

# 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: Мазурский Александр

Группа: НКАбд-02-24

МОСКВА

2024 г.

## Оглавление

1	Цель работы.....	3
3	Теоретическое введение .....	5
4	Выполнение лабораторной работы.....	7
4.1	Техническое обеспечение .....	7
4.2	Базовая настройка Git .....	7
4.3	Создание рабочего пространства и репозитория курса на основе шаблона.....	10
4.4	Создание репозитория курса на основе шаблона.	10
4.5	Настройка каталога курса.....	13
5	Задания для самостоятельной работы.....	14
6	Выводы.....	15
	Список литературы .....	16

# 1 Цель работы

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

## 2 Задание

На основе методических указаний провести работу с базовыми командами системы контроля версий git, выучить применение команд для разных случаев использования, настроить GitHub.

### 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 проекта. 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, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (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 хранилища 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 конфликтов, 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 заблокировать 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.

<code>git commit -am</code> 'Описание коммита'	сохранить все добавленные изменения и все изменённые файлы
<code>git checkout -b</code> имя_ветки	создание новой ветки, базирующейся на текущей
<code>git checkout</code> имя_ветки	переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой)
<code>git push origin</code> имя_ветки	отправка изменений конкретной ветки в центральный репозиторий
<code>git merge</code> --no-ff имя_ветки	слияние ветки с текущим деревом
<code>git branch -d</code> имя_ветки	удаление локальной уже слитой с основным деревом ветки
<code>git branch -D</code> имя_ветки	принудительное удаление локальной ветки
<code>git push origin</code> :имя_ветки	удаление ветки с центрального репозитория

Таблица 3.1 Описание некоторых команд системы контроля версий Git.

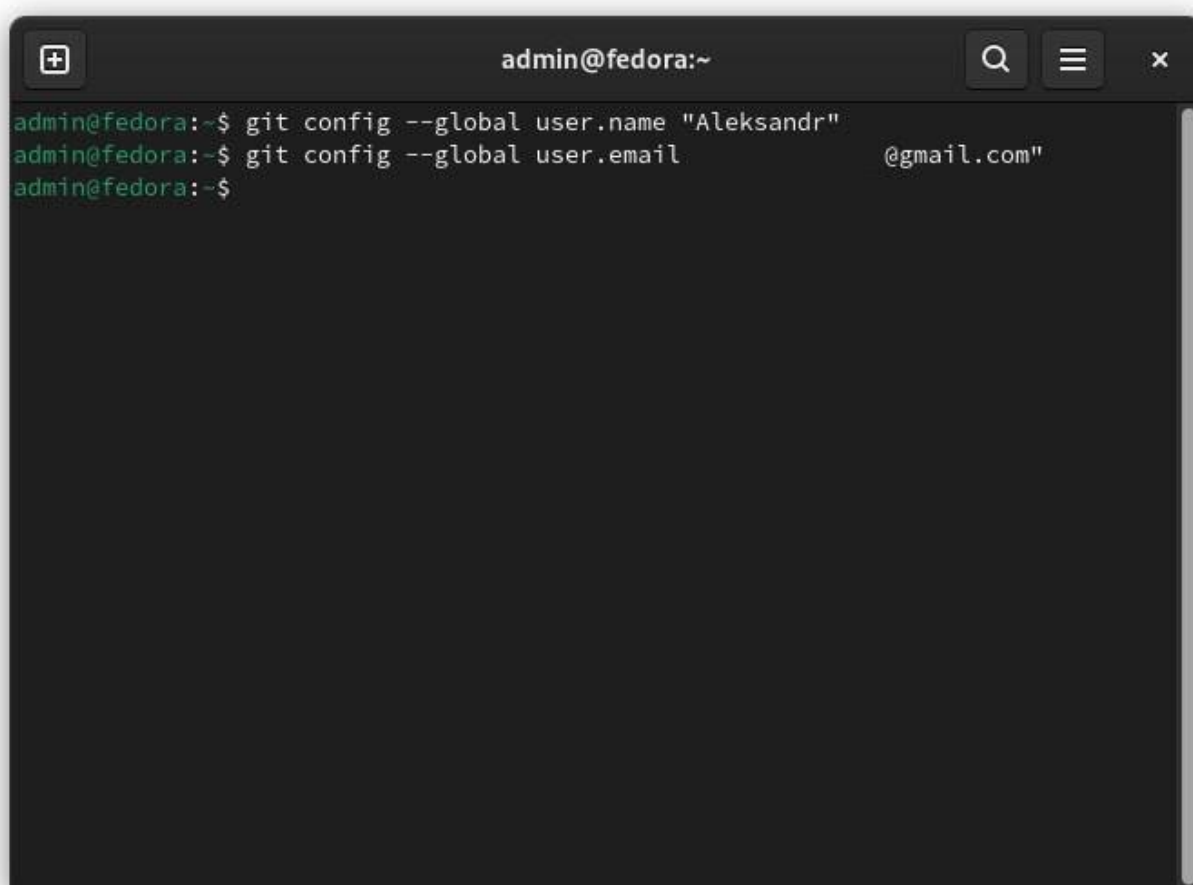
## 4 Выполнение лабораторной работы

### 4.1 Техническое обеспечение

Лабораторная работа была выполнена на домашнем компьютере под управлением операционной системы Fedora Workstation 40.

