**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: Загидуллина Э.Д.

Группа: НБИбд-01-25

**МОСКВА**

2025 г.

Оглавление

[1 Цель работы 3](#_Toc211019293)

[2 Задания 4](#_Toc211019294)

[3 Теоретическое введение 5](#_Toc211019295)

[4. Выполнение лабораторной работы 7](#_Toc211019296)

[4.1 Настройка GitHub 7](#_Toc211019297)

[4.2 Базовая настройка git 7](#_Toc211019298)

[4.3 Создание SSH-ключа 8](#_Toc211019299)

[4.4 Создание рабочего пространства и репозитория курса на основе шаблона 10](#_Toc211019300)

[4.5 Создание репозитория курса на основе шаблона 10](#_Toc211019301)

[4.6. Настройка каталога курса 12](#_Toc211019302)

[5. Задание для самостоятельной работы 14](#_Toc211019303)

[6. Вывод 17](#_Toc211019304)

# 1 Цель работы

Целью данной работы является изучить идеологию и применение средств контроля версий, а также приобрести практические навыки по работе 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 курса на основе шаблона.
  6. Настройка каталога курса.

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 неполную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

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):

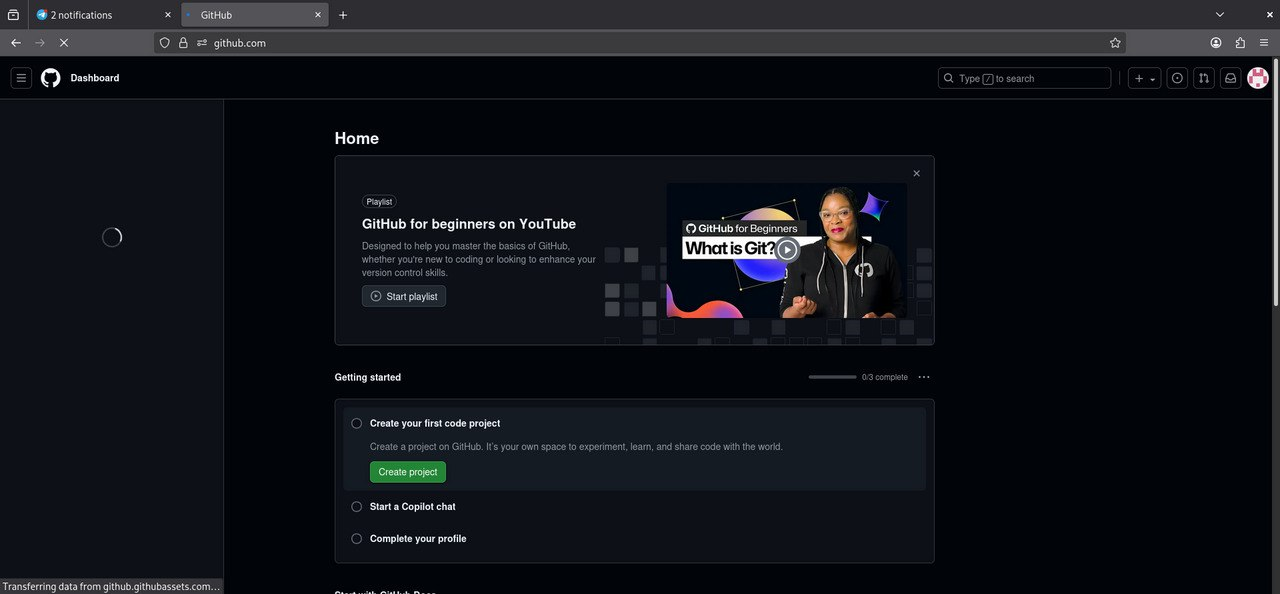


Рис. 4.1. Создание учетной записи на GitHub

## 4.2 Базовая настройка git

Открываю терминал и ввожу команды, указав имя и e-mail моего репозитория (рис. 4.2):

