РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ ФАКУЛЬТЕТ ФИЗИКОМАТЕМАТИЧЕСКИХ И ЕСТЕСТВЕННЫХ НАУК

Отчёт по лабораторной работе №2

Приспешкин Андрей Андреевич НКАбд-05-23

Содержание

## 0.1 Содержание

- 1. Цель работы..................................................................  
 - 2. Задание.........................................................................  
 - 3. Выполнение лабораторной работы.....................................

* 1. Выводы……………………………………………………………….
  2. Список литературы…………………………………………………

**1) Цель работы**

Целью работы является изучить применение средств для контроля версий, а также приобрести практические навыки по работе с системой контроля версий git.

**2) Задания** 1) Настройка GitHub.

1. Базовая настройка git.
2. Создание SHH-ключа.
3. Создание рабочего пространства.
4. Создание репозитория на основе шаблона.
5. Настройка каталога курса.
6. Задания для самостоятельной работы.

**3) Выполнение лабораторной работы**

**1)** Создадим учётную запись на GitHub (рис.1).

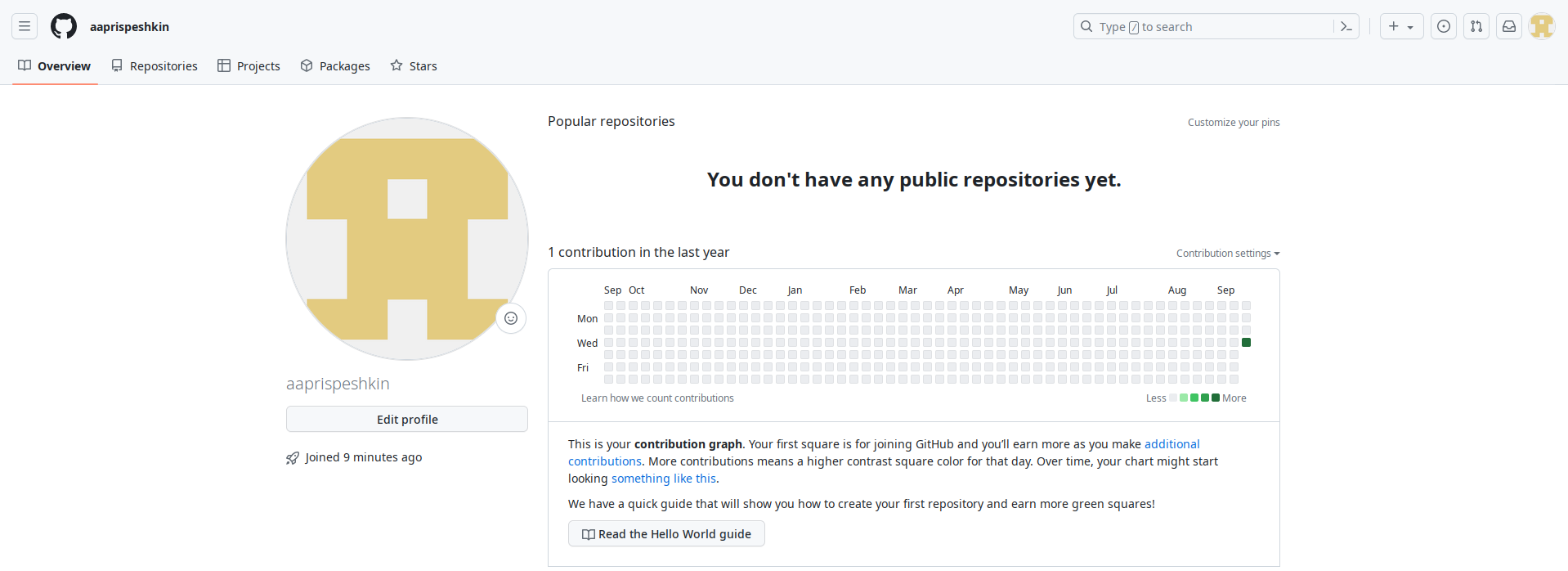


Рис.1 Созданный аккаунт на Github

**2)** Откроем терминал и введём команду git config, указав имя фамилию и электронную почту пользователя (рис.2).

Рис. 2 Настройка git с помошью команды git config

Рис. 2 Настройка git с помошью команды git config

Продолжая использовать команду git config, настраиваем кодировку utf-8 для вывода сообщений (рис.3) и создаём начальную ветку “master” (рис.4).

Рис.3 Настройка вывода сообщений в кодировке utf-8

Рис.3 Настройка вывода сообщений в кодировке utf-8

Рис.4 Создание начальной ветки

Рис.4 Создание начальной ветки

Зададим параметр autocrlf со значением input (рис.5).

Рис. 5 Задание параметра autocrlf

Рис. 5 Задание параметра autocrlf

Чтобы git мог определять преобразование на обратимость, зададим параметр autocrlf со значением warn(рис.6)

Рис.6 Задание параметра autocrlf со значением warn

Рис.6 Задание параметра autocrlf со значением warn

**3)** Для последующей работой с git и github, нам потребуется сгенерировать уникальный ssh ключ. Сделать это можно командой ssh-keygen(рис.7).

