Отчёт по лабораторной работе №2

Дисциплина: архитектура компьютера

Ахмаров Роман

Содержание

# 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 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.

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.

# 3 Выполнение лабораторной работы

## 3.1 Базовая настройка

Создаем учетную запись на сайте GitHub. Заполняем данные и регестрируем аккаунт. В моем случае мое имя пользователя: «romashalun» (рис. [1](#fig:fig1)).

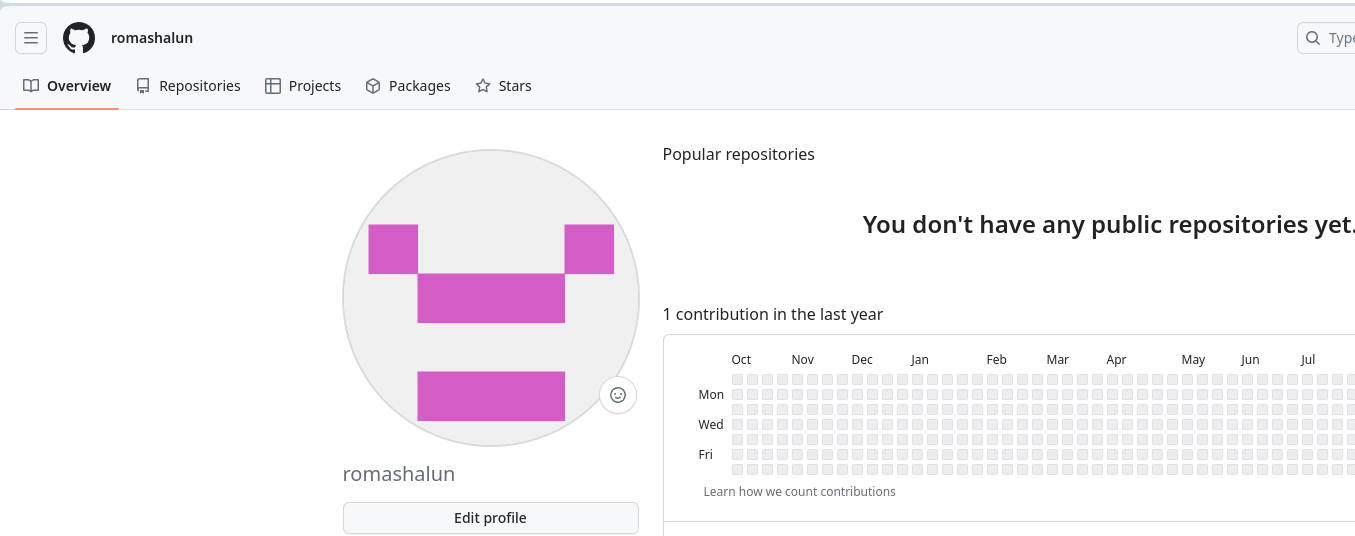


Figure 1: Учетная запись GitHub

Запускаю виртуальную машину, затем в терминале задаю предварительную конфигурацию git. Ввожу команду git config –global user.name “”, указывая свое имя и команду git config –global user.email “work@mail”, указывая в ней электронную почту владельца, то есть мою. (рис. ??)

 Конфигурация git

Настраиваю utf-8 в выводе сообщений git для корректного отображения символов(рис. [2](#fig:fig3))

Figure 2: Настройка

Figure 2: Настройка

Задаю имя «master» для начальной ветки (рис. [3](#fig:fig4))

Figure 3: Создание имени для ветки

Figure 3: Создание имени для ветки

Задаю параметр autocrlf со значением input (рис. [4](#fig:fig5))

Figure 4: Параметр autocrlf

Figure 4: Параметр autocrlf

Задаю параметр safecrlf со значением warn (рис. [5](#fig:fig6)).

Figure 5: Параметр safecrlf

Figure 5: Параметр safecrlf

## 3.2 Создание SSH-ключа

ОВвожу комнаду ssh-keygen -C “имя фамилия, почта” , указываю свои данные. Ключ сохранятся в каталоге ~/.ssh/. (рис. [6](#fig:fig7))

