

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Компьютерные и информационные науки

## ОТЧЕТ

### ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

дисциплина:    Архитектура компьютера

Студент:   Нечаева Кира

Группа: НКАбд-04-23

МОСКВА

2023\_\_ г.

# Содержание

1. Цель работы.....	3
2. Задание.....	4
3. Теоретическое введение.....	5
4. Выполнение лабораторной работы.....	7
5. Выводы.....	15
6. Вопросы для самопроверки.....	16
7. Источники.....	17

# **1 Цель работы**

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

## 2 Задание

1. Настройка github
2. Базовая настройка git
3. Создание SSH ключа
4. Создание рабочего пространства на основе шаблона
5. Создание репозитория курса на основе шаблона
6. Настройка каталога курса
7. Задание для самостоятельной работы

### 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.

Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки

ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.

Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

## 4 Выполнение лабораторной работы

### 1. Настройка github

Поскольку у меня уже есть аккаунт на Github, я пропускаю этот шаг. (рис. 1)



Рис. 1. Аккаунт на github

### 2. Базовая настройка git

Сначала сделаю предварительную конфигурацию git. Открываю терминал и ввожу команды:

```
git config --global user.name "<Kira Nechaeva>"
```

```
git config --global user.email "<kirusya1234@gmail.com>"
```

указав свои имя и email.

Настраиваю utf-8 в выводе сообщений git для верного отображения символов.

Задаю имя начальной ветки "master".

Задаю параметр autocrlf со значением input.

Задаю параметр safecrlf со значением warn, так Git будет проверять преобразование на обратимость.

Все действия отображены в рис.2 на следующей странице.

```
kanechaeva@192:~  
[kanechaeva@192 ~]$ git config --global user.name "<Kira Nechaeva>"  
[kanechaeva@192 ~]$ git config --global user.email "<kirusya1234@gmail.com>"  
[kanechaeva@192 ~]$ git config --global core.quotepath false  
[kanechaeva@192 ~]$ git config --global init.defaultBranch master  
[kanechaeva@192 ~]$ git config --global core.autocrlf input  
[kanechaeva@192 ~]$ git config --global core.safecrlf warn
```

Рис.2. Базовые настройки Git.

### 3. Создание SSH ключа

Для последующей идентификации пользователя на сервере репозитория генерирую пару ключей (приватный и открытый) с помощью команды `ssh-keygen -C "Кира Нечаева <kirusya1234@gmail.com>"`

Ключи сохраняются в каталоге `~/.ssh/` автоматически. (рис.3)

```
kanechaeva@192:~  
[kanechaeva@192 ~]$ ssh-keygen -C "Кира Нечаева <kirusya1234@gmail.com>"  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/kanechaeva/.ssh/id_rsa):  
Created directory '/home/kanechaeva/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/kanechaeva/.ssh/id_rsa  
Your public key has been saved in /home/kanechaeva/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:0sXgTxTutXKA0UMPnuhMroSveCCyyb++6yne8EeFIIsQ Кира Нечаева <kirusya1234@gmail.com>  
The key's randomart image is:  
+---[RSA 3072]-----+  
|.      + o.      |  
| E        = @      |  
|.      + O B .    |  
| . o * o B o .    |  
|  o o * S + o     |  
|+   o o .   o     |  
|+=  +           |  
|o+=...          |  
|oo*@=           |  
+---[SHA256]-----+  
[kanechaeva@192 ~]$
```

Рис.3. Создание SSH ключа.

Копирую открытый ключ из директории, в которой он был сохранен, с помощью утилиты `xclip`. (рис. 4)

```
[kanechaeva@192 ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip  
[kanechaeva@192 ~]$
```

Рис. 4. Копирование содержимого файла



Захожу на сайт GitHub. Выбираю в меню “Настройки”, затем “SSH and GPG keys”. Нажимаю кнопку «New SSH key». (рис. 5)

