**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**по лабораторной работе № 2**

*дисциплина: Архитектура компьютера*

Студент: Калашникова Дарья

Группа: НПИбд-01-24

**МОСКВА**

2024 г.

**Содержание**

[Цель работы 3](#_Toc178376079)

[Выполнение лабораторной работы 4](#_Toc178376080)

[2.4.2 Базовая настройка git. 4](#_Toc178376081)

[2.4.3 Создание SSH ключа. 5](#_Toc178376082)

[2.4.4. Создание рабочего пространства и репозитория курса на основе шаблона. 7](#_Toc178376083)

[2.4.5. Создание репозитория на основе шаблона. 8](#_Toc178376084)

[2.4.6. Настройка каталога курса. 11](#_Toc178376085)

[2.5. Задание для самостоятельной работы 14](#_Toc178376086)

[Выводы 17](#_Toc178376087)

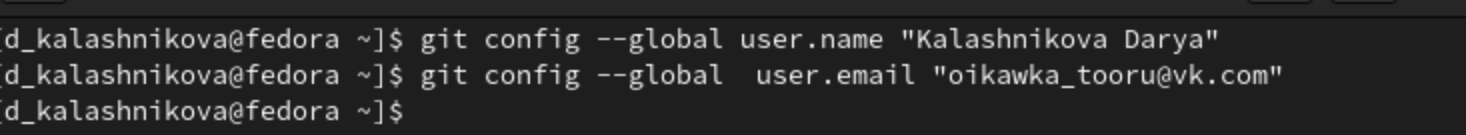
# Цель работы

Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.

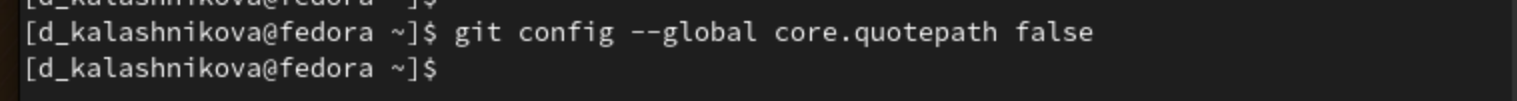
# Выполнение лабораторной работы

## 2.4.2 Базовая настройка git.

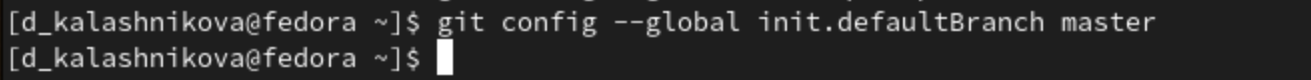
Перед началом работы с git его необходимо предварительно настроить. Для этого введем в терминал следующие команды:

 Рис. 1: Настройка имени и адреса эл. Почты.

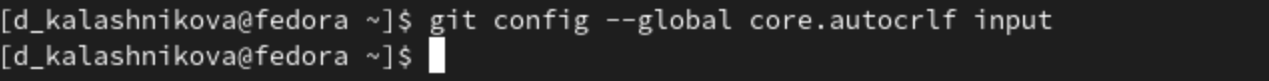
Введя их, мы задали имя и электронный адрес почты пользователя. Теперь введем следующую команду:

 Рис. 2: Настройка UTF8 в выводе сообщений Git.

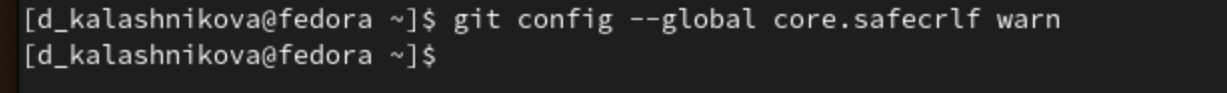
Благодаря ней, мы настроили вывод сообщений git в кодировке utf8. Теперь мы должны задать имя для начальной ветки. Мы назовем ее master:

 Рис. 3: Конфигурация имени начальной ветки.

Теперь введем следующую команду:

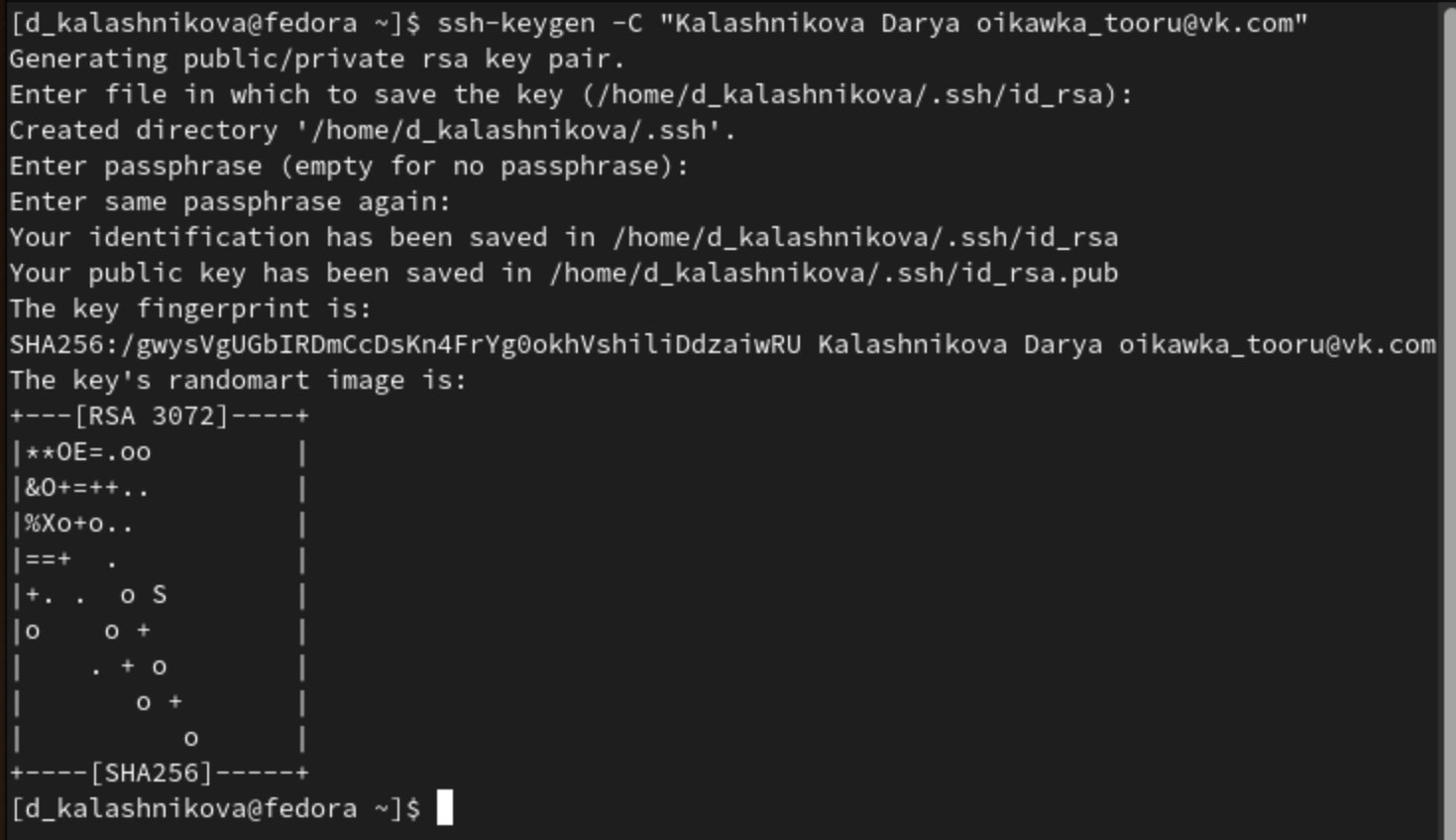
 Рис. 4: Настройка автоконвертации окончаний строк.

Таким образом мы указываем git автоматически конвертировать CRLF окончания строк в LF во время commit'а. Теперь выполним вот эту команду (рис. 5). Она будет отвечать за то, чтобы печатать предупреждение в случае, если преобразования из CRLF будут необратимыми.

