

# **Отчёт по лабораторной работе №3**

**Язык разметки Markdown**

Джаллох Ишмаил

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы №3</b>	<b>7</b>
<b>4</b>	<b>Делаем отчет лабораторной работы №2</b>	<b>12</b>
<b>5</b>	<b>Выводы</b>	<b>17</b>

# Список иллюстраций

3.1	Переходим в нужный каталог . . . . .	7
3.2	Используем команду git pull . . . . .	7
3.3	Переходим в следующий каталог . . . . .	7
3.4	Используем команду make . . . . .	8
3.5	Открывем файлы и проверяем создание документов . . . . .	8
3.6	Используем команду make clean . . . . .	8
3.7	Проверяем, как сработала команда make clean . . . . .	9
3.8	Используем команду gedit . . . . .	9
3.9	Изучаем документ . . . . .	10
3.10	Изменяем документ . . . . .	11
4.1	Задаем имя и email репозитория . . . . .	12
4.2	Настраиваем utf-8 . . . . .	12
4.3	Задаем имя начальной ветки, как master . . . . .	12
4.4	Устанавливаем настройку autocrlf . . . . .	12
4.5	Устанавливаем параметр safecrlf . . . . .	12
4.6	Генерируем пару ключей командой keygen . . . . .	13
4.7	Копируем ключ из локальной консоли в буфер обмена . . . . .	13
4.8	вставляем ключ и сохраняем . . . . .	13
4.9	Проверяем добавление ключа . . . . .	14
4.10	Создаем каталоги последовательно . . . . .	14
4.11	Создаем репозиторий по шаблону . . . . .	14
4.12	Переходим в каталог курса . . . . .	14
4.13	Клонируем созданный репозиторий . . . . .	15
4.14	Переходим в нужный каталог . . . . .	15
4.15	Удаляем лишние файлы . . . . .	15
4.16	Создаем необходимые каталоги . . . . .	15
4.17	Отправляем файлы на git . . . . .	15
4.18	Проверяем отправку ЛБ1 . . . . .	16

## **Список таблиц**

# 1 Цель работы

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

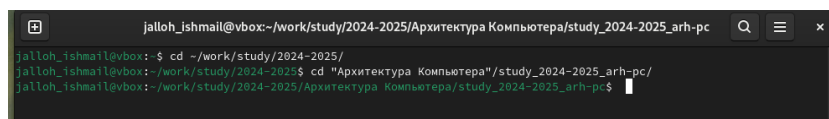
## **2 Задание**

Сформировать отчет по лабораторной работе №2 с помощью Markdown.

## 3 Выполнение лабораторной работы

### №3

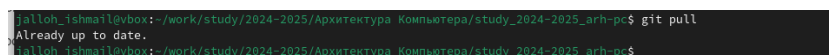
Переходим в каталог, который привязан к репозиторию Git на сайте Github.  
(рис. fig. 3.1).



```
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc
jalloh_ishmail@vbox:~$ cd ~/work/study/2024-2025/
jalloh_ishmail@vbox:~/work/study/2024-2025$ cd "Архитектура Компьютера"/study_2024-2025_arh-pc/
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc$
```

Рис. 3.1: Переходим в нужный каталог

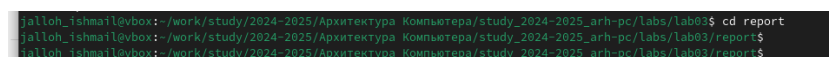
С помощью команды `git pull` обновляем локальный репозиторий, скачивая изменения. (рис. fig. 3.2).



```
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc$ git pull
Already up to date.
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc$
```

Рис. 3.2: Используем команду `git pull`

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



```
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc/labs/lab03$ cd report
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc/labs/lab03/report$
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc/labs/lab03/report$
```

Рис. 3.3: Переходим в следующий каталог

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

```
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc/labs/lab03...
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc/labs/lab03/report$ make
pandoc "report.md" --filter pandoc-crossref --number-sections --citeproc -o "report.docx"
WARNING: pandoc-crossref was compiled with pandoc 3.1.12.2 but is being run through 3.1.3. This is not supported. Strange things may (and likely will) happen silently.
[WARNING] Could not fetch resource image/1.png: replacing image with description
pandoc "report.md" --filter pandoc-crossref --pdf-engine=lualatex --pdf-engine-opt=--shell-escape --citeproc --number-sections -o "report.pdf"
```