### 4.2 Базовая настройка Git

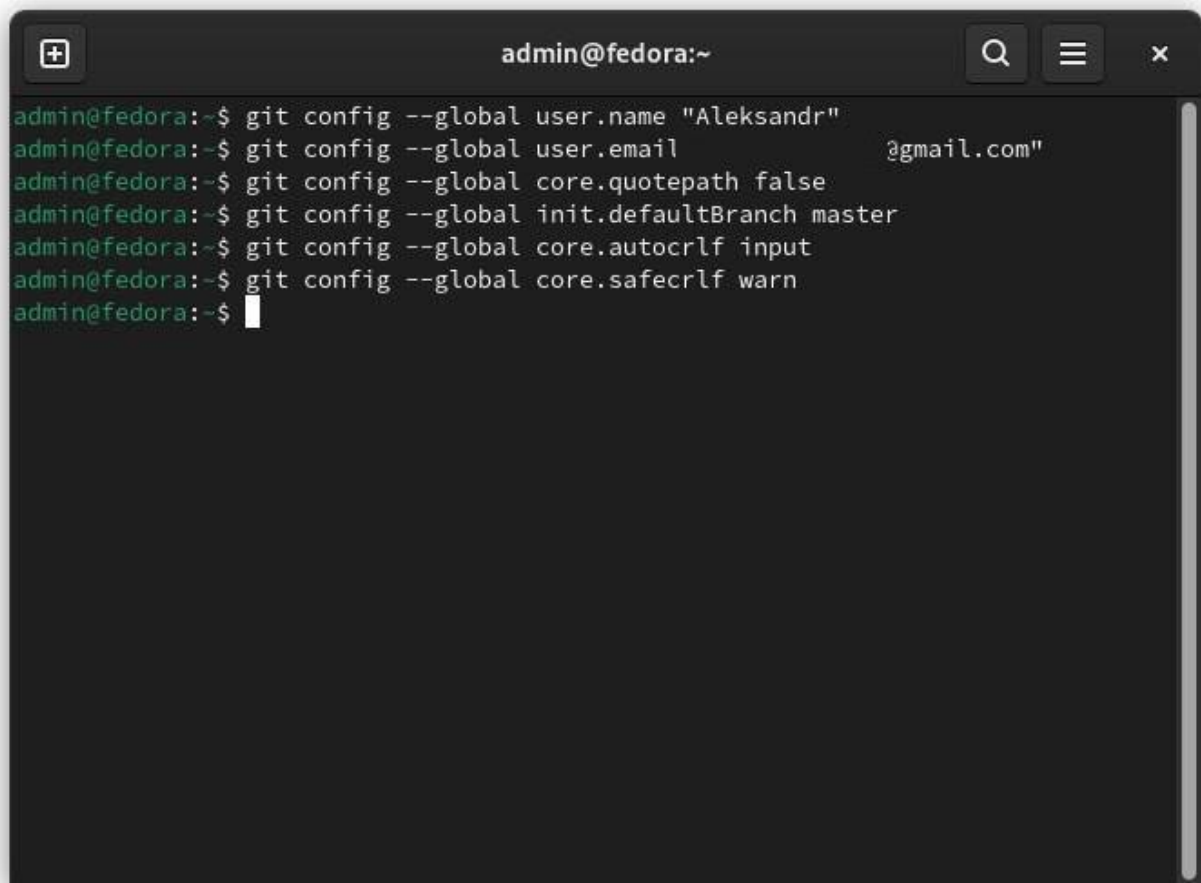
Для начала я проведу предварительную конфигурацию Git, для этого открываю терминал и ввожу команды на (рис. 4.2.1)

A screenshot of a terminal window titled 'admin@fedora:~'. The window has a dark background and a light gray border. The terminal shows three lines of text: 'admin@fedora:~\$ git config --global user.name "Aleksandr"', 'admin@fedora:~\$ git config --global user.email @gmail.com"', and 'admin@fedora:~\$'. The prompt character is a green dollar sign. The window has standard Linux window controls (minimize, maximize, close) in the top right corner and a search icon in the top left corner.

```
admin@fedora:~$ git config --global user.name "Aleksandr"
admin@fedora:~$ git config --global user.email @gmail.com"
admin@fedora:~$
```

Рис. 4.2.1 Предварительная конфигурация Git.

