

Отчёт по лабораторной работе №3

Язык разметки Markdown

Андрюшин Никита Сергеевич

Содержание

1	Цель работы	6
2	Выполнение лабораторной работы	7
3	Задание для самостоятельной работы	22
4	Выводы	28

Список иллюстраций

2.1	Настройка имени и адреса эл. почты	7
2.2	Настройка UTF8 в выводе сообщений Git	7
2.3	Конфигурация имени начальной ветки	7
2.4	Настройка автоконвертации окончаний строк	8
2.5	Настройка вывода предупреждений о необратимых преобразованиях из CRLF	8
2.6	Генерация ssh ключа	9
2.7	Внешний вид сайта GitHub	10
2.8	Вкладка настроек GitHub	11
2.9	Раздел настроек “SSH and GPG keys”	12
2.10	Поле ввода ключа в окне добавления нового SSH ключа	13
2.11	Считывание и копирование публичного ключа	13
2.12	Заполнение всех необходимых полей для добавления ключа	14
2.13	Создание каталога “Архитектура компьютера”	14
2.14	Страница шаблона на GitHub	15
2.15	Создание репозитория на основе шаблона	16
2.16	Перемещение в папку, куда скачается репозиторий	16
2.17	Ссылка на наш репозиторий	17
2.18	Скачивание репозитория командой “git clone”	17
2.19	Переход в клонированный каталог	18
2.20	Удаление файла командой “rm”	18
2.21	Создание файла и запись в него строки	18
2.22	Добавление каталога для отправки на сервер	18
2.23	Сохранение изменений и указание комментария	19
2.24	Отправка репозитория на сервер GitHub	19
2.25	Файлы на компьютере	20
2.26	Файлы на сервере	21
3.1	Начальный экран LibreOffice	22
3.2	Выбор места для сохранения файла и расширения docx	23
3.3	Сохранение отчёта в формате pdf	24
3.4	Проверка наличия отчёта в папке	25
3.5	Копирование отчёта по первой лабораторной работе в папку lab01	25
3.6	Добавление каталога для отправки на сервер	26
3.7	Сохранение изменений и указание комментария	26
3.8	Загрузка новых файлов на GitHub	26

3.9 Вид обновлённого репозитория на GitHub	27
--	----

Список таблиц

1 Цель работы

Цели работы - приобрести навыки по работе с системой контроля версий git и научиться пользоваться онлайн платформой GitHub.

2 Выполнение лабораторной работы

Перед началом работы с git его необходимо предварительно настроить. Для этого введем в терминал следующие команды (рис. 2.1):

```
nsandryushin@nsandryushin:~$ git config --global user.name "Nikita Andryushin"
nsandryushin@nsandryushin:~$ git config --global user.email "mega_nikitos111@mail.ru"
nsandryushin@nsandryushin:~$
```

Рис. 2.1: Настройка имени и адреса эл. почты

Введя их, мы задали имя и электронный адрес почты пользователя. Теперь введем следующую команду (рис. 2.2):

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

Рис. 2.2: Настройка UTF8 в выводе сообщений Git

Благодаря ней, мы настроили вывод сообщений git в кодировке utf8. Теперь мы должны задать имя для начальной ветки. Мы назовем ее master (рис. 2.3):

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

Рис. 2.3: Конфигурация имени начальной ветки

Теперь введем следующую команду (рис. 2.4)

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

Рис. 2.4: Настройка автоконвертации окончаний строк

Таким образом мы указываем git автоматически конвертировать CRLF окончания строк в LF во время commit'a. Теперь выполним вот эту команду (рис. 2.5). Она будет отвечать за то, чтобы печатать предупреждение в случае, если преобразования из CRLF будут необратимыми.

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

Рис. 2.5: Настройка вывода предупреждений о необратимых преобразованиях из CRLF

Для того, чтобы сервер мог идентифицировать пользователя, необходимо сгенерировать несколько ssh ключей. Начнем с генерации открытого ключа. Для этого мы введем следующую команду, указав имя, фамилию пользователя и его адрес электронной почты в качестве аргумента (рис. 2.6)


```

nsandryushin@nsandryushin:~$ ssh-keygen -C "Nikita Andryushin mega_nikitos111@mail.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/nsandryushin/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/nsandryushin/.ssh/id_rsa
Your public key has been saved in /home/nsandryushin/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:4IpF31Lt+739sdREi5rEM0U6Fpj0fmLjAertL24HtSo Nikita Andryushin mega_nikitos111@mail.ru
The key's randomart image is:
+---[RSA 3072]---+
|           .           |
|          o o          |
|         .o + .        |
|        . oo. *        |
|       . o. ooS+ + .   |
|      . oo.o. . + +   |
|     ...+..= o +   o.  |
|    E =.o+.=. . .+   |
|   +.+ . . . . .o.o*  |
+-----[SHA256]-----+
nsandryushin@nsandryushin:~$

```

Рис. 2.6: Генерация ssh ключа

Зайдем на сайт GitHub (рис. 2.7)

