

Отчёт по лабораторной работе

Лабораторная №2

Полина Витальевна Барабаш

Содержание

<i>1</i>	<i>Цель работы</i>	<i>4</i>
<i>2</i>	<i>Выполнение работы</i>	<i>5</i>
<i>3</i>	<i>Выполнение самостоятельной работы</i>	<i>17</i>
<i>4</i>	<i>Выводы</i>	<i>22</i>

Список иллюстраций

2.1	Созданная учётная запись на github	5
2.2	Предварительная конфигурацию git	6
2.3	Настройка utf-8 в выводе сообщений git	6
2.4	Имя начальной ветки	6
2.5	Параметр autocrlf	6
2.6	Параметр safecrlf	7
2.7	Генерация двух ключей	8
2.8	Копирование в буфер обмена	8
2.9	Загрузка открытого ключа на github	9
2.10	Создание каталога предмета “Архитектура компьютера”	9
2.11	Выбор при нажатии Use this template	10
2.12	Страница Create a new repository	10
2.13	Созданный репозиторий по шаблону	11
2.14	Переход в каталог курса через терминал	11
2.15	Клонирование созданного на github репозитория	12
2.16	Проверка, что репозиторий клонирован	13
2.17	Переход в каталог курса	13
2.18	Удаление файла package.json	14
2.19	Создание необходимых каталогов	14
2.20	Отправление файлов на сервер (Часть I)	14
2.21	Отправление файлов на сервер (Часть II)	15
2.22	Правильность создания иерархии рабочего пространства в локаль- ном репозитории	15
2.23	Правильность создания иерархии рабочего пространства на github	16
3.1	Перемещение файла	17
3.2	Создание файла	18
3.3	Перемещение предыдущего отчёта в нужную папку	19
3.4	Действия в терминале для загрузки файлов на github	20
3.5	Проверка загрузки файлов на github	21

1 Цель работы

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

2 Выполнение работы

Задание №1. Создайте учётную запись на сайте <https://github.com/> и заполните основные данные. Я создала учётную запись на сайте <https://github.com/> и заполнила основные данные.

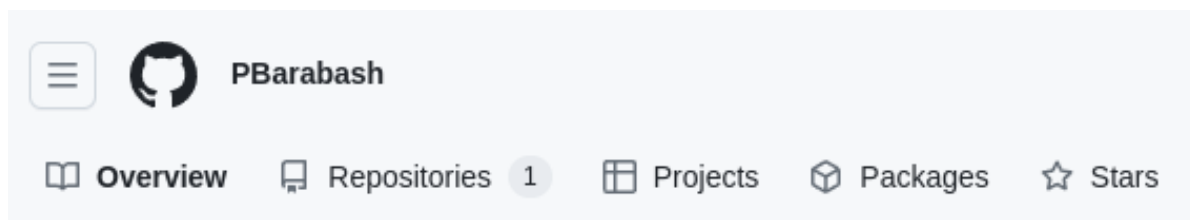


Рис. 2.1: Созданная учётная запись на github

Задание №2. Сделайте предварительную конфигурацию git. Откройте терминал и введите следующие команды, указав имя и email владельца репозитория:
`git config --global user.name ""` `git config --global user.email "work@mail"`

Настройте utf-8 в выводе сообщений git: `git config --global core.quotePath false`

Задайте имя начальной ветки (будем называть её master): `git config --global init.defaultBranch master`

Задайте параметр autocrlf и параметр safecrlf: `git config --global core.autocrlf input`
`git config --global core.safecrlf warn`

Сначала я сделала предварительную конфигурацию git.

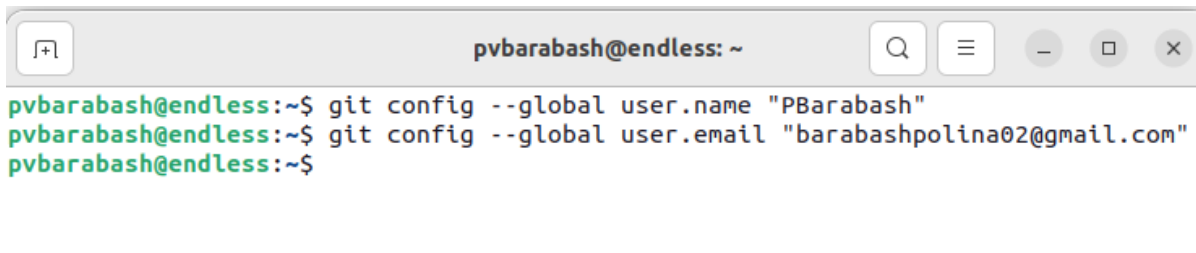
A terminal window titled 'pvbarabash@endless: ~' with search, menu, and window control icons. It shows two lines of green text: 'pvbarabash@endless:~\$ git config --global user.name "PBarabash"' and 'pvbarabash@endless:~\$ git config --global user.email "barabashpolina02@gmail.com"', followed by a prompt 'pvbarabash@endless:~\$'.

Рис. 2.2: Предварительная конфигурацию git

Затем я настроила utf-8 в выводе сообщений git.

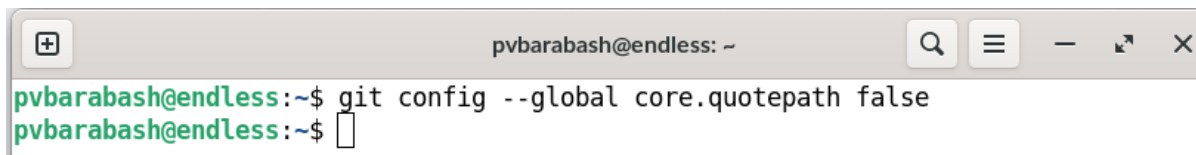
A terminal window titled 'pvbarabash@endless: ~' with search, menu, and window control icons. It shows a line of green text: 'pvbarabash@endless:~\$ git config --global core.quotepath false', followed by a prompt 'pvbarabash@endless:~\$'.

Рис. 2.3: Настройка utf-8 в выводе сообщений git

Я задала имя начальной ветки.