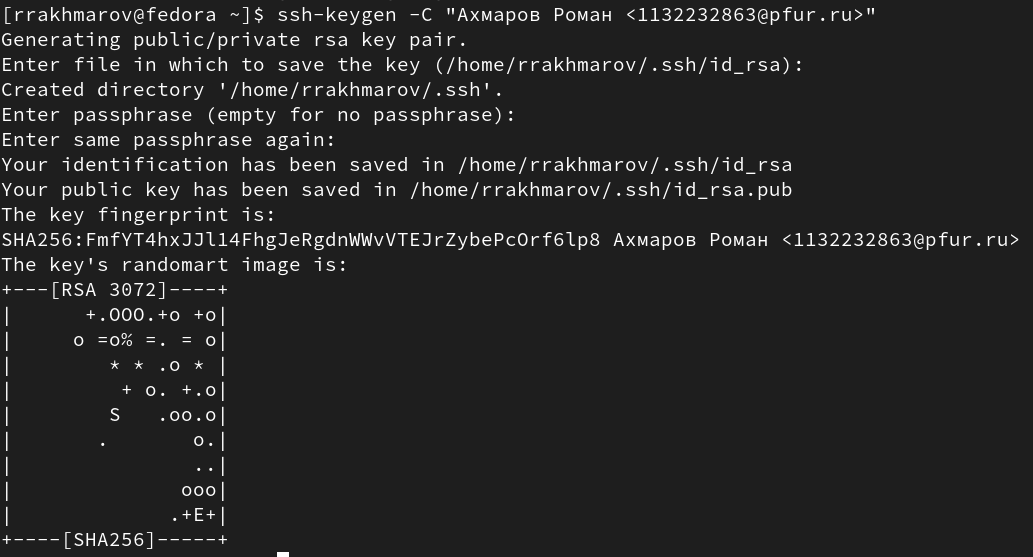


Figure 6: Генерация ssh-ключа

Копирую открытый ключ из директории, в которой он был сохранен, с помощью утилиты xclip. (рис. [7](#fig:fig8))

Figure 7: Копирование ключа

Figure 7: Копирование ключа

Открываю свой профиль и выбираю страницу SSH and GPG key. Нажимаю кнопку New SSH key и вставляю скопированный ключ в специальное поле и называю своим именем.(рис. [8](#fig:fig9))

