Отчёт по лабораторной работе №2

Дисциплина:архитуктура компьютера

Елизавета Александровна Киселева

Содержание

# 1 Цель работы

Целью данной лабораторной работы является изучить идеологию и применение средств контроля версий. 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. 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 последовательными версиями, 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, 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. 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. Работа пользователя со своей веткой начинается с проверки и получения изменений из центрального репозитория (при этом в локальное дерево до начала этой процедуры не должно было вноситься изменений). Затем можно вносить изменения в локальном дереве и/или ветке. После завершения внесения какого-то изменения в файлы и/или каталоги проекта необходимо разместить их в центральном репозитории.

# 4 Выполнение лабораторной работы

## 4.1 Настройка Github

Первым делом я создаю учетную запись на сайте GitHub (рис. 1).

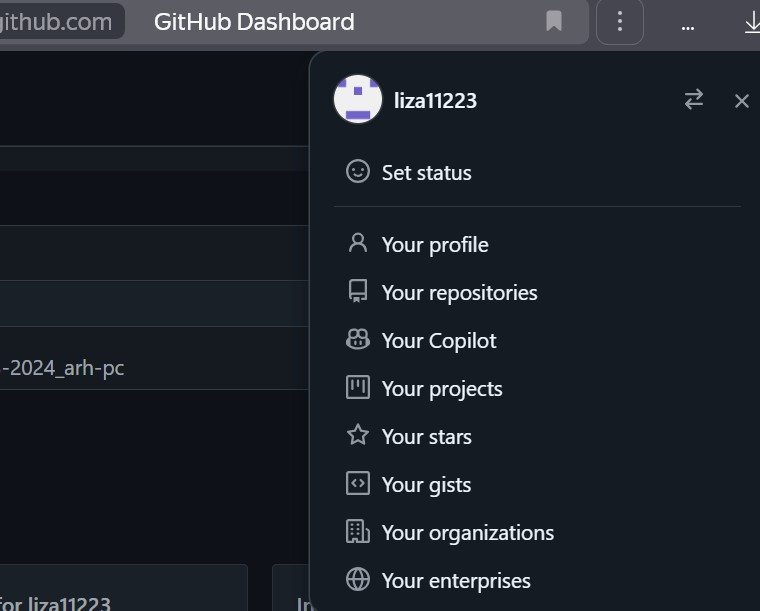


Рис. 1: учетная запись

## 4.2 Базовая настройка Git

Сначала делаю предварительную конфигурацию git. Открываю терминал и ввожу следующие команды: git config –global user.name ’’, указывая свое имя пользователя, а затем git config – global user.email ’’, указывая свою почту(рис. 2).

Рис. 2: Предварительная конфигурация git

Рис. 2: Предварительная конфигурация git

Далее настраиваю utf-8 в выводе сообщений git, задаю имя начальной ветки (master), далее использую параметр autocrlf и safecrlf(рис. 3).

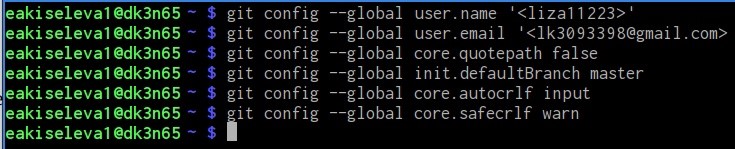


Рис. 3: настройка и создание

## 4.3 Создание SSH-ключа

Для последующей идентификации пользователя на сервере репозиториев я сгенерировала пару ключей: приватный и открытый (рис. 4).

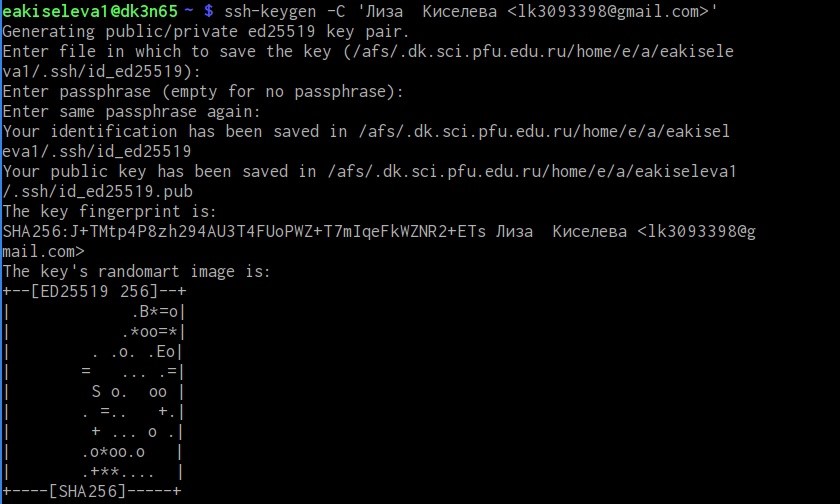


Рис. 4: Генерация SSH-ключа

Далее копирую открытый ключ из директории, в которой он был сохранен, с помощью команды xclip (рис. 5).

