

# **Лабораторная работа №2**

**Система контроля версий Git**

Калашникова Дарья Викторовна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>6</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>3</b>	<b>Задание для самостоятельной работы</b>	<b>17</b>
<b>4</b>	<b>Выводы</b>	<b>21</b>

# Список иллюстраций

2.1	Настройка имени и адреса эл. Почты. . . . .	7
2.2	Настройка UTF8 в выводе сообщений Git. . . . .	7
2.3	Конфигурация имени начальной ветки. . . . .	7
2.4	Настройка автоконвертации окончаний строк. . . . .	7
2.5	Настройка вывода предупреждений о необратимых преобразованиях из CRLF. . . . .	8
2.6	Генерация ssh ключа. . . . .	8
2.7	Перейдём в пункт settings. . . . .	9
2.8	Раздел настроек “SSH ang GPG keys”. . . . .	9
2.9	Поле ввода ключа в окне добавления нового SSH ключа. . . . .	10
2.10	Считывание и копирование публичного ключа. Команда “cat” прочитает данные из файла id_rsa.pub, а команда xclip выгрузит их в буфер обмена. Остается лишь вставить содержимое буфера обмена, то есть наш ключ, в предложенное поле “Key” и указать имя ключа в поле “Title”. . . . .	10
2.11	Заполнение всех необходимых полей для добавления ключа. . . . .	10
2.12	Создание каталога “Архитектура компьютера”. . . . .	11
2.13	Страница шаблона на GitHub. . . . .	11
2.14	Создание репозитория на основе шаблона. . . . .	12
2.15	Перемещение в папку, куда скачается репозиторий. . . . .	12
2.16	Ссылка на наш репозиторий. . . . .	13
2.17	Скачивание репозитория командой “git clone”. . . . .	13
2.18	Переход в клонированный каталог. . . . .	14
2.19	Удаление файла. . . . .	14
2.20	Создание файла и запись в него строки. . . . .	14
2.21	Добавление каталога для отправки на сервер. . . . .	14
2.22	Сохранение изменений и указание комментария. . . . .	15
2.23	Отправка репозитория на сервер GitHub. . . . .	15
2.24	Файлы на компьютере. . . . .	15
2.25	Файлы на сервере. . . . .	16
3.1	Наличие отчёта в папке. . . . .	17
3.2	Копирование отчёта по первой лабораторной работе в папку lab01. . . . .	17
3.3	Добавление каталога для отправки на сервер. . . . .	18
3.4	Сохранение изменений и указание комментария. . . . .	18
3.5	Загрузка новых файлов на GitHub. . . . .	18
3.6	Вид обновлённого репозитория на GitHub. . . . .	19

3.7	Наличие отчетов на GitHub. . . . .	19
3.8	Наличие отчетов на GitHub. . . . .	20

## **Список таблиц**

# 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

Перед началом работы с git его необходимо предварительно настроить. Для этого введем в терминал следующие команды: (рис. 2.1).

```
d_kalashnikova@fedora ~]$ git config --global user.name "Kalashnikova Darya"  
d_kalashnikova@fedora ~]$ git config --global user.email "oikawka_tooru@vk.com"  
d_kalashnikova@fedora ~]$
```

Рис. 2.1: Настройка имени и адреса эл. Почты.

Введя их, мы задали имя и электронный адрес почты пользователя. Теперь введем следующую команду: (рис. 2.2).

```
[d_kalashnikova@fedora ~]$  
[d_kalashnikova@fedora ~]$ git config --global core.quotepath false  
[d_kalashnikova@fedora ~]$
```

Рис. 2.2: Настройка UTF8 в выводе сообщений Git.

Благодаря ней, мы настроили вывод сообщений git в кодировке utf8. Теперь мы должны задать имя для (рис. 2.3). начальной ветки. Мы назовем ее master:

```
[d_kalashnikova@fedora ~]$  
[d_kalashnikova@fedora ~]$ git config --global core.quotepath false  
[d_kalashnikova@fedora ~]$
```

Рис. 2.3: Конфигурация имени начальной ветки.

