Отчёт по лабораторной работе №2

Дисциплина: архитектура компьютера

Сибомана Ламек

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 Задание

1. Техническое обеспечение
   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.
2. Выполнение заданий для самостоятельной работы.

# 3 Выполнение лабораторной работы

## 3.1 Настройка GitHub

Для начала создадим учётную запись на сайте https://github.com/ и заполните основные данные (рис. 1)

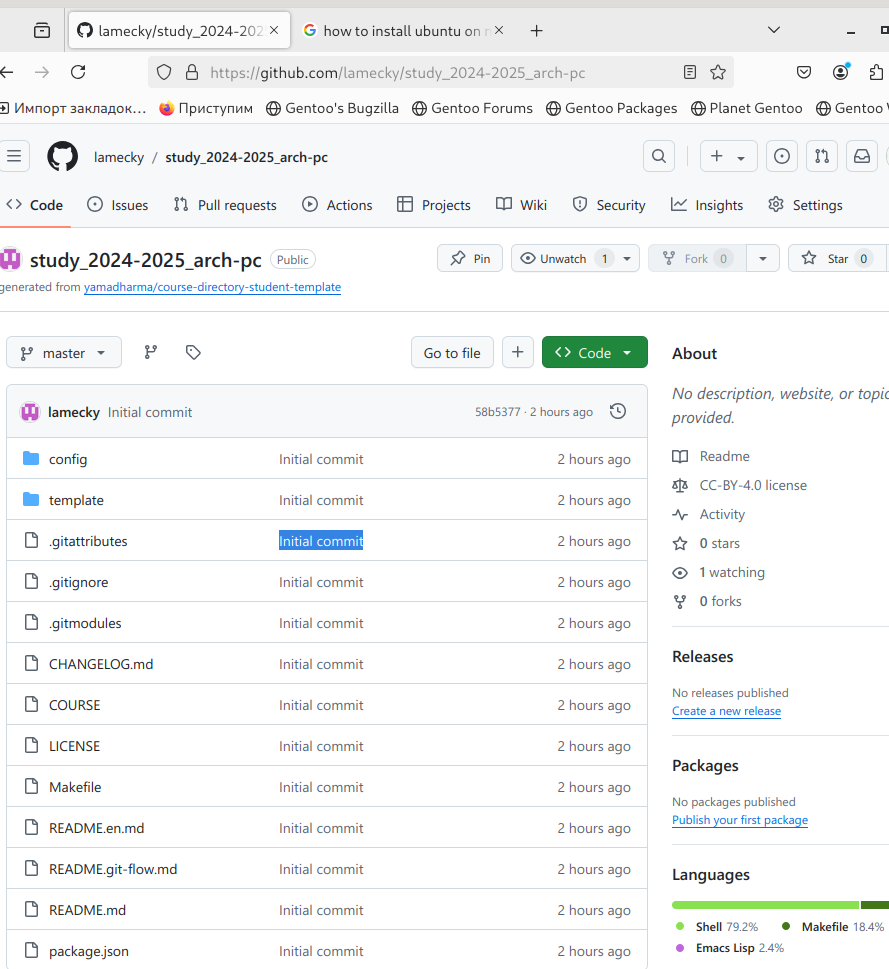


Рис. 1: Учётная запись на сайте https://github.com/

Аккаунт создан (рис. 2).

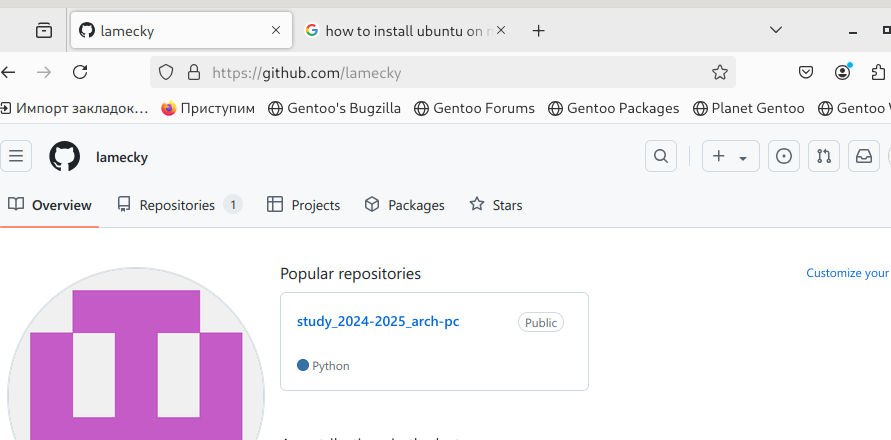


Рис. 2: Аккаунт GitHub

## 3.2 Базовая настройка Git

Открываю виртуальную машину, затем открываю терминал и делаю предварительную конфигурацию git. Ввожу команду git config –global user.name “”, указывая свое имя и команду git config –global user.email “[work@mail](mailto:work@mail)”, указывая в ней электронную почту владельца, то есть мою (рис. 3).

