Отчет по лабораторной работе №2

Дисциплина: архитектура компьютера

Луцкая Алиса Витальевна

Содержание

# 1 Цель работы

Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.

# 2 Задание

4.1 Настройка github 4.2 Базовая настройка git 4.3 Создание ssh-ключа 4.4 Создание рабочего пространства и репозитория курса на основе шаблона 4.5 Создание репозитория курса на основе шаблона 4.6 Настройка каталога курса 5. Выполнение 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 про- екта. 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, пред- полагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник про- екта (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 хранили- ща 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 кон- фликтов, 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 заблоки- ровать 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.

# 4 Выполнение лабораторной работы

##Настройка GitHub

Создаю учетную запись на сайте GitHub и заполняю основные данные для учетной записи (рис. 1).

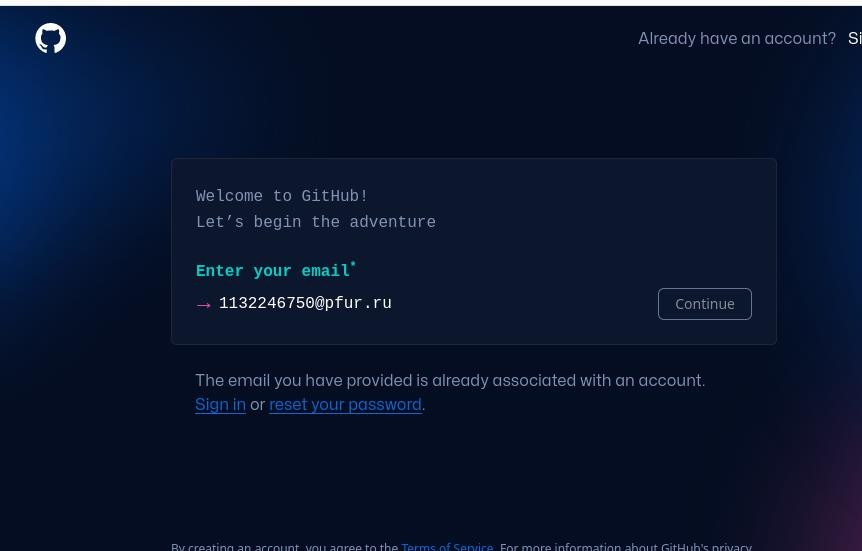


Рис. 1: Заполнение данных учетной записи GitHub

Аккаунт создан (рис. 2).

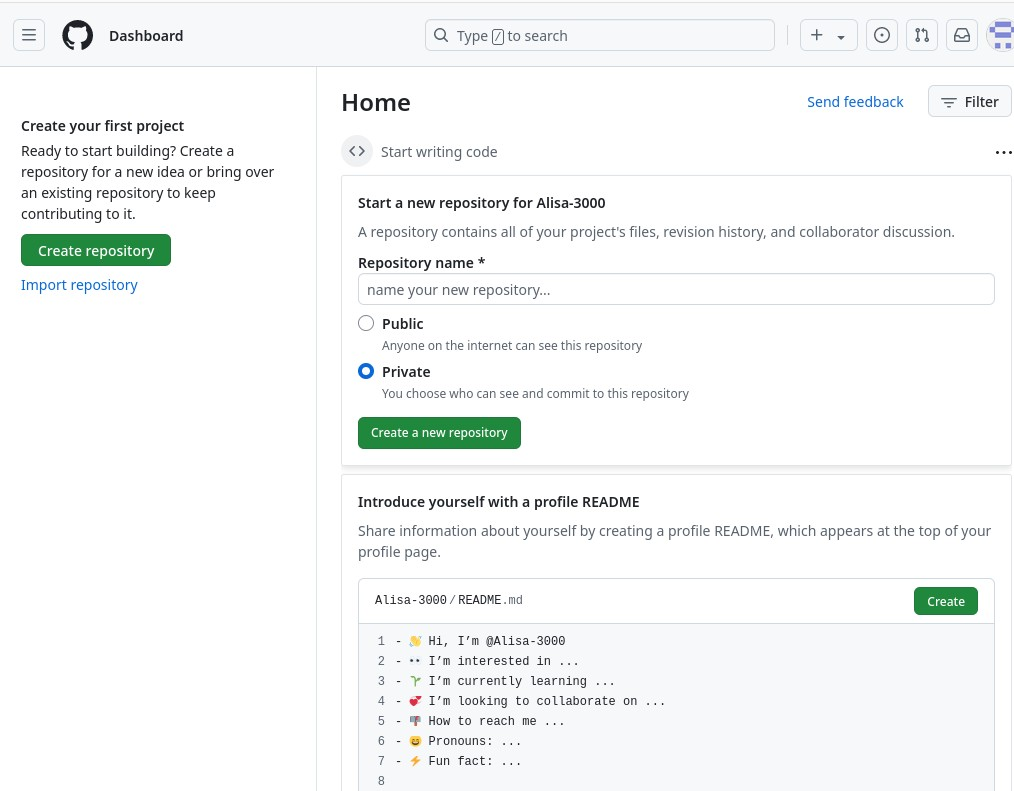


Рис. 2: Аккаунт GitHub

## 4.1 Базовая настройка Git

Произвожу базовую настройку git, для это открываю терминал и ввожу следующие команды: git config –global user.name “”, указывая свое имя и команду git config –global user.email “work@mail”, указывая в ней свою электронную почту (рис. 3).

