

# **Отчёт по лабораторной работе №1**

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

Ришард Когенгар

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Ход выполнения</b>	<b>6</b>
2.1 Установка операционной системы Rocky Linux на виртуальную машину . . . . .	6
2.2 Установка дополнений гостевой операционной системы VirtualBox . . . . .	11
2.3 Анализ загрузки системы и параметров оборудования . . . . .	12
2.4 Вывод . . . . .	14
<b>3 Контрольные вопросы</b>	<b>15</b>

# **Список иллюстраций**

2.1	Создание виртуальной машины и выбор ISO-образа . . . . .	7
2.2	Основные параметры виртуальной машины . . . . .	8
2.3	Выбор языка установки . . . . .	9
2.4	Сводка параметров установки . . . . .	10
2.5	Завершение установки Rocky Linux . . . . .	11
2.6	Установка VirtualBox Guest Additions . . . . .	12
2.7	Анализ загрузки системы и монтирования файловых систем . . . . .	13
2.8	Анализ загрузки системы и 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 Ход выполнения**

### **2.1 Установка операционной системы Rocky Linux на виртуальную машину**

1. В среде виртуализации **Oracle VirtualBox** начато создание новой виртуальной машины.

В качестве имени виртуальной машины указано **rishardkogengar**, каталог размещения – стандартный каталог VirtualBox VMs в профиле пользователя.

В качестве установочного носителя выбран ISO-образ **Rocky-10.1-x86\_64-dvd1.iso**, расположенный в каталоге загрузок.

Тип операционной системы установлен как **Linux**, подтип – **Red Hat**, версия – **Red Hat (64-bit)**.

Автоматическая установка гостевой операционной системы отключена, поскольку установка выполнялась в ручном режиме с контролем всех параметров.

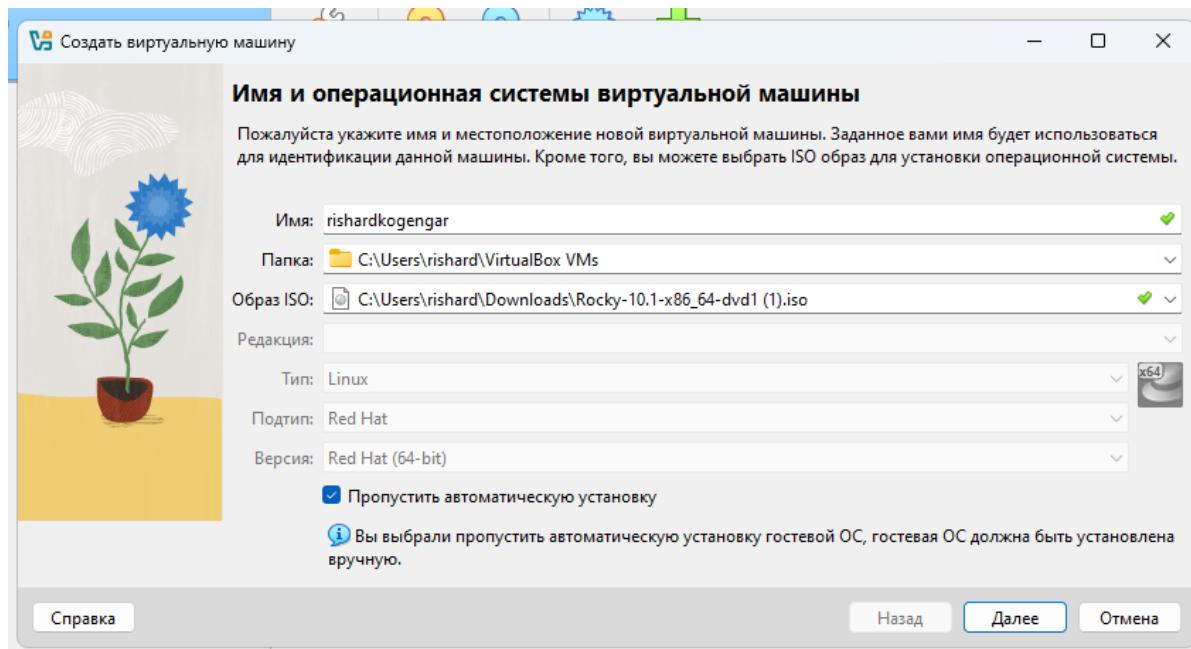


Рис. 2.1: Создание виртуальной машины и выбор ISO-образа

2. После создания виртуальной машины выполнена её базовая настройка.