Далее настраиваю параметры utf-8, имя начальной ветки, autocrlf и safecrlf (рис. 4.2.2)

A terminal window titled 'admin@fedora:~' with search, menu, and close buttons in the title bar. The terminal shows a series of 'git config' commands being executed to set global Git settings. The commands are: 'git config --global user.name "Aleksandr"', 'git config --global user.email @gmail.com', 'git config --global core.quotepath false', 'git config --global init.defaultBranch master', 'git config --global core.autocrlf input', and 'git config --global core.safecrlf warn'. The prompt returns to 'admin@fedora:~\$' after the last command.

```
admin@fedora:~$ git config --global user.name "Aleksandr"
admin@fedora:~$ git config --global user.email @gmail.com
admin@fedora:~$ git config --global core.quotepath false
admin@fedora:~$ git config --global init.defaultBranch master
admin@fedora:~$ git config --global core.autocrlf input
admin@fedora:~$ git config --global core.safecrlf warn
admin@fedora:~$
```

Рис. 4.2.2 Настройка параметров Git.

Далее создаю пару ssh ключей для интеграции с платформой GitHub (рис. 4.2.3)



```
admin@fedora:~$ ssh-keygen -C "Aleksandr Mazurskiy@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/admin/.ssh/id_ed25519):
Created directory '/home/admin/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/admin/.ssh/id_ed25519
Your public key has been saved in /home/admin/.ssh/id_ed25519.pub
The key fingerprint is:
                                Aleksandr Mazurskiy
                                @gmail.com
The key's randomart image is:
```

Рис. 4.2.3 Создание пары ssh ключей.

Далее я перехожу на сайт GitHub, авторизуюсь, перехожу в настройки аккаунта, вставляю публичный ключ в предназначенном для этого поле. (рис. 4.2.4)

