

# **Отчёт по лабораторной работе №1**

## **Информационная безопасность**

**Настройка рабочего пространства и конфигурация операционной системы на виртуальную машину. Система контроля версий Git. Язык разметки Markdown**

**Выполнил: Махорин Иван Сергеевич,  
НПИбд-02-21, 1032211221**

# Содержание

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Цель работы</b>   | <b>4</b>  |
| <b>2</b> | <b>Теоретическое введение</b>  | <b>5</b>  |
| <b>3</b> | <b>Выполнение лабораторной работы</b>  | <b>8</b>  |
| 3.1      | Настройка рабочего пространства . . . . .  | 8         |
| 3.2      | Установка и конфигурация операционной системы на виртуальную<br>машину . . . . . | 9         |
| <b>4</b> | <b>Вывод</b>   | <b>11</b> |
| <b>5</b> | <b>Список литературы. Библиография</b>   | <b>12</b> |

## Список иллюстраций

|     |  |    |
|-----|--|----|
| 3.1 | (рис. 1. Проверка рабочего пространства) . . . . . | 8  |
| 3.2 | (рис. 2. Общие настройки) . . . . .                | 9  |
| 3.3 | (рис. 3. Запуск системы) . . . . .                 | 10 |

# 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 разре-

шения 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<br>команды | Описание команды               |
|---------------------|--------------------------------|
| git clone           | Клонирование репозитория на ПК |

| Назва-<br>ние<br>коман-<br>ды              | Описание команды          |
|--|---------------------------|
| git<br>commit<br>-m<br>"Initial<br>Commit" | Оставление коммита        |
| git<br>push                                | Загрузка изменений на гит |
| 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

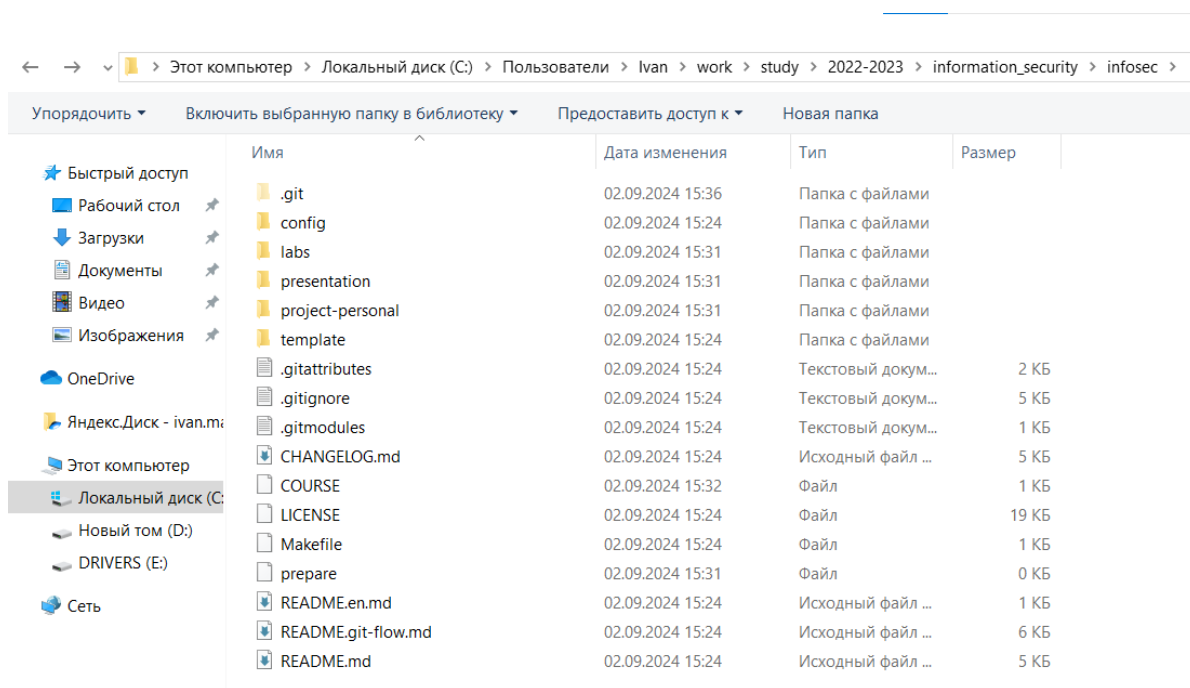


Рис. 3.1: (рис. 1. Проверка рабочего пространства)