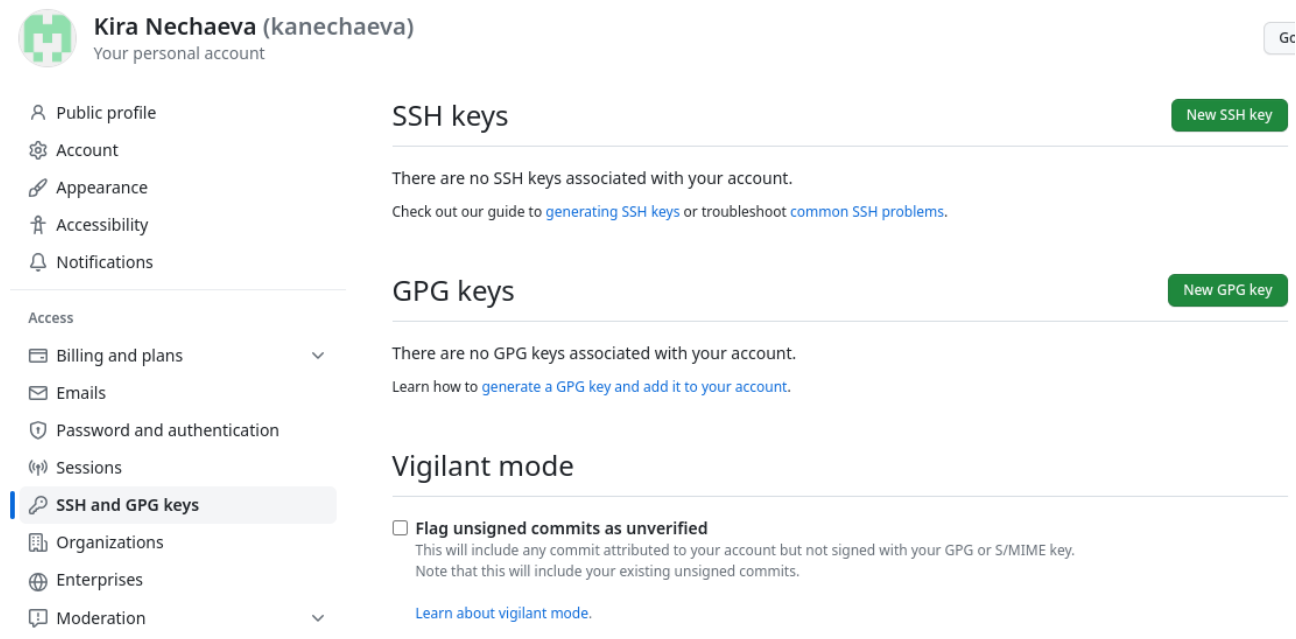


Рис. 5. Окно SSH and GPG keys.

Вставляю скопированный ключ в поле «Key». Называю его kanechaeva1910. Нажимаю «Add SSH-key», чтобы завершить добавление ключа. (рис. 6)

