

Отчет по лабораторной работе №1

Основы информационной безопасности

Дагделен Зейнап Реджеповна НКАбд-02-23

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выполнение дополнительного задания	12
5	Ответы на контрольные вопросы	15
6	Выводы	17

Список иллюстраций

3.1	Окно создания виртуальной машины	7
3.2	Окно выбора основных характеристик для гостевой ОС	8
3.3	Окно выбора объема памяти	8
3.4	Загруза операционной системы Rocky	9
3.5	Настройки	9
3.6	Настройки	9
3.7	Настройки	10
3.8	Настройки	10
3.9	Окно входа в операционную систему	11
4.1	Окно терминала	12
4.2	Версия ядра	12
4.3	Частота процессора	13
4.4	Модель процессора	13
4.5	Объем доступной оперативной памяти	13
4.6	Тип обнаруженного гипервизора	13
4.7	Тип файловой системы	14
4.8	Последовательность монтирования файловых систем	14

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. Установка и настройка операционной системы.
2. Найти следующую информацию:
 1. Версия ядра Linux (Linux version).
 2. Частота процессора (Detected Mhz processor).
 3. Модель процессора (CPU0).
 4. Объем доступной оперативной памяти (Memory available).
 5. Тип обнаруженного гипервизора (Hypervisor detected).
 6. Тип файловой системы корневого раздела.

3 Выполнение лабораторной работы

Я выполняю лабораторную работу на домашнем оборудовании, поэтому создаю новую виртуальную машину в VirtualBox, выбираю имя, местоположение и образ ISO, устанавливать будем операционную систему Rocky DVD (рис. 1).

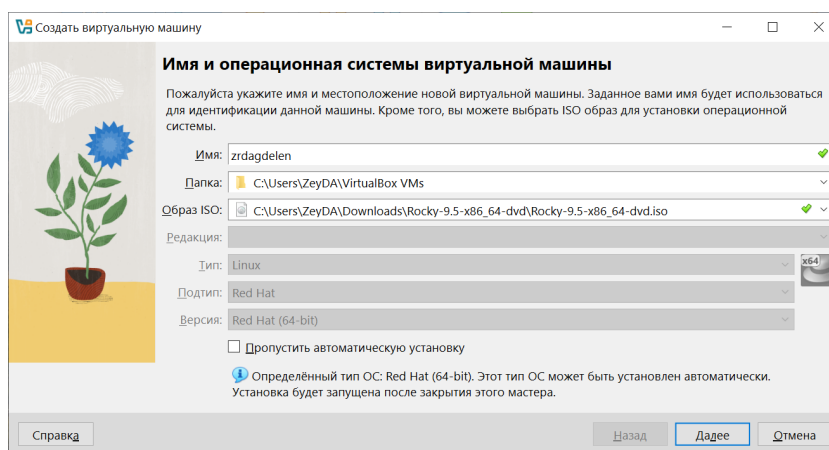


Рис. 3.1: Окно создания виртуальной машины

Выставляю основной памяти размер 2048 Мб, выбираю 2 процессора (рис. 2).

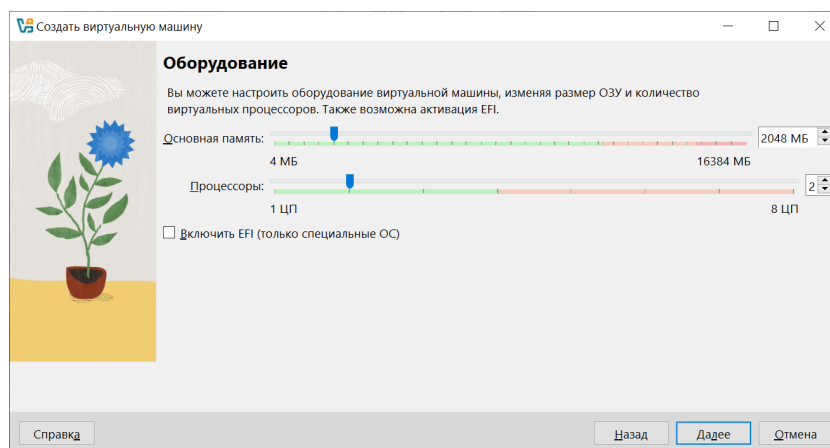


Рис. 3.2: Окно выбора основных характеристик для гостевой ОС

Выделаю 40 Гб памяти на виртуальном жестком диске (рис. 3).