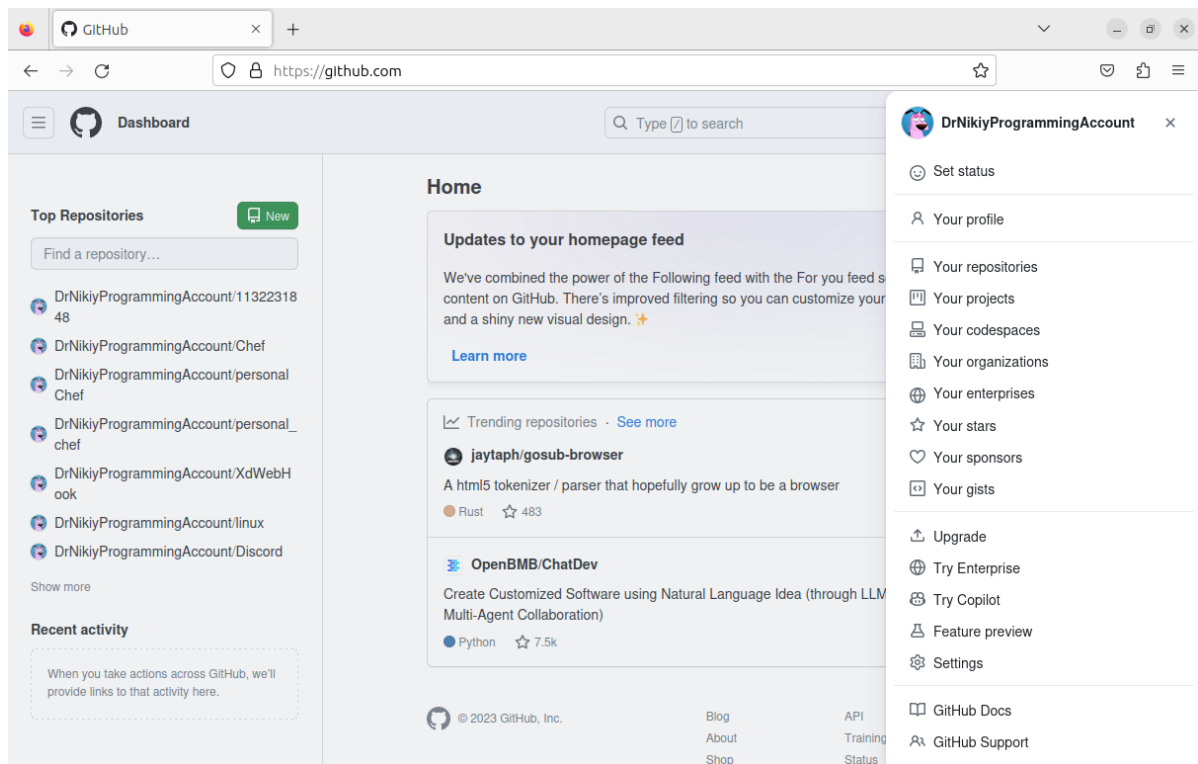


Рис. 2.7: Внешний вид сайта GitHub

Здесь мы должны перейти в пункт settings (рис. 2.8)

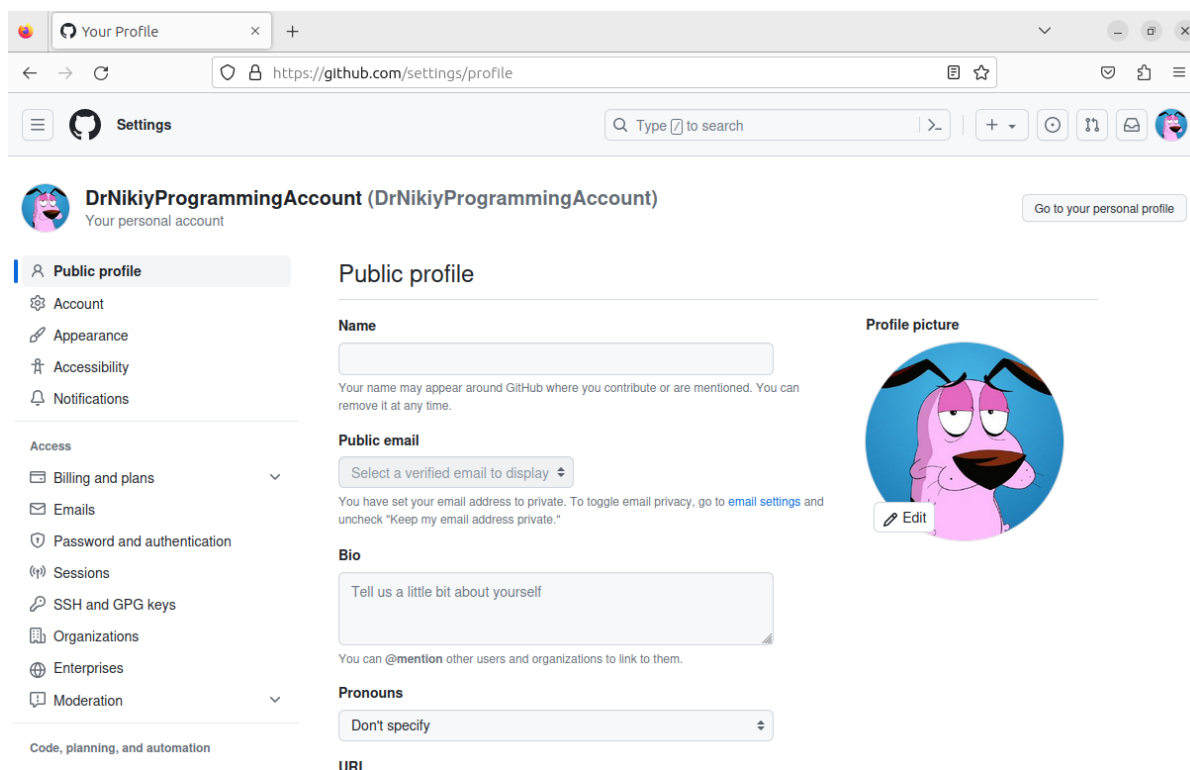


Рис. 2.8: Вкладка настроек GitHub

Находим раздел SSH and GPG keys и нажимаем New SSH key (рис. 2.9)

