Отчёт по лабораторной работе №2

Дисциплина: Архитектура компьютера

София Андреевна Кудякова

Содержание

# 1 Цель работы

Цель данной работы - изучить идеологию и применение средств контроля версий, а также научиться работать 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 на основе шаблона.
6. Настройка каталога курса
7. Выполнение заданий для самостоятельной работы.

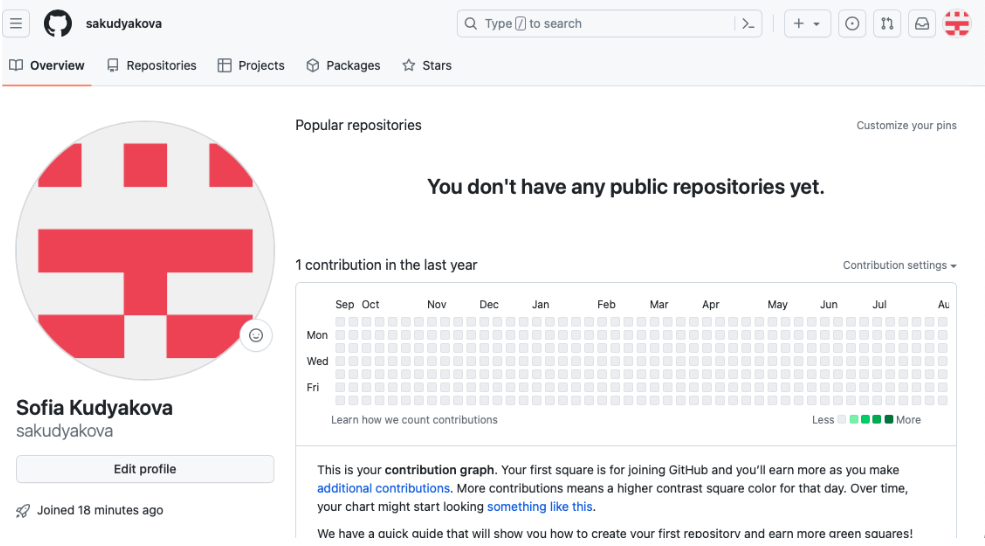
# 3 Теоретическое введение

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. 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 не полную версию изменённых файлов, а производить так называемую дельта- компрессию - сохранять только изменения между последовательнымиверсиями, что позволяет уменьшить объём хранимых данных. Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.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. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить. 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

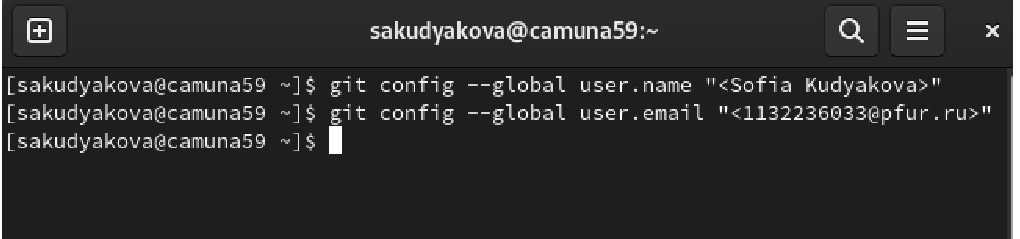
Создаю учётную запись на сайте https://github.com/ и заполняю основные данные. (рис. ??).



Создание учетной записи GitHub

## 4.2 Базовая настройка Git

Делаю предварительную конфигурацию git. Открываю терминал и ввожу следующие команды, указав имя и email своего репозитория. (рис. ??).



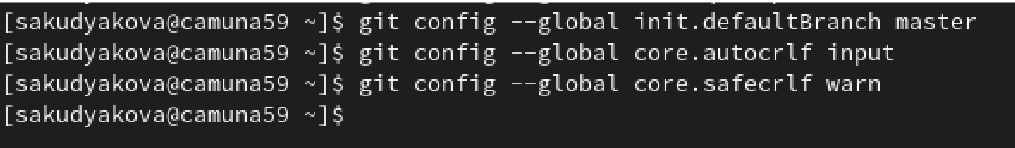
Предварительная конфигурация git

Настраиваю utf-8 в выводе сообщений git (рис. ??).

