

# **Лабораторная работа №1**

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

Ко Антон Геннадьевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>9</b>
3.1	Настройка рабочего пространства . . . . .	10
3.2	Установка и конфигурация операционной системы на виртуальную машину	12
3.3	Домашнее задание . . . . .	15
3.4	Ответы на контрольные вопросы . . . . .	22
3.4.1	1. Что содержит информация об учетной записи пользователя? .	22
3.4.2	2. Основные команды Linux . . . . .	22
3.4.3	3. Файловые системы . . . . .	23
3.4.4	4. Команда для проверки файловой системы . . . . .	23
3.4.5	5. Команда для завершения процесса . . . . .	23
<b>4</b>	<b>Вывод</b>	<b>24</b>
<b>5</b>	<b>Список литературы. Библиография</b>	<b>25</b>

# Список иллюстраций

3.1	(рис. Проверка рабочего пространства)	10
3.2	(рис. Общие настройки)	12
3.3	(рис. Запуск системы)	13
3.4	(рис. Версия ядра Linux (Linux version).)	15
3.5	(рис. Частота процессора (Detected Mhz processor).)	16
3.6	(рис. Модель процессора (CPU0).)	17
3.7	(рис. Объем доступной оперативной памяти (Memory available).)	18
3.8	(рис. Тип обнаруженного гипервизора (Hypervisor detected).)	19
3.9	(рис. Тип файловой системы корневого раздела.)	20
3.10	(рис. Последовательность монтирования файловых систем.)	21

## Список таблиц

---

# 1 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. Изучить идеологию и применение средств контроля версий, освоить умения по работе с git. Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

## 2 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 (автоматически 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:

Название	
команды	Описание команды
git clone	Клонирование репозитория на ПК
git commit -m "Initial Commit"	Оставление коммита
git push	Загрузка изменений на гит

Название	
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]





