memory at [mem 0x7ffff0200-0x7ffff023f]
[ 0.001293] ACPI: Reserving FACS table memory at [mem 0x7ffff0200-0x7ffff023f]
[ 0.001293] ACPI: Reserving APIC table memory at [mem 0x7ffff0240-0x7ffff0293]
[ 0.001294] Reserving SSDT table memory at [mem 0x7ffff02a0-0x7ffff046b]
[ 0.001746] Early memory node ranges
[ 0.002742] PM: hibernation: Registered nosave memory: [mem 0x00000000-0x0000ffff]
[ 0.002745] PM: hibernation: Registered nosave memory: [mem 0x0009f000-0x0009ffff]
[ 0.002746] PM: hibernation: Registered nosave memory: [mem 0x000a0000-0x000effff]
[ 0.002747] PM: hibernation: Registered nosave memory: [mem 0x000f0000-0x000fffff]
[ 0.014201] Memory: 260860K/2096696K available (16384K kernel code, 5626K rdata, 11748K rodata
, 3892K init, 5956K bss, 143428K reserved, 0K cma-reserved)
[ 0.065528] Freeing SMP alternatives memory: 36K
[ 0.194559] x86/mm: Memory block size: 128MB
[ 0.471782] Non-volatile memory driver v1.3
[ 1.174636] Freeing initrd memory: 55196K
[ 1.510418] Freeing unused decrypted memory: 2028K
[ 1.511134] Freeing unused kernel image (initmem) memory: 3892K
[ 1.511505] Freeing unused kernel image (rodata/data gap) memory: 540K
[ 3.358248] vmwgfx 0000:00:02.0: [drm] Legacy memory limits: VRAM = 16384 kB, FIFO = 2048 kB, surface = 507904 kB
[ 3.358255] vmwgfx 0000:00:02.0: [drm] Maximum display memory size is 16384 kB
```

Рис. 4.24: Объем доступной оперативной памяти

5. Тип обнаруженногого гипервизора (Hypervisor detected) с помощью команды “dmesg|grep ‘Hypervisor detected’ ”(рис. 4.25). Тип гипервизора: KVM

```
[eeparfenova@eeparfenova ~]$ dmesg | grep -i 'Hypervisor detected'  
[ 0.000000] Hypervisor detected: KVM
```

Рис. 4.25: Тип обнаруженногого гипервизора

6. Тип файловой системы корневого раздела с помощью команды “df -T”(рис. 4.26). Это единственная команда, которая вывела корректный результат. Здесь видим, что тип файловой системы корневого раздела - xfs

```
[eeparfenova@eeparfenova ~]$ 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   6204  396788  2% /run  
/dev/mapper/rl-root xfs   38678528 6165980 32512548 16% /  
/dev/sdal       xfs    983040  275668  707372  29% /boot  
tmpfs          tmpfs    201496   108  201388  1% /run/user/1000  
/dev/sr1        iso9660  62550   62550     0 100% /run/media/eeparfenova/VBox_GAs_6.1.  
50
```

Рис. 4.26: Тип файловой системы корневого раздела

7. Последовательность монтирования файловых систем с помощью команды “dmesg|grep ‘Mounted’ ”. Она представлена на (рис. 4.27).