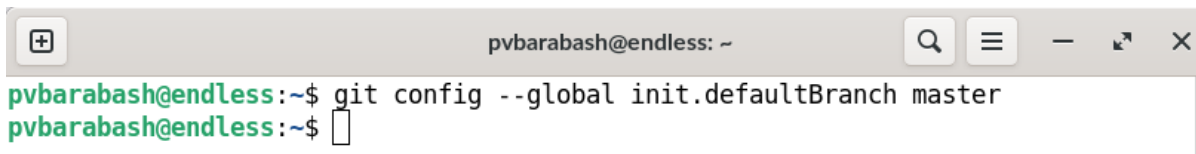
A terminal window titled 'pvbarabash@endless: ~' with search, menu, and window control icons. It shows a line of green text: 'pvbarabash@endless:~\$ git config --global init.defaultBranch master', followed by a prompt 'pvbarabash@endless:~\$'.

Рис. 2.4: Имя начальной ветки

Я задала параметр autocrlf.

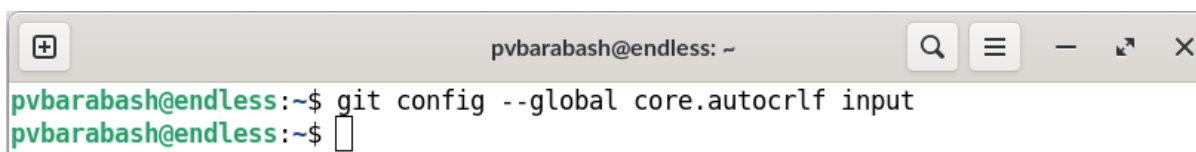
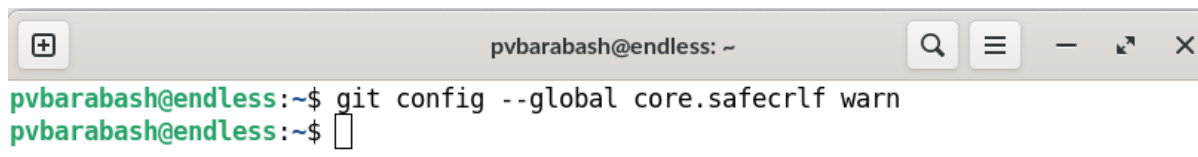
A terminal window titled 'pvbarabash@endless: ~' with search, menu, and window control icons. It shows a line of green text: 'pvbarabash@endless:~\$ git config --global core.autocrlf input', followed by a prompt 'pvbarabash@endless:~\$'.

Рис. 2.5: Параметр autocrlf

И задала параметр safecrlf.

A terminal window with a title bar showing 'pvbarabash@endless: ~'. The window contains two lines of text: 'pvbarabash@endless:~\$ git config --global core.safecrlf warn' and 'pvbarabash@endless:~\$' followed by a cursor. The window has standard macOS-style window controls (red, yellow, green buttons) and a search icon in the top right corner.

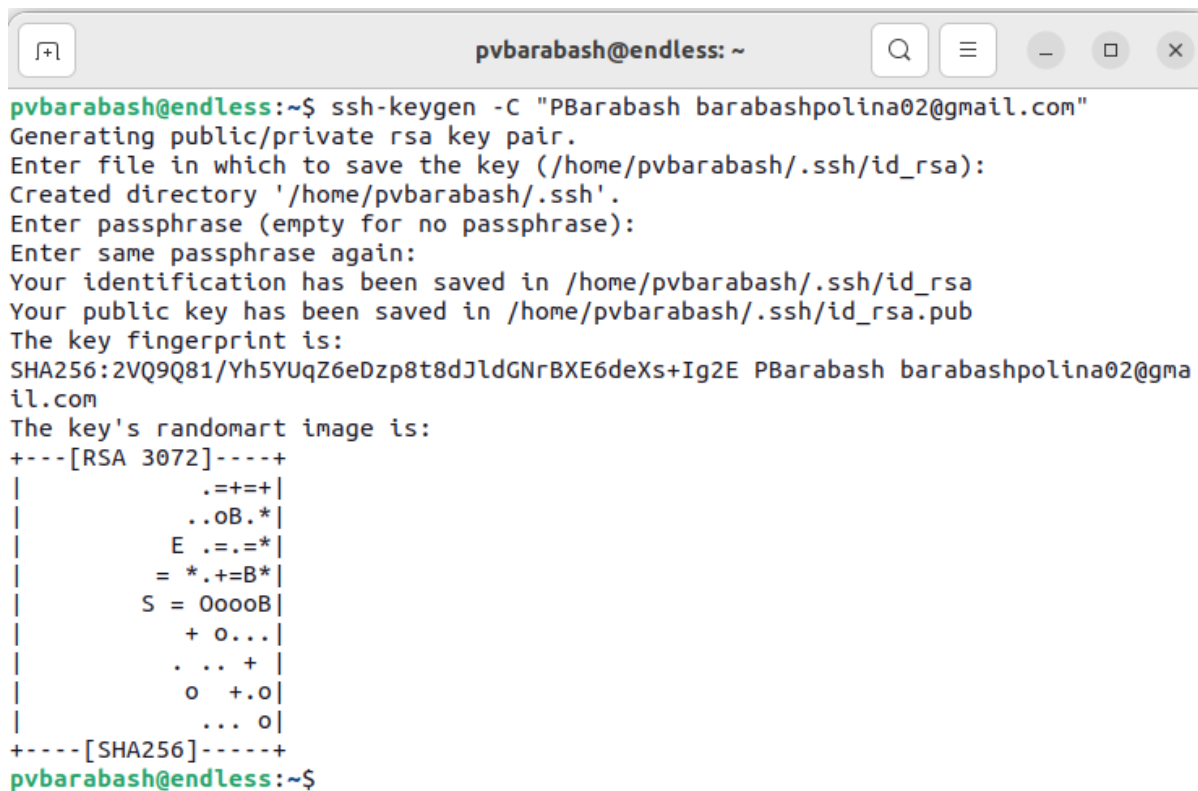
```
pvbarabash@endless:~$ git config --global core.safecrlf warn
pvbarabash@endless:~$
```

Рис. 2.6: Параметр safecrlf

Задание №3. Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый):
`ssh-keygen -C "Имя Фамилия work@mail"`

Далее необходимо загрузить сгенерённый открытый ключ. Для этого зайти на сайт <http://github.org/> под своей учётной записью и перейти в меню Setting . После этого выбрать в боковом меню SSH and GPG keys и нажать кнопку New SSH key. Скопировав из локальной консоли ключ в буфер обмена, вставить ключ в появившееся на сайте поле и указать для ключа имя (Title).

