

# Список литературы

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: Резвых Никита

Группа: НКАбд-04-24

МОСКВА

2024 г.

# Список литературы

## 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

# **Список литературы**

## **1 Цель работы**

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

## **Список литературы**

### **2 Задание**

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

# Список литературы

## 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.

## Список литературы

<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> <code>--no-ff</code> 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.

# Список литературы

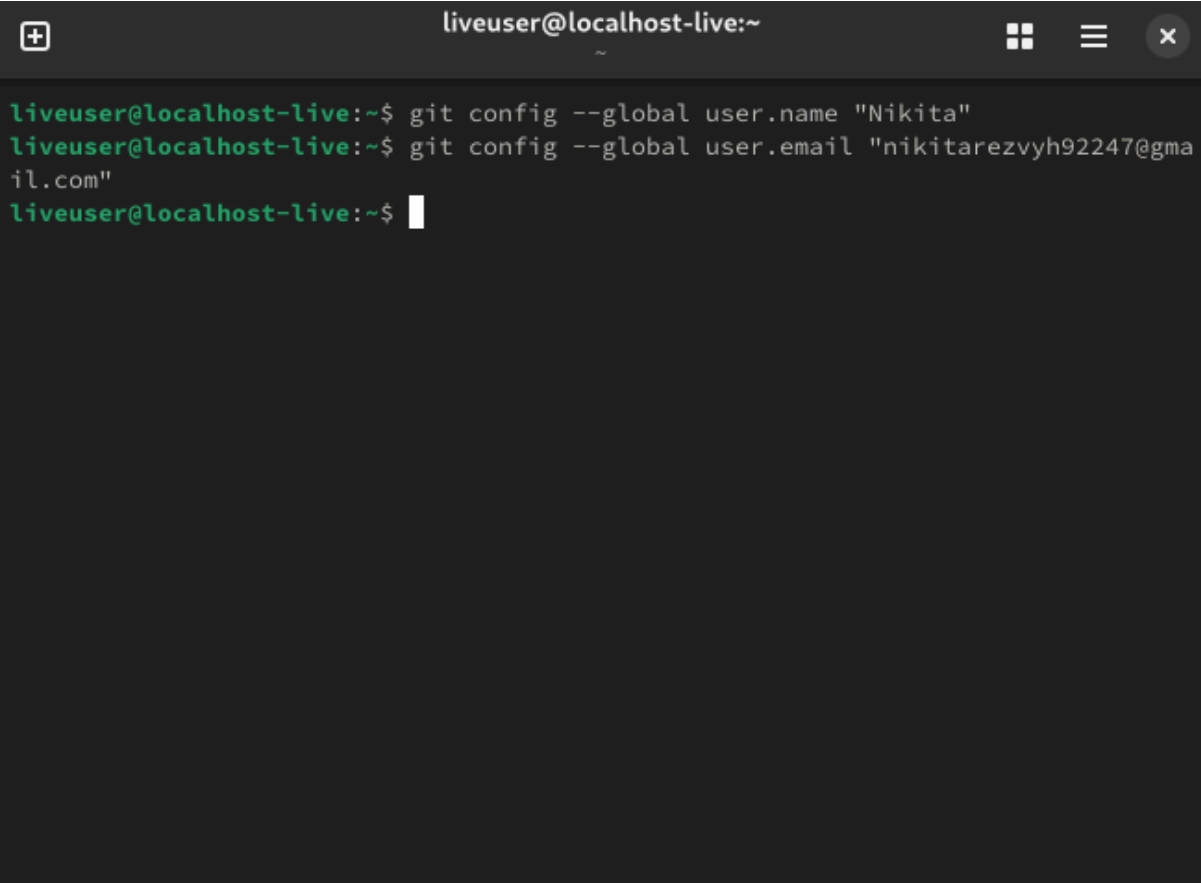
## 4 Выполнение лабораторной работы

### 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)

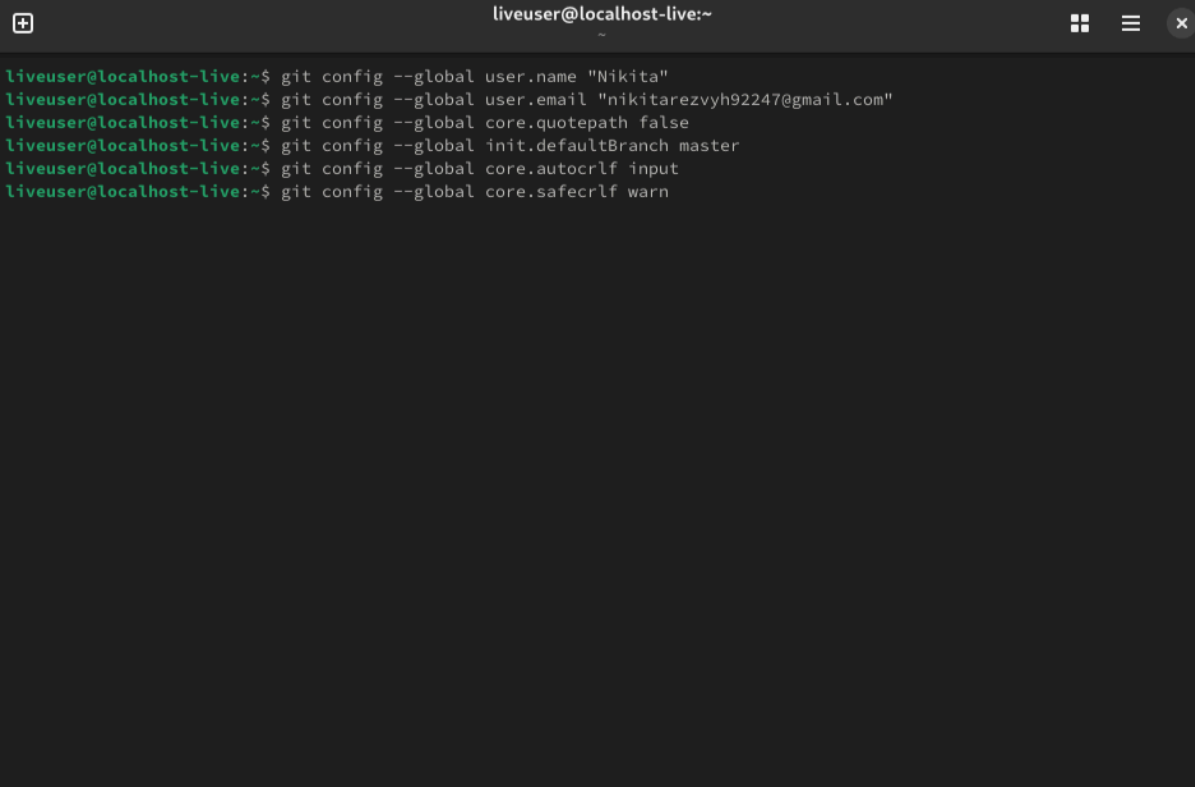
A screenshot of a terminal window titled "liveuser@localhost-live:~". The terminal shows three lines of commands being entered at the prompt "liveuser@localhost-live:~\$". The first command is "git config --global user.name \"Nikita\"", the second is "git config --global user.email \"nikitarezvyh92247@gmail.com\"", and the third is a blank line with a cursor. The terminal has a dark background and green text for the prompt and commands. The window has standard Linux window controls (minimize, maximize, close) in the top right corner.

```
liveuser@localhost-live:~$ git config --global user.name "Nikita"
liveuser@localhost-live:~$ git config --global user.email "nikitarezvyh92247@gmail.com"
liveuser@localhost-live:~$
```

Рис. 4.2.1 Предварительная конфигурация Git.

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

## Список литературы

A terminal window titled 'liveuser@localhost-live:~' with standard window controls. It displays a series of five 'git config' commands being executed in a shell. The commands are: 'git config --global user.name "Nikita"', 'git config --global user.email "nikitarezvyh92247@gmail.com"', 'git config --global core.quotepath false', 'git config --global init.defaultBranch master', and 'git config --global core.autocrlf input'. The final command 'git config --global core.safecrlf warn' is partially visible at the bottom.

```
liveuser@localhost-live:~$ git config --global user.name "Nikita"
liveuser@localhost-live:~$ git config --global user.email "nikitarezvyh92247@gmail.com"
liveuser@localhost-live:~$ git config --global core.quotepath false
liveuser@localhost-live:~$ git config --global init.defaultBranch master
liveuser@localhost-live:~$ git config --global core.autocrlf input
liveuser@localhost-live:~$ git config --global core.safecrlf warn
```

Рис. 4.2.2 Настройка параметров Git.