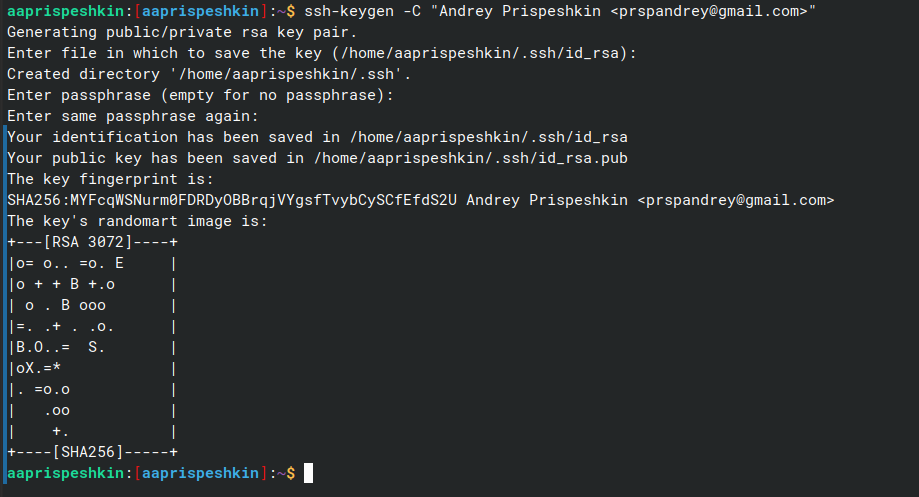


Рис. 7 Генерация ssh ключа с помощью команды ssh-keygen

Скопируем ключ из директории в которой он был сохранён, используя команды cat и xclip(рис.8)

Рис. 8 Использование команды xclup для копирования ключа из терминала в буфер обмена

Рис. 8 Использование команды xclup для копирования ключа из терминала в буфер обмена

Затем, загружаем скопированный ключ в GitHub. В настройках выберем пункт “SSH and GPG keys”(рис.9), там выберем “Add new SSH key” и вставим скопированный ранее ключ(рис.10).