Настройка utf-8 в выводе сообщений git

Настройка utf-8 в выводе сообщений git

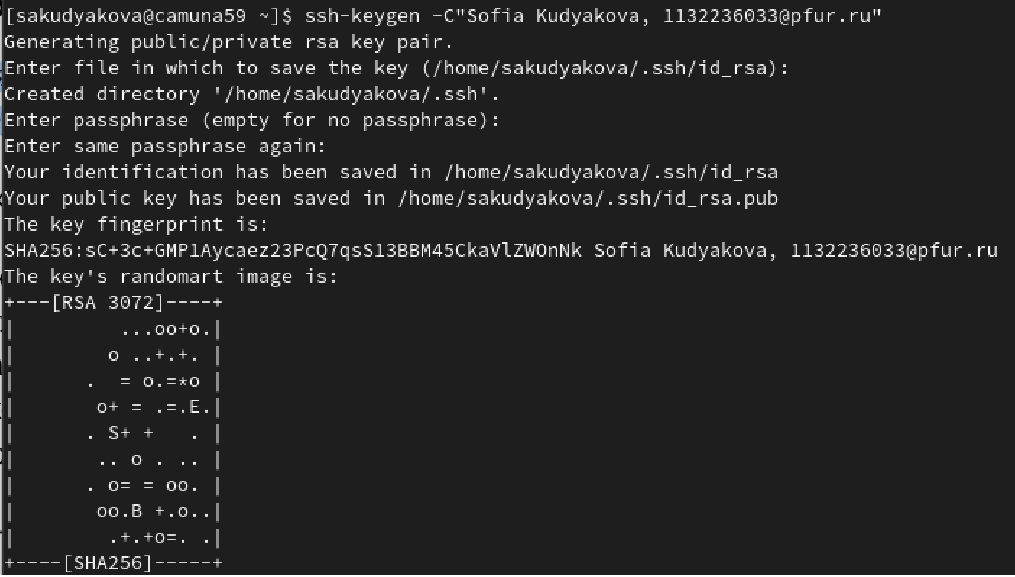
Задаю имя начальной ветки (будем называть её master). Далее устанавливаю параметры autocrlf и safecrlf (рис. ??).



Создание имени для начальной ветки. Параметры autocrlf и safecrlf

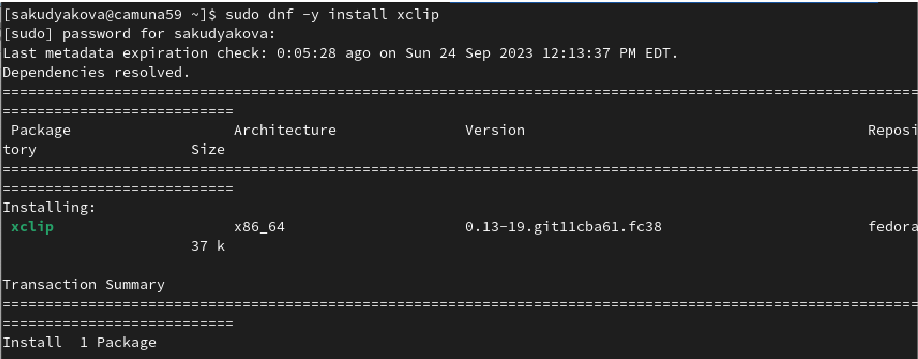
## 4.3 Создание SSH ключа

Для последующей идентификации пользователя на сервере репозиториев необходимо сгенерировать пару ключей (приватный и открытый) (рис. ??). Ключи сохранятся в каталоге ~/.ssh/



Создание SSH-ключа

Устанавливаю xclip. (рис. ??).



