

# **Отчёт по лабораторной работе №3**

**Язык разметки Markdown**

Спелов Андрей Николаевич

# Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы №3	7
4	Делаем отчет лабораторной работы №2	12

## 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	Открываем файлы и проверяем создание документов . . . . .	8
3.6	Используем команду make clean . . . . .	9
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 . . . . .	13
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 . . . . .	14
4.9	Проверяем добавление ключа . . . . .	14
4.10	Создаем каталоги последовательно . . . . .	15
4.11	Создаем репозиторий по шаблону . . . . .	15
4.12	Переходим в каталог курса . . . . .	15
4.13	Клонируем созданный репозиторий . . . . .	16
4.14	Переходим в нужный каталог . . . . .	16
4.15	Удаляем лишние файлы . . . . .	16
4.16	Создаем необходимые каталоги . . . . .	16
4.17	Отправляем файлы на git . . . . .	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. 4.1).

```
spelov@spelov-VirtualBox:~$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/  
spelov@spelov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 3.1: Переходим в нужный каталог

С помощью команды git pull обновляем локальный репозиторий, скачивая изменения. (рис. 4.2).

```
spelov@spelov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git pull  
remote: Enumerating objects: 18, done.  
remote: Counting objects: 100% (16/16), done.  
remote: Compressing objects: 100% (12/12), done.  
remote: Total 12 (delta 5), reused 0 (delta 0), pack-reused 0  
Распаковка объектов: 100% (12/12), 1.75 МиБ | 1.57 МиБ/с, готово.  
Из github.com:SpelovA/study_2023-2024 arh-pc  
1391223..968c3b8 master -> origin/master  
Обновление 1391223..968c3b8  
Fast-forward  
 labs/lab01/report/ЛБ1.pdf | Bin 0 -> 1031314 bytes  
 labs/lab02/report/ЛБ2.pdf | Bin 0 -> 956496 bytes  
 2 files changed, 0 insertions(+), 0 deletions(-)  
 create mode 100644 labs/lab01/report/ЛБ1.pdf  
 create mode 100644 labs/lab02/report/ЛБ2.pdf  
spelov@spelov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 3.2: Используем команду git pull

Переходим в каталог report 3 лабораторной работы. (рис. 4.3).

```
spelov@spelov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab03/report  
spelov@spelov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab03/report$
```

Рис. 3.3: Переходим в следующий каталог

Используем команду make для создания файлов report.pdf и report.docx (рис. 4.4).

```
[spelovandrei@fedora report]$ make
pandoc "report.md" --filter pandoc/filters/pandoc_fignos.py --filter pandoc/filters/pandoc_eqnos.py --filter pandoc/filters/pandoc_tablenos.py --filter pandoc/filters/pandoc_secnos.py --number-sections --citeproc -o "report.docx"
pandoc "report.md" --filter pandoc/filters/pandoc_fignos.py --filter pandoc/filters/pandoc_eqnos.py --filter pandoc/filters/pandoc_tablenos.py --filter pandoc/filters/pandoc_secnos.py --pdf-engine=lualatex --pdf-engine-opt=--shell-escape --citeproc --number-sections -o "report.pdf"
[spelovandrei@fedora report]$
```

Рис. 3.4: Используем команду make

Проверяем, как сработала команда make (рис. 4.5).