Рис. 3.4: Используем команду make

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

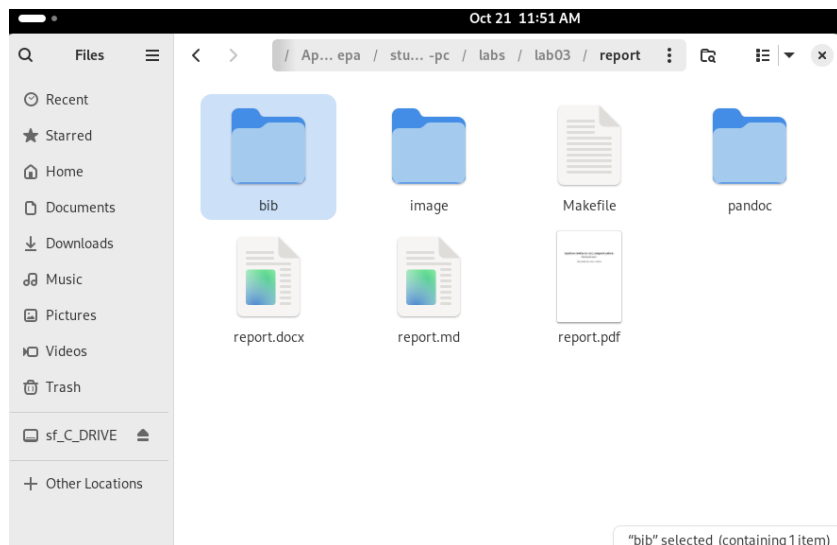


Рис. 3.5: Открываем файлы и проверяем создание документов

Используем команду make clean, которая удаляет недавно созданные документы(рис. fig. 3.6).

```
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc/labs/lab03/report$ make clean
rm report.docx report.pdf *~
rm: cannot remove '*~': No such file or directory
make: [Makefile:35: clean] Error 1 (ignored)
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc/labs/lab03/report$
```

Рис. 3.6: Используем команду make clean

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



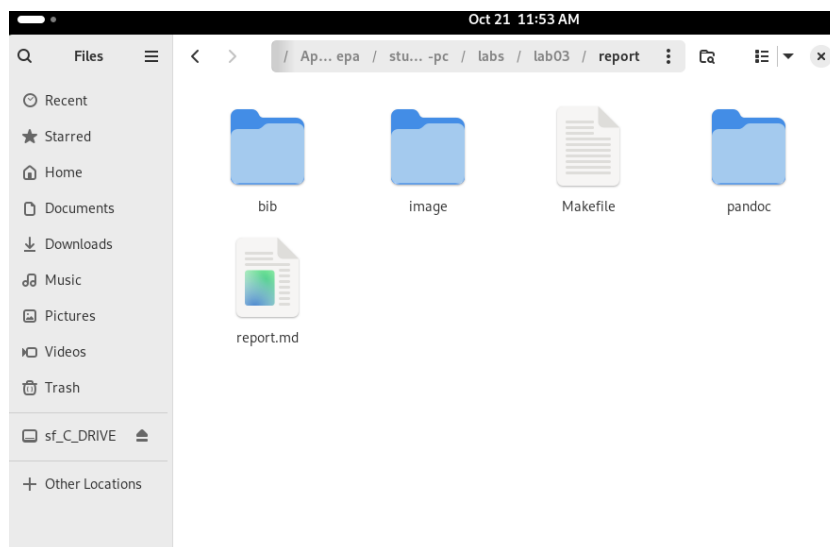


Рис. 3.7: Проверяем, как сработала команда `make clean`

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

```
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc/labs/lab03/report$ gedit report.md
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc/labs/lab03/report$
```

Рис. 3.8: Используем команду `gedit`

Изучаем открывшийся файл(рис. fig. 3.9).

```
Open ▾ + • report.md
~/work/study/2024-2025/Аспирантура Кооператива/study_2024-2025_arh-pc/labs/lab03/report

---
## Front matter
title: "Отчет по лабораторной работе №3"

subtitle: "Как писать Markdown"

author: "Александр Иванов"

group: "ИКА-05-24"


## Generic options
lang: ru-RU
doc-title: "Содержание"


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


## Pdf output format
doc: true # Table of contents
doc-depth: 2
lot: true # List of figures
lot: true # List of tables

font-size: 12pt
linespacing: 1.5
papersize: a4
documentclass: scrreprt
## Use polyglossia
polyglossia-lang:
  name: russian
  options:
    - spelling=modern
    - babelshorthands=true
polyglossia-otherlangs:
  name: english
## Use babel
babel-lang: russian
babel-otherlangs: english
## Fonts
mainfont: IBM Plex Serif
romanfont: IBM Plex Serif
sansfont: IBM Plex Sans
monofont: IBM Plex Mono
mathfont: SILX Two Math
mainfontoptions: Ligatures=Common,Ligatures=TeX,Scale=0.94
romanfontoptions: Ligatures=Common,Ligatures=TeX,Scale=0.94
sansfontoptions: Ligatures=Common,Ligatures=TeX,Scale=MatchLowercase,Scale=0.94
monofontoptions: Scale=MatchLowercase,Scale=0.94,FakeSlantish=0.9
mathfontoptions:
## Biblatex
bibtex: true
bibtex-style: "apa-numeric"
bibtexoptions:
  - parenttracker=true
```