Рис. 3: Предварительная конфигурация git

Рис. 3: Предварительная конфигурация git

Настраиваю utf-8 в выводе сообщений git(рис. 4).

Рис. 4: Настройка кодировки

Рис. 4: Настройка кодировки

Задаю имя начальной ветки, называю «master» (рис. 5).

Рис. 5: Создание имени для начальной ветки

Рис. 5: Создание имени для начальной ветки

Задаю параметр autocrlf со значением input (рис. 6)

Рис. 6: Параметр autocrlf

Рис. 6: Параметр autocrlf

Задаю параметр safecrlf со значением warn (рис. 7).

Рис. 7: Параметр safecrlf

Рис. 7: Параметр safecrlf

## 4.2 Создание SSH-ключа

Для последующей идентификации пользователя на сервере репозиториев необ- ходимо сгенерировать пару ключей (приватный и открытый). Для этого ввожу команду ssh-keygen -C “Имя Фамилия, work@email”, указывая свое имя и электронную почту (рис. 8)

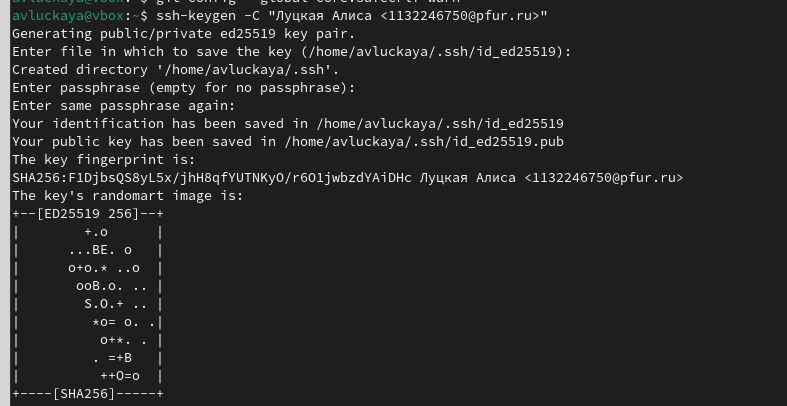


Рис. 8: Генерация SSH-ключа

Получаю сгенерированный открытый ключ, который буду вставлять на сайт http://github.org/ под своей учётной записью (рис. 9)

Рис. 9: Сгенерированный открытый ключ

Рис. 9: Сгенерированный открытый ключ

Открываю браузер, захожу на сайт GitHub. Открываю свой профиль и перехожу в настройки (рис. 10)

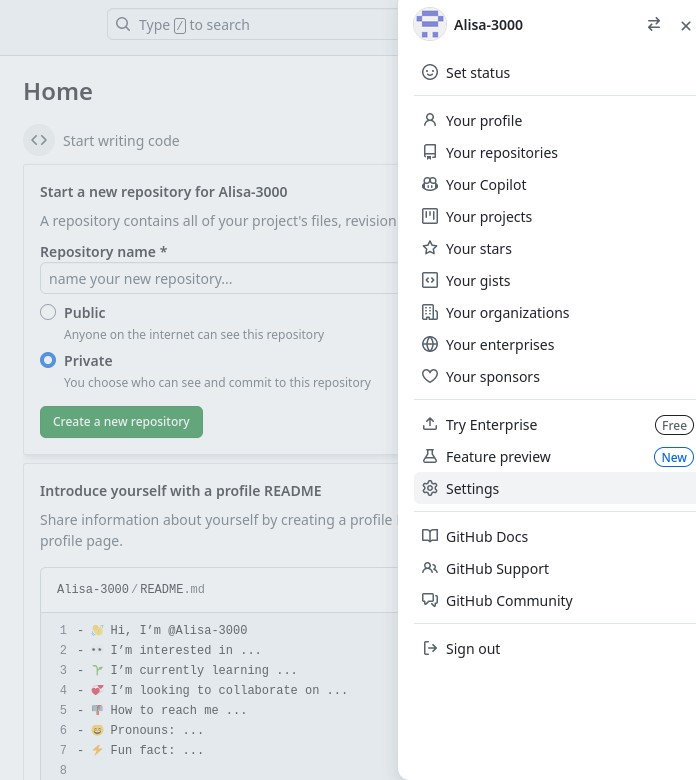


Рис. 10: Настройки

Выбираю страницу «SSH and GPG keys» (рис. 11)

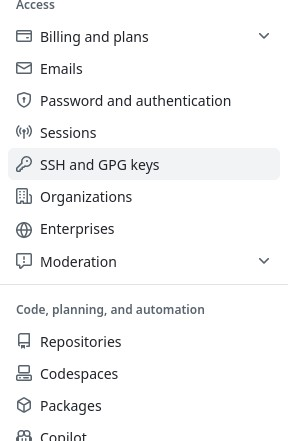


Рис. 11: SSH and GPG keys

Нажимаю кнопку «New SSH key» (рис. 12)