Для виртуальной машины выделено **4096 МБ оперативной памяти** и **4 виртуальных процессора**, что обеспечивает корректную и стабильную работу серверной системы с графическим интерфейсом.

В качестве графического контроллера выбран **VMSVGA**, объём видеопамяти установлен равным **128 МБ**.

Создан виртуальный жёсткий диск формата **VDI** объёмом **40 ГБ**, подключённый к SATA-контроллеру.

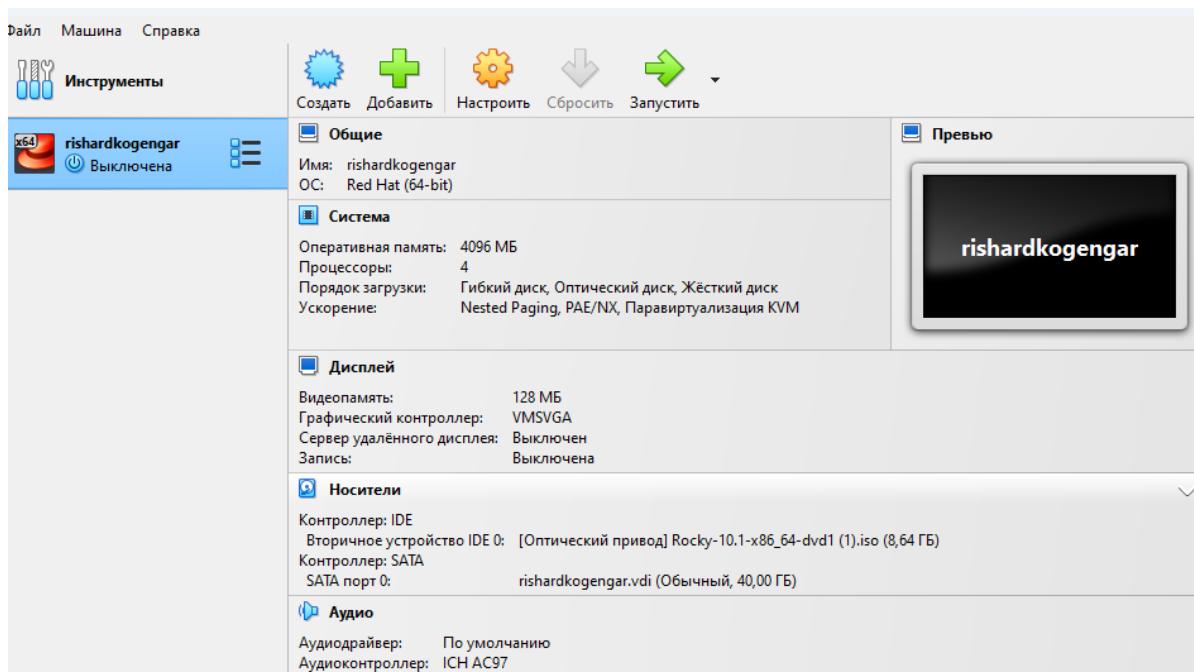


Рис. 2.2: Основные параметры виртуальной машины

3. Виртуальная машина была запущена, после чего началась загрузка установщика операционной системы Rocky Linux с подключённого ISO-образа.
4. После загрузки установщика выполнен выбор языка установки.

В качестве языка интерфейса выбран **English (United States)**, что обеспечивает корректную локализацию системных компонентов и совместимость с серверным программным обеспечением.