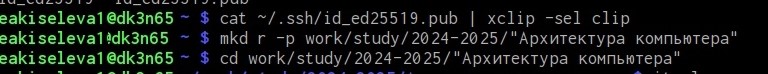


Рис. 5: Установка утилиты xclip

Далее захожу на сайт GitGub, открываю свой профиль и выбираю страницу “SSH and GPG keys”. Нажимаю на кнопку “New SSH key”, вставляю в поле скопированный ключ, в поле ‘Title’ указываю имя ключа, затем нажимаю “Add SSHkey”. (рис. 6).

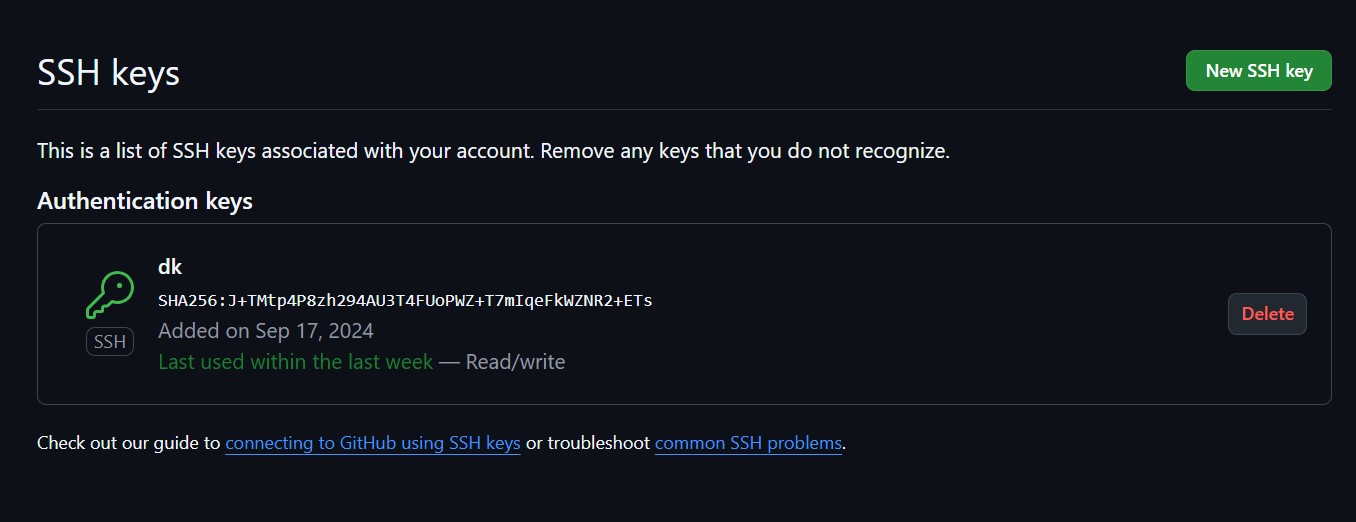


Рис. 6: Копирование содержимого файла

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

Открываю терминал, создаю директорию, рабочее пространство с помощью утилиты mkdir, с помощью ключа –p создаю рекурсивно все директории после домашней ~/work/study/20242025/”Архитектура компьютера” (рис. 7)

Рис. 7: выполнение команд

Рис. 7: выполнение команд

В браузере перехожу на страницу репозитория с шаблоном курса по адресу https://github.com/yamadharma/course-directory-student-template, выбираю use this template, чтобы использовать данный шаблон для своего репозитория. В открывшемся окне задаю название репозитория: study\_2024-2025\_arch-pc и создаю репозиторий, нажимая на кнопку “Create repository” (рис. 8)