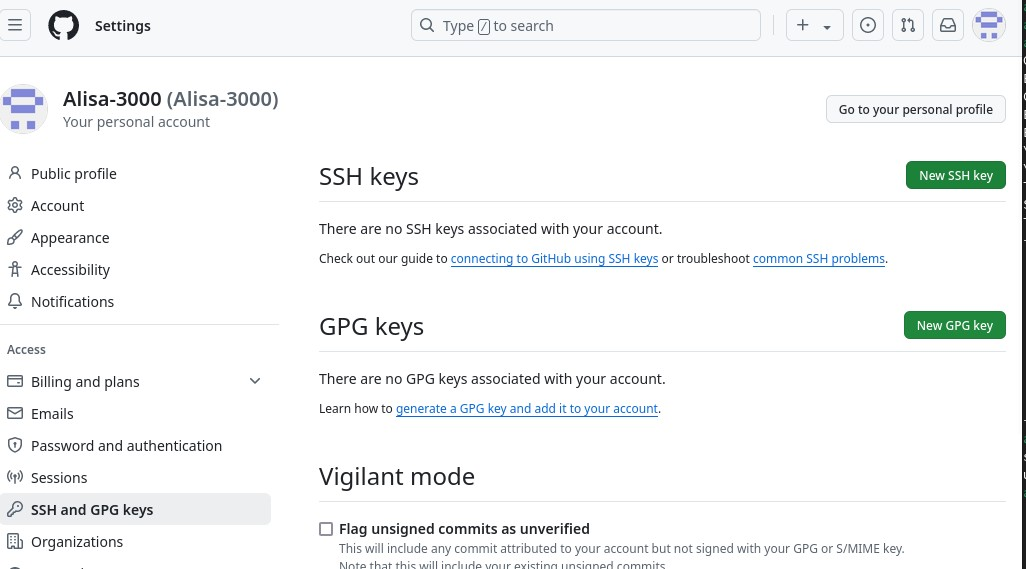


Рис. 12: New SSH key

Вставляю скопированный ключ в поле «Key». В поле Title указываю имя для ключа. Нажимаю «Add SSH-key», чтобы завершить добавление ключа (рис. 13)

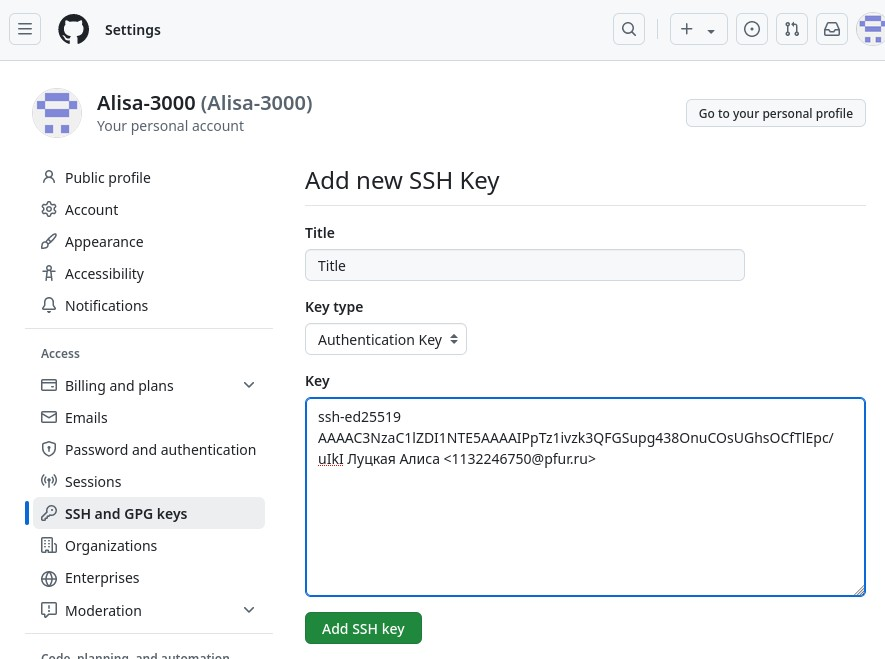


Рис. 13: Добавление ключа

Ключ создан (рис. 14)