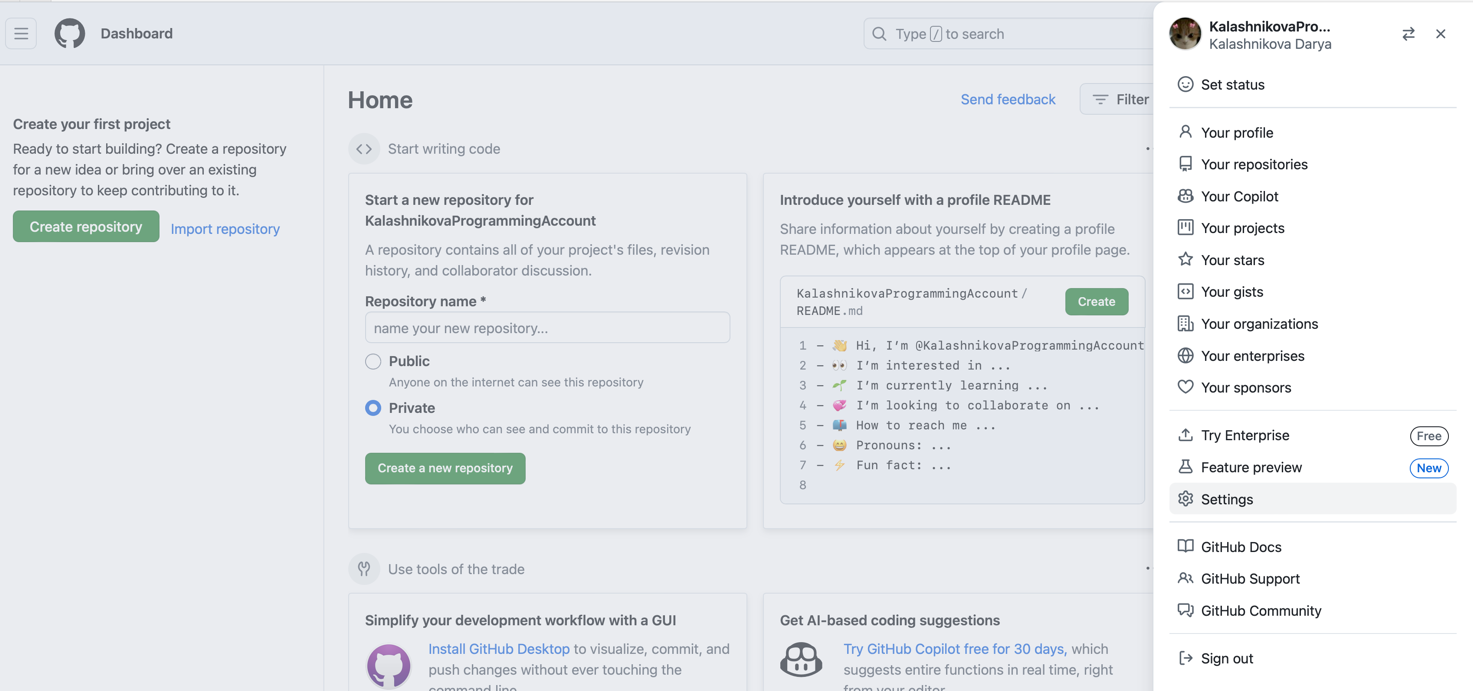
 Рис. 5: Настройка вывода предупреждений о необратимых преобразованиях из CRLF.

## 2.4.3 Создание SSH ключа.

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. 6: Генерация ssh ключа.

Зайдем на сайт GitHub:

Рис.7: Перейдём в пункт settings.

Находим раздел SSH and GPG keys и нажимаем New SSH key:

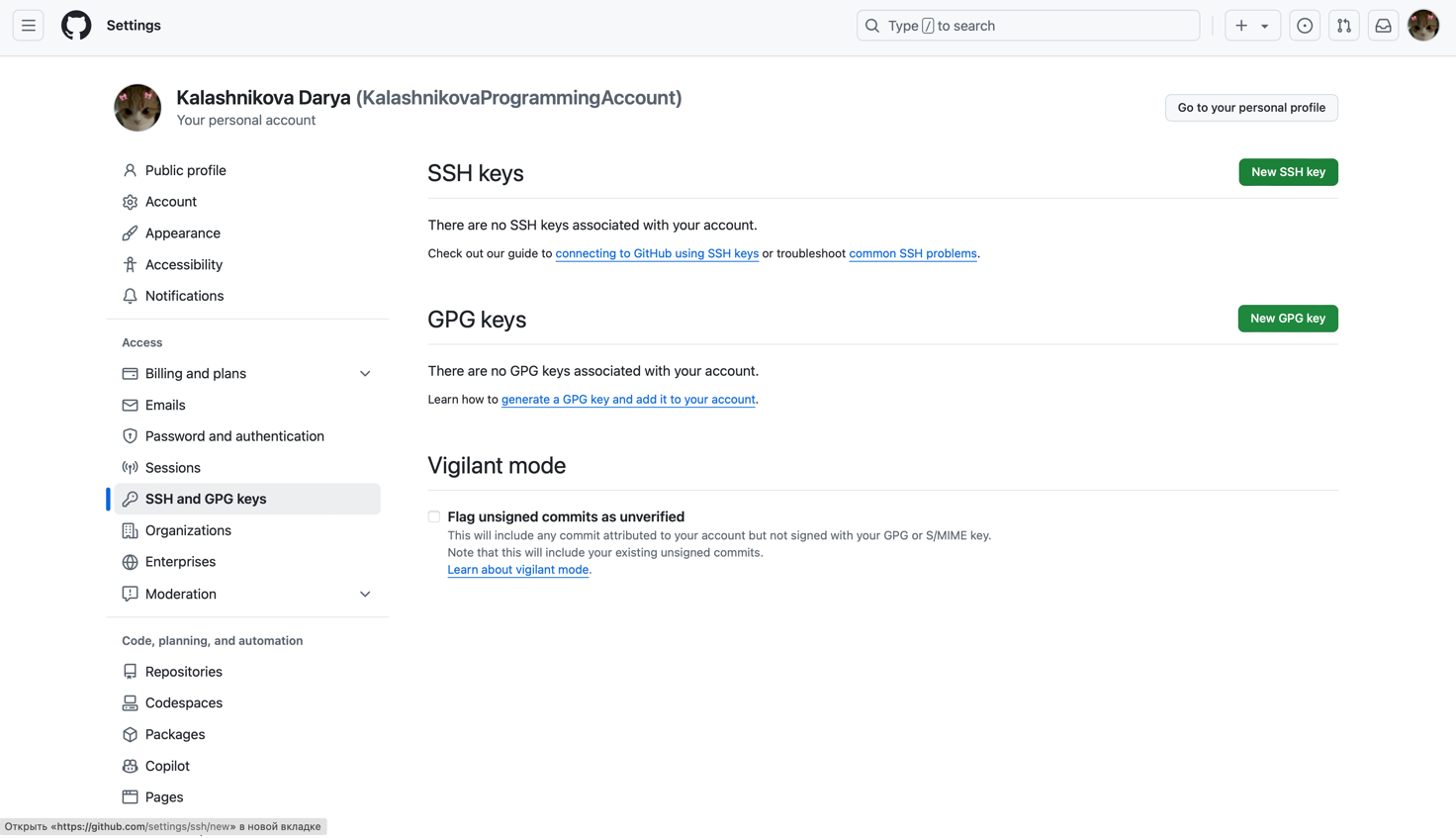
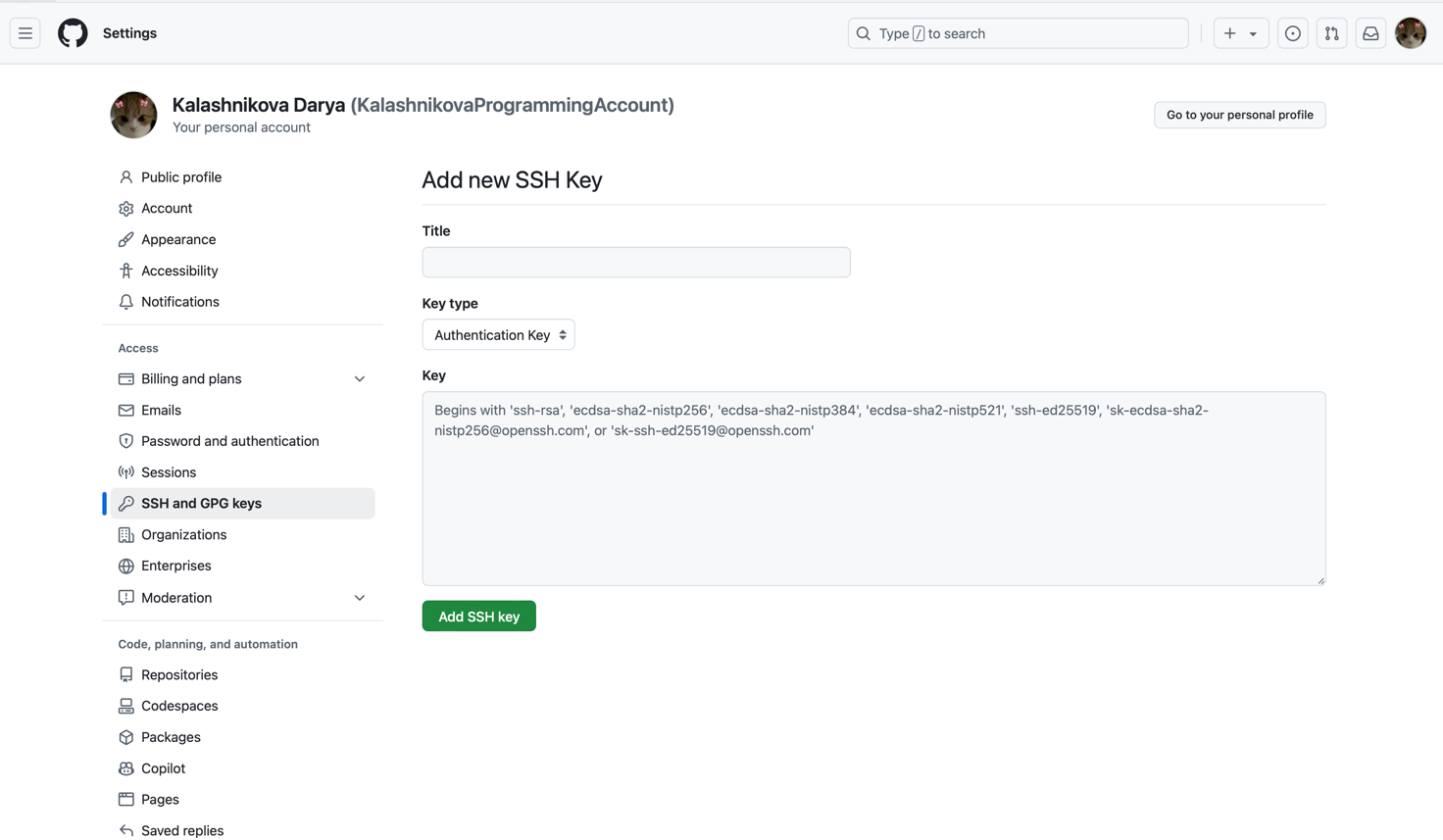


