
title: “Отчет по лабораторной работе №2” subtitle: “Архитектура компьютеров/Операционные системы” author: “Нилова Кристина Артуровна”

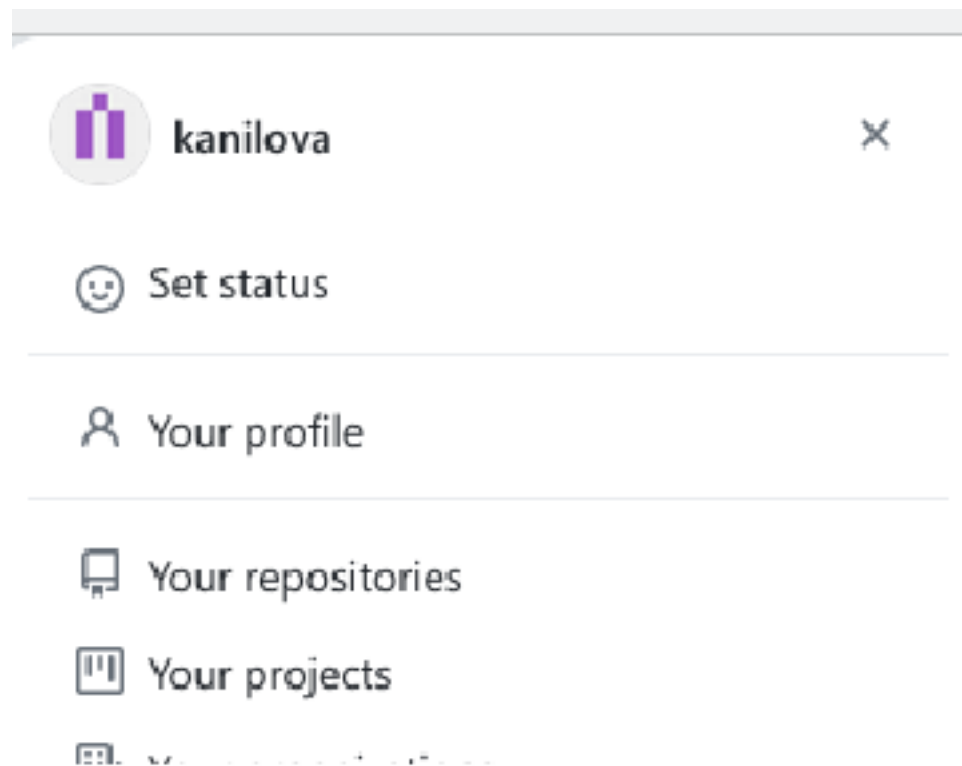
Цель работы

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

Ход работы

1. Настройка Github.

1. Так как учетная запись на <https://github.com/> ранее была создана, заходим уже в существующий аккаунт.



{#fig:001

width=70%} Рис.1.1 Аккаунт на Github.

2. Запускаем виртуальную машину. Сначала сделаем предварительную конфигурацию git. При помощи команд `git config --global user.name ""`, `git config --global user.email ""`, укажем имя и email владельца репозитория.

```
kanilova@dk3n62 ~ $ git config --global user.name "<kanilova>"
kanilova@dk3n62 ~ $ git config --global user.email "<1132230797@pfur.ru>"
kanilova@dk3n62 ~ $
```

{#fig:002 width=70%} Рис.1.2 Предварительная конфигурация git.

3. Настраиваем utf-8 в выводе сообщений git. Затем задаем имя начальной ветки (будем называть её master). Задаем параметр autocrlf со значением input (т.к. работаем в системе Linux, чтобы конвертировать CRLF в LF только при коммитах (рис. 1.3.)). CR и LF – это символы, которые используются для обозначения разрыва строки в текстовых файлах. Далее задаем параметр safecrlf со значением warn, так Git будет проверять преобразование на обратимость (рис. 1.3)

```
kanilova@dk3n62 ~ $ git config --global core.quotePath false
kanilova@dk3n62 ~ $ git config --global init.defaultBranch master
kanilova@dk3n62 ~ $ git config --global core.autocrlf input
kanilova@dk3n62 ~ $ git config --global core.safecrlf warn
kanilova@dk3n62 ~ $
```

{#fig:003 width=70%} Рис. 1.3. Настройка кодировки, создание имени для начальной ветки и выполнение параметров: autocrlf, safecrlf

2. Создание SSH ключа.

1. Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый). Вводим команду ssh-keygen -C "Имя Фамилия, work@email", указывая имя и корпоративную почту владельца. Ключи сохраняются в каталоге ~/.ssh/.(рис.1.4).

```
kanilova@dk3n62 ~ $ ssh-keygen -C "kanilova 1132230797@pfur.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/k/a/kanilova/.ssh/id_rsa):
/afs/.dk.sci.pfu.edu.ru/home/k/a/kanilova/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/k/a/kanilova/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/k/a/kanilova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:8It9WqPp5C49B/xwX1yn9rXB83p+pgsqY1YnUVPsRb4 kanilova 1132230797@pfur.ru
The key's randomart image is:
+---[RSA 3072]-----+
|
|      +o|
|      o.o|
|      .oo|
|      o  =|
|      .S o.E.|
|      o+.. o *+o|
|      ..+o+o = +*|
|      .ooB=.. o.=|
|      +B+oo.oo=o|
+---[SHA256]-----+
```

Рис.1.4. Создание SSH ключа.

2. Используем xclip утилиту, позволяющую скопировать любой текст через терминал. Для начала, в дистрибутиве Linux Ubuntu, ее необходимо установить с помощью команды apt-get install с ключом -у от имени суперпользователя, введя в начале команды sudo (рис. 1.5.). Копируем

открытый ключ из директории, в которой он был сохранен, с помощью утилиты xclip.

```
+-----[SHA256]-----+
kanilova@dk2n25 ~ $ sudo apt-get install -y xclip
```

Рис.1.5. Утилита xclip.

3. Открываем браузер, заходим на сайт <https://github.com/>. Открываем профиль и выбираем страницу «SSH and GPG keys» кнопку «New SSH key». Вставляем скопированный ключ в поле «Key». В поле Title указываем имя для ключа. Нажимаем «Add SSH-key», чтобы завершить добавление ключа (рис. 1.6.).

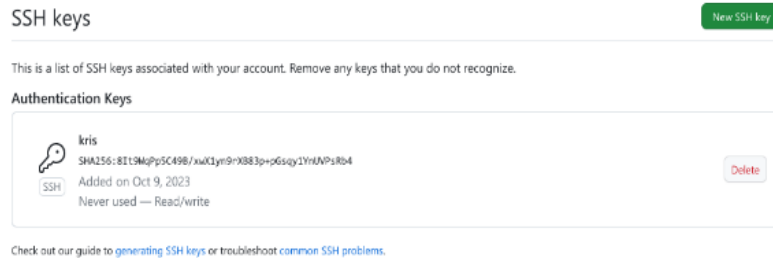


Рис.1.6. Добавление ключа.

3. Сознание рабочего пространства и репозитория курса на основе шаблона

1. Открываем терминал. Создаем директорию, рабочее пространство, с помощью утилиты mkdir, благодаря ключу -p создаем все директории после домашней ~/work/study/2023-2024/“Computer architecture” рекурсивно. Далее проверяем с помощью ls, создание необходимых каталогов (рис. 1.7.).

```
kanilova@dk2n25 ~ $ mkdir -p ~/work/study/2023-2024/"Computer architecture"
kanilova@dk2n25 ~ $ ls
chinp. dir1 dir2 hosts parentdir parentdir2 public tip bugso Разработка Общедоступные Шаблоны
chinp.pub dir2 documents newdir1 parentdir1 parentdir3 public_html work загрузки Музыка "Рабочий стол"
```

Рис.1.7.Создание рабочего

пространства

4. Создание репозитория курса на основе шаблона.