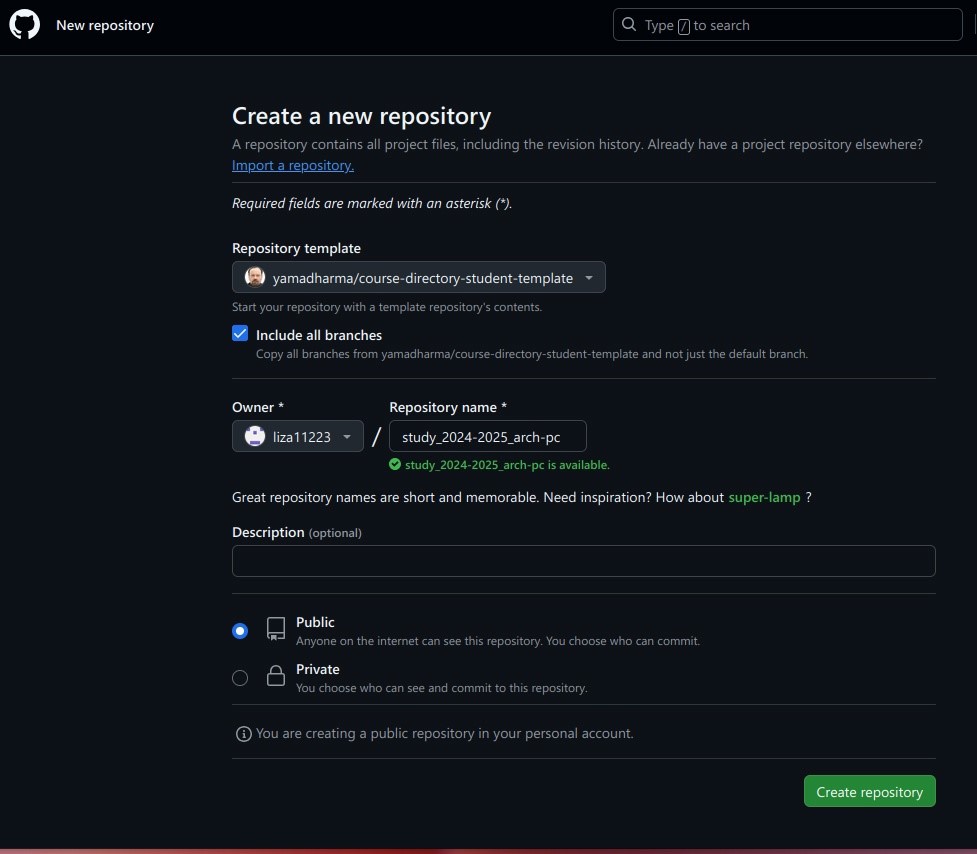


Рис. 8: Страница шаблона для репозитория

Через терминал перехожу в созданный каталог курса с помощью утилиты cd (рис. 9)

Рис. 9: Перемещение между директориями

Рис. 9: Перемещение между директориями

Далее клонирую созданный репозиторий (рис. 10).

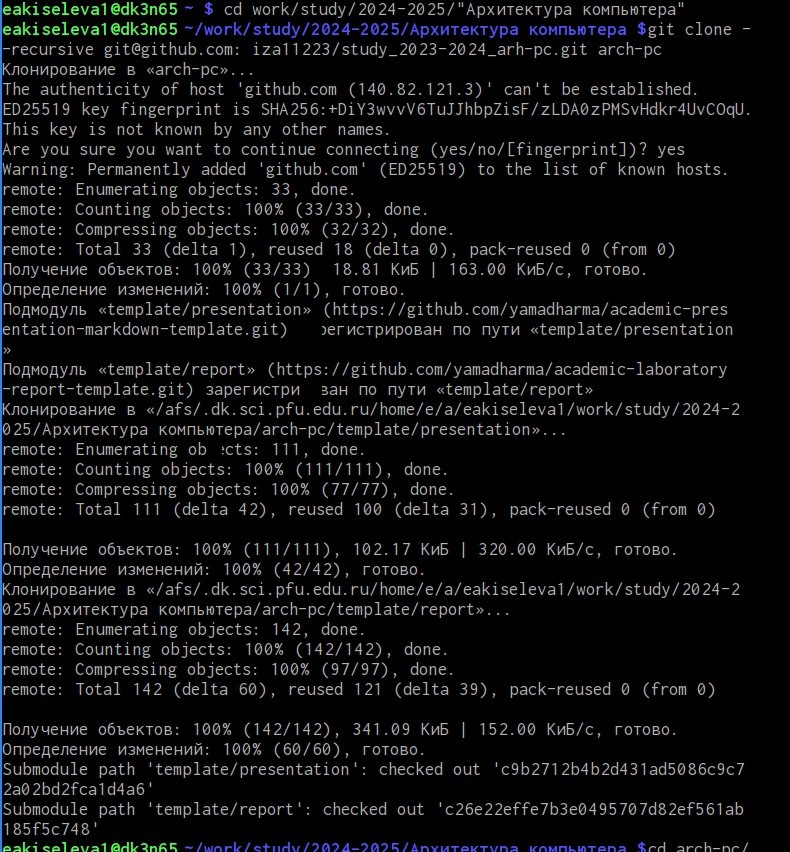


Рис. 10: Клонирование репозитория

Ссылку для клонирования копирую на странице созданного репозитория, перейдя сначала в окно ‘code’, выбрав затем вкладку ‘SSH’ (рис. 11).