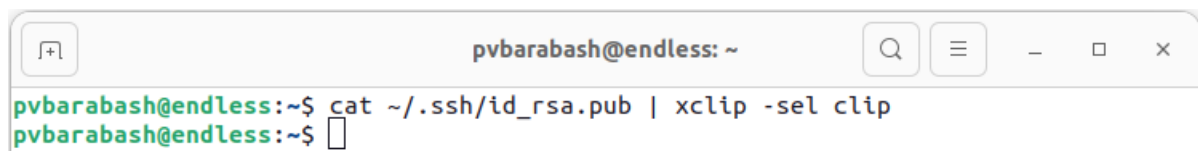
Я сгенерировала пару ключей (приватный и открытый). Ключи сохранились в каталоге `~/.ssh/`.



```
pvbarabash@endless: ~
pvbarabash@endless:~$ ssh-keygen -C "PBarabash barabashpolina02@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/pvbarabash/.ssh/id_rsa):
Created directory '/home/pvbarabash/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/pvbarabash/.ssh/id_rsa
Your public key has been saved in /home/pvbarabash/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:2VQ9Q81/Yh5YUqZ6eDzp8t8dJldGNrBXE6deXs+Ig2E PBarabash barabashpolina02@gmail.com
The key's randomart image is:
+---[RSA 3072]---+
|           .+=+  |
|          ..oB.* |
|         E .+=*  |
|        = *.+=B* |
|       S = 0oooB |
|      + o...    |
|     . . . +   |
|    o  +.o     |
|   ... o      |
+-----[SHA256]-----+
pvbarabash@endless:~$
```

Рис. 2.7: Генерация двух ключей

Я скопировала сгенерённый открытый ключ с помощью команды `cat ~/.ssh/id_rsa.pub | xclip -sel clip`.



```
pvbarabash@endless: ~
pvbarabash@endless:~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
pvbarabash@endless:~$
```

Рис. 2.8: Копирование в буфер обмена

Я зашла на сайт <http://github.org/> под своей учётной записью, перешла в меню Setting, выбрала в боковом меню SSH and GPG keys и, нажав на кнопку New SSH key, загрузила скопированный в буфер обмена сгенерённый открытый ключ. Указала имя для ключа.

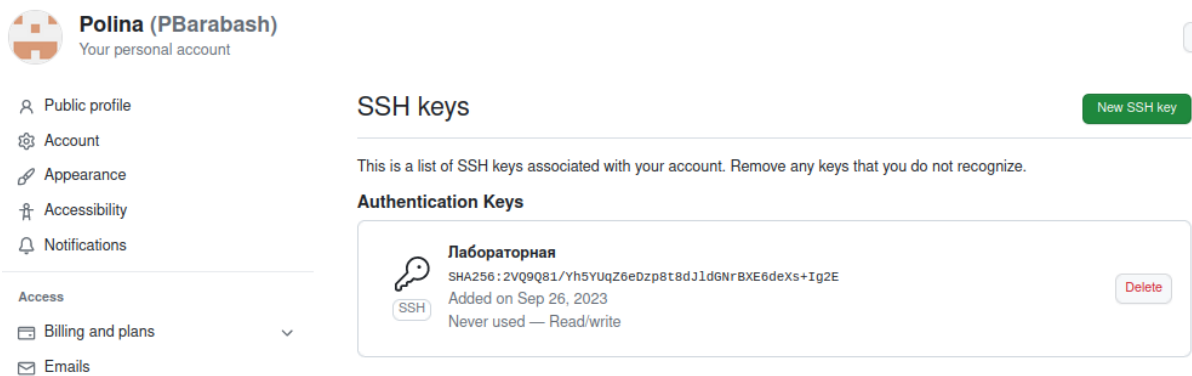


Рис. 2.9: Загрузка открытого ключа на github

Задание №4. Откройте терминал и создайте каталог для предмета «Архитектура компьютера»: `mkdir -p ~/work/study/2023-2024/«Архитектура компьютера»`

Я создала каталог предмета «Архитектура компьютера» и проверила, что он существует с помощью команды `ls`.

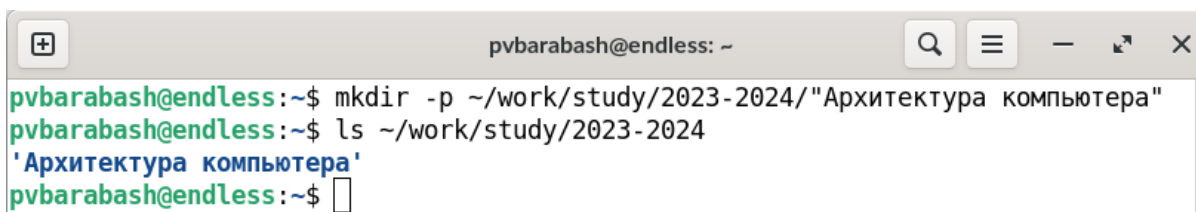


Рис. 2.10: Создание каталога предмета «Архитектура компьютера»

Задание №5. Перейдите на страницу репозитория с шаблоном курса <https://github.com/yamadharma/course-directory-student-template>. Далее выберите Use this template. В открывшемся окне задайте имя репозитория (Repository name) `n` и создайте репозиторий (кнопка Create repository from template). Откройте терминал и перейдите в каталог курса: `cd ~/work/study/2023-2024/«Архитектура компьютера»`

Клонируйте созданный репозиторий: `git clone --recursive git@github.com:/study_2023-2024_arch-pc.git`

Я перешла на страницу репозитория с шаблоном курса и выбрала Use this

template. Далее был предоставлен выбор Create a new repository или Open in a codespace.