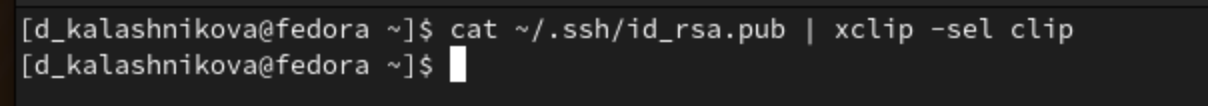
Рис. 8: Раздел настроек "SSH ang GPG keys".

В предложенное поле "Key" нам необходимо вставить ключ, который мы

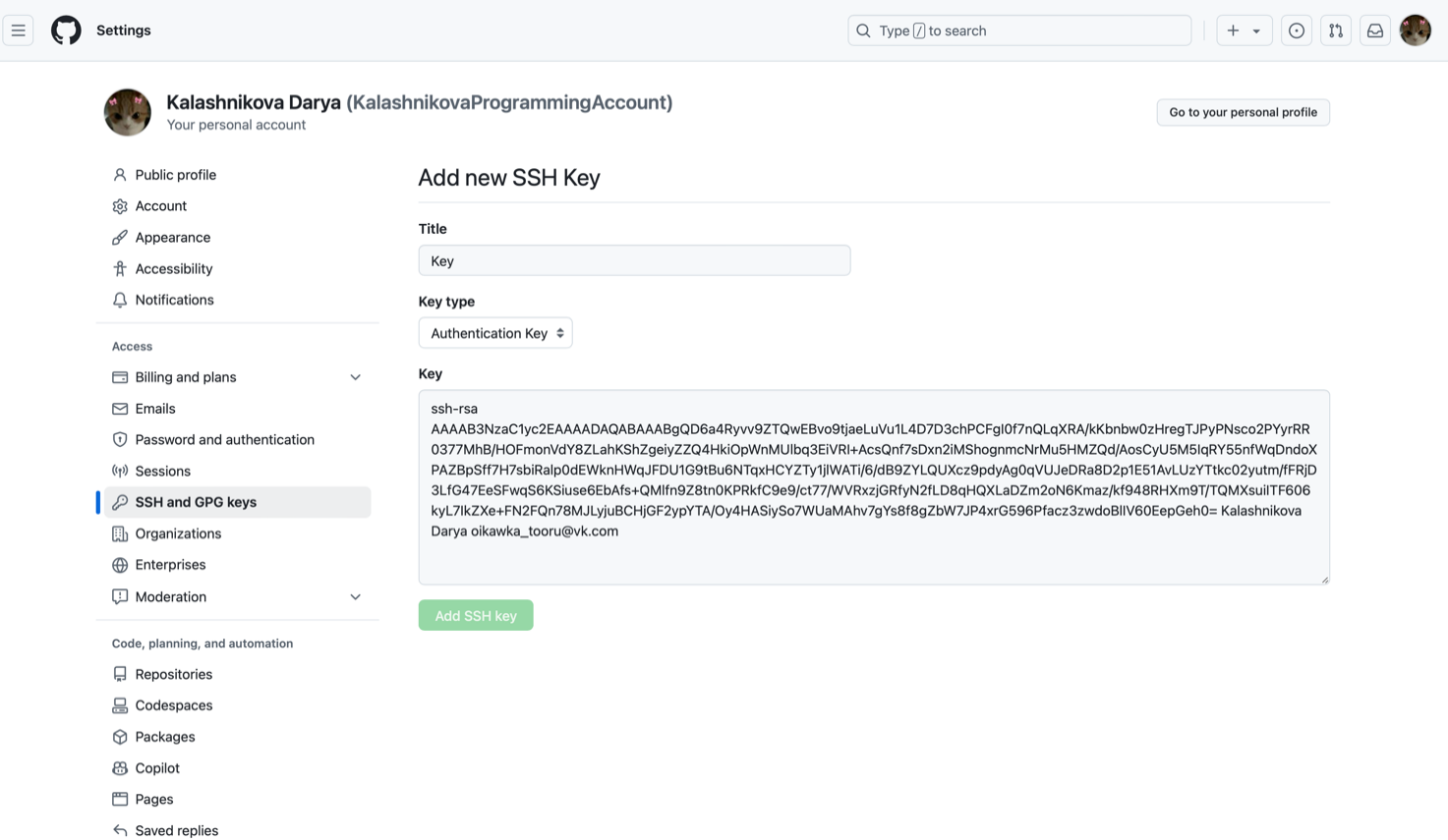
только что сгенерировали:

Рис. 9: Поле ввода ключа в окне добавления нового SSH ключа.

Для того, чтобы скопировать ключ для последующей вставки, нам необходимо ввести следующую команду:

Рис. 10: Считывание и копирование публичного ключа.

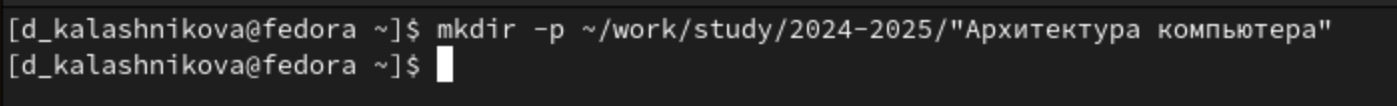
Команда "cat" прочитает данные из файла id\_rsa.pub, а команда xclip выгрузит их в буфер обмена. Остается лишь вставить содержимое буфера обмена, то есть наш ключ, в предложенное поле "Key" и указать имя ключа в поле "Title".

Рис. 11: Заполнение всех необходимых полей для добавления ключа.

## 2.4.4. Создание рабочего пространства и репозитория курса на основе шаблона.

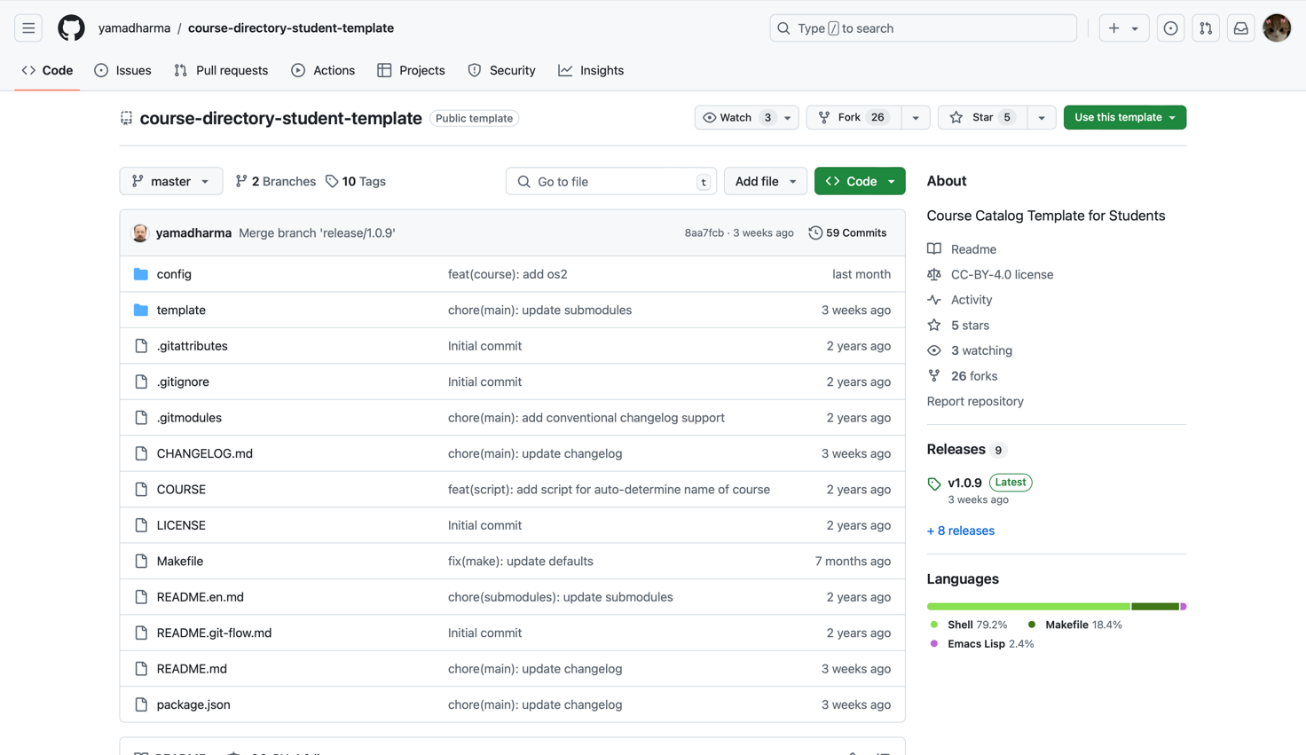
Теперь нам необходимо организовать наше рабочее пространство. Для этого