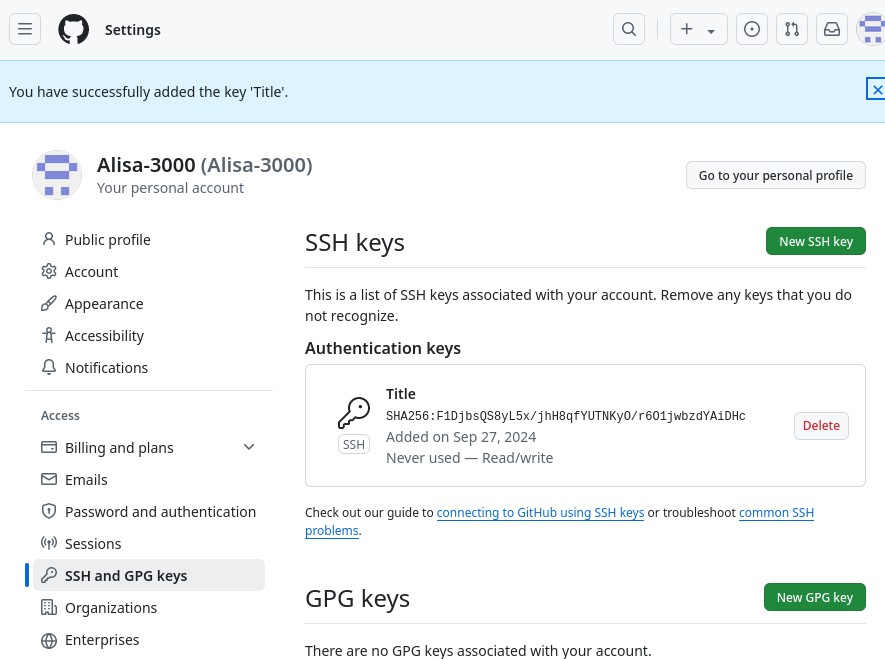


Рис. 14: Ключ Title

## 4.3 Создание рабочего пространства и репозитория курсана основе шаблона

В терминале создаю директорию, с помощью утилиты mkdir, благодаря ключу -p создаю все директории после домашней ~/work/study/2024- 2025/“Архитектура компьютера” рекурсивно. Далее проверяю с помощью ls, проверяю их создание (рис. 15)

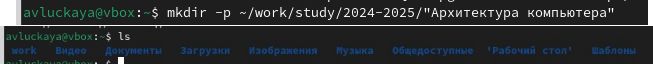


Рис. 15: Создание рабочего пространства

## 4.4 Создание репозитория курса на основе шаблона

В браузере перехожу на страницу репозитория с шаблоном курса по адресуhttps://github.com/yamadharma/course-directory-student-template. Выбираю «Use this template», чтобы использовать этот шаблон для репозитория (рис. 16)

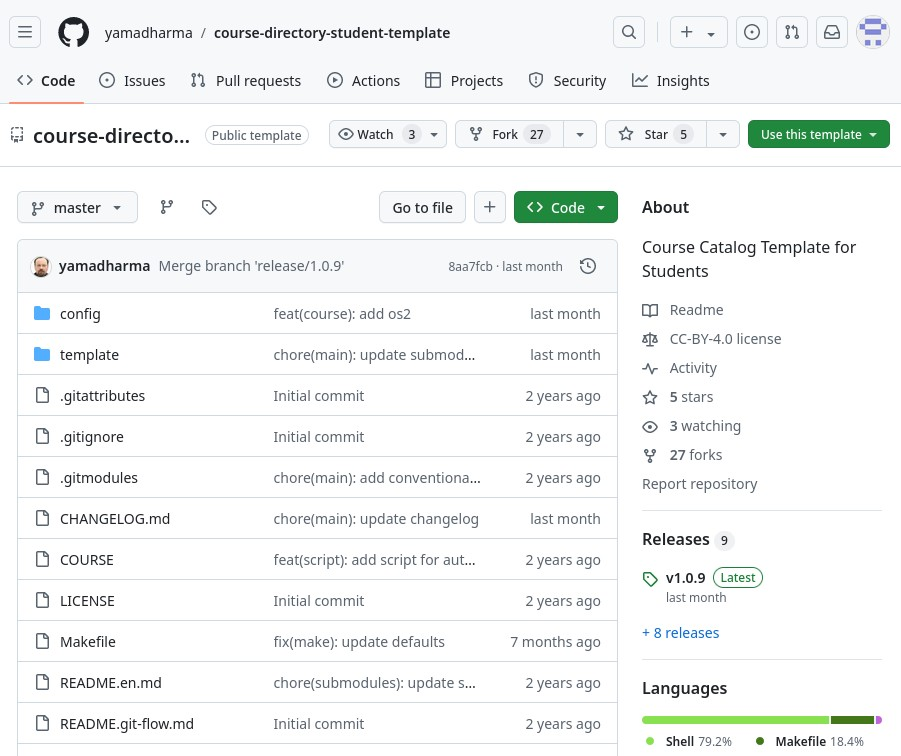


Рис. 16: Страница шаблона для репозитория

В открывшемся окне задаю имя и создаю репозиторий, нажимаю на кнопку «Create repository from template» (рис. 17)

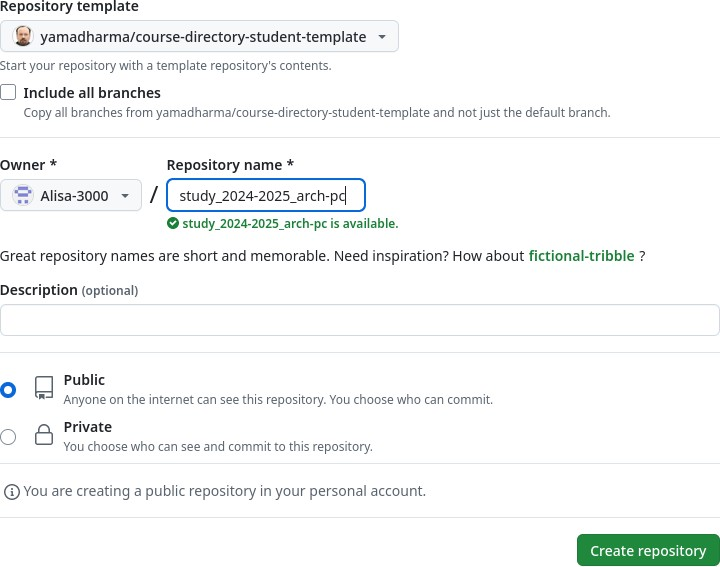


Рис. 17: Окно создания репозиторий

Репозиторий создан (рис. 18)