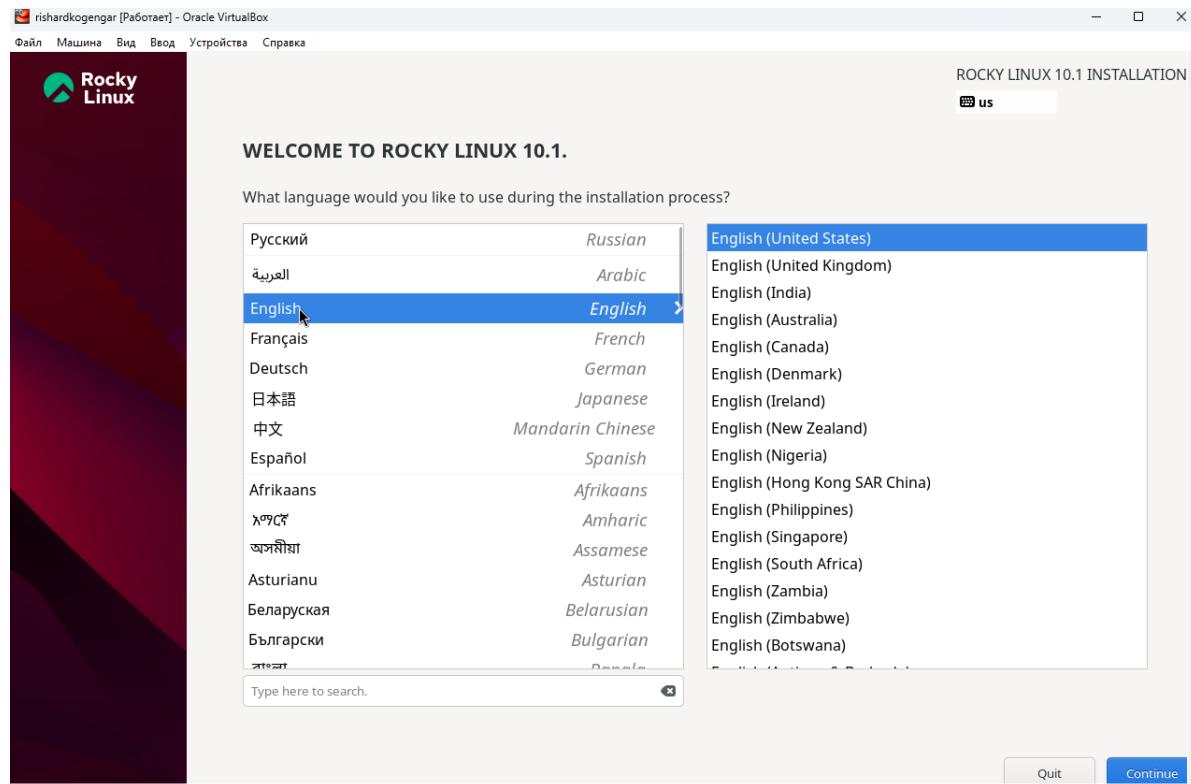


Рис. 2.3: Выбор языка установки

5. На экране **Installation Summary** автоматически определены источник установки и целевой диск.

В разделе **Software Selection** в качестве базовой среды выбрана **Server with GUI**.

Настроены параметры локализации, часовой пояс **Europe/Moscow**, сетевое подключение, а также заданы параметры учётных записей пользователя и администратора (root).

