**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**по лабораторной работе № 2**

*дисциплина: Архитектура компьютера*

Студент: Лащиков Алексей Антонович

Группа: НКАбд-04-25

**МОСКВА**

2025 г.

**Содержание**

[1 Цель работы 5](#_Toc209104649)

[2 Задание 6](#_Toc209104650)

[**2.1 Повторение заданий лабораторной работы** 6](#_Toc209104651)

[**2.2 Выполнение заданий для самостоятельной работы** 6](#_Toc209104652)

[3 Теоретическое введение 7](#_Toc209104653)

[**3.1 Системы контроля версий. Общие понятия** 7](#_Toc209104654)

[**3.2 Система контроля версий Git** 7](#_Toc209104655)

[4 Выполнение лабораторной работы 7](#_Toc209104656)

[**4.1 Перемещение по файловой системе** 7](#_Toc209104657)

[**4.2 Создание пустых каталогов и файлов** 11](#_Toc209104658)

[**4.3 Перемещение и удаление файлов и каталогов** 13](#_Toc209104659)

[**4.4 Команда cat: вывод содержимого файлов** 14](#_Toc209104660)

[5 Выполнение заданий для самостоятельной работы 15](#_Toc209104661)

[6 Выводы 19](#_Toc209104662)

**Список иллюстраций**

**Список таблиц**

# **1 Цель работы**

Изучить идеологию и применение средств контроля версий, а также приобрести практические навыки работы с системой Git.

# **2 Задание**

## **2.1 Повторение заданий лабораторной работы**

Выполнить настройку и базовую работу с Git и GitHub согласно методическим указаниям: создать рабочее пространство и репозиторий курса из шаблона, выполнить первичные commits/push, а также подготовить скриншоты каждого шага.

## **2.2 Выполнение заданий для самостоятельной работы**

* 1. Создать отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report).
  2. Скопировать отчеты по выполнению предыдущих лабораторных работ соответствующие каталоги созданного рабочего пространства.
  3. Загрузить файлы на github.

# **3 Теоретическое введение**

## **3.1 Системы контроля версий. Общие понятия**

Системы контроля версий (VCS) применяются при совместной работе над проектами: позволяют фиксировать изменения, объединять правки нескольких участников, вести историю и при необходимости откатывать проект к ранним состояниям. Различают централизованные и распределённые VCS.

## **3.2 Система контроля версий Git**

Git — распределённая система контроля версий. Каждый разработчик обладает локальным репозиторием с полной историей, что упрощает резервирование и офлайн‑работу; взаимодействие с удалёнными репозиториями осуществляется через pull/push.

## **3.3 Основные команды Git**

В таблице 1 приведены основные команды Git.

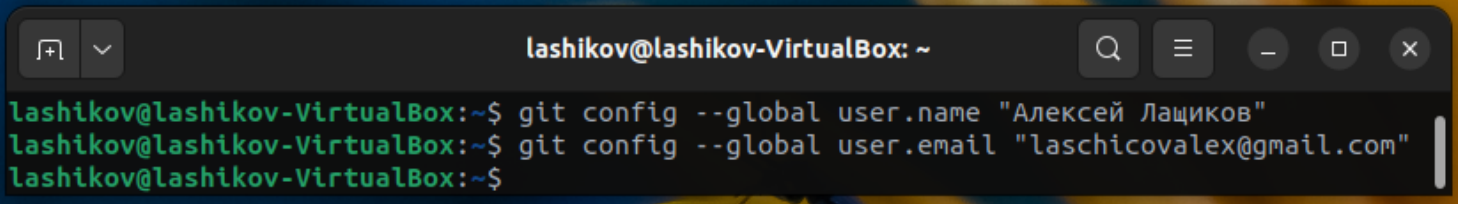
| Команда | Назначение |
| --- | --- |
| git init | Создание основного дерева репозитория |
| git pull | Получение обновлений текущего дерева из центрального репозитория |
| git push | Отправка всех произведённых изменений локального дерева в центральный репозиторий |
| git status | Просмотр списка изменённых файлов в текущей директории |
| git diff | Просмотр текущих изменений |
| git add . | Добавить все изменённые/новые файлы и каталоги в индекс |
| git add <имена\_файлов> | Добавить конкретные изменённые/новые файлы/каталоги в индекс |
| git rm <имена\_файлов> | Удалить файл/каталог из индекса (файл/каталог остаётся в ФС) |
| git commit -am 'Описание коммита' | Сохранить все добавленные изменения и изменённые файлы |
| git checkout -b <имя\_ветки> | Создать новую ветку на основе текущей и переключиться на неё |
| git checkout <имя\_ветки> | Переключиться на ветку (если нет локально — создастся и свяжется с удалённой) |
| git push origin <имя\_ветки> | Отправить изменения конкретной ветки в центральный репозиторий |
| git merge --no-ff <имя\_ветки> | Слить ветку с текущим деревом с сохранением merge-коммита |
| git branch -d <имя\_ветки> | Удалить локальную ветку, уже слитую с основным деревом |
| git branch -D <имя\_ветки> | Принудительно удалить локальную ветку |
| git push origin :<имя\_ветки> | Удалить ветку в центральном репозитории |