Рис. 3.9: Изучаем документ

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

```

indent: true
header-includes:
  - \usepackage{indentfirst}
  - \usepackage{float} # keep figures where there are in the text
  - \floatplacement{figure}{H} # keep figures where there are in the text
---

# Hello World

Знакомимся с новым форматом Markdown и оформляем текст по лабораторной работе №2 в HED.

# Задачи

Сформировать текст по лабораторной работе №2 с помощью Markdown.

# Выполнение лабораторной работы №3

Переходим в каталог, который привязан к репозиторию Git на сайте Github. (рис. @fig:001).

! [Переходим в нужный каталог] (image/1.png) {#fig:001 width=70%}

С помощью команды git pull обновляем локальный репозиторий, скачивая изменения. (рис. @fig:002).

! [Используем команду git pull] (image/2.png) {#fig:002 width=70%}

Переходим в каталог report 3 лабораторной работы. (рис. @fig:003).

! [Переходим в нужный каталог] (image/3.png) {#fig:003 width=70%}

Используем команду make для создания файла report.pdf и report.docx (рис. @fig:004).

! [Используем команду make] (image/5.png) {#fig:004 width=70%}

Повторим, как создавала команда make (рис. @fig:005).

! [Открываем файл и проверяем создание документов] (image/6.png) {#fig:005 width=70%}

Используем команду make clean, которая удалит все наши созданные документы (рис. @fig:006).

! [Используем команду make clean] (image/7.png) {#fig:006 width=70%}

Открываем файл и смотрим, создала ли команда make clean (рис. @fig:007).

! [Проверяем, как создала команда make clean] (image/8.png) {#fig:007 width=70%}

Используем команду sedit report.md, которая отредактирует файл документов (рис. @fig:008).

! [Используем команду sedit] (image/9.png) {#fig:008 width=70%}

Изучаем отредактированный файл (рис. @fig:009).

! [Изучаем документ] (image/10.png) {#fig:009 width=70%}

Изучив структуру файла, начинаем его изменять (рис. @fig:010).

```

Рис. 3.10: Изменяем документ

## 4 Делаем отчет лабораторной работы №2

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

```
jalloh_ishmail@vbox:~$ git config --global user.name "Ishmail Jalloh"
jalloh_ishmail@vbox:~$ git config --global user.email "isjalloh6040@gmail.com"
jalloh_ishmail@vbox:~$
```

Рис. 4.1: Задаем имя и email репозитория

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

```
jalloh_ishmail@vbox:~$ git config --global core.quotepath false
jalloh_ishmail@vbox:~$
```

Рис. 4.2: Настраиваем utf-8

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

```
jalloh_ishmail@vbox:~$ git config --global init.defaultBranch master
jalloh_ishmail@vbox:~$
```

Рис. 4.3: Задаем имя начальной ветки, как master

```
jalloh_ishmail@vbox:~$ git config --global core.autosafecrlf input
jalloh_ishmail@vbox:~$
```

Рис. 4.4: Устанавливаем настройку autocrlf

```
jalloh_ishmail@vbox:~$ git config --global core.safecrlf warn
jalloh_ishmail@vbox:~$
```

Рис. 4.5: Устанавливаем параметр safecrlf

Создаем SSH ключ(рис. fig. 4.6).

```
jalloh_ishmail@vbox:~$ ssh-keygen -t rsa -b 4096 -C"Ishmail Jalloh isjalloh6040@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jalloh_ishmail/.ssh/id_rsa):
/home/jalloh_ishmail/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jalloh_ishmail/.ssh/id_rsa
Your public key has been saved in /home/jalloh_ishmail/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:fBP/lc7fpMf705Qg0gHyqp3t3UPeM+/3fUIt4F6qEQc Ishmail Jalloh isjalloh6040@gmail.com
The key's randomart image is:
+---[RSA 4096]-----+
|
| . .
| o . E
| o . =.. .
| . S *.+.oo|
| . + =..+..|
| o o =..+o=|
| . o .. .+o.+o|
| .. ..o. *B#|
+---[SHA256]-----+
jalloh_ishmail@vbox:~$
```

Рис. 4.6: Генерируем пару ключей командой keygen

```
jalloh_ishmail@vbox:~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
jalloh_ishmail@vbox:~$
```

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).