```
[eeparfenova@eeparfenova ~]$ dmesg | grep -i 'Mounted'  
[ 5.585138] systemd[1]: Mounted Huge Pages File System.  
[ 5.588430] systemd[1]: Mounted POSIX Message Queue File System.  
[ 5.589491] systemd[1]: Mounted Kernel Debug File System.  
[ 5.589838] systemd[1]: Mounted Kernel Trace File System.
```

Рис. 4.27: Последовательность монтирования файловых систем

4.3 Настройка git

На самом деле, так как я работаю на операционной системе, на которой уже выполняла лабораторные работы из прошлых курсов, git был подключен к моему локальному серверу еще полгода назад, на курсе “Математическое моделирование”. Была произведена базовая настройка git с заданием имени email и имени пользователя владельца, а также настроены ключи SSH и GPG (рис. 4.28).

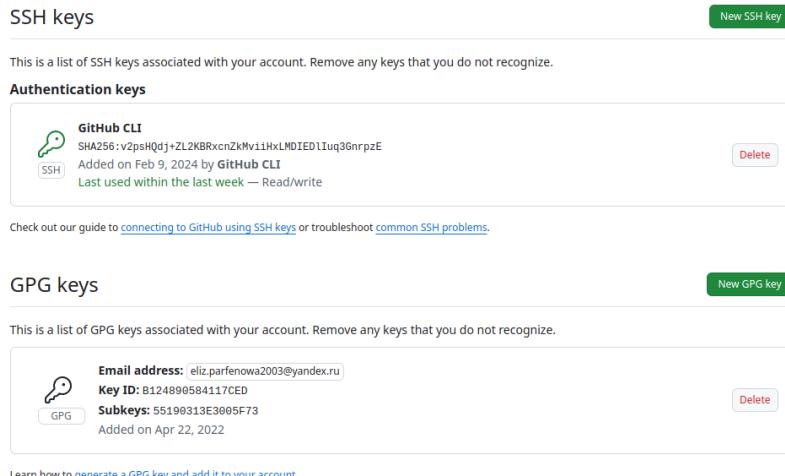


Рис. 4.28: Успешно подключенные ключи SSH и GPG

4.4 Создание и настройка репозитория курса

Мы, как обычно, сделали репозиторий на основе шаблона (рис. 4.29), я делала это через консоль. В начале я создала необходимый каталог с правильным названием командой `mkdir -p ~/work/study/2024-2025/“Информационная безопасность”`, затем перешла в него и создала репозиторий, клонировав шаблон из свой гитхаб:

```
gh repo create study_2024-2025_infosec -template=yamadharma/course-directory-student-template -public
```

```
git clone -recursive git@github.com:parfenovaee/study_2024-2025_infosec.git infosec
```

Рис. 4.29: Создание репозитория курса

Далее перешла в каталог курса командой `cd ~/work/study/2024-2025/“Информационная безопасность”/infosec` и удалила лишние файлы командой `rm package.json`. Создала

необходимые каталоги комнадами `echo infosec > COURSE` и `make prepare`. Автоматически репозиторий наполнился каталогами ко всем лабораторным, индивидуальному проекту и презентации. Затем я все загрузила на сервер командами `git add .`, `git commit -am 'feat(main): make course structure'`, `git push`. (рис. 4.30) (рис. 4.31)

```
elizabeth-pc@elizabeth-Pc:~/work/study/2024-2025/Информационная безопасность/infosec$ rm package.json
elizabeth-pc@elizabeth-Pc:~/work/study/2024-2025/Информационная безопасность/infosec$ echo Infosec > COURSE
elizabeth-pc@elizabeth-Pc:~/work/study/2024-2025/Информационная безопасность/infosec$ make prepare
elizabeth-pc@elizabeth-Pc:~/work/study/2024-2025/Информационная безопасность/infosec$ git add .
elizabeth-pc@elizabeth-Pc:~/work/study/2024-2025/Информационная безопасность/infosec$ git commit -am 'feat(main): make course structure'
[main 9619bfc] feat(main): make course structure
 261 files changed, 62629 insertions(+), 14 deletions(-)
create mode 100644 .gitignore
create mode 100644 .labi/README.ru.md
create mode 100644 .labi/presentation/projecttitle
create mode 100644 .labi/presentation/projectoutline
create mode 100644 .labi/presentation/releasenote
create mode 100644 .labi/presentation/makefile
create mode 100644 .labi/presentation/makefilebulkybov.jpg
elizabeth-pc@elizabeth-Pc:~/work/study/2024-2025/Информационная безопасность/infosec$
```

Рис. 4.30: Заполнение репозитория и отправка на сервер

