

# **Отчёт по лабораторной работе №2**

**Дисциплина: Архитектура компьютера**

София Андреевна Кудякова

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задания</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
4.1	Настройка GitHub . . . . .	8
4.2	Базовая настройка Git . . . . .	8
4.3	Создание SSH ключа . . . . .	9
4.4	Создание рабочего пространства и репозитория курса на основе шаблона . . . . .	11
4.5	Создание репозитория курса на основе шаблона . . . . .	11
4.6	Настройка каталога курса . . . . .	14
<b>5</b>	<b>Выводы</b>	<b>19</b>

## Список иллюстраций

4.1	Создание учетной записи GitHub . . . . .	8
4.2	Предварительная конфигурация git . . . . .	9
4.3	Настройка utf-8 в выводе сообщений git . . . . .	9
4.4	Создание имени для начальной ветки. Параметры autocrlf и safecrlf . . . . .	9
4.5	Создание SSH-ключа . . . . .	10
4.6	Установка xclip . . . . .	10
4.7	Копирование с помощью xclip . . . . .	10
4.8	Добавление SSH key . . . . .	11
4.9	Создание рабочего пространства . . . . .	11
4.10	Выбор шаблона . . . . .	12
4.11	Создание репозитория . . . . .	12
4.12	Созданный репозиторий . . . . .	13
4.13	Переход в каталог курса . . . . .	13
4.14	Копирование ссылки . . . . .	14
4.15	Клонирование репозитория . . . . .	14
4.16	Переход в каталог курса . . . . .	15
4.17	Удаление файлов . . . . .	15
4.18	Создание необходимых каталогов . . . . .	15
4.19	Отправка файлов на сервер . . . . .	15
4.20	Проверка иерархии создания рабочего пространства на GitHub . . . . .	16
4.21	Создание файла . . . . .	16
4.22	Переход в каталог . . . . .	17
4.23	Содержимое “Downloads” . . . . .	17
4.24	Копирование файла в каталог . . . . .	17
4.25	Загрузка файлов на GitHub . . . . .	18
4.26	Добавление файла на GitHub . . . . .	18

# 1 Цель работы

Цель данной работы - изучить идеологию и применение средств контроля версий, а также научиться работать с системой git.

## 2 Задания

1. Настройка GitHub.
2. Базовая настройка Git.
3. Создание SSH-ключа.
4. Создание рабочего пространства и репозитория курса на основе шаблона.
5. Создание репозитория на основе шаблона.
6. Настройка каталога курса
7. Выполнение заданий для самостоятельной работы.

### 3 Теоретическое введение

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется. В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию - сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных. Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зави-

симости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом. Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить. В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным. Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд. Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями. Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

## 4 Выполнение лабораторной работы

### 4.1 Настройка GitHub

Создаю учётную запись на сайте <https://github.com/> и заполняю основные данные. (рис. 4.1).

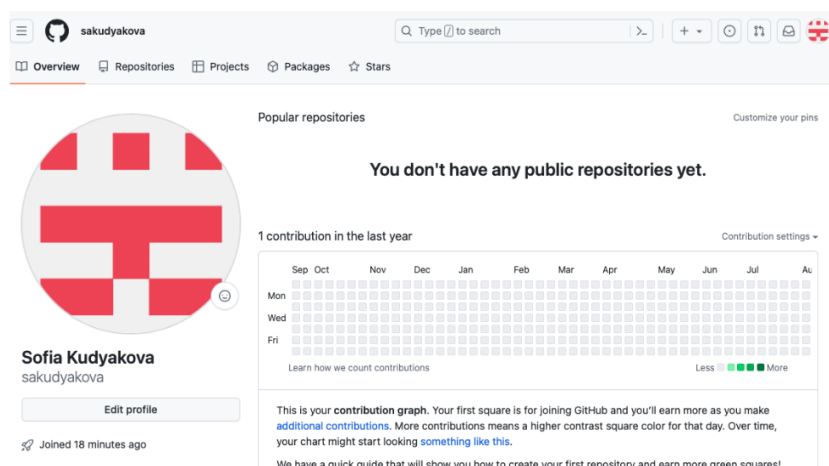
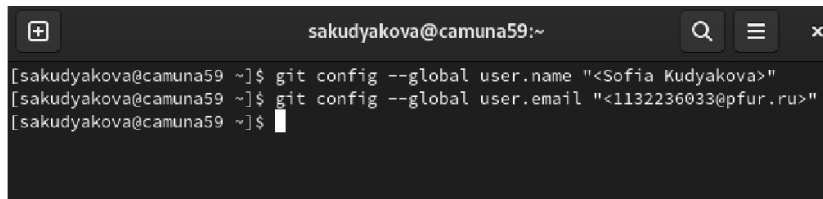