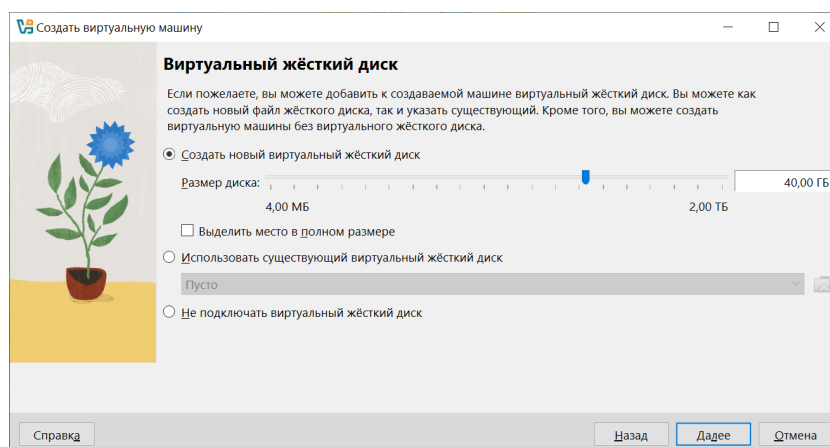


Рис. 3.3: Окно выбора объема памяти

Начинается загрузка операционной системы (рис. 4).

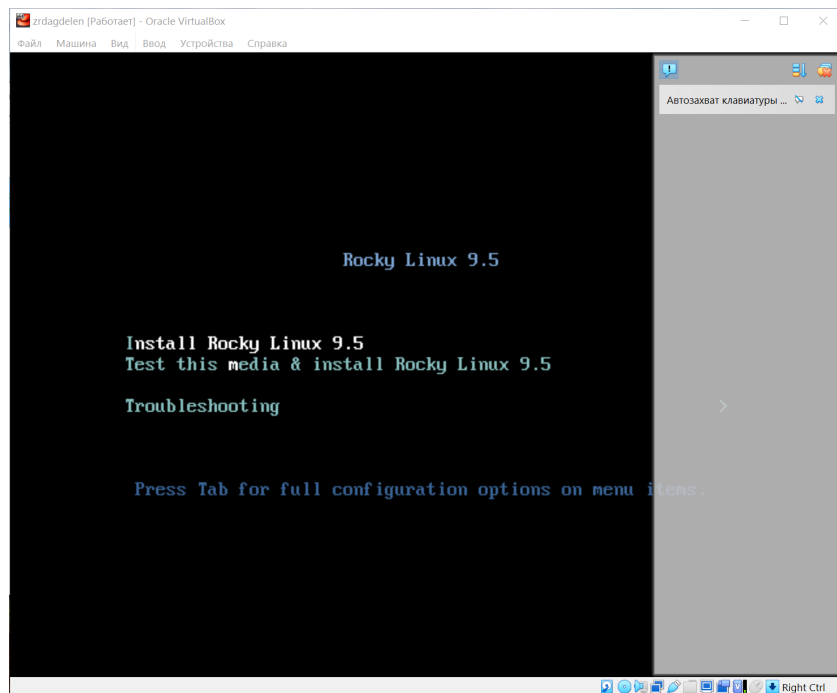


Рис. 3.4: Загрузка операционной системы Rocky

Выбираю нужные настройки (как требуется в лабораторной работе) (рис. 5-8).