### Add new SSH Key

Title

Key type

Authentication Key ↕

Key

ssh-ed25519

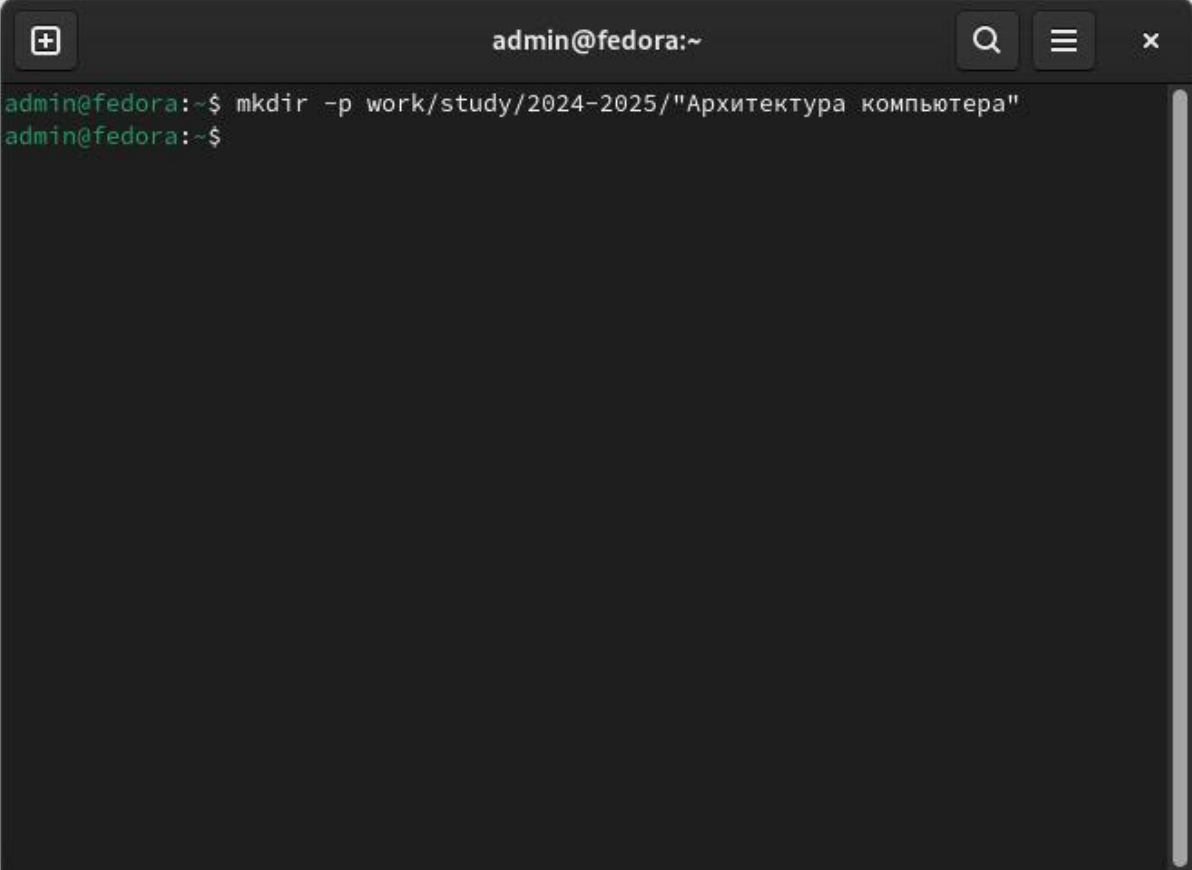
@gmail.com

Aleksandr Mazurskiy

Add SSH key

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

Рабочее пространство при выполнении лабораторных работ должно придерживаться определённой структурной иерархии, для этого я создаю директорию на своем рабочем компьютере (рис. 4.3.1)

A terminal window titled 'admin@fedora:~' with search, menu, and close buttons. It shows the command 'mkdir -p work/study/2024-2025/"Архитектура компьютера"' being executed, followed by a new prompt.

```
admin@fedora:~$ mkdir -p work/study/2024-2025/"Архитектура компьютера"
admin@fedora:~$
```

### 4.4 Создание репозитория курса на основе шаблона.

Создаю репозиторий на основе имеющего шаблона (рис. 4.4.1) через функционал клонирования интерфейса GitHub. (рис 4.4.2)

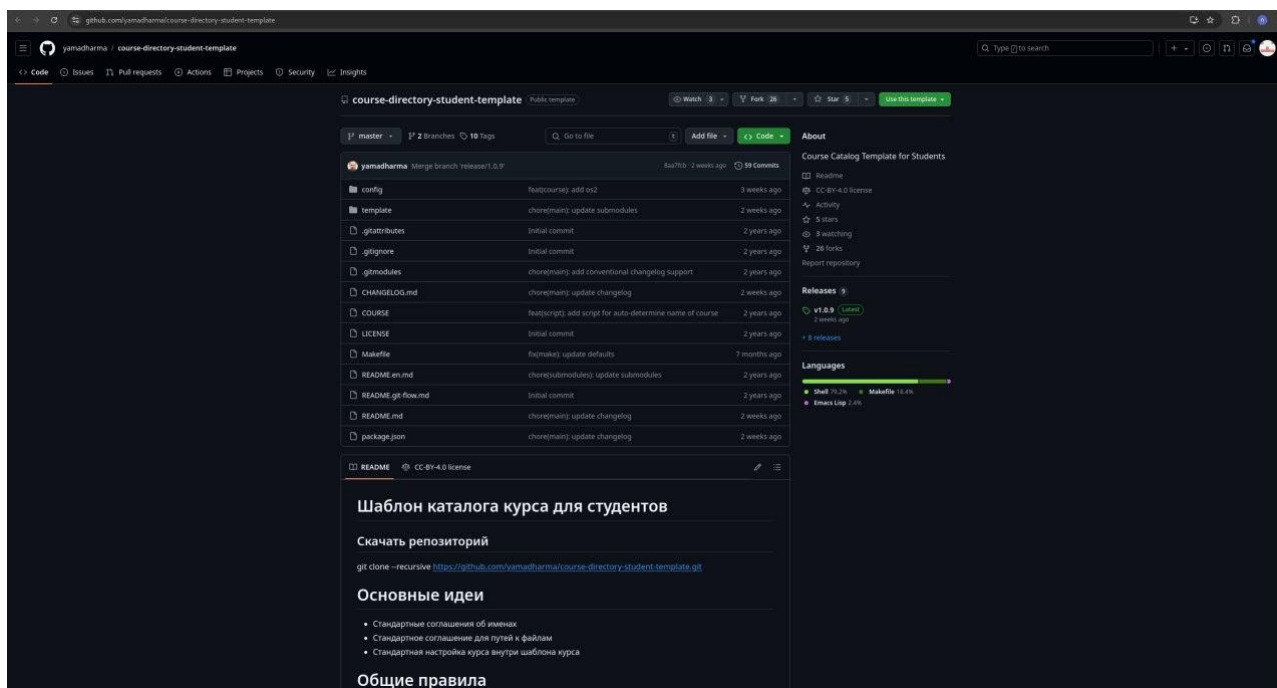



Рис. 4.4.1 Шаблон для клонирования на платформе GitHub.

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

*Required fields are marked with an asterisk (\*).*


**Repository template**

 yamadharm/course-directory-student-template

Start your repository with a template repository's contents.

☐ **Include all branches**  
Copy all branches from yamadharm/course-directory-student-template and not just the default branch.