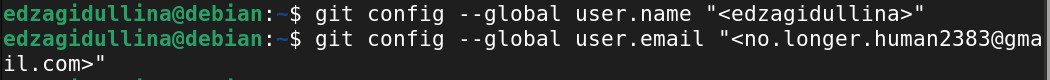


Рис. 4.2 Настройка GitHub в терминале

Настраиваю utf-8 в выводе сообщений git и задаю начальное имя ветки (рис.4.3):



Рис. 4.3 Создание начальной ветки master

Параметры autocrlf и safecrlf (рис. 4.4) (рис. 4.5):



Рис. 4.4. Параметр autocrlf



Рис. 4.5 Параметр safecrlf

## 4.3 Создание SSH-ключа

Для последующей идентификации пользователя на сервере репозиториев необходимо сгенерировать пару ключей (приватный и открытый).

Генерирую ключи (рис. 4.6):



Рис. 4.6 Генерация ключа

Генерирую открытый ключ. Захожу на сайт http://github.org/ и перехожу в меню “Settings”. После этого в боковом меню выбираю “SSH and GPG keys” и нажимаю кнопку “New SSH key”. Копирую из локальной консоли ключ в буфер обмена (рис. 4.7):

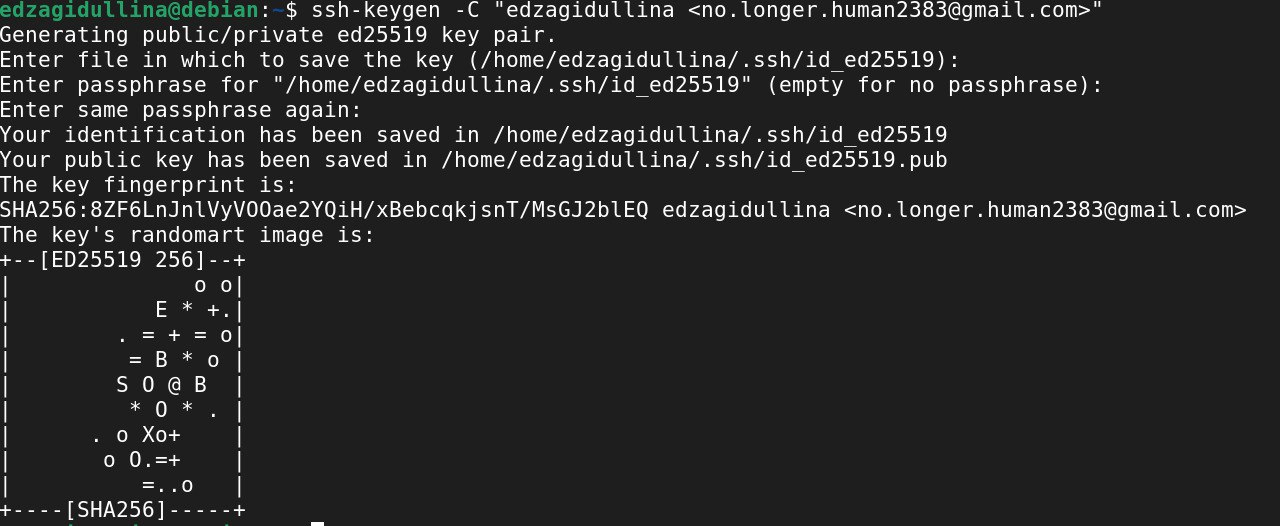


Рис. 4.7 Генерация приватного и открытого ключа

Копирую ссылку (рис. 4.8):



Рис. 4.8 Копирование ссылки

Вставляю ссылку в поле “key” на GitHub (рис. 4.9) (рис.4.10):