# 3 Выполнение лабораторной работы

## 3.1 Настройка рабочего пространства

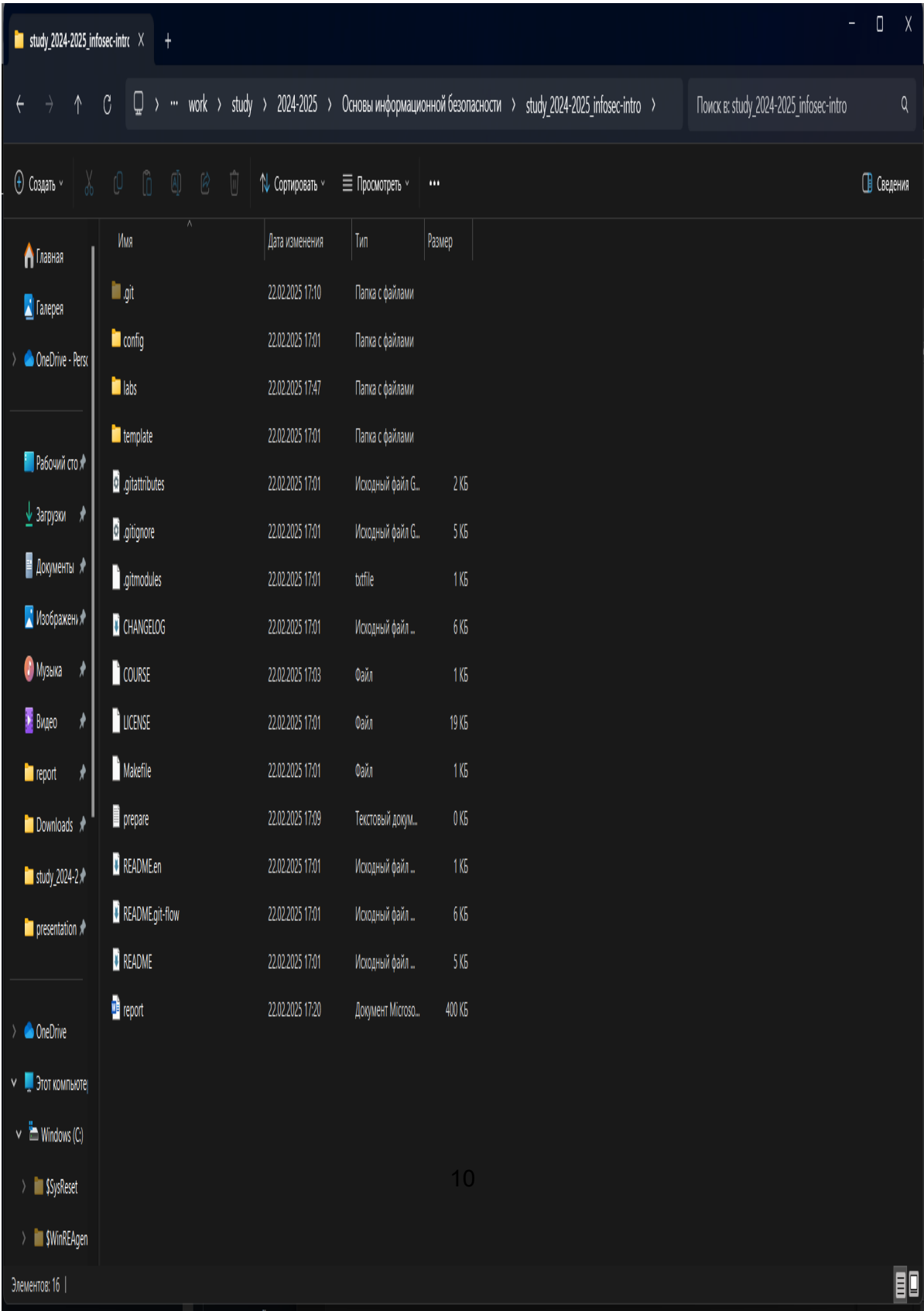


Рис. 3.1: (рис. Проверка рабочего пространства)



### 3.2 Установка и конфигурация операционной системы на виртуальную машину

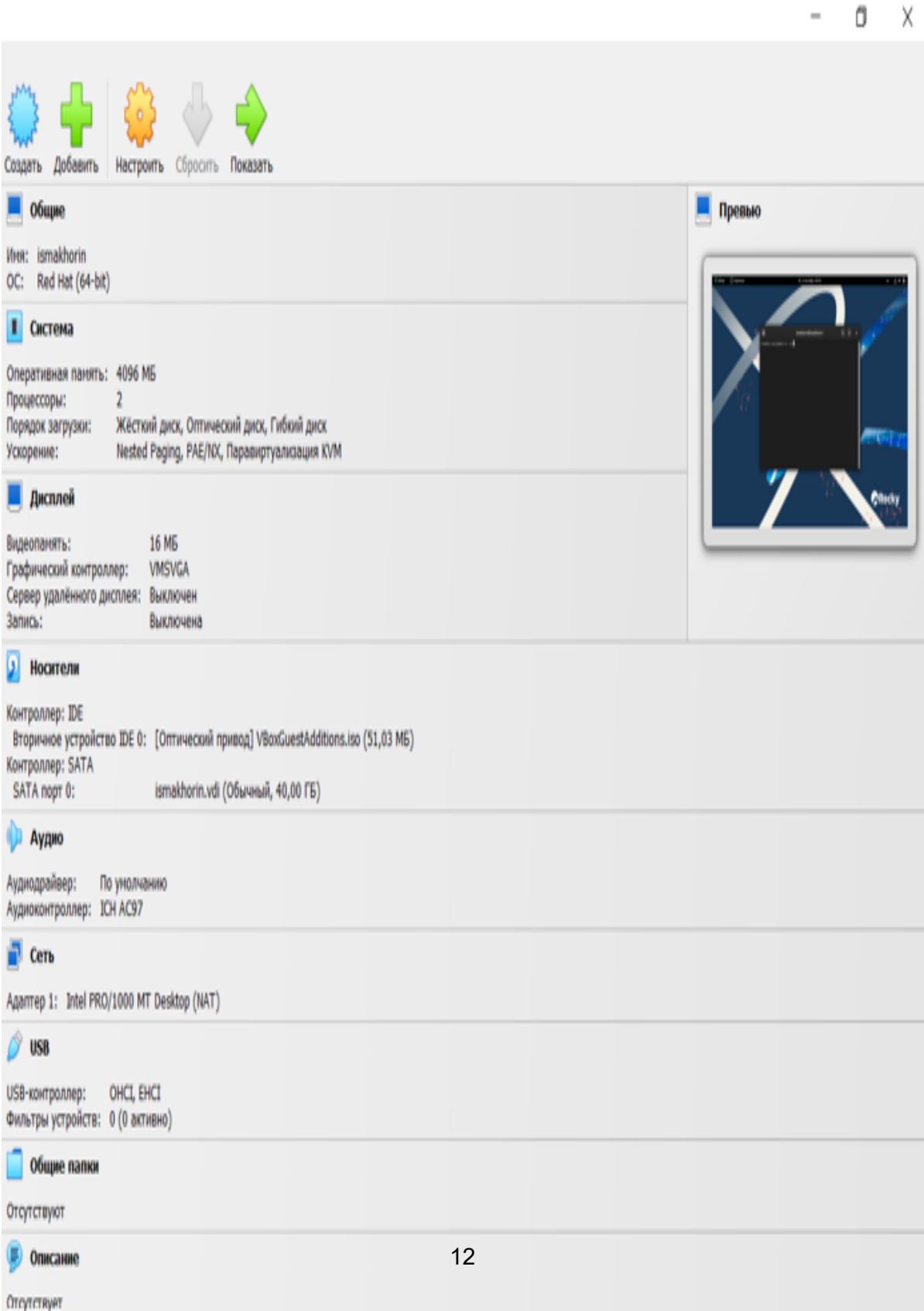


Рис. 3.2: (рис. Общие настройки)