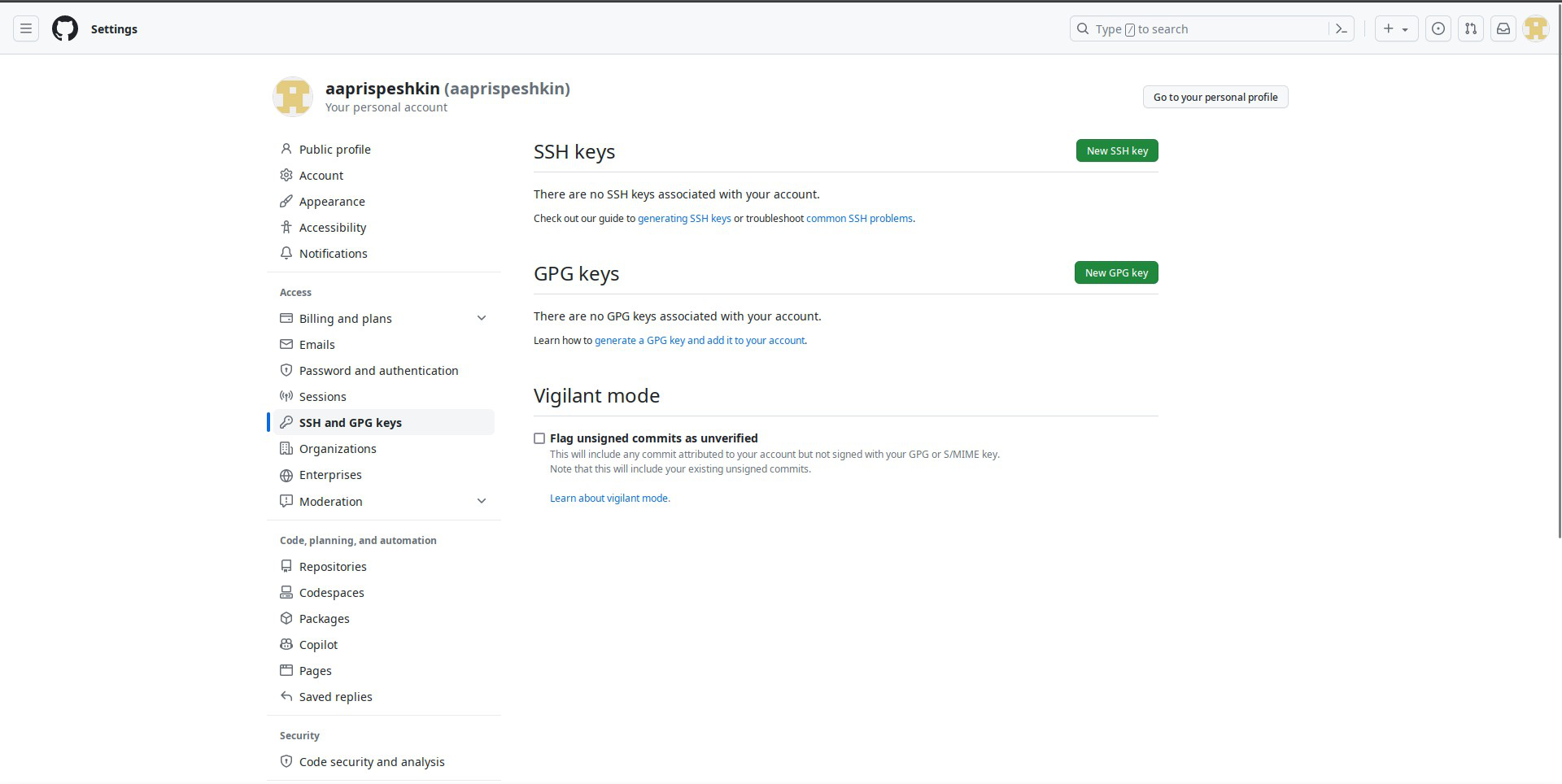


Рис. 9 Меню SSH and GPG keys в настройках

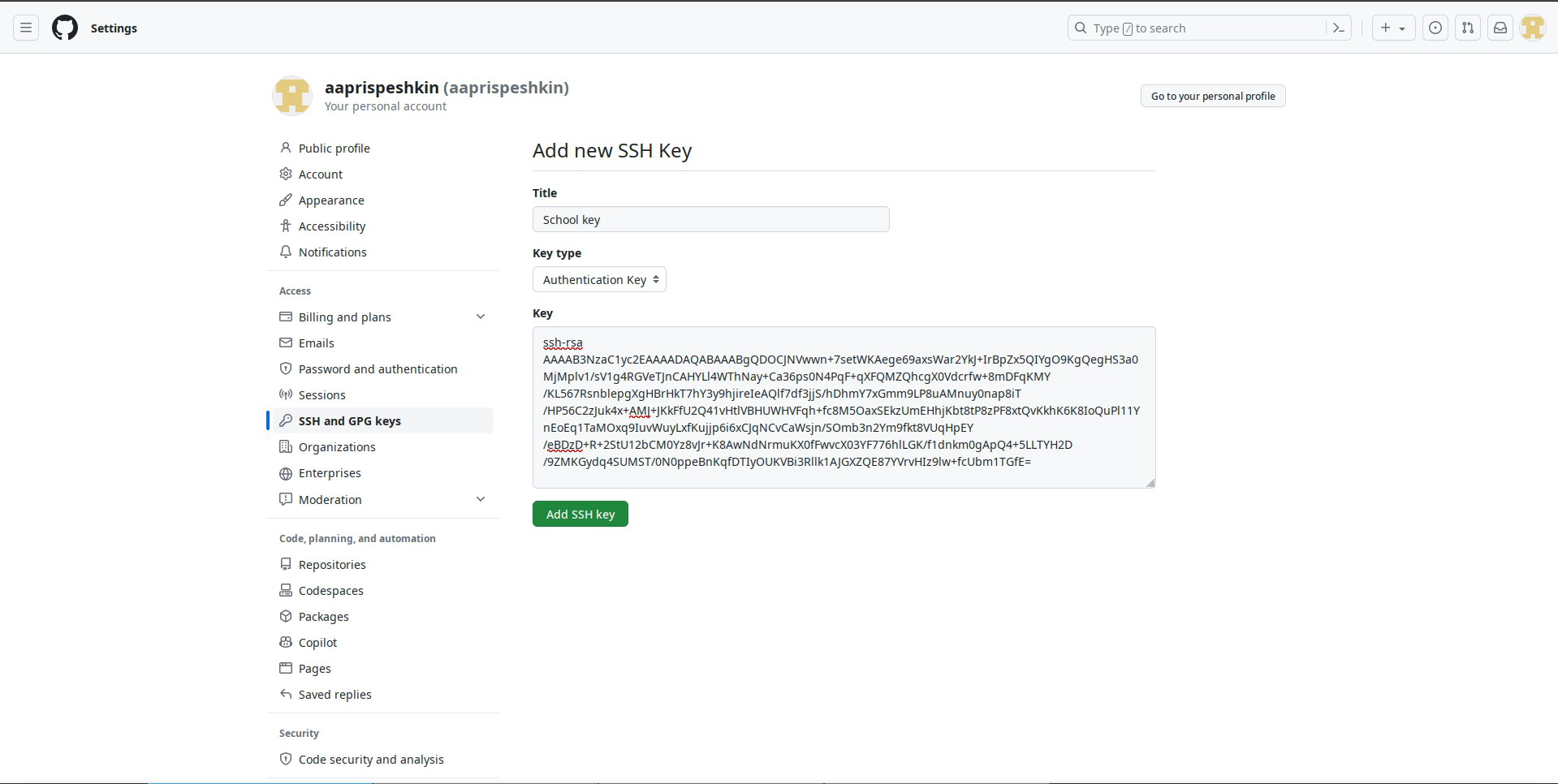


Рис.10 Вставленный SSH ключ сгенерированный ранее

**4)** С помощью утилиты mkdir с ключом -р создадим рабочую директорию(рис.11).

Рис. 11 Создание рабочей директории

Рис. 11 Создание рабочей директории

**5)** Перейдём на страницу с шаблоном курса (“https://github.com/yamadharma/course-directory-student-template”) и скопируем шаблон (рис.12, 13 и 14)