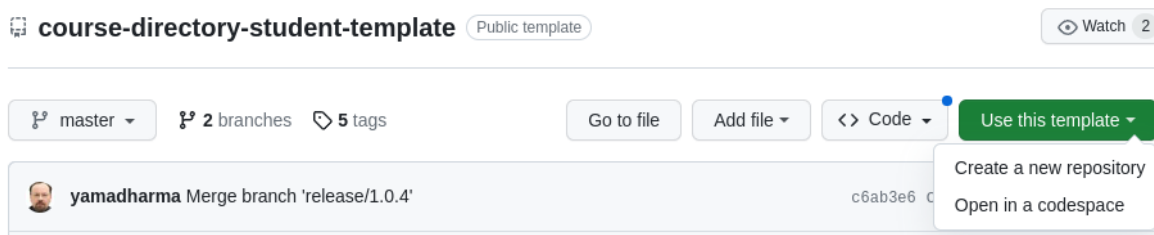


Рис. 2.11: Выбор при нажатии Use this template

Я выбрала Create a new repository и увидела нужные поля для заполнения.

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 (*).

Owner * **Repository name ***
PBarabash /

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

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. 2.12: Страница Create a new repository

Я задала имя репозитория (Repository name) study_2023–2024_arhpc и создала

репозиторий (кнопка Create repository).

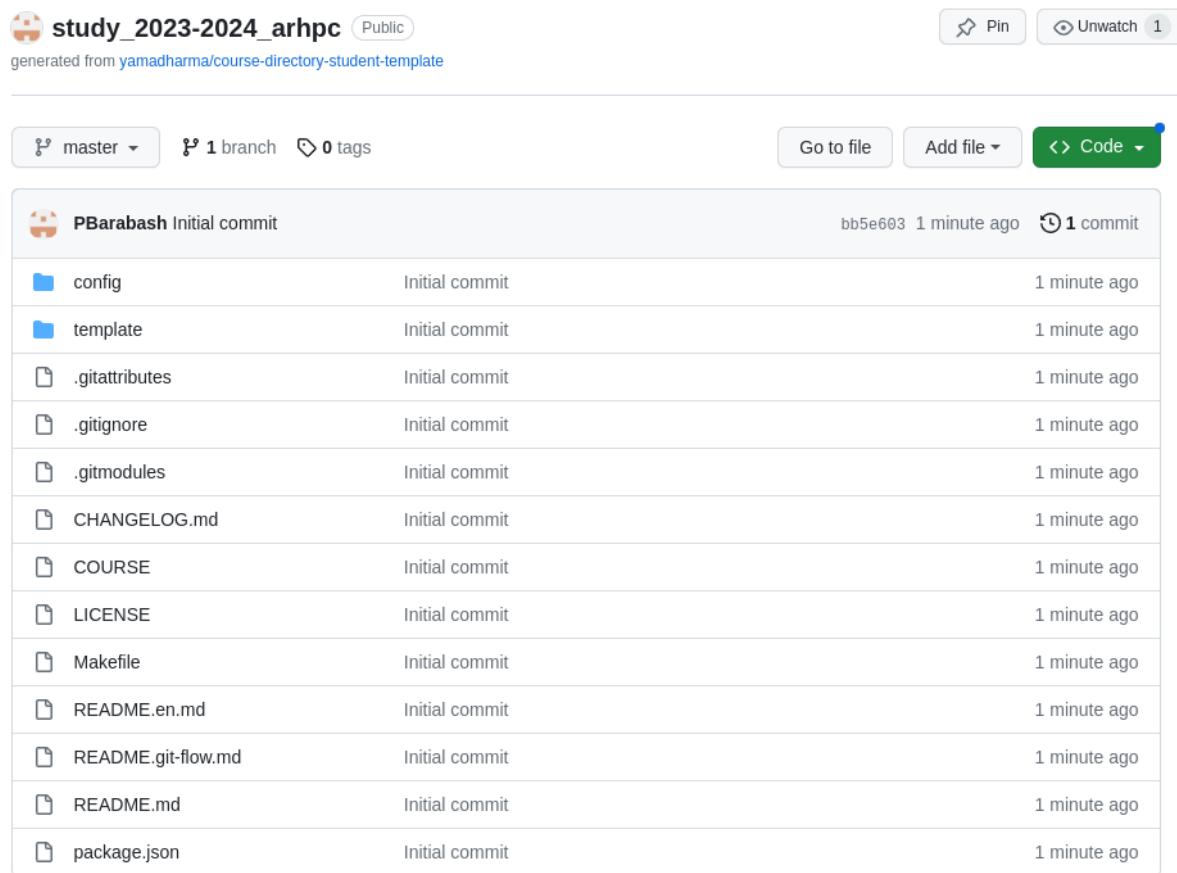


Рис. 2.13: Созданный репозиторий по шаблону

Я открыла терминал и перешла в каталог курса.