**Owner \*** **Repository name \***

 nowherewasthere / study\_2024\_2025\_arch-pc

✔ study\_2024\_2025\_arch-pc is available.

Great repository names are short and memorable. Need inspiration? How about [solid-octo-spork](#) ?

**Description (optional)**

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

*You are creating a public repository in your personal account.*

[Create repository](#)

Рис. 4.4.2 Процесс клонирования репозитория через интерфейс GitHub.

Сгенерированный репозиторий на основе шаблона клонирую на свой рабочий компьютер, для этого беру ссылку для клонирования через интерфейс GitHub (рис. 4.4.3) и затем ввожу в терминале `git clone`. (рис 4.4.4)

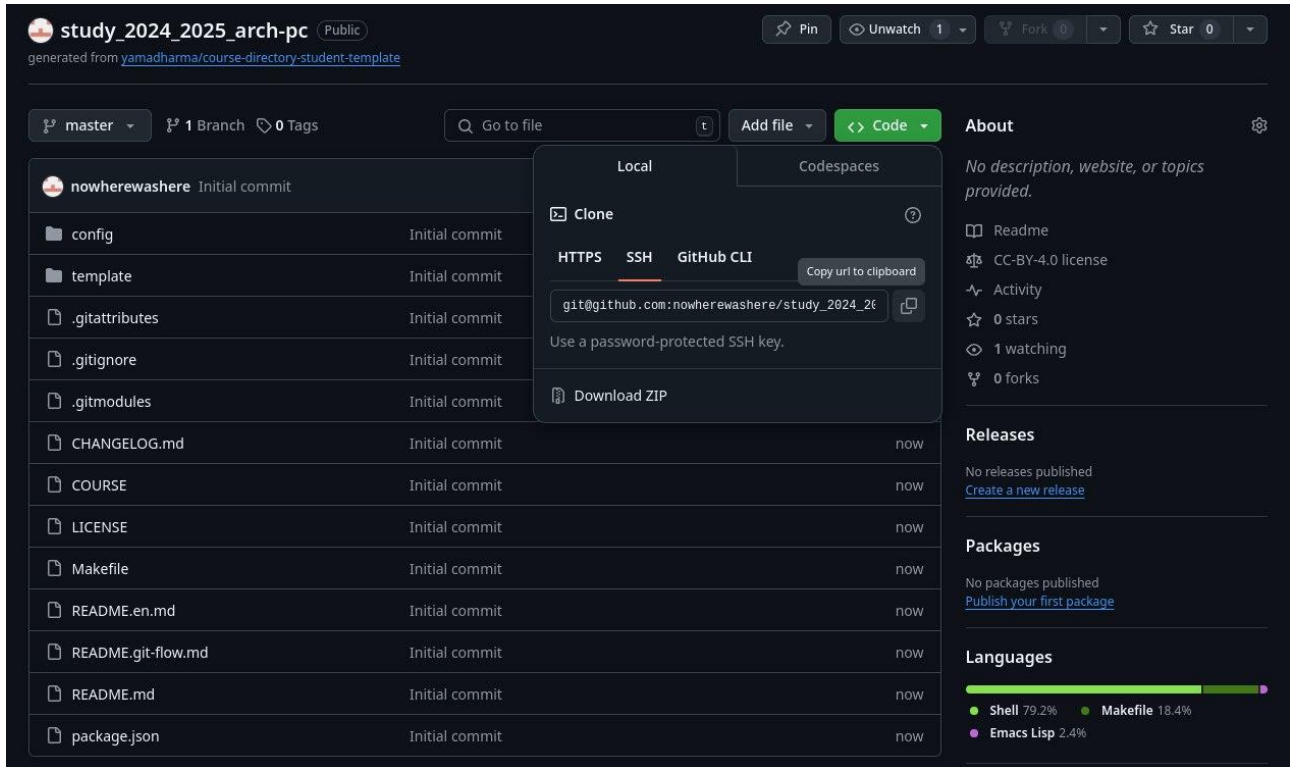


Рис. 4.4.3 Копирование ссылки для последующей вставки в терминал.