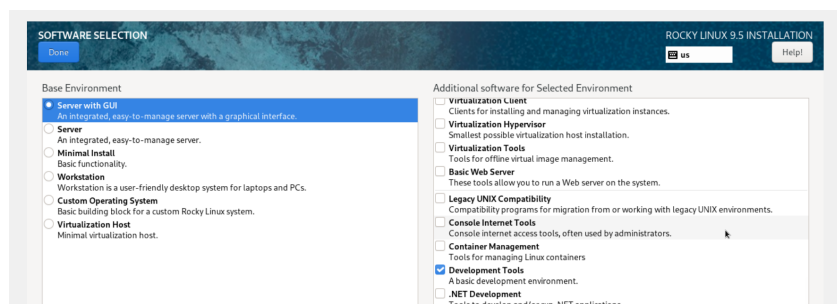


Рис. 3.5: Настройки

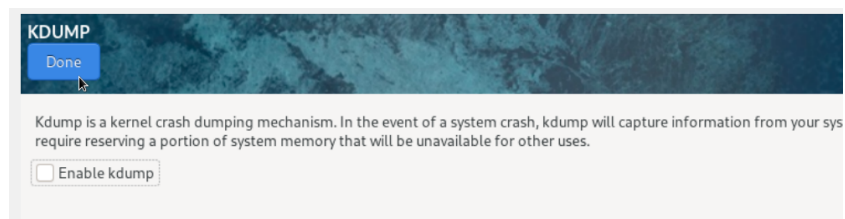


Рис. 3.6: Настройки

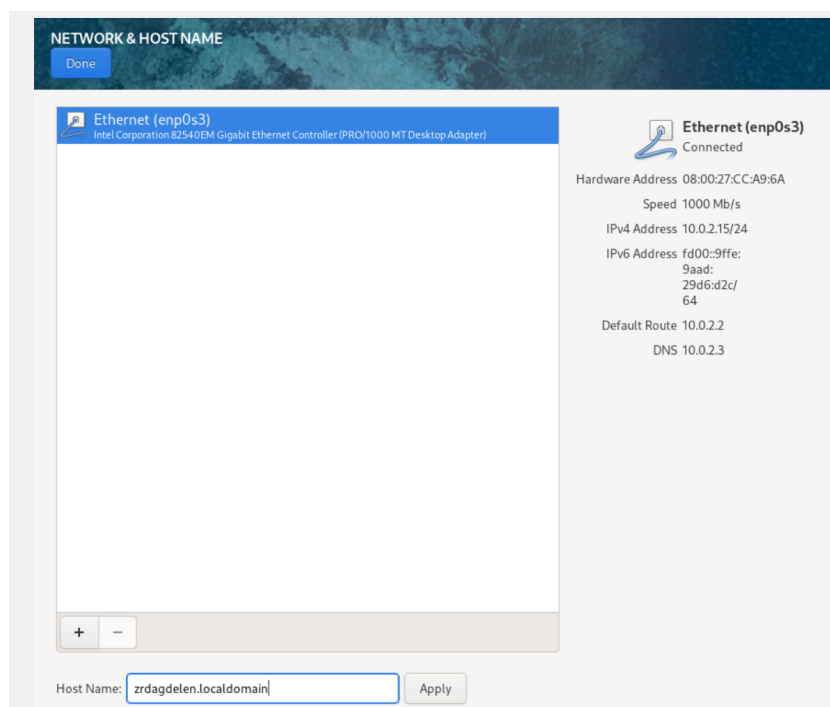


Рис. 3.7: Настройки