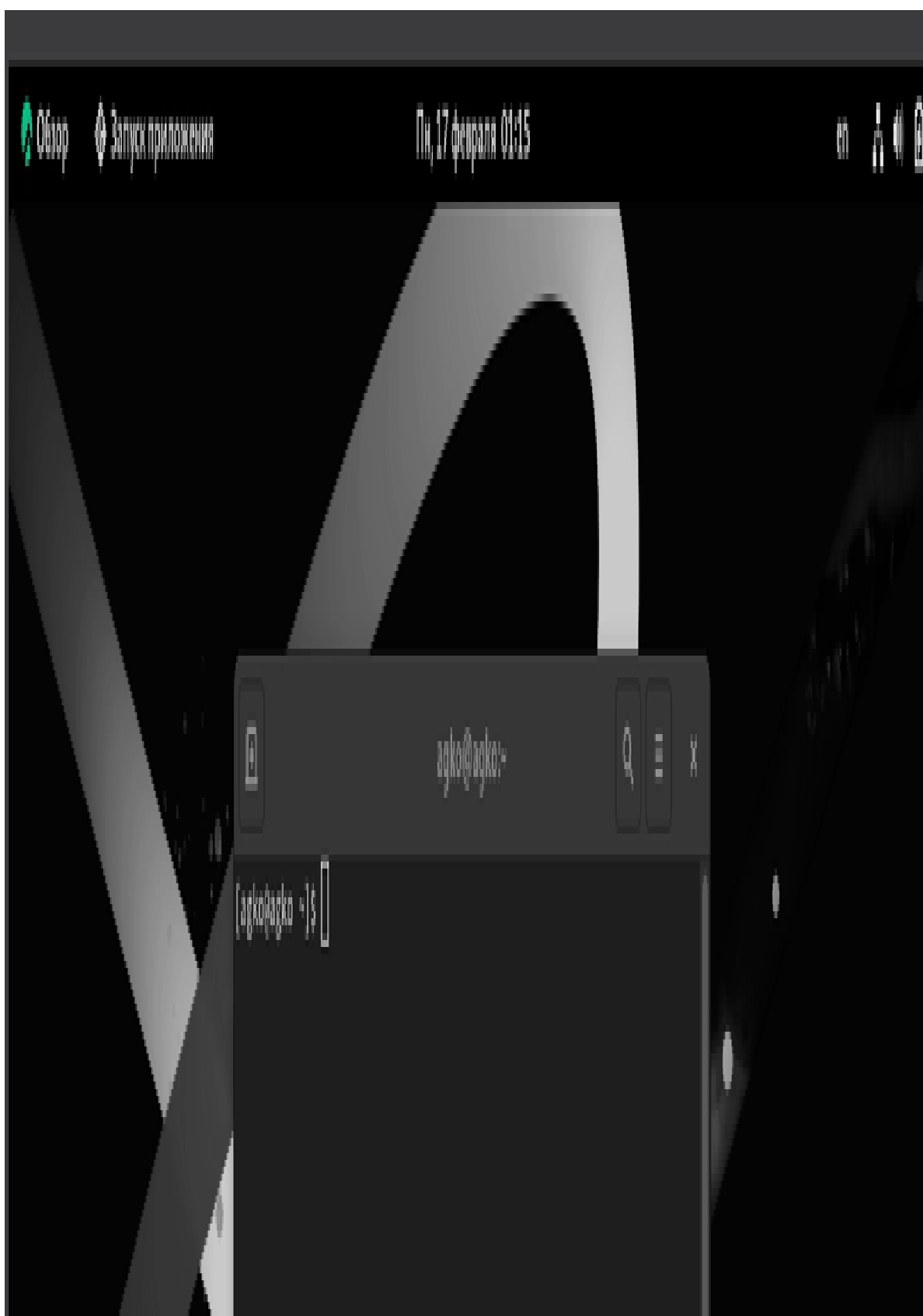


Рис. 3.3: (рис. Запуск системы)



### 3.3 Домашнее задание

```
аркобазко ~$ mesg | grep -i "Linux Version"
@.00000000 Linux version 5.14.0-503.14.1.el9_5.x86_64 (mockbuild@ad1-prod-b
170001.blk.eui.rockylinux.org) (gcc (GCC) 11.5.0 20240719 (Red Hat 11.5.0-2), G
nu ld version 2.35.2-54.el9) #1 SMP PREEMPT_DYNAMIC Fri Nov 15 12:04:32 UTC 2024
аркобазко ~$ mesg | grep -i "De
```

```

[agko@agko ~]$ dmesg | grep -i "Mhz"
[ 0.000017] tsc: Detected 2419.202 MHz processor
[ 2.527835] e1000 0000:00:03:0 eth0: (PCI:33MHz:32-bit) 08:00:27:f4:36:65
[agko@agko ~]$ free -m

```

	total	used	free	shared	buff/cache	available
Mem:	3914	1144	2484	21	524	2769
Swap:	4043	0	4043			

```

[agko@agko ~]$

```

Рис. 3.5: (рис. Частота процессора (Detected Mhz processor).)