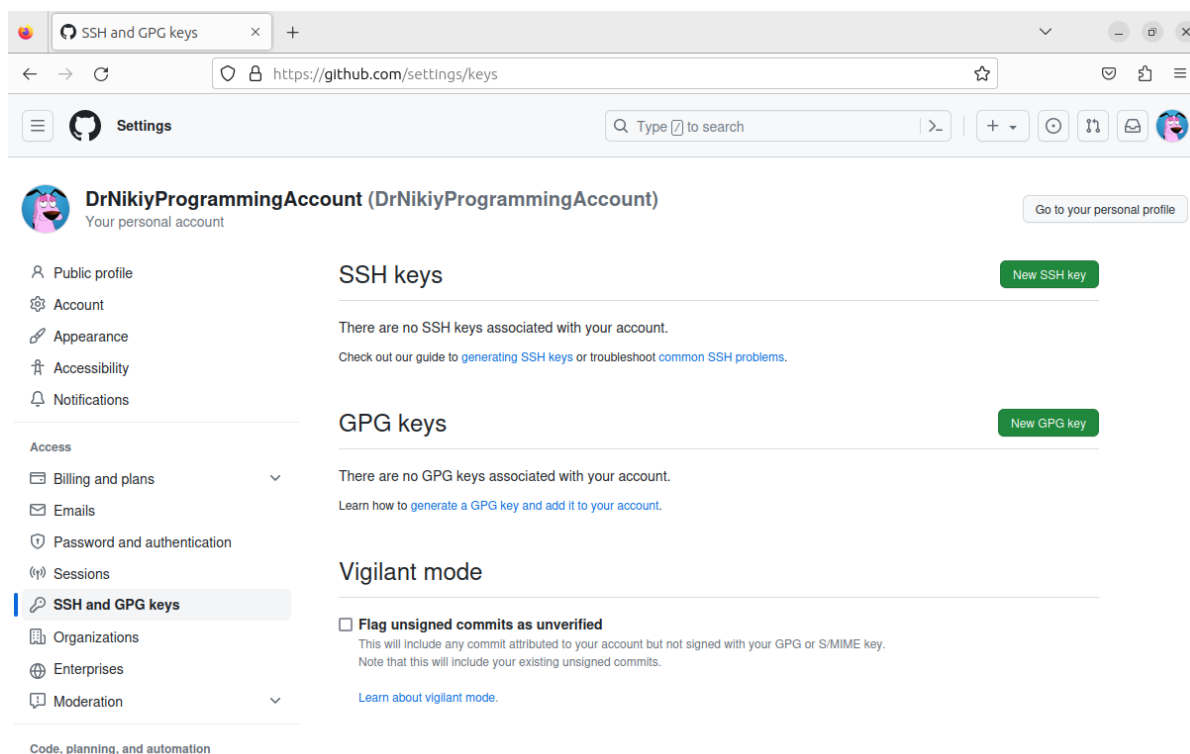


Рис. 2.9: Раздел настроек “SSH ang GPG keys”

В предложенное поле “Key” нам необходимо вставить ключ, который мы только что сгенерировали (рис. 2.10)

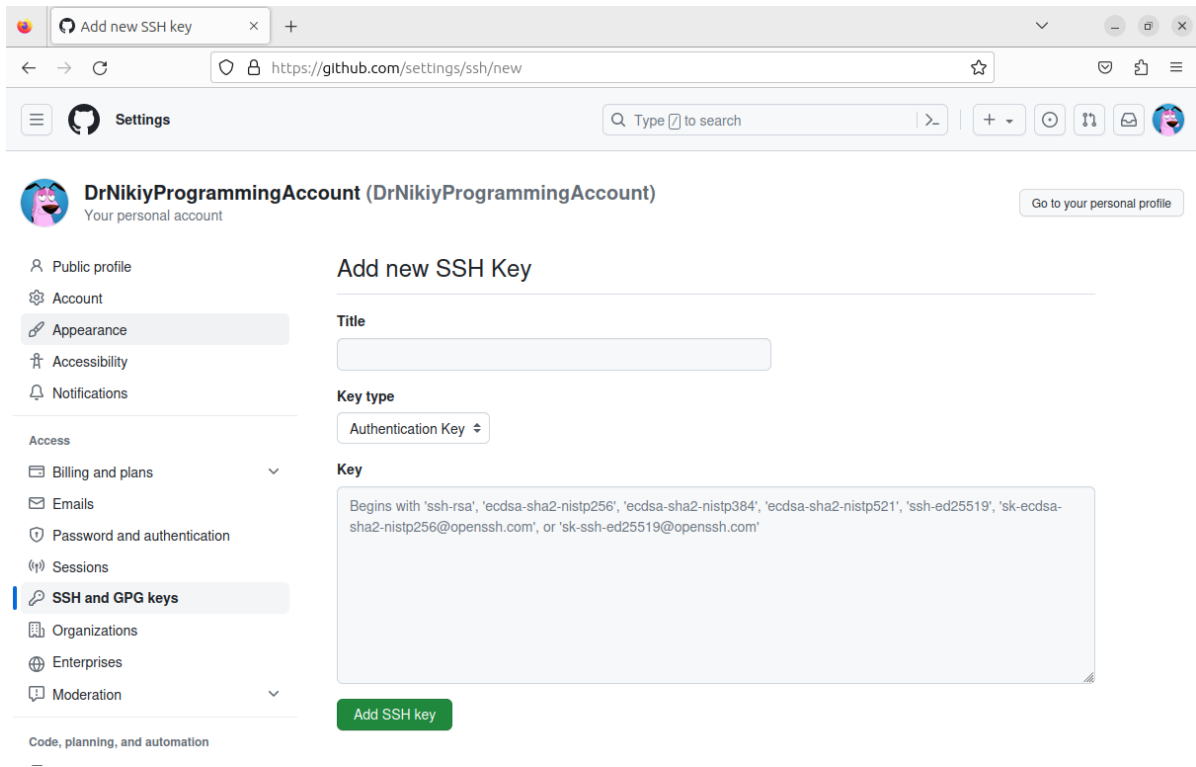


Рис. 2.10: Поле ввода ключа в окне добавления нового SSH ключа

Для того, чтобы скопировать ключ для последующей вставки, нам необходимо ввести следующую команду (рис. 2.11)

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

Рис. 2.11: Считывание и копирование публичного ключа

Команда “cat” прочитает данные из файла `id_rsa.pub`, а команда `xclip` вгрузит их в буфер обмена. Остается лишь вставить содержимое буфера обмена, то есть наш ключ, в предложенное поле “Key” и указать имя ключа в поле “Title” (рис. 2.12)