1. В браузере переходим на страницу репозитория с шаблоном курса по адресу <https://github.com/yamadharma/course-directory-student-template>. Далее выбираем «Use this template», чтобы использовать этот шаблон для репозитория (рис. 1.8.).

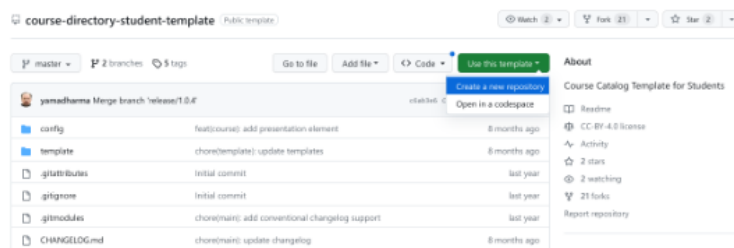


Рис.1.8. Страница репозитория

с шаблоном курса.

2. Задаем имя репозитория (Repository name): study_2022– 2023_arch- pc и создаем репозиторий, нажав на кнопку «Create repository» (рис. 1.9.).

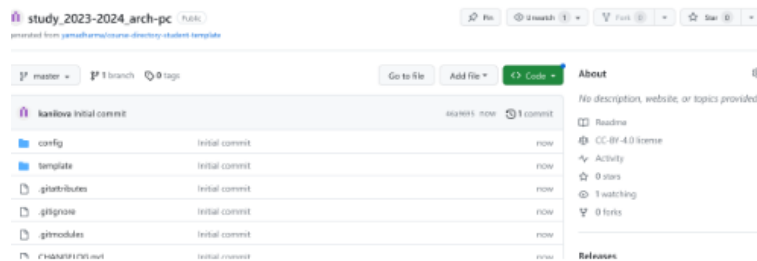


Рис.1.9. Созданный

репозиторий.

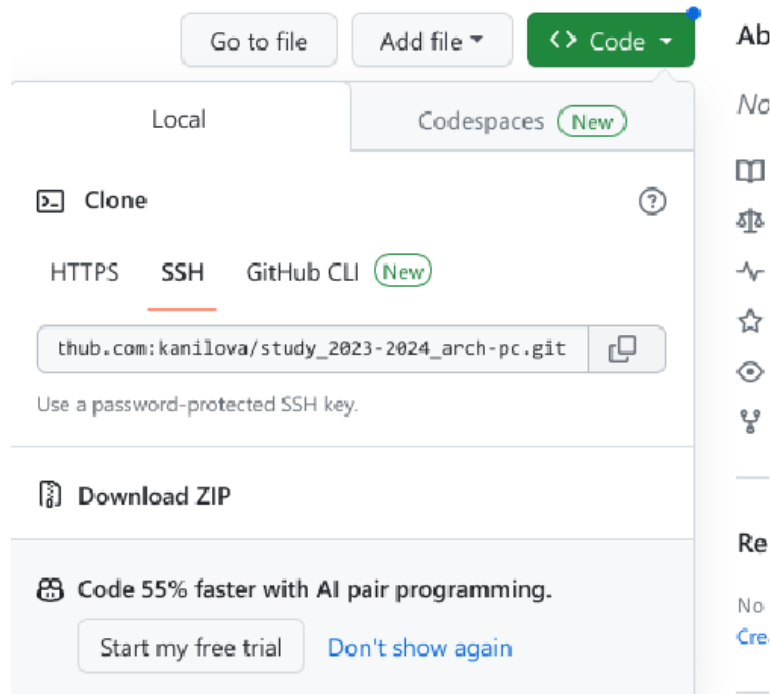
3. Через терминал переходим в созданный каталог курса с помощью утилиты `cd` (рис. 1.10.).

```
kanilova@dk3n62 ~ $ cd ~/work/study/2023-2024/"Computer architecture"
kanilova@dk3n62 ~/work/study/2023-2024/Computer architecture $
```

Рис.1.10. Перемещение между

директориями.