Рис. 3: Предварительная конфигурация git

Рис. 3: Предварительная конфигурация git

Настраиваю utf-8 в выводе сообщений git для корректного отображения символов (рис. 4).

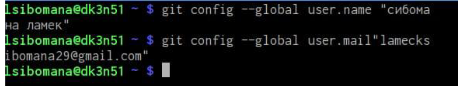


Рис. 4: Настройка кодировки

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

Рис. 5: Создание имени для начальной ветки

Рис. 5: Создание имени для начальной ветки

Задаю параметр autocrlf со значением input, так как я работаю в системе Linux, чтобы конвертировать CRLF в LF только при коммитах (рис. 6). CR и LF – это символы, которые можно использовать для обозначения разрыва строки в текстовых файлах.

Рис. 6: Параметр autocrlf

Рис. 6: Параметр autocrlf

Задаю параметр safecrlf со значением warn, так Git будет проверять преобразование на обратимость (рис. 7). При значении warn Git только выведет предупреждение, но будет принимать необратимые конвертации.

Рис. 7: Параметр safecrlf

Рис. 7: Параметр safecrlf

## 3.3 Создание SSH-ключа

Для последующей идентификации пользователя на сервере репозиториев необходимо сгенерировать пару ключей (приватный и открытый). Для этого ввожу команду ssh-keygen -C “Имя Фамилия, [work@email](mailto:work@email)”, указывая имя владельца и электронную почту владельца (рис. 8). Ключ автоматически сохранится в каталоге ~/.ssh/.

Рис. 8: Генерация SSH-ключа

Рис. 8: Генерация SSH-ключа

Xclip – утилита, позволяющая скопировать любой текст через терминал. Оказывается, в дистрибутиве Linux Kali ее сначала надо установить. Устанавливаю xclip с помощью команды apt-get install с ключом -y отимени суперпользователя, введя в начале команды sudo (рис. 9).

Рис. 9: Установка утилиты xclip

Рис. 9: Установка утилиты xclip

Копирую открытый ключ из директории, в которой он был сохранен, с помощью утилиты xclip (рис. 10).

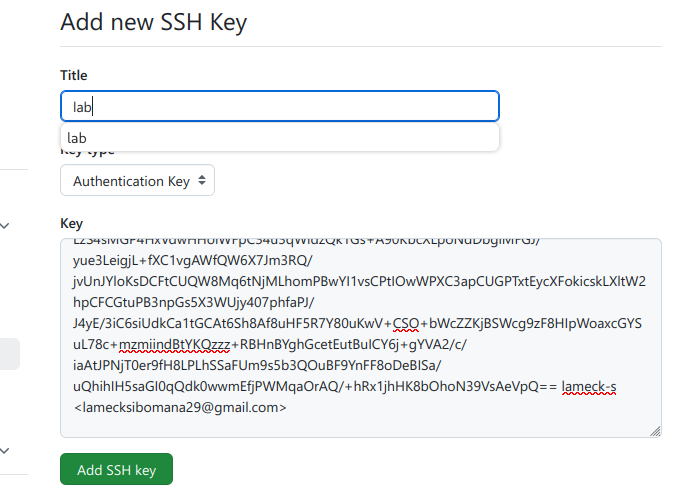


Рис. 10: Копирование содержимого файла

Открываю браузер, захожу на сайт GitHub. Открываю свой профиль и выбираю страницу «SSH and GPG keys». Нажимаю кнопку «New SSH key» (рис. 11).