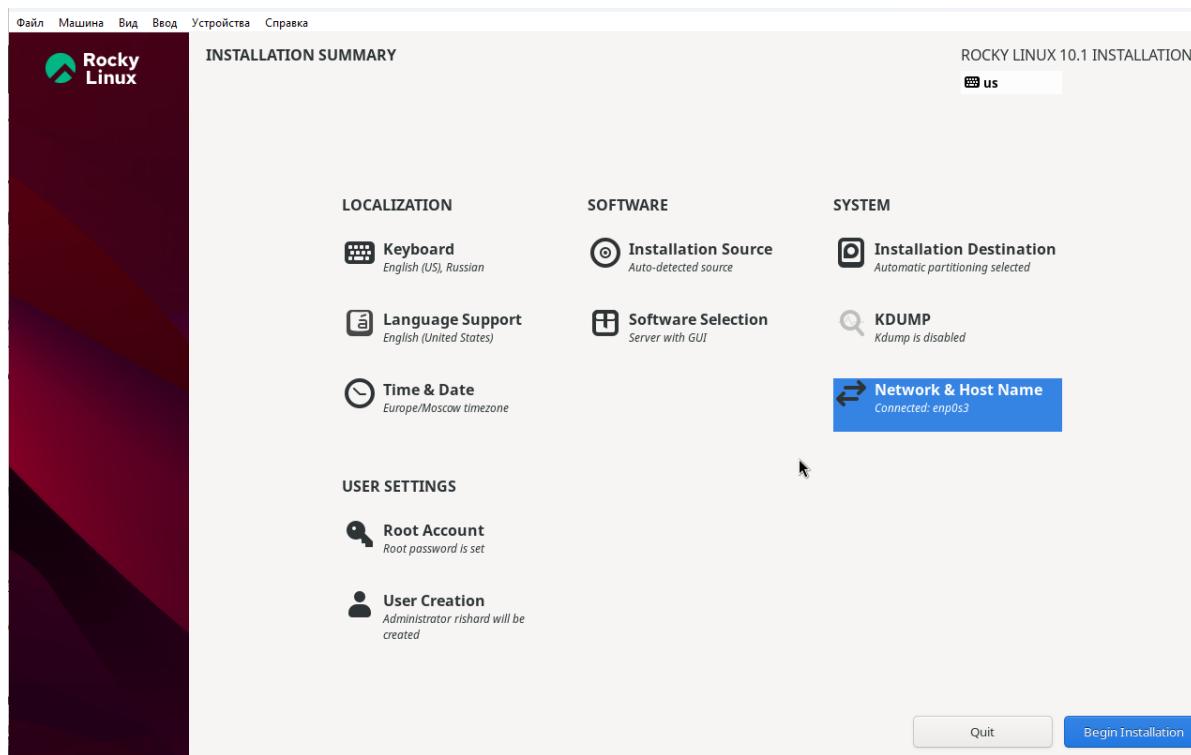


Рис. 2.4: Сводка параметров установки

6. После завершения настройки параметров запущен процесс установки операционной системы.

В ходе установки выполнялось копирование файлов и конфигурирование системных компонентов.

По завершении процесса система уведомила об успешной установке и необходимости перезагрузки.



Рис. 2.5: Завершение установки Rocky Linux

## 2.2 Установка дополнений гостевой операционной системы **VirtualBox**

7. После первого входа в систему выполнена установка дополнений гостевой операционной системы **VirtualBox Guest Additions**, необходимых для корректной работы графического интерфейса и повышения производительности.

Пользователь получил полномочия администратора, перешёл в каталог с образом дополнений гостевой ОС и запустил установочный файл **VBoxLinuxAdditions.run**.

В процессе установки были успешно собраны и загружены модули ядра VirtualBox, после чего система была корректно перезагружена.

```
root@rishardkogengar:/home/rishard#
root@rishardkogengar:/home/rishard# cd /run/media/rishard/VBox_GAs_7.1.12/
root@rishardkogengar:/run/media/rishard/VBox_GAs_7.1.12# ./VBoxLinuxAdditions.run
Verifying archive integrity... 100% MD5 checksums are OK. All good.
Uncompressing VirtualBox 7.1.12 Guest Additions for Linux 100%
VirtualBox Guest Additions installer
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
6.12.0-124.8.1.el10_1.x86_64.

VirtualBox Guest Additions: Look at /var/log/vboxadd-setup.log to find out what
went wrong
VirtualBox Guest Additions: reloading kernel modules and services
VirtualBox Guest Additions: kernel modules and services 7.1.12 r169651 reloaded
VirtualBox Guest Additions: NOTE: you may still consider to re-login if some
user session specific services (Shared Clipboard, Drag and Drop, Seamless or
Guest Screen Resize) were not restarted automatically
root@rishardkogengar:/run/media/rishard/VBox_GAs_7.1.12# █
```

Рис. 2.6: Установка VirtualBox Guest Additions

## 2.3 Анализ загрузки системы и параметров оборудования

8. После загрузки графического окружения был открыт терминал и выполнен анализ последовательности загрузки системы с использованием команды **dmesg**.

Для поиска необходимой информации применялась фильтрация вывода.

В результате анализа получены следующие сведения:

