

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

Лабораторная работа 3.

РИДОЙ ХАЙАТУННАБИ

ГРУППА: НКАБД-05-2023

Содержание

1Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

2Задание

- 1.Сделайте отчёт по предыдущей лабораторной работе в формате Markdown.
- 2.В качестве отчёта просьба предоставить отчёты в 3 форматах: pdf, docx и md (в архиве, поскольку он должен содержать скриншоты, Makefile и т.д.)

3Теоретическое введение

Базовые сведения о Markdown

Чтобы создать заголовок, используйте знак (#), например:

1. # This is heading 1
2. ## This is heading 2
3. ### This is heading 3
4. #### This is heading 4

Чтобы задать для текста полужирное начертание, заключите его в двойные звездочки:

- 1.This text is **bold**

Чтобы задать для текста курсивное начертание, заключите его в одинарные звездочки:

- 1.This text is *italic*.

Чтобы задать для текста полужирное и курсивное начертание, заключите его в тройные звездочки:

- 1.This is text is both ***bold and italic***.

Блоки цитирования создаются с помощью символа >:

The drought had lasted now for ten million years, and the reign of the terrible lizards had long since ended. Here on the Equator, in the continent which would one day be known as Africa, the battle for existence had reached a new climax of ferocity, and the victor was not yet in sight. In this barren and desiccated land, only the small or the swift or the fierce could flourish, or even hope to survive.

Неупорядоченный (маркированный) список можно отформатировать с помощью звездочек или тире:

- List item 1

- List item 2

- List item 3

Упорядоченный список можно отформатировать с помощью соответствующих цифр:

1.First instruction

1.Sub-instruction

2.Sub-instruction

2.Second instruction

Обработка файлов в формате Markdown

Для обработки файлов в формате Markdown будем использовать Pandoc

<https://pandoc.org/>. Конкретно, нам понадобится программа pandoc , pandoc-citeproc

<https://github.com/jgm/pandoc/releases>, pandoc-crossref

<https://github.com/lierdakil/pandoc-crossref/releases>. Преобразовать файл README.md можно следующим образом:

1.pandoc README.md -o README.pdf

или

2.pandoc README.md -o README.docx

Оформление отчета по лабораторной работе

Лабораторная работа является небольшой научно-исследовательской работой, которую и оформлять следует по всем утверждённым требованиям. При подготовке отчета по лабораторной работе вы освоите ряд важных элементов, которые в дальнейшем пригодятся вам при написании курсовой и дипломной работы.

Структура отчёта

Согласно ГОСТ 7.32-2001, любая научно-исследовательская работа должна обязательно содержать следующие элементы: - титульный лист; - реферат; - введение; - основную часть; - заключение. Также ГОСТ рекомендует включить в работу и такие элементы: - список исполнителей; - содержание; - нормативные ссылки; - определения; - обозначения и сокращения; - список использованных

источников; - приложения. Если вы проводите сложную работу, выполняемую в несколько этапов, то вам может понадобиться включить в работу часть или все элементы второго списка.

4Выполнение лабораторной работы

После выполнения лабораторной работы №2 я приступил к созданию отчёта с помощью Markdown. И на первом этапе создания отчёта я произвёл редактирование титульного листа в соответствии с шаблоном (рис. [1]).

Figure 1:

```
## Generic options
lang: ru-RU
toc-title: "Содержание"

## Bibliography
bibliography: bib/cite.bib
csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl

## Pdf output format
toc: true # Table of contents
toc-depth: 2
lof: true # List of figures
lot: true # List of tables
fontsize: 12pt
linestretch: 1.5
papersize: a4
documentclass: scrreprt

## I18n polyglossia
polyglossia-lang:
  name: russian
  options:
    - spelling=modern
    - babelshorthands=true
polyglossia-otherlangs:
  name: english

## I18n babel
babel-lang: russian
babel-otherlangs: english

## Fonts
mainfont: PT Serif
romanfont: PT Serif
sansfont: PT Sans
monofont: PT Mono
mainfontoptions: ligatures=TeX
```

Figure 1: Редактирование титульного листа отчёта

Далее я заполнил пункты “Цель работы”, “Задание” и “Выполнение лабораторной работы” (рис. [2]).