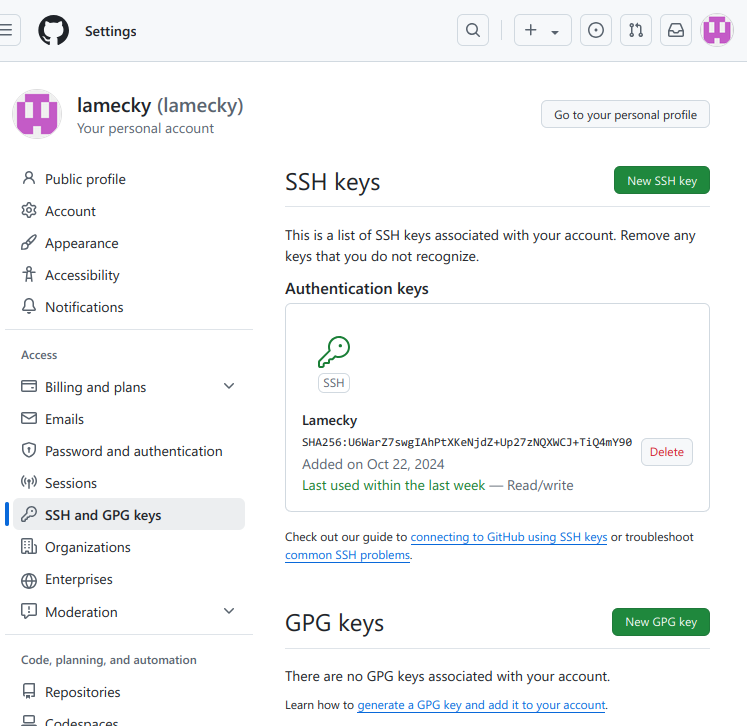


Рис. 11: Окно SSH and GPG keys

Вставляю скопированный ключ в поле «Key». В поле Title указываю имя для ключа. Нажимаю «Add SSH-key», чтобы завершить добавление ключа (рис. 12).

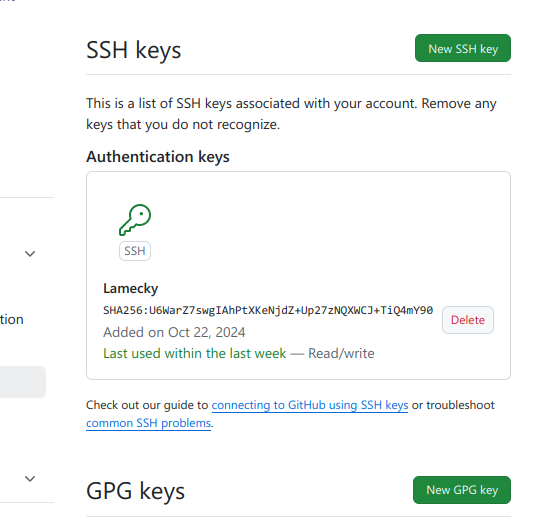


Рис. 12: Добавление ключа

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

Закрываю браузер, открываю терминал. Создаю директорию, рабочее пространство, с помощью утилиты mkdir, блягодаря ключу -p создаю все директории после домашней ~/work/study/2022-2023/“Архитектура компьютера” рекурсивно. Далее проверяю с помощью ls, действительно ли были созданы необходимые мне каталоги (рис. 13).

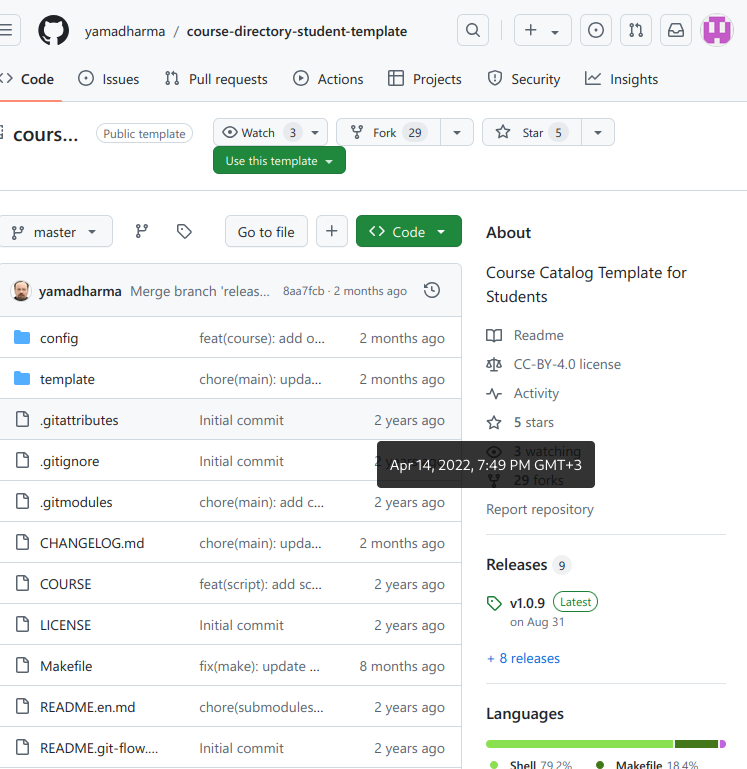


Рис. 13: Создание рабочего пространства

## 3.5 Создание репозитория курса на основе шаблона

В браузере перехожу на страницу репозитория с шаблоном курса по адресу https://github.com/yamadharma/course-directory-student-template. Далее выбираю «Use this template», чтобы использовать этот шаблон для своего репозитория (рис. 14).