Теперь введем следующую команду: (рис. 2.4).

```
[d_kalashnikova@fedora ~]$ git config --global core.autocrlf input  
[d_kalashnikova@fedora ~]$
```

Рис. 2.4: Настройка автоконвертации окончаний строк.

Таким образом мы указываем git автоматически конвертировать CRLF окончания строк в LF во время commit'a. Теперь выполним вот эту команду (рис. 5). Она

будет отвечать за то, чтобы печатать предупреждение в случае, если преобразования из CRLF будут необратимыми. (рис. 2.5).

```
[d_kalashnikova@fedora ~]$ git config --global core.safecrlf warn
[d_kalashnikova@fedora ~]$
```

Рис. 2.5: Настройка вывода предупреждений о необратимых преобразованиях из CRLF.

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. 2.6).

```
[d_kalashnikova@fedora ~]$ ssh-keygen -C "Kalashnikova Darya oikawka_tooru@vk.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/d_kalashnikova/.ssh/id_rsa):
Created directory '/home/d_kalashnikova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/d_kalashnikova/.ssh/id_rsa
Your public key has been saved in /home/d_kalashnikova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:/gwysVgUGbIRDmCcDsKn4FrYg0okhVshiliDdzaiwRU Kalashnikova Darya oikawka_tooru@vk.com
The key's randomart image is:
+---[RSA 3072]-----+
|*OE=.oo          |
|&O+=+...         |
|Xo+o..           |
|==+ .            |
|+. . o S         |
|o o +            |
| . + o           |
| o +             |
| o               |
+-----[SHA256]-----+
[d_kalashnikova@fedora ~]$
```

Рис. 2.6: Генерация ssh ключа.

Зайдем на сайт GitHub: (рис. 2.7).



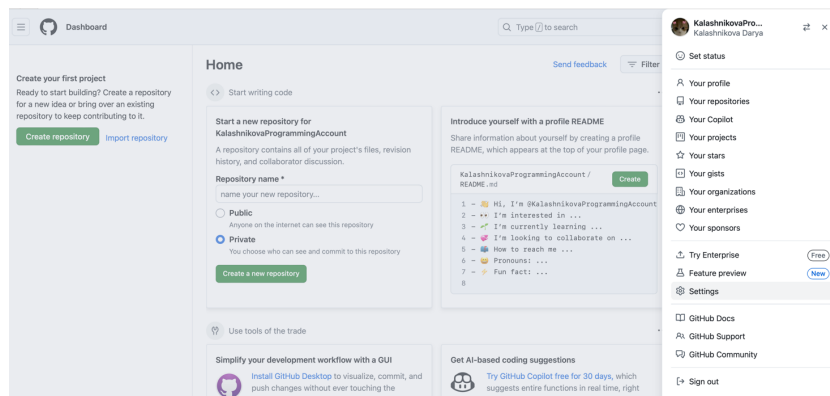


Рис. 2.7: Перейдём в пункт settings.

Находим раздел SSH and GPG keys и нажимаем New SSH key: (рис. 2.8).