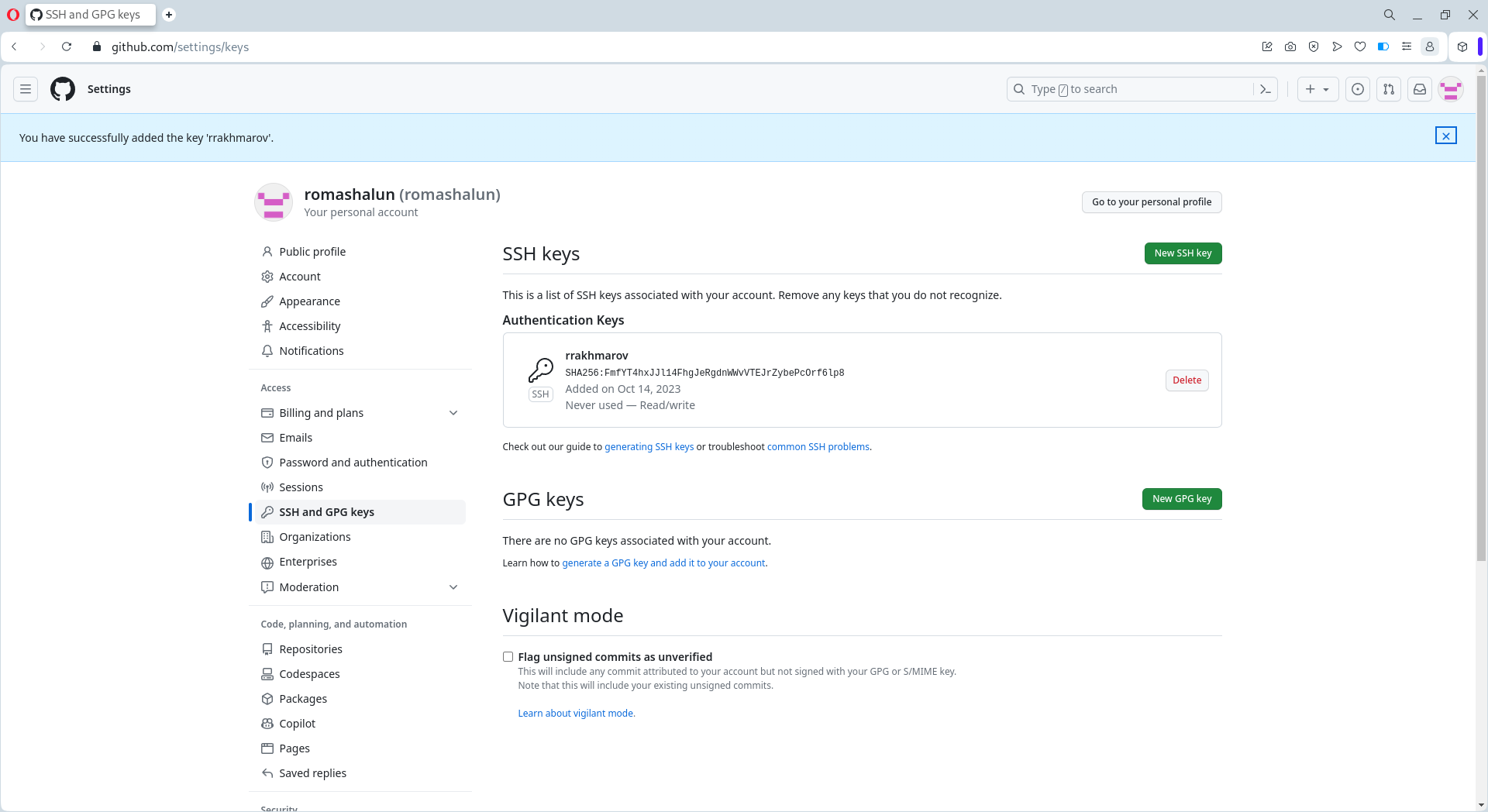


Figure 8: Добавление ключа

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

Открываю терминал. Создаю директорию, рабочее пространство, с помощью утилиты mkdir, благодаря ключу -p создаю все директории после домашней ~/work/study/2023-2024/“Computer architecture” рекурсивно. Далее проверяю с помощью ls, действительно ли были созданы необходимые мне каталоги (рис. [9](#fig:fig10))