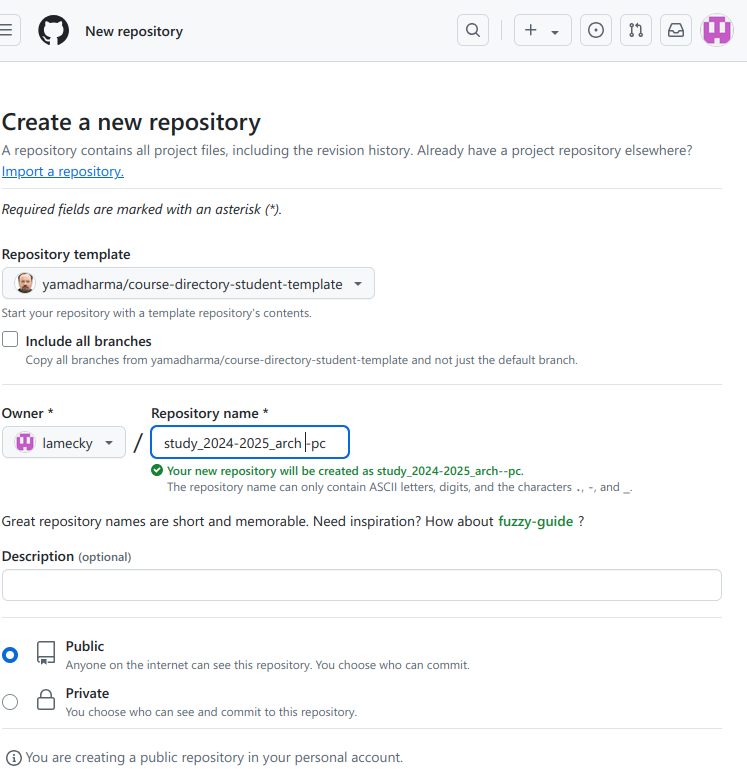


Рис. 14: Страница шаблона для репозитория

В открывшемся окне задаю имя репозитория (Repository name): study\_2022–2023\_arh-pc и создаю репозиторий, нажимаю на кнопку «Create repository from template» (рис. 15).

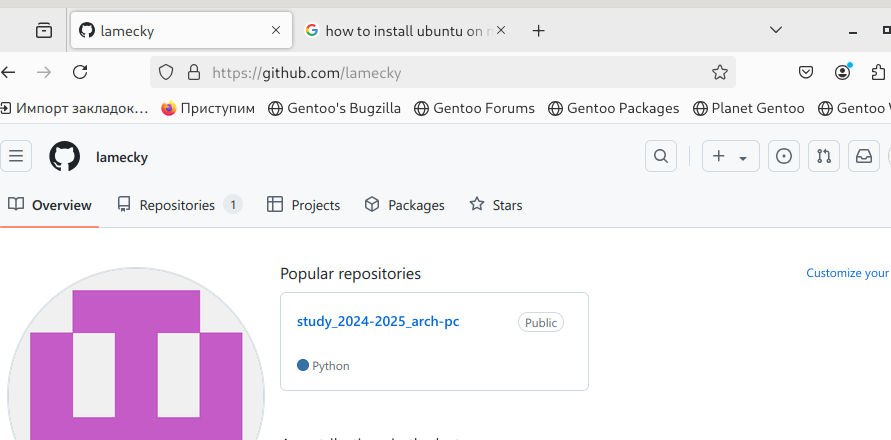


Рис. 15: Окно создания репозитория

Репозиторий создан (рис. 16).

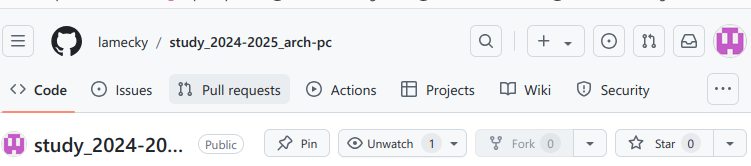


Рис. 16: Созданный репозиторий

Через терминал перехожу в созданный каталог курса с помощью утилиты cd (рис. 17).

