Лабораторная работа №2

Система контроля версий Git

Яковлева Дарья Сергеевна

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

Сначала сделаем предварительную конфигурацию git. Откроем терминал и введём следующие команды, указав имя и email владельца репозитория. (рис. 1)

Рисунок 1. Настройка имени и эл. почты

Рисунок 1. Настройка имени и эл. почты

Настроим utf-8 в выводе сообщений git. (рис. 2)

Рисунок 2. Настройка UTF8 в выводе сообщений Git

Рисунок 2. Настройка UTF8 в выводе сообщений Git

Зададим имя начальной ветки (будем называть её master). (рис. 3)

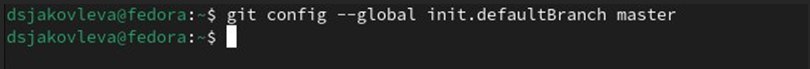


Рисунок 3. Конфигурация имени начальной ветки

Настроим параметр autocrlf и параметр safecrlf. (рис. 4 и рис. 5)

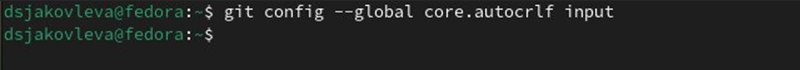


Рисунок 4. Настройка параметра autocrlf

Рисунок 5. Настройка параметра safecrlf

Рисунок 5. Настройка параметра safecrlf

Для того, чтобы сервер мог идентифицировать пользователя, необходимо сгенерировать несколько ssh ключей. Начнем с генерации открытого ключа. Для этого мы введем следующую команду, указав имя, фамилию пользователя и его адрес электронной почты в качестве аргумента. (рис. 6)

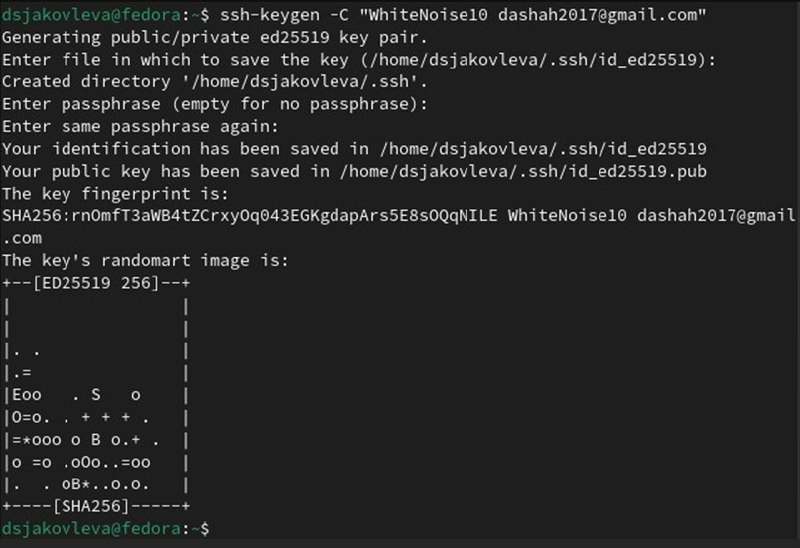


Рисунок 6. Генерация ssh ключа

Зайдем на сайт GitHub. (рис. 7)

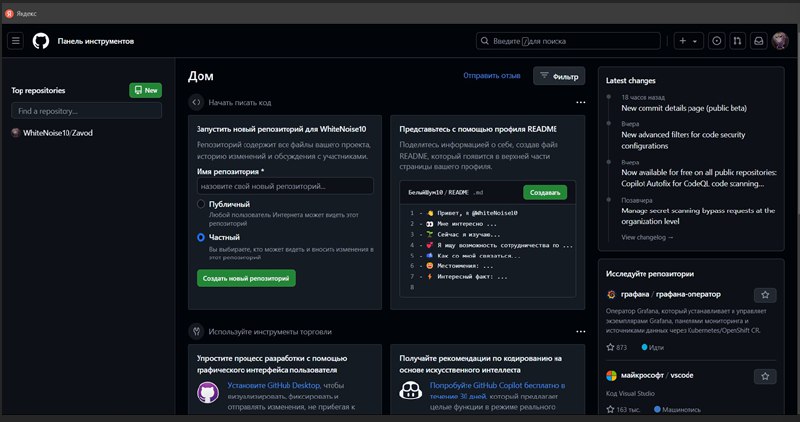


Рисунок 7. Внешний вит сайта GitHub

Перейдём в пункт settings. (рис. 8)