Рис. 4.8: вставляем ключ и сохраняем

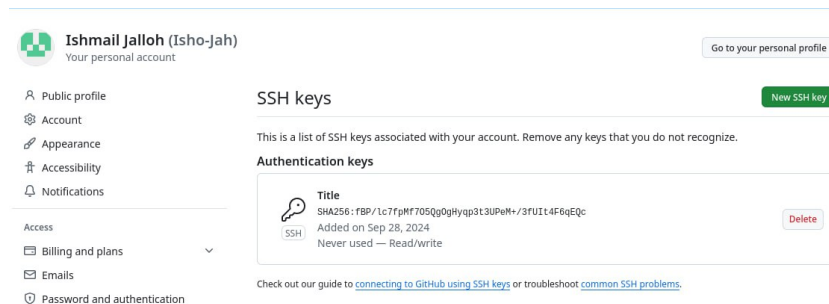


Рис. 4.9: Проверяем добавление ключа

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

```
jalloh_ishmail@vbox:~$ mkdir -p ~/work/study/2024-2025/"Архитектура Компьютера"
jalloh_ishmail@vbox:~$
```

Рис. 4.10: Создаем каталоги последовательно


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

**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 \*** **Repository name \***

 Isho-Jah / study\_2024-2025\_arh-pc

study\_2024-2025\_arh-pc is available.

Great repository names are short and memorable. Need inspiration? How about [probable-dollop](#) ?

**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).

```
jalloh_ishmail@vbox:~$ cd ~/work/study/2024-2025/"Архитектура Компьютера"
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера$
```

Рис. 4.12: Переходим в каталог курса

```
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера$ git clone --recursive git@github.com:
Isho-Jah/study_2024-2025_arh-pc.git arch-pc
Cloning into 'arch-pc'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 33 (delta 1), reused 18 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (33/33), 18.82 KiB | 6.27 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markdown-template
.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharma/academic-laboratory-report-template.git) regi
stered for path 'template/report'
Cloning into '/home/jalloh_ishmail/work/study/2024-2025/Архитектура Компьютера/arch-pc/template/presenta
tion'...
remote: Enumerating objects: 111, done.
remote: Counting objects: 100% (111/111), done.
remote: Compressing objects: 100% (77/77), done.
remote: Total 111 (delta 42), reused 100 (delta 31), pack-reused 0 (from 0)
Receiving objects: 100% (111/111), 102.17 KiB | 1.62 MiB/s, done.
Resolving deltas: 100% (42/42), done.
Cloning into '/home/jalloh_ishmail/work/study/2024-2025/Архитектура Компьютера/arch-pc/template/report'.
...
remote: Enumerating objects: 142, done.
```

Рис. 4.13: Клонировем созданный репозиторий

Переходим в каталог arch-pc(рис. fig. 4.13).

```
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера$ cd study_2024-2025_arh-pc
```

Рис. 4.14: Переходим в нужный каталог

```
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc$ rm package.json
```

Рис. 4.15: Удаляем лишние файлы

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

```
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc$ echo os-intro > COURSE
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc$ make

Usage:
  make <target>

Targets:
  list           List of courses
  prepare        Generate directories structure
  submodule      Update submules
```

Рис. 4.16: Создаем необходимые каталоги

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

```
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc$ git add .
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-2025_arh-pc$ git commit -am 'feat(main): make structure'
[master 151559a] feat(main): make structure
2 files changed, 1 insertion(+), 14 deletions(-)
delete mode 100644 package.json
```

Рис. 4.17: Отправляем файлы на git

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

<div> <div>master</div> <div>study_2024-2025_arh-pc / labs / lab01 / report</div> <div> <div>Go to file</div> <div>Add file</div> <div>...</div> </div> </div>		
<div> <div>Isho-jah feat(main): make structure</div> <div>dd84531 · 1 hour ago</div> <div>History</div> </div>		
Name	Last commit message	Last commit date
..		
bib	feat(main): make structure	1 hour ago
pandoc	feat(main): make structure	1 hour ago
Lab 01.pdf	feat(main): make structure	1 hour ago
Makefile	feat(main): make structure	1 hour ago
report.md	feat(main): make structure	1 hour ago

Рис. 4.18: Проверяем отправку ЛБ1



## 5 Выводы

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