*Таблица 1 Список основных команд Git*

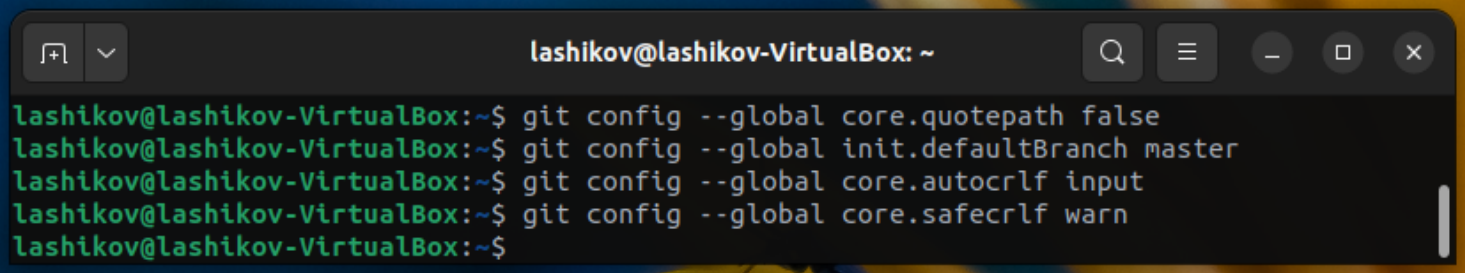
# **4 Выполнение лабораторной работы**

## **4.1 Базовая настройка git**

Сделал предварительную конфигурацию git. Открыл терминал и ввёл следующие команды, указав имя и email владельца репозитория.