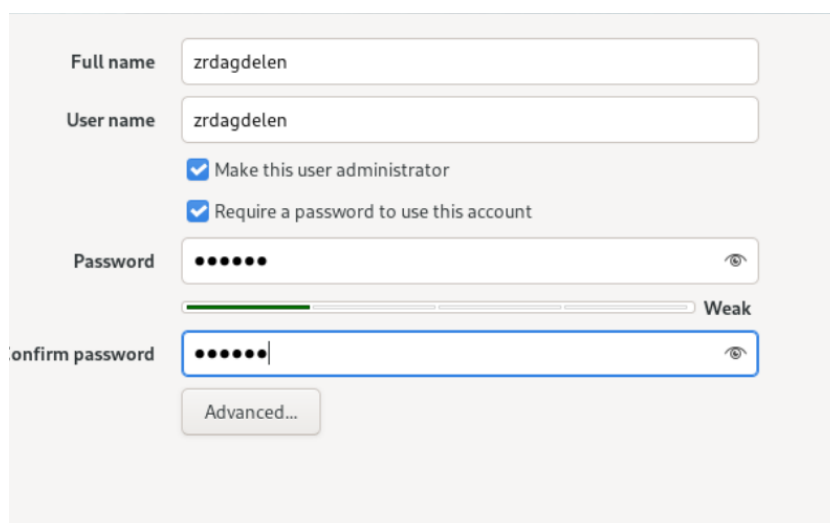


Рис. 3.8: Настройки

В итоге все скачалось и загрузилось (рис. 9).

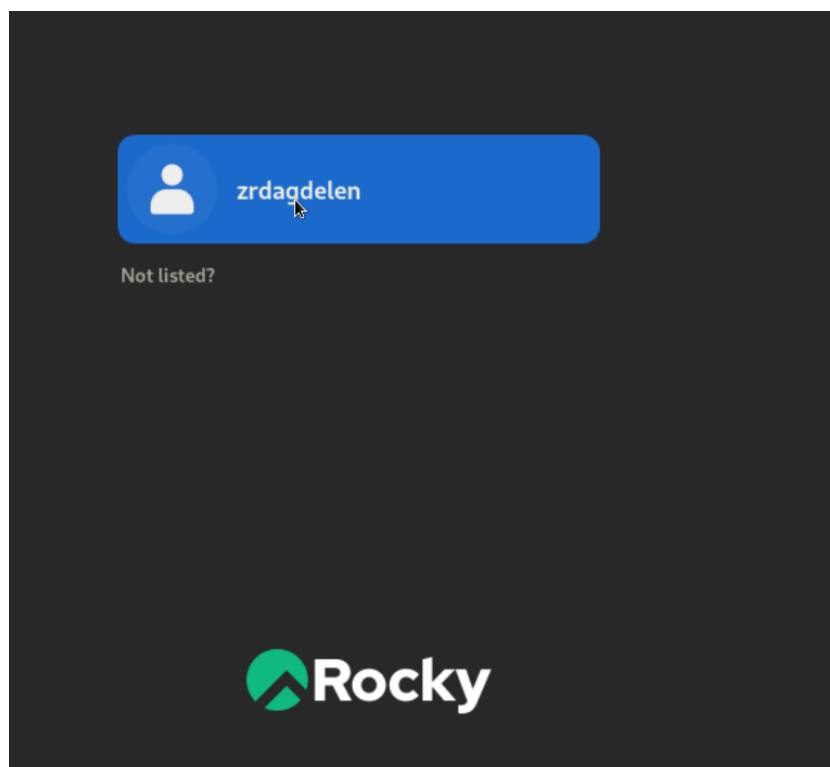
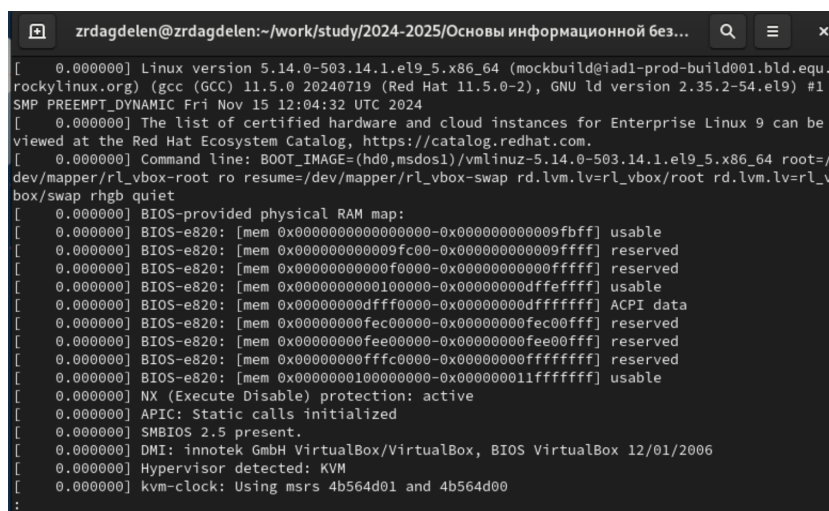