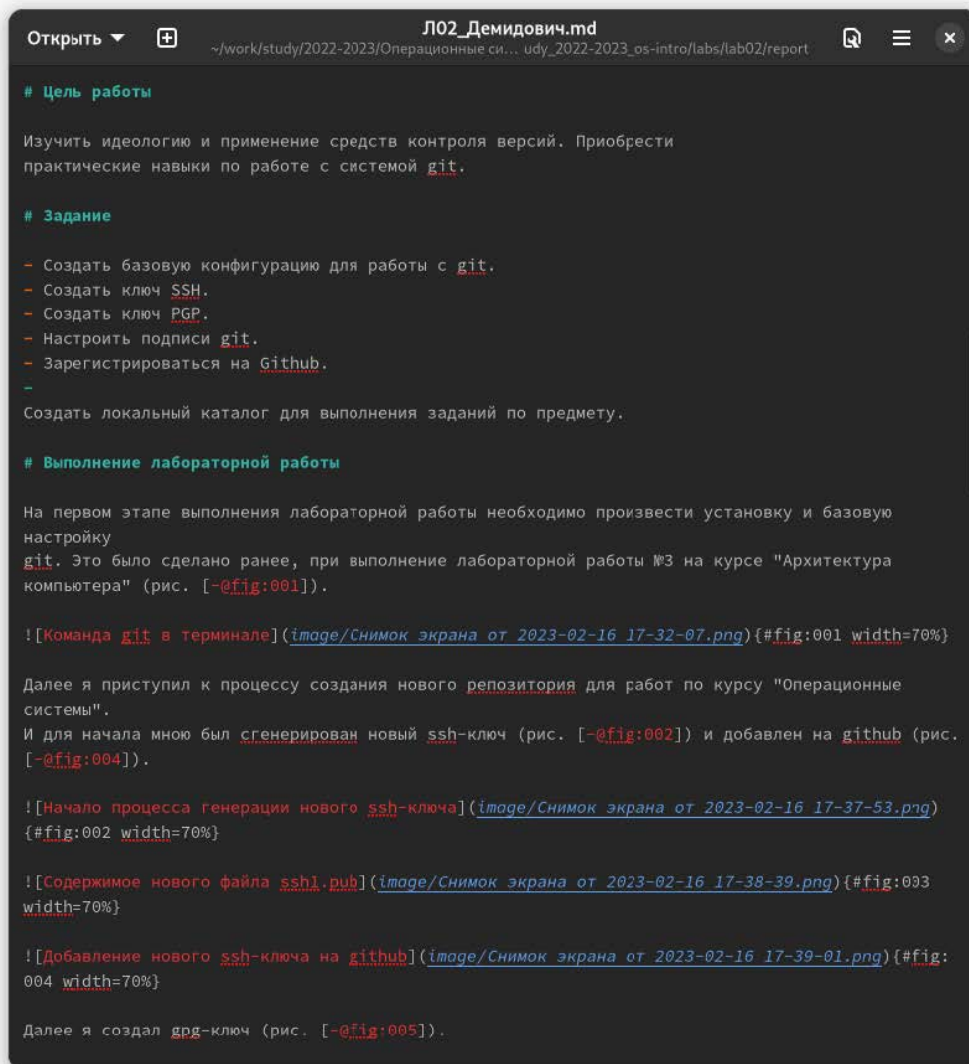


Figure 2: Редактирование пунктов “Цель работы”, “Задание” и “Выполнение лабораторной работы”

Затем я приступил к записи ответов на контрольные вопросы (рис. [3], [4]).