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



## Список литературы

```
liveuser@localhost-live:~$ ssh-keygen -C "nikitarezvyh92247@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/liveuser/.ssh/id_ed25519):
Created directory '/home/liveuser/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/liveuser/.ssh/id_ed25519
Your public key has been saved in /home/liveuser/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:s/q21JRFsZXGgxKRvk6ioUss5tRB0qBqVbzgD8288Fo nikitarezvyh92247@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|      o+o+... |
|      .   o..o= |
|     ..o   . .o. |
|    ...o*   . .o |
|   . o=.   S o.  |
|   . .+=.o . =o  |
|  ..+ +E oo+.   |
| .+ o+ .o. .    |
|  ....oo.       |
+-----[SHA256]-----+
liveuser@localhost-live:~$
```

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

fedora

Key type

Authentication Key ↕

Key

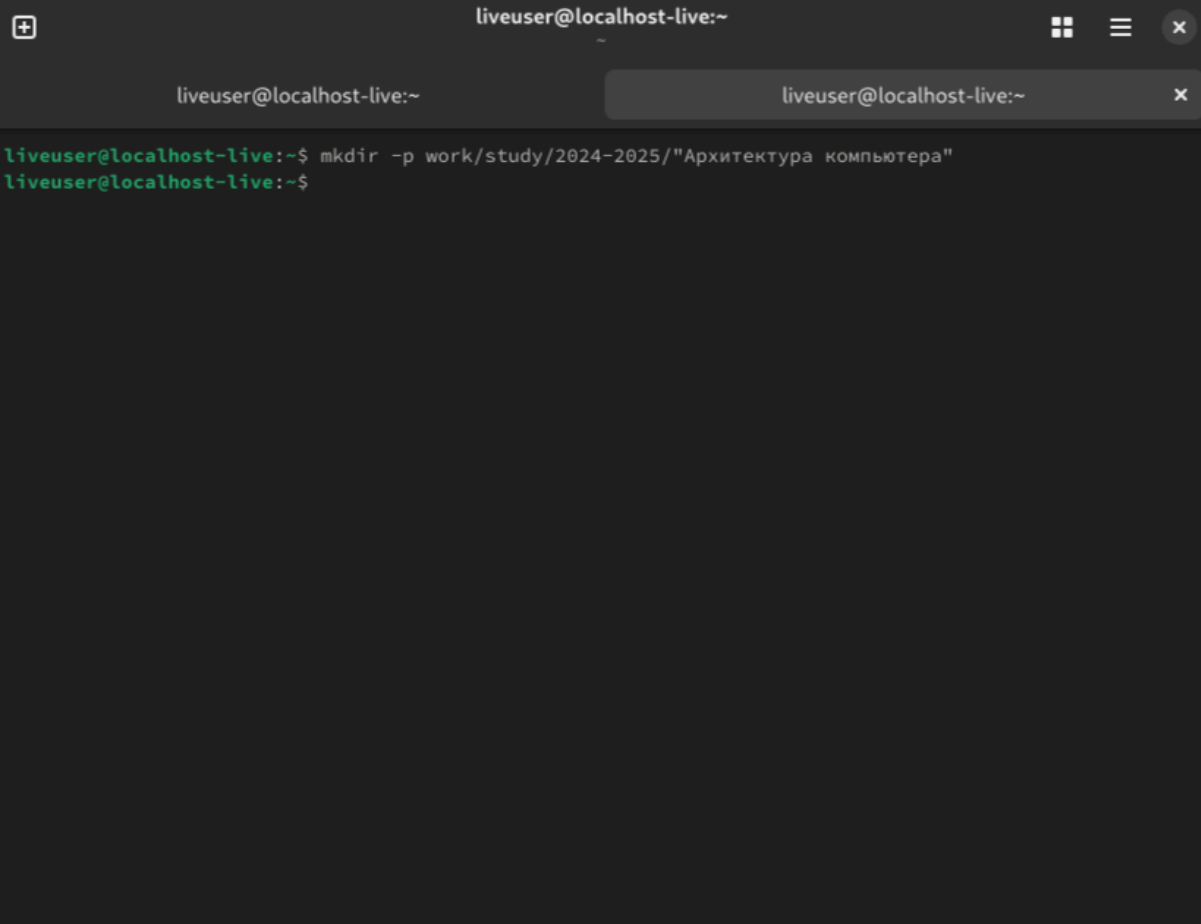
```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIMZQJbiYOhWdBmuHecQTAWES2MGn4JRznF4F5rAjA2o
nikitarezvyh92247@gmail.com
```

