

# **Отчёт по лабораторной работе №3**

**Язык разметки Markdown**

Власов Артем Сергеевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы №3</b>	<b>6</b>
<b>4</b>	<b>Делаем отчет лабораторной работы №2</b>	<b>11</b>
<b>5</b>	<b>Выводы</b>	<b>16</b>

## Список иллюстраций

3.1	Переходим в нужный каталог . . . . .	6
3.2	Используем команду git pull . . . . .	6
3.3	Переходим в следующий каталог . . . . .	6
3.4	Используем команду make . . . . .	7
3.5	Открываем файлы и проверяем создание документов . . . . .	7
3.6	Используем команду make clean . . . . .	7
3.7	Проверяем, как сработала команда make clean . . . . .	8
3.8	Используем команду gedit . . . . .	8
3.9	Изучаем документ . . . . .	9
3.10	Изменяем документ . . . . .	10
4.1	Задаем имя и email репозитория . . . . .	11
4.2	Настраиваем utf-8 . . . . .	11
4.3	Задаем имя начальной ветки, как master . . . . .	11
4.4	Устанавливаем настройку autocrlf . . . . .	11
4.5	Устанавливаем параметр safecrlf . . . . .	11
4.6	Генерируем пару ключей командой keygen . . . . .	12
4.7	Копируем ключ из локальной консоли в буфер обмена . . . . .	12
4.8	вставляем ключ и сохраняем . . . . .	13
4.9	Проверяем добавление ключа . . . . .	13
4.10	Создаем каталоги последовательно . . . . .	13
4.11	Создаем репозиторий по шаблону . . . . .	14
4.12	Переходим в каталог курса . . . . .	14
4.13	Клонируем созданный репозиторий . . . . .	14
4.14	Переходим в нужный каталог . . . . .	14
4.15	Удаляем лишние файлы . . . . .	15
4.16	Создаем необходимые каталоги . . . . .	15
4.17	Отправляем файлы на git . . . . .	15
4.18	Проверяем отправку ЛБ1 . . . . .	15

# 1 Цель работы

Ознакомиться с языком разметки Markdown и оформить отчет по лабораторной работе №2 в ней.

## 2 Задание

Сформировать отчет по лабораторной работе №2 с помощью Markdown.

### 3 Выполнение лабораторной работы №3

Переходим в каталог, который привязан к репозиторию Git на сайте Github. (рис. fig. 3.1).

```
vlasovas@vbox:~$ cd ~/work/study/2024-2025/Архитектура\ компьютера/arch-pc/  
vlasovas@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 3.1: Переходим в нужный каталог

С помощью команды `git pull` обновляем локальный репозиторий, скачивая изменения. (рис. fig. 3.2).

```
vlasovas@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git pull  
remote: Enumerating objects: 209, done.  
remote: Counting objects: 100% (201/201), done.  
remote: Compressing objects: 100% (195/195), done.  
remote: Total 195 (delta 53), reused 0 (delta 0), pack-reused 0 (from 0)  
Получение объектов: 100% (195/195), 7.00 МиБ | 669.00 КиБ/с, готово.  
Определение изменений: 100% (53/53), завершено с 2 локальными объектами.  
Из github.com:vlasovas52/study_2024-2025_arch-pc  
d26f9a8..d723853 master -> origin/master
```

Рис. 3.2: Используем команду `git pull`

Переходим в каталог report 3 лабораторной работы. (рис. fig. 3.3).

```
vlasovas@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ cd ~/work/s  
tudy/2024-2025/Архитектура\ компьютера/arch-pc/labs/lab03/report/  
vlasovas@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab03/r  
eport$
```

Рис. 3.3: Переходим в следующий каталог

Используем команду `make` для создания файлов `report.pdf` и `report.docx` (рис. fig. 3.4).