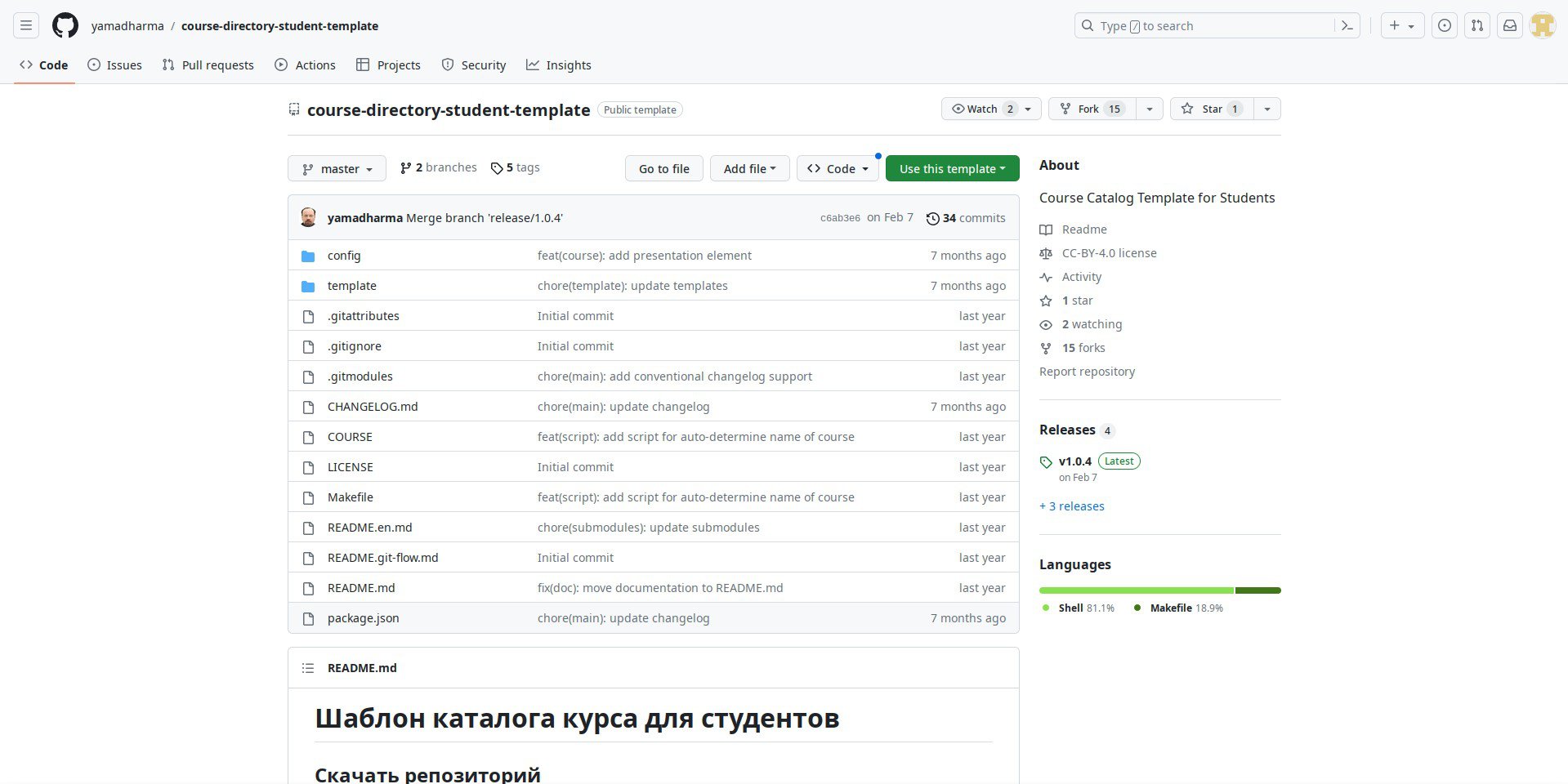


Рис. 12 Шаблон каталога на Github

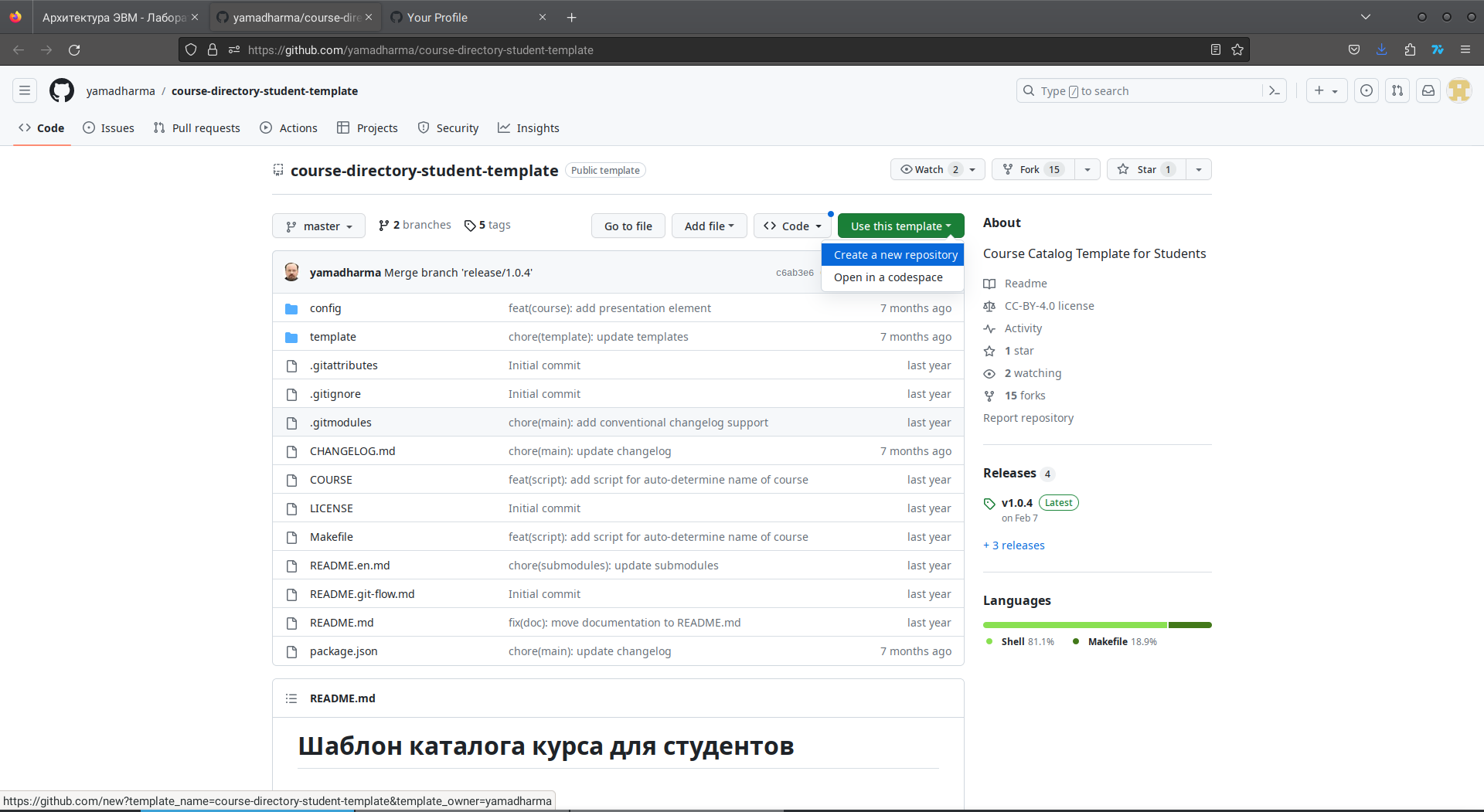


Рис.13 Создание нового репозитория на основе шаблона

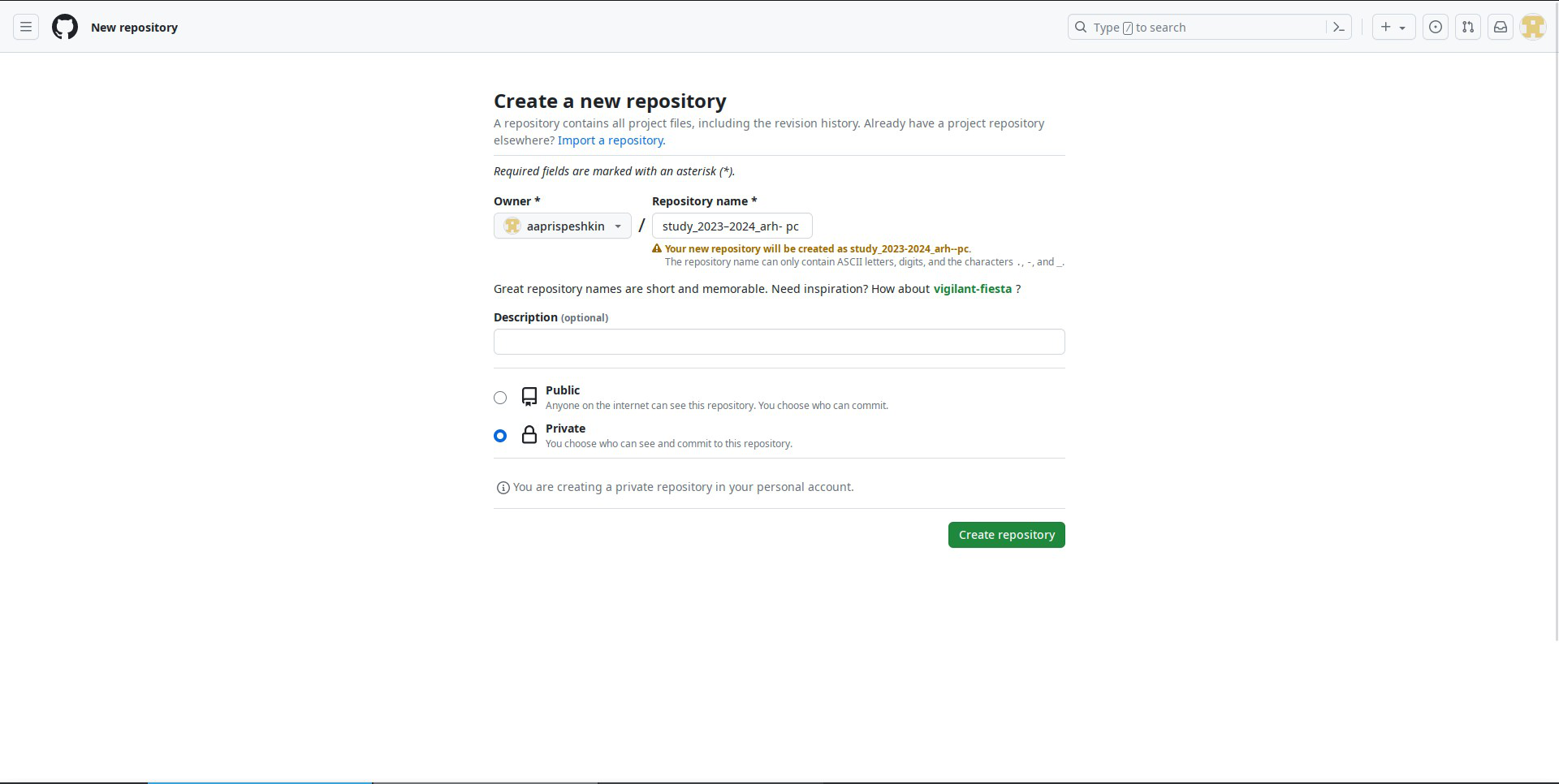


Рис.14 Окно создания нового репозитория

Пользуясь утилитой cd перейдём в каталог курса(рис.15).

Рис.15 Переход в созданный каталог курса