Add SSH key

## Список литературы

### 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)



```
liveuser@localhost-live:~  
liveuser@localhost-live:~  
liveuser@localhost-live:~$ mkdir -p work/study/2024-2025/"Архитектура компьютера"  
liveuser@localhost-live:~$
```

### 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)

# Список литературы

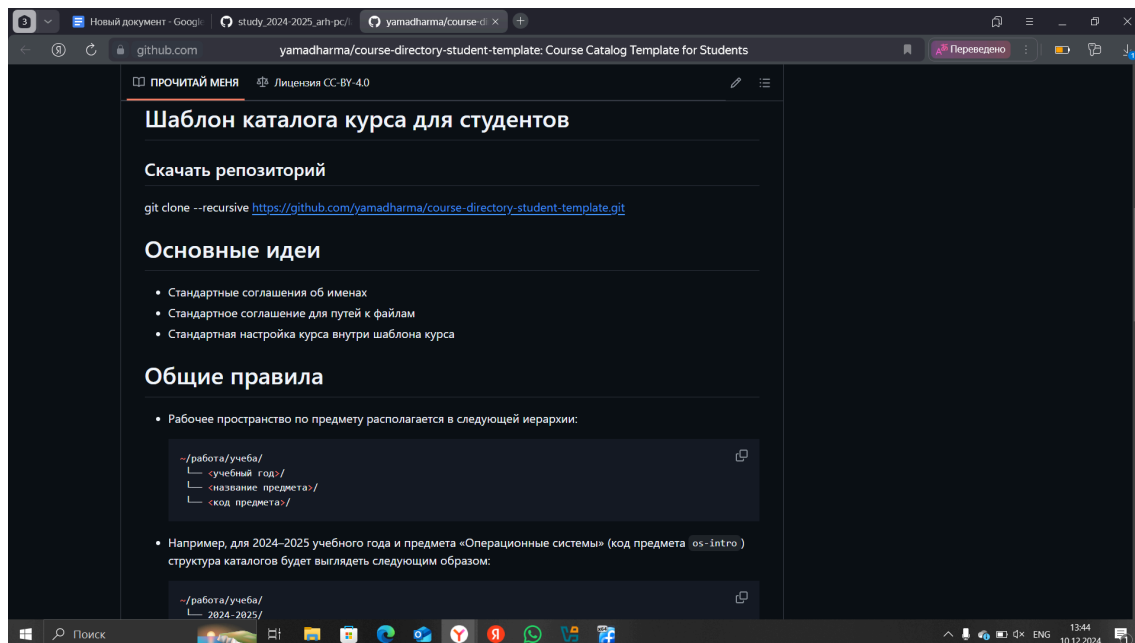


Рис. 4.4.1 Шаблон для клонирования на платформе GitHub.