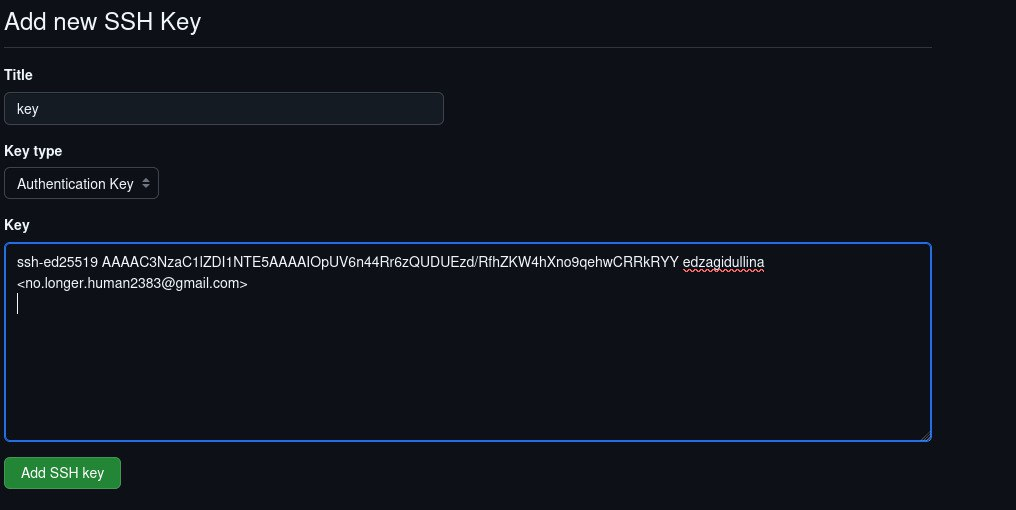


Рис. 4.9 Вставка ссылки

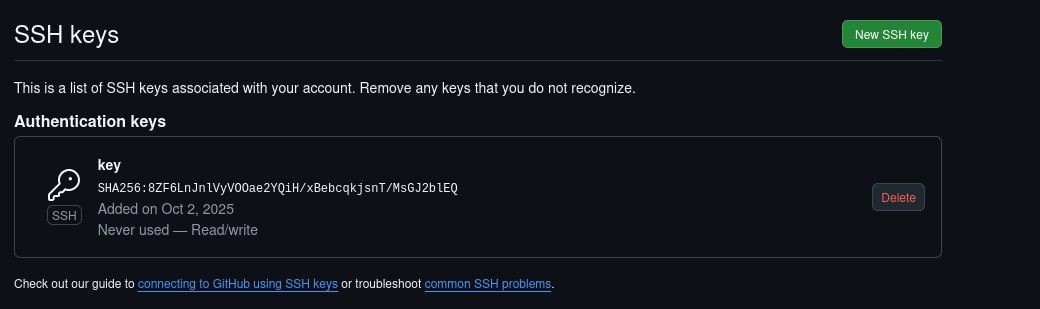


Рис. 4.10 Сохранение ключа

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

Открываю терминал и создаю каталог для предмета «Архитектура компьютера» (рис. 4.11):



Рис.4.11 Создание каталога «Архитектура компьютера»

## 4.5 Создание репозитория курса на основе шаблона

Перехожу на станицу репозитория с шаблоном курса https://github.com/yamadharma/cour. Далее выбираю Use this template (рис. 4.12) (рис.4.13):