- **Версия ядра Linux:**

Linux version 6.12.0-124.8.1.el10\_1.x86\_64

- **Частота процессора:**

Detected 3187.198 MHz processor

- **Модель процессора:**

Виртуальный процессор, предоставляемый гипервизором

- **Объём доступной оперативной памяти:**

Memory: 3934248K / 4193804K available

- **Тип обнаруженного гипервизора:**

Hypervisor detected: KVM

- **Тип файловой системы корневого раздела:**

Корневой раздел использует файловую систему **XFS**

- **Последовательность монтирования файловых систем:**

Информация о монтировании файловых систем получена с помощью команды **mount**, включая корневой, системные и временные разделы.

```
root@rishardkogengar:/run/media/rishard/VBox_GAs_7.1.12# dmseg | grep "Linux ver"
[    0.000000] Linux version 6.12.0-124.8.1.el10_1.x86_64 (mockbuild@iad1-prod-build001.bld.equ.rockyli
nux.org) (gcc (GCC) 14.3.1 20250617 (Red Hat 14.3.1-2), GNU ld version 2.41-58.el10) #1 SMP PREEMPT_DYN
AMIC Tue Nov 11 22:54:28 UTC 2025
root@rishardkogengar:/run/media/rishard/VBox_GAs_7.1.12# dmseg | grep "MHz"
[    0.000004] tsc: Detected 3187.198 MHz processor
[    4.708964] e1000 0000:00:03.0 eth0: (PCI:33MHz:32-bit) 08:00:27:8e:fd:ca
root@rishardkogengar:/run/media/rishard/VBox_GAs_7.1.12# dmseg | grep "memory ava"
root@rishardkogengar:/run/media/rishard/VBox_GAs_7.1.12# dmseg | grep "Memory ava"
root@rishardkogengar:/run/media/rishard/VBox_GAs_7.1.12# dmseg | grep "avai"
[    0.004439] On node 0, zone DMA: 1 pages in unavailable ranges
[    0.004453] On node 0, zone DMA: 97 pages in unavailable ranges
[    0.007840] On node 0, zone Normal: 16 pages in unavailable ranges
[    0.008118] [mem 0xe0000000-0xfeffff] available for PCI devices
[    0.155117] Memory: 3943248K/4193848K available (18432K kernel code, 5804K rwdta, 14268K rodata, 43
44K init, 6696K bss, 246060K reserved, 0K cma-reserved)
root@rishardkogengar:/run/media/rishard/VBox_GAs_7.1.12# dmseg | grep "Hyper"
[    0.00000] Hypervisor detected: KVM
root@rishardkogengar:/run/media/rishard/VBox_GAs_7.1.12#
```

Рис. 2.7: Анализ загрузки системы и монтирования файловых систем

```

root@rishardkogengar:/run/media/rishard/VBox_GAs_7.1.12# mount
/dev/mapper/r1_vbox-root on / type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=1973892k,nr_inodes=493473,mode=755,inode64)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,seclabel,nsdelegate,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime,seclabel)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=800548k,nr_inodes=819200,mode=755,inode64)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autos (rw,relatime,fd=36,pgrp=1,timeo=0,minproto=5,maxproto=5,direct,pipe_ino=7277)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
hugegetlbfs on /dev/hugepages type hugegetlbfs (rw,nosuid,nodev,relatime,seclabel,pagesize=2M)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabel)
tmpfs on /run/credentials/systemd-journald.service type tmpfs (ro,nosuid,nodev,noexec,relatime,nosymfollow,seclabel,size=1024k,nr_inodes=1024,mode=700,inode64,noswap)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
/dev/sda2 on /boot type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=400272k,nr_inodes=100068,mode=700,uid=1000,gid=1000,inode64)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
portal on /run/user/1000/doc type fuse.portal (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
tmpfs on /run/user/0 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=400272k,nr_inodes=100068,mode=700,inode64)
/dev/sr0 on /run/media/rishard/VBox_GAs_7.1.12 type iso9660 (ro,nosuid,nodev,relatime,nojoliet,check=s,map=n,blocksize=2048,uid=1000,gid=1000,mode=500,fmode=400,iocharset=utf8,uhelper=udisks2)
root@rishardkogengar:/run/media/rishard/VBox_GAs_7.1.12#

```

Рис. 2.8: Анализ загрузки системы и монтирования файловых систем

## 2.4 Вывод

В ходе выполнения работы была успешно установлена операционная система **Rocky Linux 10.1** на виртуальную машину в среде **Oracle VirtualBox**.

Выполнена базовая настройка системы, установка дополнений гостевой операционной системы и анализ параметров загрузки, оборудования и файловых систем.

Полученные результаты подтверждают корректность установки и готовность системы к дальнейшей эксплуатации.