```
vlasovas@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab03/r
eport$ make
pandoc "report.md" --filter pandoc-crossref --number-sections --citeproc -o "rep
ort.docx"
pandoc "report.md" --filter pandoc-crossref --pdf-engine=lualatex --pdf-engine-o
pt=--shell-escape --citeproc --number-sections -o "report.pdf"
```

Рис. 3.4: Используем команду make

Проверяем, как сработала команда make (рис. fig. 3.5).

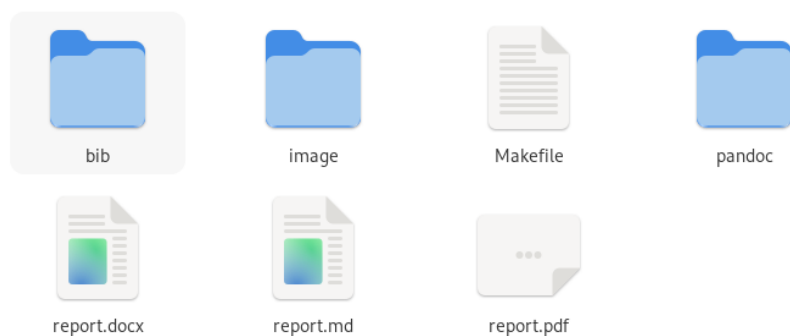


Рис. 3.5: Открываем файлы и проверяем создание документов

Используем команду make clean, которая удаляет недавно созданные докумен-  
ты(рис. fig. 3.6).

```
vlasovas@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab03/r
eport$ make clean
rm report.docx report.pdf *~
rm: невозможно удалить '*~': Нет такого файла или каталога
make: [Makefile:35: clean] Ошибка 1 (игнорирование)
vlasovas@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab03/r
eport$
```

Рис. 3.6: Используем команду make clean

Открываем файлы и смотрим, сработала ли команда make clean(рис. fig. 3.7).

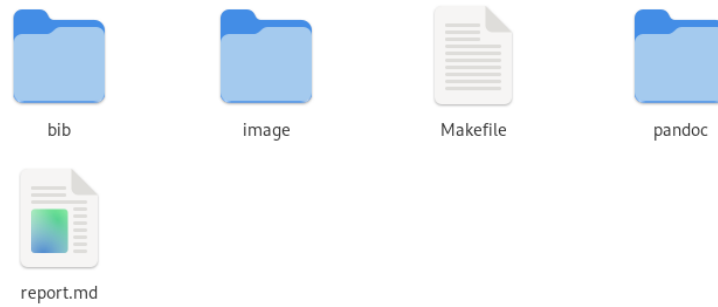


Рис. 3.7: Проверяем, как сработала команда `make clean`

Используем команду `gedit report.md`, которая открывает редактор данного документа (рис. fig. 3.8).

```
vlasovas@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab03/r  
eport$ gedit report.md
```

Рис. 3.8: Используем команду `gedit`

Изучаем открывшийся файл(рис. fig. 3.9).



```

1 |---
2 ## Front matter
3 title: "Шаблон отчёта по лабораторной работе"
4 subtitle: "Простейший вариант"
5 author: "Дмитрий Сергеевич Кулябов"
6
7 ## Generic otions
8 lang: ru-RU
9 toc-title: "Содержание"
0
1 ## Bibliography
2 bibliography: bib/cite.bib
3 csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl
4
5 ## Pdf output format
6 toc: true # Table of contents
7 toc-depth: 2
8 lof: true # List of figures
9 lot: true # List of tables
0 fontsize: 12pt
1 linestretch: 1.5
2 papersize: a4
3 documentclass: scrreprt
4 ## I18n polyglossia
5 polyglossia-lang:
6   name: russian
7   options:
8     - spelling=modern
9     - babelshorthands=true
0 polyglossia-otherlangs:
1   name: english
2 ## I18n babel
3 babel-lang: russian
4 babel-otherlangs: english
5 ## Fonts
6 mainfont: IBM Plex Serif
7 romanfont: IBM Plex Serif

```

Рис. 3.9: Изучаем документ

Изучив структуру файла, начинаем его изменять(рис. fig. 3.10).

