

# **Лабораторная работа 3**

**Язык разметки Markdown**

Головина Мария Игоревна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>10</b>
<b>5</b>	<b>Выводы</b>	<b>15</b>
	<b>Список литературы</b>	<b>16</b>

# Список иллюстраций

4.1	Обновление локального репозитория . . . . .	10
4.2	Каталог с шаблоном отчета по лабораторной работе №3 . . . . .	11
4.3	Компиляция шаблона с использованием Makefile . . . . .	11
4.4	Удаление файлов . . . . .	12
4.5	Структура файла report.md . . . . .	12
4.6	Загрузка файлов на Github . . . . .	13
4.7	Отчёт по лабораторной работе № 2 . . . . .	13
4.8	Проверка файлов на github . . . . .	14

## **Список таблиц**

# 1 Цель работы

Освоить процедуры оформления отчетов с помощью легковесного языка разметки Markdown.

## 2 Задание

1. Открыть терминал. Перейти в каталог курса сформированный при выполнении лабораторной работы №2 . Обновить локальный репозиторий, скачав изменения из удаленного репозитория.
2. Перейти в каталог с шаблоном отчета по лабораторной работе №3.
3. Провести компиляцию шаблона с использованием Makefile. Открыть и проверить успешность компиляции.
4. Удалить полученные файлы с использованием Makefile. Проверить, что после этой команды файлы report.pdf и report.docx были удалены.
5. Открыть файл report.md с помощью текстового редактора gedit gedit report.md. Изучить структуру этого файла.
6. Заполнить отчет, скомпилировать его с использованием Makefile. Проверить корректность полученных файлов. Загрузить файлы на Github.

Задание для самостоятельной работы 1. В соответствующем каталоге сделать отчёт по лабораторной работе № 2 в формате Markdown. В качестве отчёта представить отчёты в 3 форматах: pdf, docx и md. Загрузить файлы на github.

## 3 Теоретическое введение

Базовые сведения о Markdown Чтобы создать заголовок, используйте знак #, например: # This is heading 1 ## This is heading 2 ### This is heading 3 #### This is heading 4

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

This text is **bold**.

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

This text is *italic*.

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

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.

Чтобы вложить один список в другой, добавьте отступ для элементов дочернего списка:

1. First instruction
2. Second instruction
3. Third instruction

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

- List item 1
- List item 2
- List item 3

Чтобы вложить один список в другой, добавьте отступ для элементов дочернего списка:

1. List item 1
  1. List item A
  2. List item B
2. List item 2

Синтаксис Markdown для встроенной ссылки состоит из части [link text], представляющей текст гиперссылки, и части (file-name.md) – URL-адреса или имени файла, на который дается ссылка:

link text

или

link text

Оформление изображений в Markdown

В Markdown вставить изображение в документ можно с помощью непосредственного указания адреса изображения. Синтаксис данной команды выглядит следующим образом:



[Подпись к рисунку] (/путь/к/изображению.jpg “Необязательная подсказка”){#fig:fig1 width=70% }

Здесь:

в квадратных скобках указывается подпись к изображению;

в круглых скобках указывается URL-адрес или относительный путь изображения, а так- же (необязательно) всплывающую подсказку, заключённую в двойные или одиночные кавычки;

в фигурных скобках указывается идентификатор изображения (#fig:fig1) для ссылки на него по тексту и размер изображения относительно ширины страницы (width=90%).

Обработка файлов в формате Markdown

Преобразовать файл README.md можно следующим образом:

```
pandoc README.md -o README.pdf
```

или так

```
pandoc README.md -o README.docx
```

## 4 Выполнение лабораторной работы

1. Открыли терминал. Перешли в каталог курса сформированный при выполнении лабораторной работы №2 . Обновили локальный репозиторий, скачав изменения из удаленного репозитория (рис. 4.1 Обновление локального репозитория).

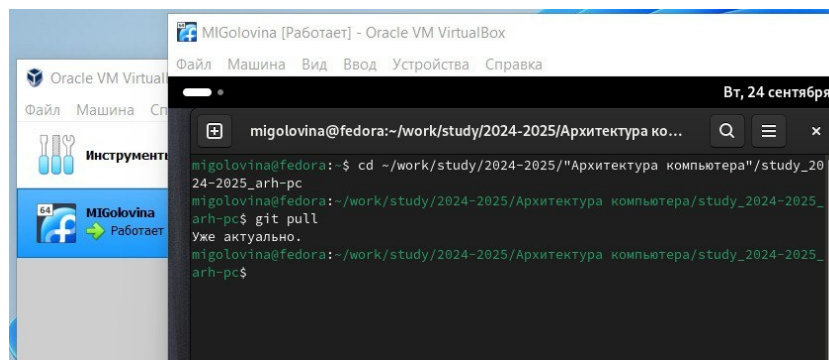


Рис. 4.1: Обновление локального репозитория

2. Перешли в каталог с шаблоном отчета по лабораторной работе №3 (рис. 4.2 Каталог с шаблоном отчета по лабораторной работе №3).