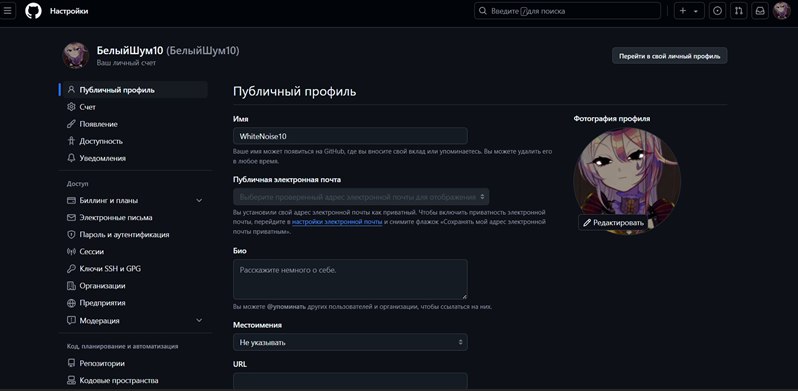


Рисунок 8. Вкладка настроек GitHub

Находим раздел SSH and GPG keys и нажимаем New SSH key. (рис. 9)

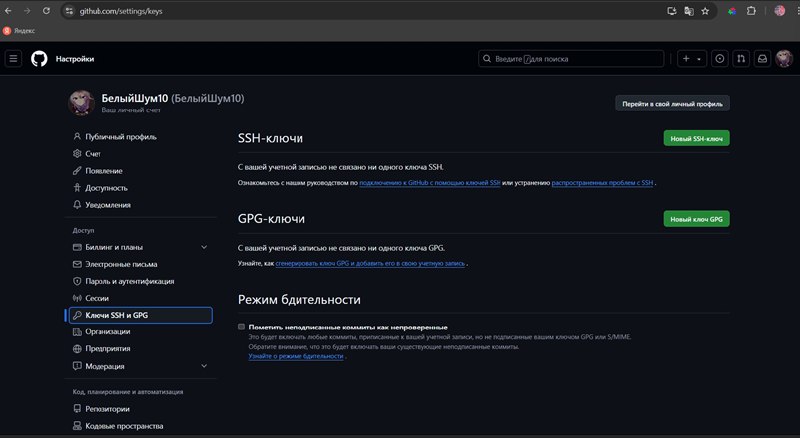


Рисунок 9. Раздел настроек “SSH and GPG keys”

В предложенное поле “Key” необходимо вставить ключ, который мы только что сгенерировали. (рис. 10)

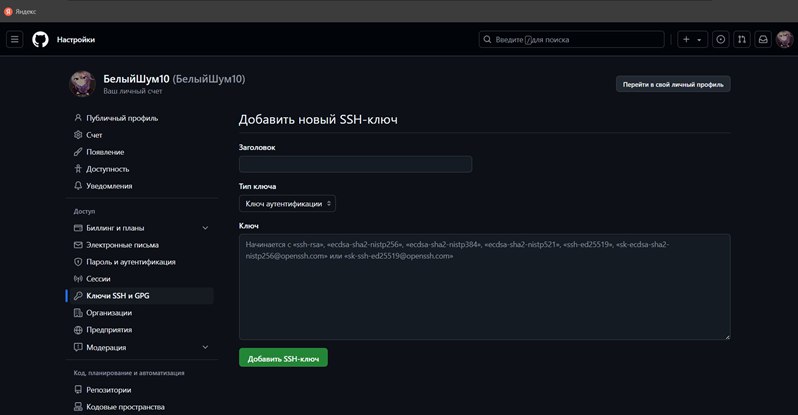


Рисунок 10. Поле ввода ключа

Для того, чтобы скопировать ключ для последующей вставки, нам введём следующую команду. (рис. 11)

Рисунок 11. Считывание и копирование публичного ключа

Рисунок 11. Считывание и копирование публичного ключа

Остается лишь вставить содержимое буфера обмена, то есть наш ключ, в предложенное поле “Key” и указать имя ключа в поле “Title”. (рис. 12)