Рис. 4.1: Создание учетной записи GitHub

### 4.2 Базовая настройка Git

Делаю предварительную конфигурацию git. Открываю терминал и ввожу следующие команды, указав имя и email своего репозитория. (рис. 4.2).



A terminal window with a dark background. The title bar shows 'sakudyakova@camuna59:~'. The terminal contains three lines of text: the first line is a prompt '[sakudyakova@camuna59 ~]\$' followed by the command 'git config --global user.name "<Sofia Kudyakova>"'; the second line is a prompt '[sakudyakova@camuna59 ~]\$' followed by the command 'git config --global user.email "<1132236033@pfur.ru>"'; the third line is a prompt '[sakudyakova@camuna59 ~]\$' followed by a cursor.

```
sakudyakova@camuna59:~  
[sakudyakova@camuna59 ~]$ git config --global user.name "<Sofia Kudyakova>"  
[sakudyakova@camuna59 ~]$ git config --global user.email "<1132236033@pfur.ru>"  
[sakudyakova@camuna59 ~]$
```

Рис. 4.2: Предварительная конфигурация git

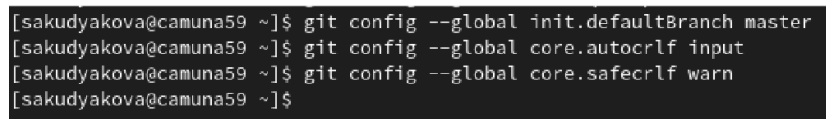
Настраиваю utf-8 в выводе сообщений git (рис. 4.3).

A terminal window with a dark background. The title bar shows 'sakudyakova@camuna59 ~'. The terminal contains two lines of text: the first line is a prompt 'sakudyakova@camuna59 ~]\$' followed by the command 'git config --global core.quotepath false'; the second line is a prompt 'sakudyakova@camuna59 ~]\$' followed by a cursor.

```
sakudyakova@camuna59 ~]$ git config --global core.quotepath false  
sakudyakova@camuna59 ~]$
```

Рис. 4.3: Настройка utf-8 в выводе сообщений git

Задаю имя начальной ветки (будем называть её master). Далее устанавливаю параметры autocrlf и safecrlf (рис. 4.4).

A terminal window with a dark background. The title bar shows 'sakudyakova@camuna59 ~'. The terminal contains four lines of text: the first line is a prompt '[sakudyakova@camuna59 ~]\$' followed by the command 'git config --global init.defaultBranch master'; the second line is a prompt '[sakudyakova@camuna59 ~]\$' followed by the command 'git config --global core.autocrlf input'; the third line is a prompt '[sakudyakova@camuna59 ~]\$' followed by the command 'git config --global core.safecrlf warn'; the fourth line is a prompt '[sakudyakova@camuna59 ~]\$' followed by a cursor.

```
[sakudyakova@camuna59 ~]$ git config --global init.defaultBranch master  
[sakudyakova@camuna59 ~]$ git config --global core.autocrlf input  
[sakudyakova@camuna59 ~]$ git config --global core.safecrlf warn  
[sakudyakova@camuna59 ~]$
```

Рис. 4.4: Создание имени для начальной ветки. Параметры autocrlf и safecrlf

## 4.3 Создание SSH ключа

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый) (рис. 4.5). Ключи сохраняются в каталоге ~/.ssh/

```
[sakudyakova@camuna59 ~]$ ssh-keygen -C"Sofia Kudyakova, 1132236033@pfur.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/sakudyakova/.ssh/id_rsa):
Created directory '/home/sakudyakova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/sakudyakova/.ssh/id_rsa
Your public key has been saved in /home/sakudyakova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:5C+3c+GMP1Aycaez23PcQ7qsS13BBM45CkaVLZW0Nk Sofia Kudyakova, 1132236033@pfur.ru
The key's randomart image is:
+---[RSA 3072]-----+
|
| ..oo+o.|
| o ..+.+. |
| . = o.=+o |
| o+ = .=.E.|
| . S+ + . |
| .. o . .. |
| . o= = oo. |
| oo.B +.o..|
| .+.+o=. .|
+---[SHA256]-----+
```