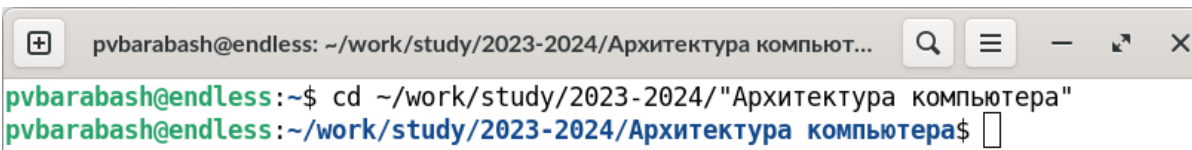


Рис. 2.14: Переход в каталог курса через терминал

Я клонировала созданный репозиторий, скопировав ссылку для клонирования на странице созданного репозитория Code.

```
pvbarabash@endless: ~/work/study/2023-2024/Архитектура ...
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера$ git clone --recursive git@github.com:PBarabash/study_2023-2024_arhpc.git arch-pc
Клонирование в «arch-pc»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
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: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 0
Получение объектов: 100% (27/27), 16.93 КиБ | 4.23 МиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharm/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharm/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/pvbarabash/work/study/2023-2024/Архитектура компьютера/arch-pc/template/presentation»...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
Получение объектов: 100% (82/82), 92.90 КиБ | 1022.00 КиБ/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/home/pvbarabash/work/study/2023-2024/Архитектура компьютера/arch-pc/template/report»...
remote: Enumerating objects: 101, done.
remote: Counting objects: 100% (101/101), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 101 (delta 40), reused 88 (delta 27), pack-reused 0
Получение объектов: 100% (101/101), 327.25 КиБ | 2.06 МиБ/с, готово.
Определение изменений: 100% (40/40), готово.
Submodule path 'template/presentation': checked out 'b1be3800ee91f5809264cb755d316174540b753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a83917b82e3aef11a33b1e3b2'
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера$
```

Рис. 2.15: Клонирование созданного на github репозитория

Через файловый менеджер я проверила, что репозиторий клонирован.

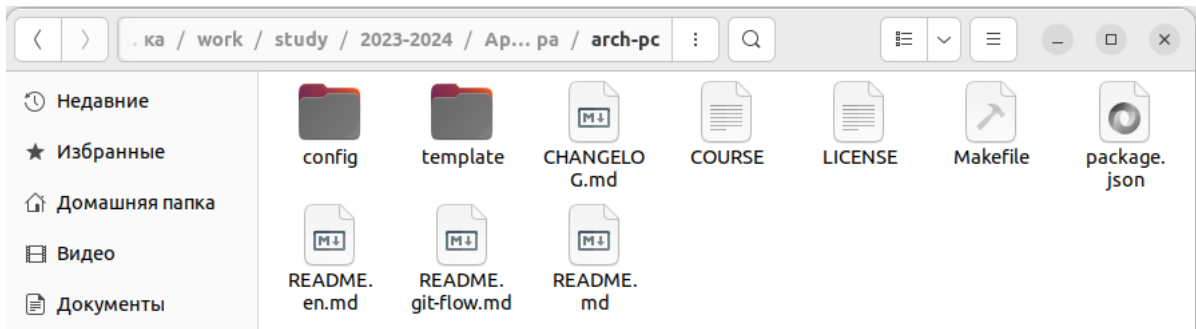


Рис. 2.16: Проверка, что репозиторий клонирован

Задание № 6. Перейдите в каталог курса: `cd ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc`

Удалите лишние файлы: `rm package.json`

Создайте необходимые каталоги: `echo arch-pc > COURSE make`

Отправьте файлы на сервер: `git add . git commit -am 'feat(main): make course structure' git push`

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

Я перешла в каталог курса, используя данный абсолютный путь.

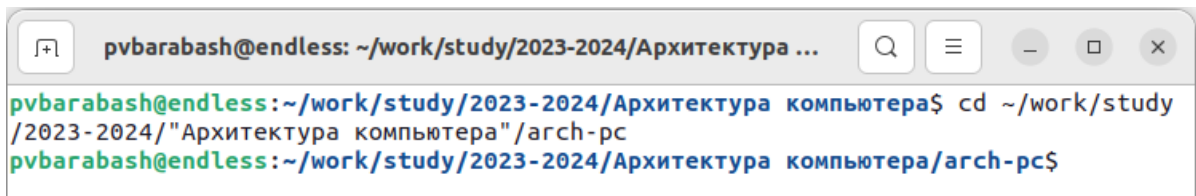
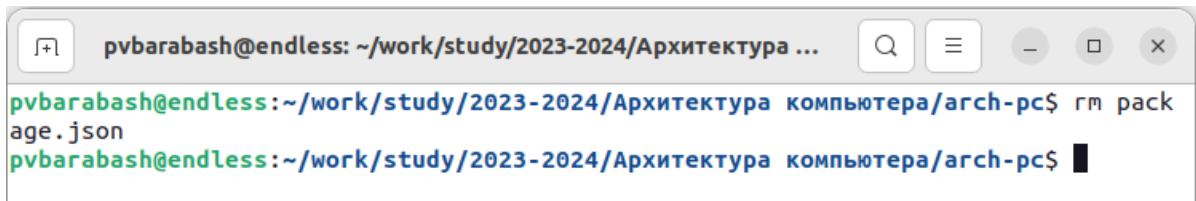


Рис. 2.17: Переход в каталог курса

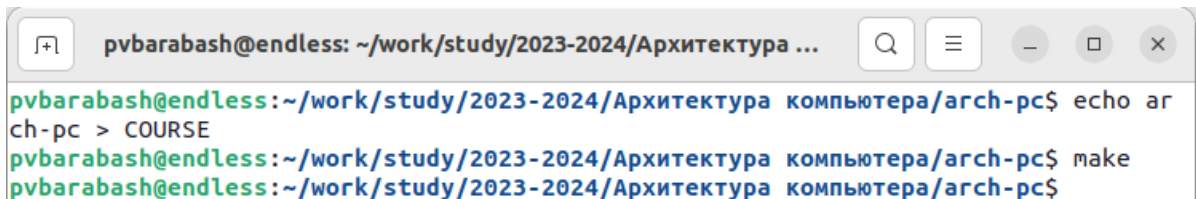
Я удалила лишние файлы.



```
pvbarabash@endless: ~/work/study/2023-2024/Архитектура ...
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ rm package.json
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 2.18: Удаление файла package.json

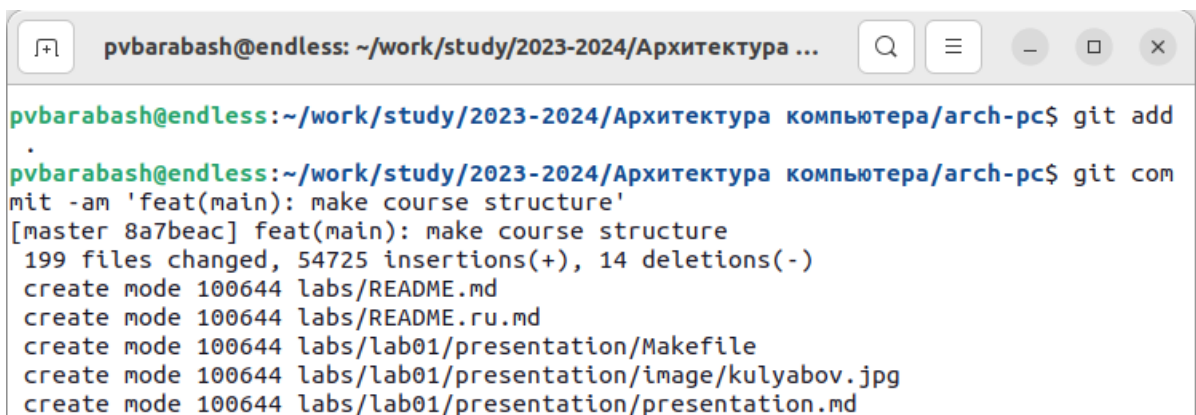
Я создала необходимые каталоги.