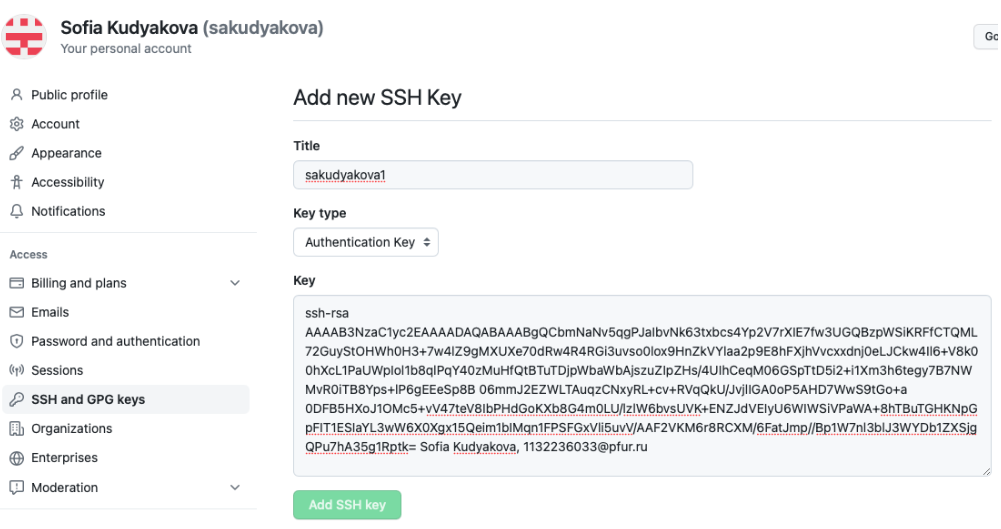
Установка xclip

С помощью xclip копирую сгенерированный ключ. (рис. ??).

Копирование с помощью xclip

Копирование с помощью xclip

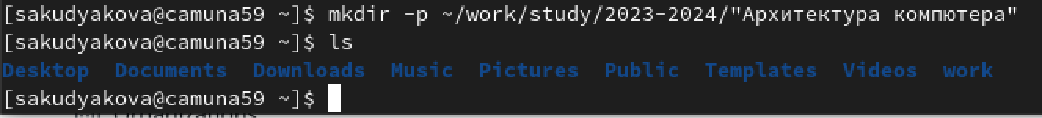
Далее загружаю сгенерированный открытый ключ. Захожу на GitHub под своей учётной записью и перехожу в меню, выбираю в боковом меню “SSH and GPG keys”. Копирую ключ из локальной консоли в буфер обмена. Вставляю ключ в появившееся на сайте поле и указываю для ключа имя (Title). (рис. ??).



Добавление SSH key

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

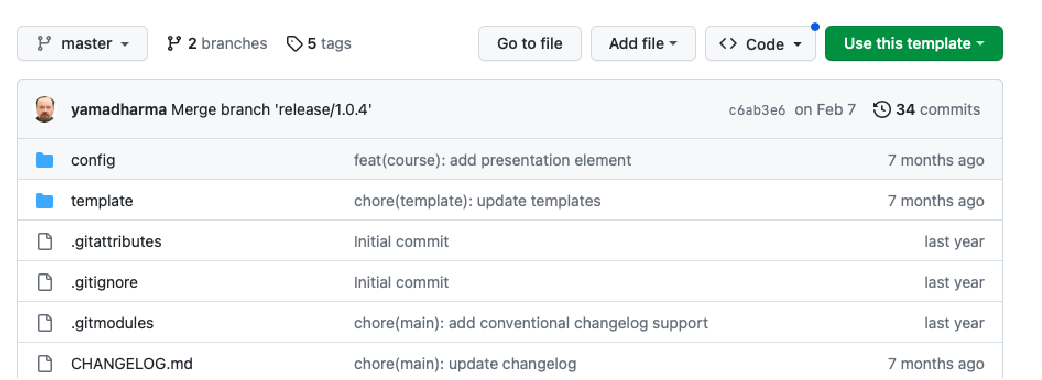
Открываю терминал и создаю каталог для предмета «Архитектура компьютера», использовав команду mkdir с ее опцией -p, которая, в свою очередь, позволяет рекрусивно создать все директории. С помощью команды ls проверяю корректность совершенных действий. (рис. ??).