создадим каталог "Архитектура компьютера" по следующему адресу:

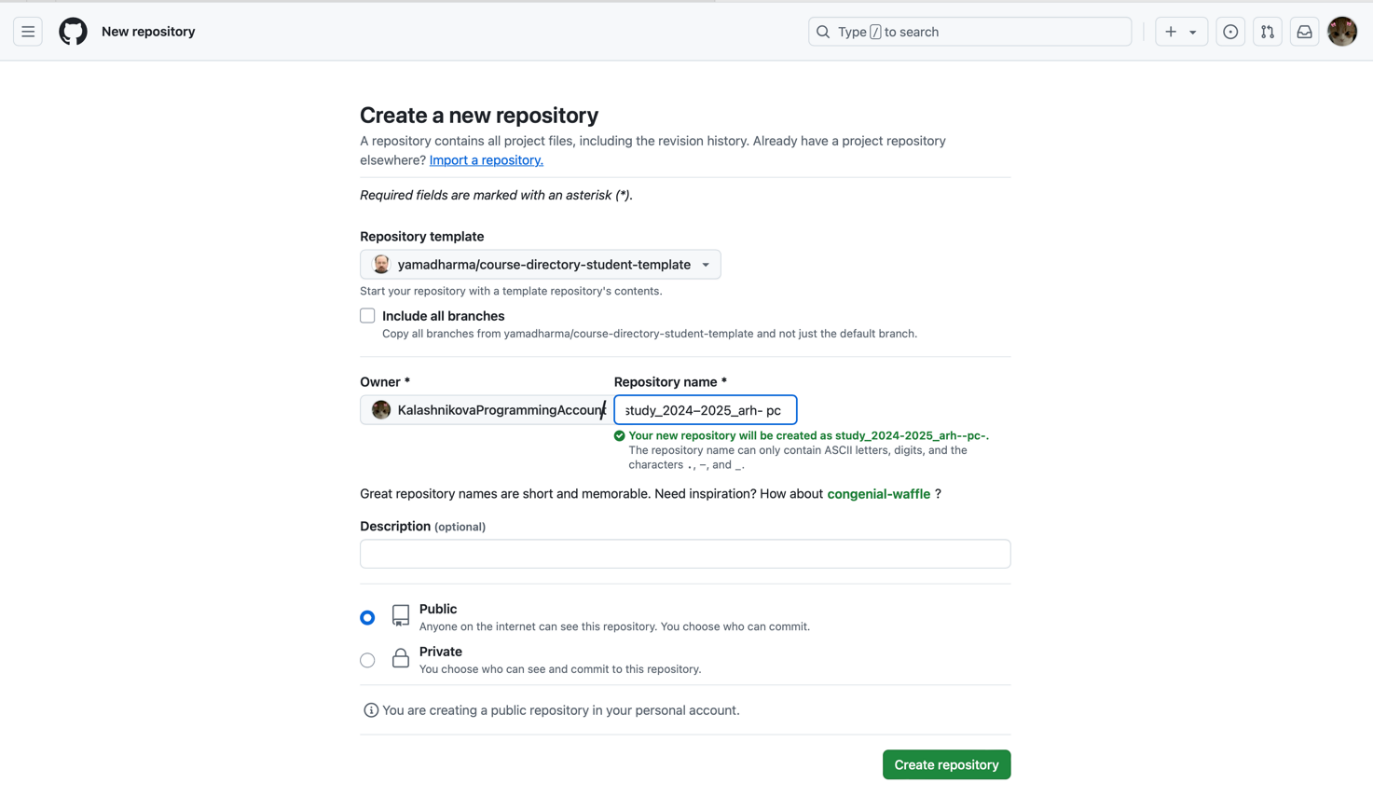
Рис. 12: Создание каталога "Архитектура компьютера".

## 2.4.5. Создание репозитория на основе шаблона.

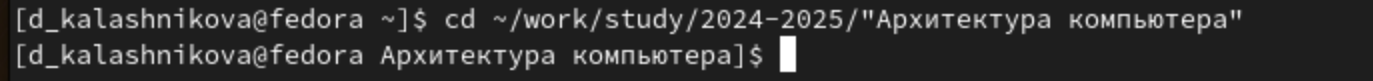
После этого нам нужно будет создать репозиторий. Мы будем его создавать на основе шаблона, который находится по следующему адресу: https://github.com/yamadharma/cour se-directory-student-template.

 Рис. 13: 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. 15). Здесь задаём имя нашего репозитория. Он будет называться так: study\_2024-2025\_arh-pc:

 Рис. 14: 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 заранее созданная папка:

Рис. 15: Перемещение в папку, куда скачается репозиторий.

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

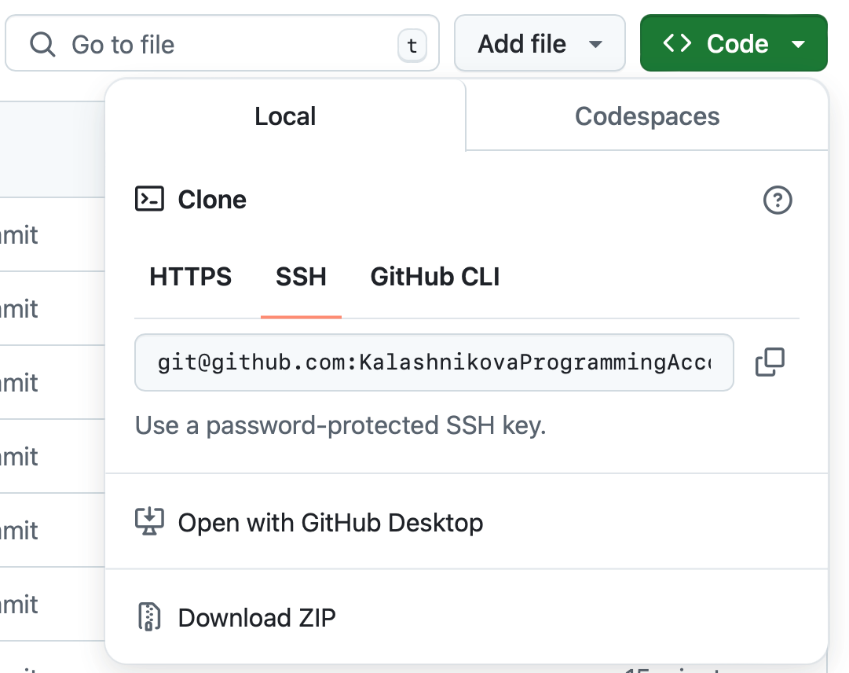
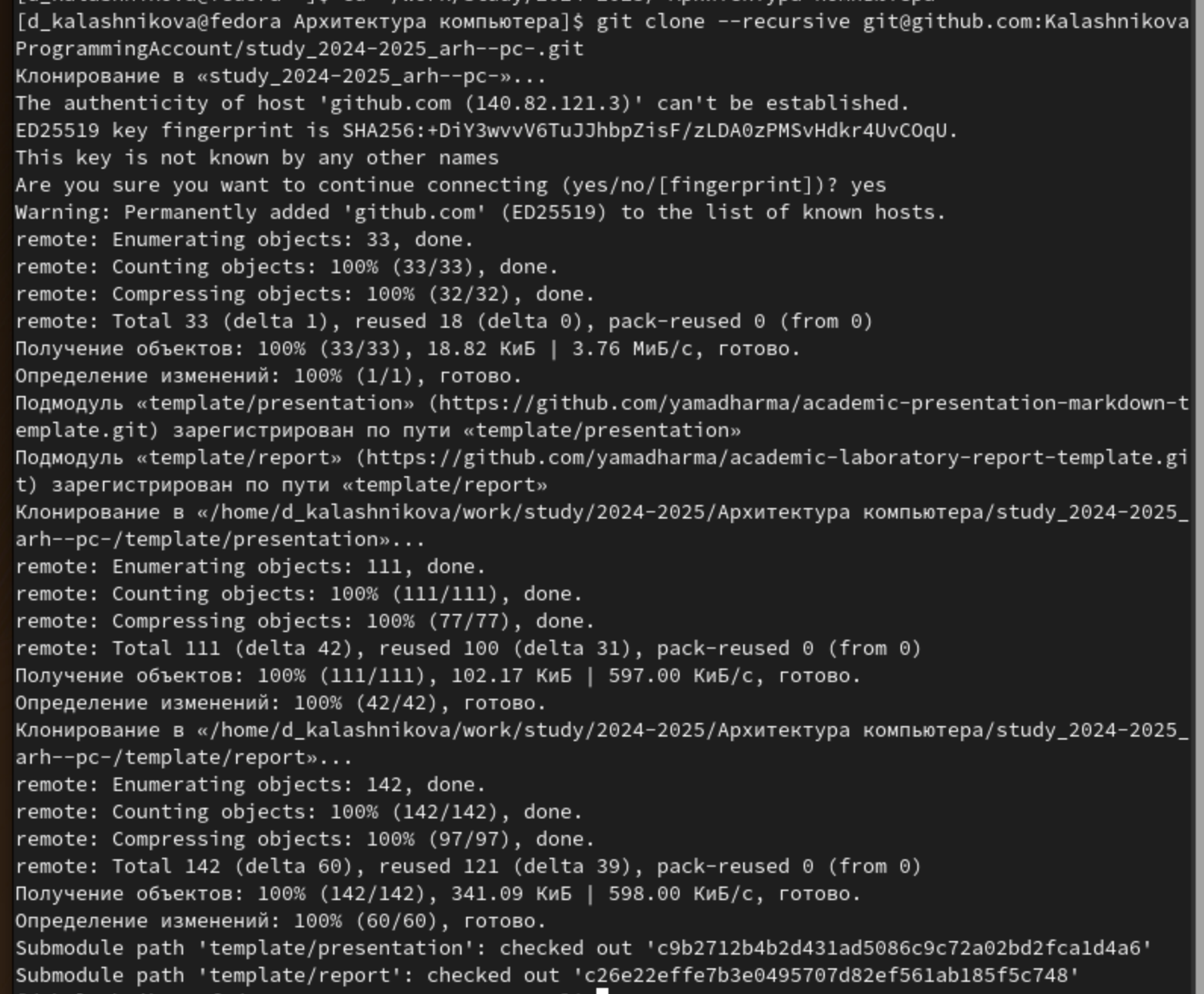