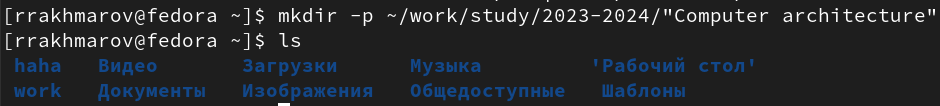
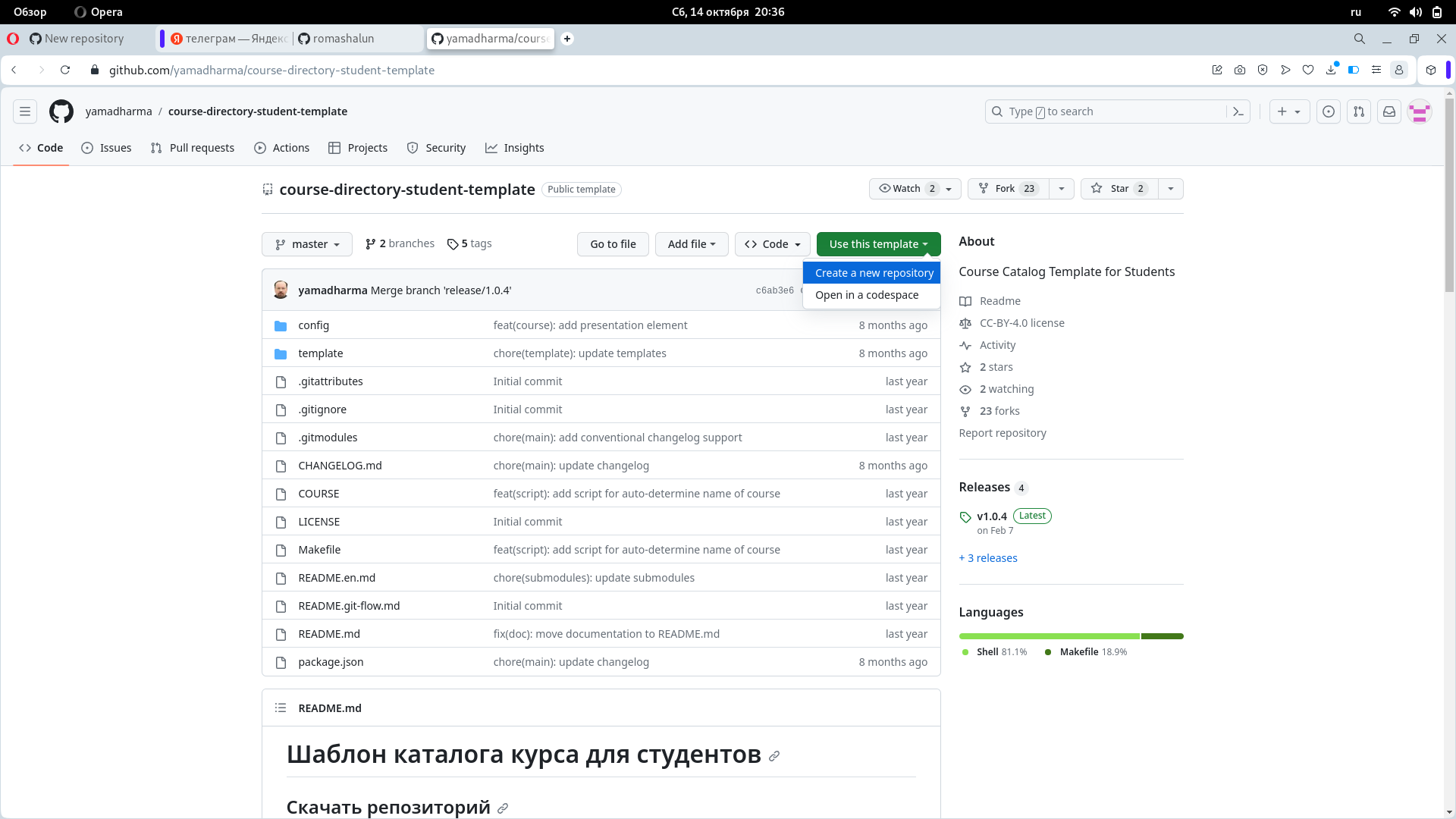
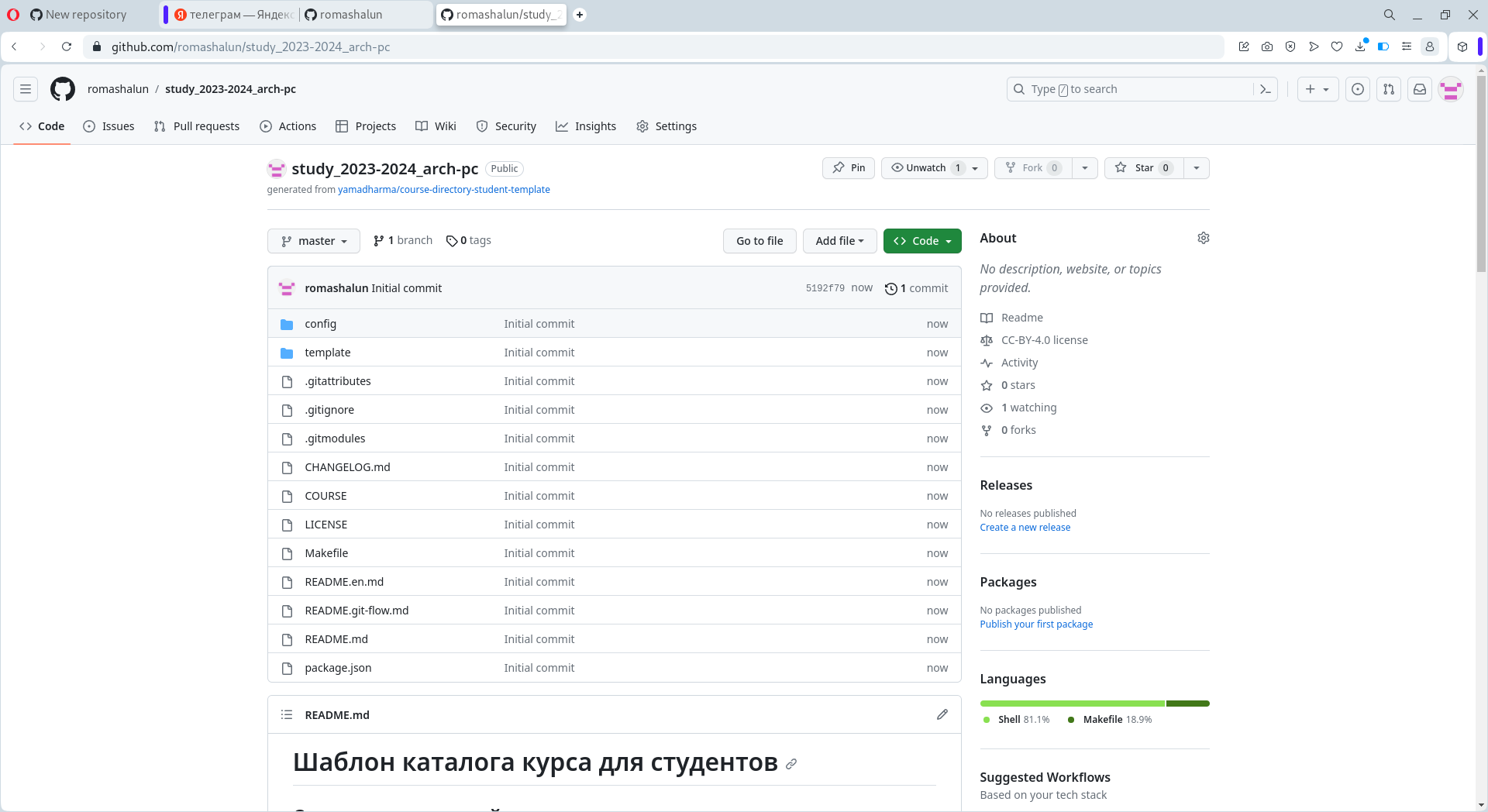


Figure 9: Создание рабочего пространства

## 3.4 Создание репозитория курса на основе шаблона

В браузере перехожу на страницу репозитории с шаблонами курса. Далее выбираю «Use this template», чтобы использовать этот шаблон для своего репозитория.(рис. ??)