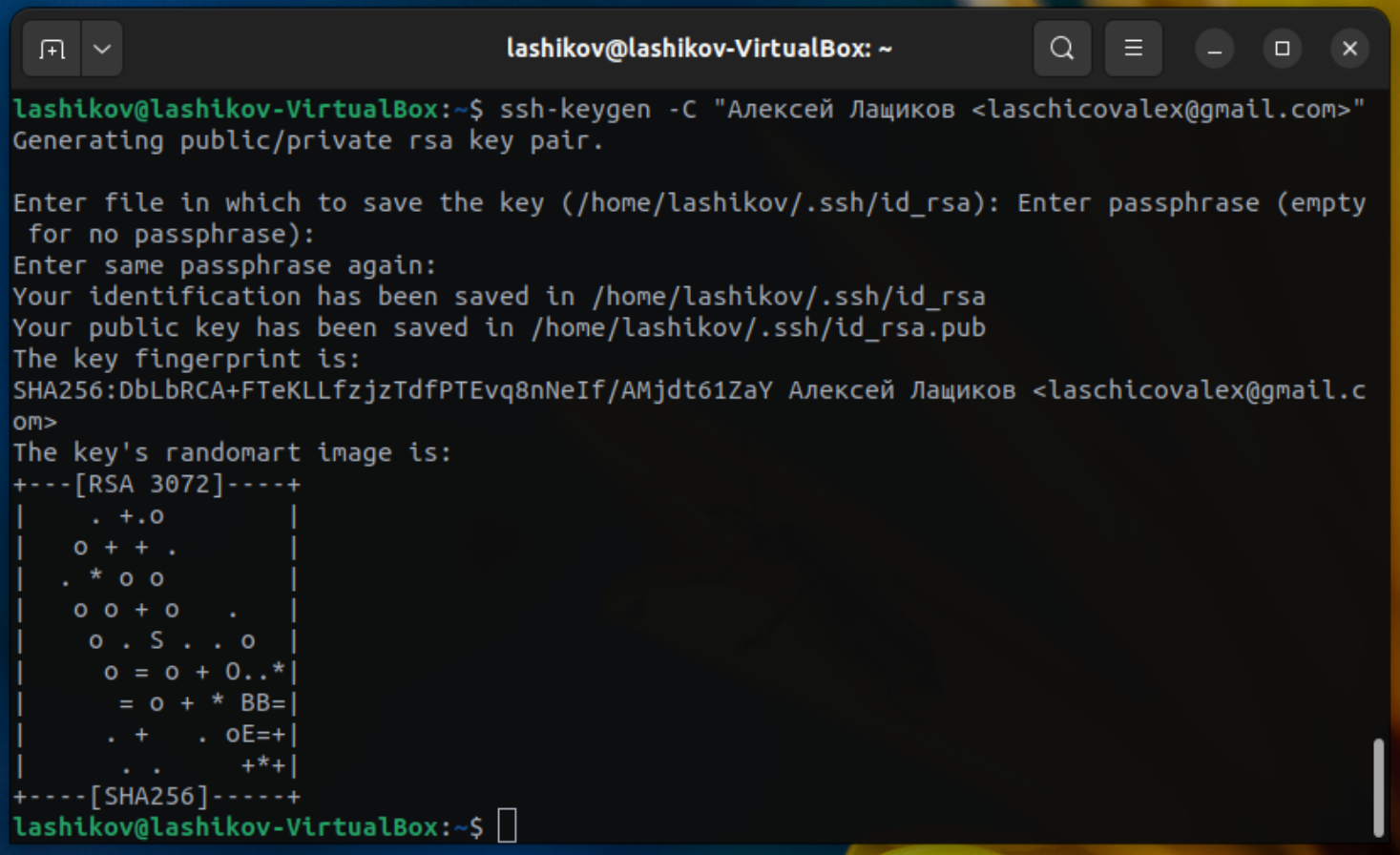


Настроил utf-8 в выводе сообщений git, задал имя начальной ветки (master), ввёл параметры autocrlf и safecrlf.