Рис. 16: Ссылка на наш репозиторий.

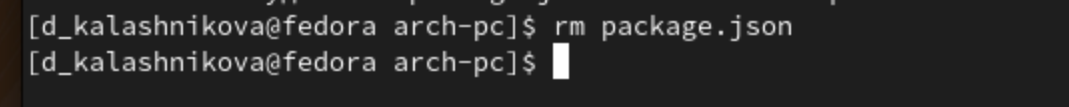
 Рис. 17: Скачивание репозитория командой "git clone".

## 2.4.6. Настройка каталога курса.

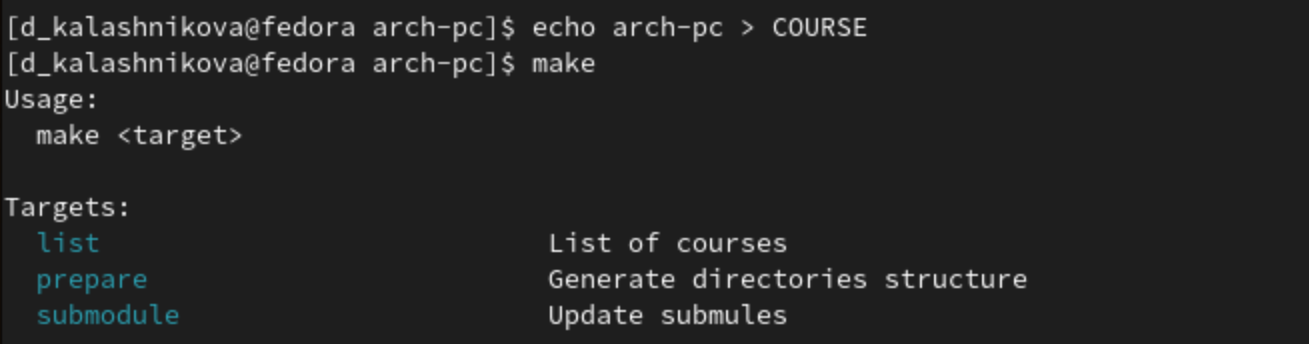
Теперь перейдем к настройке клонированного каталога. Для начала перейдем в него с помощью команды "cd":

Рис. 18: Переход в клонированный каталог.

Удалим с помощью команды "rm" лишний файл:

Рис. 19: Удаление файла.

Теперь создадим необходимые файлы. В нашем случае, это будет файл COURSE. Чтобы его создать, мы воспользуемся командой echo, которая запишет в файл строку "acrh-pc" и автоматически создаст его, т.к. этого файла раньше не существовало:

Рис. 20: Создание файла и запись в него строки.

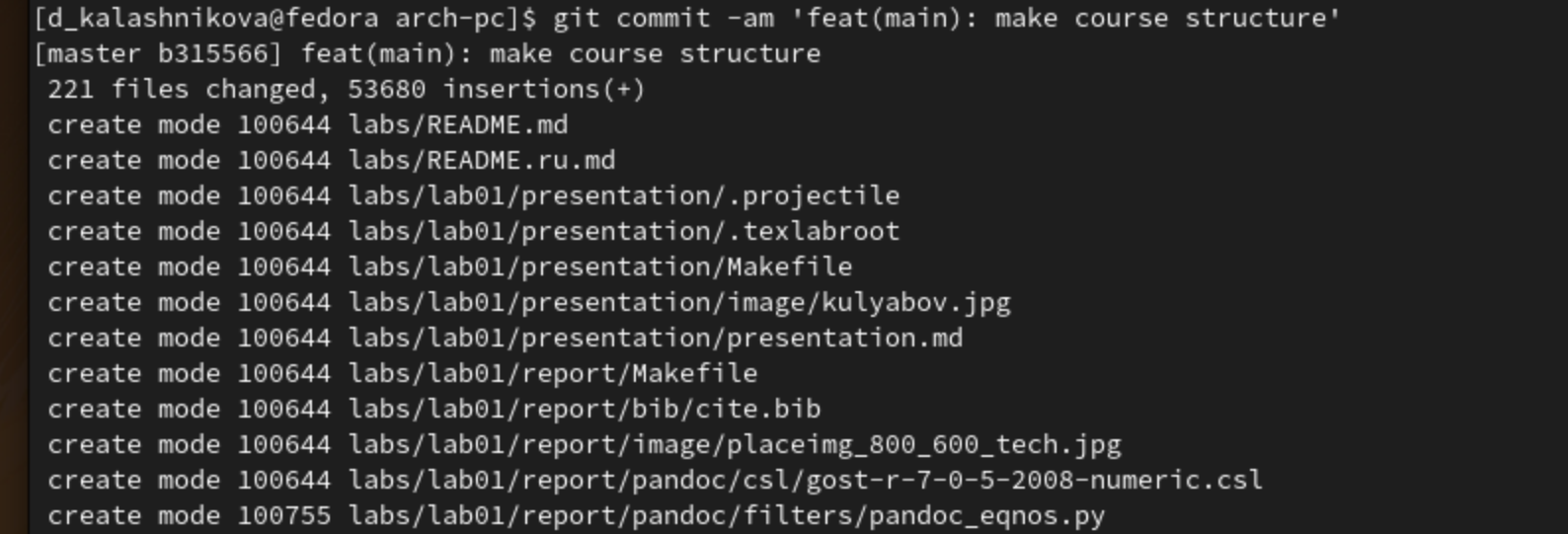
Теперь нам остается лишь отправить файлы на сервер. Для этого с помощью команды git add мы добавим каталоги, которые должны отправляться на сервер. В качестве аргумента мы возьмем точку, которая укажет на то, что мы должны

отправить на сервер все файлы и каталоги, которые по иерархии находятся ниже нашего текущего расположения:

Рис.21: Добавление каталога для отправки на сервер.