```
elizabeth-pc@elizabeth-Pc:~/work/study/2024-2025/Информационная безопасность/infosec$ git push
Перенесено объектов: 49, готово.
Несколько объектов были изменены, но не скопированы.
При скопии изменений используется до 12 потоков
Скоплено объектов: 108K (39/39), готово.
Всего 49 (изменений 4), скоплено 13 из 49 (2.94 МиБ/с), готово.
Всего 38 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
Репозиторий был обновлен с помощью 1 локального объекта.
Для git clone https://github.com/yamadharma/course-directory-student-template/infosec.git
elizabeth-pc@elizabeth-Pc:~/work/study/2024-2025/Информационная безопасность/infosec$
```

Рис. 4.31: Отправка на сервер

Все прошло успешно (рис. 4.32)

Commit	Message	Date
9619bfc · 2 days ago	feat(main): make course structure	2 Commits
	Initial commit	2 days ago
	feat(main): make course structure	2 days ago
	feat(main): make course structure	2 days ago
	feat(main): make course structure	2 days ago
	Initial commit	2 days ago
	feat(main): make course structure	2 days ago
	feat(main): make course structure	2 days ago
	Initial commit	2 days ago
	Initial commit	2 days ago
	Initial commit	2 days ago
	Initial commit	2 days ago
	feat(main): make course structure	2 days ago

Рис. 4.32: Репозиторий на сервере

4.5 Настройка конвертирования md-файлов

Так как Pandoc и LaTex уже были установлены на мой компьютер, я просто попробовала конвертировать шаблон отчета к первой лабораторной работе в doc и pdf, а также сконвертировать презентацию из md в pdf и html. Все это выполнялось командой *make* (рис. 4.33).

```
elizabed-p@elizabed-P:~/work/study/2024-2025/Информационная Безопасность/infosec/labs/lab1/report$ make
pandoc "report.md" --filter pandoc-crossref --number-sections --cteproc -o "report.docx"
pandoc report.md --filter pandoc-crossref --pdf-engine=luatex --pdf-engine-opt="--shell-escape" --cteproc --number-sections -o "report.pdf"
[WARNING] [makePDF] LaTeX Warning: Empty bibliography on input line 290.
```

Рис. 4.33: Успешное конвертирование

В начале возникли небольшие трудности:

- При конвертации отчета я дополнительно загрузила большое количество шрифтов для корректности выполнения, только тогда все сработало
- для конвертации презентации я использовала немного измененный make-файл, который был на прошлых курсах, с ним ошибок не возникло

5 Выводы

- Мы приобрели практические навыки установки операционной системы на виртуальную машину и минимально настроили ее для дальнейшей работы сервисов.
- Мы создали репозиторий для дальнейшего выполнения лабораторных работ и заполнили его необходимыми каталогами
- Мы убедились в правильности работы git с нашим устройством
- Мы проверили возможность конвертирования md-файлов в необходимые форматы

6 Контрольные вопросы

1. Какую информацию содержит учётная запись пользователя?

Имя пользователя, пароль, UID, полное имя, домашний каталог, оболочка, права доступа и группы.

2. Укажите команды терминала и приведите примеры:

- для получения справки по команде: `man` , например `man cd`
- для перемещения по файловой системе: `cd` , например `cd ~/work/study/2024-2025/“Информационная безопасность”/infosec`
- для просмотра содержимого каталога: `ls`, например `ls -l`
- для определения объёма каталога: `du -sh` , например `du -sh /home/user`
- для создания / удаления каталогов / файлов: `mkdir` / `rmdir` / `touch`
- для задания определённых прав на файл / каталог: `chmod` , например `chmod 755 script.sh`
- для просмотра истории команд: `history`

3. Что такое файловая система? Приведите примеры с краткой характеристикой

Файловая система в Linux — это способ организации и хранения данных на носителе.

Основные примеры: - ext4: современная файловая система для Linux, поддерживающая большие объёмы данных и улучшенные функции.

- XFS: высокопроизводительная файловая система, оптимизированная для работы с большими файлами.
- Btrfs: файловая система с поддержкой снимков и встроенной RAID-функциональностью.

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

Использовать команду *df -h* или “*dmesg/grep ‘Mounted’*”, как мы сделали это в файле лабы

5. Как удалить зависший процесс?

Командой *kill*

Список литературы

1. Виртуальная машина [Электронный ресурс]. Wikimedia Foundation, Inc., 2024. URL: https://ru.wikipedia.org/wiki/Виртуальная_машина.
2. Download VirtualBox [Электронный ресурс]. Oracle, 2024. URL: <https://www.virtualbox.org/wiki/Downloads>.
3. Linux [Электронный ресурс]. Wikimedia Foundation, Inc., 2024. URL: <https://www.virtualbox.org/wiki/Downloads>.
4. Rocky Linux [Электронный ресурс]. Wikimedia Foundation, Inc., 2024. URL: <https://www.virtualbox.org/wiki/Downloads>.
5. Rocky Linux [Электронный ресурс]. Linux, Red Hat, Inc., 2024. URL: <https://rockylinux.org/ru>.