```
pvbarabash@endless: ~/work/study/2023-2024/Архитектура ...
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ make
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 2.19: Создание необходимых каталогов

Я отправила файлы на сервер. Выдача после `git commit -am 'feat(main): make course structure'` очень длинная, на скриншоте приведено начало.



```
pvbarabash@endless: ~/work/study/2023-2024/Архитектура ...
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
[master 8a7beac] feat(main): make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/presentation/presentation.md
```

Рис. 2.20: Отправление файлов на сервер (Часть I)

```
pvbarabash@endless: ~/work/study/2023-2024/Архитектура ...
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push
h
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.14 КиБ | 2.65 МиБ/с, готово.
Всего 35 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:PBBarabash/study_2023-2024_arhpc.git
   ac05f51..8a7beac master -> master
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 2.21: Отправление файлов на сервер (Часть II)

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

```
pvbarabash@endless: ~/work/study/2023-2024/Архитектура ...
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ ls
CHANGELOG.md  COURSE  LICENSE  prepare  README.en.md  README.md
config        labs    Makefile  presentation  README.git-flow.md  template
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ ls labs
lab01 lab03 lab05 lab07 lab09 lab11  README.ru.md
lab02 lab04 lab06 lab08 lab10 README.md
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 2.22: Правильность создания иерархии рабочего пространства в локальном репозитории

study_2023-2024_arhpc / labs /



PBarabash feat(main): make course structure

Name



..



lab01



lab02



lab03



lab04



lab05



lab06



lab07



lab08



lab09



lab10



lab11



README.md



README.ru.md

Рис. 2.23: Правильность создания иерархии рабочего пространства на github

3 Выполнение самостоятельной работы

Задание №1. Создайте отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report).

Я переместила отчёт по выполнению лабораторной работы с помощью команды mv в каталог labs>lab02>report из каталога Загрузки, так как составляла отчет в гугл документах.

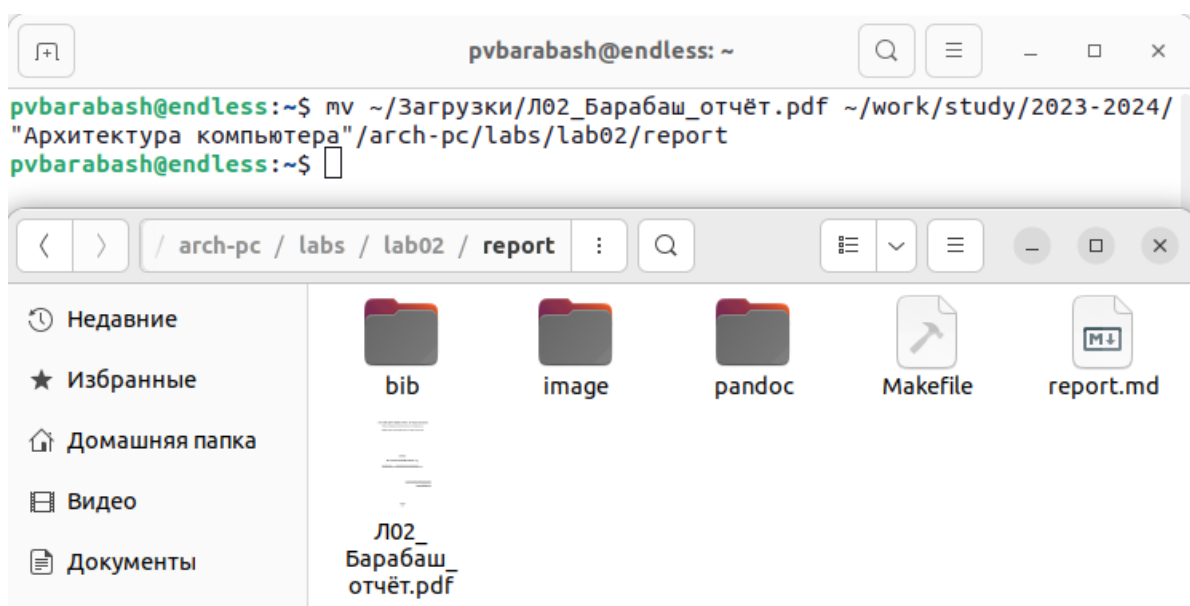


Рис. 3.1: Перемещение файла

Чтобы создать отчёт, нужно использовать команду touch.