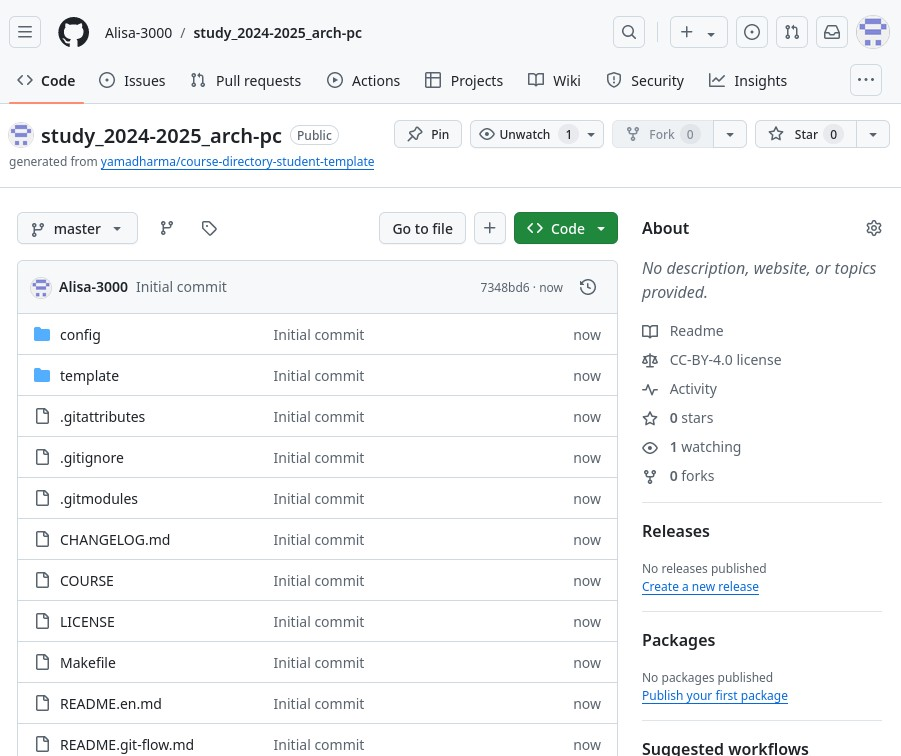


Рис. 18: Созданный репозиторий

Через терминал перехожу в созданный каталог курса с помощью утилиты cd (рис. 19)

Рис. 19: Перемещение между директориями

Рис. 19: Перемещение между директориями

Копирую ссылку для клонирования на странице созданного репозитория, сна-чала перейдя в окно «code», далее выбрав в окне вкладку «SSH» (рис. 20)

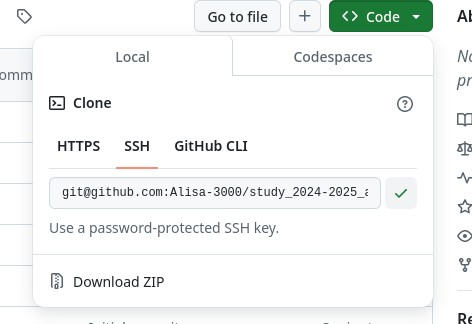


Рис. 20: Окно с ссылкой для копирования репозитория

Клонирую созданный репозиторий с помощью команды git clone – recursivegit@github.com:/study\_2024–2025\_arh-pc.git arch-pc (рис. 21)