Рис.15 Переход в созданный каталог курса

Командой git clone с ключом –recursive и пользуясь ссылкой на github клонируем созданный репозиторий(рис.16).

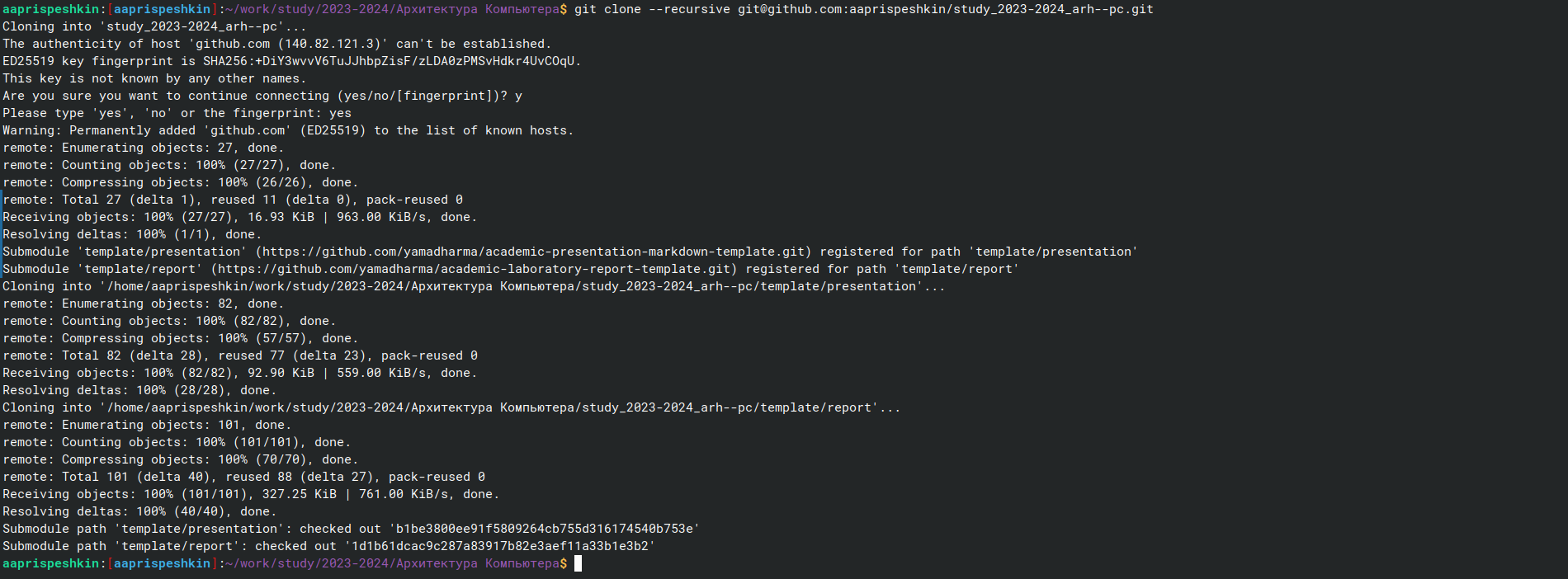


Рис.16 Клонирование репозитория

**6)** Утилитой rm удалим лишние файлы из каталога курса(рис.17).

Рис.17 Удаление файла package.json из каталога курса

Рис.17 Удаление файла package.json из каталога курса

Командами echo и make создадим необходимые каталоги(рис.18).

Рис.18 Создание нужных каталогов командами echo и make

Рис.18 Создание нужных каталогов командами echo и make

Командой git add добавим созданные каталоги на сервер, а командой git commit сохраним сделанные изменения(рис.19).