Add new SSH Key

Title

Key

Key type

Authentication Key ↕

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCAJ2eOAPA8y5oub3WSyc42AnbxprQd2Qn8ceztshq2l8f+VyJsG2jf2Fa85jT
4TxJHEcZNT1+PjR5P0KDDu3Yo8mitUP7LvzPmo2RghFRwX61CRdruG1ms3ravPdZ72eduM/vF
/mY8OY1fdP6fEQfBC6Fxfhj3iSlfj3mDyJV8Q61NrhssUtbpvlbeuRHGjP0B8klNLwpat+AMZS4R10ZowaqA8ooJHVPPR6l
KaipKiyjD7PtElitL/21pZcRwvXs10M/ucWX6v4H4DvKnH5gMIU515tVj5z8nCen
/2bRJJHcht+P+Di6rGjE6WV4NOJThYhTwJCmkOxRECHB8Sc88q+BprlaP6Rxuh
/OTPvmmnOQGquJhSLs7vNr7d7b+1Calbq4mOdle8YfwEozdUIFP3cRy4V1yMoym7+gTnt9xmsvjXG4q5NVJSbZSm6+K
jrga9xHx2Q49LtcU6c3BuhOTMMcf7BZZ1bfZm+1afPaEZalrR2ClljuWWzIZqx34HrkbEk= Nikita Andryushin
mega_nikitos111@mail.ru
```

Add SSH key

Рис. 2.12: Заполнение всех необходимых полей для добавления ключа

Теперь нам необходимо организовать наше рабочее пространство. Для этого создадим каталог “Архитектура компьютера” по следующему адресу (рис. 2.13)

```
nsandryushin@nsandryushin:~$ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
nsandryushin@nsandryushin:~$
```

Рис. 2.13: Создание каталога “Архитектура компьютера”

После этого нам нужно будет создать репозиторий. Мы будем его создавать на основе шаблона, который находится по следующему адресу: <https://github.com/yamadharma/course-directory-student-template> (рис. 14)

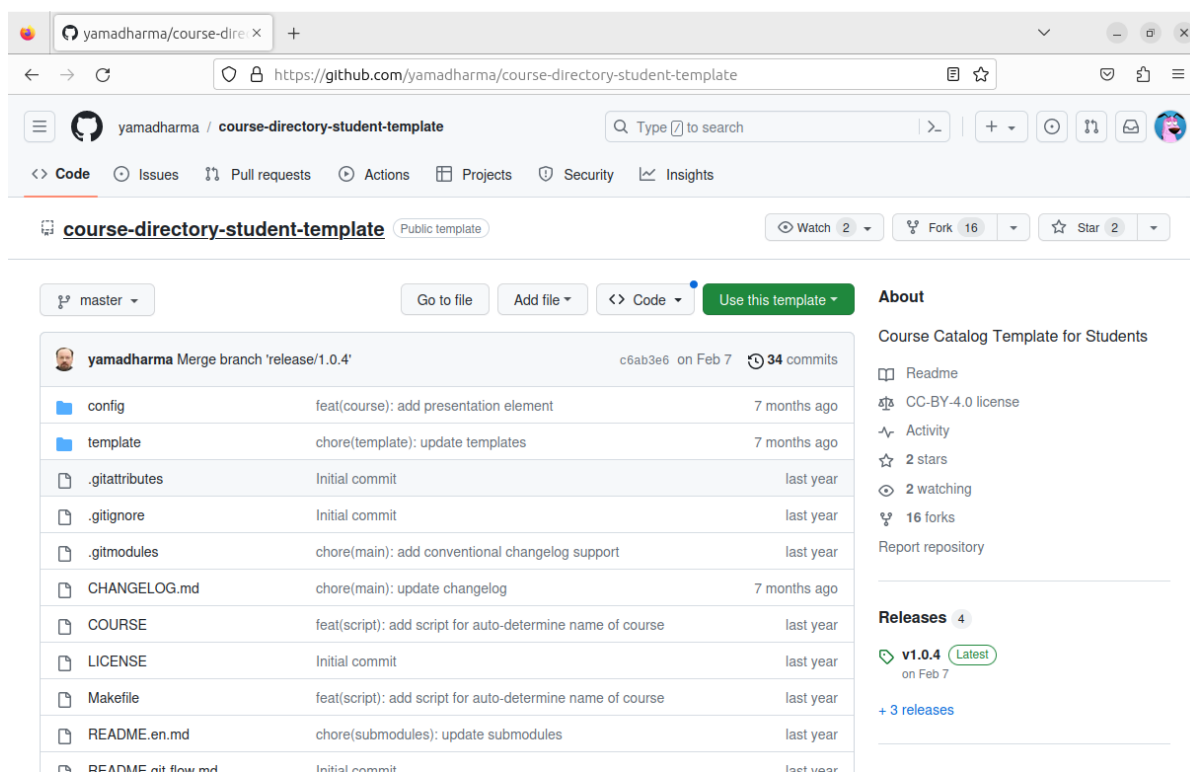


Рис. 2.14: Страница шаблона на GitHub

Нажимаем на кнопку “Use this template”, и в предложенных опциях выбираем “Create new repository”. Далее нас переносит на следующую страницу (рис. 2.15). Здесь задаём имя нашего репозитория. Он будет называться так: study_2023-2024_arh-ps

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 ***

DrNikiyProgrammingAccount / study_2023-2024_arh-pc

✔ study_2023-2024_arh-pc is available.

Great repository names are short and memorable. Need inspiration? How about **potential-sniffle** ?

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. 2.15: Создание репозитория на основе шаблона

Теперь нам нужно клонировать репозиторий на наш компьютер. Для этого перейдем в папку, в которую мы хотим скопировать репозиторий. В нашем случае это ранее созданная папка (рис. 2.16)

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

Рис. 2.16: Перемещение в папку, куда скачается репозиторий

Теперь перейдем непосредственно к клонированию. Для этого воспользуемся командой `git clone`, в аргументе указав ссылку на репозиторий (рис. 2.17). Ссылку можно найти при нажатии на кнопку код на странице нашего репозитория (рис. 2.18)

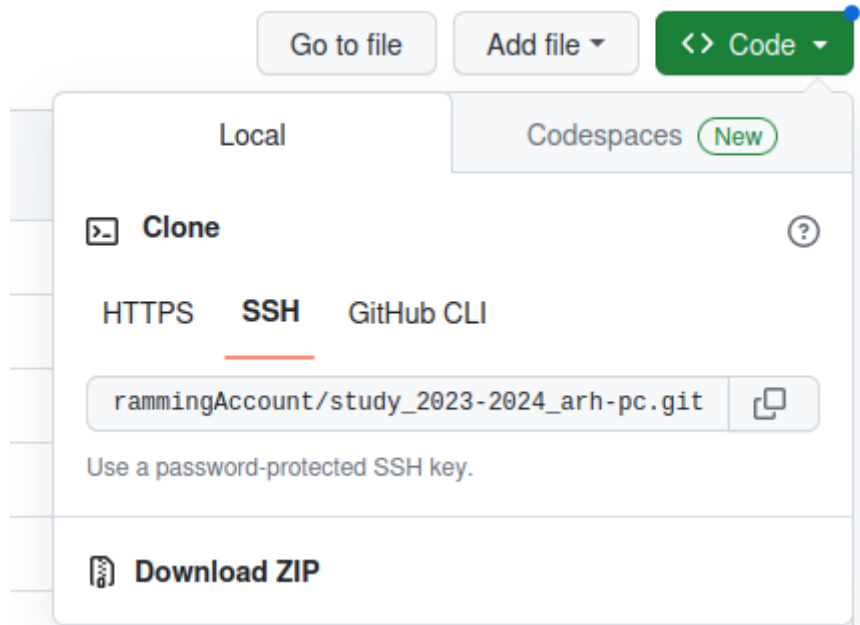


Рис. 2.17: Ссылка на наш репозиторий