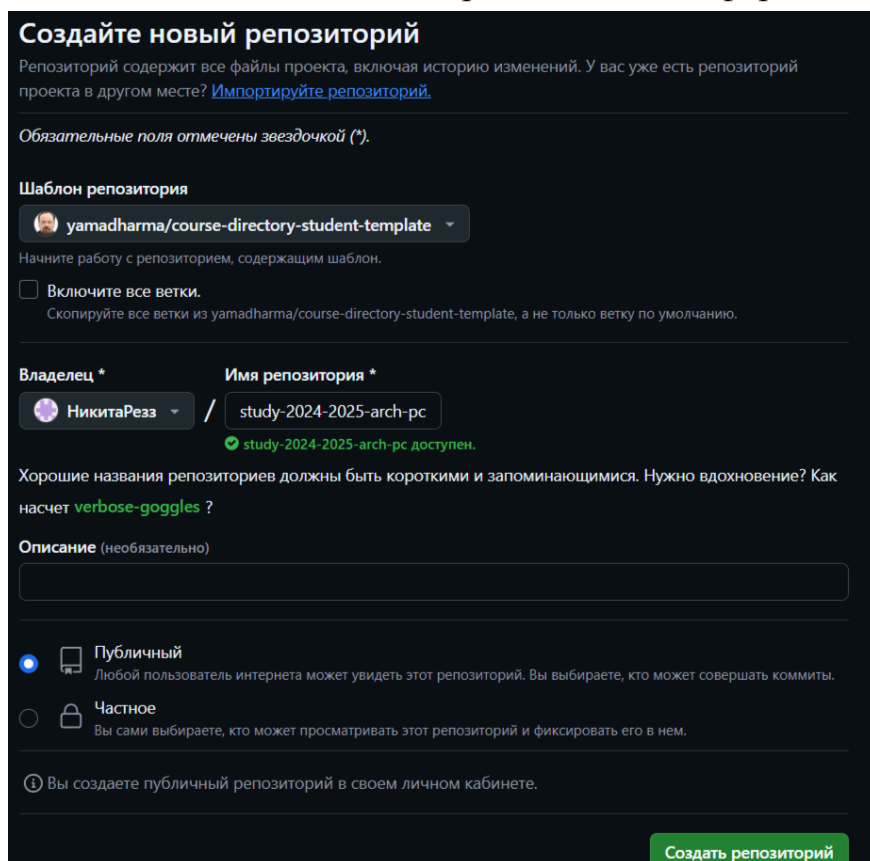


Рис. 4.4.2 Процесс клонирования репозитория через интерфейс GitHub.

# Список литературы

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)