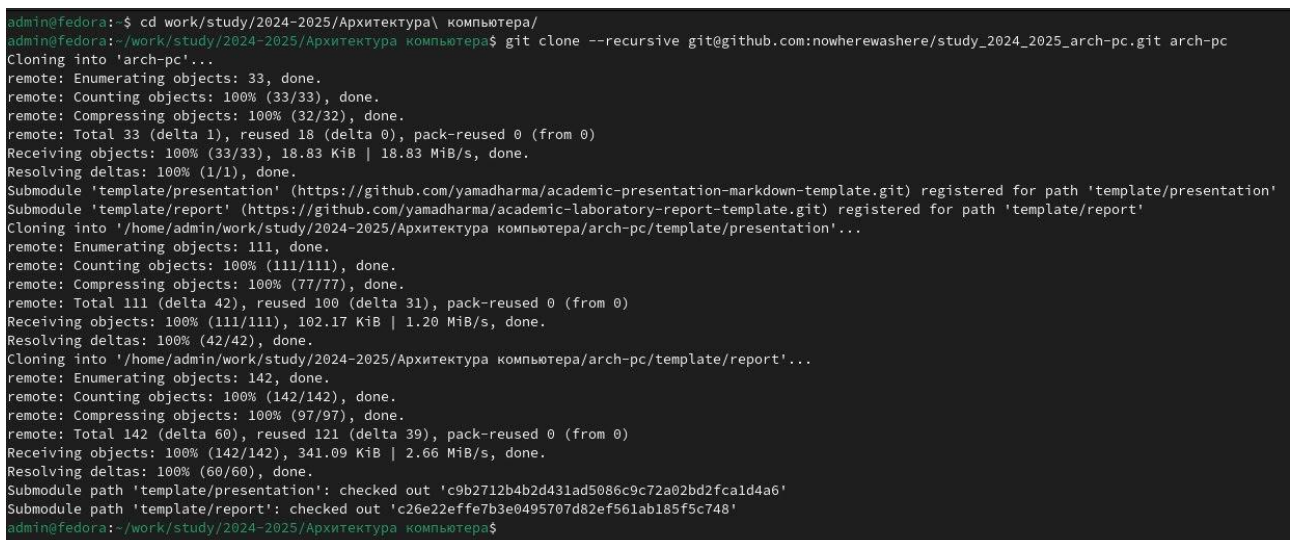
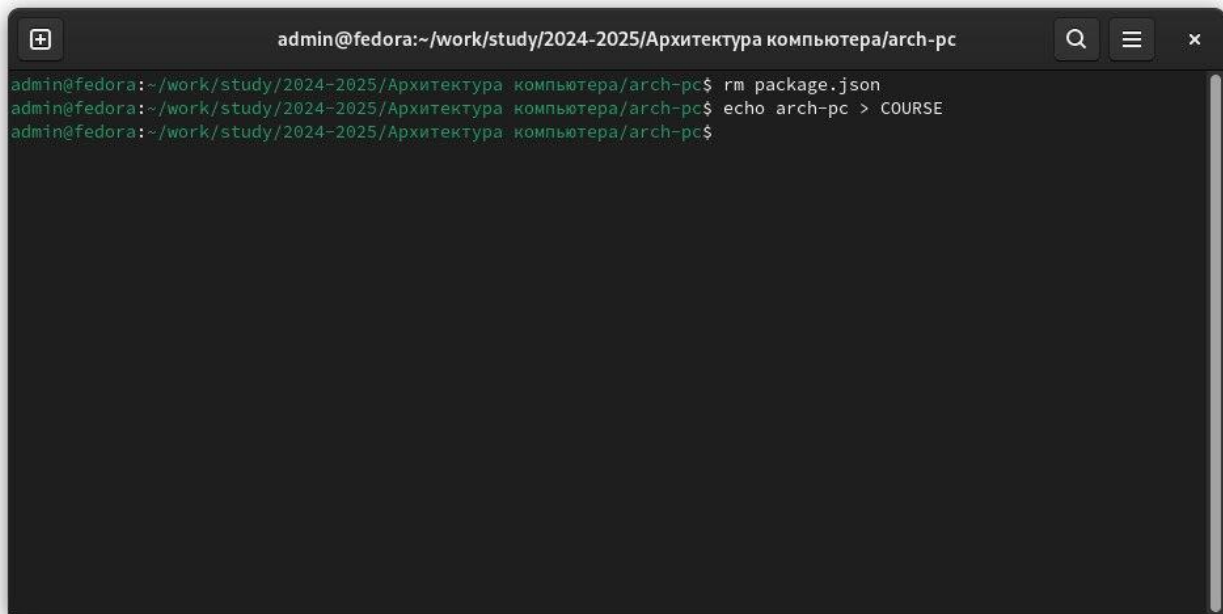


Рис. 4.4.4 Копирование репозитория на рабочий компьютер.

