Отчет по лабораторной работе №2

Дисциплина: Архитектура компьютера

Карпачев Ярослав Олегович

Содержание

# 1 Цель работы

Целью работы является изучить Git. Приобрести практические навыки по работе с системой git.

# 2 Задание

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.
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. 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. Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения.  
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.  
В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.  
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.

Основные команды git{#tbl:std-dir}

| Команда | Описание |
| --- | --- |
| git init | создание основного дерева репозитория |
| git pull | получение обновлений (изменений) текущего дерева из центрального репозитория |
| git push | отправка всех произведённых изменений локального дерева в центральный репозиторий |
| git status | просмотр списка изменённых файлов в текущей директории |
| git diff | просмотр текущих изменения |
| git add | добавить все изменённые и/или созданные файлы и/или каталоги |
| git add имена\_файлов | добавить конкретные изменённые и/или созданные файлы и/или каталоги |
| git rm имена\_файлов | удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории) |
| git commit -am 'Описание коммита' | сохранить все добавленные изменения и все изменённые файлы |
| git checkout -b имя\_ветки | создание новой ветки, базирующейся на текущей |
| git checkout имя\_ветки | переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой) |
| git push origin имя\_ветки | отправка изменений конкретной ветки в центральный репозиторий |
| git merge --no-ff имя\_ветки | слияние ветки с текущим деревом |
| git branch -d имя\_ветки | удаление локальной уже слитой с основным деревом ветки |
| git branch -D имя\_ветки | принудительное удаление локальной ветки |
| git push origin :имя\_ветки | удаление ветки с центрального репозитория |

# 4 Выполнение лабораторной работы

## 4.1 Настройка GitHub

Создал учетную запись на сайте https://github.com/ (рис. 1).

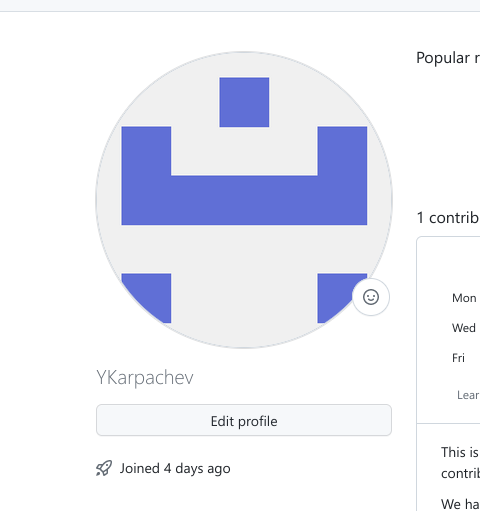


Рис. 1: Мой профиль в github

## 4.2 Базовая настройка Git

Произвел базовые настройки Git, emain, name и тд (рис. 2).

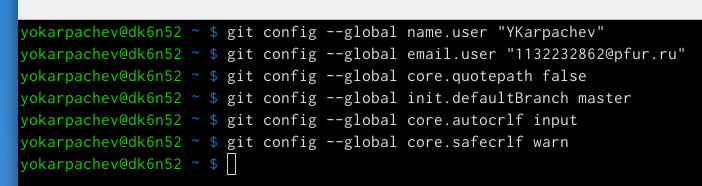


Рис. 2: Настройка git

## 4.3 Создание SSH-ключа

Сгенерировал ключи приватный и открытый (рис. 3). Просмотрел с помощью cat, скопировал его, вставил в Github (рис. 4). Авторизовала ключ (рис. 5).