```
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера$ git clone --recursive git@github.com:DrNikiyProgrammingAccount/study_2023-2024_arh-pc.git arch-pc
Клонирование в «arch-pc»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 0
Получение объектов: 100% (27/27), 16.94 КиБ | 8.47 МиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharm/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharm/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/nsandryushin/work/study/2023-2024/Архитектура компьютера/arch-pc/template/presentation»...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
```

Рис. 2.18: Скачивание репозитория командой “git clone”

Теперь перейдем к настройке клонированного каталога. Для начала перейдем в него с помощью команды “cd” (рис. 2.19).

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

Рис. 2.19: Переход в клонированный каталог

Удалим с помощью команды “rm” лишний файл (рис. 2.20)

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

Рис. 2.20: Удаление файла командой “rm”

Теперь создадим необходимые файлы. В нашем случае, это будет файл COURSE. Чтобы его создать, мы воспользуемся командой echo, которая запишет в файл строку “arch-pc” и автоматически создаст его, т.к. этого файла раньше не существовало (рис. 2.21)

```
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE  
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ make  
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 2.21: Создание файла и запись в него строки

Теперь нам остается лишь отправить файлы на сервер. Для этого с помощью команды git add мы добавим каталоги, которые должны отправляться на сервер. В качестве аргумента мы возьмем точку, которая укажет на то, что мы должны отправить на сервер все файлы и каталоги, которые по иерархии находятся ниже нашего текущего расположения (рис. 2.22)

```
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .  
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 2.22: Добавление каталога для отправки на сервер

Теперь с помощью команды `git commit` мы сохраним изменения и укажем комментарий, в котором будет поясняться, какие изменения мы сделали. В данном случае в комментарии мы напишем, что создали структуру курса (рис. 2.23)

```
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit -am "feat(main): make course structure"
[master 59cd95e] 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
```

Рис. 2.23: Сохранение изменений и указание комментария

Теперь нам осталось окончательно загрузить изменения на сервер. Для этого мы воспользуемся командой `git push` (рис. 2.24)

```
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.14 КиБ | 2.46 МиБ/с, готово.
Всего 35 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:DrNikiyProgrammingAccount/study_2023-2024_arh-pc.git
 1b7cc74..59cd95e master -> master
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 2.24: Отправка репозитория на сервер GitHub

Остается лишь проверить, сохранились ли файлы на сервере. Сравним файлы, которые находятся на GitHub с теми файлами, что находятся на нашем компьютере (рис. 2.25 и рис. 2.26)