## 4.5 Настройка каталога курса.

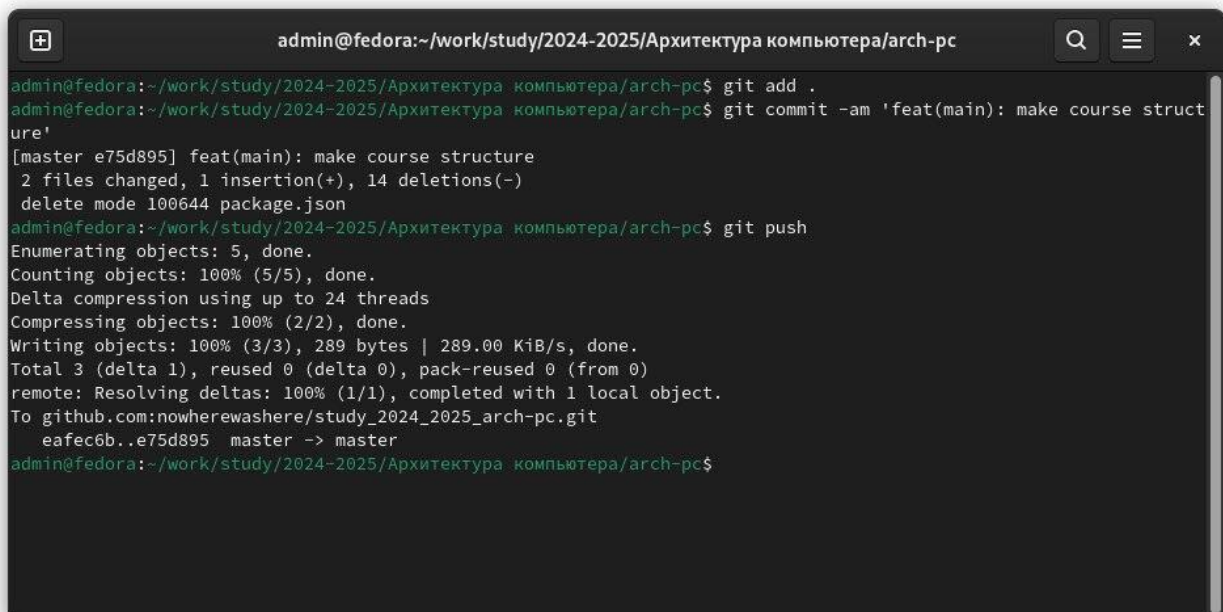
В каталоге курса удаляю лишние файлы и формирую необходимые каталоги (рис. 4.3.1)

A terminal window with a dark background and light green text. The title bar shows the user 'admin@fedora' and the current directory '~/work/study/2024-2025/Архитектура компьютера/arch-pc'. The terminal contains three lines of commands and their outputs: 'rm package.json', 'echo arch-pc > COURSE', and a prompt for the next command.

```
admin@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ rm package.json
admin@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
admin@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 4.3.1 Настройка каталога курса.

Делаю снимок сделанных изменений и push'у их на свой репозиторий в GitHub. (рис 4.3.2)

A terminal window with a dark background and light green text. The title bar shows the user 'admin@fedora' and the current directory '~/work/study/2024-2025/Архитектура компьютера/arch-pc'. The terminal contains several lines of git commands and their outputs: 'git add .', 'git commit -am 'feat(main): make course structure'', and 'git push'. The output shows the commit details and the push progress to the GitHub repository.

```
admin@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
admin@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
[master e75d895] feat(main): make course structure
 2 files changed, 1 insertion(+), 14 deletions(-)
 delete mode 100644 package.json
admin@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 24 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 289 bytes | 289.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:nowherewashere/study_2024_2025_arch-pc.git
 eafec6b..e75d895 master -> master
admin@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис 4.3.2 Отправка изменений на удаленный репозиторий.

# 5 Задания для самостоятельной работы.

Через терминал отправляю предыдущий отчет по лабораторной работе на свой удаленный репозиторий в GitHub (рис. 5.1), затем проверяю изменения на самом GitHub. (рис 5.2)