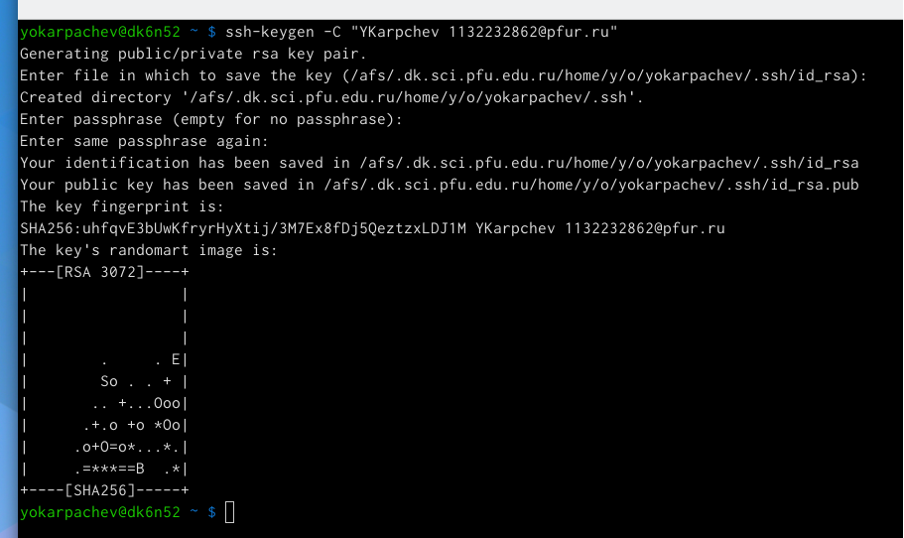


Рис. 3: Создание SHH ключа

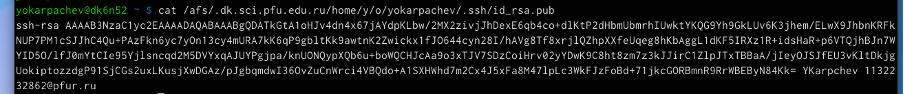


Рис. 4: Открытый ключ SHH

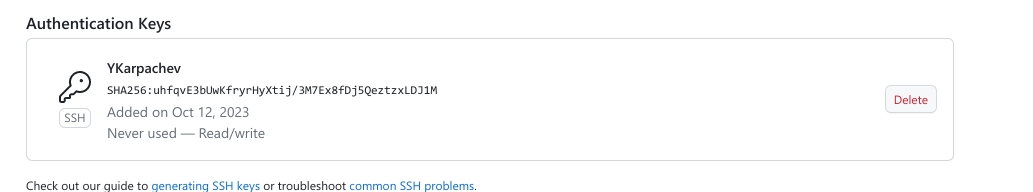


Рис. 5: Лист авторизованных ключей

## 4.4 Создание репозитория курса на основе шаблона

На основе шаблона создал свой репозиторий (рис 6.)