Рис. 4.5: Создание SSH-ключа

Устанавливаю xclip. (рис. 4.6).

```
[sakudyakova@camuna59 ~]$ sudo dnf -y install xclip
[sudo] password for sakudyakova:
Last metadata expiration check: 0:05:28 ago on Sun 24 Sep 2023 12:13:37 PM EDT.
Dependencies resolved.
=====
Package Architecture Version Repository
Size
-----
Installing:
xclip x86_64 0.13-19.git11cba61.fc38 fedora
37 k
Transaction Summary
-----
Install 1 Package
```

Рис. 4.6: Установка xclip

С помощью xclip копирую сгенерированный ключ. (рис. 4.7).

```
[sakudyakova@camuna59 ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

Рис. 4.7: Копирование с помощью xclip

Далее загружаю сгенерированный открытый ключ. Захожу на GitHub под своей учётной записью и перехожу в меню, выбираю в боковом меню “SSH and GPG keys”. Копирую ключ из локальной консоли в буфер обмена. Вставляю ключ в появившееся на сайте поле и указываю для ключа имя (Title). (рис. 4.8).

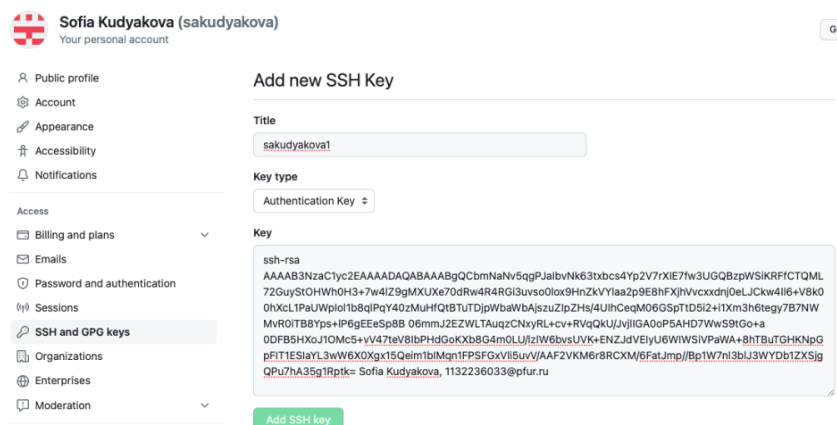


Рис. 4.8: Добавление SSH key

## 4.4 Создание рабочего пространства и репозитория курса на основе шаблона

Открываю терминал и создаю каталог для предмета «Архитектура компьютера», используя команду `mkdir` с ее опцией `-p`, которая, в свою очередь, позволяет рекурсивно создать все директории. С помощью команды `ls` проверяю корректность совершенных действий. (рис. 4.9).