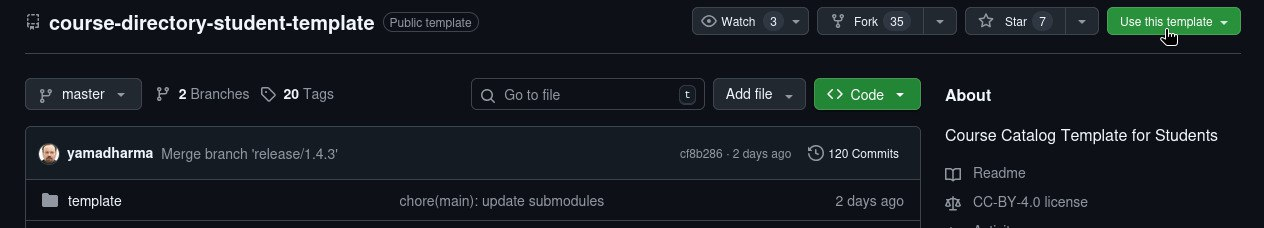


Рис. 4.12 Использую шаблон курса

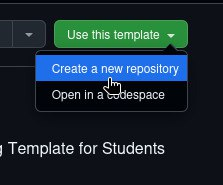


Рис. 4.13

Задаю имя репозитория (рис. 4.14):

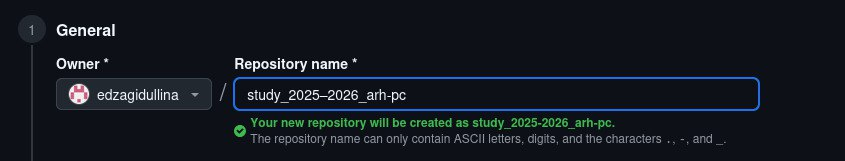


Рис. 4.14 Название репозитория

Созданный репозиторий (рис. 4.15):

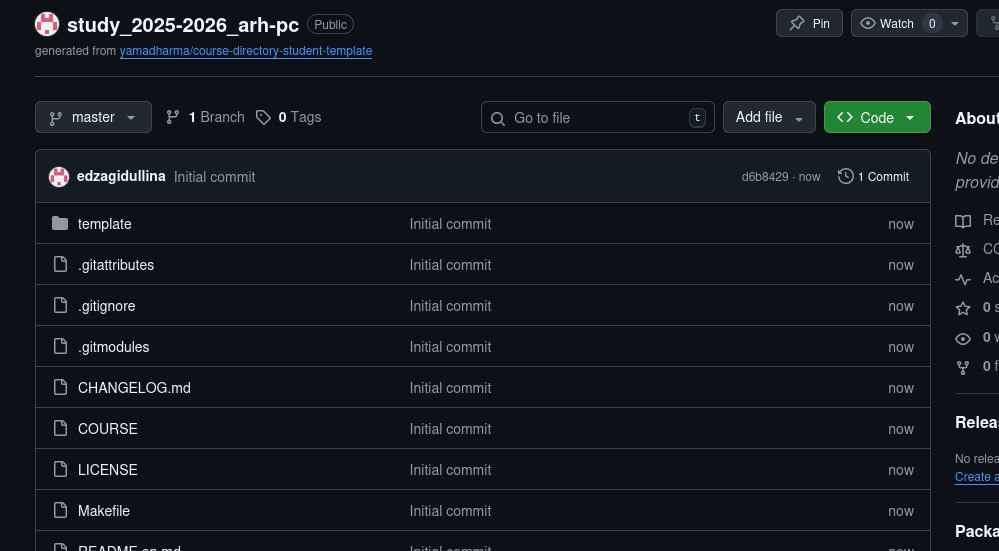


Рис. 4.15. Созданный репозиторий

Открываю терминал и перехожу в каталог курса (рис.4.16):



Рис. 4.16 Переход в каталог

Клонирую созданный репозиторий (рис. 4.17) (рис. 4.18):