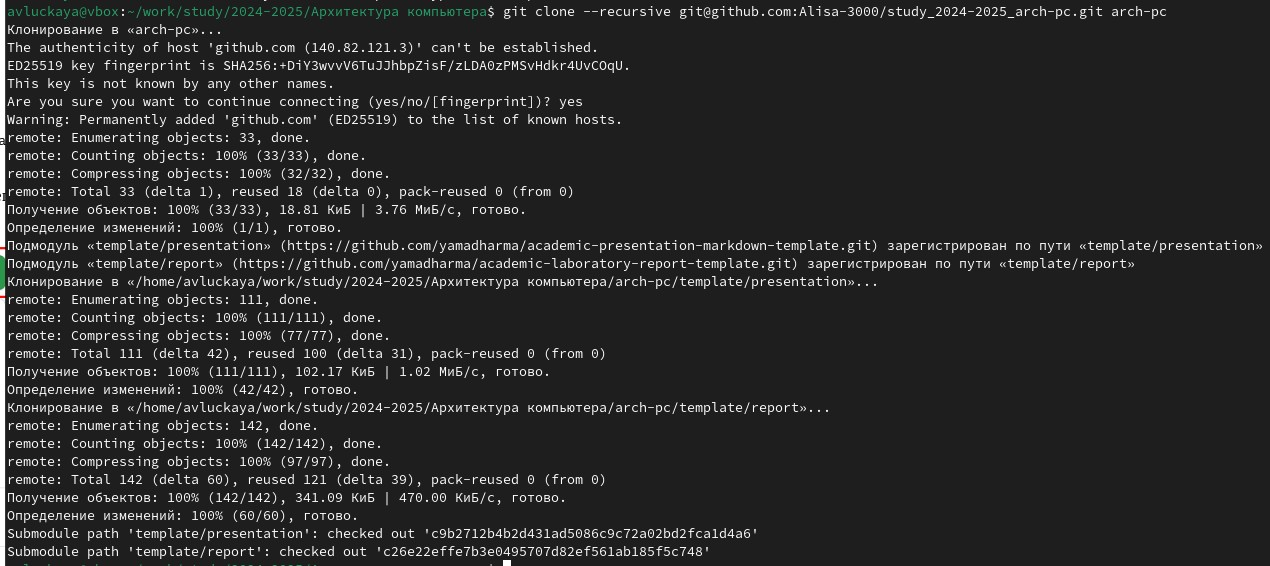


Рис. 21: Клонирование репозитория

## 4.5 Настройка каталога курса

Перехожу в каталог arch-pc с помощью утилиты cd (рис. 22)

Рис. 22: Перемещение между директориями

Рис. 22: Перемещение между директориями

Удаляю лишние файлы с помощью утилиты rm (рис. 23)

Рис. 23: Удаление файлов

Рис. 23: Удаление файлов

Создаю необходимые каталоги и устанавливаю пакет make, тк до этого он был не установлен (рис. 24)

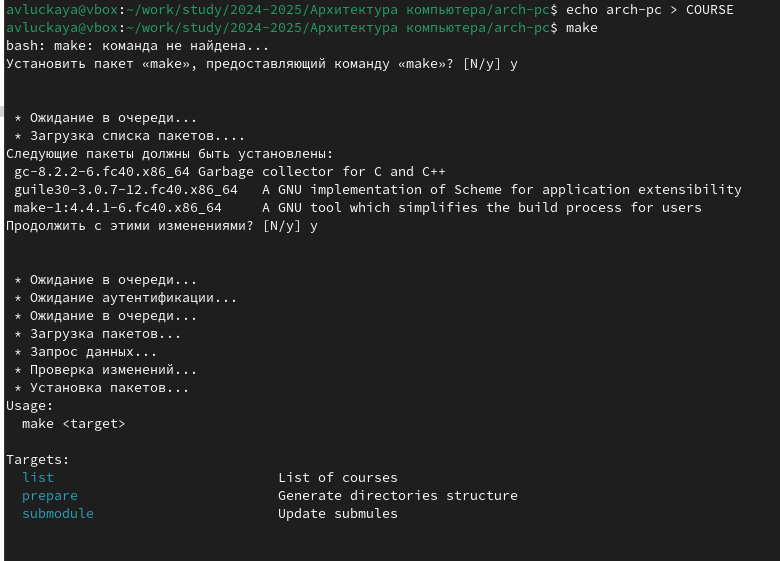


Рис. 24: Создание каталов

Добавляем все изменения с помощью команды git add . Комментируем и сохраняемвсе изменения с сообщением о создании структуры курса (рис. 25)

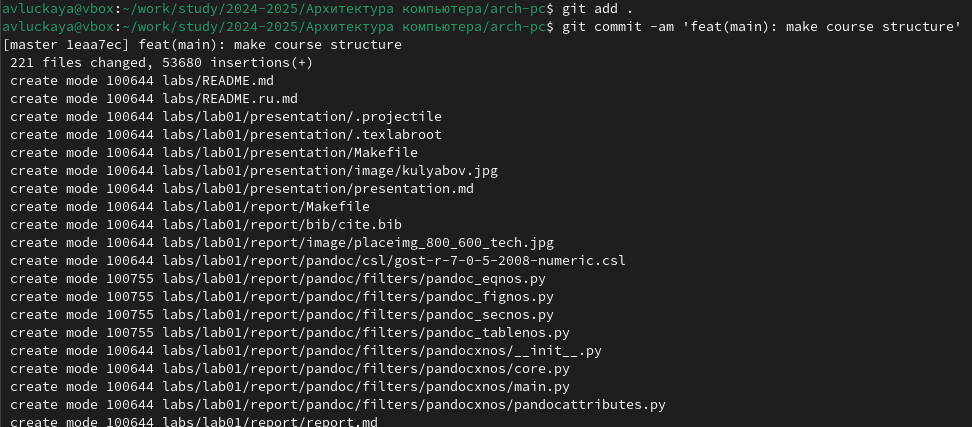


Рис. 25: Добавление файла на сервер

Отправляем все на сервер с помощью push (рис. 26)

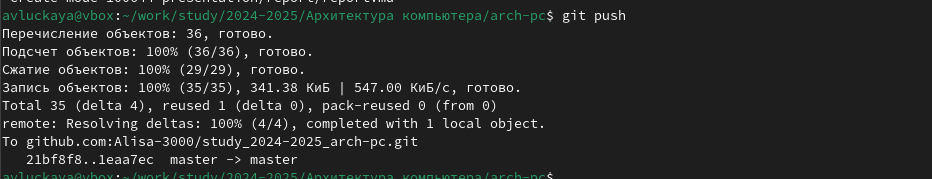


Рис. 26: Отправка в центральный репозиторий

Проверяем на github (рис. 27)