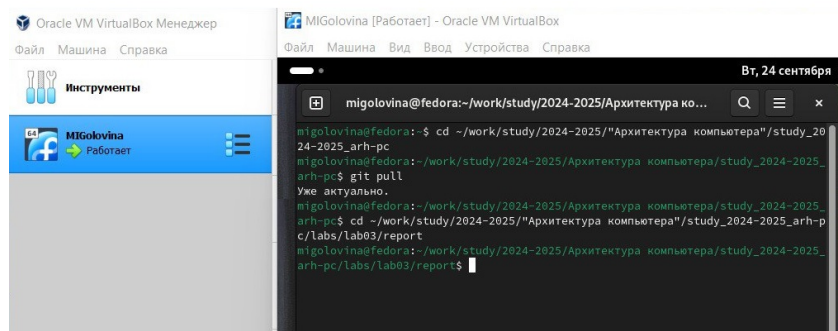


Рис. 4.2: Каталог с шаблоном отчета по лабораторной работе №3

3. Провели компиляцию шаблона с использованием Makefile. Открыли и проверили успешность компиляции. Были сгенерированы файлы report.pdf и report.docx (рис. 4.3 Компиляция шаблона с использованием Makefile).

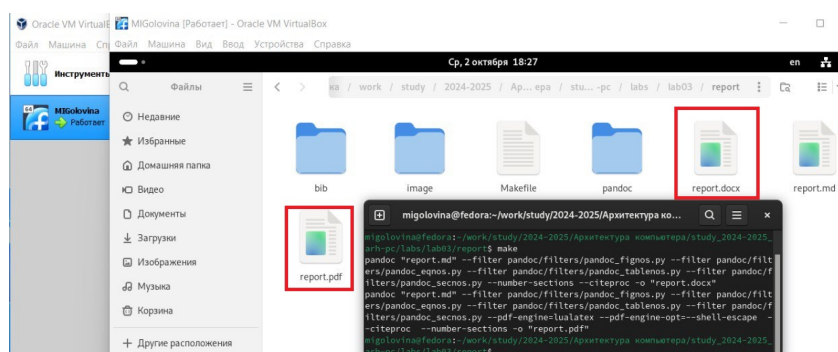


Рис. 4.3: Компиляция шаблона с использованием Makefile

4. Удалили полученные файлы с использованием Makefile. Проверили, что после этой команды файлы report.pdf и report.docx были удалены (рис. 4.4 Удаление файлов).

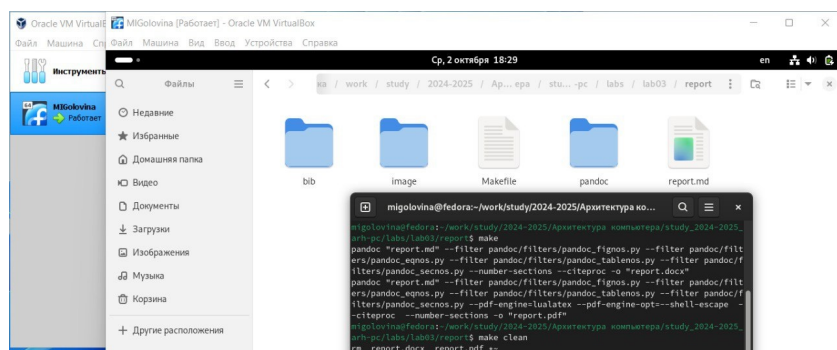


Рис. 4.4: Удаление файлов

5.Открыли файл report.md с помощью текстового редактора gedit gedit report.md. Изучили структуру этого файла (рис. 4.5 Структура файла report.md).