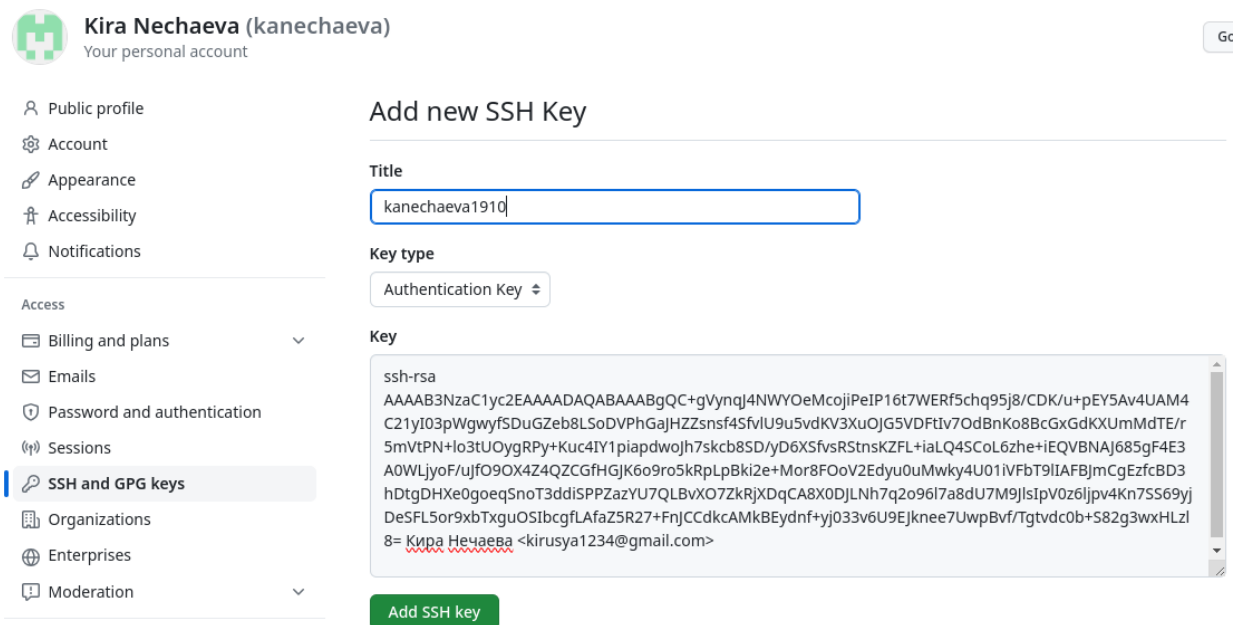


Рис. 6. Добавление ключа

## 4. Создание рабочего пространства на основе шаблона

Захожу в терминал. Создаю директорию, рабочее пространство, с помощью утилиты mkdir, благодаря ключу -p создаю все директории после домашней

~/work/study/2023-2024/"Архитектура компьютера" рекурсивно.

Далее проверяю с помощью ls корректность выполнения своих действий. (рис. 7).

```
[kanechaeva@192 ~]$ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
[kanechaeva@192 ~]$ ls
work      Документы  Изображения  Общедоступные  Шаблоны
Видео     Загрузки   Музыка       'Рабочий стол'
```

Рис. 7. Создание рабочего пространства.

## 5. Создание репозитория курса на основе шаблона

Репозиторий на основе шаблона создаю через web-интерфейс github. Перехожу на страницу репозитория с шаблоном курса <https://github.com/yamadharma/course-directory-student-template>. Далее выбираю Use this template. (рис. 8)

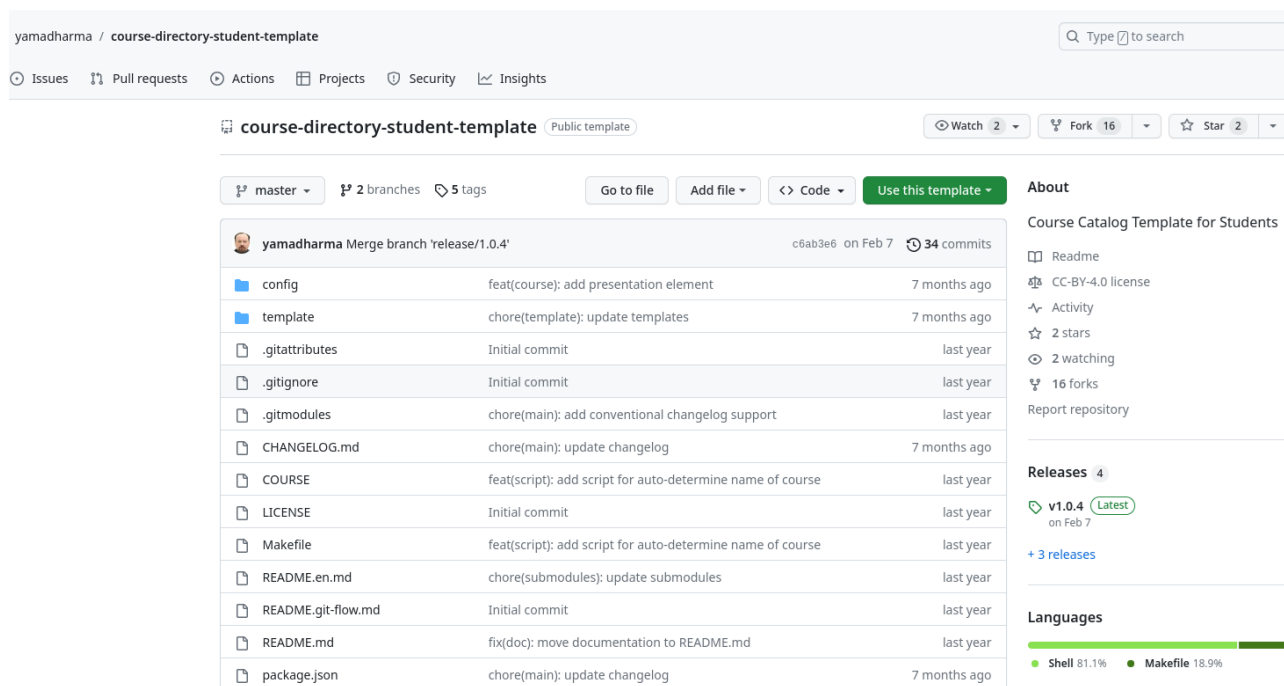


Рис. 8. Шаблон для репозитория.