```
[sakudyakova@camuna59 ~]$ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
[sakudyakova@camuna59 ~]$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos work
[sakudyakova@camuna59 ~]$
```

Рис. 4.9: Создание рабочего пространства

## 4.5 Создание репозитория курса на основе шаблона

Создаю репозиторий на основе шаблона через web-интерфейс GitHub. Перехожу на страницу репозитория с шаблоном курса и выбираю “Use this template”. (рис. 4.10).

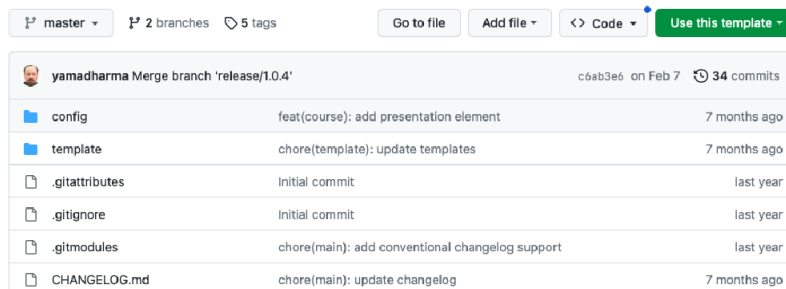


Рис. 4.10: Выбор шаблона

В открывшемся окне задаю имя репозитория (Repository name) study\_2023–2024\_arh-  
pc и создаю репозиторий, нажав на кнопку “Create repository” (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 (\*).

Owner \* / Repository name \*

sakudyakova / study\_2023-2024\_arh-pc

study\_2023-2024\_arh-pc is available.

Great repository names are short and memorable. Need inspiration? How about congenial-spork ?

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.

① You are creating a public repository in your personal account.

**Create repository**

Рис. 4.11: Создание репозитория

Убеждаюсь в том, что репозиторий был создан. (рис. 4.12).

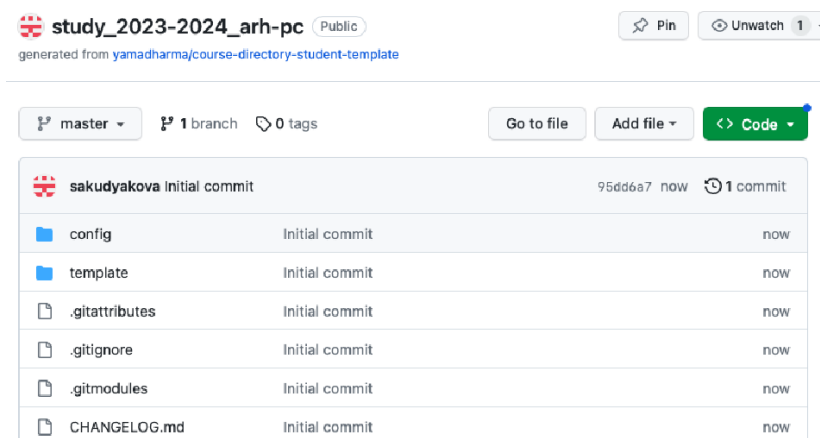


Рис. 4.12: Созданный репозиторий

Далее открываю терминал и перехожу в каталог курса, используя команду `cd`. (рис. 4.13).

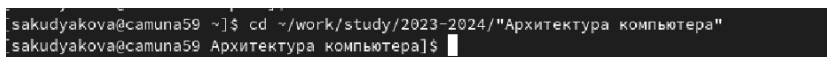


Рис. 4.13: Переход в каталог курса

Копирую ссылку на странице созданного репозитория Code – SSH. (рис. 4.14).

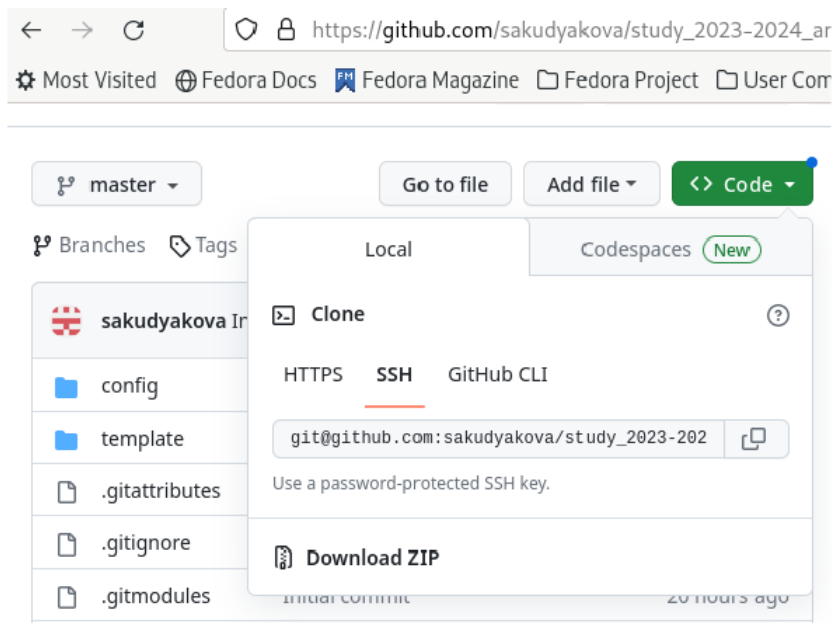


Рис. 4.14: Копирование ссылки

Клонирую созданный репозиторий. (рис. 4.15).

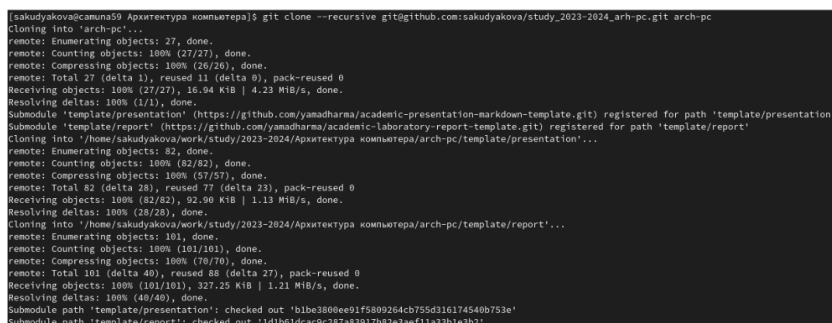


Рис. 4.15: Клонирование репозитория

## 4.6 Настройка каталога курса

Перехожу в каталог курса. (рис. 4.16).