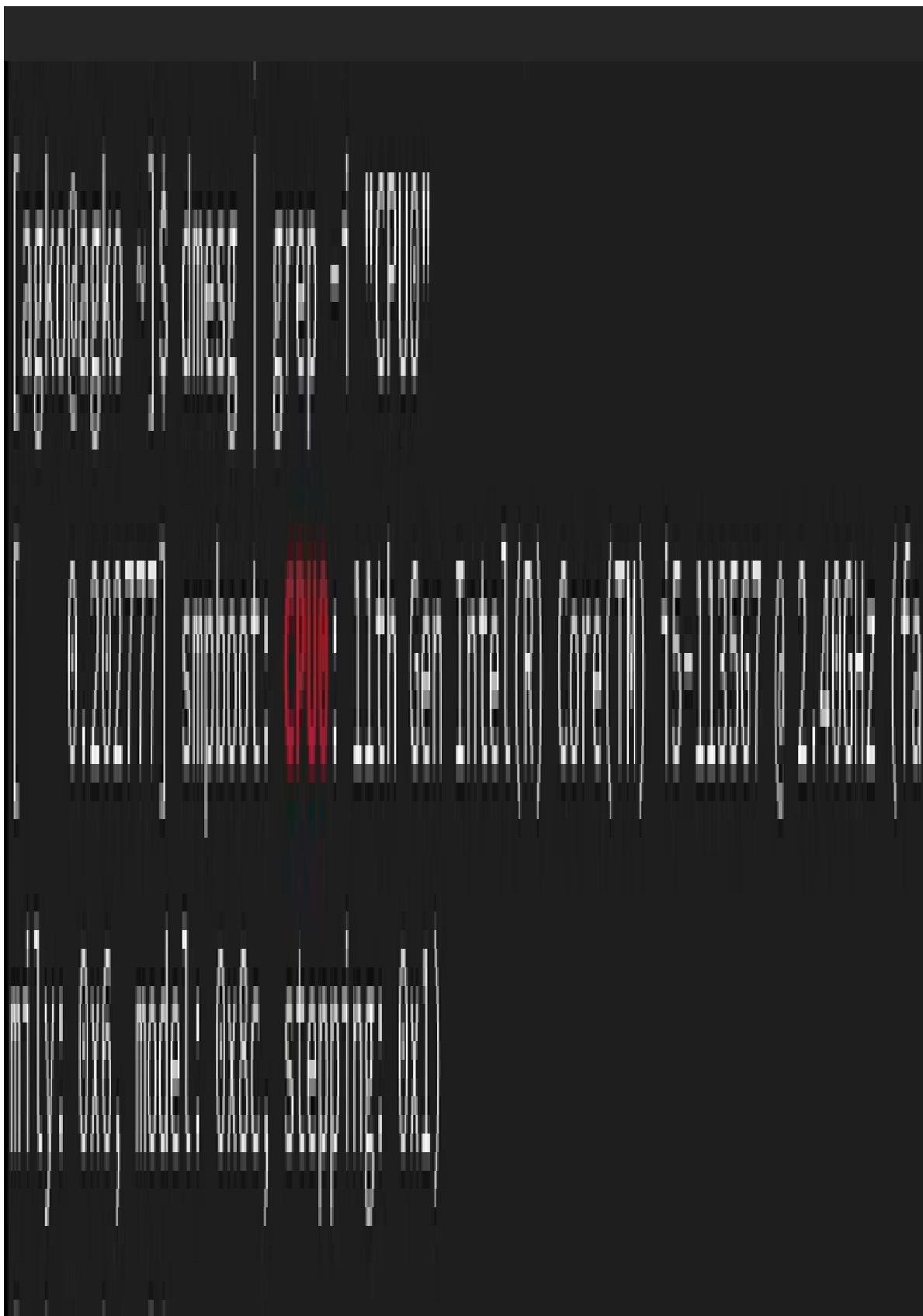


Рис. 3.6: (рис. Модель процессора (CPU0).)