Создание рабочего пространства

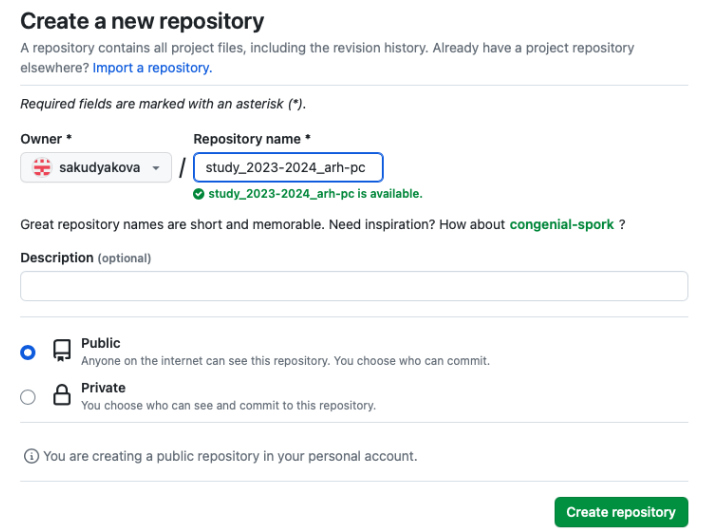
## 4.5 Создание репозитория курса на основе шаблона

Создаю репозиторий на основе шаблона через web-интерфейс GitHub. Перехожу на страницу репозитория с шаблоном курса и выбираю “Use this template”. (рис. ??).



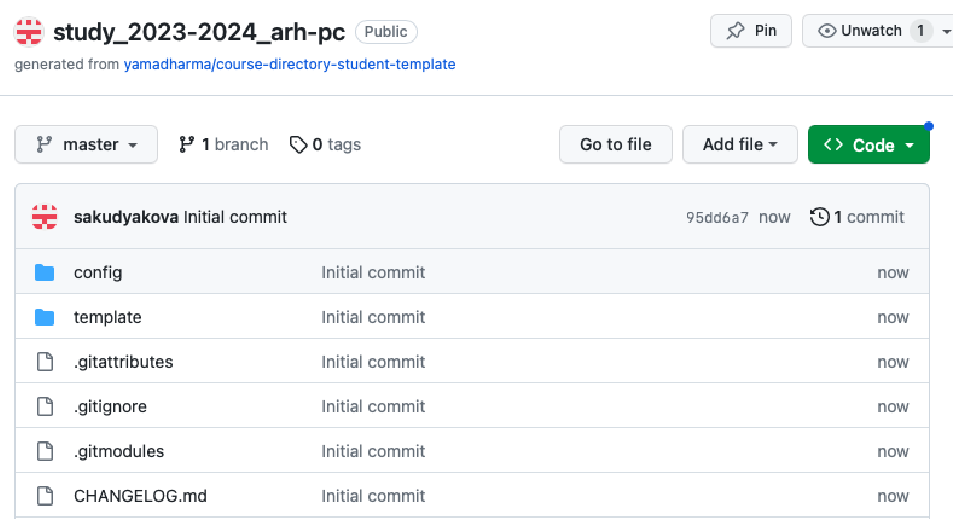
Выбор шаблона

В открывшемся окне задаю имя репозитория (Repository name) study\_2023–2024\_arh-pc и создаю репозиторий, нажав на кнопку “Create repository” (рис. ??).



Создание репозитория

Убеждаюсь в том, что репозиторий был создан. (рис. ??).