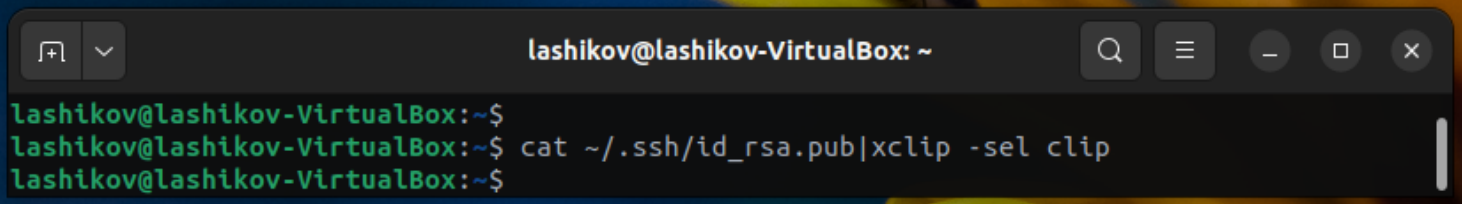


## **4.2 Создание SSH ключа**

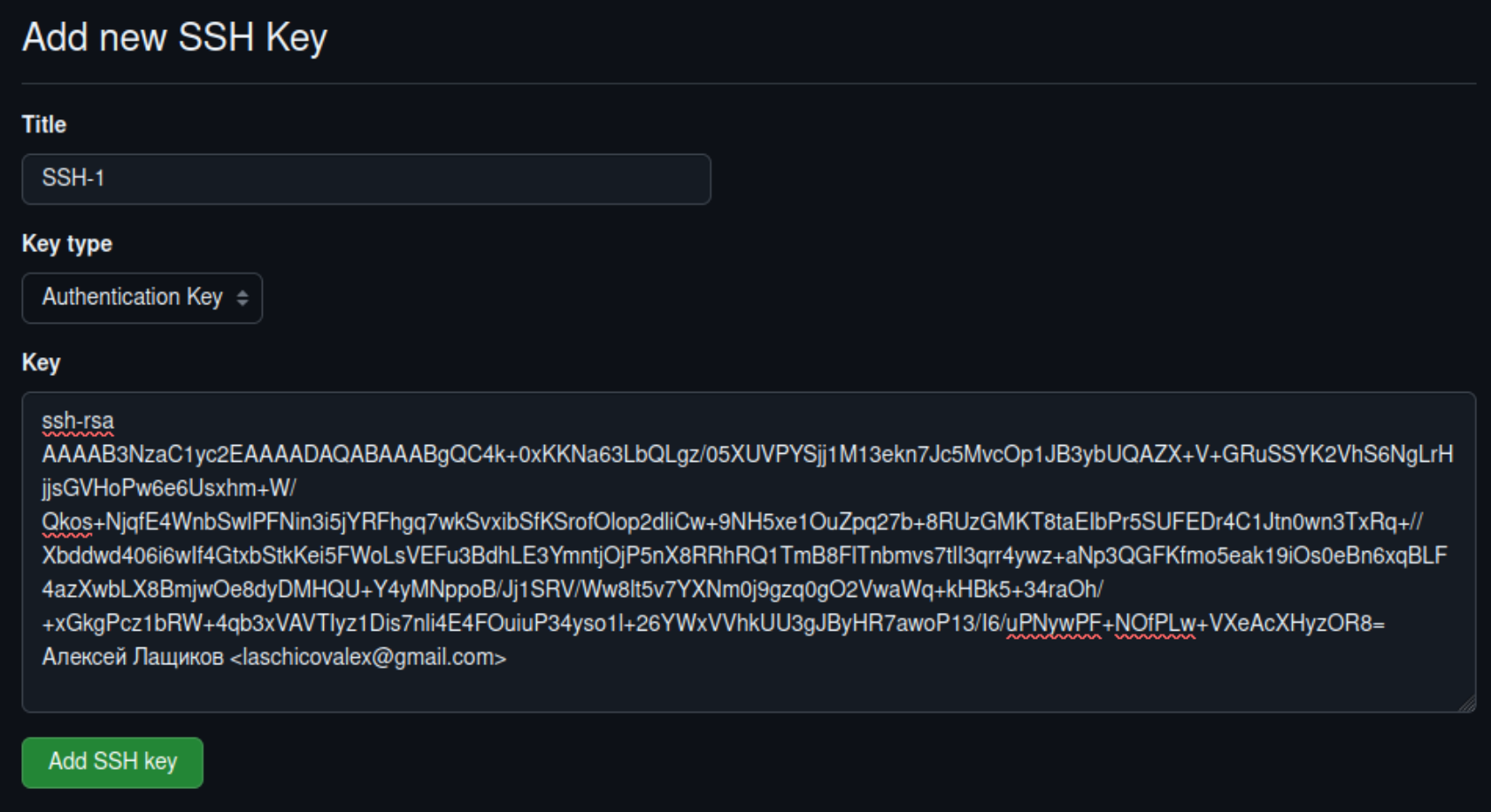
Сгенерировал новый SHH ключ.