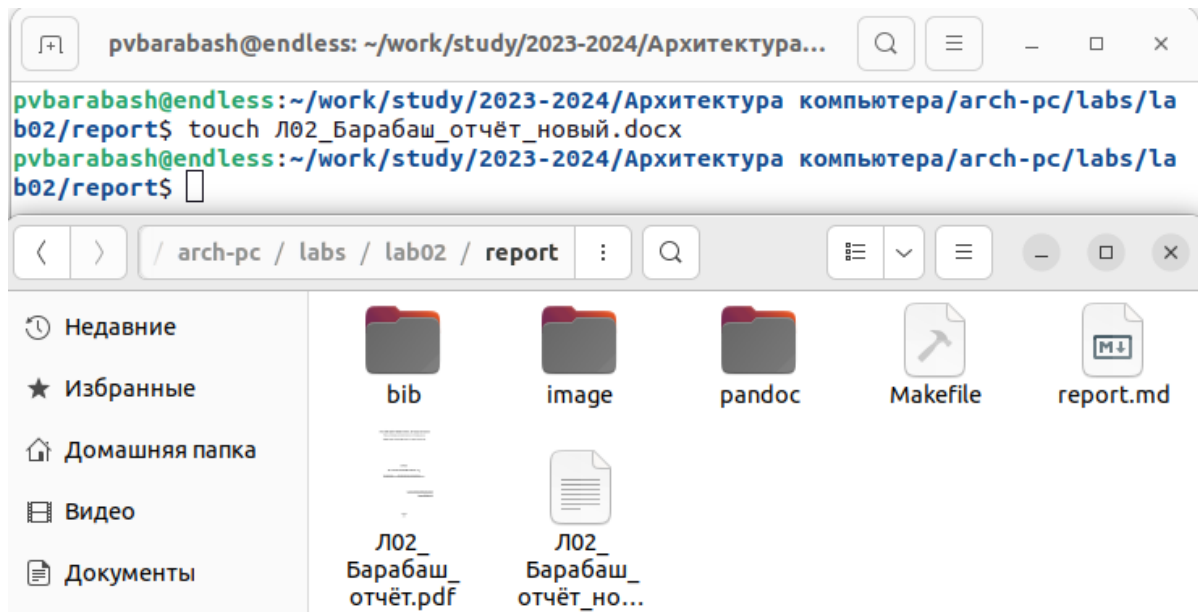


Рис. 3.2: Создание файла

Задание №2. Скопируйте отчеты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства.

Я также переместила отчёт по первой лабораторной работе из загрузок в соответствующий каталог рабочего пространства.

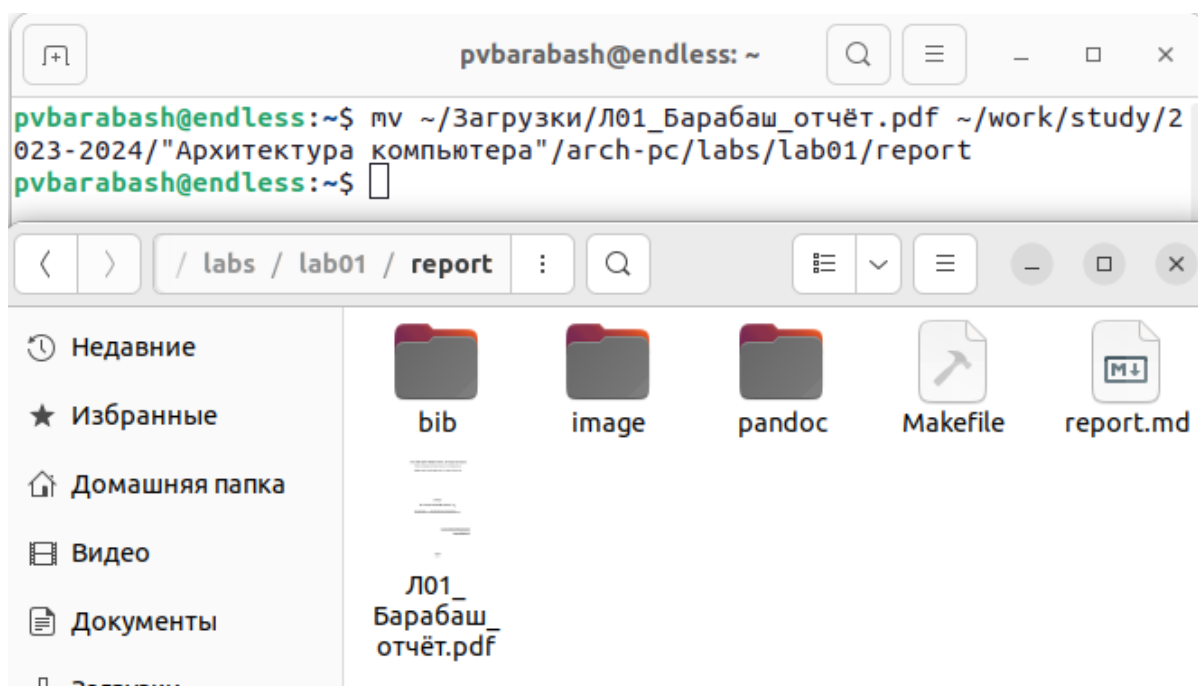
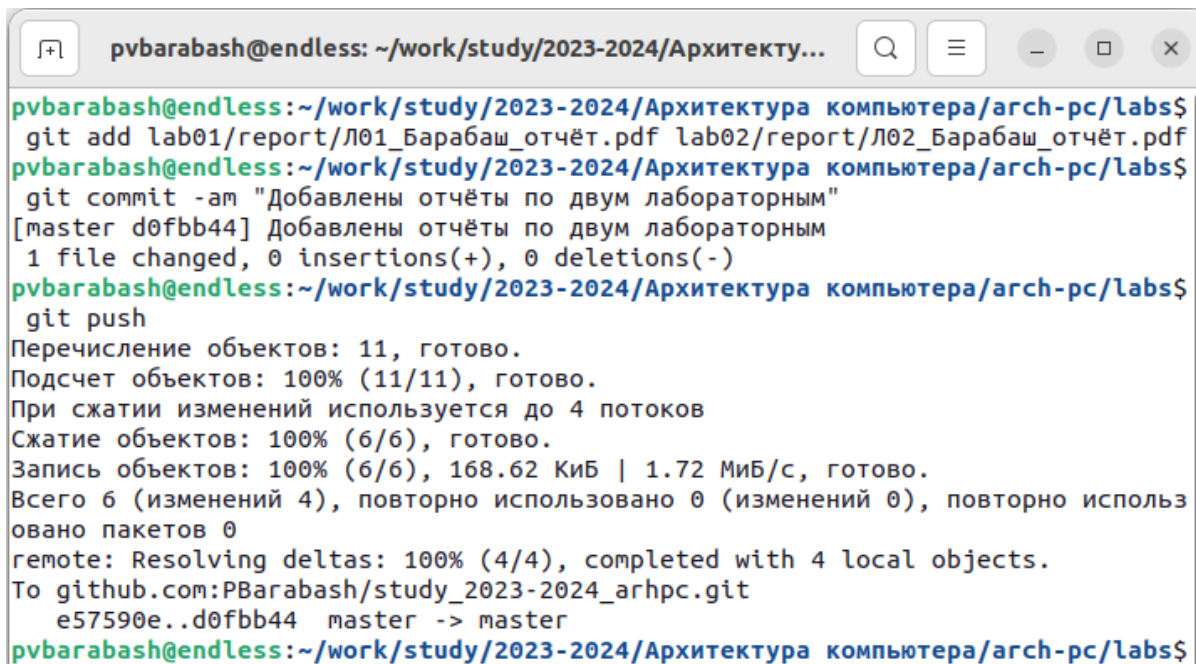