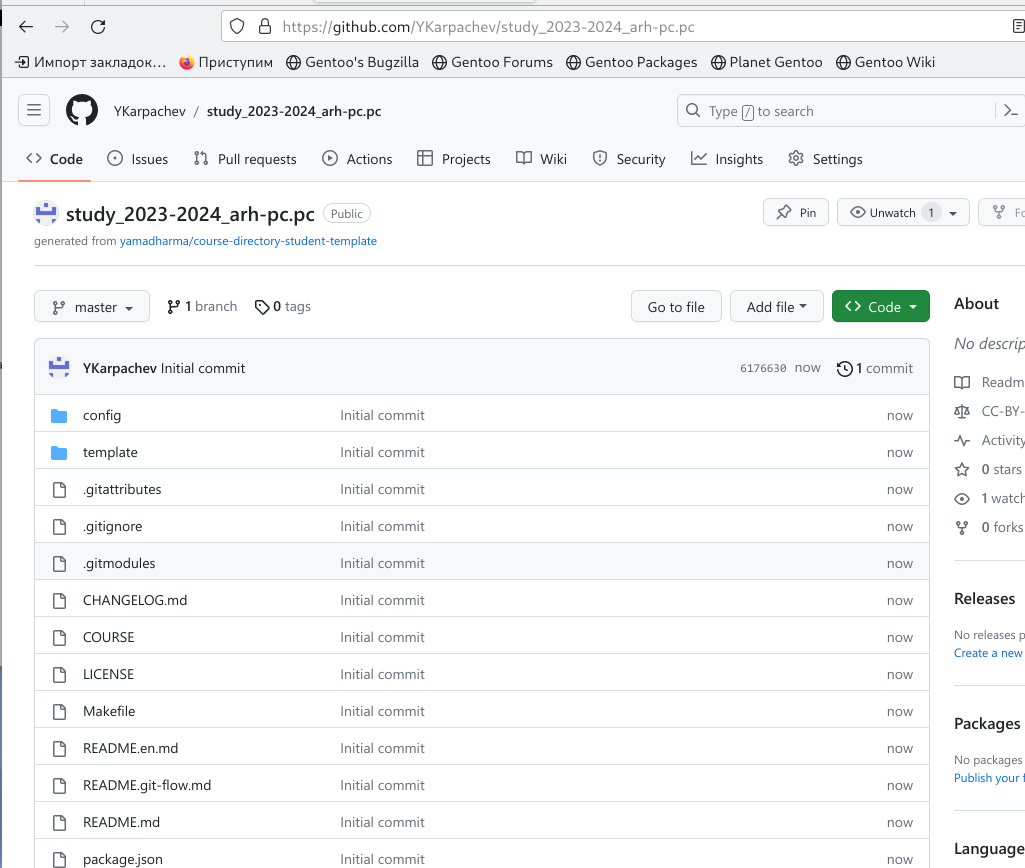


Рис. 6: Страница созданого репозитория

Открыл терминал и перешел в каталог. Клонировала созданный репозиторий, скопировав ссылку для клонирования на странице созданного репозитория Code -> SSH (рис. 7).

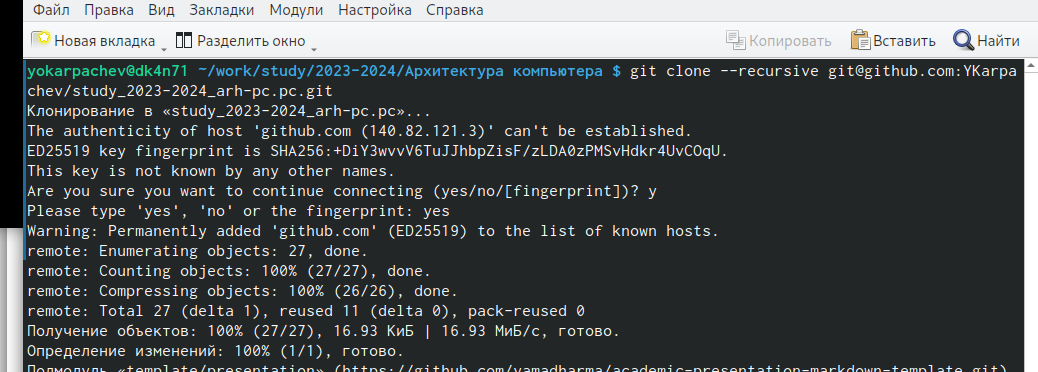


Рис. 7: Клонирование репозитория

## 4.5 Настройка каталога курса

Удалил лишние файлы с помощью команды rm (рис. 8).

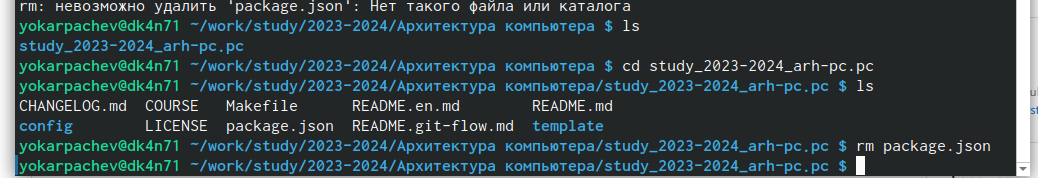


Рис. 8: Удаление лишних файлов

создал необходимые каталоги с помощью git add, прокомментировал и сохранил изменения на сервере как добавление с помощью git commit (рис. 9), и отправил изменения на сервер (рис. 10) и проверил на гитхаб равильность действий (рис. 11)