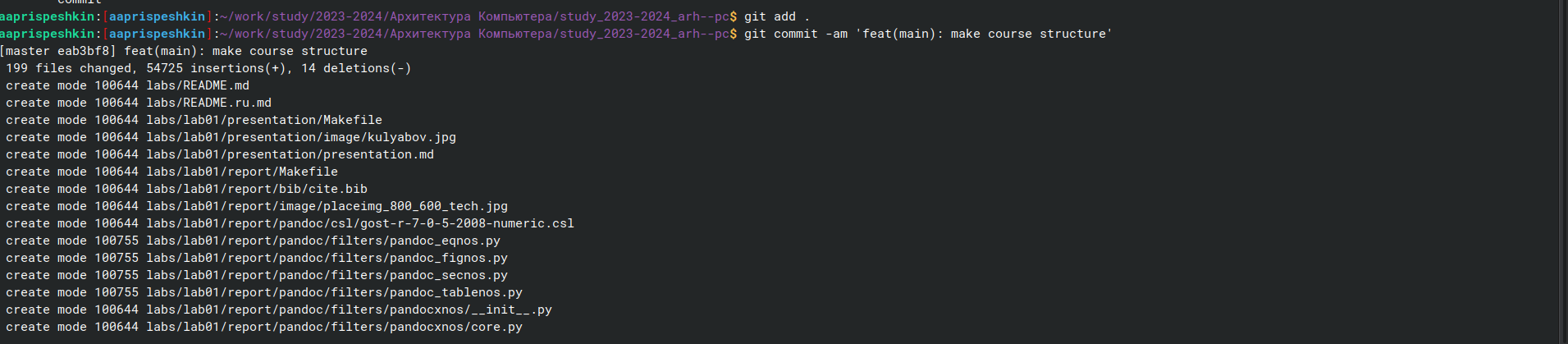


Рис.19 Использование команд git add и git commit

Командой git push отправим все сделанные изменения на сервер(рис.20).

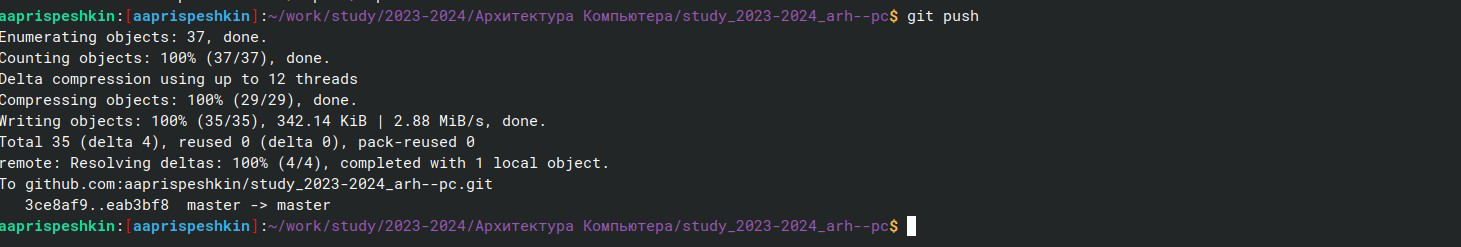


Рис.20 Отправка сделанных изменений на сервер

Проверим правильность сделанных изменений на github(рис.21).

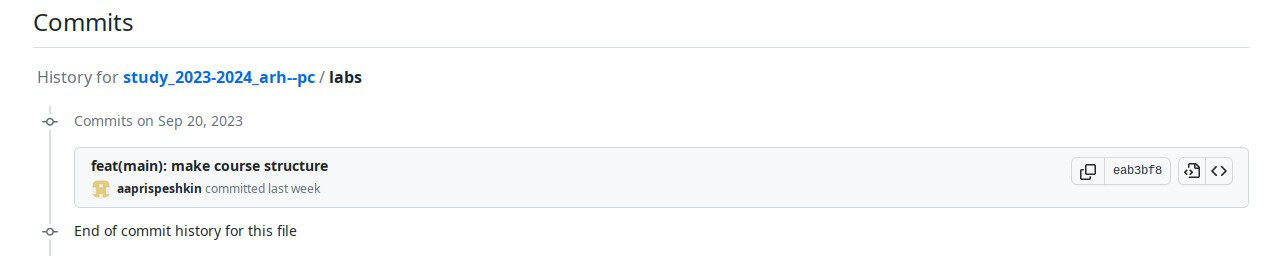


Рис.21 Проверка истории изменений на github

**7)** Перейдём в подкаталог lab02 и с помощью утилиты touch создадим файл отчёта по лабораторной работе(рис.22).

Рис.22 Создание отчёта по лабороторной работе (составлен отчёт будет в текстовом редакторе LibreOffice Writer)

Рис.22 Создание отчёта по лабороторной работе (составлен отчёт будет в текстовом редакторе LibreOffice Writer)

Скопируем отчёт по первой лабораторной работе из каталога Documents в каталог lab01/report и проверим правильность копирования утилитой ls(рис.23).