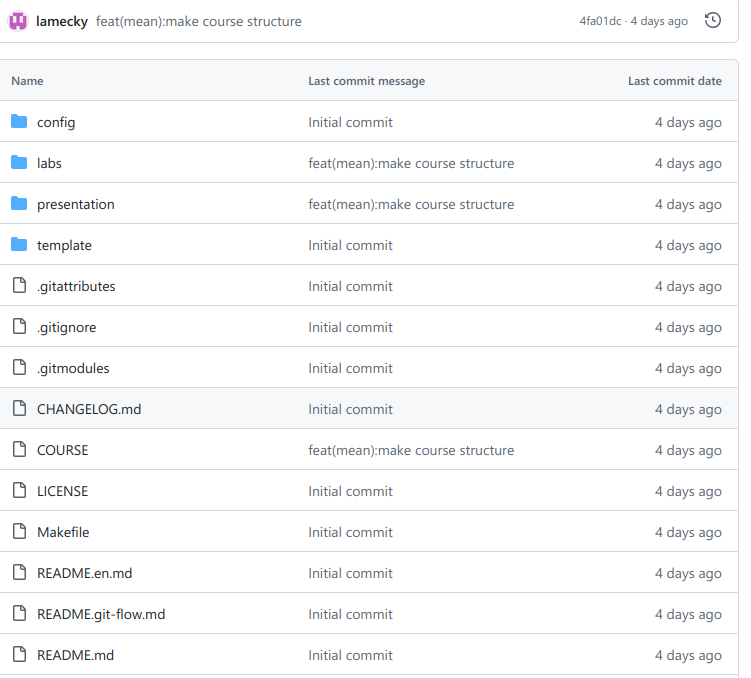


Рис. 17: Перемещение между директориями

Клонирую созданный репозиторий с помощью команды git clone –recursive git@github.com:/study\_2022–2023\_arh-pc.git arch-pc (рис. 18).

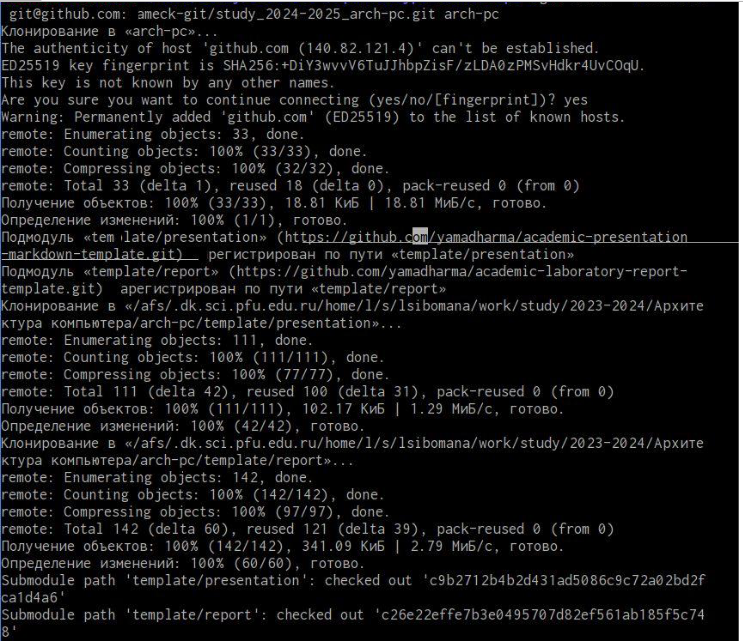


Рис. 18: Клонирование репозитория

Копирую ссылку для клонирования на странице созданного репозитория, сначала перейдя в окно «code», далее выбрав в окне вкладку «SSH» (рис. 19).

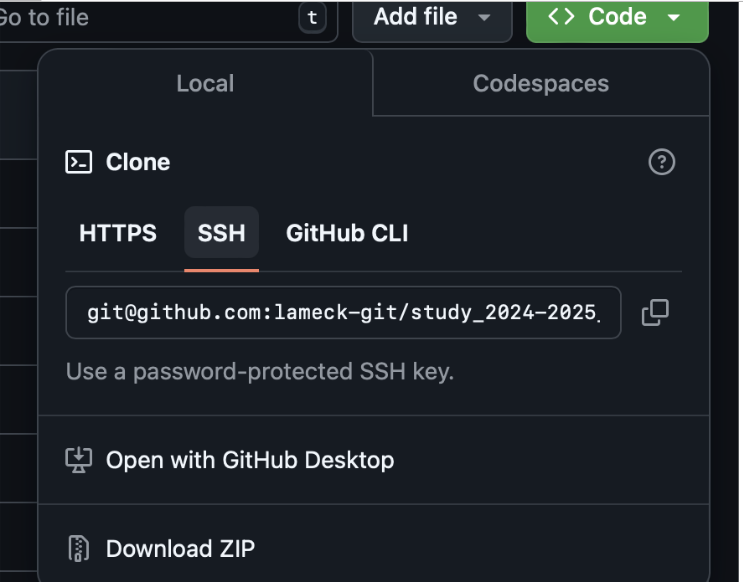


Рис. 19: Окно с ссылкой для копирования репозитория

## 3.6 Настройка каталога курса

Перехожу в каталог arch-pc с помощью утилиты cd (рис. 20).