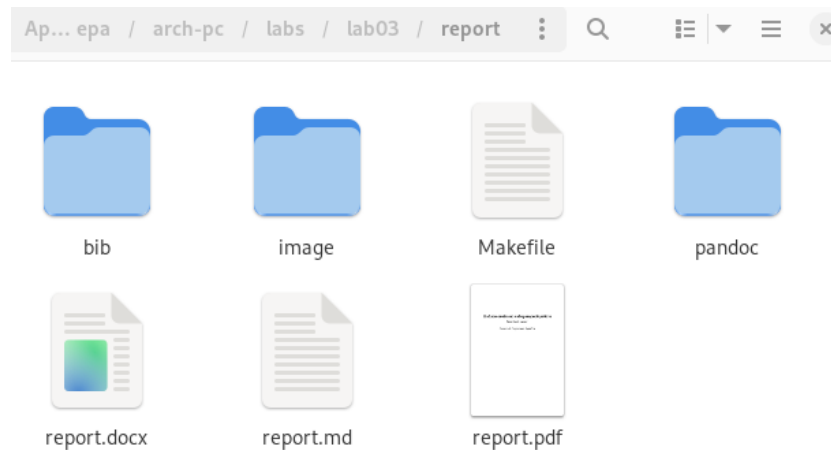


Рис. 3.5: Открываем файлы и проверяем создание документов

Используем команду make clean, которая удаляет недавно созданные документы(рис. 4.6).



```
[spelovandrei@fedora report]$ make clean
rm report.docx report.pdf *~
rm: невозможно удалить '*~': Нет такого файла или каталога
make: [Makefile:34: clean] Ошибка 1 (игнорирование)
[spelovandrei@fedora report]$
```

Рис. 3.6: Используем команду make clean

Открываем файлы и смотрим, сработала ли команда make clean(рис. 4.7).

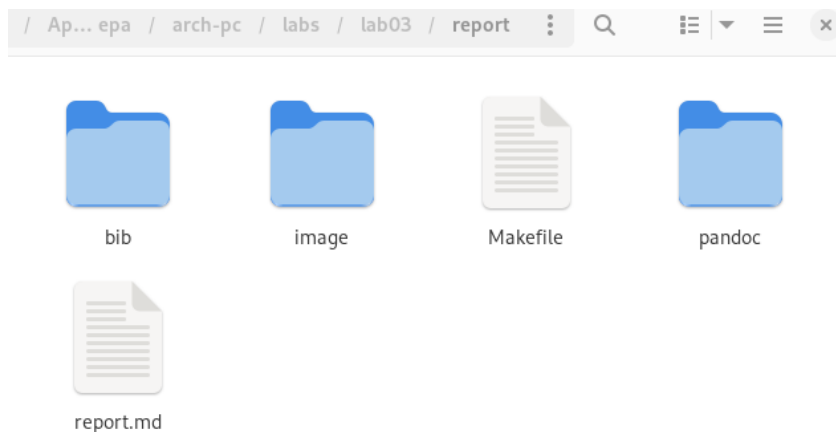


Рис. 3.7: Проверяем, как сработала команда make clean

Используем команду `gedit report.md`, которая открывает редактор данного документа (рис. 4.8).

```
[spelovandrei@fedora report]$ gedit report.md
```

Рис. 3.8: Используем команду gedit

Изучаем открывшийся файл(рис. 4.9).

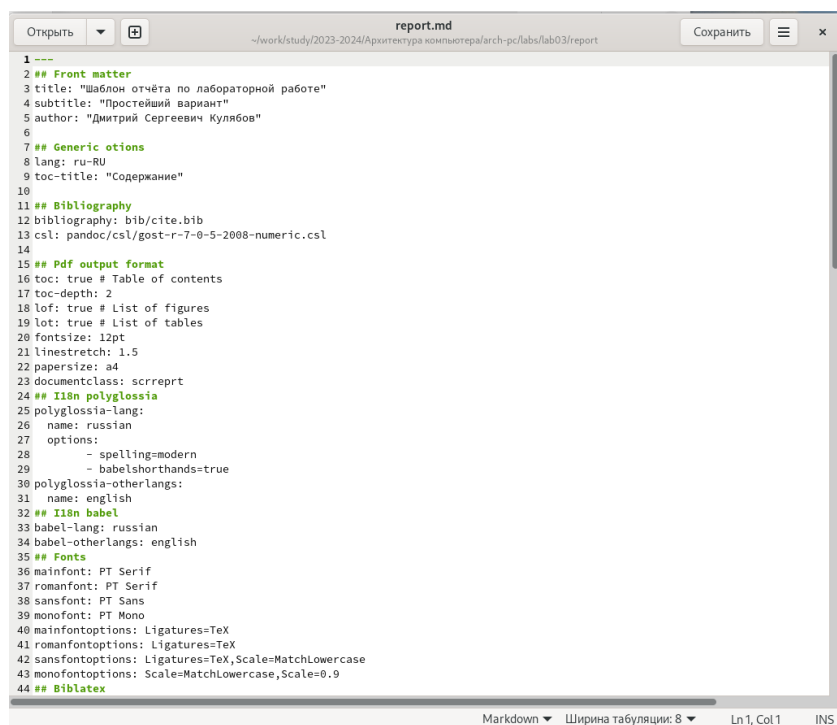


Рис. 3.9: Изучаем документ

Изучив структуру файла, начинаем его изменять(рис. 4.10).