В открывшемся окне задаю имя репозитория study\_2023-2024\_arhpc и создаю репозиторий, нажав на кнопку Create repository from template. К сожалению, я не успела сделать скриншот создания репозитория, так что прикрепляю фотографию с созданным репозиторием. (рис. 9)

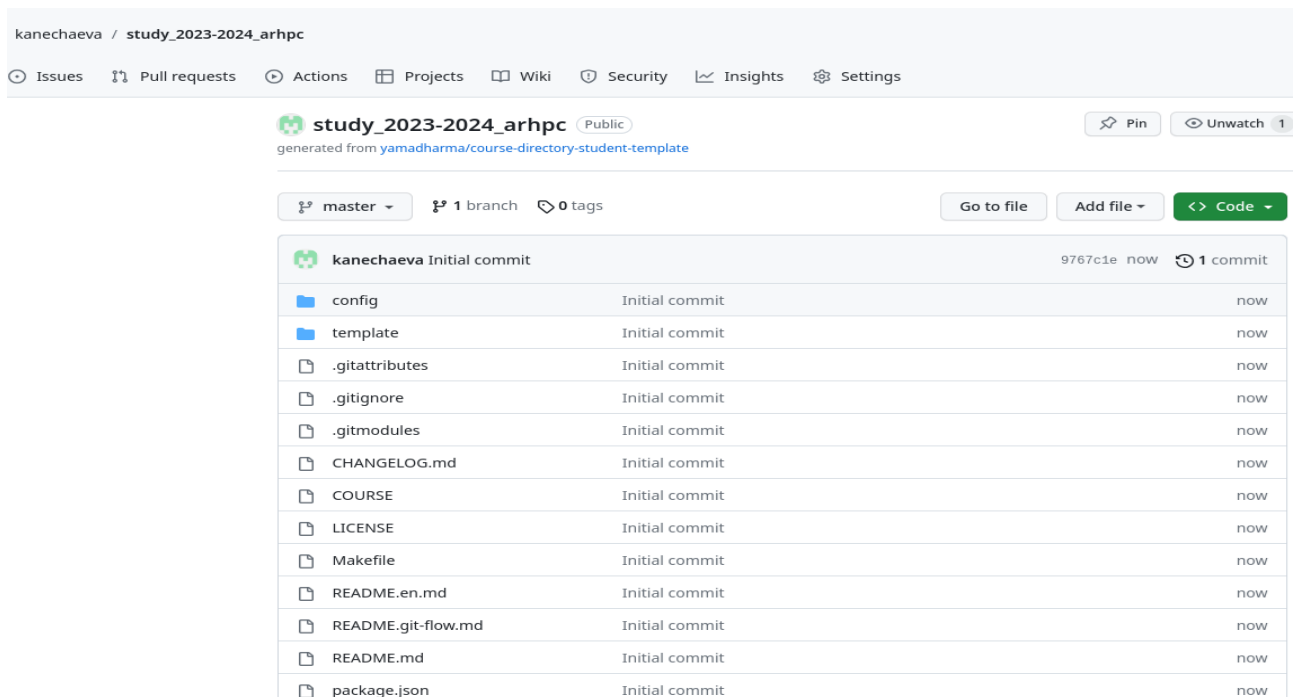


Рис. 9. Созданный репозиторий.

Открывю терминал и перехожу в каталог курса с помощью команды cd (рис. 10).

```
[kanechaeva@fedora ~]$ cd ~/work/study/2023-2024/"archcomp"
```

Рис. 10. Перемещение в каталог курса.