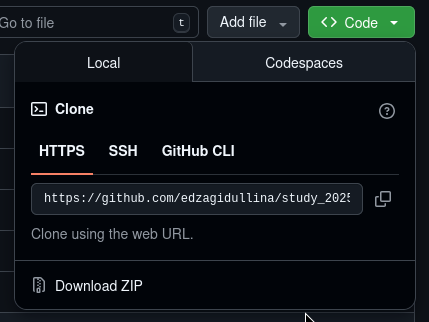


Рис. 4.17 Копирование ссылки из GitHub

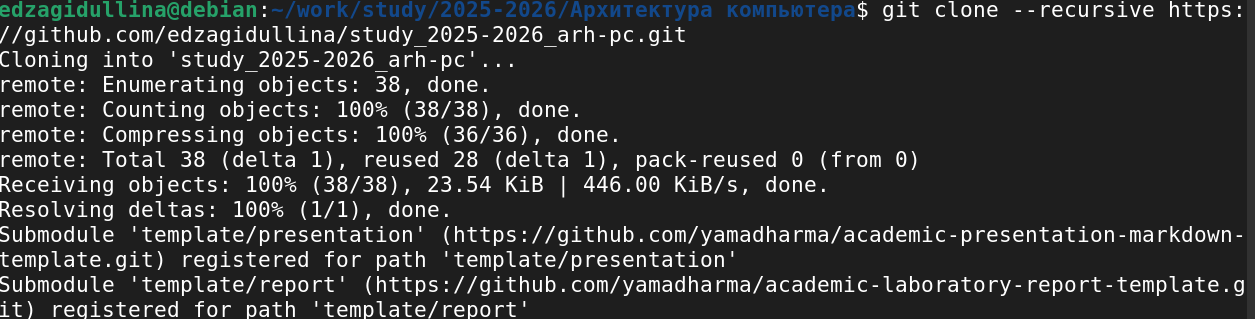


Рис. 4.18 Клонирование репозитория

## 4.6. Настройка каталога курса

Перехожу в каталог курса (рис.4.19):



Рис. 4.19 Переход в каталог

Создаю необходимые каталоги (рис.4.20):



Рис. 4.20 Создание каталогов

Через терминал перехожу в созданный каталог при помощи cd (рис. 4.21):



Рис. 4.21 Переход в кталог

Удаляю ненужные файлы (рис. 4.22):



Рис. 4.22 Удаление ненужных файлов

Создаю необходимые каталоги (рис. 4.23):

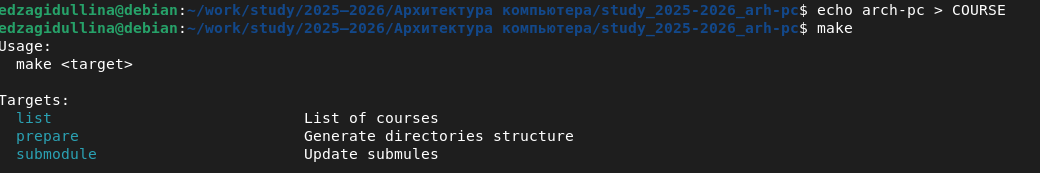


Рис. 4.23 Создание каталогов

Отправляю файлы на сервер (рис. 4.24) (рис. 4.25):

