

# **Лабораторная работа 3**

**Оформление отчета в markdown**

Майзингер Элина Сергеевна

# 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>8</b>
4.1	Использование заголовков разного уровня . . . . .	8
4.2	Прикрепляем картиночки ( с ссылочками!) . . . . .	8
4.3	Используем нумерацию . . . . .	9
4.4	Описываем ход работы . . . . .	9
4.5	Форматируем отчет и презентацию в терминале . . . . .	11
<b>5</b>	<b>Выводы</b>	<b>12</b>
	<b>Список литературы</b>	<b>13</b>

## Список иллюстраций

4.1	Заголовки разного уровня . . . . .	8
4.2	картиночки с ссылочками . . . . .	8
4.3	нумерованный список . . . . .	9
4.4	ход работы в презентации . . . . .	10
4.5	ход работы в отчете . . . . .	10
4.6	make clean . . . . .	11
4.7	make . . . . .	11
4.8	ls . . . . .	11

## Список таблиц

# 1 Цель работы

Научиться без труда подготавливать отчёт по лабораторным работам, а также познакомиться с основными возможностями разметки Markdown.

## 2 Задание

Сделать отчет к предыдущей лабораторной работе в md

## 3 Теоретическое введение

Markdown поддерживает как встраивание фрагментов кода в предложение, так и их размещение между предложениями в виде отдельных огражденных блоков. Огражденные блоки кода — это простой способ выделить синтаксис для фрагментов кода.

## 4 Выполнение лабораторной работы

### 4.1 Использование заголовков разного уровня

Для того, чтобы создать отчет, я использовала заголовки разного уровня. Количество хештегов = уровню заголовков (т.е. если перед заголовком стоит один хештэг - заголовок первого уровня, если два - второго и т.д...) (рис. [4.1])

```
# Выполнение лабораторной работы
## Настройка github
На курсе "Архитектура компьютера" мы уже создавали учетную запись на github, поэтому будем использовать ее. (рис. [-@fig:001])
! [Аккаунт github] (image/01.png) { #fig:001 width=70% }
## Базовая настройка git
```

Рис. 4.1: Заголовки разного уровня

### 4.2 Прикрепляем картиночки ( с ссылочками!)

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. (рис. [4.2])

```
git config --global core.safecrlf warn (рис. [-@fig:002])
! [Первоначальная настройка git] (image/02.png) { #fig:002 width=70% }
```

Рис. 4.2: картиночки с ссылочками



## 4.3 Используем нумерацию

Чтобы создать нумерованный список, мы делаем все как в обычном текстовом редакторе(цифра с точкой) (рис. [4.3])

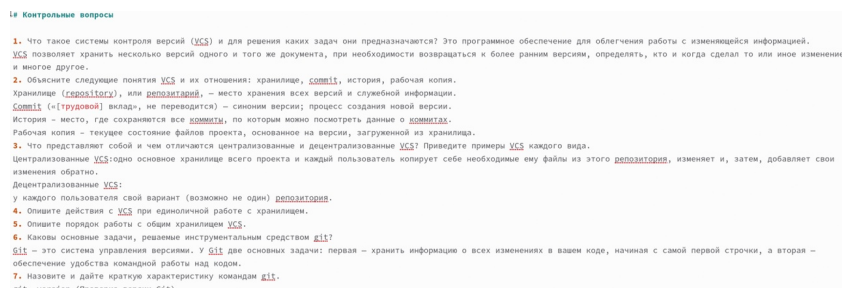


Рис. 4.3: нумерованный список

## 4.4 Описываем ход работы

Для описания хода работы, как и в отчете, так и в презентации, я использовала заголовки 2-го уровня и фотографии с ссылками, так же нумерованный список (рис. [4.4]) и (рис. [4.5])

```
## Создание ssh ключа

![создание ключа] (image/03.png) {width=30%}
![добавление ключа] (image/04.png) {width=30%}

## Авторизация на гитхаб через терминал

![авторизации] (image/05.png) {width=60%}

## Клонировем репозиторий

![клонирование репозитория] (image/06.png) {width=60%}

## Настройка каталога курса

![настройка каталога курса] (image/07.png) {width=60%}

## Отправка изменений на гитхаб

![базовая команда git push] (image/08.png) {width=60%}

# Вывод

В ходе работы мы научились взаимодействовать с сервисом гитхаб
```