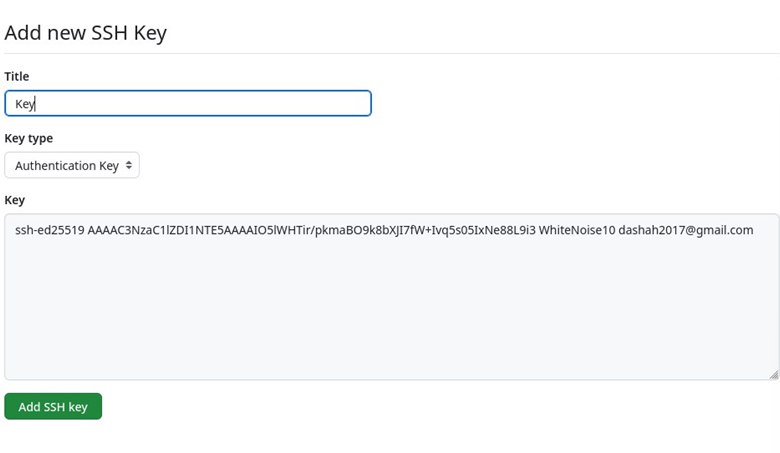


Рисунок 12. Заполнение необходимых полей для добавления ключа

Откроем терминал и создадим каталог для предмета «Архитектура компьютера». (рис. 13)

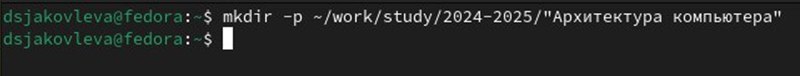


Рисунок 13. создание каталога «Архитектура компьютера»

Создадим репозиторий. Будем создавать его на основе шаблона, который находится по следующему адресу: https://github.com/yamadharma/course-directory-student-template (рис. 14)

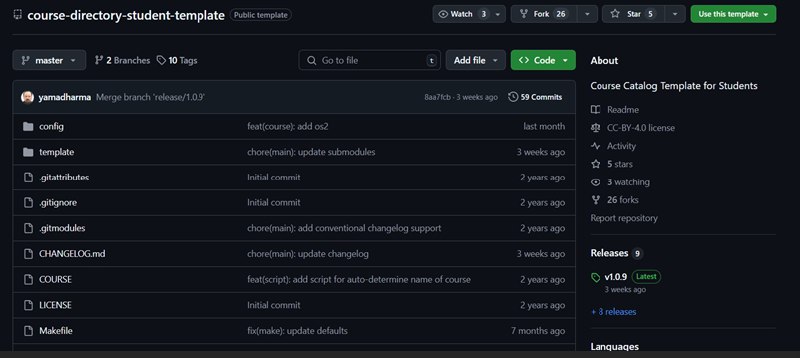


Рисунок 14. Страница шаблона на GitHub

Нажимаем на кнопку “Use this template”, и в предложенных опциях выбираем “Create new repository”. (рис. 15)

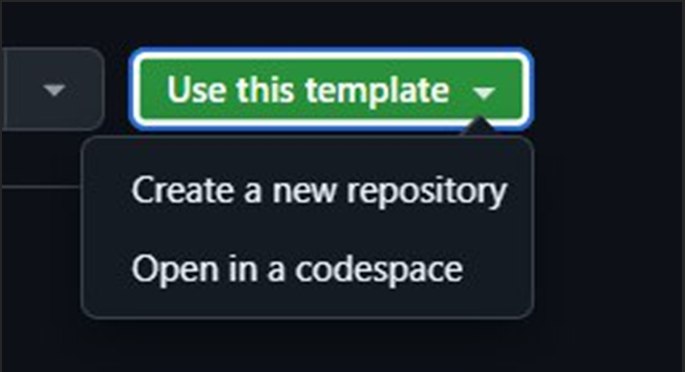


Рисунок 15. Создание репозитория

Задаём имя репозитория. Он будет называться: study\_2024-2025\_arh-pc. Далее нам нужно клонировать репозиторий на наш компьютер. Для этого перейдем в папку, в которую мы хотим скопировать репозиторий. В нашем случае это ранее созданная папка. (рис. 16)

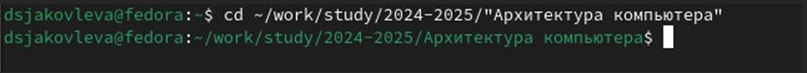


Рисунок 16. Перемещение в необходимую папку

Перейдем непосредственно к клонированию. Для этого воспользуемся командой git clone, в аргументе указав ссылку на репозиторий (рис. 17). Ссылку можно найти при нажатии на кнопку код на странице нашего репозитория (рис. 18)