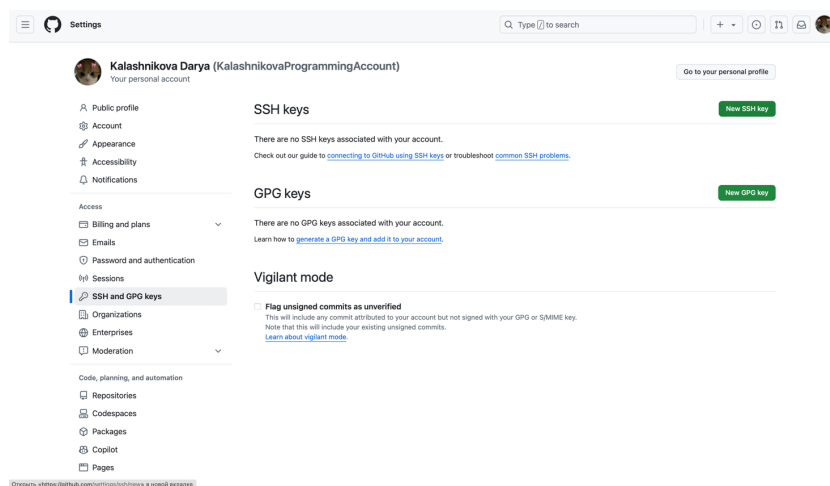


Рис. 2.8: Раздел настроек “SSH ang GPG keys”.

В предложенное поле “Key” нам необходимо вставить ключ, который мы только что сгенерировали: (рис. 2.9).

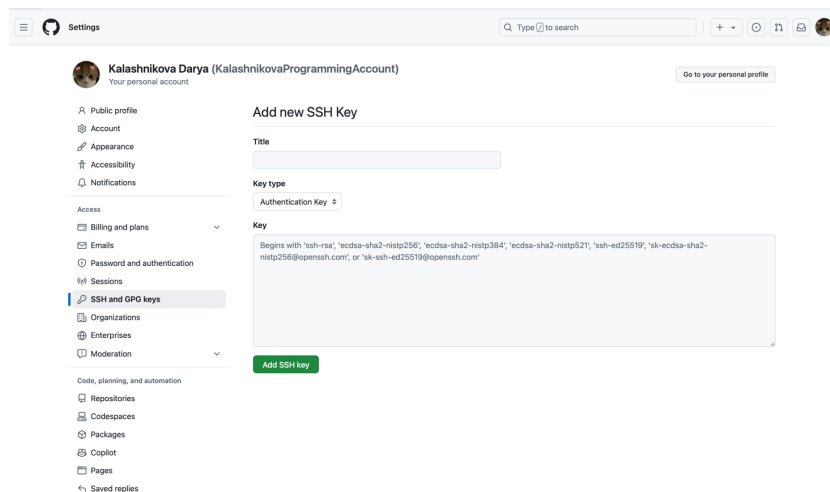


Рис. 2.9: Поле ввода ключа в окне добавления нового SSH ключа.

Для того, чтобы скопировать ключ для последующей вставки, нам необходимо ввести следующую команду: (рис. 2.10).(рис. 2.11).

```
[d_kalashnikova@fedora ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
[d_kalashnikova@fedora ~]$
```

Рис. 2.10: Считывание и копирование публичного ключа. Команда “cat” прочитает данные из файла id\_rsa.pub, а команда xclip выгрузит их в буфер обмена. Остается лишь вставить содержимое буфера обмена, то есть наш ключ, в предложенное поле “Key” и указать имя ключа в поле “Title”.

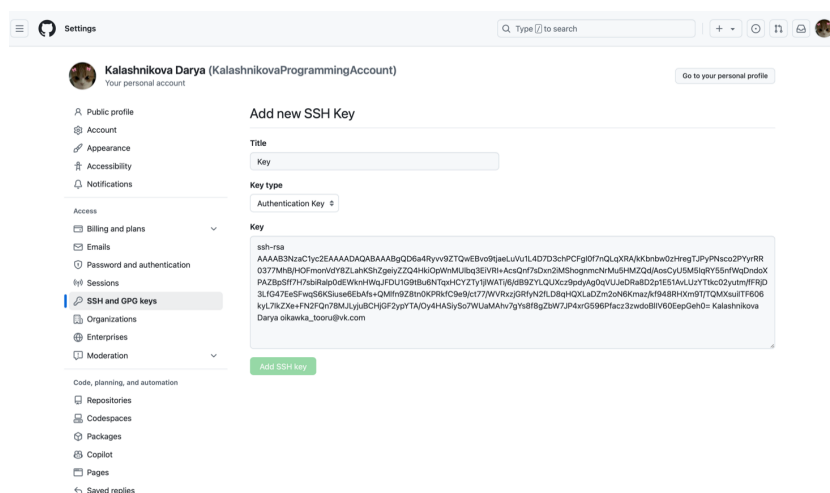


Рис. 2.11: Заполнение всех необходимых полей для добавления ключа.