4. Копируем ссылку для клонирования на странице созданного репозитория, сначала перейдя в окно «code», далее выбрав в окне вкладку «SSH» (рис. 1.11.).



5. Клонировем созданный репозиторий с помощью команды `git clone -recursive git@github.com:/study_2023-2024_arh-pc.git arch-pc` (рис. 1.12.).

```
kanilova@dk3n62 ~ $ git clone --recursive git@github.com:kanilova/study_2023-2024_arch-pc.git arch-pc
Клонирование в arch-pc...
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 KiB | 125.00 KiB/c, готово.
Определение изменений: 100% (1/1), готово.
Получены <template/presentation> (https://github.com/yasashirna/academic-presentation-markdown-template.git) сверхвысокая скорость на нуте <template/presentation>
Получены <template/report> (https://github.com/yasashirna/academic-laboratory-report-template.git) сверхвысокая скорость на нуте <template/report>
Клонирование в /afs/dk.aci.pfu.edu.ru/home/ka/kanilova/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 20), reused 77 (delta 13), pack-reused 0
Получение объектов: 100% (82/82), 42.98 KiB | 178.00 KiB/c, готово.
Определение изменений: 100% (26/26), готово.
Клонирование в /afs/dk.aci.pfu.edu.ru/home/ka/kanilova/arch-pc/template/report...
remote: Enumerating objects: 101, done.
remote: Counting objects: 100% (101/101), done.
remote: Compressing objects: 100% (78/78), done.
remote: Total 101 (delta 48), reused 88 (delta 27), pack-reused 0
Получение объектов: 100% (101/101), 327.25 KiB | 137.00 KiB/c, готово.
Определение изменений: 100% (40/40), готово.
Submodule path 'template/presentation': checked out 'b1be1800ee91f5a89294c4375d3161743406751e'
Submodule path 'template/report': checked out 'f1b61d4ac9c28fa83917b82e3aff1a31b1e3b2'
kanilova@dk3n62 ~ $
```

5. Настройка каталога курса.

1. Переходим в каталог arch-pc с помощью утилиты cd (рис. 1.13.).

```
kanilova@dk3n62 ~/work/study/2023-2024/Computer architecture $ cd ~/work/study/2023-2024/Computer architecture/arch-pc
kanilova@dk3n62 ~/work/study/2023-2024/Computer architecture/arch-pc $
```

2. Удаляем лишние файлы с помощью утилиты rm. Создаем необходимые каталоги (рис. 1.14).

```
kanilova@dk3n62 ~/work/study/2023-2024/Computer architecture/arch-pc $ rm package.json
kanilova@dk3n62 ~/work/study/2023-2024/Computer architecture/arch-pc $ echo arch-pc > COURSE
kanilova@dk3n62 ~/work/study/2023-2024/Computer architecture/arch-pc $ make
```

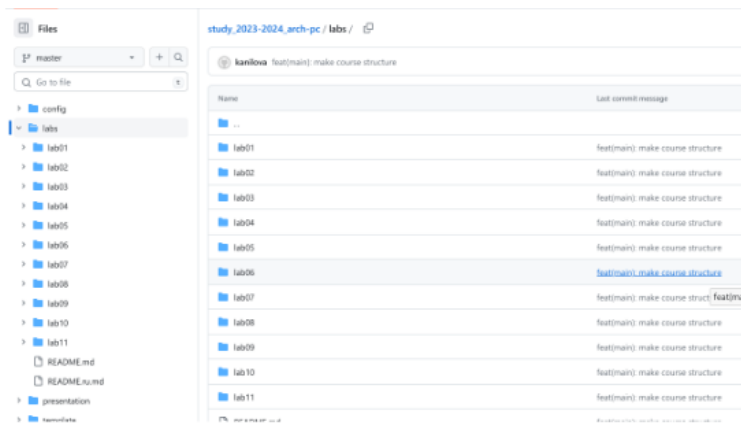
3. Отправляем созданные каталоги с локального репозитория на сервер: добавляем все созданные каталоги с помощью git add, комментируем и сохраняем изменения на сервере, как добавление курса с помощью git commit (рис. 1.15.).

```
kanilova@dk3n62 ~/work/study/2023-2024/Computer architecture/arch-pc $ git add .
kanilova@dk3n62 ~/work/study/2023-2024/Computer architecture/arch-pc $ git commit -am 'feat(main): make course structure'
[master 476e3bc] 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/placing_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-r-b-y-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
```