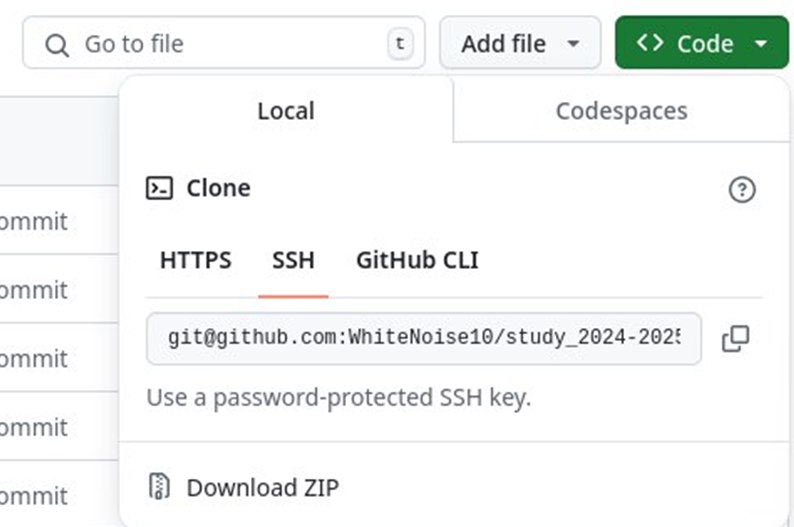


Рисунок 17. Ссылка на репозиторий

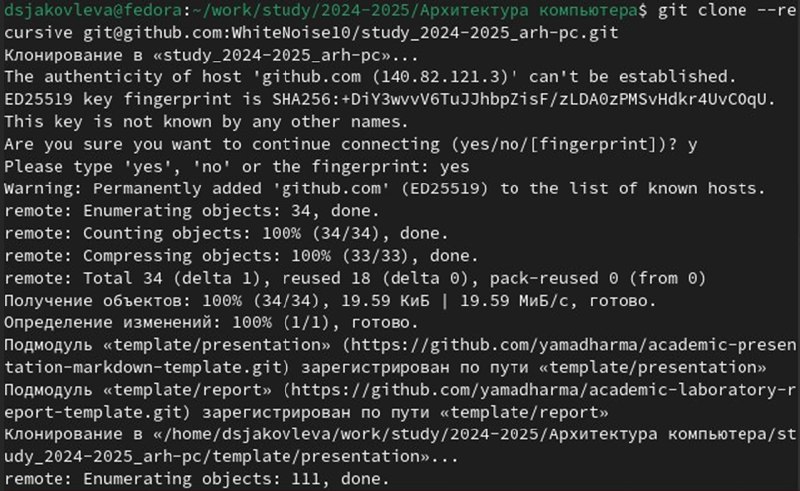


Рисунок 18. Использование команды “git clone”

Теперь перейдем к настройке клонированного каталога. Для начала перейдем в него с помощью команды “cd”. (рис. 19)

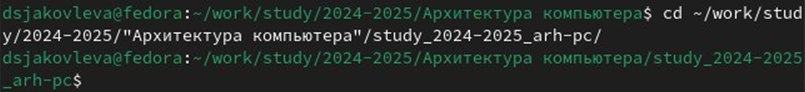


Рисунок 19. Переход в клонированный каталог

Удалим лишний файл с помощью команды “rm”. (рис. 20)

Рисунок 20. Удаление файла команды “rm”

Рисунок 20. Удаление файла команды “rm”

Создадим необходимые каталоги. (рис. 21)

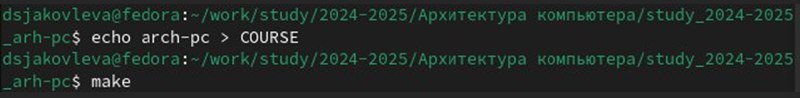


Рисунок 21. Создание необходимых каталогов

Теперь нам остается отправить файлы на сервер. Для этого с помощью команды git add мы добавим каталоги, которые должны отправляться на сервер. (рис. 22)

Рисунок 22. Добавление каталога для отправки на сервер

Рисунок 22. Добавление каталога для отправки на сервер

Теперь с помощью команды git commit мы сохраним изменения и укажем комментарий, в котором будет поясняться, какие изменения мы сделали. В данном случае в комментарии мы напишем, что создали структуру курса. (рис. 23)