Затем клонирую созданный репозиторий. (рис. 11)

```
kanechaeva@fedora:~/work/study/2023-2024/archcomp

[kanechaeva@fedora archcomp]$ git clone --recursive git@github.com:kanechaeva>/study_2023-2024_arhpc.git arch-pc
bash: kanechaeva: Нет такого файла или каталога
[kanechaeva@fedora archcomp]$ ls
[kanechaeva@fedora archcomp]$ git clone --recursive git@github.com:kanechaeva/study_2023-2024_arhpc.git
Клонирование в «study_2023-2024_arhpc»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
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), 16.94 КиБ | 251.00 КиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadhama/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadhama/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/kanechaeva/work/study/2023-2024/archcomp/study_2023-2024_arhpc/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 | 131.00 КиБ/с, готово.
```

Рис. 11. Клонирование репозитория

Затем проверяю корректность своих действий с помощью утилиты ls (рис.12).

```
[kanechaeva@fedora archcomp]$ ls
study_2023-2024_arhpc
[kanechaeva@fedora archcomp]$
```

Рис. 12. Наличие папки в директории archcomp

## 6. Настройка каталога курса

Перехожу в каталог курса с помощью утилиты cd.

Затем удаляю лишний файл package.json, используя команду rm.

После чего создаю необходимые каталоги.

Все действия отображены в рис. 13.

```
[kanechaeva@fedora ~]$ cd /home/kanechaeva/work/study/2023-2024/archcomp/study_2023-2024_arhpc
[kanechaeva@fedora study_2023-2024_arhpc]$ rm package.json
[kanechaeva@fedora study_2023-2024_arhpc]$ echo arch-pc > COURSE
[kanechaeva@fedora study_2023-2024_arhpc]$ make
```

Рис. 13. Удаление и создание необходимых файлов и каталогов.

Далее отправляю файлы на сервер (рис. 14 и рис. 15).

```
[kanechaeva@fedora study_2023-2024_arhpc]$ git add .
[kanechaeva@fedora study_2023-2024_arhpc]$ git commit -am 'feat(main): make course structure'
[master 7bb9349] 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
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
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 labs/lab01/report/report.md
create mode 100644 labs/lab02/presentation/Makefile
create mode 100644 labs/lab02/presentation/image/kulyabov.jpg
create mode 100644 labs/lab02/presentation/presentation.md
create mode 100644 labs/lab02/report/Makefile
create mode 100644 labs/lab02/report/bib/cite.bib
create mode 100644 labs/lab02/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab02/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
```

Рис. 14. Добавление и сохранение изменений на сервере

```
[kanechaeva@fedora study_2023-2024_arhpc]$ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 12 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.13 КиБ | 2.48 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:kanechaeva/study_2023-2024_arhpc.git
   9767c1e..7bb9349  master -> master
[kanechaeva@fedora study_2023-2024_arhpc]$
```

Рис. 15. Выгрузка изменений.

Проверяю правильность создания иерархии рабочего пространства в локальном репозитории и на странице github. (рис. 16)