Скопировал SSH ключ.

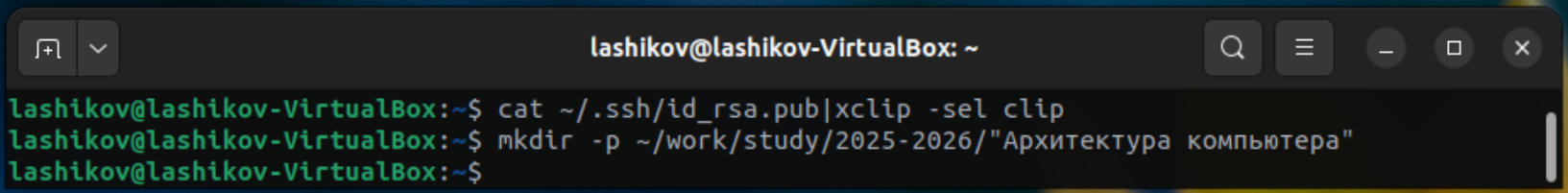


Загрузил сгенерённый открытый ключ на сайт http://github.org/.



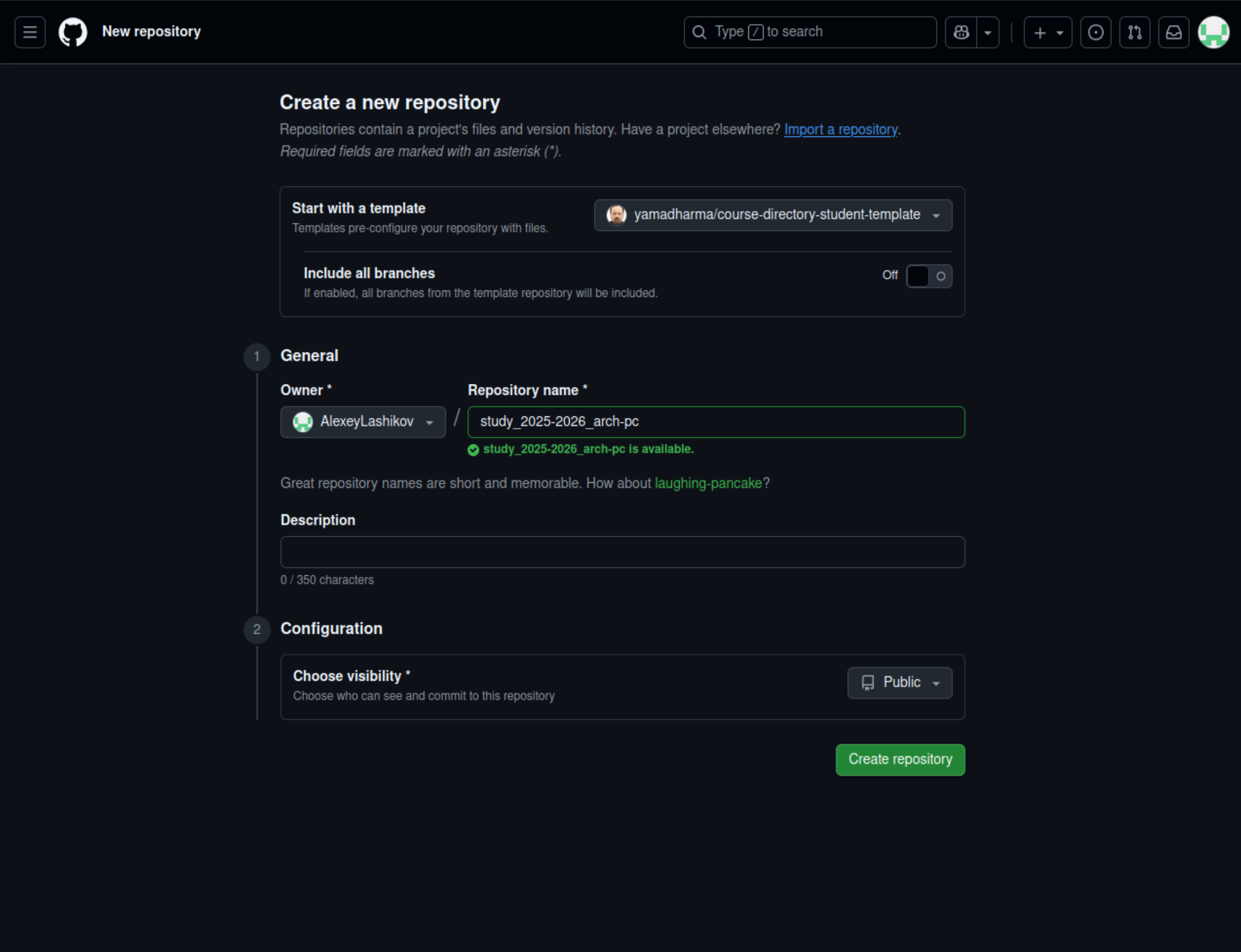
## **4.3 Создание рабочего пространства и репозитория курса на основе шаблона**