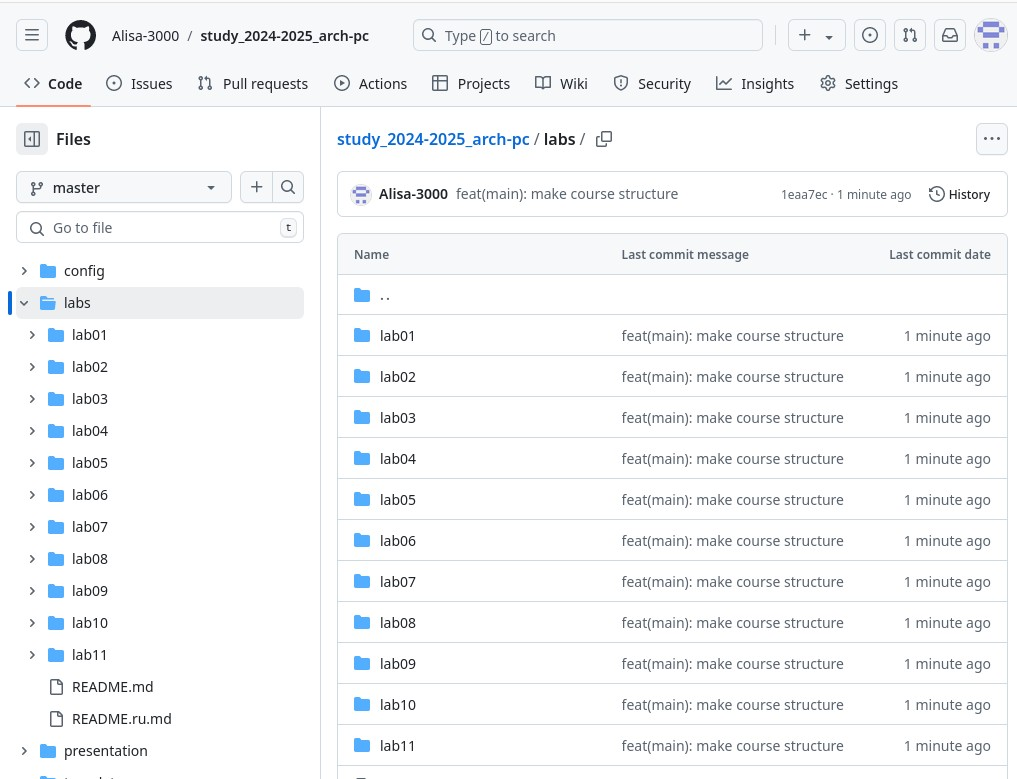


Рис. 27: Проверка файлов

## 4.6 Выполнение заданий для самостоятельной работы

Переходим в каталог с отчетами по лабораторной работе 2 (рис. 28)

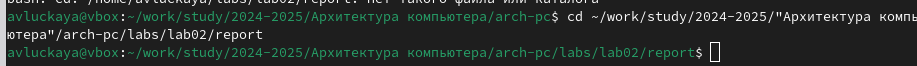


Рис. 28: Перемещение между директориями

Создаем пустой файл с именем отчет.odt в текущем каталоге (рис. 29)

Рис. 29: Создание отчета

Рис. 29: Создание отчета

Открываем файл в текстовом редакторе LibreOffice для редактирования (рис. 30)

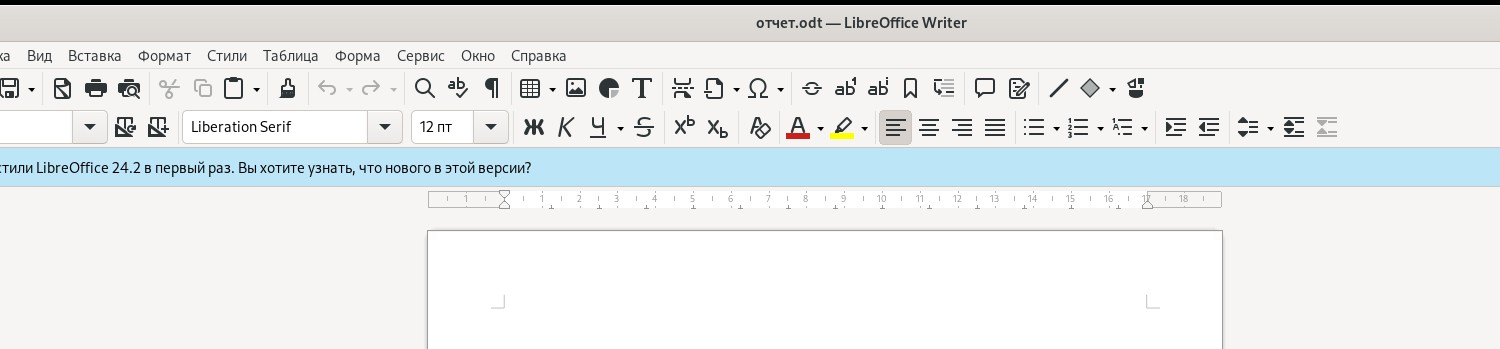


Рис. 30: LibreOffice

Комментируем изменения в репозитории с сообщением «Добавлен отчет по лабораторной работе №2» (рис. 31)