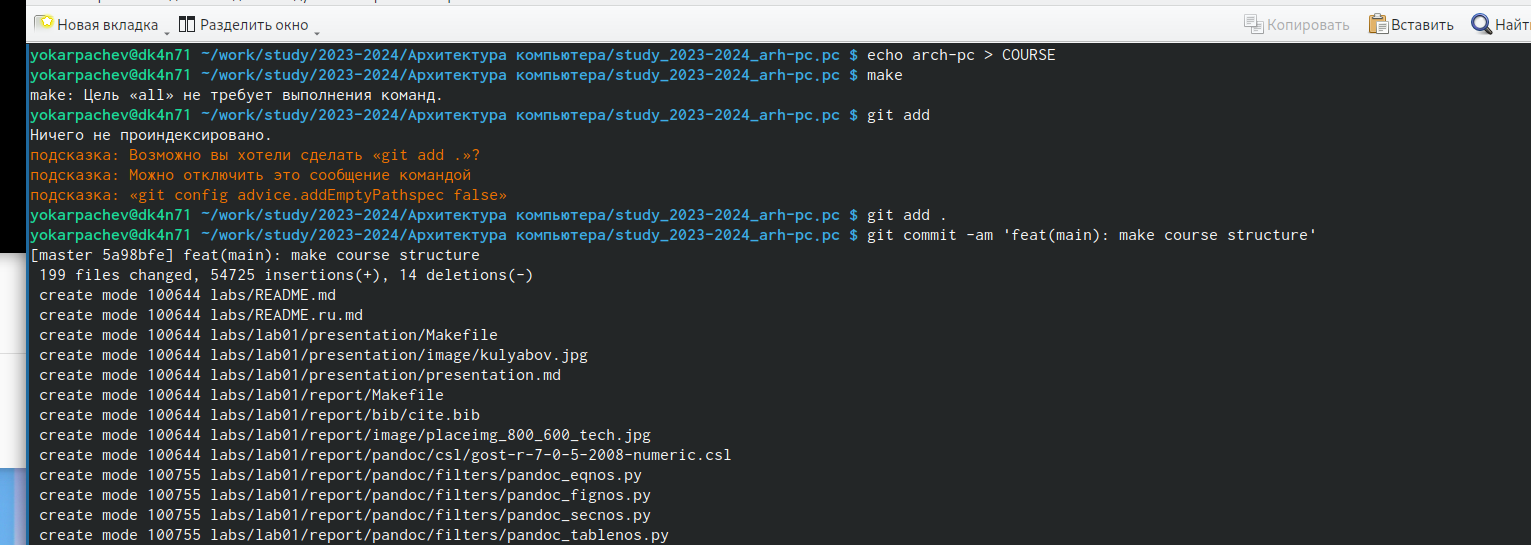


Рис. 9: Создание необходимых каталогов и сохранение изменений с git commit

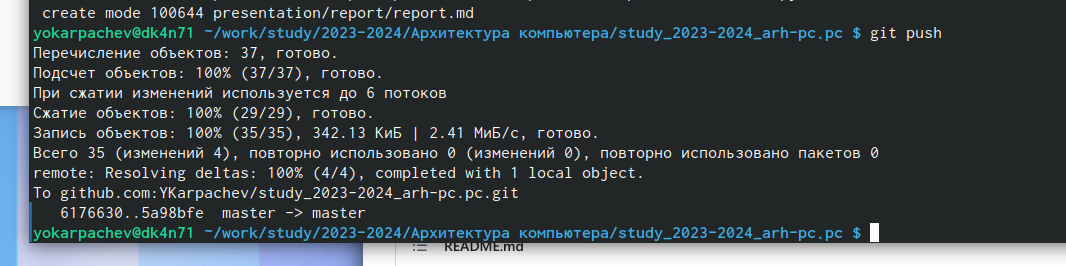


Рис. 10: Отправка файлов на сервер

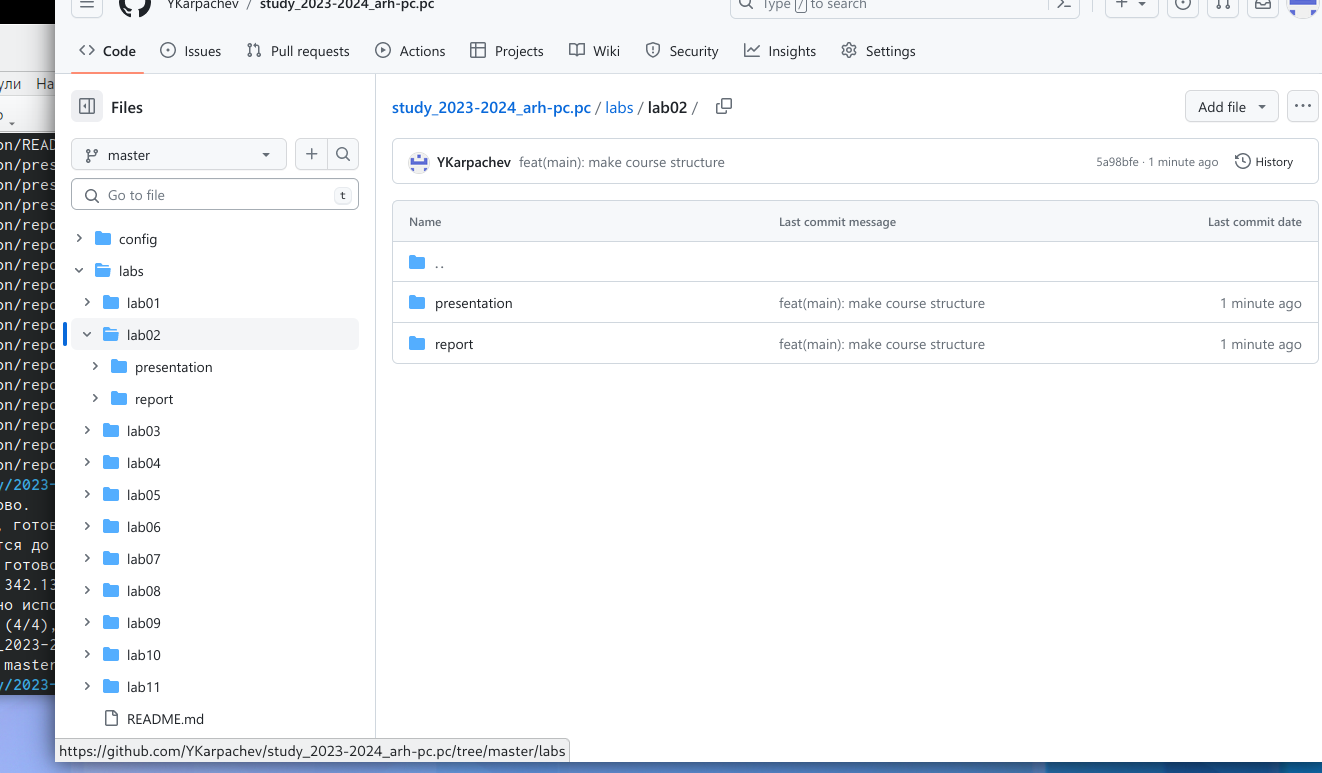


Рис. 11: Страница репозитория

## 4.6 Выполнение заданий для самостоятельной работы

Перехожу в директорию labs/lab02/report с помощью утилиты cd. Создаю в каталоге файл для отчета по второй лабораторной работе с помощью утилиты touch, оформио его и отправил на сервер. (рис. 12).

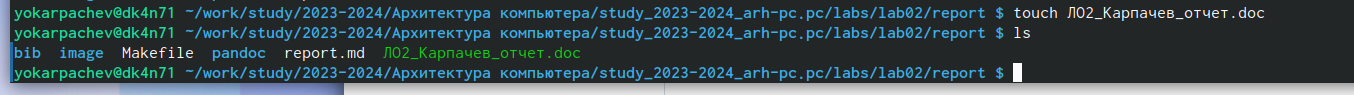


Рис. 12: Создание текстового документа для оформления отчета и загрузка его на сервер

Перенес (копировал) и загрузил предыдущий отчет на сервер в каталог labs/lab01/report. И проверил правильность действий на Github.