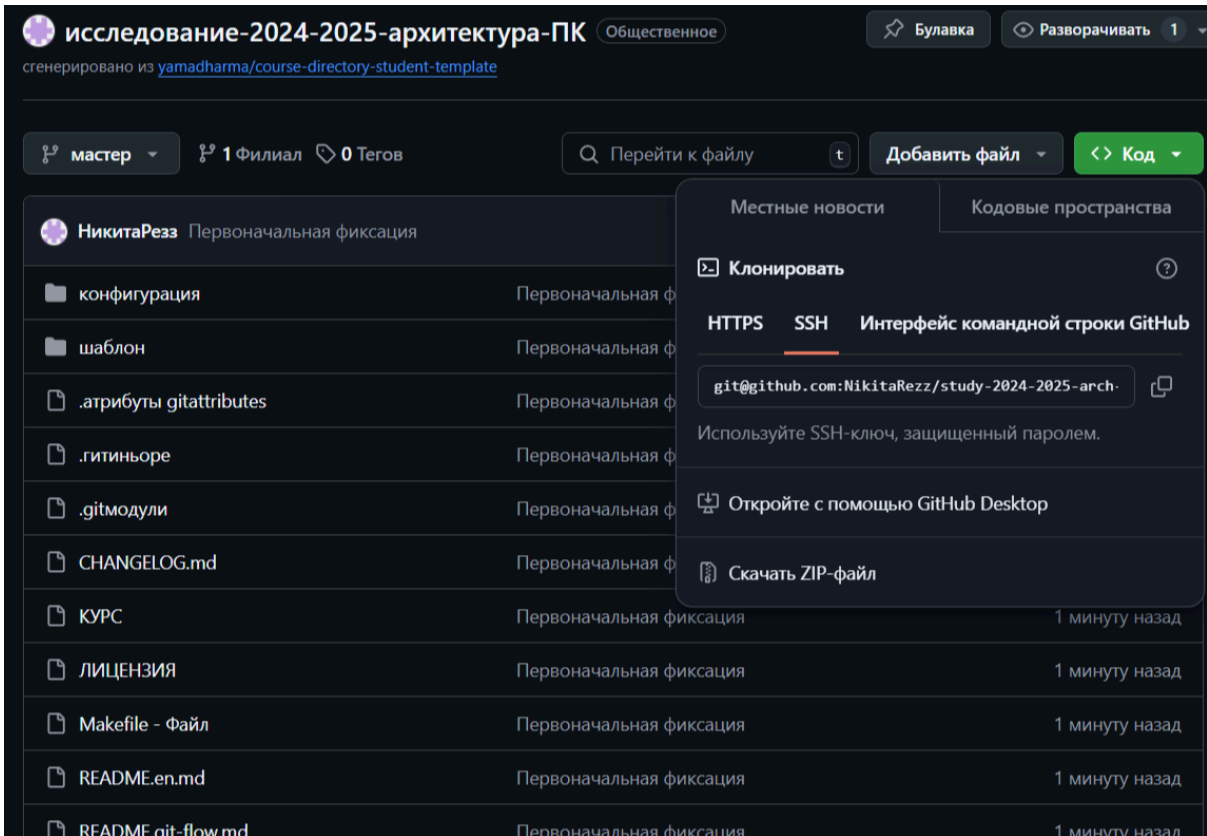


Рис. 4.4.3 Копирование ссылки для последующей вставки в терминал.

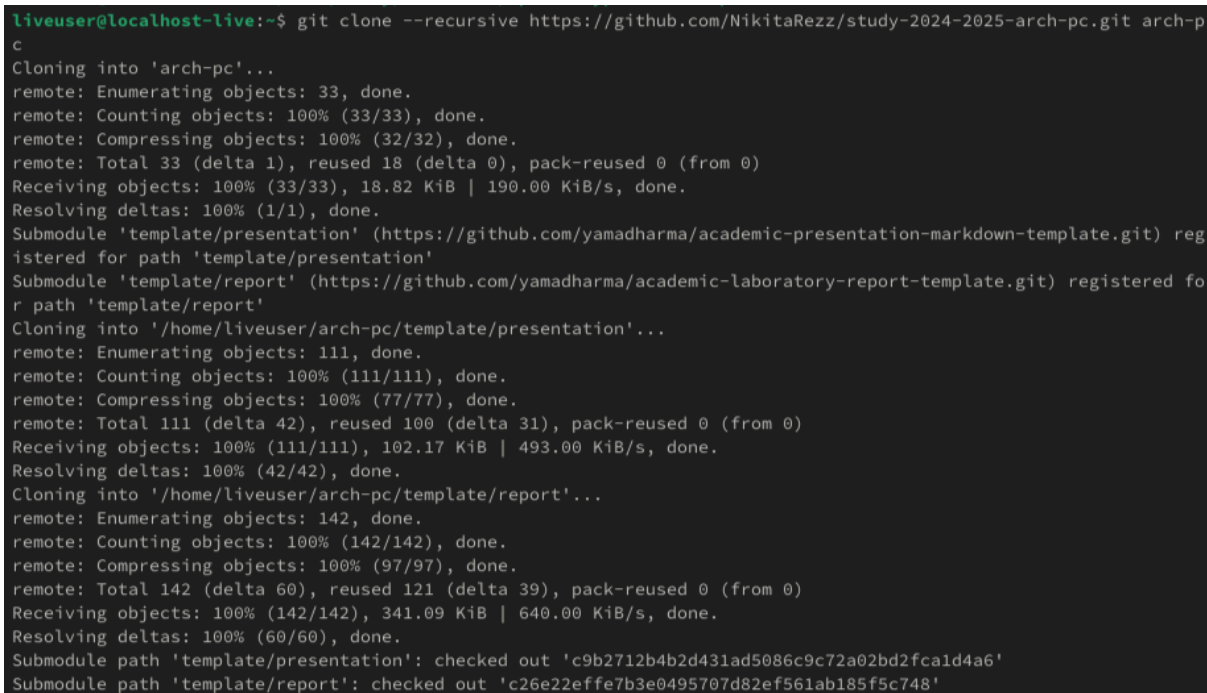


Рис. 4.4.4 Копирование репозитория на рабочий компьютер.

# Список литературы

## 4.5 Настройка каталога курса.

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

```
liveuser@localhost-live:~$ rm arch-pc/package.json
liveuser@localhost-live:~$ echo arch-pc > COURSE
```

Рис. 4.3.1 Настройка каталога курса.