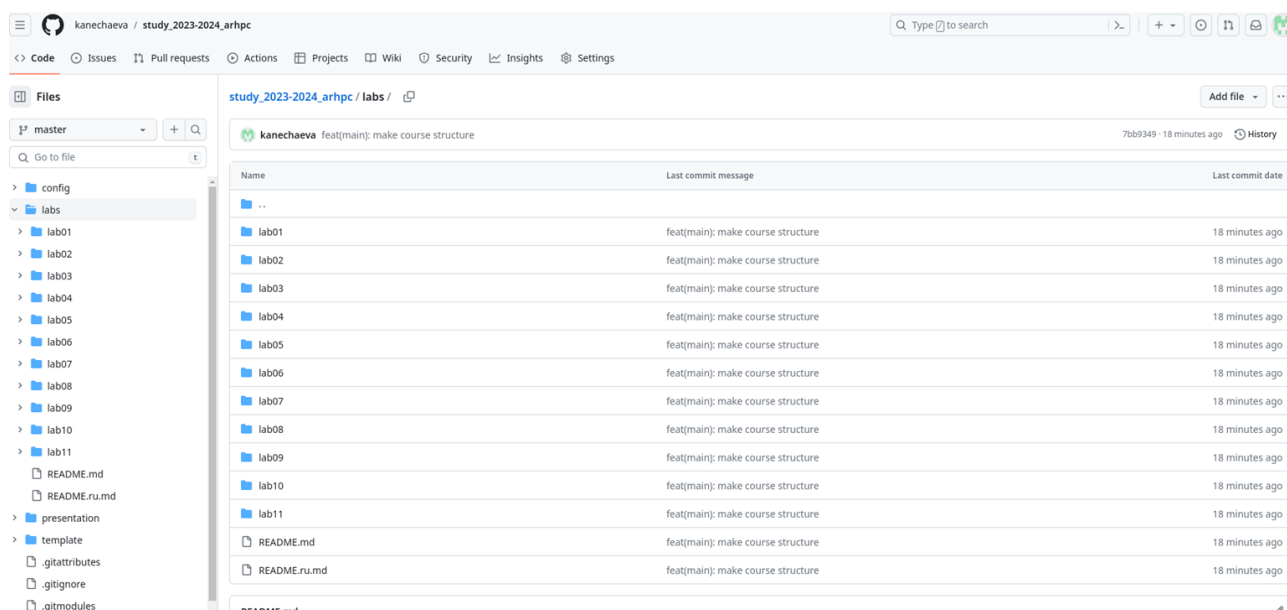


Рис. 16. Проверка проделанных действий.

## 7. Задание для самостоятельной работы

### №1

Перехожу в директорию labs/lab02/report. Создаю в каталоге файл для отчета по второй лабораторной работе.(рис. 17)

```
[kanechaeva@192 ~]$ cd /home/kanechaeva/work/study/2023-2024/archcomp/study_2023-2024_arhpc/labs/lab02/report
[kanechaeva@192 report]$ touch Л02_Нечаева_отчет
```

Рис. 17. Создание файла для отчета по лабораторной работе

Оформляю отчет я в текстовом процессоре LibreOffice Writer.

## №2

Далее я перехожу в каталог “Загрузки”, чтоб скопировать из него отчет по первой лабораторной работе в директорию labs/lab1/report. (рис. 18)

```
[kanechaeva@192 report]$ cd
[kanechaeva@192 ~]$ cd Загрузки
[kanechaeva@192 Загрузки]$ cp Л01_Нечаева_отчет.pdf /home/kanechaeva/work/study/
2023-2024/archcomp/study_2023-2024_arhpc/labs/lab01/report
[kanechaeva@192 Загрузки]$ cd /home/kanechaeva/work/study/2023-2024/archcomp/stu
dy_2023-2024_arhpc/labs/lab01/report
[kanechaeva@192 report]$ ls
bib image Makefile pandoc report.md Л01_Нечаева_отчет.pdf
[kanechaeva@192 report]$
```

Рис. 18. копирование файла Л01\_Нечаева\_отчет в другую директорию

## №3

Загружаю файл Л01\_Нечаева\_отчет с помощью таких команд, как:

git add – добавление файла

git commit -am 'feat(main): make course structure' – сохранение изменений

git push – отправка изменений в репозиторий.

Все действия отображены на рис. 19.

```
[kanechaeva@192 report]$ git add Л01_Нечаева_отчет.pdf
[kanechaeva@192 report]$ git commit -am 'feat(main): make course structure'
[master 7d75c3a] feat(main): make course structure
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Л01_Нечаева_отчет.pdf
[kanechaeva@192 report]$ git push
Перечисление объектов: 11, готово.
Подсчет объектов: 100% (11/11), готово.
При сжатии изменений используется до 12 потоков
Сжатие объектов: 100% (7/7), готово.
Запись объектов: 100% (7/7), 688.57 КиБ | 5.03 МиБ/с, готово.
Всего 7 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 3 local objects.
To github.com:kanechaeva/study_2023-2024_arhpc.git
f859c8f..9aeaf32 master -> master
```

Рис. 19. Загрузка файлов на github

Так как сейчас я взаимодействую с отчетом по второй лабораторной работе, его загрузку на github я осуществлю позднее по тому же алгоритму.

## **5 Вывод**

При выполнении данной лабораторной работы я изучила идеологию и применение средств контроля версий и приобрела практические навыки по работе с системой git.

## **7 Источники**

1. ТУИС – Архитектура ЭВМ – [Электронный ресурс] -  
<https://esystem.rudn.ru/mod/resource/view.php?id=1030548>