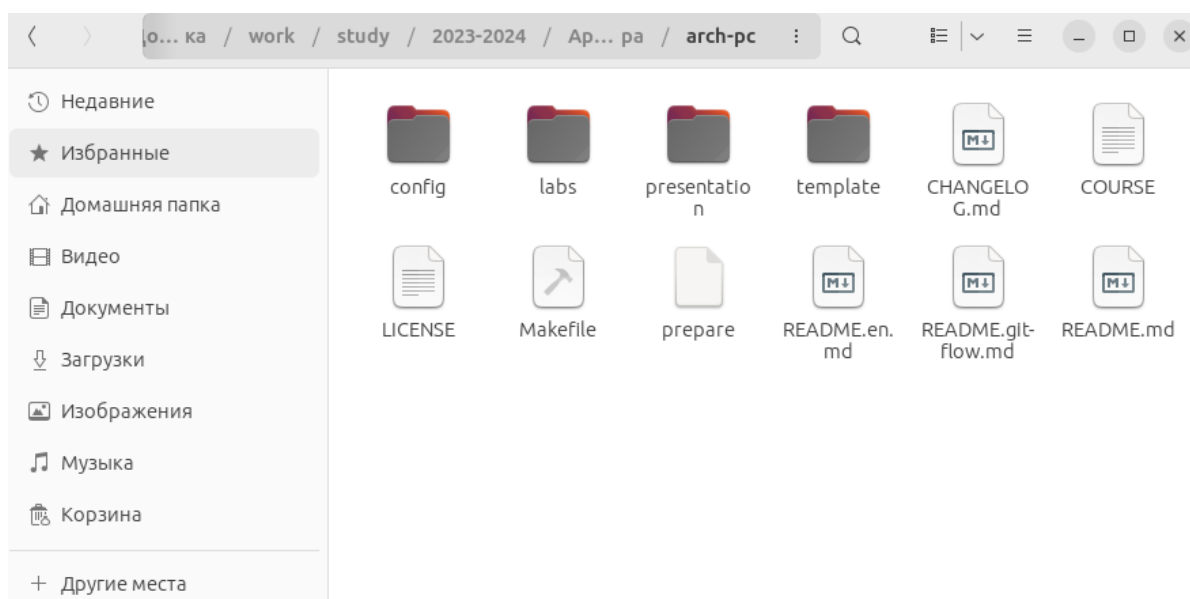


Рис. 2.25: Файлы на компьютере


 DrNikiyProgrammingAccount feat(main): make course structure 59cd95e 2 minutes ago 🕒 2 commits		
📁 config	Initial commit	10 minutes ago
📁 labs	feat(main): make course structure	2 minutes ago
📁 presentation	feat(main): make course structure	2 minutes ago
📁 template	Initial commit	10 minutes ago
📄 .gitattributes	Initial commit	10 minutes ago
📄 .gitignore	Initial commit	10 minutes ago
📄 .gitmodules	Initial commit	10 minutes ago
📄 CHANGELOG.md	Initial commit	10 minutes ago
📄 COURSE	feat(main): make course structure	2 minutes ago
📄 LICENSE	Initial commit	10 minutes ago
📄 Makefile	Initial commit	10 minutes ago
📄 README.en.md	Initial commit	10 minutes ago
📄 README.git-flow.md	Initial commit	10 minutes ago
📄 README.md	Initial commit	10 minutes ago
📄 prepare	feat(main): make course structure	2 minutes ago

Рис. 2.26: Файлы на сервере

Как видим, все совпало.

3 Задание для самостоятельной работы

Теперь приступим к выполнению самостоятельной работы. Для начала мы создадим файл отчета для нашей лабораторной работы в папке labs/lab02/report с помощью LibreOffice (рис. 3.1, рис. 3.2, рис. 3.3 и рис. 3.4)