Теперь нам необходимо организовать наше рабочее пространство. Для этого создадим каталог “Архитектура компьютера” по следующему адресу: (рис. 2.12).

```
[d_kalashnikova@fedora ~]$ mkdir -p ~/work/study/2024-2025/"Архитектура компьютера"
[d_kalashnikova@fedora ~]$
```

Рис. 2.12: Создание каталога “Архитектура компьютера”.

После этого нам нужно будет создать репозиторий. Мы будем его создавать на основе шаблона, который находится по следующему адресу: <https://github.com/yamadharma/course-directory-student-template>. (рис. 2.13).

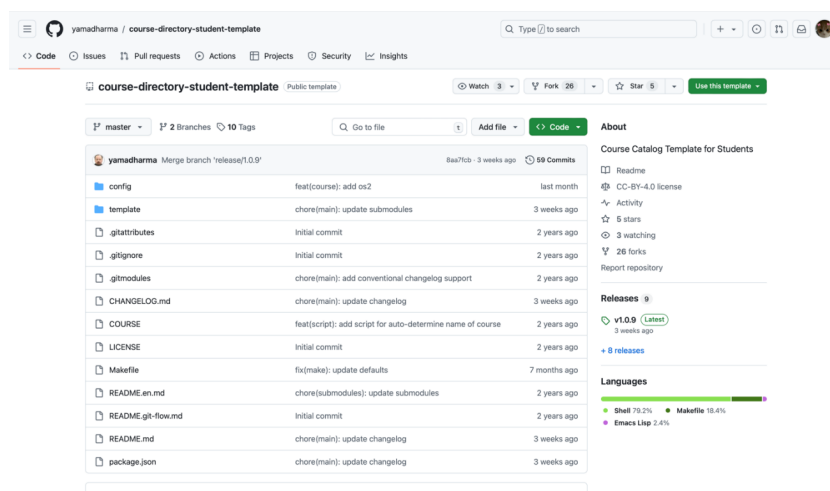


Рис. 2.13: Страница шаблона на GitHub.

Нажимаем на кнопку “Use this template”, и в предложенных опциях выбираем “Create new repository”. Далее нас переносит на следующую страницу (рис. 15). Здесь задаём имя нашего репозитория. Он будет называться так: study\_2024-2025\_arh-pc: (рис. 2.14).

**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**  
yamadharma/course-directory-student-template  
Start your repository with a template repository's contents.  
☐ Include all branches  
Copy all branches from yamadharma/course-directory-student-template and not just the default branch.

**Owner \*** KalashnikovaProgrammingAccount **Repository name \*** study\_2024-2025\_arh-pc  
Your new repository will be created as study\_2024-2025\_arh-pc-. The repository name can only contain ASCII letters, digits, and the characters -, ., and \_.

Great repository names are short and memorable. Need inspiration? How about [congenial-waffle](#)?

**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. 2.14: Создание репозитория на основе шаблона.

Теперь нам нужно клонировать репозиторий на наш компьютер. Для этого перейдем в папку, в которую мы хотим скопировать репозиторий. В нашем случае это заранее созданная папка:(рис. 2.15).

```
[d_kalashnikova@fedora ~]$ cd ~/work/study/2024-2025/"Архитектура компьютера"
[d_kalashnikova@fedora Архитектура компьютера]$
```

Рис. 2.15: Перемещение в папку, куда скачается репозиторий.

Теперь перейдем непосредственно к клонированию. Для этого воспользуемся командой `git clone`, в аргументе указав ссылку на репозиторий. Ссылку можно найти при нажатии на кнопку код на странице нашего репозитория. (рис. 2.16).(рис. 2.17).