Рис. 20: Перемещение между директориями

Рис. 20: Перемещение между директориями

Удаляю лишние файлы с помощью утилиты rm (рис. 21).

Рис. 21: Удаление файлов

Рис. 21: Удаление файлов

Создаю необходимые каталоги (рис. 22).

Рис. 22: Создание каталогов

Рис. 22: Создание каталогов

Отправляю созданные каталоги с локального репозитория на сервер: добавляю все созданные каталоги с помощью git add, комментирую и сохраняю изменения на сервере как добавление курса с помощью git commit (рис. 23).

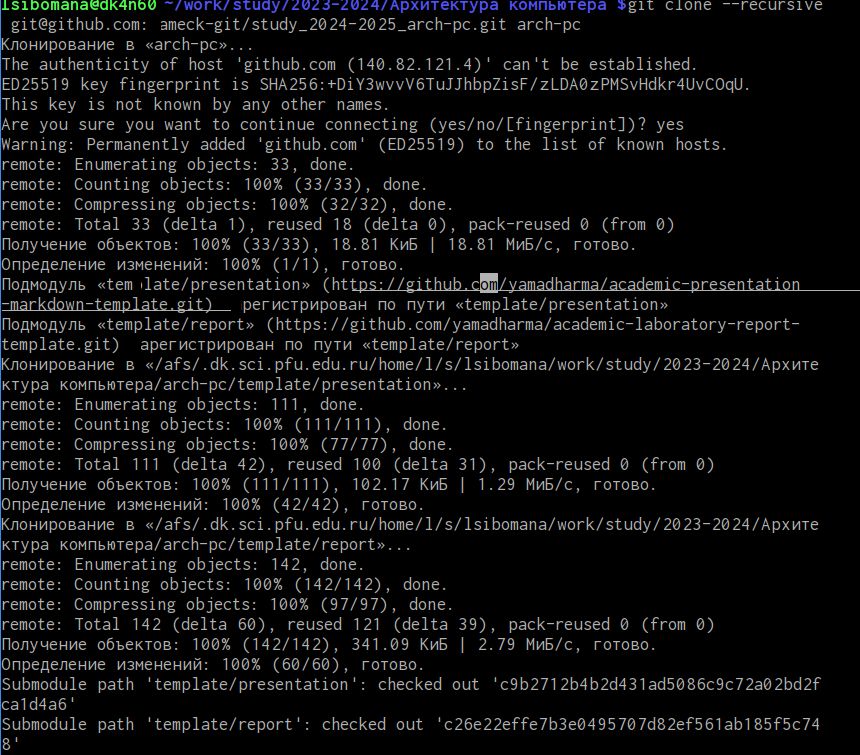


Рис. 23: Добавление и сохранение изменений на сервере

Отправляю все на сервер с помощью push (рис. 24).

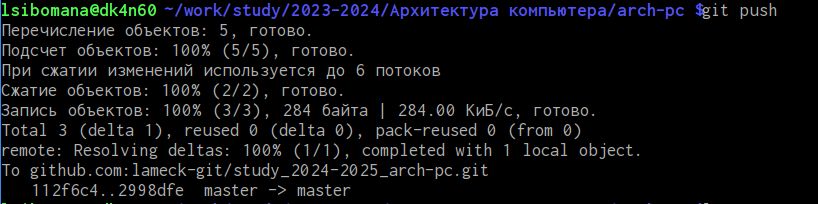


Рис. 24: Выгрузка изменений на сервер

Проверяю правильность выполнения работы сначала на самом сайте GitHub (рис. 25).

Рис. 25: Страница репозитория

Рис. 25: Страница репозитория

## 3.7 Выполнение заданий для самостоятельной работы

1. Перехожу в директорию labs/lab03/report с помощью утилиты cd. Создаю в каталоге файл для отчета по третьей лабораторной работе с помощью утилиты touch (рис. 26).