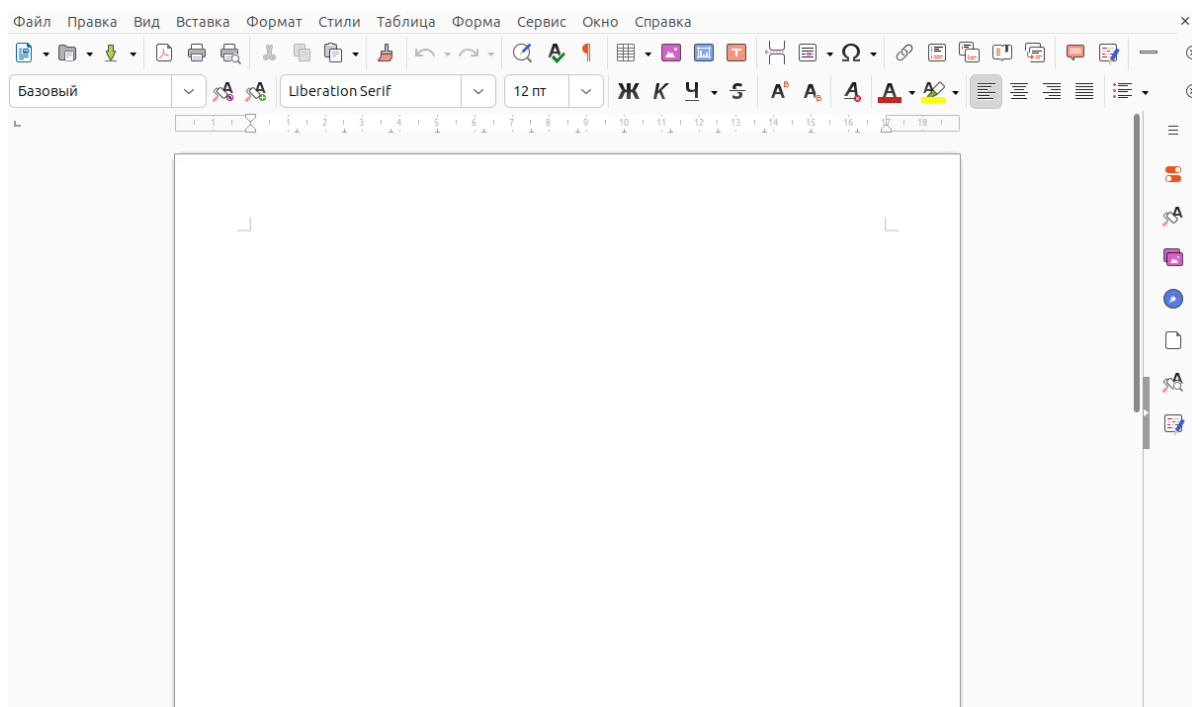


Рис. 3.1: Начальный экран LibreOffice

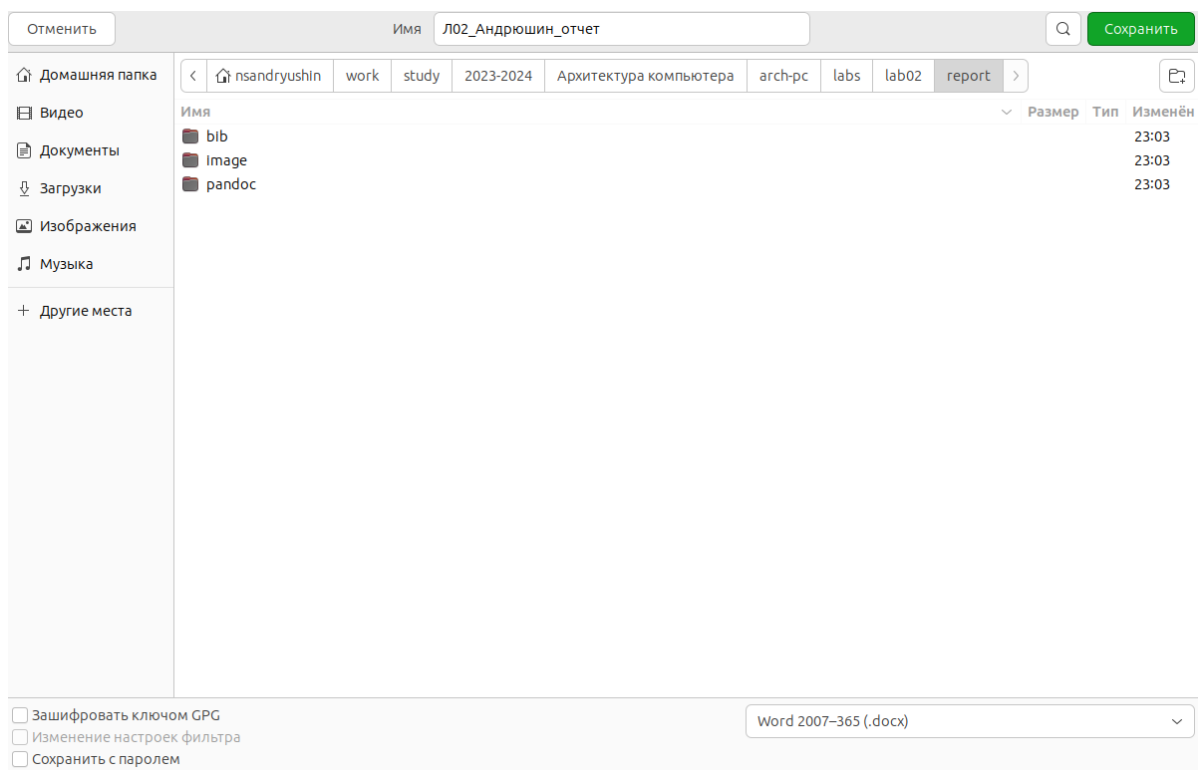


Рис. 3.2: Выбор места для сохранения файла и расширения docx

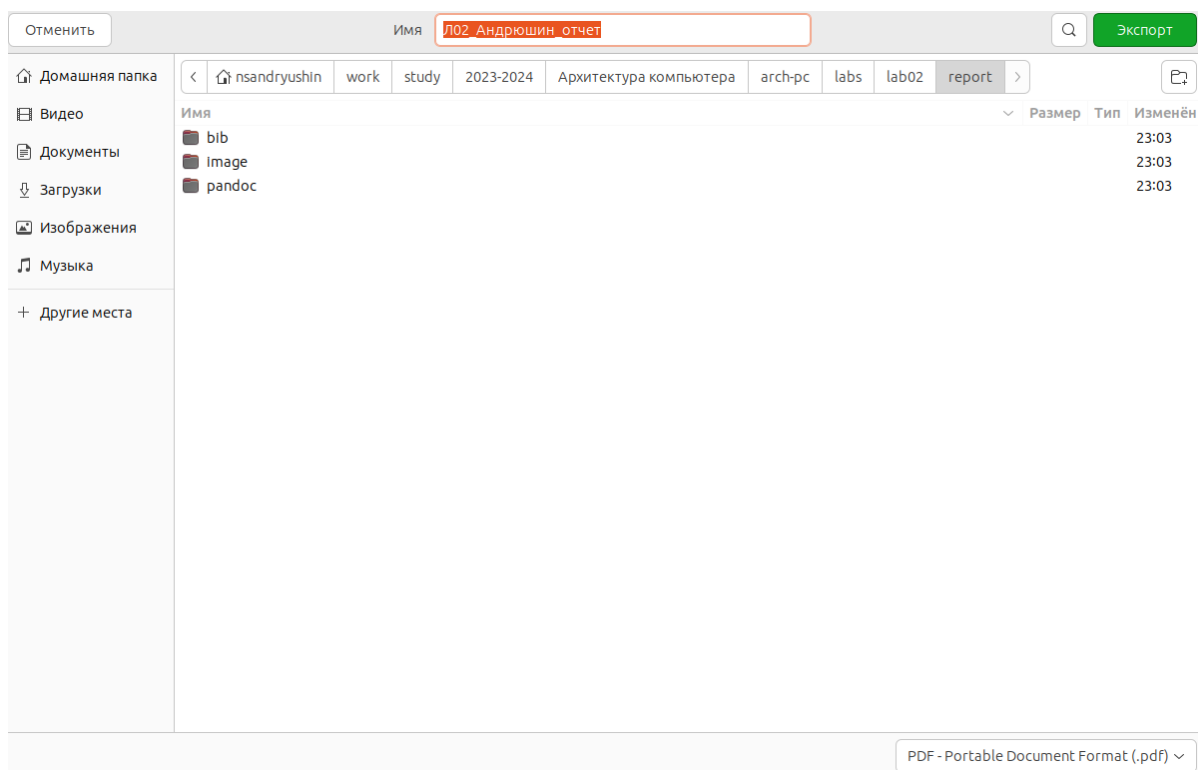


Рис. 3.3: Сохранение отчёта в формате pdf

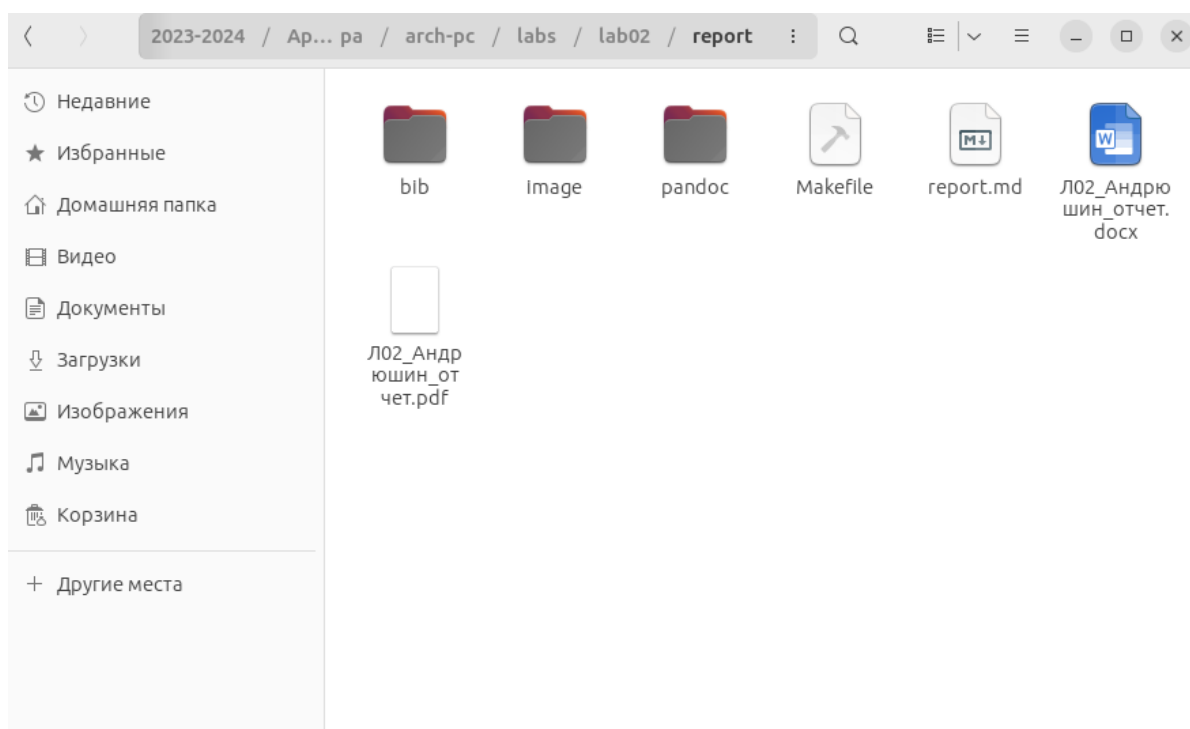


Рис. 3.4: Проверка наличия отчёта в папке

После этого скопируем отчет по нашей предыдущей лабораторной работе в соответствующую папку созданного нами рабочего пространства, то есть в папку `labs/lab01/report`. Для копирования воспользуемся командой “`cp`” (рис. 3.4)

```
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ cp ~/Документы/Л01_Андрюшин_отчет.docx labs/lab01/report/
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ cp ~/Документы/Л01_Андрюшин_отчет.pdf labs/lab01/report/
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 3.5: Копирование отчёта по первой лабораторной работе в папку `lab01`

Теперь нам осталось лишь загрузить изменения на GitHub. Для этого нам нужно повторить те шаги, которые мы выполнили в предыдущих пунктах лабораторной работы, а именно, воспользоваться командой “`git add .`” для того, чтобы указать, что мы хотим сохранить изменения во всех файлах, находящихся в нашем каталоге (рис. 3.5)

```
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 3.6: Добавление каталога для отправки на сервер

После этого с помощью команды “git commit” мы укажем комментарий и сохраним изменения. В комментарии мы укажем, что мы загрузили первую и вторую лабораторные работы (рис. 3.6)

```
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit -am "feat(main): added first 2 labs"
[master ddb0b52] feat(main): added first 2 labs
4 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Л01_Андрюшин_отчет.docx
create mode 100644 labs/lab01/report/Л01_Андрюшин_отчет.pdf
create mode 100644 labs/lab02/report/Л02_Андрюшин_отчет.docx
create mode 100644 labs/lab02/report/Л02_Андрюшин_отчет.pdf