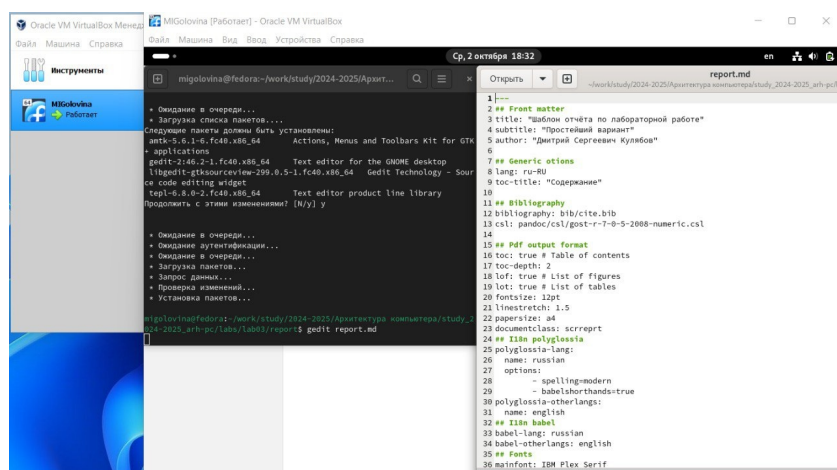


Рис. 4.5: Структура файла report.md

6.Заполнили отчет, скомпилировали его с использованием Makefile. Проверите корректность полученных файлов. Загрузили файлы на Github. (рис. 4.6 Загрузка файлов на Github).

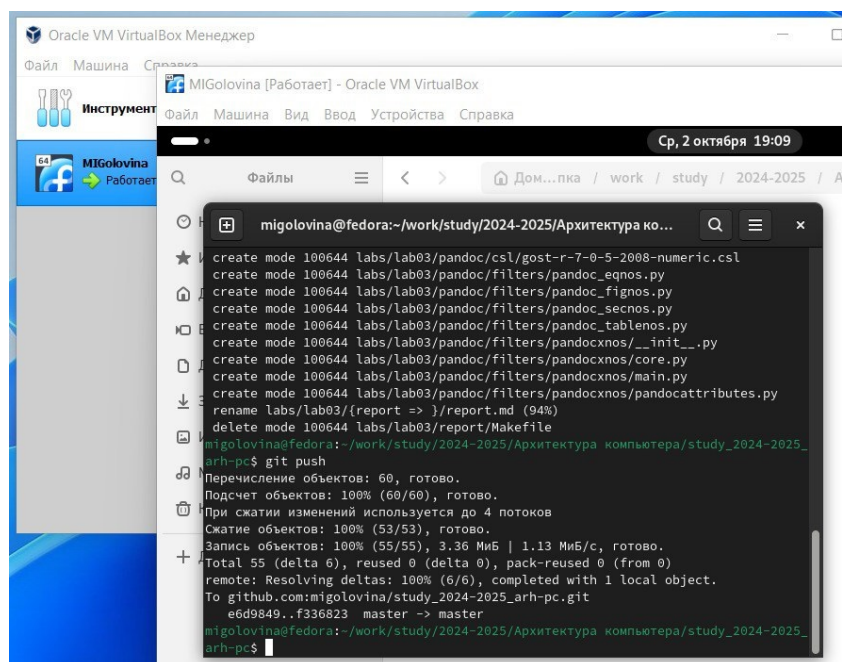


Рис. 4.6: Загрузка файлов на Github

#### Задание для самостоятельной работы

В соответствующем каталоге сделали отчёт по лабораторной работе № 2 в формате Markdown. В качестве отчёта предоставили отчёты в 3 форматах: pdf, docx и md (рис. 4.7 Отчёт по лабораторной работе № 2).

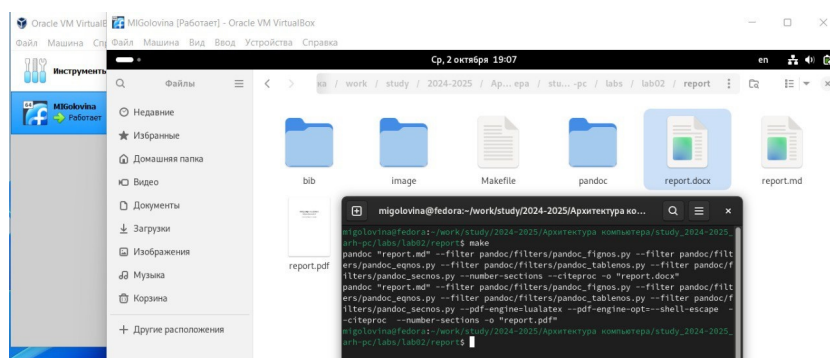


Рис. 4.7: Отчёт по лабораторной работе № 2

Загрузили файлы на github (рис. 4.8 Проверка файлов на github).



## 5 Выводы

Освоили процедуры оформления отчетов с помощью легковесного языка разметки Markdown.

## Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: [http://www.stolyarov.info/books/asm\\_unix](http://www.stolyarov.info/books/asm_unix).



15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).