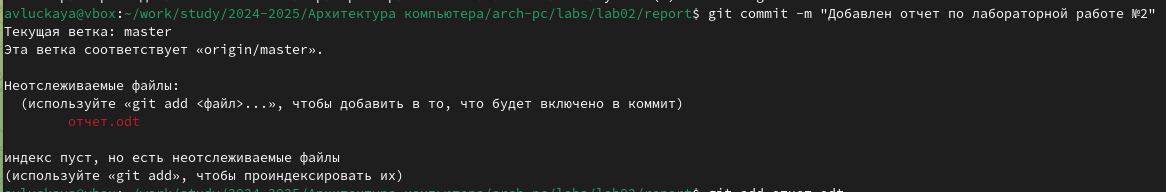


Рис. 31: Комментарий

Добавляем файл отчета (рис. 32)

Рис. 32: Добавление файла

Рис. 32: Добавление файла

Отправляем все изменения в удаленный репозиторий на ветку master (рис. 33)

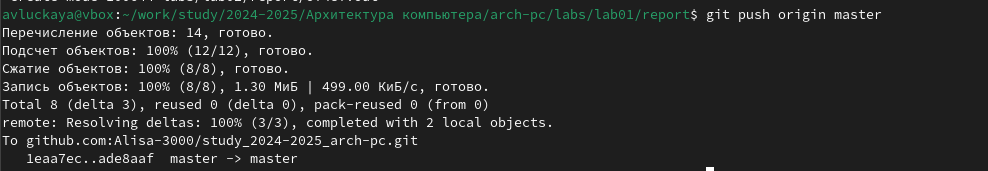


Рис. 33: Отправление файлов в центральный репозиторий

С помощью команды cp копируем файл Л01\_Луцкая\_отчет.pdf из папки Загрузки в каталог /work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report/ , с помощью команды cd переходим в каталог с отчетами по лабораторной работе 1, с помощью -ls выводим список файлов в текущем каталоге (рис. 34)

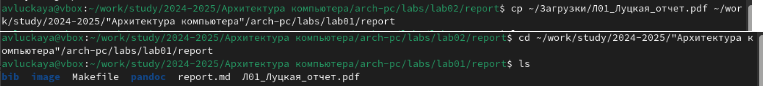


Рис. 34: Копирование и проверка файлов

Добавляем файл Л01\_Луцкая\_отчет.pdf с помощью git add и добавляем комментарий с помощьюgit commit (рис. 35)

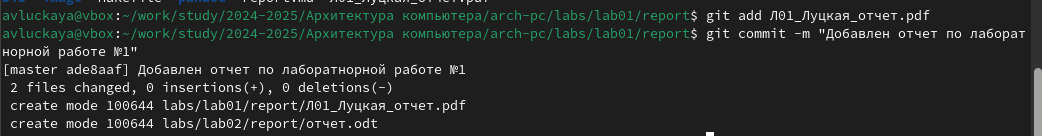


Рис. 35: Добавление файла

Отправляем все изменения в репозиторий (рис. 36)

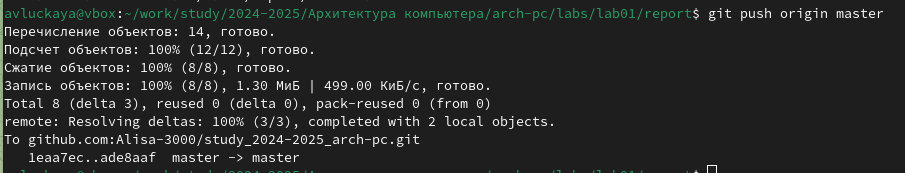


Рис. 36: Отправление изменений

Проверяем файлы в репозитории (рис. 37)

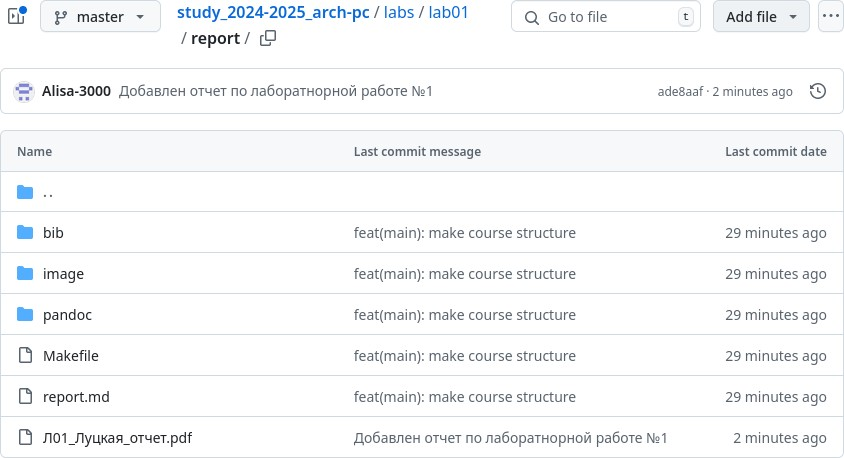


Рис. 37: Файлы в репозитории

# 5 Выводы

В ходе выполнения лабораторной работы была изучена концепция и применение систем контроля версий, а также приобретены практические навыки работы с системой git.

# 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)