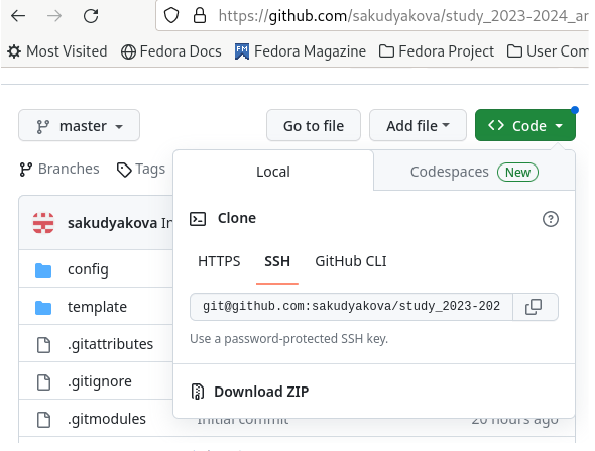
Созданный репозиторий

Далее открываю терминал и перехожу в каталог курса, используя команду cd. (рис. ??).

Переход в каталог курса

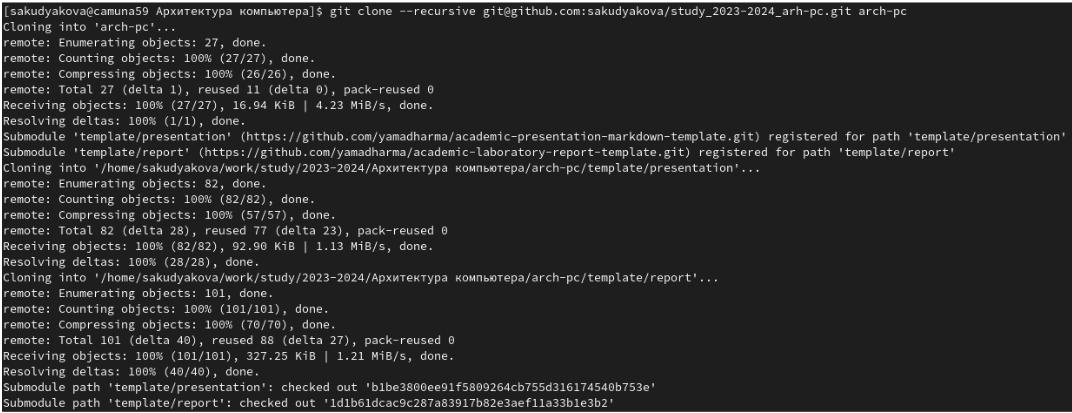
Переход в каталог курса

Копирую ссылку на странице созданного репозитория Code – SSH. (рис. ??).



Копирование ссылки

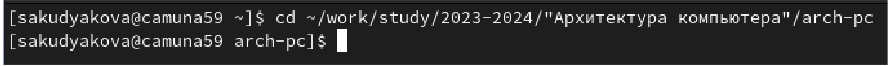
Клонирую созданный репозиторий. (рис. ??).



Клонирование репозитория

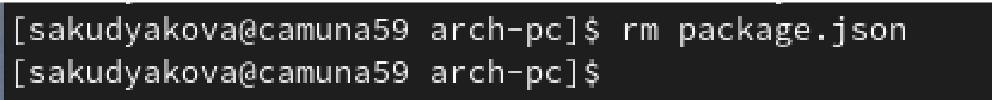
## 4.6 Настройка каталога курса

Перехожу в каталог курса. (рис. ??).



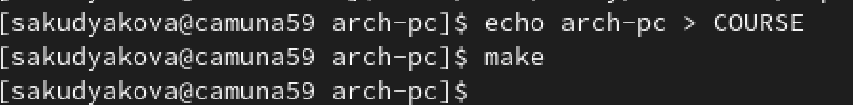
Переход в каталог курса

С помощью команды rm удаляю лишние файлы, а именно файл “package.json”. (рис. ??).