```
[sakudyakova@camuna59 ~]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc
[sakudyakova@camuna59 arch-pc]$
```

Рис. 4.16: Переход в каталог курса

С помощью команды `rm` удаляю лишние файлы, а именно файл “`package.json`”. (рис. 4.17).

```
[sakudyakova@camuna59 arch-pc]$ rm package.json
[sakudyakova@camuna59 arch-pc]$
```

Рис. 4.17: Удаление файлов

Создаю необходимые каталоги. (рис. 4.18).

```
[sakudyakova@camuna59 arch-pc]$ echo arch-pc > COURSE
[sakudyakova@camuna59 arch-pc]$ make
[sakudyakova@camuna59 arch-pc]$
```

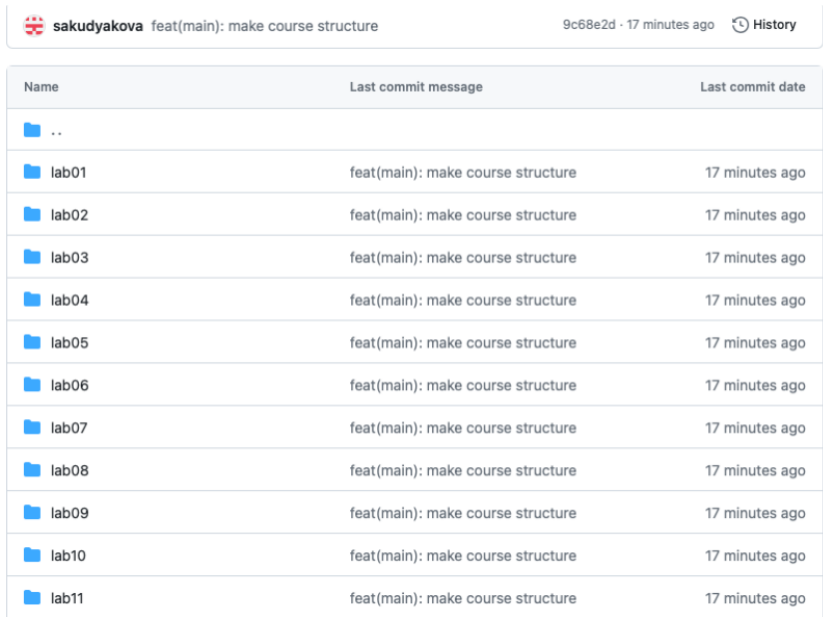
Рис. 4.18: Создание необходимых каталогов

Отправляю файлы на сервер. (рис. 4.19).