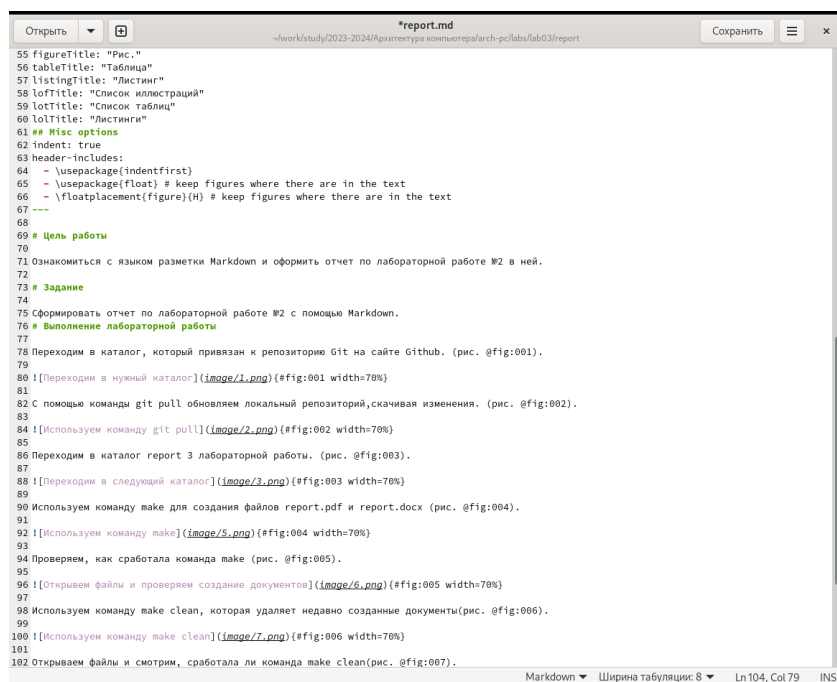


Рис. 3.10: Изменяем документ

## 4 Делаем отчет лабораторной работы №2

Делаем предварительную конфигурацию git. (рис. 4.1).

```
spelov@spelov-VirtualBox:~$ git config --global user.name "SpelovA"  
spelov@spelov-VirtualBox:~$ git config --global user.email "1132231839@pfur.ru"  
spelov@spelov-VirtualBox:~$
```

Рис. 4.1: Задаем имя и email репозитория

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

```
spelov@spelov-VirtualBox:~$ git config --global core.quotepath false  
spelov@spelov-VirtualBox:~$
```

Рис. 4.2: Настраиваем utf-8

Задаем имя начальной ветки. (рис. 4.3).

```
spelov@spelov-VirtualBox:~$ git config --global init.defaultBranch master  
spelov@spelov-VirtualBox:~$
```

Рис. 4.3: Задаем имя начальной ветки, как master

```
spelov@spelov-VirtualBox:~$ git config --global core.autocrlf input  
spelov@spelov-VirtualBox:~$
```

Рис. 4.4: Устанавливаем настройку autocrlf

```
spelov@spelov-VirtualBox:~$ git config --global core.safecrlf warn
spelov@spelov-VirtualBox:~$
```

Рис. 4.5: Устанавливаем параметр safecrlf

Создаем SSH ключ(рис. 4.6).

```
spelov@spelov-VirtualBox:~$ ssh-keygen -C "SpelovA 1132231839@pfur.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/spelov/.ssh/id_rsa):
/home/spelov/.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/spelov/.ssh/id_rsa
Your public key has been saved in /home/spelov/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:3eRFgkX3yTKoe8DkzLGTWbR1UUPwpivaUQ0S7u7M6Z4 SpelovA 1132231839@pfur.ru
The key's randomart image is:
+---[RSA 3072]-----+
|
|  o  *+.=+|
| . 0.=.=|
|+.++o.B|
|S*..oo.|
|o = .|
|+ = .|
|B =|
|.oE|
+---[SHA256]-----+
spelov@spelov-VirtualBox:~$
```