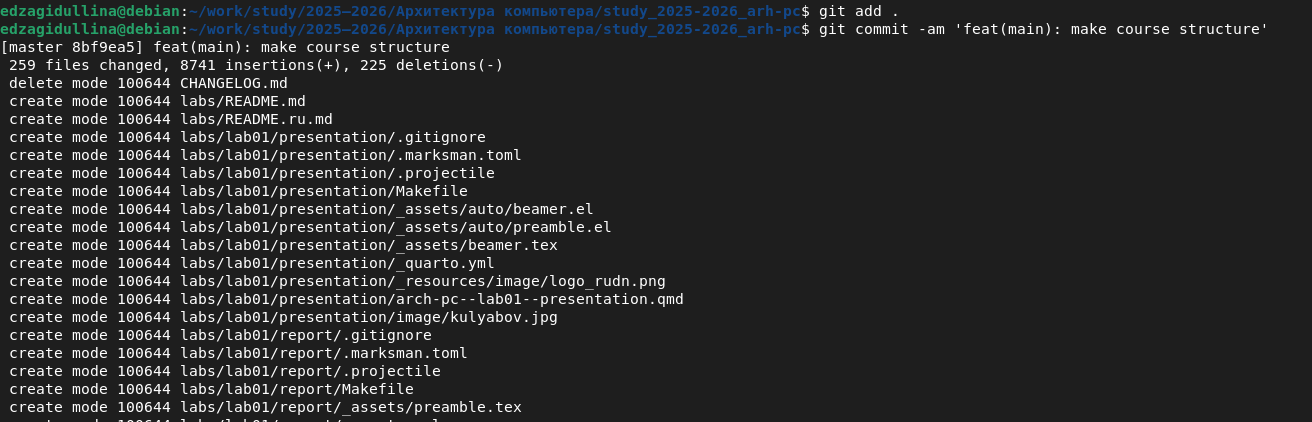


Рис. 4.24

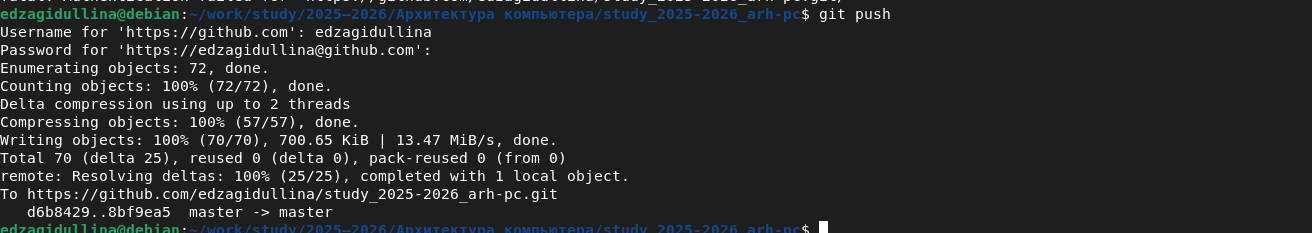


Рис. 4.25 Отправка файлов на сервер

Проверяю правильность создания рабочего пространства на странице GitHub (4.26):