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

```
liveuser@localhost-live:~/1/study-2024-2025-arch-pc$ git add .
liveuser@localhost-live:~/1/study-2024-2025-arch-pc$ git commit "11"
error: pathspec '11' did not match any file(s) known to git
liveuser@localhost-live:~/1/study-2024-2025-arch-pc$ git commit -m "1"
[master 0d17af9] 1
1 file changed, 14 deletions(-)
delete mode 100644 package.json
liveuser@localhost-live:~/1/study-2024-2025-arch-pc$ git pull
Already up to date.
liveuser@localhost-live:~/1/study-2024-2025-arch-pc$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 220 bytes | 220.00 KiB/s, done.
Total 2 (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:NikitaRezz/study-2024-2025-arch-pc.git
13398cb..0d17af9 master -> master
liveuser@localhost-live:~/1/study-2024-2025-arch-pc$ █
```

Рис 4.3.2 Отправка изменений на удаленный репозиторий.

# Список литературы

## 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)

```
liveuser@localhost-live:~/1/study-2024-2025-arch-pc$ git add .
liveuser@localhost-live:~/1/study-2024-2025-arch-pc$ git commit "11"
error: pathspec '11' did not match any file(s) known to git
liveuser@localhost-live:~/1/study-2024-2025-arch-pc$ git commit -m "1"
[master 0d17af9] 1
1 file changed, 14 deletions(-)
delete mode 100644 package.json
liveuser@localhost-live:~/1/study-2024-2025-arch-pc$ git pull
Already up to date.
liveuser@localhost-live:~/1/study-2024-2025-arch-pc$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 220 bytes | 220.00 KiB/s, done.
Total 2 (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:NikitaRezz/study-2024-2025-arch-pc.git
13398cb..0d17af9 master -> master
liveuser@localhost-live:~/1/study-2024-2025-arch-pc$
```

Рис 5.1 Отправка изменений на удаленный репозиторий

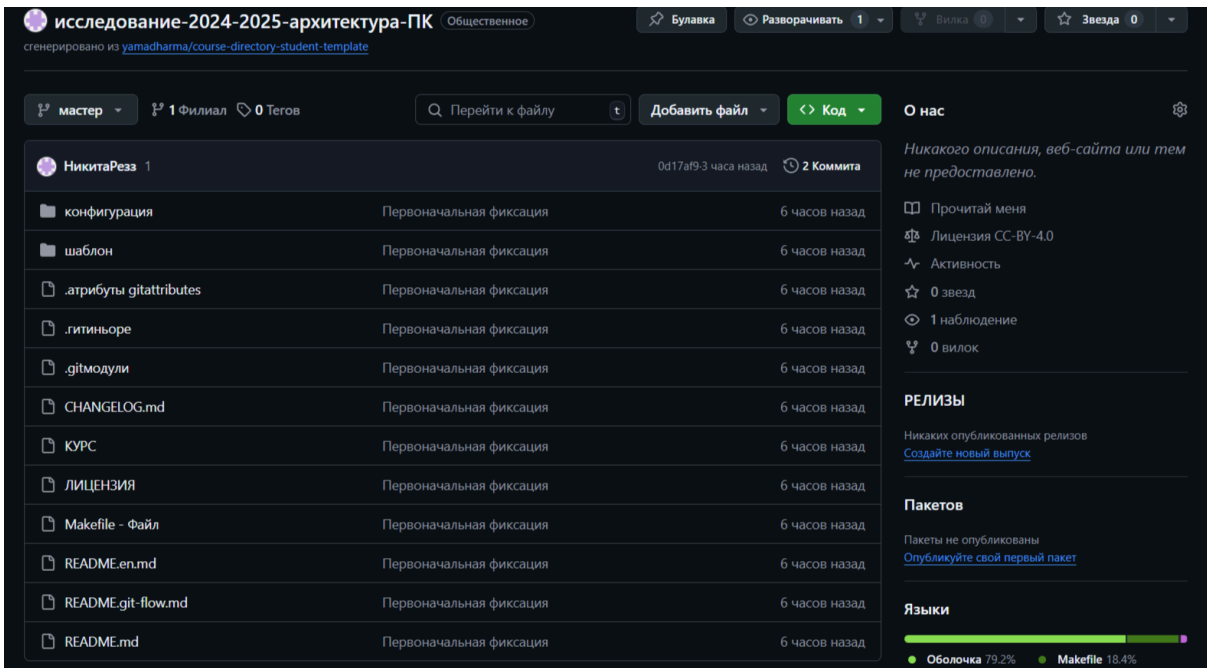


Рис 5.2 Проверка выполненных изменений на GitHub.

## **Список литературы**

## **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)