Через терминал перехожу в созданный каталог курса с помощью утилиты cd(рис. [10](#fig:fig12))

Figure 10: Перемещение с помощью cd

Figure 10: Перемещение с помощью cd

Копирую ссылку для клонирования на странице созданного репозитория, сначала перейдя в окно «code», далее выбрав в окне вкладку «SSH» (рис. [11](#fig:fig13))

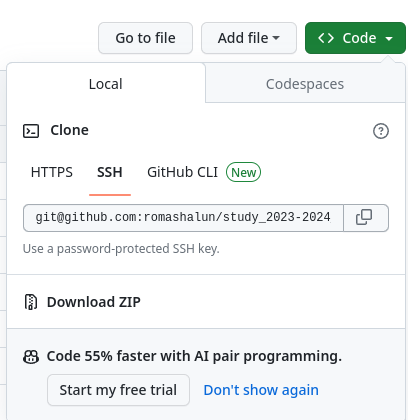


Figure 11: Окно с ссылкой для копирования репозитория

Клонирую созданный репозиторий с помощью команды git clone – recursive git@github.com:/study\_2023–2024\_arh-pc.git arch-pc (рис. [12](#fig:fig14))

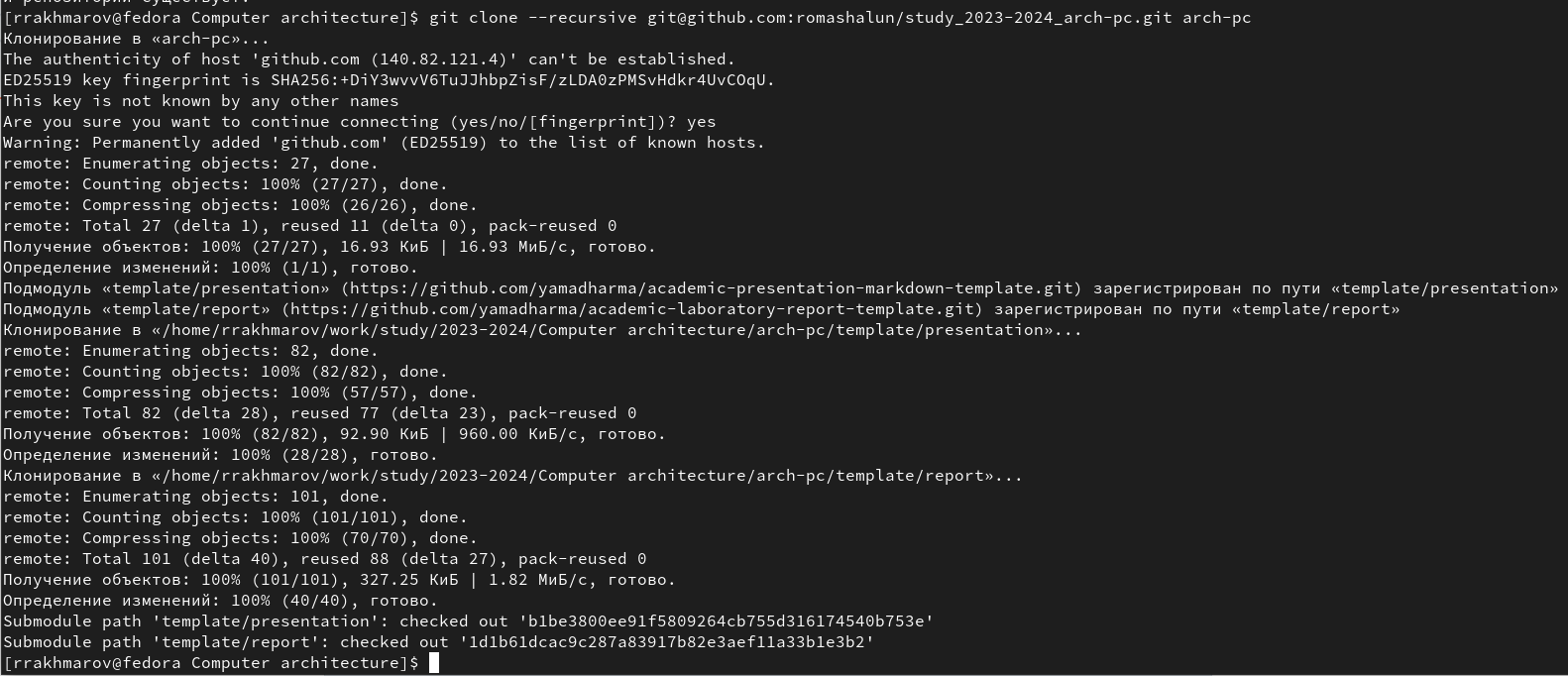


Figure 12: Клонирование репозитория

## 3.5 Настройка каталога курса

Перехожу в каталог arch-pc с помощью утилиты cd (рис. [13](#fig:fig15))