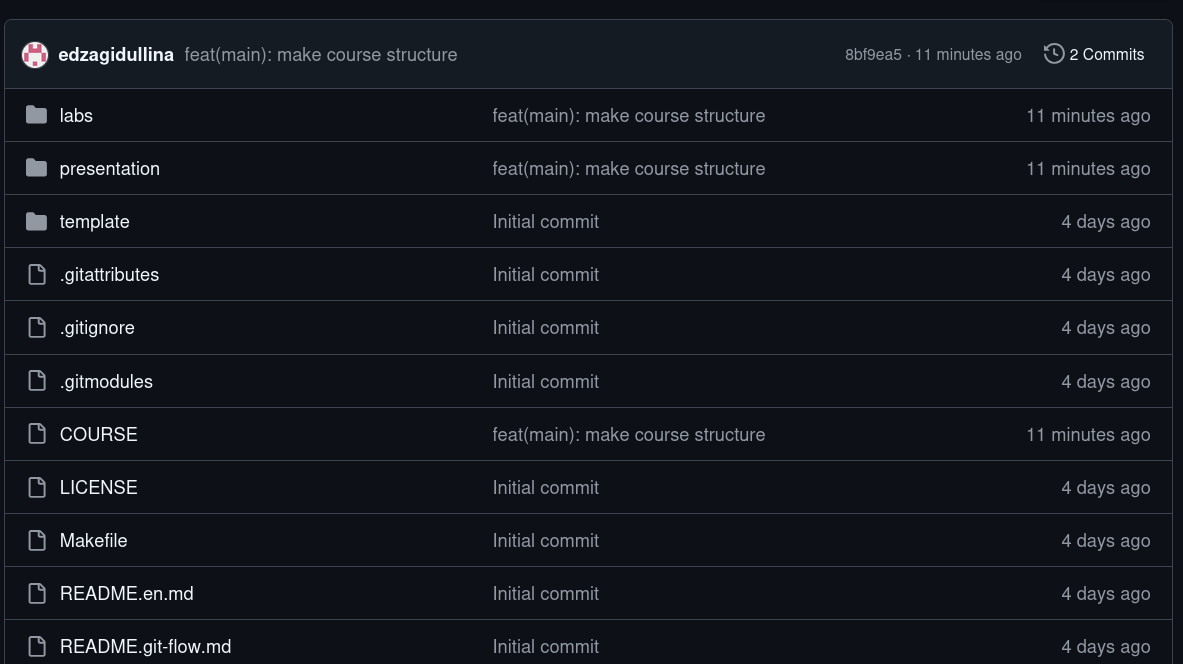


Рис. 4.26 Рабочее пространство на GitHub

# 5. Задание для самостоятельной работы

Создаю отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs/lab02/report) (рис. 4.27):



Рис. 4.27 Создание пространства для отчета

Перехожу из каталога lab02 в lab01 (рис.4.28):

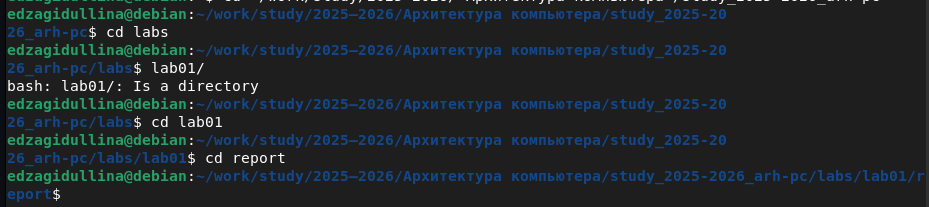


Рис. 4.28 Переход в lab01

Проверяю местонахождение файлов (рис. 4.29):

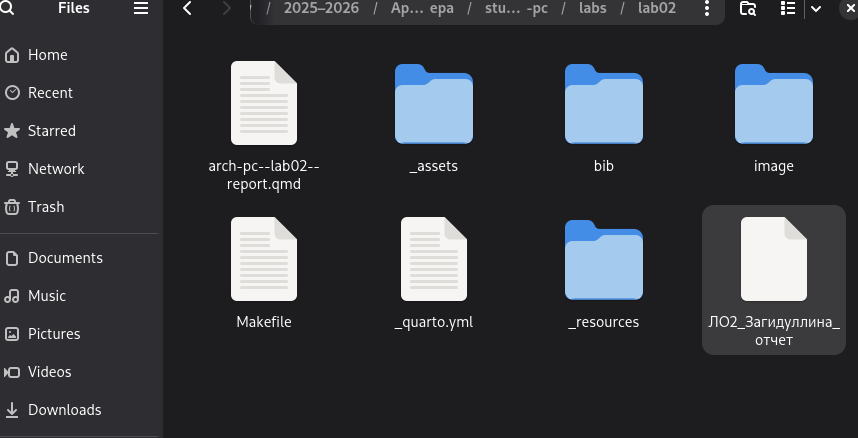


Рис. 4.29 Отчет в папке labs

Загружаем остальные отчеты на GitHub (рис 4.30) (рис. 4.31):

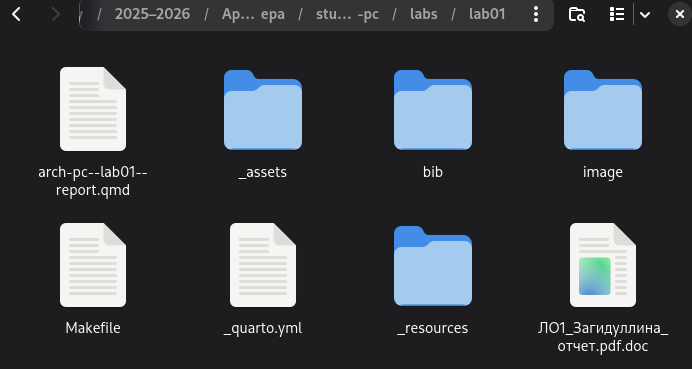


Рис. 4.30 Отчет 1 в папке “lab01”