Создал каталог для предмета «Архитектура Компьютера».

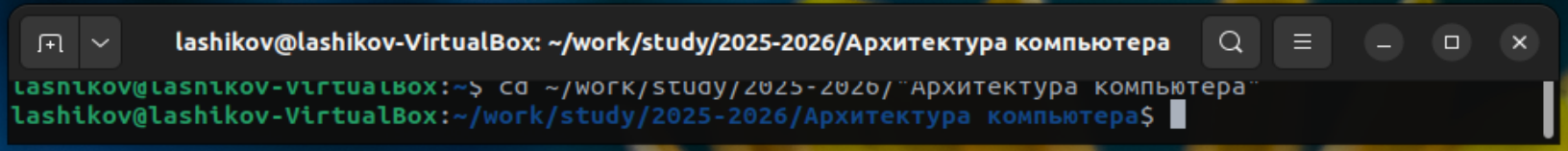


## **4.4 Создание репозитория курса на основе шаблона**

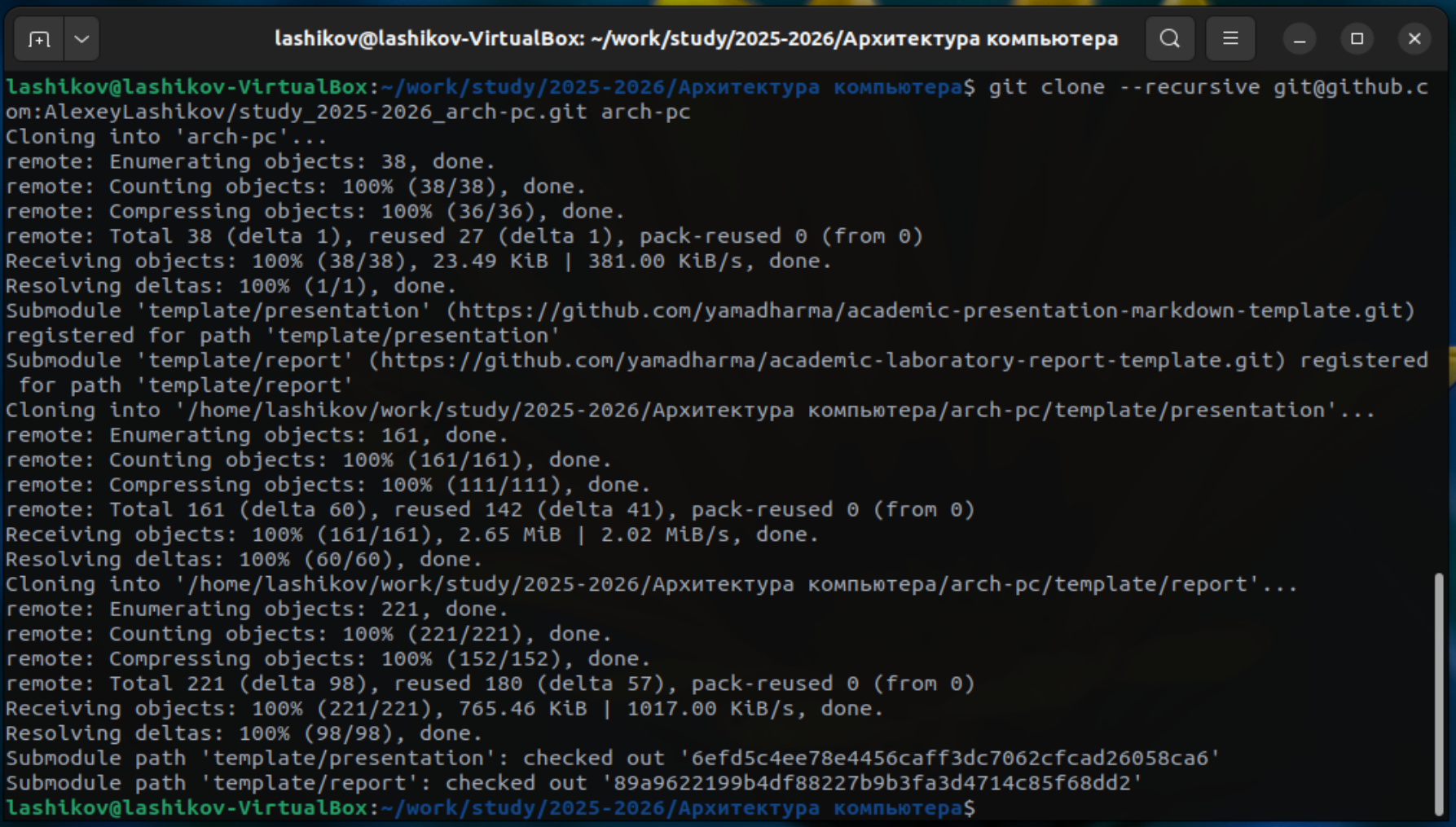
Перешёл на страницу репозитория с шаблоном курса, выбрал Use this template. Затем в открывшемся окне задал имя репозитория (study\_2025–2026\_arh-pc) и создал репозиторий.