Рис. 3.9: Окно входа в операционную систему

4 Выполнение дополнительного задания

Открываю терминал, в нем прописываю `dmesg | less` (рис. 10).



```
zrdagdelen@zrdagdelen:~/work/study/2024-2025/Основы информационной без...
[ 0.000000] Linux version 5.14.0-503.14.1.el9_5.x86_64 (mockbuild@iad1-prod-build001.bld.equ.
rockylinux.org) (gcc (GCC) 11.5.0 20240719 (Red Hat 11.5.0-2), GNU ld version 2.35.2-54.el9) #1
SMP PREEMPT_DYNAMIC Fri Nov 15 12:04:32 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-5.14.0-503.14.1.el9_5.x86_64 root=/
dev/mapper/rl_vbox-root ro resume=/dev/mapper/rl_vbox-swap rd.lvm.lv=rl_vbox/root rd.lvm.lv=rl_v
box/swap rhgb quiet
[ 0.000000] BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
[ 0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000000ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000000f0000-0x000000000000ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000100000-0x000000000000ffff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000000ff0000-0x000000000000ffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000100000000-0x000000011fffffffff] usable
[ 0.000000] NX (Execute Disable) protection: active
[ 0.000000] APIC: Static calls initialized
[ 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
:
```

Рис. 4.1: Окно терминала

Версия ядра: (рис. 11).



```
[zrdagdelen@zrdagdelen infosec-intro]$ dmesg | grep -i "Linux version"
[ 0.000000] Linux version 5.14.0-503.14.1.el9_5.x86_64 (mockbuild@iad1-prod-build001.bld.equ.
rockylinux.org) (gcc (GCC) 11.5.0 20240719 (Red Hat 11.5.0-2), GNU ld version 2.35.2-54.el9) #1
SMP PREEMPT_DYNAMIC Fri Nov 15 12:04:32 UTC 2024
[zrdagdelen@zrdagdelen infosec-intro]$
```

Рис. 4.2: Версия ядра

Частота процессора: (рис. 12).

```

[zrdagdelen@zrdagdelen infosec-intro]$ dmesg | grep -i "Detected"
[ 0.000000] Hypervisor detected: KVM
[ 0.000016] tsc: Detected 2693.890 MHz processor
[ 0.068893] Warning: Deprecated Hardware is detected: x86_64-v2:GenuineIntel:Intel(R) Core(TM)
) i7-3740QM CPU @ 2.70GHz will not be maintained in a future major release and may be disabled
[ 0.659995] hub 1-0:1.0: 12 ports detected
[ 0.670314] hub 2-0:1.0: 12 ports detected
[ 2.892652] systemd[1]: Detected virtualization oracle.
[ 2.892733] systemd[1]: Detected architecture x86-64.
[ 4.514251] Warning: Unmaintained driver is detected: e1000
[ 8.457126] systemd[1]: Detected virtualization oracle.
[ 8.457153] systemd[1]: Detected architecture x86-64.
[ 14.925379] Warning: Unmaintained driver is detected: ip_set
[zrdagdelen@zrdagdelen infosec-intro]$

```

Рис. 4.3: Частота процессора

Модель процессора: (рис. 13).

```

[zrdagdelen@zrdagdelen infosec-intro]$ dmesg | grep -i "CPU0"
[ 0.315901] smpboot: CPU0: Intel(R) Core(TM) i7-3740QM CPU @ 2.70GHz (family: 0x6, model: 0x3
a, stepping: 0x9)
[zrdagdelen@zrdagdelen infosec-intro]$