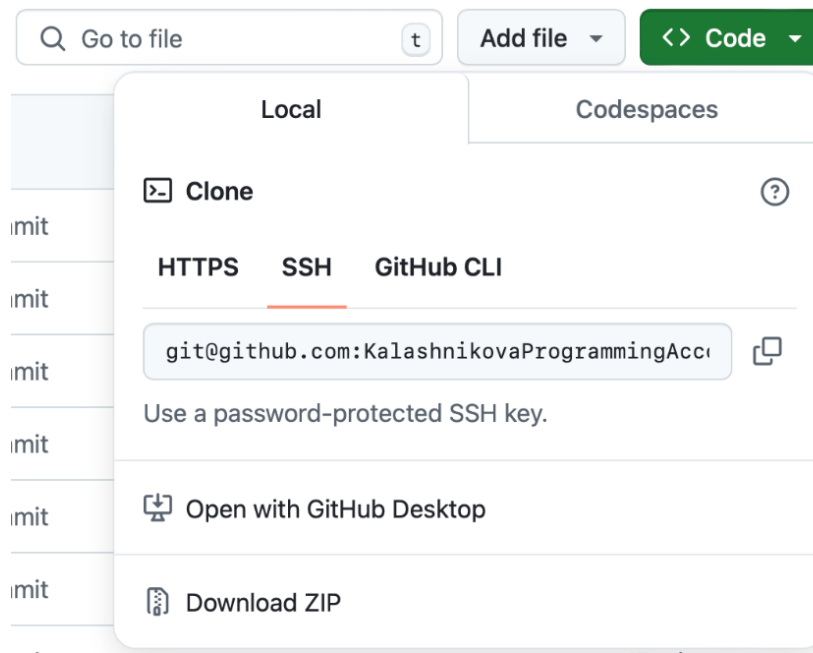


Рис. 2.16: Ссылка на наш репозиторий.

```
[d_kalashnikova@fedora Архитектура компьютера]$ git clone --recursive git@github.com:KalashnikovaProgrammingAccount/study_2024-2025_arh--pc-.git
Клонирование в «study_2024-2025_arh--pc-»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TujZisF/zLDA0zPMSvHdK4UvCoQU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 33 (delta 1), reused 18 (delta 0), pack-reused 0 (from 0)
Получение объектов: 100% (33/33), 18.82 КиБ | 3.76 МиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/d_kalashnikova/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh--pc-/template/presentation»...
remote: Enumerating objects: 111, done.
remote: Counting objects: 100% (111/111), done.
remote: Compressing objects: 100% (77/77), done.
remote: Total 111 (delta 42), reused 100 (delta 31), pack-reused 0 (from 0)
Получение объектов: 100% (111/111), 102.17 КиБ | 597.00 КиБ/с, готово.
Определение изменений: 100% (42/42), готово.
Клонирование в «/home/d_kalashnikova/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh--pc-/template/report»...
remote: Enumerating objects: 142, done.
remote: Counting objects: 100% (142/142), done.
remote: Compressing objects: 100% (97/97), done.
remote: Total 142 (delta 60), reused 121 (delta 39), pack-reused 0 (from 0)
Получение объектов: 100% (142/142), 341.09 КиБ | 598.00 КиБ/с, готово.
Определение изменений: 100% (60/60), готово.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
```

Рис. 2.17: Скачивание репозитория командой “git clone”.

Теперь перейдем к настройке клонированного каталога. Для начала перейдем в него с помощью команды “cd”: (рис. 2.18).

```
[d_kalashnikova@fedora Архитектура компьютера]$ cd ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/  
[d_kalashnikova@fedora arch-pc]$
```

Рис. 2.18: Переход в клонированный каталог.

Удалим с помощью команды “rm” лишний файл: (рис. 2.19).

```
[d_kalashnikova@fedora arch-pc]$ rm package.json  
[d_kalashnikova@fedora arch-pc]$
```

Рис. 2.19: Удаление файла.