Рис. 4.6: Генерируем пару ключей командой keygen

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

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

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

https://github.com/settings/ssh/new

## Add new SSH Key

**Title**

**Key type**

Authentication Key

**Key**

```
/OUjaJGn5M+yMvNAgLnwhEVxHcCygGgygEdbTThUv8GMFGAcB  
4+wgHlnK+Yvr1fPv8UmhbydM+qFwx2UGNs9YBZb  
/UEhz/9f93Sdl0A+wxzu9gPRNgulspZaY7NVNMnri4h6ltt3Imqe05C  
wYPWl1Ba518cDupleELRvAzA551jHmMru7k  
/yur62Jy1ScJIWonA0JJCZ2+e6+NYDyUqPkebCZ2BCzw55S18KV5  
igt3ArevnkL5M5EnZzkM6Uuc32Eiq4G0XyN6N  
/sprqyO7E59sXEsT+7UdgHqFRDBcctRc= SpelovA  
1132231839@pfur.ru  
|
```


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



SSH

**Title**

SHA256:3eRFgkX3yTKoe8Dkz1GTWbR1UUPwpivaUQ0S7u7M6Z4

Added on Sep 23, 2023

Never used — Read/write

Delete

Рис. 4.9: Проверяем добавление ключа

Открываем терминал и создаем каталоги для предмета “Архитектура компью-

тера”(рис. 4.10).

```
spelov@spelov-VirtualBox:~$ mkdir -p ~/work/study/2023-2024/"Архитектура компью  
ера"  
spelov@spelov-VirtualBox:~$
```

Рис. 4.10: Создаем каталоги последовательно

Переходим на страницу репозитория с шаблоном(рис. 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**

yamadharm/course-directory-student-template ▾

Start your repository with a template repository's contents.

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

**Owner \*** SpelovA ▾ / **Repository name \***

⚠ **Your new repository will be created as study\_2023-2024\_arh-pc.**  
The repository name can only contain ASCII letters, digits, and the characters -, ., and \_.

Great repository names are short and memorable. Need inspiration? How about **ubiquitous-octo-barnacle** ?

**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. 4.12).

```
spelov@spelov-VirtualBox:~$ cd ~/work/study/2023-2024/"Архитектура компьютера"  
spelov@spelov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера$
```

Рис. 4.12: Переходим в каталог курса

```
spelov@spelov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера$ git clone --recursive git@github.com:SpelovA/study_2023-2024_arh-pc.git arch-pc
Клонирование в «arch-pc»...
```

Рис. 4.13: Клонировем созданный репозиторий

Переходим в каталог arch-рс(рис. 4.14).

```
spelov@spelov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера$ cd ~/work/study/2023-2024/
"Архитектура компьютера"/arch-pc
spelov@spelov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 4.14: Переходим в нужный каталог

```
spelov@spelov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ rm package.json
```

Рис. 4.15: Удаляем лишние файлы

Создаем папки по образцу(рис. 4.16).

```
spelov@spelov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ echo arch-pc > CPU
RSE
spelov@spelov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ make
```

Рис. 4.16: Создаем необходимые каталоги

Отправляем файлы на сервер(рис. 4.17).

```
spelo@spelo-VirtualBox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .
spelo@spelo-VirtualBox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit -am "feat
at(main): make course structure"
[master 1391223] feat(main): make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
```

Рис. 4.17: Отправляем фалы на git

Отправляем прошлую лабораторную работу(рис. ??).

The screenshot shows a file explorer window with a search bar at the top containing 'study\_2023-2024\_ahw-pc / lab01 / report /'. Below the search bar, there's a list of files and folders. The files are: 'lab01', 'image', 'practical', 'Makefile', 'report.md', and 'lab1.pdf'. Each file has a small icon to its left and a 'Last commit message' column to its right. The 'Last commit message' column contains the text 'Add files via upload' for all files. The 'lab1.pdf' file is highlighted in blue.

## # Выводы

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