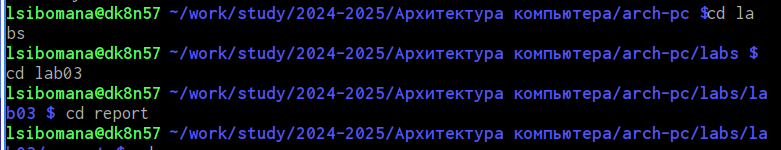
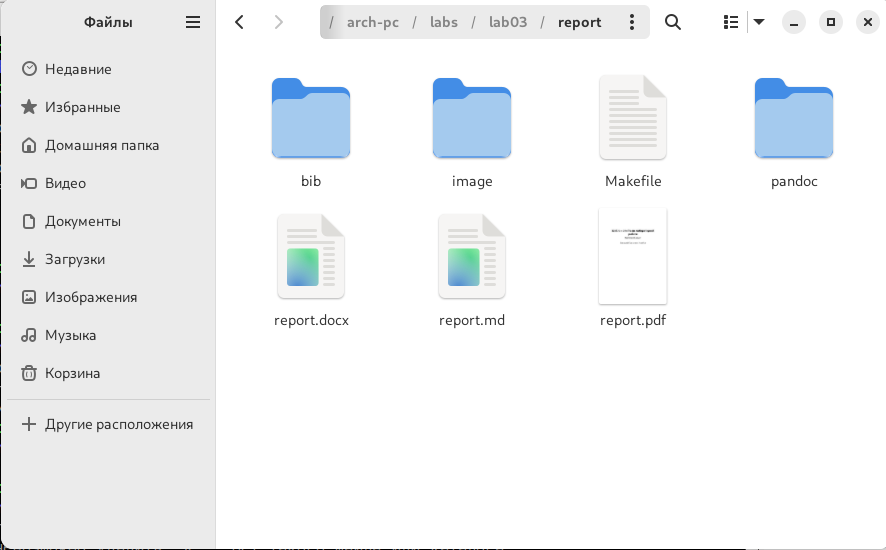
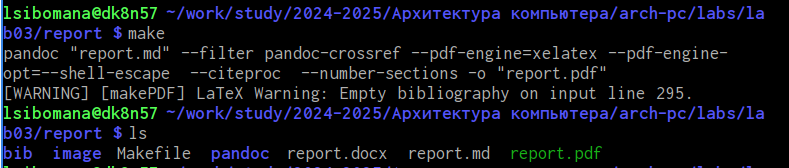


Рис. 26: Создание файла

Оформить отчет я смогу в текстовом процессоре LibreOffice Writer, найдя его в меню приложений (рис. **¿fig:028?**).

+

После открытия текстового процессора открываю в нем созданный файл и могу начать в нем работу над отчетом. 2. Перехожу из подкаталога lab03/report в подкаталог lab01/report с помощью утилиты cd (рис. **¿fig:031?**). 

Проверяю местонахождение файлов с отчетами по первой и второй лабораторным работам. Они должны быть в подкаталоге домашней директории «Загрузки», для проверки использую команду ls (рис. 27).

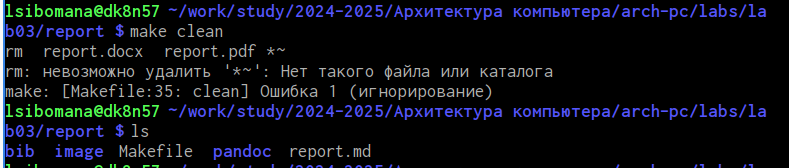


Рис. 27: Проверка местонахождения файлов

Копирую первую лабораторную с помощью утилиты cp и проверяю правильность выполнения команды cp с помощью ls (рис. 28).

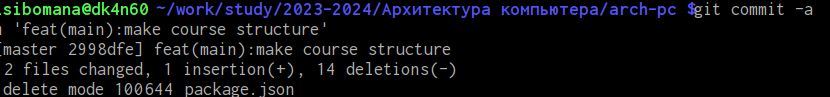


Рис. 28: Копирование файла

Перехожу из подкаталога lab01/report в подкаталог lab02/report с помощью утилиты cd (рис. 29).

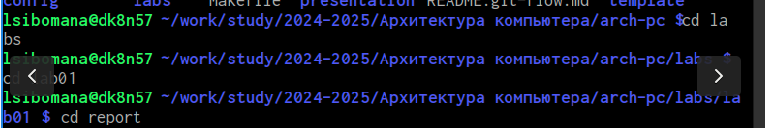


Рис. 29: Перемещение между директориями

Копирую вторую лабораторную с помощью утилиты cp и проверяю правильность выполнения команды cp с помощью ls (рис. 30).