```

Рис. 4.4: Модель процессора

Доступно: (рис. 14).

```

[zrdagdelen@zrdagdelen infosec-intro]$ dmesg | grep -i "Memory:"
[ 0.068841] PM: hibernation: Registered nosave memory: [mem 0x00000000-0x00000fff]
[ 0.068846] PM: hibernation: Registered nosave memory: [mem 0x0009f000-0x0009ffff]
[ 0.068849] PM: hibernation: Registered nosave memory: [mem 0x000a0000-0x000affff]
[ 0.068852] PM: hibernation: Registered nosave memory: [mem 0x000f0000-0x000fffff]
[ 0.068857] PM: hibernation: Registered nosave memory: [mem 0xdfff0000-0xdfffffff]
[ 0.068860] PM: hibernation: Registered nosave memory: [mem 0xe0000000-0xfefbffff]
[ 0.068863] PM: hibernation: Registered nosave memory: [mem 0xfec00000-0xfec0ffff]
[ 0.068866] PM: hibernation: Registered nosave memory: [mem 0xfec01000-0xfedfffff]
[ 0.068869] PM: hibernation: Registered nosave memory: [mem 0xfef00000-0xfef0ffff]
[ 0.068872] PM: hibernation: Registered nosave memory: [mem 0xfef01000-0xffffbffff]
[ 0.068875] PM: hibernation: Registered nosave memory: [mem 0xffffc0000-0xffffffffff]
[ 0.112304] Memory: 3625352K/4193848K available (16384K kernel code, 5685K rwddata, 12904K ro
ata, 3976K init, 5672K bss, 250560K reserved, 0K cma-reserved)
[ 0.212876] Freeing SMP alternatives memory: 40K
[ 2.039856] Freeing initrd memory: 57584K
[ 2.677179] Freeing unused decrypted memory: 2028K
[ 2.678001] Freeing unused kernel image (initmem) memory: 3976K
[ 2.679190] Freeing unused kernel image (rodata/data gap) memory: 1432K
[zrdagdelen@zrdagdelen infosec-intro]$

```

Рис. 4.5: Объем доступной оперативной памяти

Обнаруженный гипервизор типа KVM (рис. 15).

```

[zrdagdelen@zrdagdelen infosec-intro]$ dmesg | grep -i "Hypervisor"
[ 0.000000] Hypervisor detected: KVM
[ 0.177019] SRBDS: Unknown: Dependent on hypervisor status
[ 5.096299] vmwgfx 0000:00:02.0: [drm] *ERROR* vmwgfx seems to be running on an unsupported h
ypervisor.
[zrdagdelen@zrdagdelen infosec-intro]$

```

Рис. 4.6: Тип обнаруженного гипервизора

`sudo fdisk -l` показывает тип файловой системы, типа Linux, Linux LVM (рис. 16).

```
[zrdagdelen@zrdagdelen infosec-intro]$ sudo fdisk -l
[sudo] пароль для zrdagdelen:
Попробуйте ещё раз.
[sudo] пароль для zrdagdelen:
Диск /dev/sda: 40 GiB, 42949672960 байт, 83886080 секторов
Disk model: VBOX HARDDISK
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
Тип метки диска: dos
Идентификатор диска: 0x81488912

Устр-во      Загрузочный  i
/dev/sda1  *          начало      Конец      Секторы  Размер  Идентификатор  Тип
/dev/sda1  *          2048        2099199    2097152    1G      83 Linux
/dev/sda2          2099200    83886079   81786880    39G      8e Linux LVM

Диск /dev/mapper/rl_vbox-root: 35,05 GiB, 37631295488 байт, 73498624 секторов
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт

Диск /dev/mapper/rl_vbox-swap: 3,95 GiB, 4240441344 байт, 8282112 секторов
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
[zrdagdelen@zrdagdelen infosec-intro]$
```