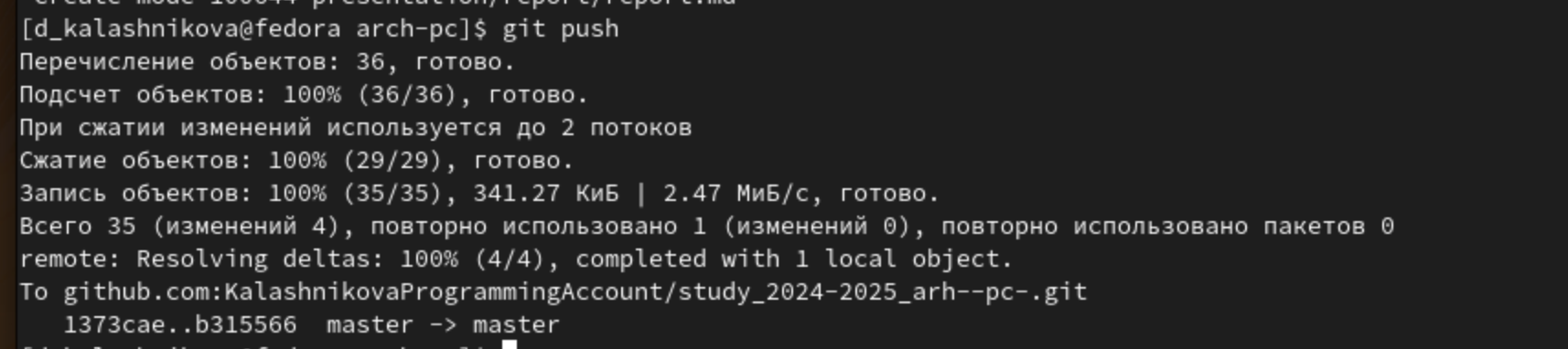
Теперь с помощью команды git commit мы сохраним изменения и укажем

комментарий, в котором будет поясняться, какие изменения мы сделали. В данном случае в комментарии мы напишем, что создали структуру курса:

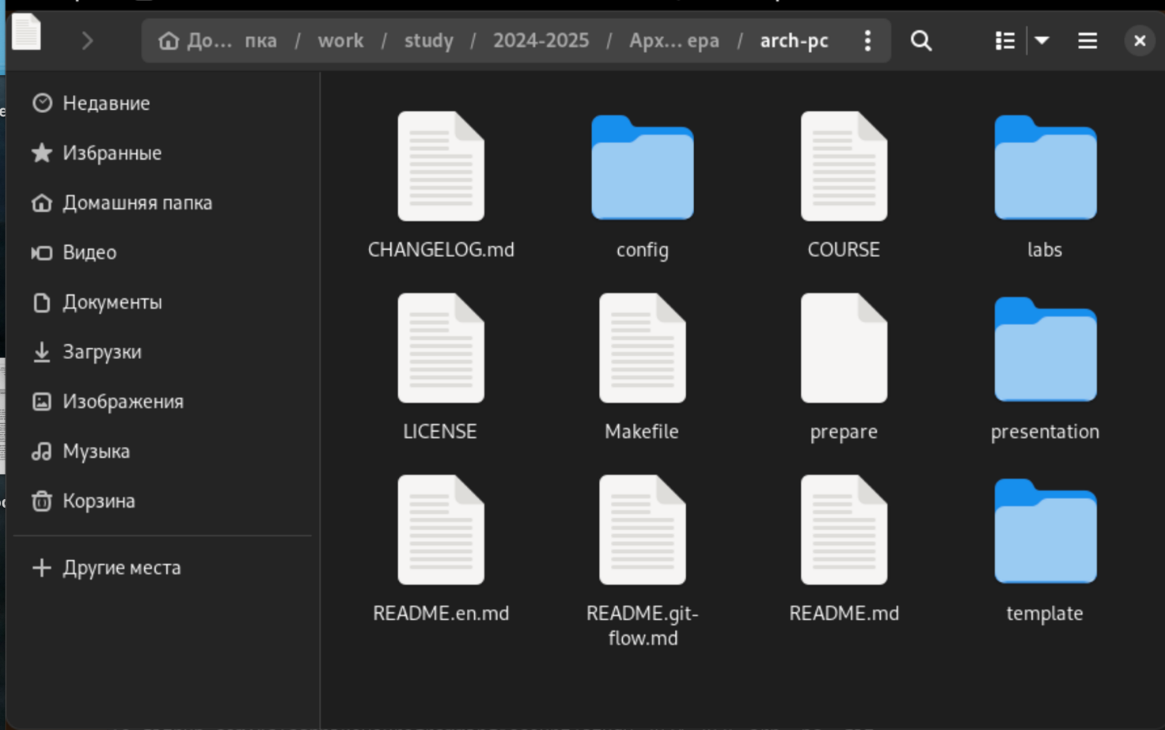
Рис. 22: Сохранение изменений и указание комментария.

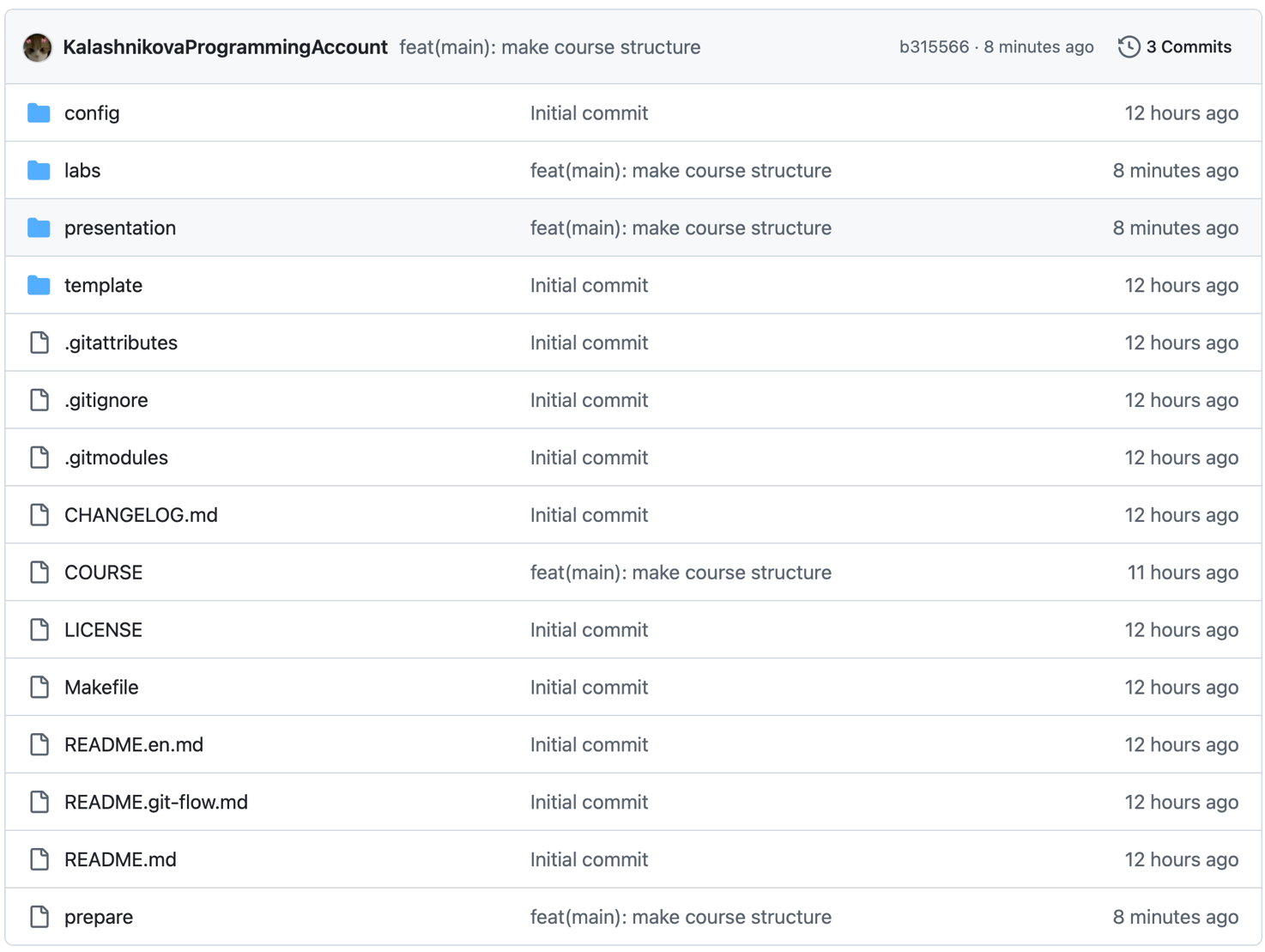
Теперь нам осталось окончательно загрузить изменения на сервер. Для этого

мы воспользуемся командой git push:

Рис. 23: 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. 24: Файлы на компьютере.