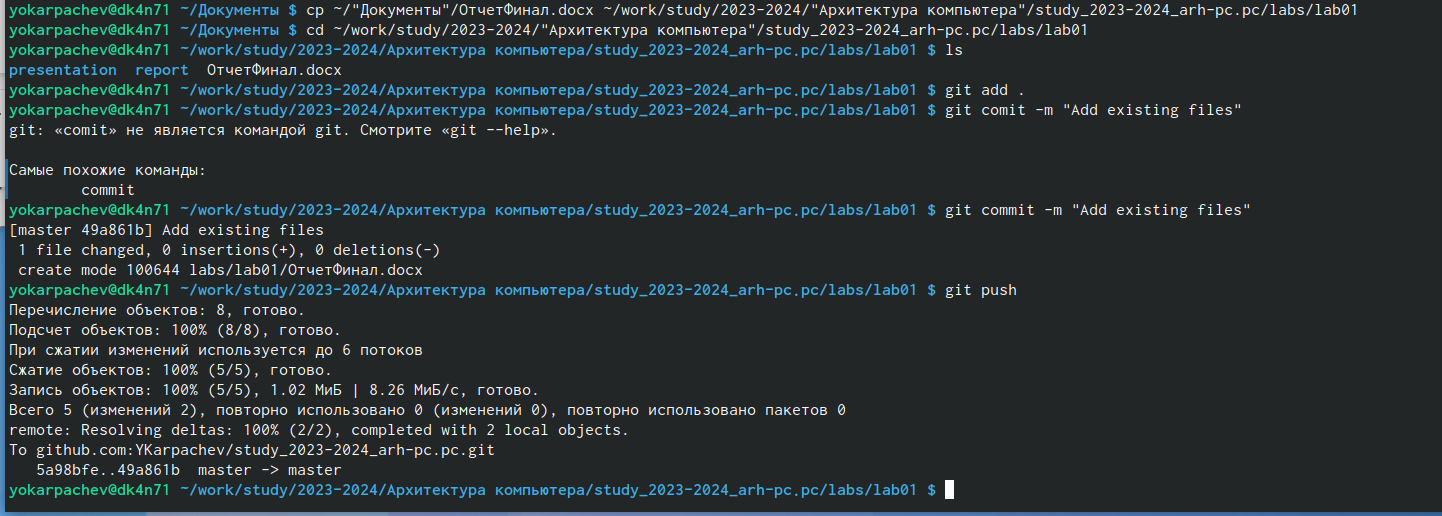


Рис. 13: добавление и отправка файлов на сервер

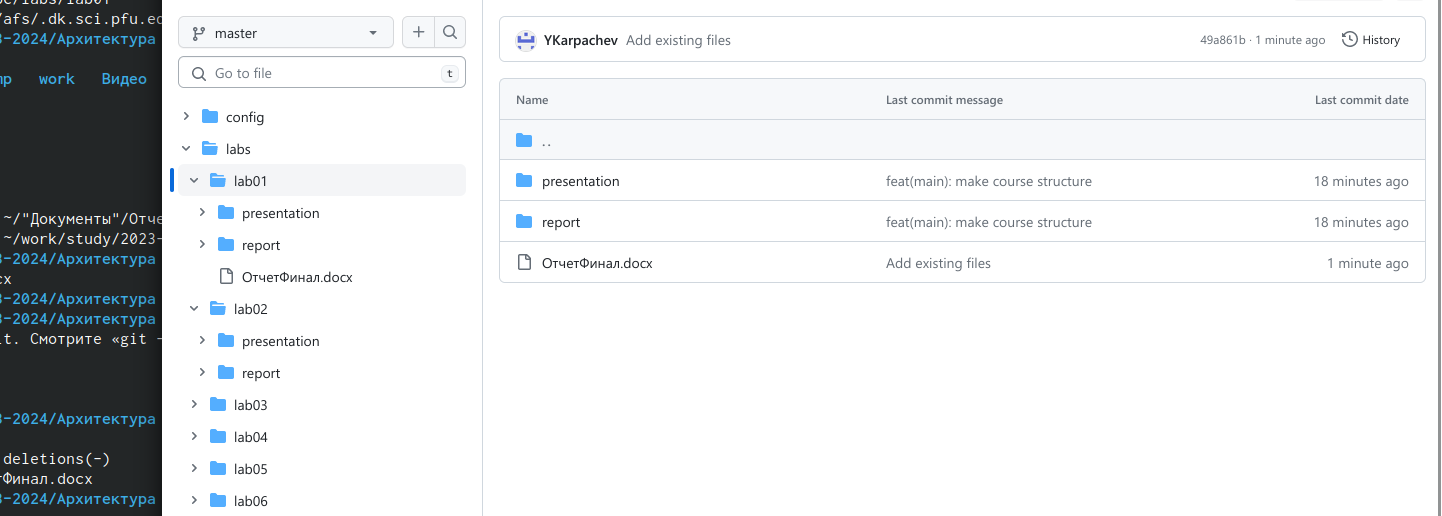


Рис. 14: проверка своих действий

# 5 Выводы

В ходе лабораторной работы я получил практически навыки работы с системой git, а также изучил Git посредством выполнения практических заданий.