4. Отправляем все на сервер с помощью push (рис. 1.16.).

```
kanilova@dk3n62 ~/work/study/2023-2024/Computer architecture/arch-pc $ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 6 потоков.
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.13 КиБ | 2.46 МБ/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:kanilova/study_2023-2024_arch-pc.git
  46a9695..476e3bc master -> master
kanilova@dk3n62 ~/work/study/2023-2024/Computer architecture/arch-pc $
```

5. Проверяем правильность выполнения работы на самом сайте GitHub (рис. 1.17.).



6. Самостоятельная работа.

1. Переходим в директорию labs/lab02/report с помощью утилиты cd. Создаем в каталоге файл для отчета по второй лабораторной работе с помощью утилиты touch (рис.1.17.).

```
kanilova@dk3n62 ~ $ cd ~/work/study/2023-2024/Computer_architecture/arch-pc/labs/lab02/report
kanilova@dk3n62 ~/work/study/2023-2024/Computer_architecture/arch-pc/labs/lab02/report $ touch L02_Kanilova_Report
kanilova@dk3n62 ~/work/study/2023-2024/Computer_architecture/arch-pc/labs/lab02/report $
```

Рис.1.17. Создание файла

2. Оформляем отчет в текстовом процессоре LibreOffice Writer. После открытия текстового редактора, открываем в нем созданный файл.
3. Переходим из подкаталога lab02/report в подкаталог lab01/report с помощью утилиты cd (рис. 1.19.).

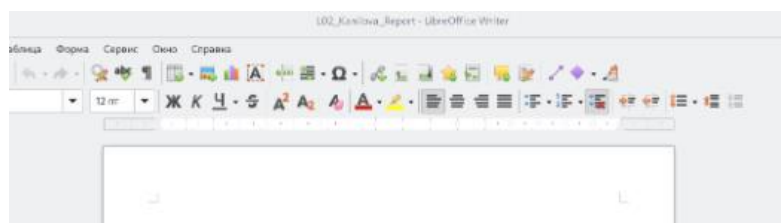


Рис.1.19. Перемещение между

директориями

4. Проверяем местонахождение файлов с отчетами по первой лабораторной работе. Она должна быть в подкаталоге домашней директории «Documents», для проверки используем команду ls (рис. 1.20.).

```
kanilova@dk3n62 ~/work/study/2023-2024/Computer_architecture/arch-pc/labs/lab02/report $ cd ..
kanilova@dk3n62 ~/work/study/2023-2024/Computer_architecture/arch-pc/labs/lab02 $ cd ..
kanilova@dk3n62 ~/work/study/2023-2024/Computer_architecture/arch-pc/labs $ cd lab01
kanilova@dk3n62 ~/work/study/2023-2024/Computer_architecture/arch-pc/labs/lab01 $ cd report
kanilova@dk3n62 ~/work/study/2023-2024/Computer_architecture/arch-pc/labs/lab01/report $
```

Рис.1.20. Проверка

местонахождения файлов.