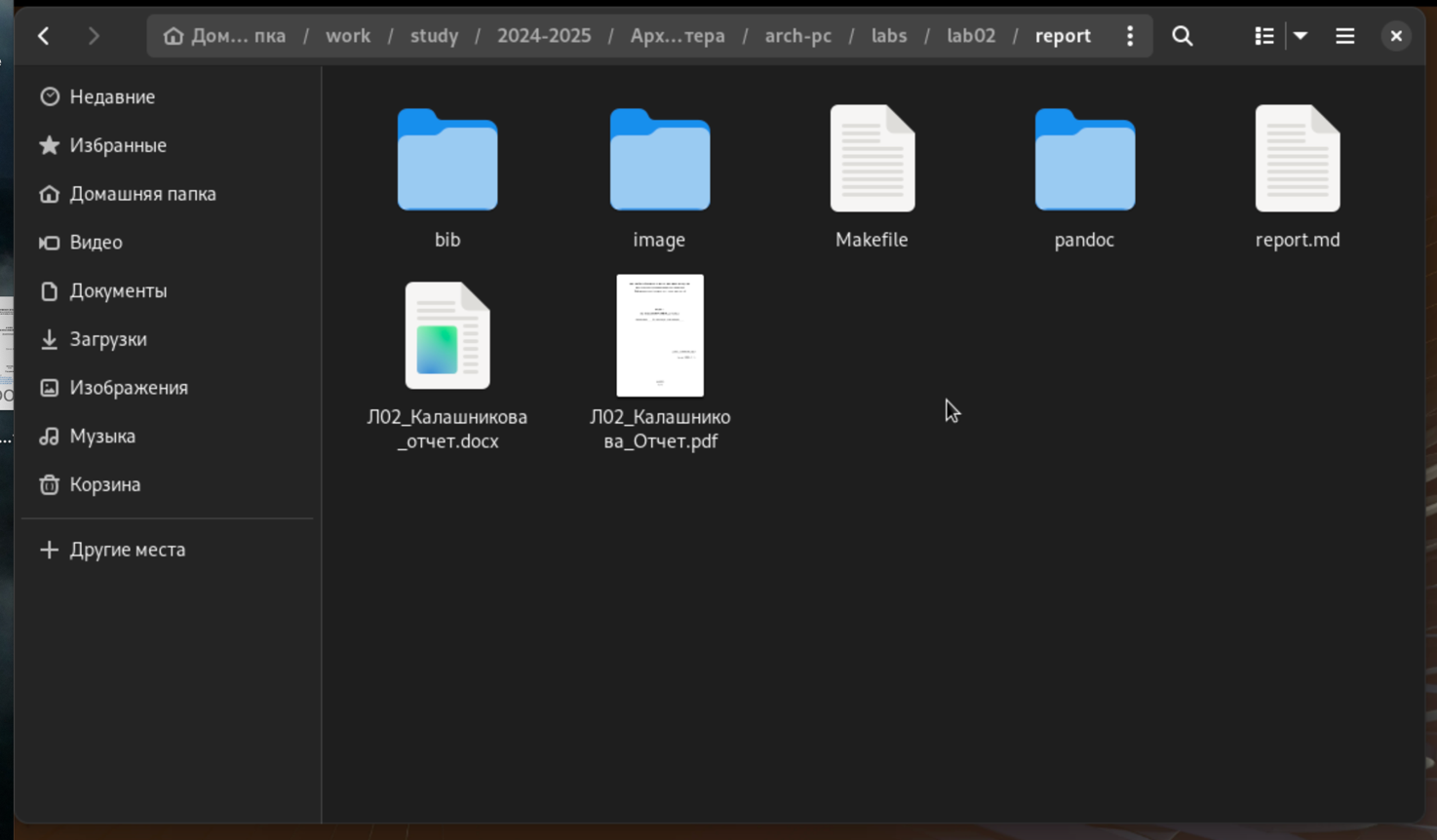
Рис. 25: Файлы на сервере.

Всё совпало.

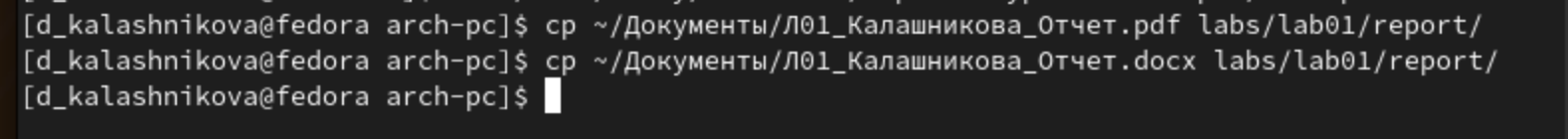
## 2.5. Задание для самостоятельной работы

**Задание №1.** 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. 26: Наличие отчёта в папке.

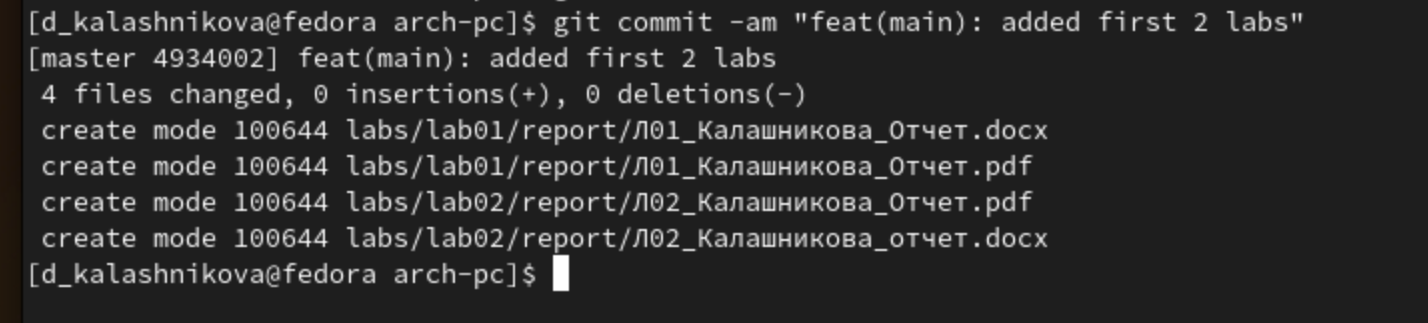
**Задание №2.** Копируем отчет по нашей предыдущей лабораторной работе в соответствующую папку созданного нами рабочего пространства, то есть в папку labs/lab01/report. Для копирования воспользуемся командой "cp":

Рис. 27: Копирование отчёта по первой лабораторной работе в папку lab01.

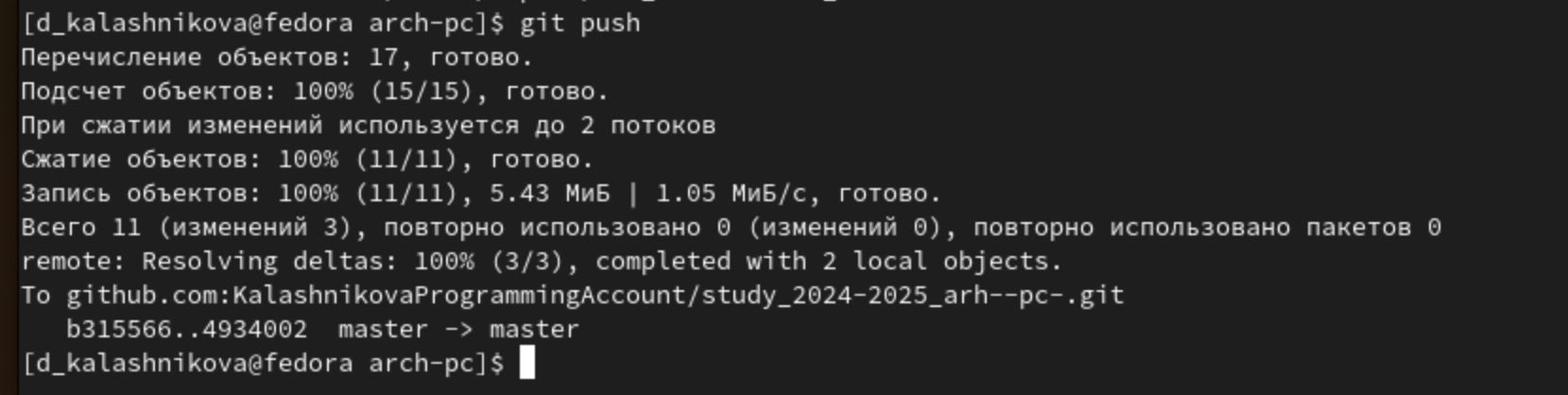
**Задание №3.** Теперь нам осталось лишь загрузить изменения на GitHub. Для этого нам нужно повторить те шаги, которые мы выполнили в предыдущих пунктах лабораторной работы, а именно, воспользоваться командой "git add ." для того, чтобы указать, что мы хотим сохранить изменения во всех файлах, находящихся в нашем каталоге.

Рис. 27: Добавление каталога для отправки на сервер.