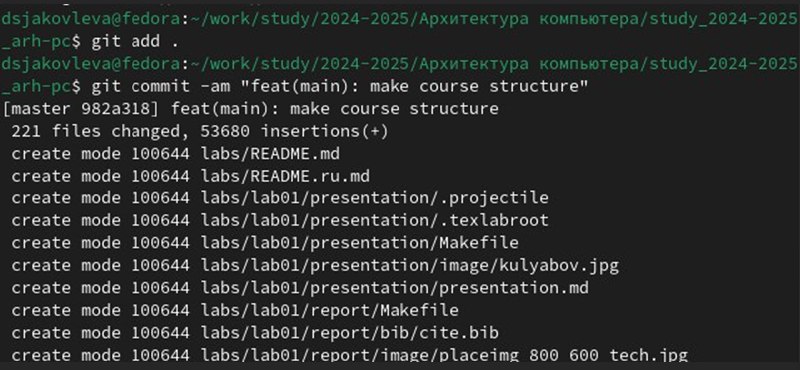


Рисунок 23. Сохранение изменений и комментарий

Теперь нам осталось окончательно загрузить изменения на сервер. Для этого мы воспользуемся командой git push. (рис. 24)

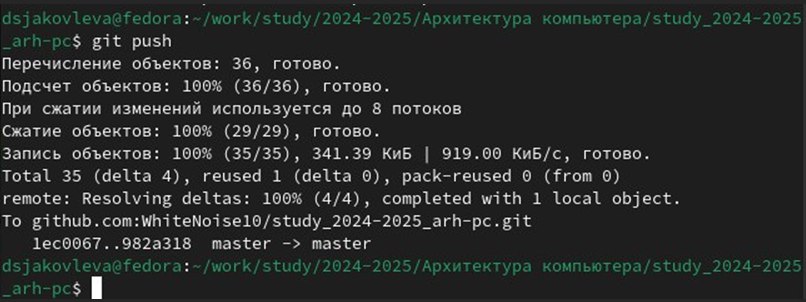


Рисунок 24. Отправка репозитория на GitHub

Проверим, сохранились ли файлы на сервере. Сравним файлы, которые находятся на GitHub с теми файлами, что находятся на нашем компьютере (рис. 25 и рис. 26)