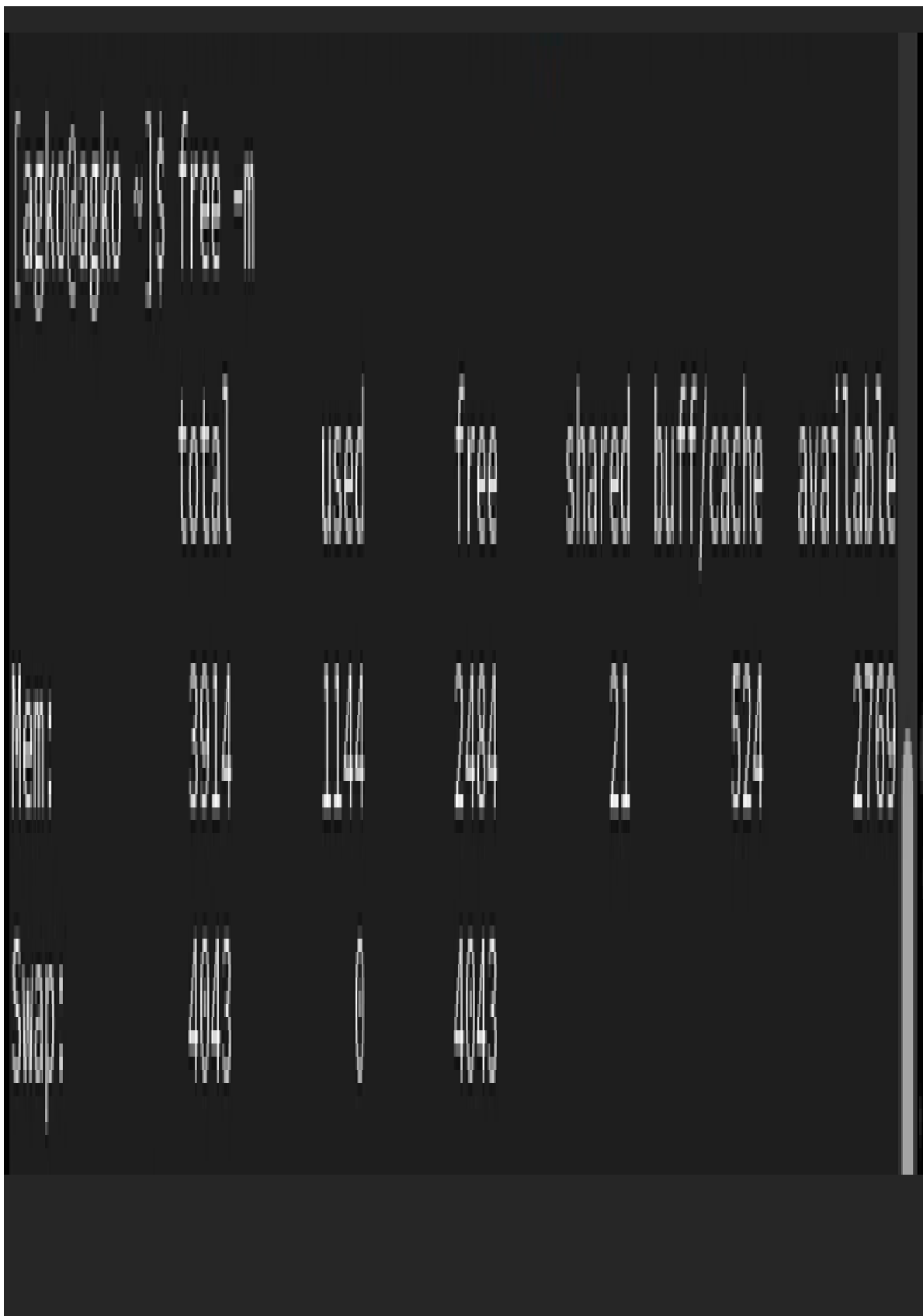


Рис. 3.7: (рис. Объем доступной оперативной памяти (Memory available).)