Рис. 4.7: Тип файловой системы

Далее показана последовательно монтирования файловых систем (рис. 17).

```
[zrdagdelen@zrdagdelen infosec-intro]$ dmesg | grep -i "Mount"
[ 0.213243] Mount-cache hash table entries: 8192 (order: 4, 65536 bytes, linear)
[ 0.213253] Mountpoint-cache hash table entries: 8192 (order: 4, 65536 bytes, linear)
[ 7.144152] XFS (dm-0): Mounting V5 Filesystem 95536fba-2e3d-4e30-9b50-3ca20250ca25
[ 7.179279] XFS (dm-0): Ending clean mount
[ 9.728823] systemd[1]: Set up automount Arbitrary Executable File Formats File System Automount Point
.
[ 9.765456] systemd[1]: Mounting Huge Pages File System...
[ 9.769863] systemd[1]: Mounting POSIX Message Queue File System...
[ 9.774111] systemd[1]: Mounting Kernel Debug File System...
[ 9.777610] systemd[1]: Mounting Kernel Trace File System...
[ 9.840863] systemd[1]: Starting Remount Root and Kernel File Systems...
[ 9.874744] systemd[1]: Mounted Huge Pages File System.
[ 9.875636] systemd[1]: Mounted POSIX Message Queue File System.
[ 9.876643] systemd[1]: Mounted Kernel Debug File System.
[ 9.877492] systemd[1]: Mounted Kernel Trace File System.
[ 9.908175] systemd[1]: Mounting FUSE Control File System...
[ 9.916317] systemd[1]: Mounting Kernel Configuration File System...
[ 11.752873] XFS (sda1): Mounting V5 Filesystem e7bab821-d380-4273-bd1a-8dbb3a45eaca
[ 12.380022] XFS (sda1): Ending clean mount
```

Рис. 4.8: Последовательность монтирования файловых систем

5 Ответы на контрольные вопросы

1. Учетная запись содержит необходимые для идентификации пользователя при подключении к системе данные, а так же информацию для авторизации и учета: системного имени (user name) (оно может содержать только латинские буквы и знак нижнее подчеркивание, еще оно должно быть уникальным), идентификатор пользователя (UID) (уникальный идентификатор пользователя в системе, целое положительное число), идентификатор группы (GID) (группа, к к-рой относится пользователь. Она, как минимум, одна, по умолчанию - одна), полное имя (full name) (Могут быть ФИО), домашний каталог (home directory) (каталог, в к-рый попадает пользователь после входа в систему и в к-ром хранятся его данные), начальная оболочка (login shell) (командная оболочка, к-рая запускается при входе в систему).
2. Для получения справки по команде: `—help`; для перемещения по файловой системе - `cd`; для просмотра содержимого каталога - `ls`; для определения объёма каталога - `du` ; для создания / удаления каталогов - `mkdir/rmdir`; для создания / удаления файлов - `touch/rm`; для задания определённых прав на файл / каталог - `chmod`; для просмотра истории команд - `history`
3. Файловая система - это порядок, определяющий способ организации и хранения и именования данных на различных носителях информации. Примеры: FAT32 представляет собой пространство, разделенное на три части: одна область для служебных структур, форма указателей в виде таблиц и зона для хранения самих файлов. ext3/ext4 - журналируемая файловая система, используемая в основном в ОС с ядром Linux.

4. С помощью команды `df`, введя ее в терминале. Это утилита, которая показывает список всех файловых систем по именам устройств, сообщает их размер и данные о памяти. Также посмотреть подмонтированные файловые системы можно с помощью утилиты `mount`.
5. Чтобы удалить зависший процесс, вначале мы должны узнать, какой у него `id`: используем команду `ps`. Далее в терминале вводим команду `kill < id процесса >`. Или можно использовать утилиту `killall`, что “убьет” все процессы, которые есть в данный момент, для этого не нужно знать `id` процесса.

6 Выводы

Я приобрела практические навыки установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.