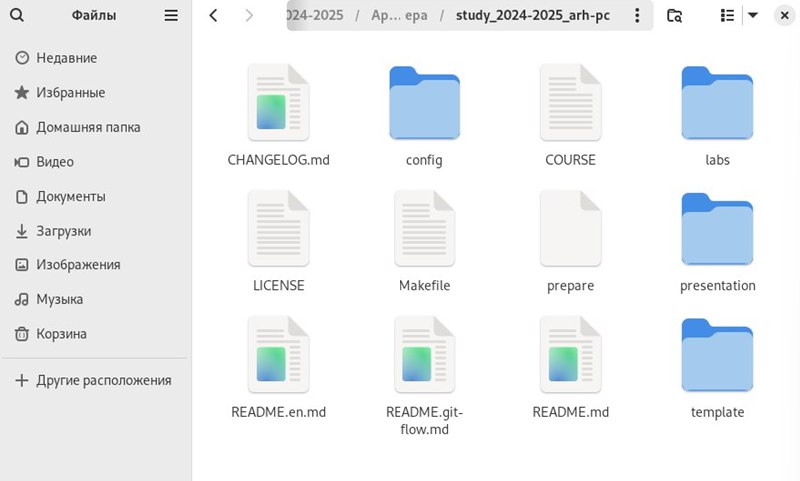


Рисунок 25. Файлы на компьютере

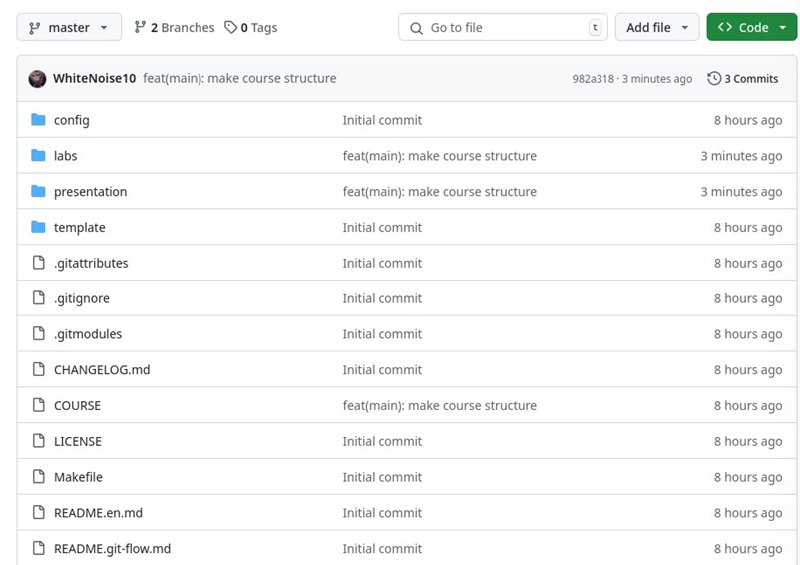


Рисунок 26. Файлы на сервере

Файлы совпали

# 3 Задание для самостоятельной работы

Теперь приступим к выполнению самостоятельной работы. Для начала мы создадим файл отчета для нашей лабораторной работы в папке labs/lab02/report с помощью LibreOffice (рис. 27, рис. 28 и рис. 29).

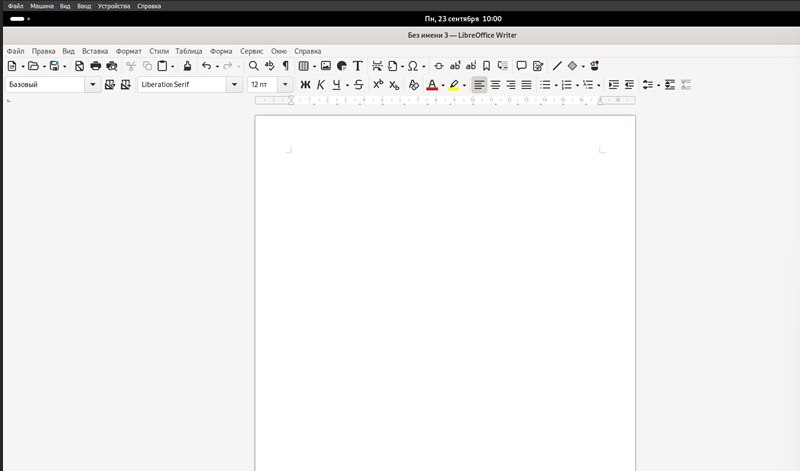


Рисунок 27. Начальный экран LibreOffice

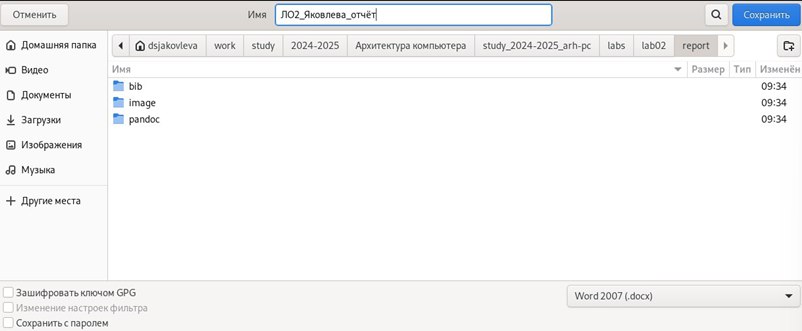


Рисунок 28. Выбор места для хранения файла в расширении docx

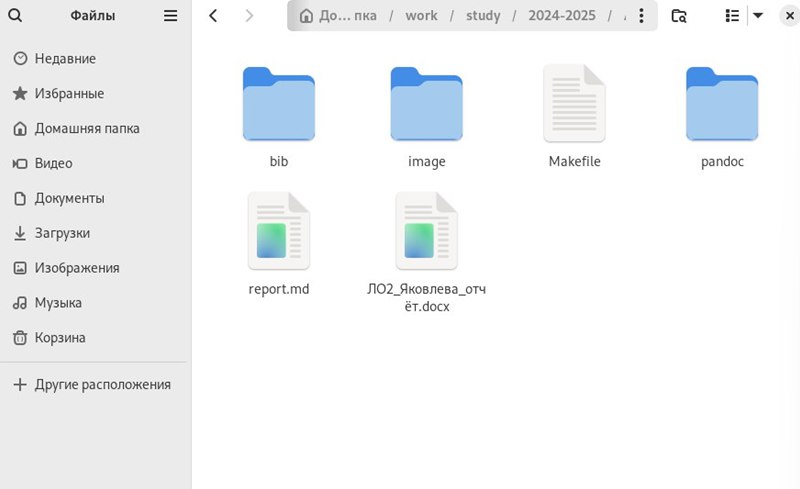


Рисунок 29. Проверка наличие отчёта в папке

После этого скопируем отчет по нашей предыдущей лабораторной работе в соответствующую папку созданного нами рабочего пространства, то есть в папку labs/lab01/report. Для копирования воспользуемся командой “cp” (рис. 30)