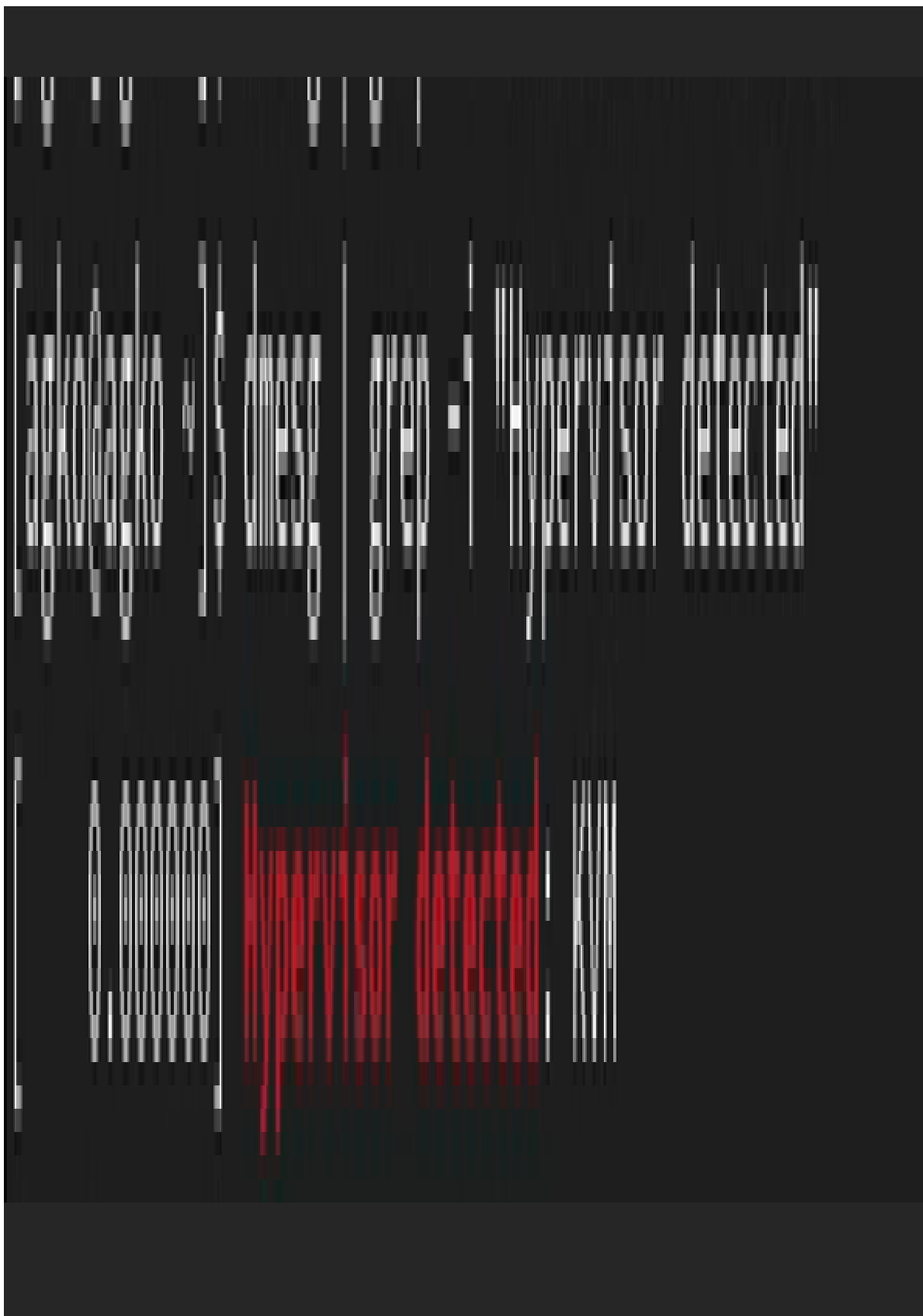


Рис. 3.8: (рис. Тип обнаруженного гипервизора (Hypervisor detected).)