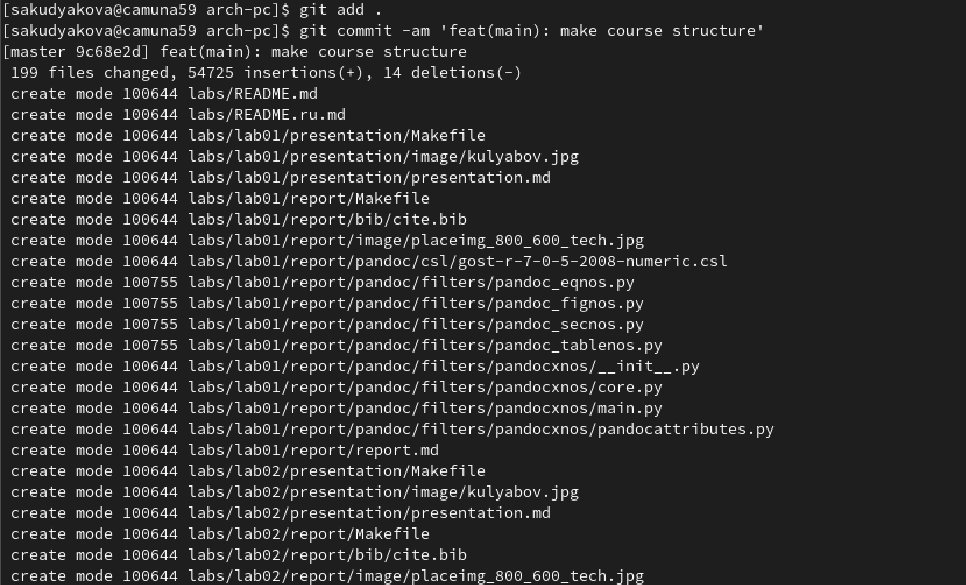
Удаление файлов

Создаю необходимые каталоги. (рис. ??).



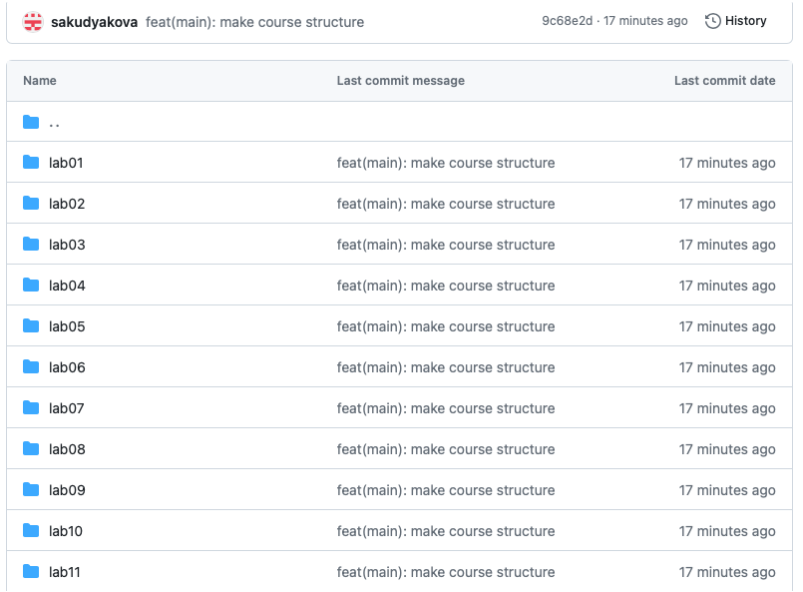
Создание необходимых каталогов

Отправляю файлы на сервер. (рис. ??).



Отправка файлов на сервер

Проверяю корректность создания иерархии рабочего пространства в локальном репозитории и на странице GitHub. (рис. ??).



Проверка иерархии создания рабочего пространства на GitHub

#Задания для самостоятельной работы

1. Создание отчета по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report).

С помощью команды cd перехожу в каталог labs/lab02/report. Далее с помощью утилиты touch создаю файл для отчета по выполнению второй лабораторной работы. (рис. ??).