Рис. 4.4: ход работы в презентации

```
## Создаём ключ SSH

(рис. [-@fig:003]) В терминале вводим данную команду:

ssh-keygen -t rsa -b 4096

Далее во всех пунктах пользуемся клавишей Enter и получаем наш ключ.

![Создание ключа] (image/03.png) { #fig:003 width=70% }

Используя команду cat ~/ .ssh/id_rsa/pub копируем наш ключ и далее добавляем его на гитхаб (рис. [-@fig:004])

![Ключ на гитхабе] (image/04.png) { #fig:004 width=70% }

## Создание репозитория

Переходим в терминал и авторизуемся с помощью команды gh auth login (рис. [-@fig:005]).

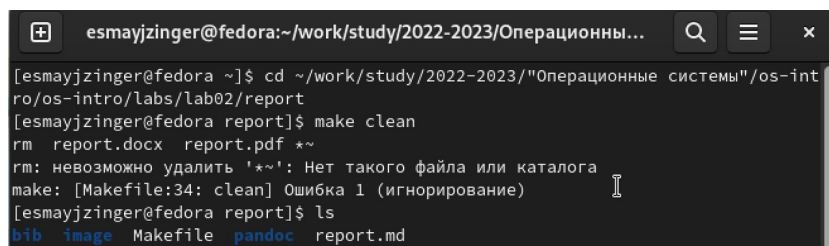
Создаём репозиторий курса на основе шаблона, с помощью команд, указанных в методичке к лабораторной работе:

1. mkdir -p ~/work/study/2021-2022/"Операционные системы"
2. cd ~/work/study/2021-2022/"Операционные системы"
3. gh repo create study_2021-2022_os-intro --template=yamadharma/course-directory-student-template --public
4. git clone --recursive git@github.com:<владелец>/study_2021-2022_os-intro.git os-intro
(рис. [-@fig:006])
```

Рис. 4.5: ход работы в отчете

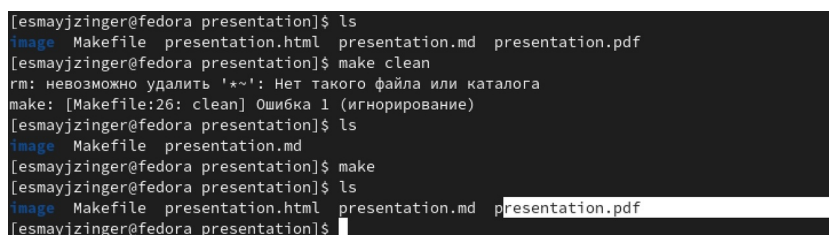
## 4.5 Форматируем отчет и презентацию в терминале

Для этого используем команду “make”, но так как я уже до этого создавала отчеты, я использовала команду “make clean”, чтобы удалить ранее созданные файлы, после создаю их снова и с помощью команды “ls” проверяю получила ли я нужные файлы. (рис. [4.6]) и (рис. [4.7]) и (рис. [4.8])



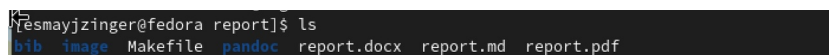
```
esmayjzinger@fedora:~/work/study/2022-2023/Операционные системы/os-intro/os-intro/labs/lab02/report
[esmayjzinger@fedora ~]$ cd ~/work/study/2022-2023/"Операционные системы"/os-intro/os-intro/labs/lab02/report
[esmayjzinger@fedora report]$ make clean
rm report.docx report.pdf *~
rm: невозможно удалить '*~': Нет такого файла или каталога
make: [Makefile:34: clean] Ошибка 1 (игнорирование)
[esmayjzinger@fedora report]$ ls
bib image Makefile pandoc report.md
```

Рис. 4.6: make clean



```
[esmayjzinger@fedora presentation]$ ls
image Makefile presentation.html presentation.md presentation.pdf
[esmayjzinger@fedora presentation]$ make clean
rm: невозможно удалить '*~': Нет такого файла или каталога
make: [Makefile:26: clean] Ошибка 1 (игнорирование)
[esmayjzinger@fedora presentation]$ ls
image Makefile presentation.md
[esmayjzinger@fedora presentation]$ make
[esmayjzinger@fedora presentation]$ ls
image Makefile presentation.html presentation.md presentation.pdf
[esmayjzinger@fedora presentation]$
```

Рис. 4.7: make



```
[esmayjzinger@fedora report]$ ls
bib image Makefile pandoc report.docx report.md report.pdf
```

Рис. 4.8: ls

## **5 Выводы**

Здесь кратко описываются итоги проделанной работы.

# Список литературы

esystem rudn