# 3 Контрольные вопросы

## 1. Укажите команды терминала и приведите примеры

### Получение справки по команде

`man <команда>` — вывод полной справочной информации. Пример: `man ls`  
`<команда> --help` — краткая встроенная справка. Пример: `cp --help`  
`info <команда>` — справка в формате info. Пример: `info coreutils`

### Перемещение по файловой системе

`pwd` — отображение текущего каталога. Пример: `pwd`  
`cd <путь>` — переход в каталог. Примеры: `cd /etc`, `cd ..`, `cd ~`, `cd -`

### Просмотр содержимого каталога

`ls` — вывод списка файлов и каталогов.

Примеры: `ls`, `ls -l`, `ls -la`

### Определение объёма каталога

`du` — определение занимаемого места каталогом.

Примеры: `du -sh /var/log`, `du -h --max-depth=1 /home`

`df` — информация о заполнении файловых систем. Пример: `df -h`

### Создание и удаление каталогов и файлов

`mkdir <каталог>` — создание каталога. Примеры: `mkdir test`, `mkdir -p /tmp/a/b/c`

`rmdir <каталог>` — удаление пустого каталога. Пример: `rmdir empty_dir`

`rm -r <каталог>` — рекурсивное удаление каталога. Пример: `rm -r test`

`touch <файл>` — создание пустого файла. Пример: `touch file.txt`

`rm <файл>` — удаление файла. Пример: `rm file.txt`

## **Задание прав доступа к файлам и каталогам**

`chmod` – изменение прав доступа.

Примеры: `chmod 644 file.txt`, `chmod 755 script.sh`, `chmod u+x script.sh`

`chown` – изменение владельца и группы. Пример: `chown user:user file.txt`

`chgrp` – изменение группы. Пример: `chgrp developers file.txt`

## **Просмотр истории команд**

`history` – вывод истории команд пользователя. Пример: `history`

`Ctrl + R` – интерактивный поиск по истории команд в оболочке bash.

## **2. Какую информацию содержит учётная запись пользователя? Какие команды позволяют посмотреть информацию о пользователе?**

Учётная запись пользователя в Linux содержит следующие сведения: имя пользователя (login), идентификатор пользователя (UID), идентификатор основной группы (GID), список дополнительных групп, домашний каталог, используемую командную оболочку, а также параметры пароля и политики его действия.

Основная информация о пользователях хранится в файлах `/etc/passwd`, `/etc/shadow` и `/etc/group`.

Команды для просмотра информации о пользователе:

`id` – отображает UID, GID и группы пользователя. Пример: `id username`

`whoami` – показывает имя текущего пользователя. Пример: `whoami`

`groups` – список групп пользователя. Пример: `groups username`

`getent passwd <user>` – получение записи пользователя из системной базы.

`finger <user>` – расширенная информация о пользователе (если утилита установлена).

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

Файловая система – это способ организации хранения данных на носителе, определяющий структуру каталогов и файлов, правила доступа к ним, хранение метаданных и механизмы обеспечения целостности данных.

Примеры файловых систем:

**ext4** – журналируемая файловая система Linux, надёжная и универсальная.

**XFS** – высокопроизводительная журналируемая файловая система, оптимизированная для работы с большими объёмами данных.

**Btrfs** – современная файловая система с поддержкой снапшотов, сжатия и контрольных сумм.

**FAT32 / exFAT** – файловые системы с высокой совместимостью, часто применяются на съёмных носителях.

**NTFS** – основная файловая система Windows, поддерживается в Linux через специальные драйверы.

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

Для просмотра подмонтированных файловых систем используются команды:

`mount` – выводит список всех смонтированных файловых систем.

`findmnt` – отображает дерево монтирования и параметры файловых систем.

`df -T` – показывает тип файловой системы и её заполненность.

Также информацию можно получить, просмотрев системные файлы `/proc/mounts` или `/etc/mtab`.

#### **5. Как удалить зависший процесс?**

Для удаления зависшего процесса необходимо определить его идентификатор (PID) и завершить выполнение процесса.

Для поиска процесса используются команды:

`ps aux | grep <имя>`

`pgrep <имя_процесса>`

`top` или `htop`

Для завершения процесса применяются команды:

`kill <PID>` – корректное завершение процесса.

`kill -9 <PID>` – принудительное завершение процесса.

`pkill <имя>` – завершение процесса по имени.

`killall <имя>` – завершение всех процессов с указанным именем.