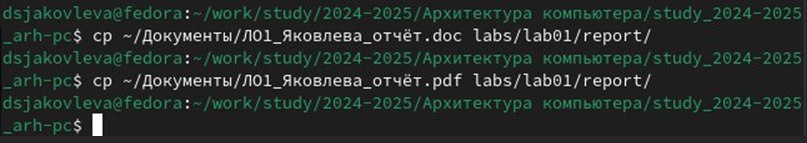


Рисунок 30. Копирование отчёта по первой лабораторной в форматах doc и pdf

Теперь нам осталось лишь загрузить изменения на GitHub. Воспользуемся командой “git add .” для того, чтобы указать, что мы хотим сохранить изменения во всех файлах, находящихся в нашем каталоге. После этого с помощью команды “git commit” мы укажем комментарий и сохраним изменения. (рис. 31)

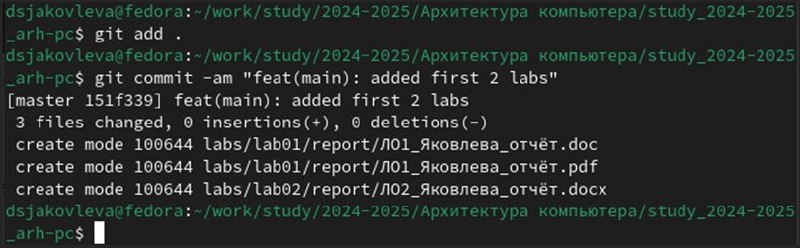


Рисунок 31. Добавление каталога для отправки на сервер, сохранение изменений и комментарий к ним

После этого введём команду “git push” для того, чтобы загрузить файлы на GitHub (рис. 32)

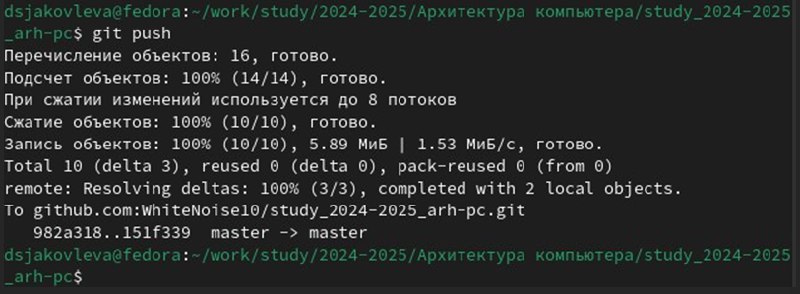


Рисунок 32. Загрузка файлов на GitHub

Остается лишь проверить, правильно ли мы все загрузили. Для этого посмотрим время обновления файлов в папке labs в GitHub. (рис. 33)

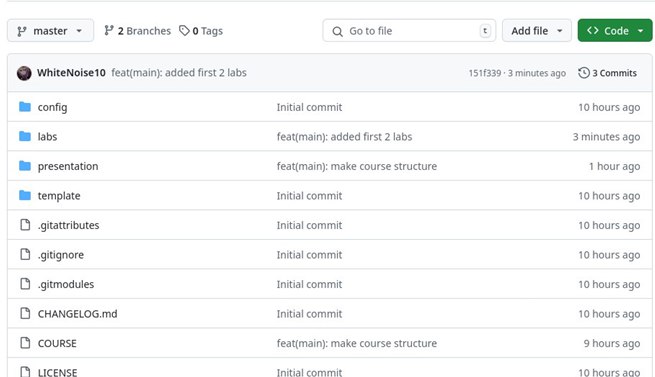


Рисунок 33. Обновлённый репозиторий на GitHub

# 4 Выводы

В результате выполнения лабораторной работы появились практические навыки работы с системой контроля версий Git, была произведена её первоначальная настройка в linux. Было изучено, как создавать репозитории, сохранять изменения и добавлять к ним комментарии, а также как выгружать файлы на сервер. Были приобретены навыки работы с платформой GitHub.