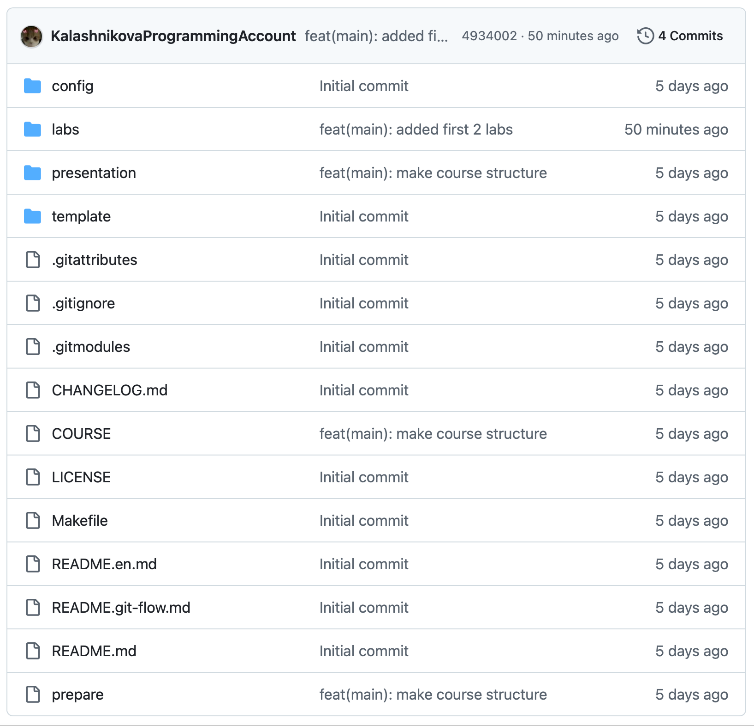
После этого с помощью команды "git commit" мы укажем комментарий и сохраним изменения. В комментарии мы укажем, что мы загрузили первую и вторую лабораторные работы.

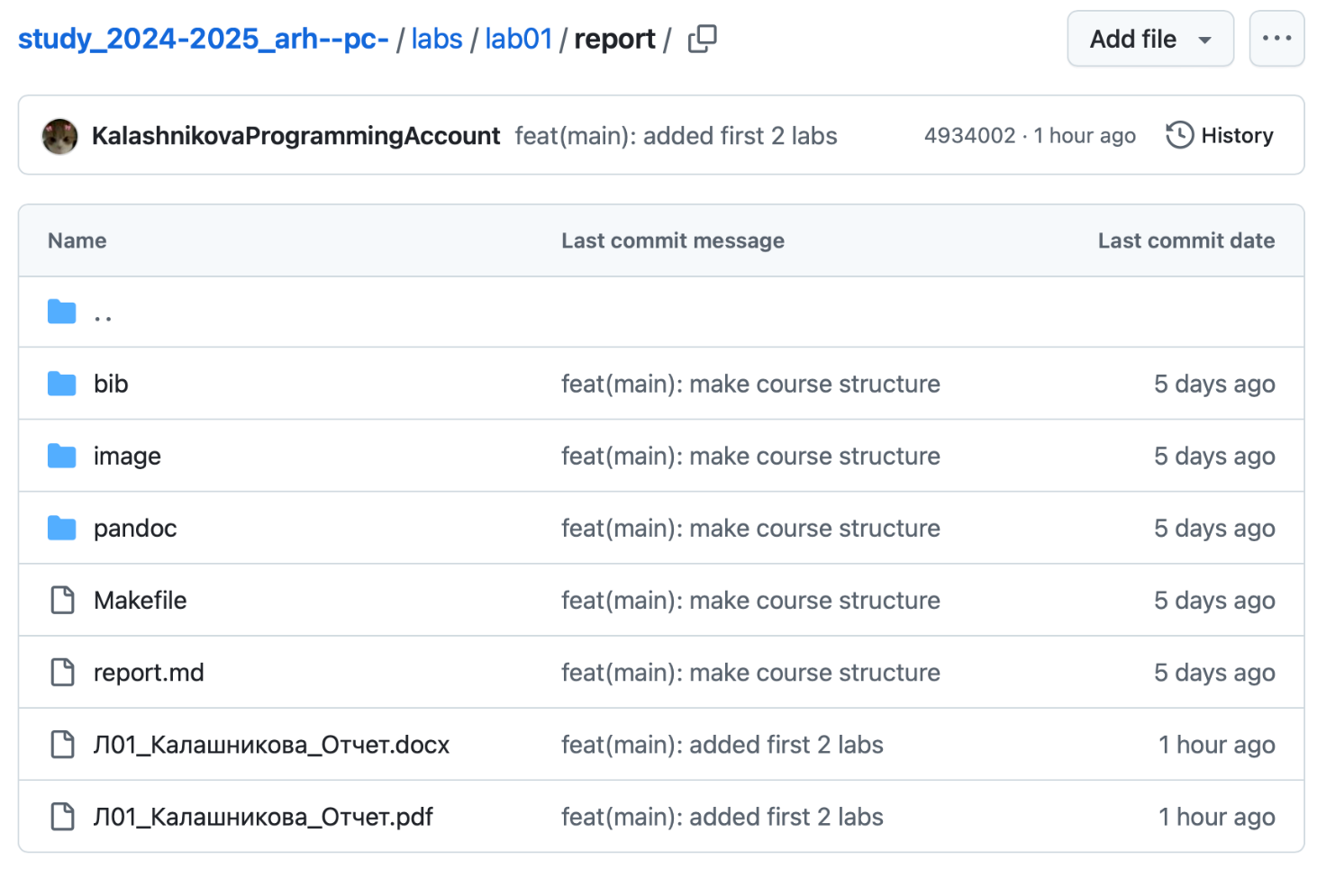
Рис. 28: Сохранение изменений и указание комментария.

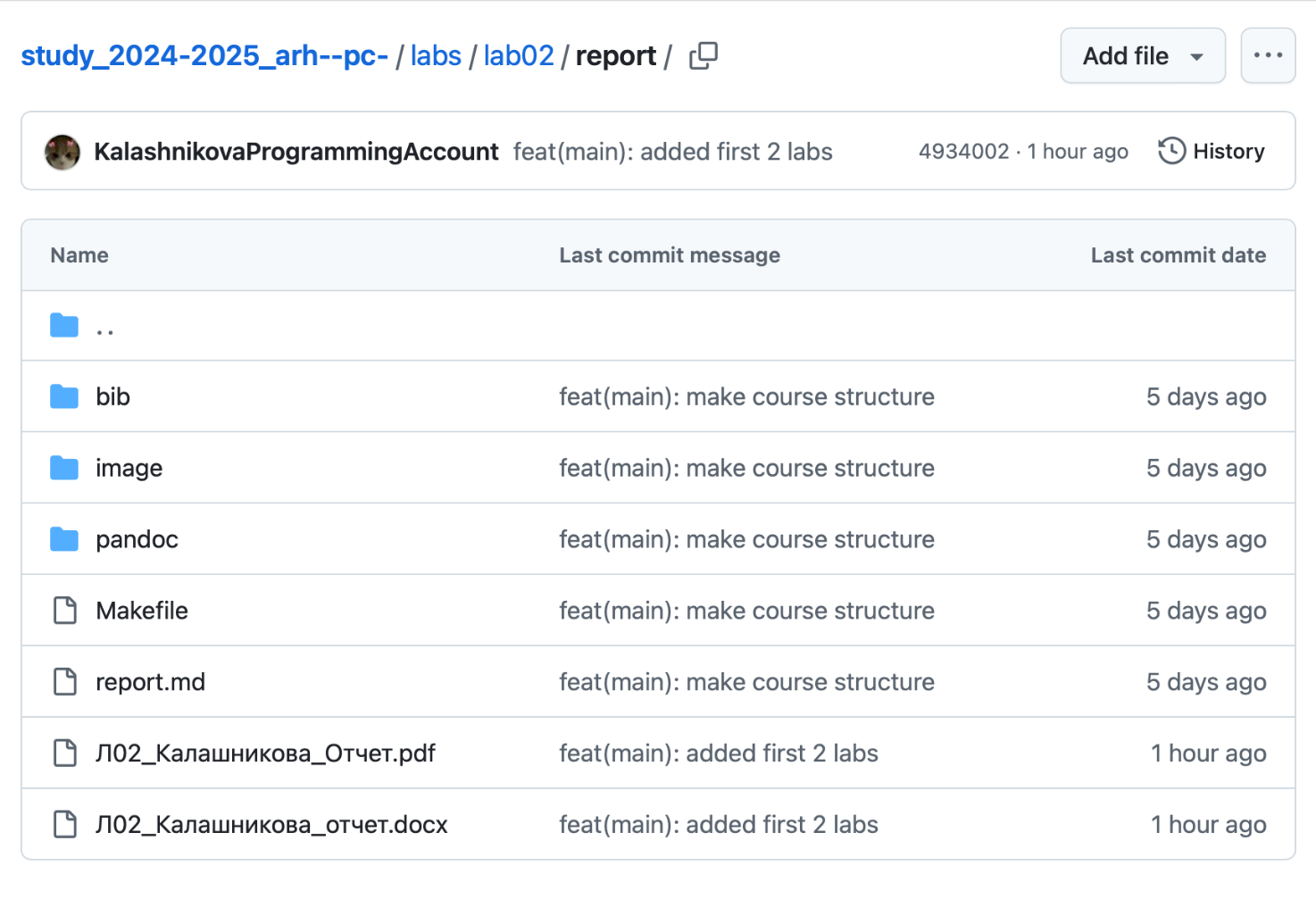
После этого вводим команду "git push" для того, чтобы загрузить наши файлы на GitHub.

Рис. 29: 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. 30: Вид обновлённого репозитория на GitHub.

Рис. 31: Наличие отчетов на GitHub.

Рис. 32: Наличие отчетов на GitHub.

# 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.