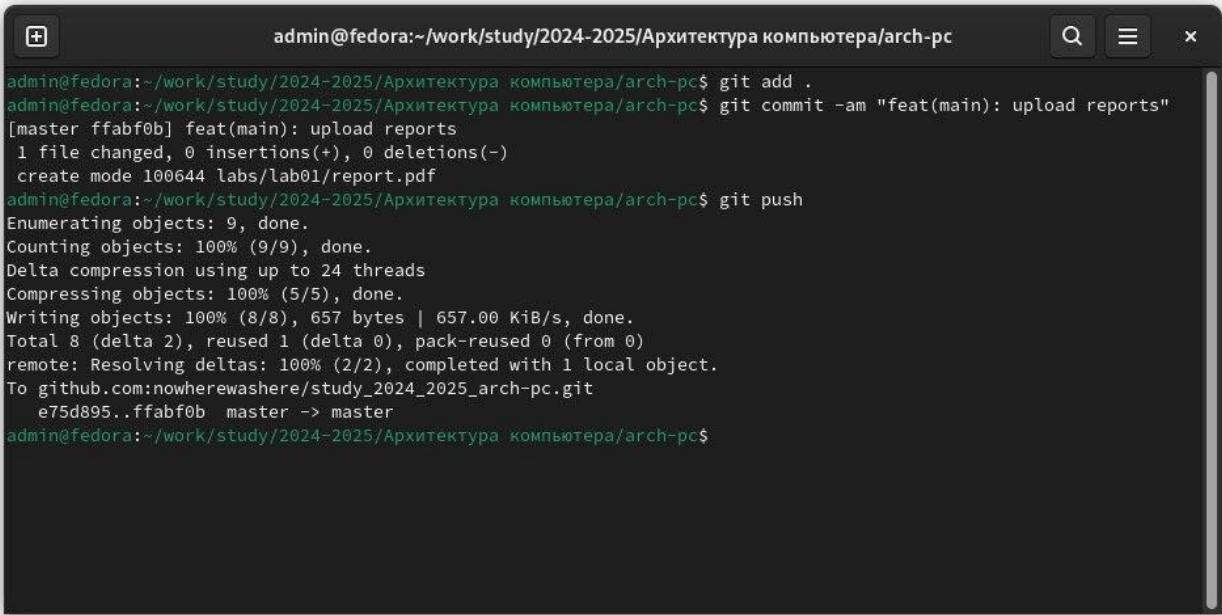


Рис 5.1 Отправка изменений на удаленный репозиторий.

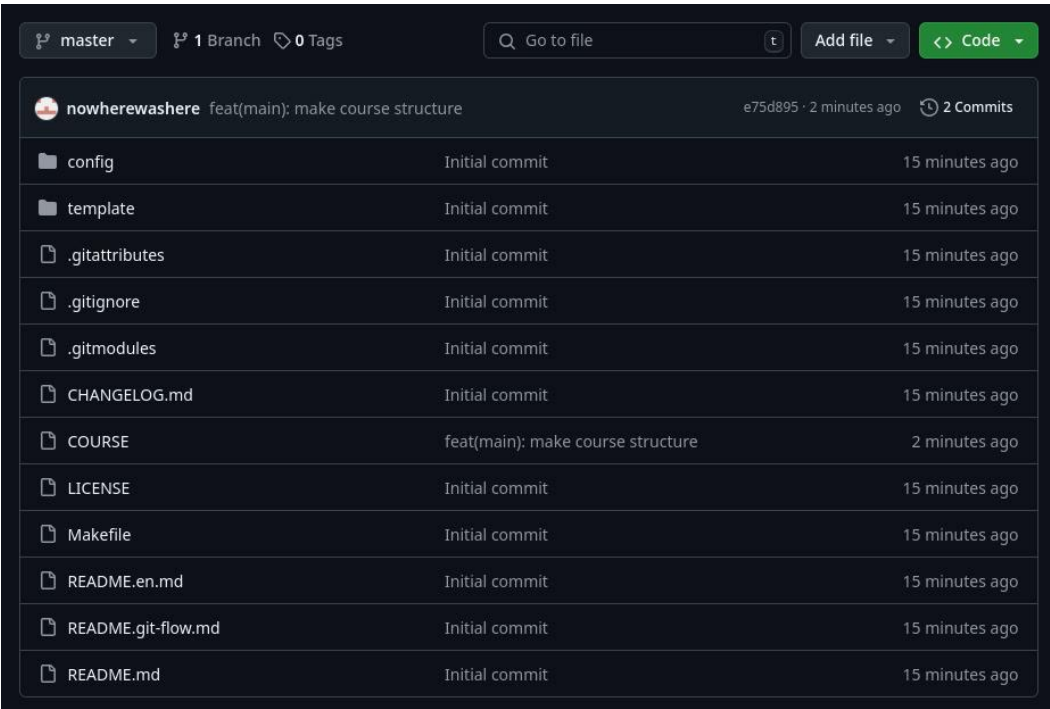


Рис 5.2 Проверка выполненных изменений на GitHub.

## **6 Выводы**

При выполнении данной лабораторной работы я изучил идеологию и применение средств контроля версий, а также приобрел практические навыки по работе с реализацией VSC git.



## Список литературы

1. [https://esystem.rudn.ru/pluginfile.php/2089082/mod\\_resource/content/0/%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%962.%20%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0%20%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8F%20%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B9%20Git.pdf](https://esystem.rudn.ru/pluginfile.php/2089082/mod_resource/content/0/%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%962.%20%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0%20%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8F%20%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B9%20Git.pdf)
2. <https://esystem.rudn.ru/mod/page/view.php?id=1030492>
3. <https://esystem.rudn.ru/mod/resource/view.php?id=1030495>
4. <https://esystem.rudn.ru/mod/resource/view.php?id=1030496>
5. [https://github.com/evdvorkina/study\\_2022-2023\\_arh-pc/tree/master](https://github.com/evdvorkina/study_2022-2023_arh-pc/tree/master)