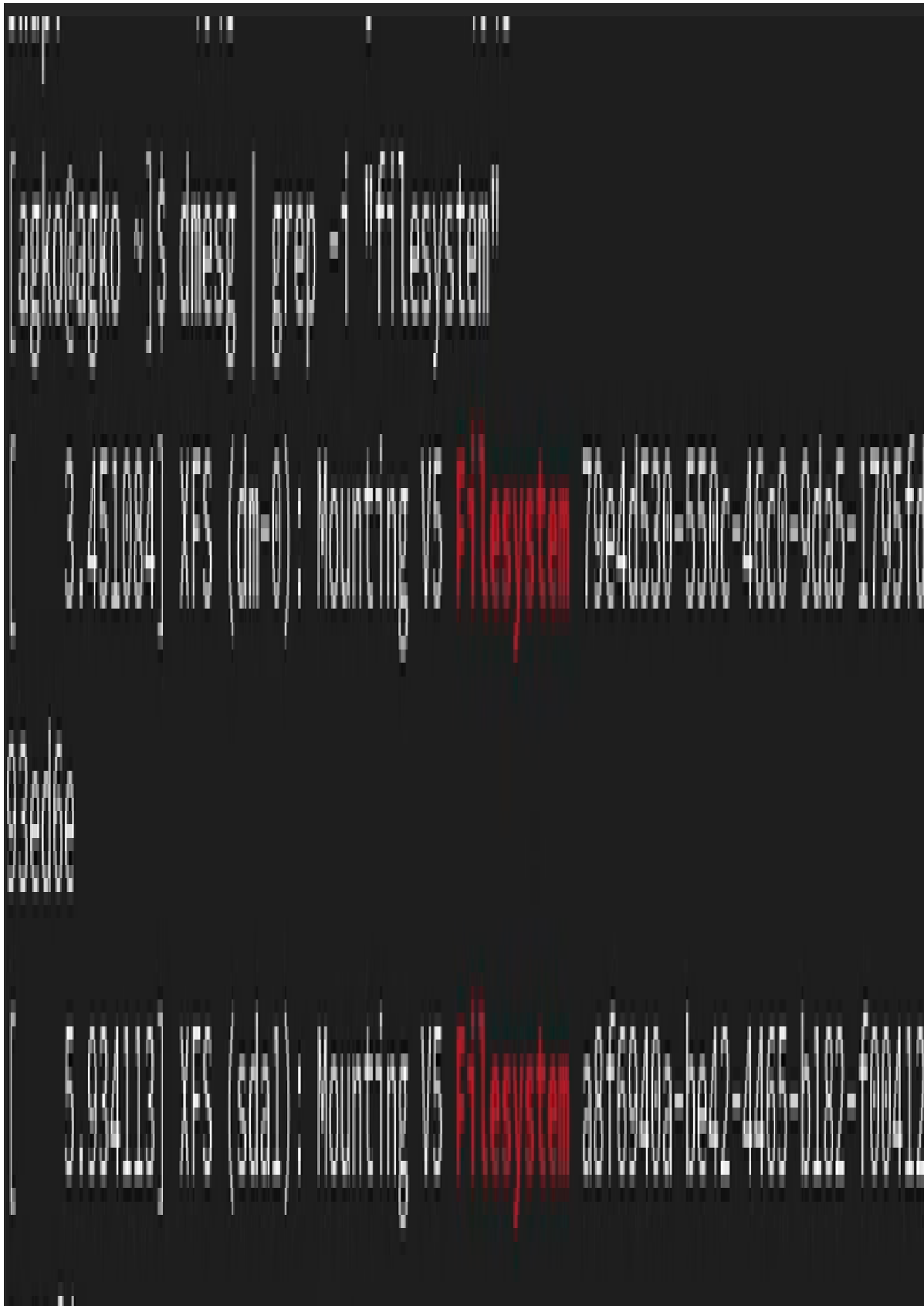
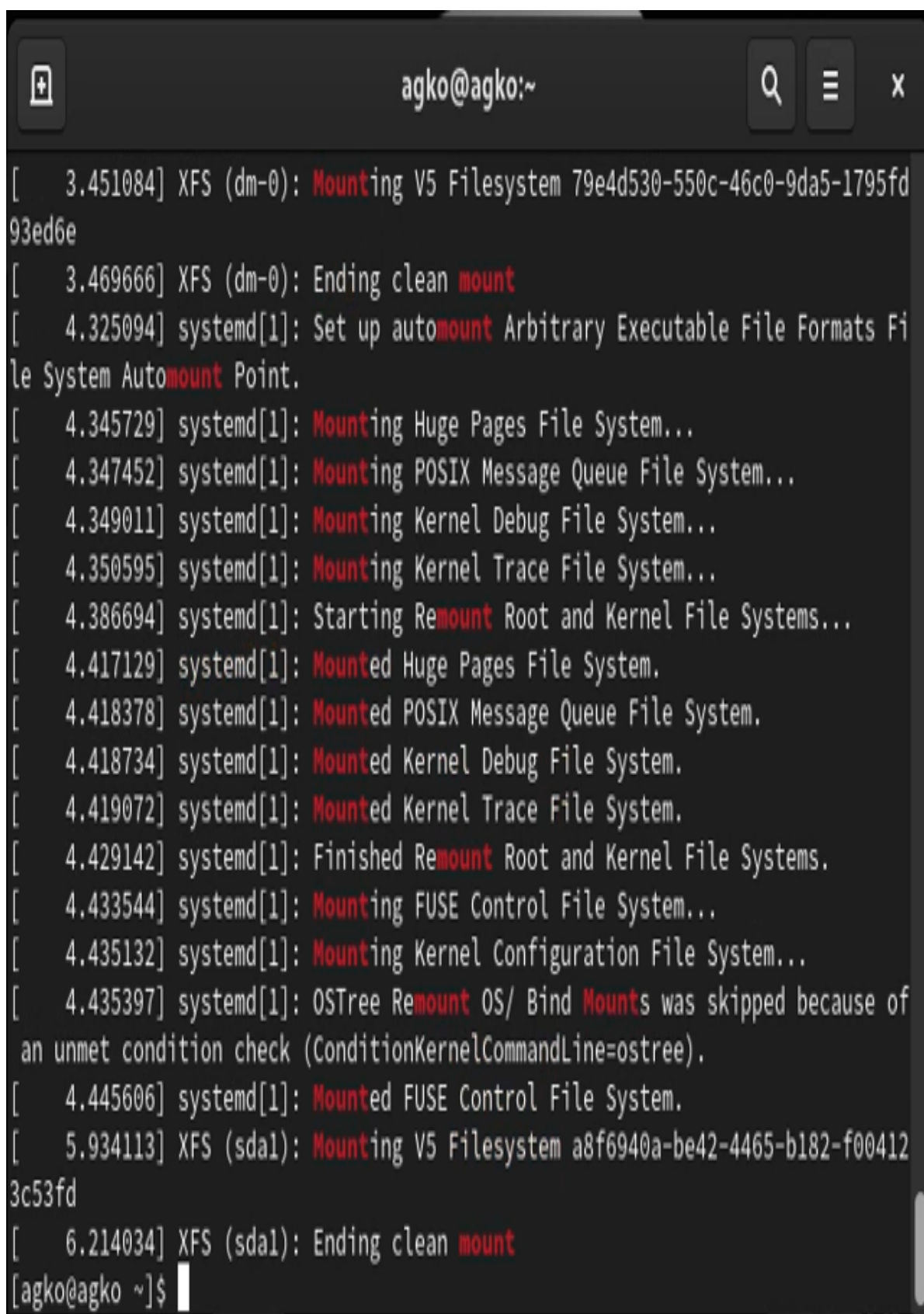


Рис. 3.9: (рис. Тип файловой системы корневого раздела.)