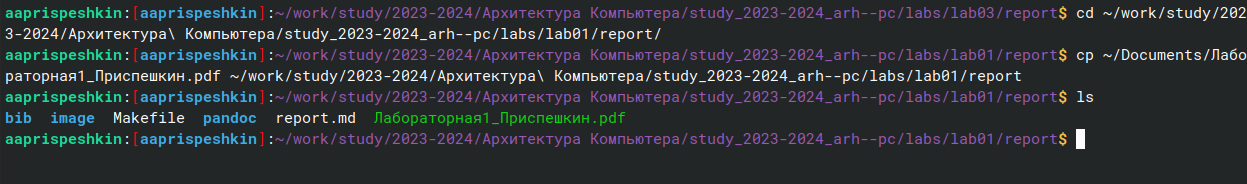


Рис.23 Копирование отчёта в рабочий репозиторий

Командой git add добавим отчёт по первой лабораторной работе на сервер, повторим эти действия с отчётом по второй лабораторной работе(рис.24, 25 и 26)

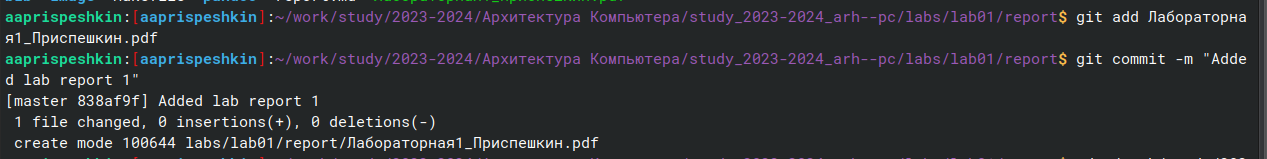


Рис. 24 Добавление отчёта по первой лаборатнорной работе

Рис.25 Добавление отчёта по второй лабораторной работе

Рис.25 Добавление отчёта по второй лабораторной работе

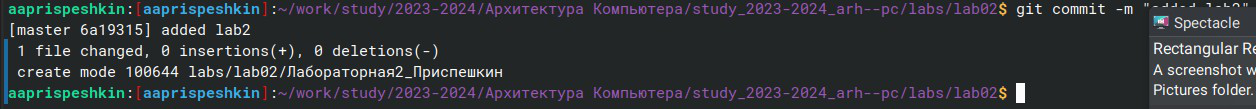


Рис.26 Использование команды git commit для сохранения изменений по отчёту второй лабораторной работы

Командой git push отправим все созданные файлы в центральный репозиторий(рис.27).

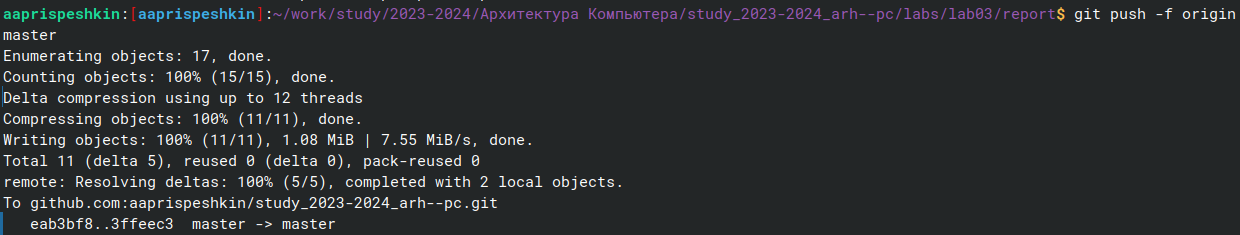


Рис.27 Отправка добавленных нами файлов в центральный репозиторий командой git push

Проверим правильность введённых нами команд в GitHub(рис.28).

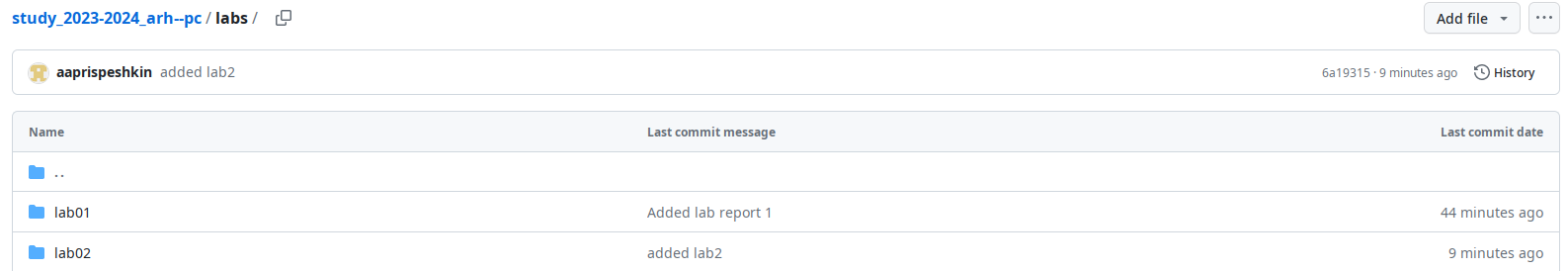


Рис.28 Проверка правильности добавления отчётов по лабораторной работе

**4) Вывод** В результате выполнения данной лабораторной работы я научился пользоваться системой контроля версий git в терминале Linux, а также создавать и обновлять репозитории в GitHub.

**5) Список литературы**

1. Лабораторная работа No2. Система контроля версий

Git