Теперь создадим необходимые файлы. В нашем случае, это будет файл COURSE. Чтобы его создать, мы воспользуемся командой echo, которая запишет в файл строку “arch-pc” и автоматически создаст его, т.к. этого файла раньше не существовало: (рис. 2.20).

```
[d_kalashnikova@fedora arch-pc]$ echo arch-pc > COURSE  
[d_kalashnikova@fedora arch-pc]$ make  
Usage:  
  make <target>  
  
Targets:  
  list           List of courses  
  prepare        Generate directories structure  
  submodule      Update submodules
```

Рис. 2.20: Создание файла и запись в него строки.

Теперь нам остается лишь отправить файлы на сервер. Для этого с помощью команды git add мы добавим каталоги, которые должны отправляться на сервер. В качестве аргумента мы возьмем точку, которая укажет на то, что мы должны отправить на сервер все файлы и каталоги, которые по иерархии находятся ниже нашего текущего расположения: (рис. 2.21).

```
[d_kalashnikova@fedora arch-pc]$ git add .
```

Рис. 2.21: 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. 2.22).

```
[d_kalashnikova@fedora arch-pc]$ git commit -am 'feat(main): make course structure'
[master b315566] feat(main): make course structure
221 files changed, 53680 insertions(+)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/.projectile
create mode 100644 labs/lab01/presentation/.texlabroot
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
```

Рис. 2.22: Сохранение изменений и указание комментария.

Теперь нам осталось окончательно загрузить изменения на сервер. Для этого мы воспользуемся командой `git push`: (рис. 2.23).

```
[d_kalashnikova@fedora arch-pc]$ git push
Перечисление объектов: 36, готово.
Подсчет объектов: 100% (36/36), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 341.27 Киб | 2.47 Миб/с, готово.
Всего 35 (изменений 4), повторно использовано 1 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:KalashnikovaProgrammingAccount/study_2024-2025_arh--pc-.git
1373cae..b315566 master -> master
```

Рис. 2.23: Отправка репозитория на сервер GitHub.

Остается лишь проверить, сохранились ли файлы на сервере. Сравним файлы, которые находятся на GitHub с теми файлами, что находятся на нашем компьютере: (рис. 2.24).(рис. 2.25).

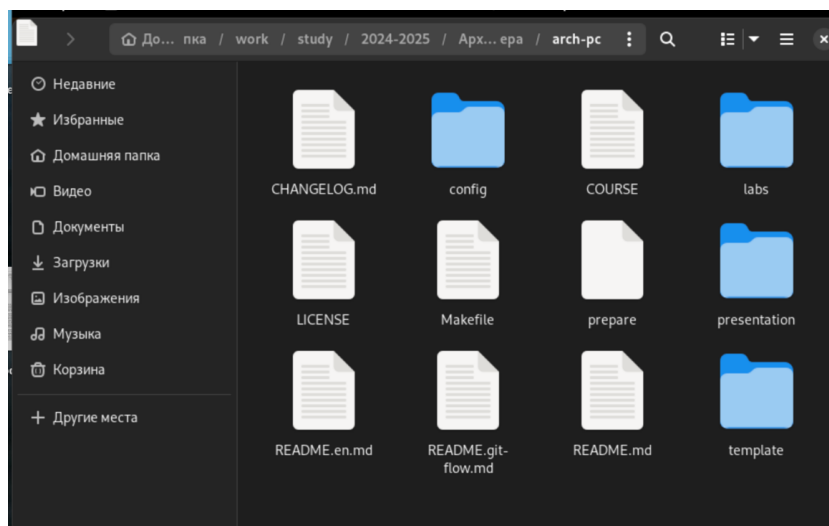


Рис. 2.24: Файлы на компьютере.