Figure 13: Перемещение между директориями

Figure 13: Перемещение между директориями

Удаляю лишние файлы с помощью утилиты rm (рис. [14](#fig:fig16))

Figure 14: Удаление файлов

Figure 14: Удаление файлов

Создаю необходимые каталоги. (рис. [15](#fig:fig17))

Figure 15: Создание каталогов

Figure 15: Создание каталогов

Отправляю созданные каталоги на сервер и сохраняю изменения. (рис. [16](#fig:fig18))

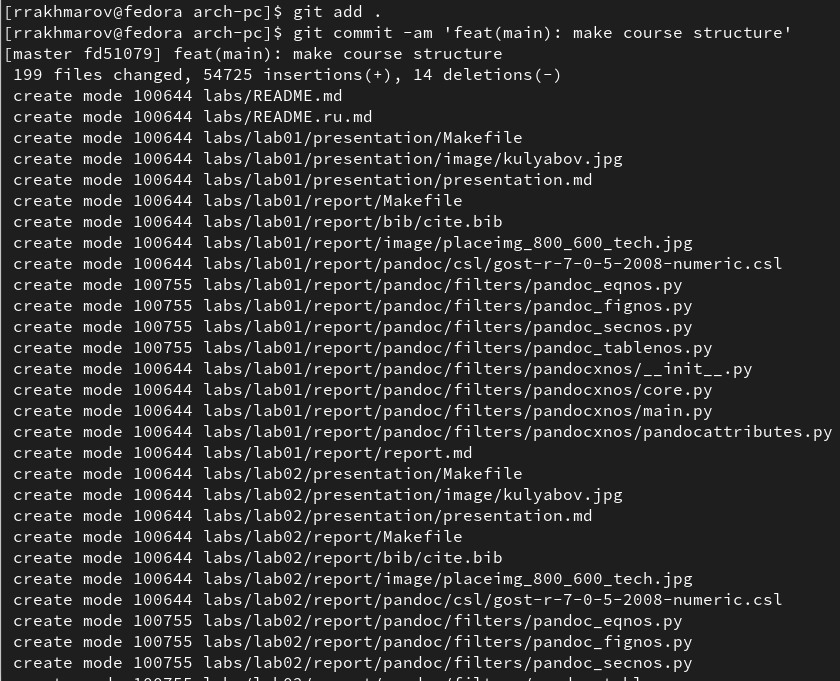


Figure 16: Добавление и сохранение изменений на сервере

Отправляю все на сервер с помощью команды git push (рис. [17](#fig:fig19))

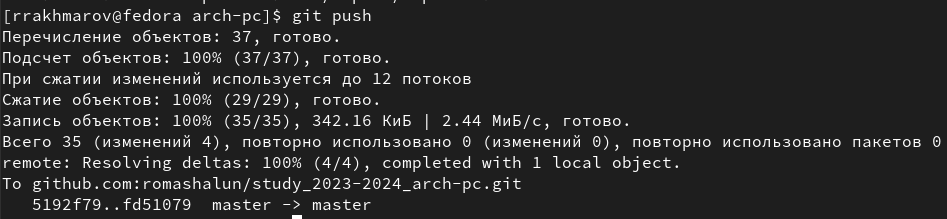


Figure 17: Выгрузка изменений на сервер

Проверяю правильность выполнения работы на самом сайте GitHub. Вижу что каталоги созданы недавно, это те которые мы отправили на сервер. (рис. [18](#fig:fig20))