```

3 - \usepackage{float} # keep figures where there are in the text
4 - \floatplacement{figure}{H} # keep figures where there are in the text
5 ---
6
7 # Цель работы
8
9 Ознакомиться с языком разметки Markdown и оформить отчет по лабораторной работе №2 в ней.
10
11 # Задание
12
13 Сформировать отчет по лабораторной работе №2 с помощью Markdown.
14
15 # Выполнение лабораторной работы №3
16
17 Переходим в каталог, который привязан к репозиторию Git на сайте Github. (рис. @fig:001).
18
19 ![Переходим в нужный каталог](image/1.png){#fig:001 width=70%}
20
21 С помощью команды git pull обновляем локальный репозиторий, скачивая изменения. (рис. @fig:002).
22
23 ![Используем команду git pull](image/2.png){#fig:002 width=70%}
24
25 Переходим в каталог report 3 лабораторной работы. (рис. @fig:003).
26
27 ![Переходим в следующий каталог](image/3.png){#fig:003 width=70%}
28
29 Используем команду make для создания файлов report.pdf и report.docx (рис. @fig:004).
30
31 ![Используем команду make](image/5.png){#fig:004 width=70%}
32
33 Проверяем, как сработала команда make (рис. @fig:005).
34
35 ![Открываем файлы и проверяем создание документов](image/6.png){#fig:005 width=70%}
36
37 Используем команду make clean, которая удаляет недавно созданные документы (рис. @fig:006).

```

Markdown ▾ Ширина таблицы: 8 ▾ Ln 1,

Рис. 3.10: Изменяем документ

## 4 Делаем отчет лабораторной работы №2

Делаем предварительную конфигурацию git. (рис. fig. 4.1).

```
liveuser@localhost-live:~$ git config --global user.name "vlasovas52"  
liveuser@localhost-live:~$ git config --global user.email "1132246841@pfur.ru"
```

Рис. 4.1: Задаем имя и email репозитория

Настраиваем utf-8 в выводе сообщения git. (рис. fig. 4.2).

```
liveuser@localhost-live:~$ git config --global core.quotePath false
```

Рис. 4.2: Настраиваем utf-8

Задаем имя начальной ветки. (рис. fig. 4.3).

```
liveuser@localhost-live:~$ git config --global init.defaultBranch master
```

Рис. 4.3: Задаем имя начальной ветки, как master

```
liveuser@localhost-live:~$ git config --global core.autocrlf input
```

Рис. 4.4: Устанавливаем настройку autocrlf

```
liveuser@localhost-live:~$ git config --global core.safecrlf warn
```

Рис. 4.5: Устанавливаем параметр safecrlf

Создаем SSH ключ(рис. fig. 4.6).

```
liveuser@localhost-live:~$ ssh-keygen -C "vlasovas52 1132246841@pfur.ru"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/liveuser/.ssh/id_ed25519):
Created directory '/home/liveuser/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/liveuser/.ssh/id_ed25519
Your public key has been saved in /home/liveuser/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:Zzh2Wl6kebvAdd38Q+9fNa0D26xz+mugZRyCZdLk3SY vlasovas52 1132246841@pfur.ru
The key's randomart image is:
+--[ED25519 256]--+
|      o.      |
|    ..+. .   |
|   =. E o    |
|  ...+.o oo   |
| S Bo=..o*   |
| . X +=B..=  |
| . ++o.=.+   |
|  ...oooo    |
|   ++o.+    |
+-----[SHA256]-----+
liveuser@localhost-live:~$
```

Рис. 4.6: Генерируем пару ключей командой keygen

```
liveuser@localhost-live:~$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAINQmnwfkyyvIpl5vTjdBmzH6UDUbQKgemqrPs39AxxaCy
vlasovas52 1132246841@pfur.ru
liveuser@localhost-live:~$
```

Рис. 4.7: Копируем ключ из локальной консоли в буфер обмена

Заходим в свой аккаунт на сайте github. Переходим в настройки, SSH ключи.  
(рис. fig. 4.8).

### Add new SSH Key

Title

Key type

Authentication Key ↕

Key

```
ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAINQmnwfkYvIpl5vTjdBmzH6UDUbQKgemqrPs39AxkaCy
```


Add SSH key

Рис. 4.8: вставляем ключ и сохраняем

### SSH keys New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

#### Authentication keys

	<b>master</b> SHA256: Zzh2Wl6kebvAdd38Q+9fNa0D26xz+mugZRyCZdLk3SY Added on Sep 28, 2024 Never used — Read/write	<span>Delete</span>
---	--	---------------------

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

Рис. 4.9: Проверяем добавление ключа

Открываем терминал и создаем каталоги для предмета “Архитектура компьютера”(рис. fig. 4.10).