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 3.7: Сохранение изменений и указание комментария

После этого вводим команду “git push” для того, чтобы загрузить наши файлы на GitHub (рис. 3.7)

```
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 17, готово.
Подсчет объектов: 100% (15/15), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (11/11), готово.
Запись объектов: 100% (11/11), 2.58 МиБ | 2.64 МиБ/с, готово.
Всего 11 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:DrNikiyProgrammingAccount/study_2023-2024_arh-pc.git
 59cd95e..ddb0b52 master -> master
nsandryushin@nsandryushin:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 3.8: Загрузка новых файлов на GitHub

Остается лишь проверить, правильно ли мы все загрузили. Для этого посмотрим время обновления файлов в папке labs в GitHub (рис. 3.8)


















 DrNikiyProgrammingAccount feat(main): added 2 first labs 05c1a27 1 minute ago  4 commits		
 config	Initial commit	5 days ago
 labs	feat(main): added 2 first labs	1 minute ago
 presentation	feat(main): make course structure	5 days ago
 template	Initial commit	5 days ago
 .gitattributes	Initial commit	5 days ago
 .gitignore	Initial commit	5 days ago
 .gitmodules	Initial commit	5 days ago
 CHANGELOG.md	Initial commit	5 days ago
 COURSE	feat(main): make course structure	5 days ago
 LICENSE	Initial commit	5 days ago
 Makefile	Initial commit	5 days ago
 README.en.md	Initial commit	5 days ago
 README.git-flow.md	Initial commit	5 days ago
 README.md	Initial commit	5 days ago
 prepare	feat(main): make course structure	5 days ago

Рис. 3.9: Вид обновлённого репозитория на GitHub

4 Выводы

В результате выполнения лабораторной работы появились практические навыки работы с системой контроля версий Git, была произведена её первоначальная настройка в linux. Было изучено, как создавать репозитории, сохранять изменения и добавлять к ним комментарии, а также как выгружать файлы на сервер. Были приобретены навыки работы с платформой GitHub