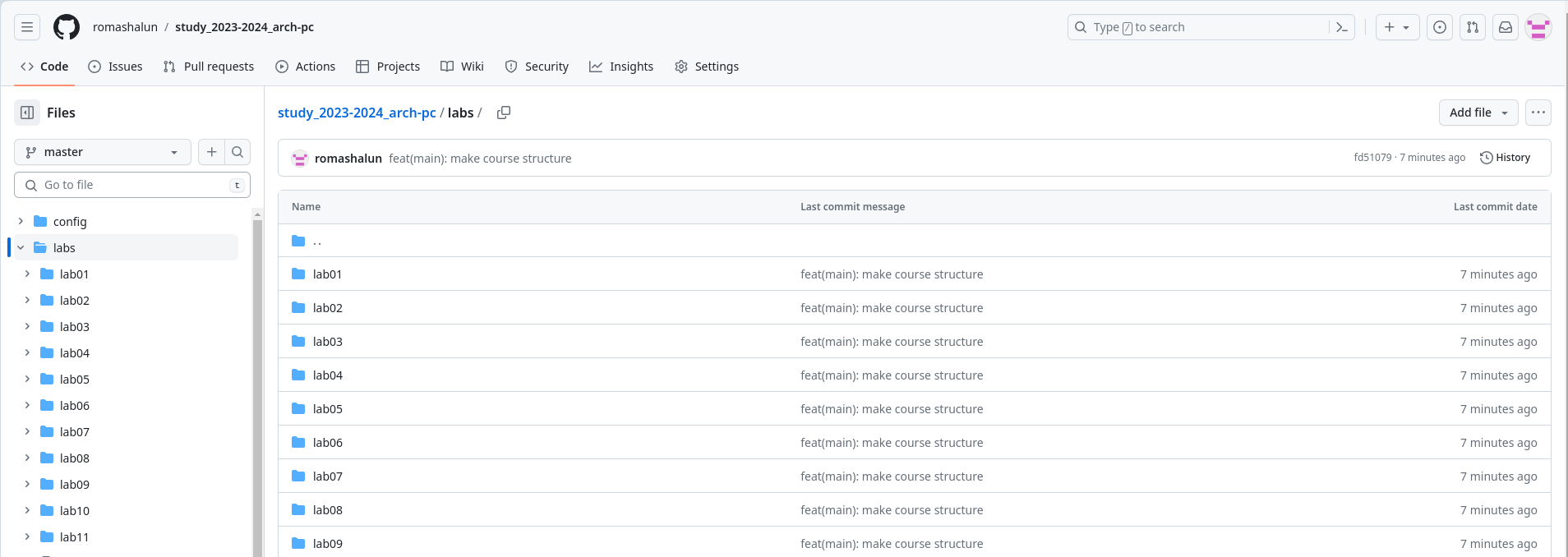


Figure 18: Страница репозитория

## 3.6 Выполнение заданий для самостоятельной работы.

Перехожу в директорию labs/lab02/report с помощью утилиты cd. Создаю в каталоге файл для отчета по третьей лабораторной работе с помощью утилиты touch. (рис. [19](#fig:fig21))

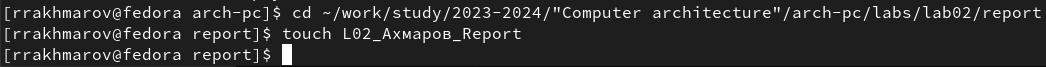


Figure 19: Создание файла

После открытия текстового редактора открываю в нем созданный файл и могу начать в нем работу над отчетом. (рис. [20](#fig:fig22))

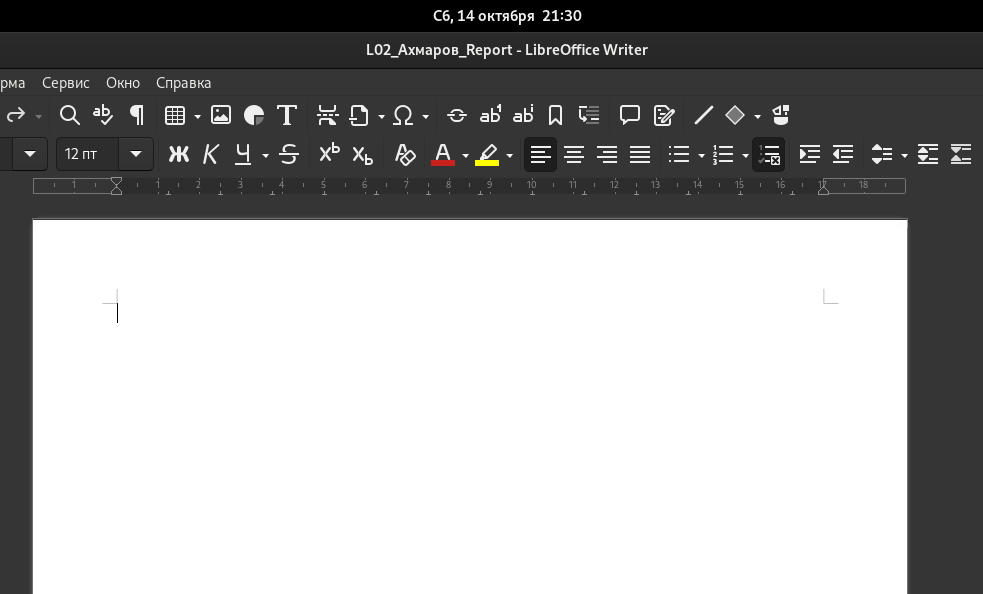


Figure 20: Работа с отчетом в текстовом редакторе

Перехожу из подкаталога lab02/report в подкаталог lab01/report с помощью утилиты cd. (рис. [21](#fig:fig23))