 <b>KalashnikovaProgrammingAccount</b> feat(main): make course structure <span>b315566 · 8 minutes ago</span> <span>🕒 3 Commits</span>		
📁 config	Initial commit	12 hours ago
📁 labs	feat(main): make course structure	8 minutes ago
📁 presentation	feat(main): make course structure	8 minutes ago
📁 template	Initial commit	12 hours ago
📄 .gitattributes	Initial commit	12 hours ago
📄 .gitignore	Initial commit	12 hours ago
📄 .gitmodules	Initial commit	12 hours ago
📄 CHANGELOG.md	Initial commit	12 hours ago
📄 COURSE	feat(main): make course structure	11 hours ago
📄 LICENSE	Initial commit	12 hours ago
📄 Makefile	Initial commit	12 hours ago
📄 README.en.md	Initial commit	12 hours ago
📄 README.git-flow.md	Initial commit	12 hours ago
📄 README.md	Initial commit	12 hours ago
📄 prepare	feat(main): make course structure	8 minutes ago

Рис. 2.25: Файлы на сервере.

Всё совпало.



### 3 Задание для самостоятельной работы

Задание №1. 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. 3.1).

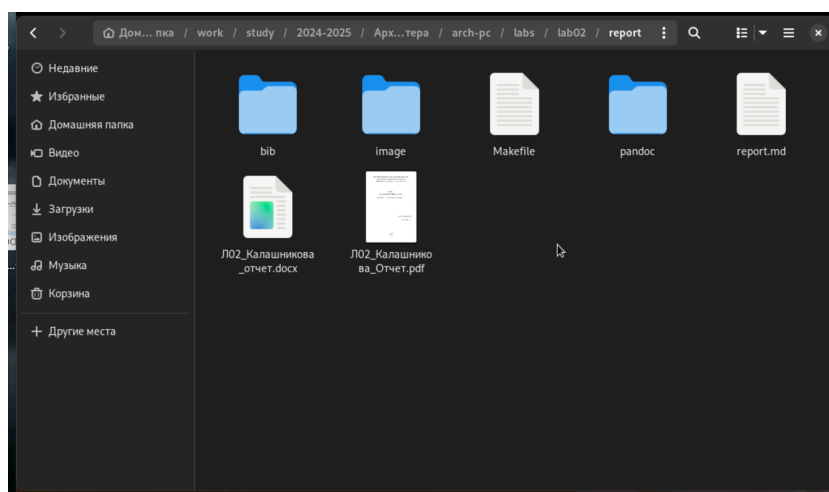


Рис. 3.1: Наличие отчёта в папке.

Задание №2. Копируем 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. 3.2).

```
[d_kalashnikova@fedora arch-pc]$ cp ~/Документы/Л01_Калашникова_Отчет.pdf labs/lab01/report/  
[d_kalashnikova@fedora arch-pc]$ cp ~/Документы/Л01_Калашникова_Отчет.docx labs/lab01/report/  
[d_kalashnikova@fedora arch-pc]$
```

Рис. 3.2: Копирование отчёта по первой лабораторной работе в папку lab01.

Задание №3. Теперь нам осталось лишь загрузить изменения на GitHub. Для этого нам нужно повторить те шаги, которые мы выполнили в предыдущих пунктах лабораторной работы, а именно, воспользоваться командой “git add.” для того, чтобы указать, что мы хотим сохранить изменения во всех файлах, находящихся в нашем каталоге. (рис. 3.3).

```
[d_kalashnikova@fedora arch-pc]$ git add .  
[d_kalashnikova@fedora arch-pc]$
```

Рис. 3.3: Добавление каталога для отправки на сервер.

После этого с помощью команды “git commit” мы укажем комментарий и сохраним изменения. В комментарии мы укажем, что мы загрузили первую и вторую лабораторные работы. (рис. 3.4).

```
[d_kalashnikova@fedora arch-pc]$ git commit -am "feat(main): added first 2 labs"  
[master 4934002] feat(main): added first 2 labs  
4 files changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 labs/lab01/report/Л01_Калашникова_Отчет.docx  
create mode 100644 labs/lab01/report/Л01_Калашникова_Отчет.pdf  
create mode 100644 labs/lab02/report/Л02_Калашникова_Отчет.pdf  
create mode 100644 labs/lab02/report/Л02_Калашникова_Отчет.docx  
[d_kalashnikova@fedora arch-pc]$
```

Рис. 3.4: Сохранение изменений и указание комментария.