Рис. 3.3: Перемещение предыдущего отчёта в нужную папку

Задание №3. Загрузите файлы на github.

Я загрузила файлы на github. Для этого с помощью команды `git add` я выбрала оба отчёта, которые и были добавлены по сравнению с предыдущей версией. Затем я внесла описание изменений с помощью `git commit -am`. Отправила изменения с помощью `git push`.



```
pvbarabash@endless: ~/work/study/2023-2024/Архитекту...
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs$
git add lab01/report/Л01_Барабаш_отчёт.pdf lab02/report/Л02_Барабаш_отчёт.pdf
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs$
git commit -am "Добавлены отчёты по двум лабораторным"
[master d0fbb44] Добавлены отчёты по двум лабораторным
1 file changed, 0 insertions(+), 0 deletions(-)
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs$
git push
Перечисление объектов: 11, готово.
Подсчет объектов: 100% (11/11), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 168.62 КиБ | 1.72 МиБ/с, готово.
Всего 6 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To github.com:PBarabash/study_2023-2024_arhpc.git
e57590e..d0fbb44 master -> master
pvbarabash@endless:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs$
```

Рис. 3.4: Действия в терминале для загрузки файлов на github

На следующем скриншоте видно, что файлы загружены на github. В структуре слева видно, что файл “Л01_Барабаш_отчёт” находится в каталоге lab1/report, а сверху видно, что открыт каталог lab2/report, где также есть файл “Л02_Барабаш_отчёт”, также отображается последнее изменение – комментарий “Добавлены отчёты по двум лабораторным”.

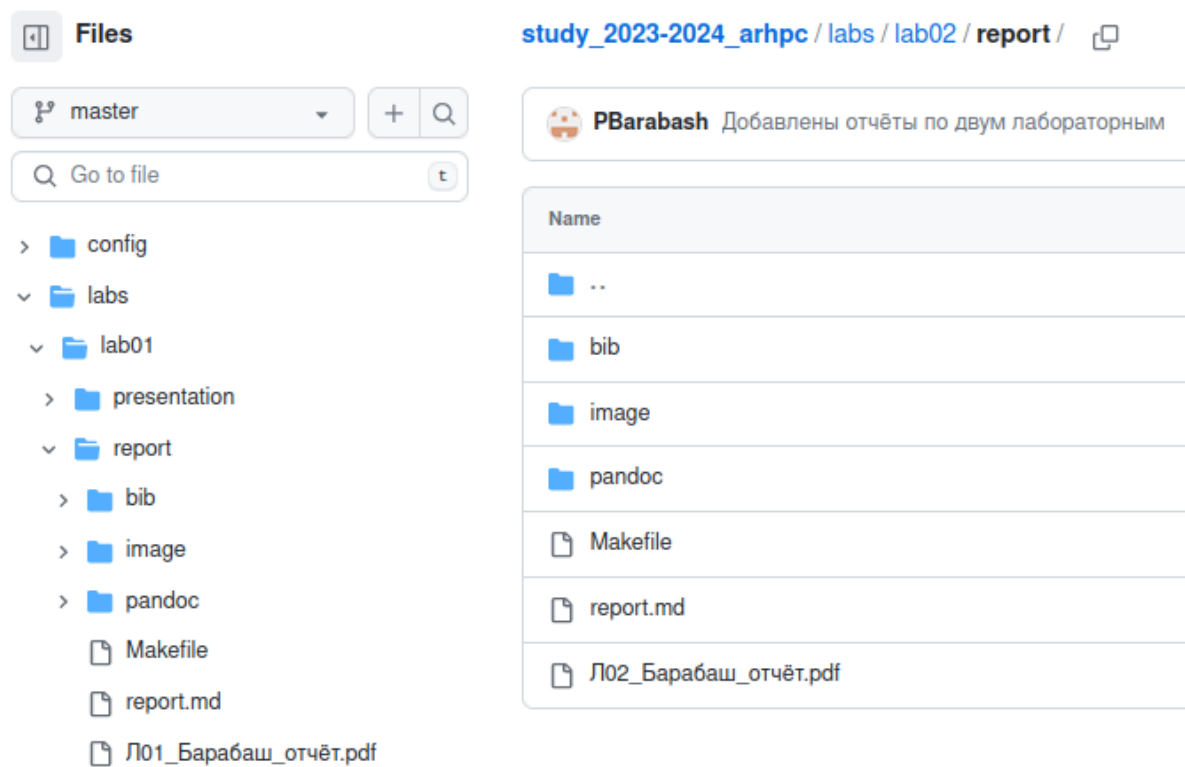


Рис. 3.5: Проверка загрузки файлов на github

4 Выводы

Я изучила идеологию и применение средств контроля версий. Узнала о локальном или удалённом репозитории, о том, что к удалённому может быть настроен доступ у нескольких пользователей. Узнала, что 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.

Я приобрела практические навыки по работе с системой git. Узнала, как создавать репозиторий на github, клонировать его на свой компьютер, добавлять локальные изменения на github.

...