## 3.2 Установка и конфигурация операционной системы на виртуальную машину

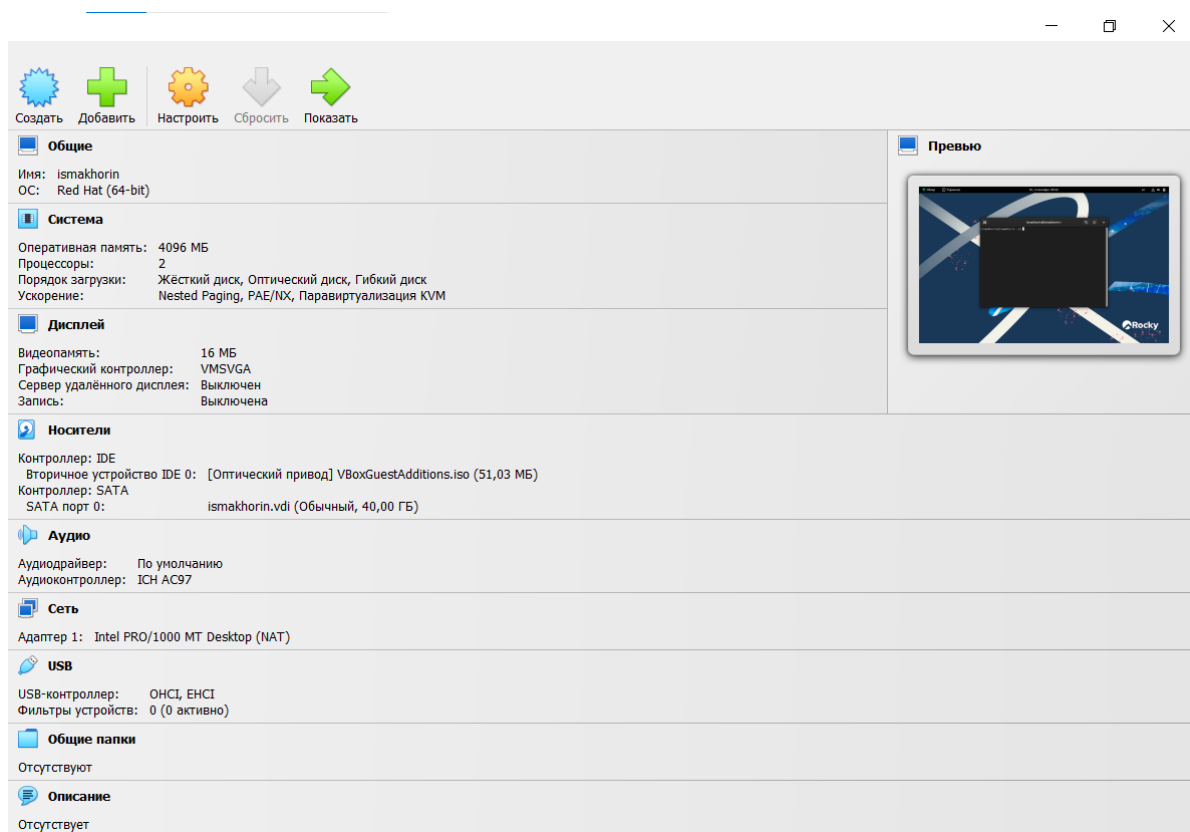


Рис. 3.2: (рис. 2. Общие настройки)

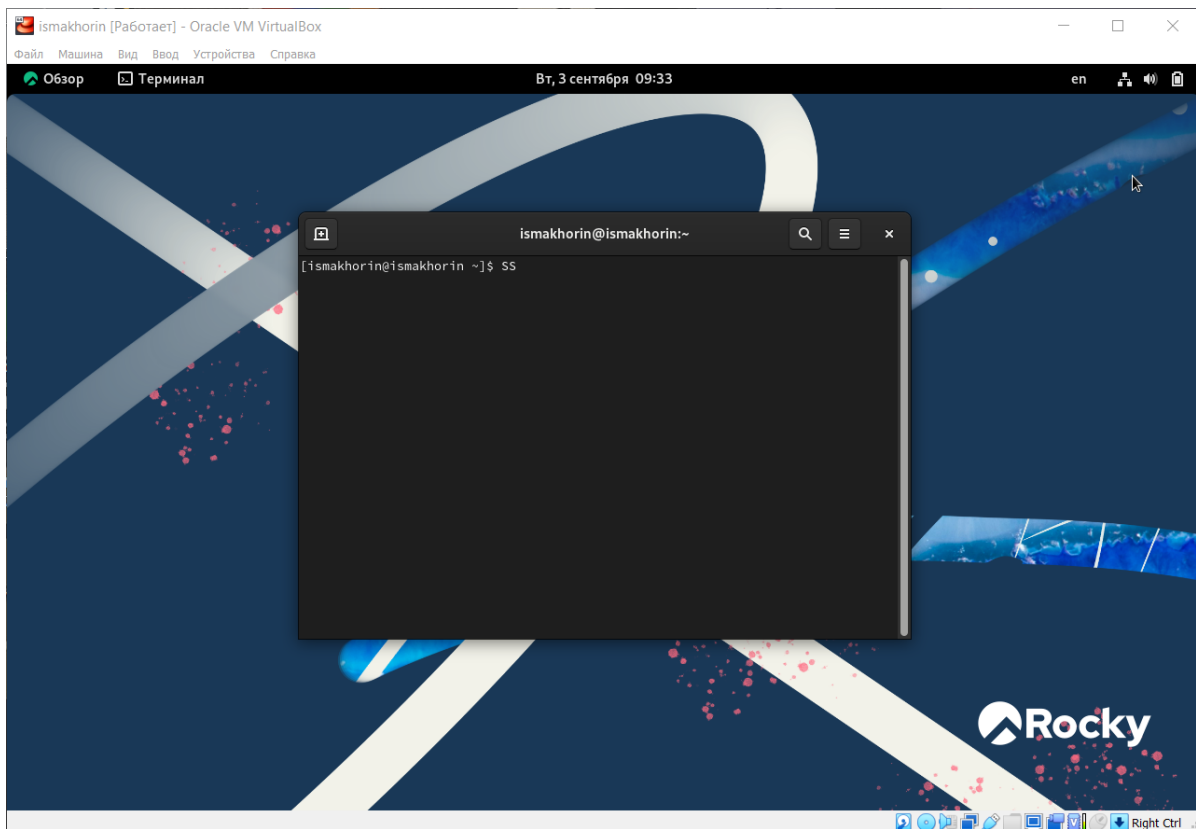


Рис. 3.3: (рис. 3. Запуск системы)

## 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>