Перешёл в каталог курса

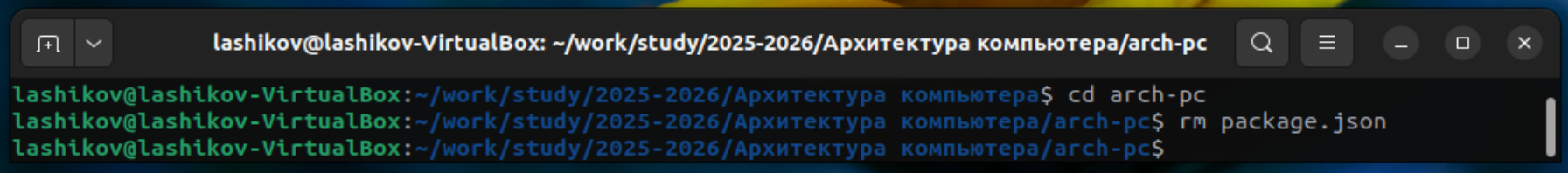


Клонировал созданный репозиторий

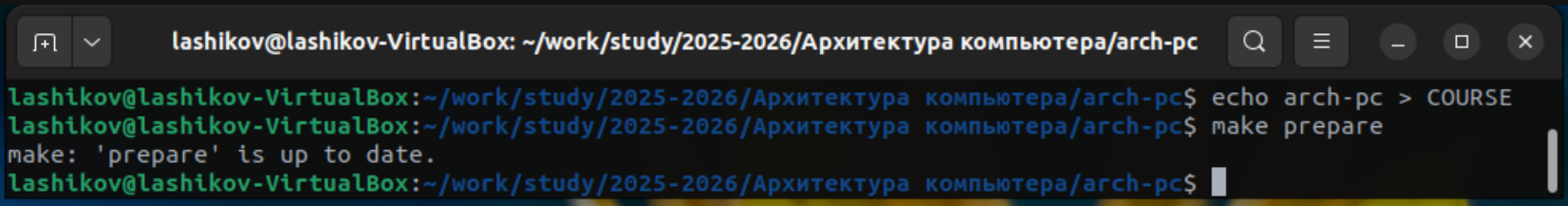


## **4.5 Настройка каталога курса**

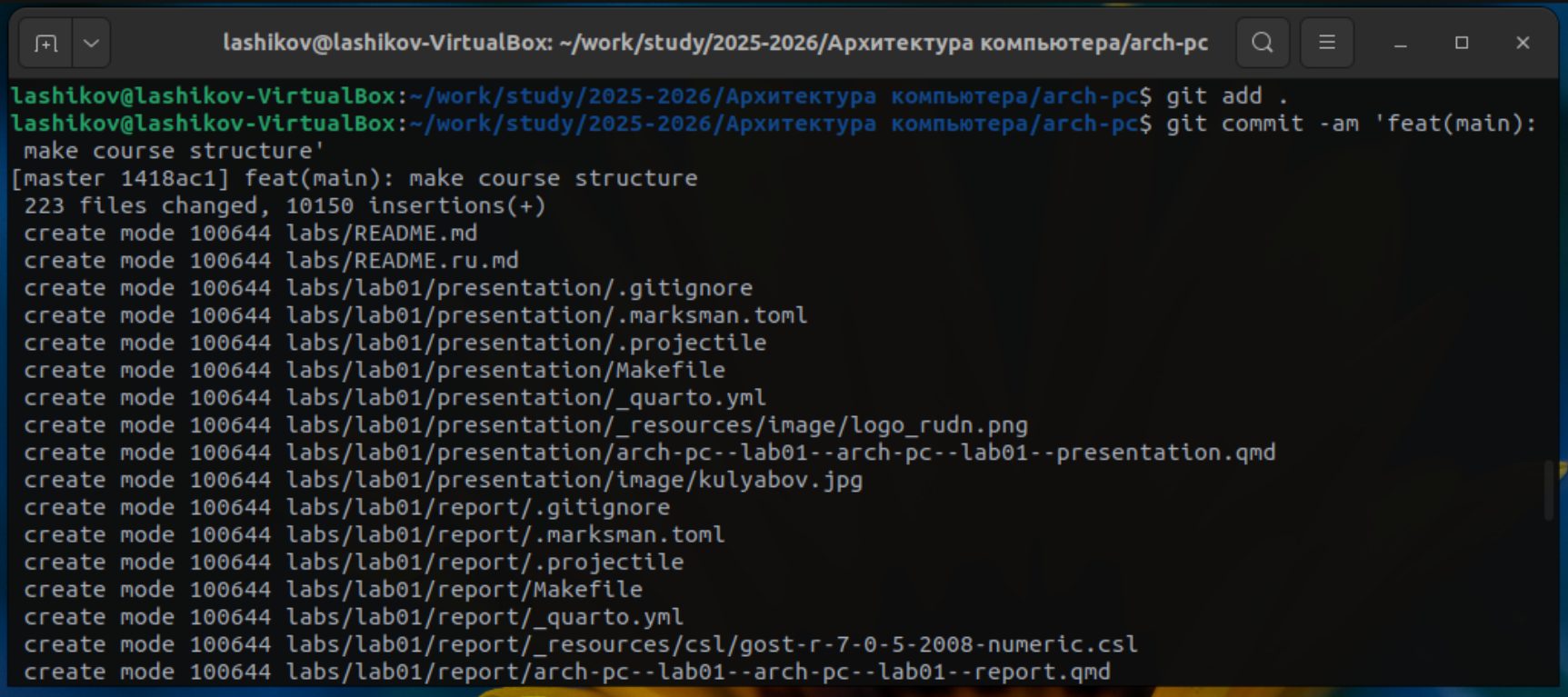
Перешёл в каталог курса и удалил лишние файлы