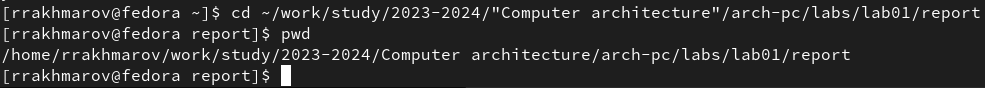


Figure 21: Перемещение между директориями

Проверяю местонахождение файлов с отчетами по первой лабораторной работe. Она должна быть в подкаталоге домашней директории «Документы», для проверки использую команду ls.(рис. [22](#fig:fig24))

Figure 22: Проверка местанахождения файлов

Figure 22: Проверка местанахождения файлов

Копирую первую лабораторную с помощью утилиты cp и проверяю правильность выполнения команды cp с помощью ls.(рис. [23](#fig:fig25))



Figure 23: Копирование файла

Добавляю файл L01\_Akhmarov\_Report.doc на сервер.(рис. [24](#fig:fig26))

Figure 24: Добавление файла на сервер

Figure 24: Добавление файла на сервер

Сохраняю изменения на сервере командой git commit -m “…”, поясняя, что добавил файлы. То же самое делаю для отчета по третьей лабораторной работе: перехожу в директорию labs/lab02/report с помощью cd, добавляю с помощью git add нужный файл, сохраняю изменения с помощью git commit. (рис. [25](#fig:fig27))

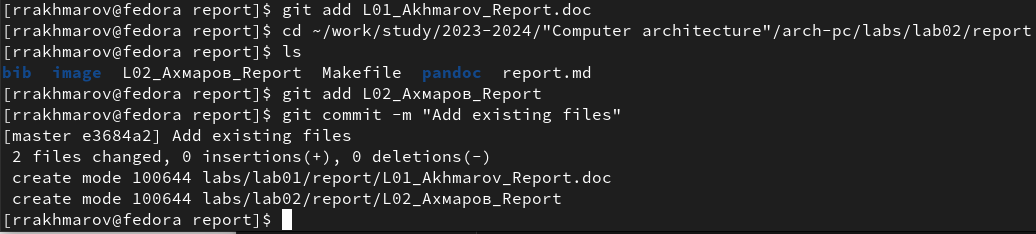


Figure 25: Подкаталоги и файлы в репозитории

Отправляю в центральный репозиторий сохраненные изменения командой git push -f origin master. (рис. [26](#fig:fig28))

Figure 26: Отправка в центральный репозиторий сохраненных изменений

Figure 26: Отправка в центральный репозиторий сохраненных изменений

Проверяю на сайте GitHub правильность выполнения заданий. Вижу, что отчеты по лабораторным работам находятся в соответствующих каталогах репозитория.(рис. [27](#fig:fig29))

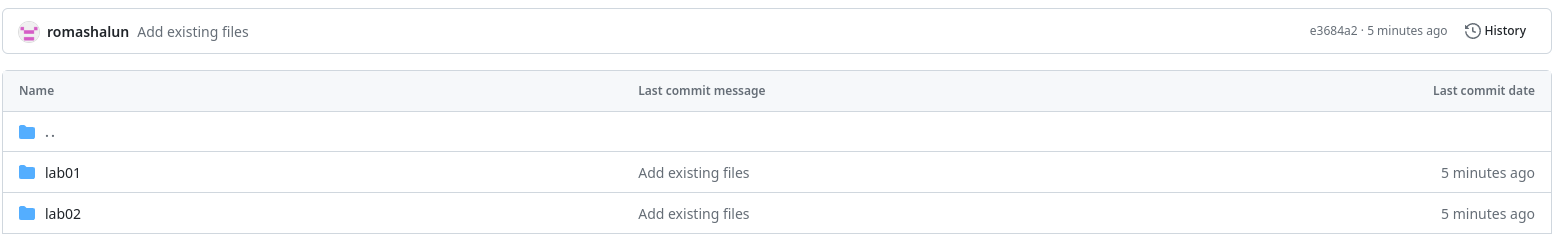


Figure 27: Каталог lab01, lab02/report

# 4 Вывод

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

# 5 Список литературы

1. https://esystem.rudn.ru/pluginfile.php/2091228/mod\_resource/content/0/Лабораторная%20работа%20№2.%20Система%20контроля%20версий%20Git.pdf