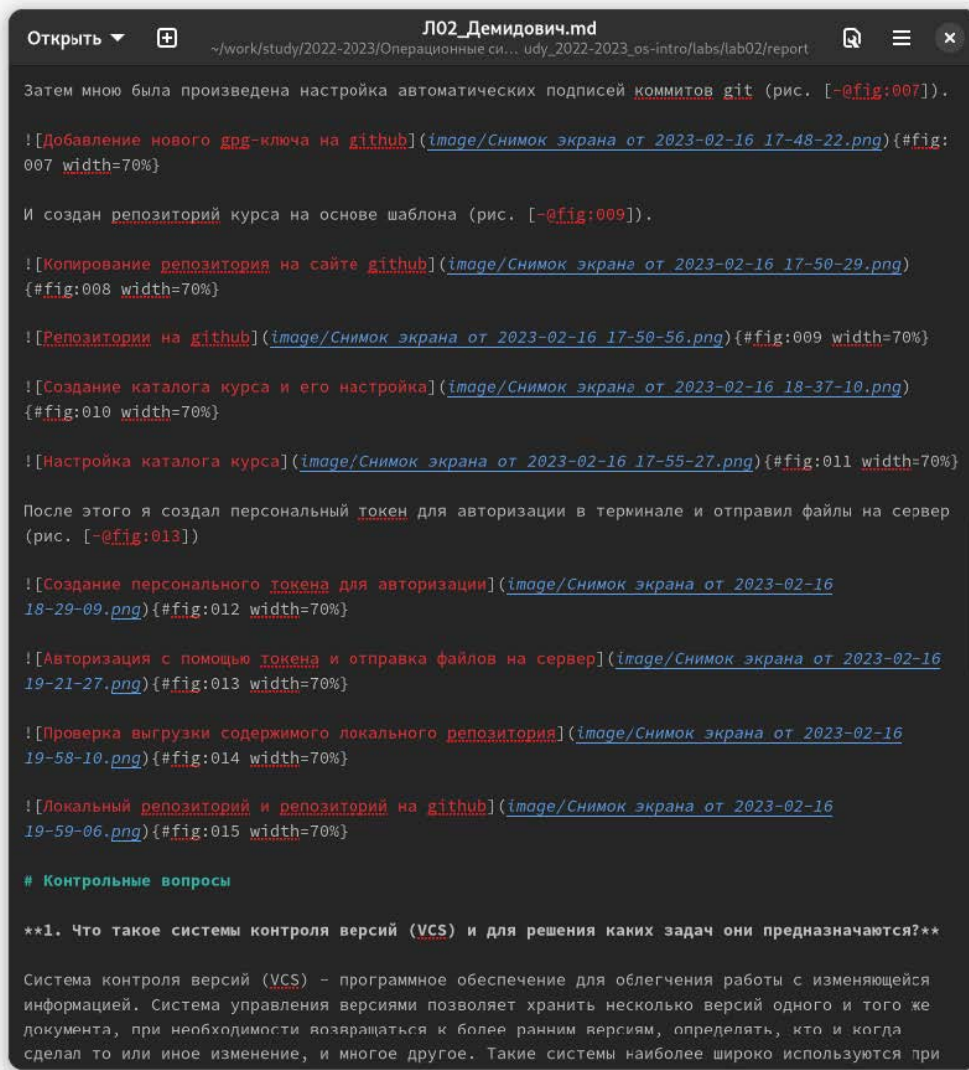
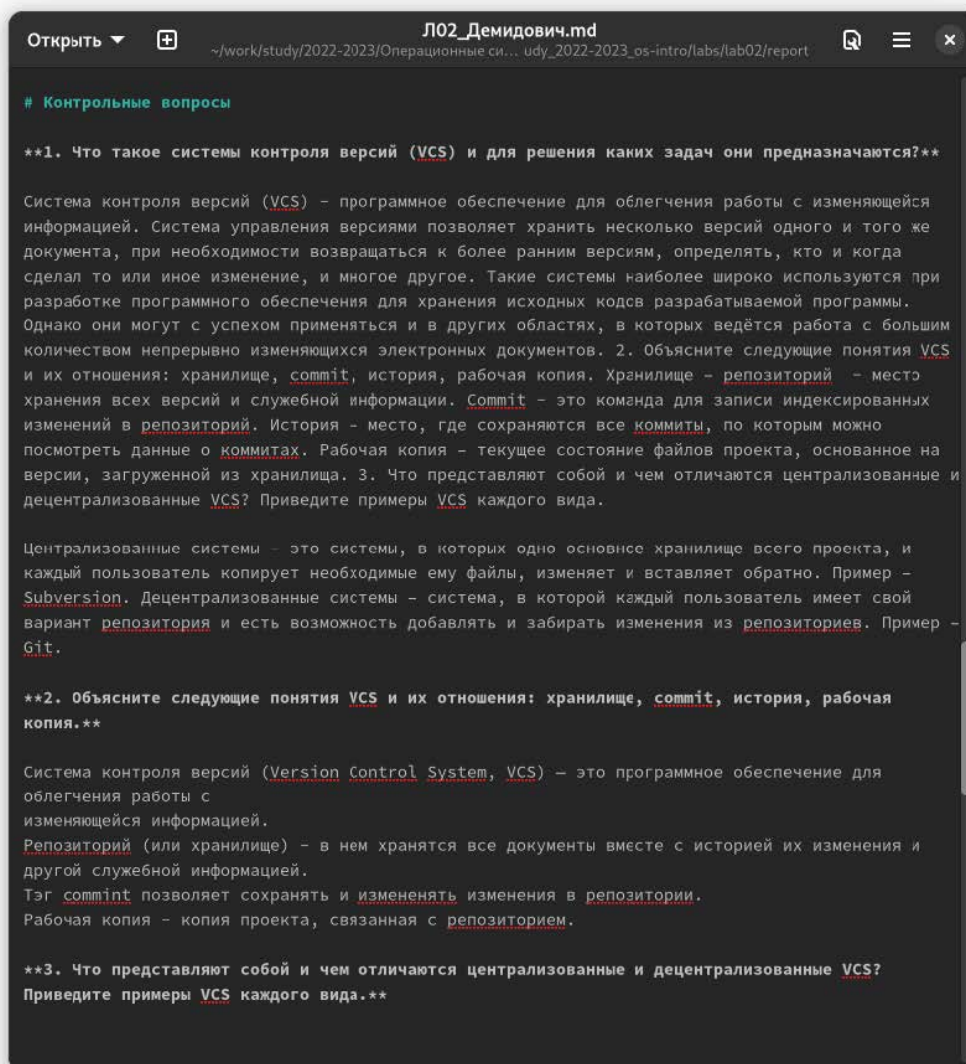