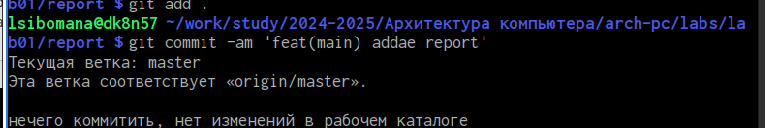


Рис. 30: Копирование файла

1. Добавляю с помощью команды git add в коммит созданные файлы: Л02\_Дворкина отчет (рис. 31). и

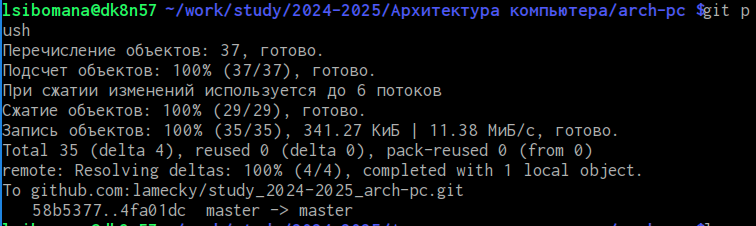


Рис. 31: Добавление файла на сервер

Вижу, что отчеты по лабораторным работам находятся в соответствующих каталогах репозитория: отчет по первой - в lab01/report (рис. 32), по второй – в lab02/report (рис. 33), по третьей в - lab03/report (рис. 34).

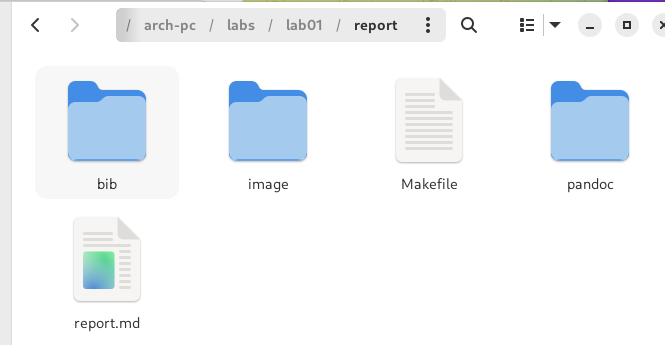


Рис. 32: Каталог lab01/report

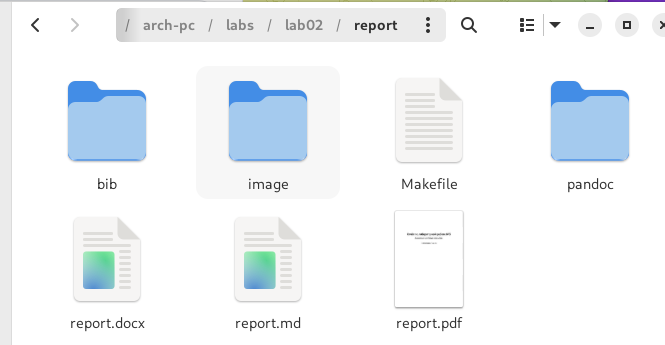


Рис. 33: Каталог lab02/report

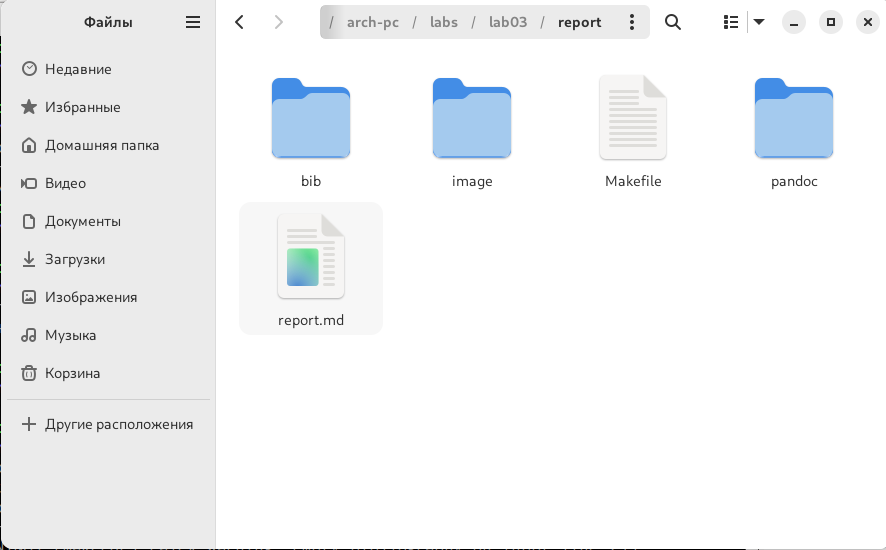


Рис. 34: Каталог lab03/report

# 4 Выводы

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