```
[sakudyakova@camuna59 arch-pc]$ git add .
[sakudyakova@camuna59 arch-pc]$ git commit -am 'feat(main): make course structure'
[master 9c68e2d] feat(main): make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 labs/lab01/report/report.md
create mode 100644 labs/lab02/presentation/Makefile
create mode 100644 labs/lab02/presentation/image/kulyabov.jpg
create mode 100644 labs/lab02/presentation/presentation.md
create mode 100644 labs/lab02/report/Makefile
create mode 100644 labs/lab02/report/bib/cite.bib
create mode 100644 labs/lab02/report/image/placeimg_800_600_tech.jpg
```

Рис. 4.19: Отправка файлов на сервер

Проверяю корректность создания иерархии рабочего пространства в локальном репозитории и на странице GitHub. (рис. 4.20).



Name	Last commit message	Last commit date
..		
lab01	feat(main): make course structure	17 minutes ago
lab02	feat(main): make course structure	17 minutes ago
lab03	feat(main): make course structure	17 minutes ago
lab04	feat(main): make course structure	17 minutes ago
lab05	feat(main): make course structure	17 minutes ago
lab06	feat(main): make course structure	17 minutes ago
lab07	feat(main): make course structure	17 minutes ago
lab08	feat(main): make course structure	17 minutes ago
lab09	feat(main): make course structure	17 minutes ago
lab10	feat(main): make course structure	17 minutes ago
lab11	feat(main): make course structure	17 minutes ago

Рис. 4.20: Проверка иерархии создания рабочего пространства на GitHub

#Задания для самостоятельной работы

1. Создание отчета по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report).

С помощью команды cd перехожу в каталог labs/lab02/report. Далее с помощью утилиты touch создаю файл для отчета по выполнению второй лабораторной работы. (рис. 4.21).

```
[sakudyakova@camuna59 arch-pc]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab02/report
[sakudyakova@camuna59 report]$ touch Л02_Кудякова_отчет
```

Рис. 4.21: Создание файла

2. Копирование отчетов по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства.



С помощью команды `cd` перехожу в каталог `lab01/report`. (рис. 4.22).

```
[sakudyakova@camuna59 report]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab01/report
```

Рис. 4.22: Переход в каталог

С помощью команды `ls` проверяю, есть ли в подкаталоге “Downloads” файл с отчетом по первой лабораторной работы. (рис. 4.23).

```
[sakudyakova@camuna59 ~]$ ls Downloads
Л01_Кудякова_отчет.pdf
```

Рис. 4.23: Содержимое “Downloads”

Убеждаюсь в том, что файл есть. Далее копирую отчет в каталог `lab01/report` с помощью `cd` и проверяю корректность выполненных действий с помощью `ls`. (рис. 4.24).

```
[sakudyakova@camuna59 ~]$ cp ~/Downloads/Л01_Кудякова_отчет.pdf ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab01/report
[sakudyakova@camuna59 ~]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab01/report
[sakudyakova@camuna59 report]$ ls
Л01_Кудякова_отчет.pdf
Makefile  pandoc  report.md
[sakudyakova@camuna59 report]$
```

Рис. 4.24: Копирование файла в каталог

### 3. Загрузка файлов на GitHub.

С помощью команды `git add` добавляю файлы. Далее сохраняю изменения, используя команду `git commit -m` и добавляя пояснение “Add existing file”. После чего использую команду `git push`, чтобы отправить файлы в репозиторий. (рис. 4.25).

```
[sakudyakova@camuna59 report]$ git add Л01_Кудрякова_отчет.pdf
[sakudyakova@camuna59 report]$ git commit -m "Add existing file"
[master 3d53f10] Add existing file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Л01_Кудрякова_отчет.pdf
[sakudyakova@camuna59 report]$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 931.73 KiB | 5.68 MiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:sakudyakova/study_2023-2024_arh-pc.git
9c68e2d..3d53f10 master -> master
```

Рис. 4.25: Загрузка файлов на GitHub

Захожу на GitHub, чтобы проверить действительно ли загрузились файлы. (рис. 4.26).

study\_2023-2024\_arh-pc / labs / lab01 / report / Add file ...

sakudyakova Add existing file 3d53f10 · 7 minutes ago History

Name	Last commit message	Last commit date
..		
bib	feat(main): make course structure	1 hour ago
image	feat(main): make course structure	1 hour ago
pandoc	feat(main): make course structure	1 hour ago
Makefile	feat(main): make course structure	1 hour ago
report.md	feat(main): make course structure	1 hour ago
Л01_Кудрякова_отчет.pdf	Add existing file	7 minutes ago

Рис. 4.26: Добавление файла на GitHub

Такие же действия выполняю, чтобы добавить отчет по второй лабораторной работе на GitHub.

## 5 Выводы

В ходе данной лабораторной работы я изучила идеологию и применение средств контроля версий, а также научилась работать с системой git.