Figure 3: Редактирование пункта “Выполнение лабораторной работы” и начало редактирования ответов на контрольные вопросы



The image shows a screenshot of a code editor window with a dark theme. The title bar at the top reads "Л02_Демидович.md" and shows the file path "~\work\study\2022-2023\Операционные си... udy_2022-2023_os-intro\labs\lab02\report". The editor content is as follows:

```
# Контрольные вопросы

**1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначены?**

Система контроля версий (VCS) – программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое. Такие системы наиболее широко используются при разработке программного обеспечения для хранения исходных кодов разрабатываемой программы. Однако они могут с успехом применяться и в других областях, в которых ведётся работа с большим количеством непрерывно изменяющихся электронных документов.

2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия. Хранилище – репозиторий – место хранения всех версий и служебной информации. Commit – это команда для записи индексированных изменений в репозиторий. История – место, где сохраняются все коммиты, по которым можно посмотреть данные о коммитах. Рабочая копия – текущее состояние файлов проекта, основанное на версии, загруженной из хранилища.

3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

Централизованные системы – это системы, в которых одно основное хранилище всего проекта, и каждый пользователь копирует необходимые ему файлы, изменяет и вставляет обратно. Пример – Subversion. Децентрализованные системы – система, в которой каждый пользователь имеет свой вариант репозитория и есть возможность добавлять и забирать изменения из репозитория. Пример – Git.

**2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.**

Система контроля версий (Version Control System, VCS) – это программное обеспечение для облегчения работы с изменяющейся информацией.
Репозиторий (или хранилище) – в нем хранятся все документы вместе с историей их изменения и другой служебной информацией.
Тэг commit позволяет сохранять и изменять изменения в репозитории.
Рабочая копия – копия проекта, связанная с репозиторием.

**3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.**
```

Figure 4: Ответы на контрольные вопросы
И на заключающем этапе создания отчёта я записал выводы и указал источники, которые были использованы при выполнении лабораторной работы №2 (рис. [5]).


```
Открыть ▾ + Л02_Демидович.md
~/work/study/2022-2023/Операционные си... udy_2022-2023_os-intro/labs/lab02/report

- переключение на некоторую ветку: git checkout имя_ветки (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой) - отправка изменений конкретной ветки в центральный репозиторий: 1 git push origin имя_ветки - слияние ветки с текущим деревом: 1 git merge --no-ff имя_ветки - удаление локальной уже слитой с основным деревом ветки: git branch -d имя_ветки - принудительное удаление локальной ветки: git branch -D имя_ветки - удаление ветки с центрального репозитория: git push origin :имя_ветки

**8. Приведите примеры использования при работе с локальным и удалённым репозиториями.**

Работа с удалённым репозиторием: git remote - просмотр списка настроенных удалённых репозиториях.

Работа с локальным репозиторием: git status - выводит информацию обо всех изменениях, внесенных в дерево директорий проекта по сравнению с последним коммитом рабочей ветки.

**9. Что такое и зачем могут быть нужны ветки (branches)?**

Ветка (англ. branch) - это последовательность коммитов, в которой ведётся параллельная разработка какого-либо функционала. Ветки нужны, чтобы несколько программистов могли вести работу над одним и тем же проектом или даже файлом одновременно, при этом не мешая друг другу. Кроме того, ветки используются для тестирования экспериментальных функций: чтобы не повредить основному проекту, создается новая ветка специально для экспериментов.

**10. Как и зачем можно игнорировать некоторые файлы при commit?**

Игнорируемые файлы - это, как правило, артефакты сборки и файлы, генерируемые машиной из исходных файлов в вашем репозитории, либо файлы, которые по какой-либо иной причине не должны попадать в коммиты. В Git нет специальной команды для указания игнорируемых файлов: вместо этого необходимо вручную отредактировать файл .gitignore. Временно игнорировать изменения в файле можно командой git update-index --assume-unchanged.

# Выводы

В результате выполнения данной лабораторной работы я изучил идеологию и применение средств контроля версий, а также приобрел практические навыки по работе с системой git.

# Список литературы{.unnumbered}

Лабораторная работа №2 (Архитектура ОС).
```

Figure 5: Заполнение пунктов “Выводы” и “Список литературы”

5Выводы

В результате выполнения данной лабораторной работы я научился оформлять отчёты с помощью легковесного языка разметки Markdown.

Список литературы

Лабораторная работа №3 (Архитектура ОС).