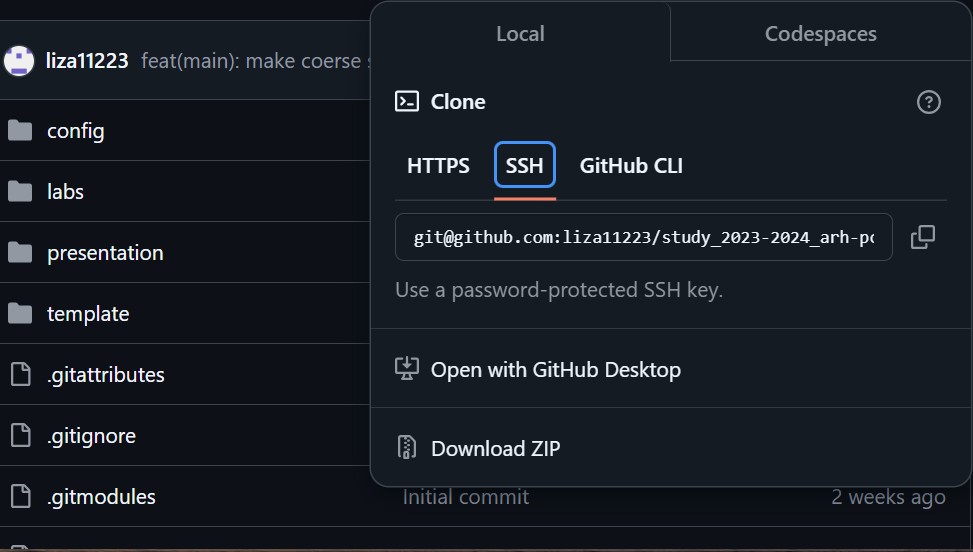


Рис. 11: Окно с ссылкой для копирования репозитория

## 4.5 Настройка каталога курса

Перехожу в каталог arch-pc с помощью утилиты cd и удаляю лишние файлы с помощью утилиты rm (рис. 12).

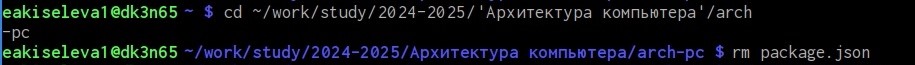


Рис. 12: Перемещение между директориями и удаление файлов

Затем все отправляю на сервер с помощью команды git push (рис. 13).

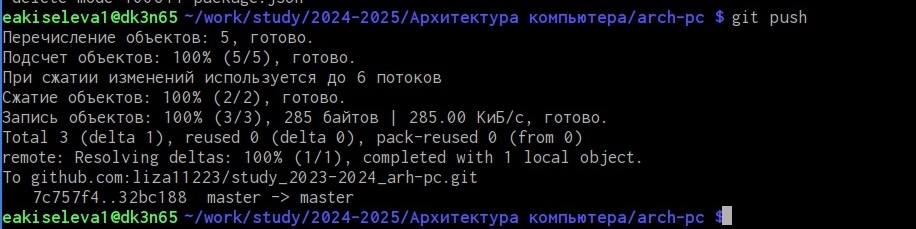


Рис. 13: Выгрузка изменений на сервер

## 4.6 Выполнение заданий для самостоятельной работы

Перехожу в директорию labs/lab02/report и создаю в каталоге файл для отчета по второй лабораторной работе с помощью команды touch (рис. 14).

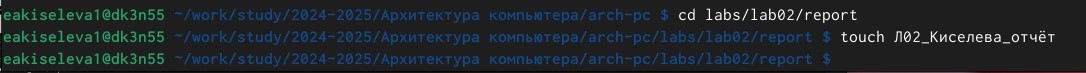


Рис. 14: Создание файла

# 5 Выводы

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

# Список литературы

1. [Архитектура ЭВМ](https://esystem.rudn.ru/pluginfile.php/1584622/mod_resource/content/1/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%963.pdf)
2. [Git - gitattributes Документация](https://git-scm.com/docs/gitattributes)