5. Добавляем файл L01_Kanilova_Report.pdf. Сохраняем изменения на сервере командой git commit -m "...". То же самое делаем для отчета по второй лабораторной работе. Переходим в директорию labs/lab02/report с помощью cd, добавляем с помощью git add нужный файл, сохраняем изменения с помощью git commit (рис. 1.22./1.23.)

```
kanilova@dk3n62 ~ $ cd ~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab01/report
kanilova@dk3n62 ~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab01/report $ git add L01_Kanilova_Report.pdf
kanilova@dk3n62 ~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab01/report $ git commit -m "Add existing files"
[master cd513ac] Add existing files
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/L01_Kanilova_Report.pdf
```

6. Отправляем в центральный репозиторий сохраненные изменения командой `git push -f origin master` (рис. 1.24.)

```
kanilova@dk3n62 ~ $ cd ~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab02/report
kanilova@dk3n62 ~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab02/report $ ls
lab01 image L02_Kanilova_Report Makefile pandoc report.md
kanilova@dk3n62 ~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab02/report $ git add L02_Kanilova_Report
kanilova@dk3n62 ~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab02/report $ git commit -m "Add existing files"
[master 576c740] Add existing files
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab02/report/L02_Kanilova_Report
kanilova@dk3n62 ~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab02/report $
```

Рис.1.24. Отправка изменений

в центральный репозиторий.

7. Проверяем на сайте <https://github.com/> правильность выполнения заданий.(рис. 1.25.)

```
kanilova@dk3n62 ~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab02/report $ git push -f origin master
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сканировании изменений используется до 6 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 477 байтов | 477.00 Кб/с, готово.
Всего 5 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:kanilova/study_2023-2024_arch-pc.git
476e3bc..576c740 master -> master
```

8. Просматриваем изменения. Видим, что добавлены файлы с отчетами по лабораторным работам.(рис. 1.26.).

| Name | Last commit message |
|-------|---------------------|
| .. | |
| lab01 | Add existing files |
| lab02 | Add existing files |

Рис.1.26. Проверка добавления

отчетов по лабораторным работам.

9. Проверяем, находятся ли отчеты по лабораторным работам в нужных каталогах репозитория. (рис.1.27.)

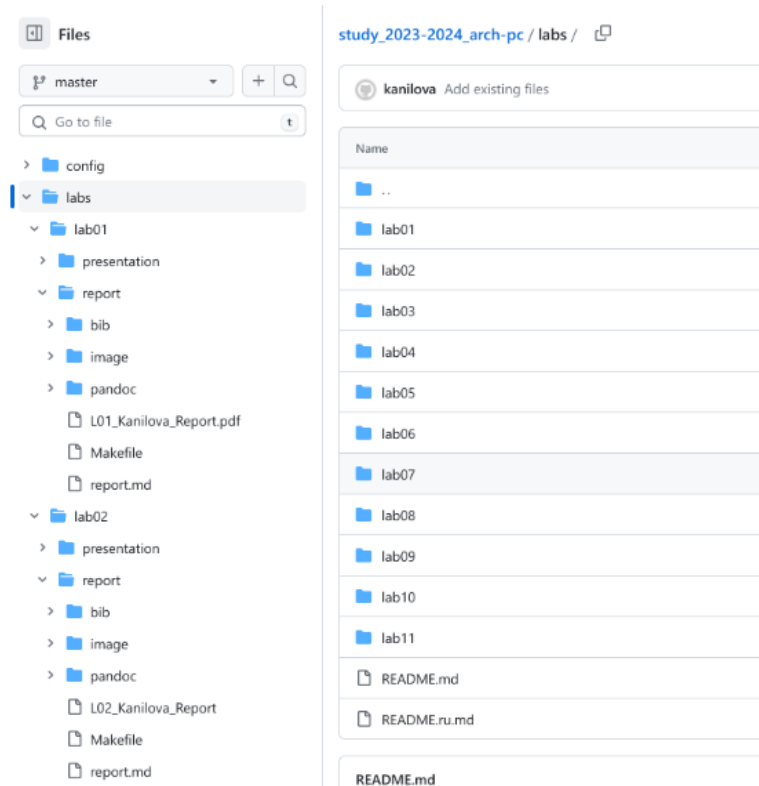


Рис.1.27. Каталог lab01/report.

Вывод

В ходе выполнения лабораторной работы изучены идеология и применение средств контроля версий. Приобретены практические навыки по работе с системой git.