The image shows a terminal window with a dark background. The title bar at the top contains a window icon, the text 'agko@agko:~', and standard window controls (search, menu, close). The terminal displays a series of log messages from the systemd service, indicating the mounting of various file systems. The messages are timestamped and include the name of the file system and the action being performed. The sequence of events includes mounting XFS on dm-0, setting up automount for arbitrary executable file formats, mounting huge pages, POSIX message queue, kernel debug, and kernel trace file systems, remounting root and kernel file systems, mounting FUSE control and kernel configuration file systems, skipping OSTree remount, and finally mounting XFS on sda1. The terminal ends with a prompt '[agko@agko ~]\$'.

```
[ 3.451084] XFS (dm-0): Mounting V5 Filesystem 79e4d530-550c-46c0-9da5-1795fd93ed6e
[ 3.469666] XFS (dm-0): Ending clean mount
[ 4.325094] systemd[1]: Set up automount Arbitrary Executable File Formats File System Automount Point.
[ 4.345729] systemd[1]: Mounting Huge Pages File System...
[ 4.347452] systemd[1]: Mounting POSIX Message Queue File System...
[ 4.349011] systemd[1]: Mounting Kernel Debug File System...
[ 4.350595] systemd[1]: Mounting Kernel Trace File System...
[ 4.386694] systemd[1]: Starting Remount Root and Kernel File Systems...
[ 4.417129] systemd[1]: Mounted Huge Pages File System.
[ 4.418378] systemd[1]: Mounted POSIX Message Queue File System.
[ 4.418734] systemd[1]: Mounted Kernel Debug File System.
[ 4.419072] systemd[1]: Mounted Kernel Trace File System.
[ 4.429142] systemd[1]: Finished Remount Root and Kernel File Systems.
[ 4.433544] systemd[1]: Mounting FUSE Control File System...
[ 4.435132] systemd[1]: Mounting Kernel Configuration File System...
[ 4.435397] systemd[1]: OSTree Remount OS/ Bind Mounts was skipped because of an unmet condition check (ConditionKernelCommandLine=ostree).
[ 4.445606] systemd[1]: Mounted FUSE Control File System.
[ 5.934113] XFS (sda1): Mounting V5 Filesystem a8f6940a-be42-4465-b182-f004123c53fd
[ 6.214034] XFS (sda1): Ending clean mount
[agko@agko ~]$
```

Рис. 3.10: (рис. Последовательность монтирования файловых систем.)

---

## 3.4 Ответы на контрольные вопросы

### 3.4.1 1. Что содержит информация об учетной записи пользователя?

- Идентификатор учетной записи пользователя и ее имя
- Идентификатор основной группы пользователя и ее название

### 3.4.2 2. Основные команды Linux

- **Справка по команде:**  
info “название команды”  
“название команды” –help
- **Перемещение по файловой системе:**  
cd “путь”
- **Просмотр содержимого каталога:**  
ls или dir
- **Определение объема каталога:**  
du -sh “путь”
- **Создание каталога:**  
mkdir “название”
- **Удаление каталога:**

`rmdir "название"`

- **Создание файла:**

`touch "название"`

`cat > "название"`

- **Удаление файла:**

`rm "название"`

- **Изменение прав доступа:**

`chmod "права" "файл"`

- **Просмотр истории команд:**

`history`

### 3.4.3 3. Файловые системы

	Максимальный размер	Максимальное число	Максимальный размер
Название	файла	файлов	тома
NTFS	2 <sup>64</sup> байт	2 <sup>32</sup> - 1	256 ТБ
EXT4	2 <sup>44</sup> байт	2 <sup>32</sup> - 1	1048576 ТБ

### 3.4.4 4. Команда для проверки файловой системы

`dmesg | grep "filesystem"`

### 3.4.5 5. Команда для завершения процесса

`pkill "название процесса"`

## 4 Вывод

Было настроено рабочее пространство для лабораторных работ, приобретены практические навыки установки операционной системы на виртуальную машину и настройки минимально необходимых для дальнейшей работы сервисов. Были изучены идеология и применение средств контроля версий, освоены умения по работе с git. Были приобретены практические навыки оформления отчётов с помощью легковесного языка разметки Markdown.



## 5 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>