Создание файла

Создание файла

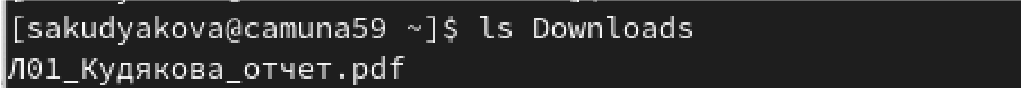
1. Копирование отчетов по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства.

С помощью команды cd перехожу в каталог lab01/report. (рис. ??).

Переход в каталог

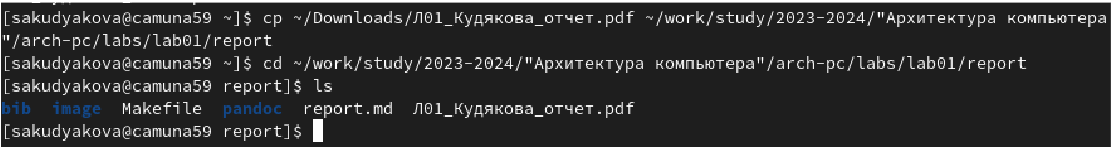
Переход в каталог

С помощью команды ls проверяю, есть ли в подкаталоге “Downloads” файл с отчетом по первой лабораторной работы. (рис. ??).



Содержимое “Downloads”

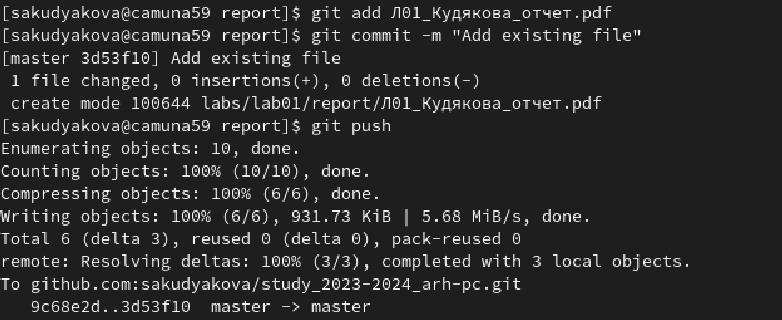
Убеждаюсь в том, что файл есть. Далее копирую отчет в каталог lab01/report с помощью cd и проверяю корректность выполненных действий с помощью ls. (рис. ??).



Копирование файла в каталог

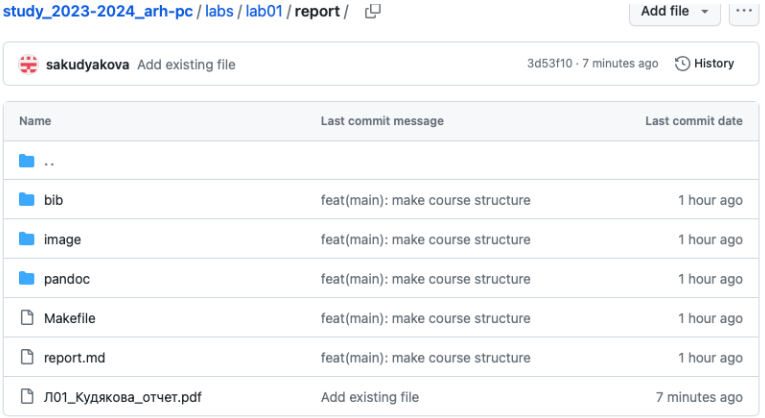
1. Загрузка файлов на GitHub.

С помощью команды git add добавляю файлы. Далее сохраняю изменения, используя команду git commit -m и добавляя пояснение “Add existing file”. После чего использую команду git push, чтобы отправить файлы в репозиторий. (рис. ??).



Загрузка файлов на GitHub

Захожу на GitHub, чтобы проверить действительно ли загрузились файлы. (рис. ??).



Добавление файла на GitHub

Такие же действия выполняю, чтобы добавить отчет по второй лабораторной работе на GitHub.

# 5 Выводы

В ходе данной лабораторной работы я изучила идеологию и применение средств контроля версий, а также научилась работать с системой git.