После этого вводим команду “git push” для того, чтобы загрузить наши файлы на GitHub. (рис. 3.5).

```
[d_kalashnikova@fedora arch-pc]$ git push  
Перечисление объектов: 17, готово.  
Подсчет объектов: 100% (15/15), готово.  
При сжатии изменений используется до 2 потоков  
Сжатие объектов: 100% (11/11), готово.  
Запись объектов: 100% (11/11), 5.43 МБ | 1.05 МБ/с, готово.  
Всего 11 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0  
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.  
To github.com:KalashnikovaProgrammingAccount/study_2024-2025_arh--pc-.git  
b315566..4934002 master -> master  
[d_kalashnikova@fedora arch-pc]$
```

Рис. 3.5: Загрузка новых файлов на GitHub.

Остается лишь проверить, правильно ли мы все загрузили. Для этого посмотрим время обновления файлов в папке labs в GitHub: (рис. 3.6).(рис. 3.7).(рис. 3.8).


















 <b>KalashnikovaProgrammingAccount</b>	feat(main): added fi...	4934002 · 50 minutes ago	 4 Commits
 config	Initial commit	5 days ago	
 labs	feat(main): added first 2 labs	50 minutes ago	
 presentation	feat(main): make course structure	5 days ago	
 template	Initial commit	5 days ago	
 .gitattributes	Initial commit	5 days ago	
 .gitignore	Initial commit	5 days ago	
 .gitmodules	Initial commit	5 days ago	
 CHANGELOG.md	Initial commit	5 days ago	
 COURSE	feat(main): make course structure	5 days ago	
 LICENSE	Initial commit	5 days ago	
 Makefile	Initial commit	5 days ago	
 README.en.md	Initial commit	5 days ago	
 README.git-flow.md	Initial commit	5 days ago	
 README.md	Initial commit	5 days ago	
 prepare	feat(main): make course structure	5 days ago	

Рис. 3.6: Вид обновлённого репозитория на GitHub.













study_2024-2025_arh--pc- / labs / lab01 / report 			Add file ▾	⋮
 <b>KalashnikovaProgrammingAccount</b>	feat(main): added first 2 labs	4934002 · 1 hour ago	 History	
Name	Last commit message	Last commit date		
 ..				
 bib	feat(main): make course structure	5 days ago		
 image	feat(main): make course structure	5 days ago		
 pandoc	feat(main): make course structure	5 days ago		
 Makefile	feat(main): make course structure	5 days ago		
 report.md	feat(main): make course structure	5 days ago		
 Л01_Калашникова_Отчет.docx	feat(main): added first 2 labs	1 hour ago		
 Л01_Калашникова_Отчет.pdf	feat(main): added first 2 labs	1 hour ago		

Рис. 3.7: Наличие отчетов на GitHub.

[study\\_2024-2025\\_arh--pc-](#) / [labs](#) / [lab02](#) / [report](#)

Add file
...


**KalashnikovaProgrammingAccount**
feat(main): added first 2 labs
4934002 · 1 hour ago
History

Name	Last commit message	Last commit date
..		
bib	feat(main): make course structure	5 days ago
image	feat(main): make course structure	5 days ago
pandoc	feat(main): make course structure	5 days ago
Makefile	feat(main): make course structure	5 days ago
report.md	feat(main): make course structure	5 days ago
Л02_Калашникова_Отчет.pdf	feat(main): added first 2 labs	1 hour ago
Л02_Калашникова_отчет.docx	feat(main): added first 2 labs	1 hour ago

Рис. 3.8: Наличие отчетов 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.