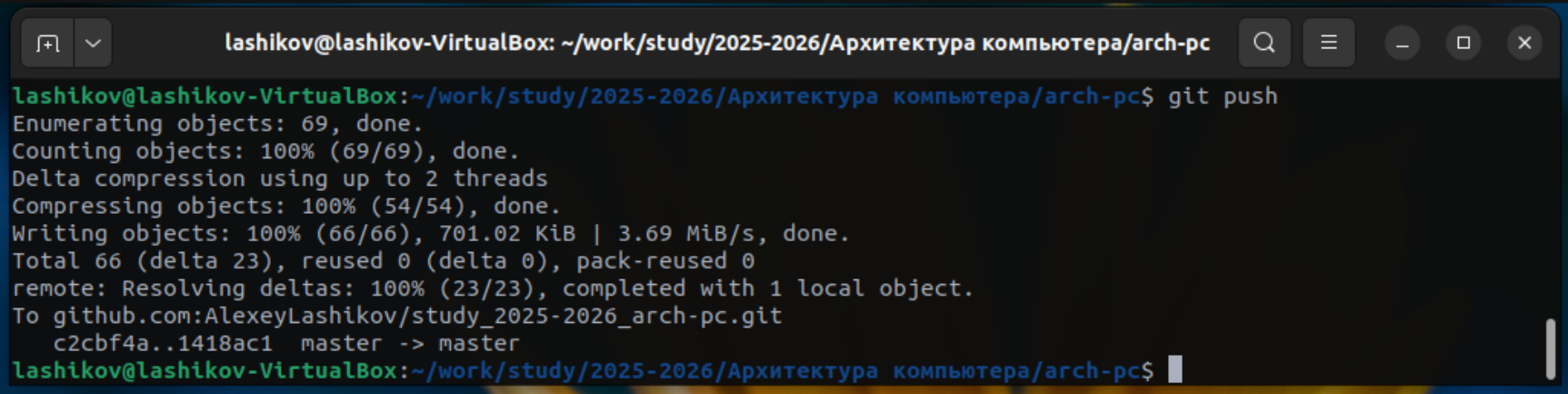


Создал необходимые каталоги

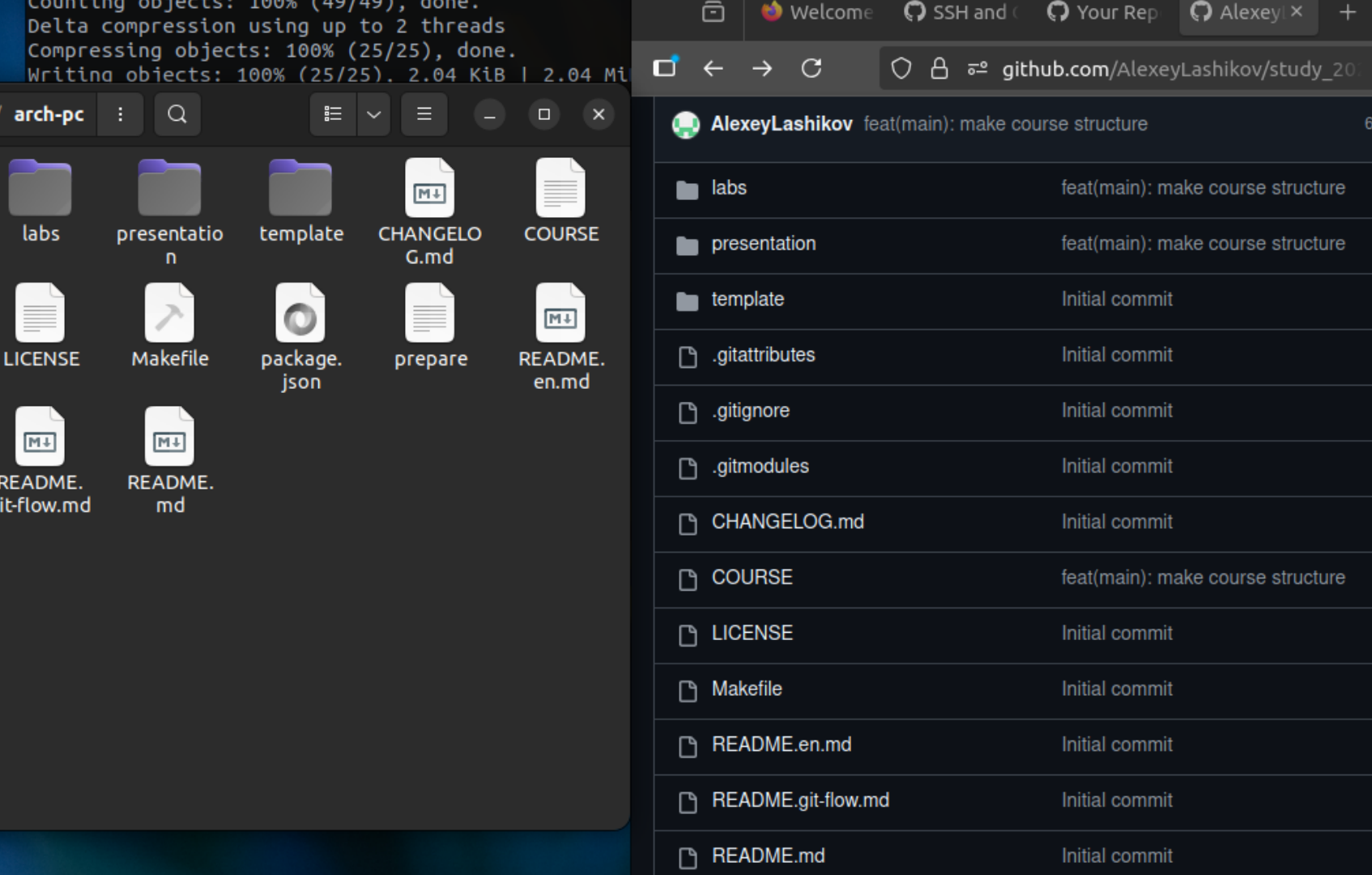


Отправил файлы на сервер





Проверил правильность создания иерархии рабочего пространства в локальном репозитории и на странице github.



# **5 Выполнение заданий для самостоятельной работы**

1. Создал отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report).
2. Скопировал отчеты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства.
3. Загрузил файлы на github.

# **6 Выводы**

В процессе выполнения работы я освоил базовые команды Linux для навигации по файловой системе, создания, копирования, перемещения и удаления файлов и каталогов. Научился использовать команды ‘cd’, ‘ls’, ‘mkdir’, ‘touch’, ‘cp’, ‘mv’, ‘rm’, ‘cat’.