```
liveuser@localhost:~$ mkdir -p ~/work/study/2024-2025/"Архитектура компьютера"
```

Рис. 4.10: Создаем каталоги последовательно

Переходим на страницу репозитория с шаблоном(рис. fig. 4.11).

Import a repository.'. A note states 'Required fields are marked with an asterisk (\*)'. The 'Repository template' section shows 'yamadharma/course-directory-student-template' selected. Below this, there's a checkbox for 'Include all branches' with a description: 'Copy all branches from yamadharma/course-directory-student-template and not just the default branch.' The 'Owner' field shows 'vlasovas52' and the 'Repository name' field shows 'study\_2024-2025\_arch-pc'. A green checkmark indicates 'study\_2024-2025\_arch-pc is available.' Below this, a suggestion says 'Great repository names are short and memorable. Need inspiration? How about curly-octo-garbanzo?'. The 'Description (optional)' field is empty. The 'Visibility' section has two options: 'Public' (selected) and 'Private'. A note at the bottom says 'You are creating a private repository in your personal account.' A green 'Create repository' button is at the bottom right."/>

**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

**Repository template**

yamadharma/course-directory-student-template ▾

Start your repository with a template repository's contents.

☐ **Include all branches**  
Copy all branches from yamadharma/course-directory-student-template and not just the default branch.

**Owner \*** vlasovas52 ▾ / **Repository name \*** study\_2024-2025\_arch-pc

✔ study\_2024-2025\_arch-pc is available.

Great repository names are short and memorable. Need inspiration? How about curly-octo-garbanzo?

**Description (optional)**

☐ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☒ **Private**  
You choose who can see and commit to this repository.

ⓘ You are creating a private repository in your personal account.

**Create repository**

Рис. 4.11: Создаем репозиторий по шаблону

Переходим в папку с предметом(рис. fig. 4.12).

```
liveuser@localhost-live:~$ cd ~/work/study/2024-2025/"Архитектура компьютера"
liveuser@localhost-live:~/work/study/2024-2025/Архитектура компьютера$
```

Рис. 4.12: Переходим в каталог курса

```
liveuser@localhost-live:~/work/study/2024-2025/Архитектура компьютера$ git clone --recursive git@github.com:vlasovas52/study_2024-2025_arch-pc.git arch-pc
Cloning into 'arch-pc'...
```

Рис. 4.13: Клонировем созданный репозиторий

Переходим в каталог arch-pc(рис. fig. 4.14).

```
liveuser@localhost-live:~/work/study/2024-2025/Архитектура компьютера$ cd ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc
liveuser@localhost-live:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 4.14: Переходим в нужный каталог

```
liveuser@localhost-live:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ rm package.json
liveuser@localhost-live:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 4.15: Удаляем лишние файлы

Создаем папки по образцу(рис. fig. 4.16).

```
liveuser@localhost-live:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
liveuser@localhost-live:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make
```

Рис. 4.16: Создаем необходимые каталоги

Отправляем файлы на сервер(рис. fig. 4.17).

```
liveuser@localhost-live:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
liveuser@localhost-live:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -am "feat(main): m
ake course structure"
[master 1ceec39] feat(main): make course structure
221 files changed, 53680 insertions(+)
```

Рис. 4.17: Отправляем файлы на git

Отправляем прошлую лабораторную работу(рис. fig. 4.18).

master
study\_2024-2025\_arch-pc / labs / lab02

Add file
...

report

vlasovas52 Create report.pdf 1c428fe · now

Name	Last commit message	Last commit date
..		
bib	feat(main): make course structure	2 months ago
image	feat(main): make course structure	2 months ago
pandoc	feat(main): make course structure	2 months ago
Makefile	feat(main): make course structure	2 months ago
report.md	feat(main): make course structure	2 months ago
report.pdf	Create report.pdf	now

Рис. 4.18: Проверяем отправку ЛБ1

## 5 Выводы

Мы познакомились с языком разметки Markdown и оформили отчет в ней и загрузили на Github.