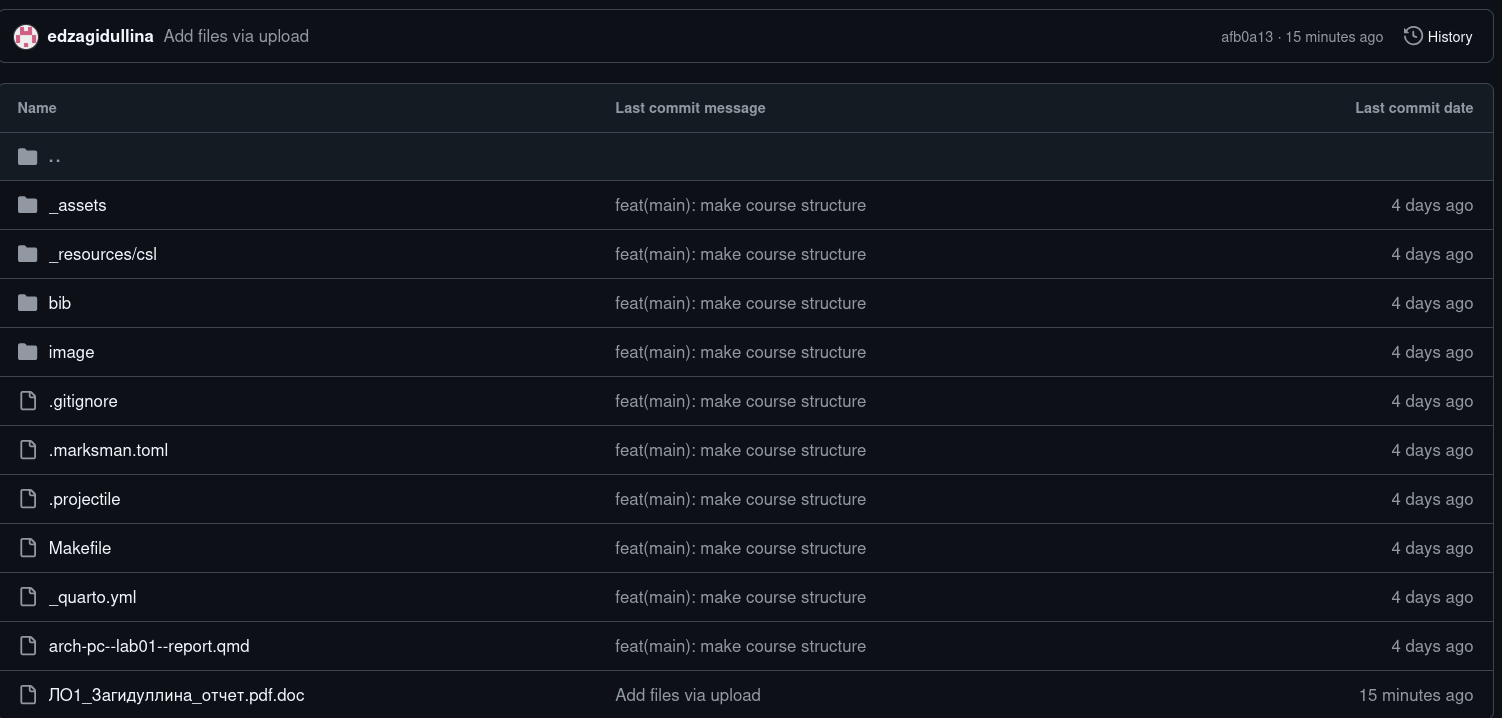


Рис. 4.31 Отчет 1 в lab01/report

ЛО­2\_Загидуллина\_отчет также загружаем на Github

# 6. Вывод

При выполнении данной лабораторной работы я изучила идеологию и применение средств контроля версий, а также приобрела практические навыки по работе с системой git.