

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
на тему «Система контроля Git»
дисциплина: Архитектура компьютера

Студент: Дагделен З.Р.

Группа: НКАбд-02-23

МОСКВА
2023 г.

Содержание

1 Цель работы	3
2 Задание.....	4
3 Теоретическое введение.....	5
4 Выполнение лабораторной работы.....	7
5 Выводы.....	19
6 Список литературы	20

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 и репозитория курса на основе шаблона
5. Создание рабочего курса на основе шаблона
6. Настройка каталога курса
7. Выполнение заданий для сам.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.

Система контроля версий Git представляет собой набор программ командной строки.

Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями.

Благодаря тому, что *Git* является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

4. Выполнение лабораторной работы

1. Настройка github

Существует несколько доступных серверов репозитория с возможностью бесплатного размещения данных.

Для выполнения лабораторных работ буду использовать Github.

Создаю учётную запись на сайте <https://github.com/> и заполняю основные данные. Аккаунт создан (рис.1.1)

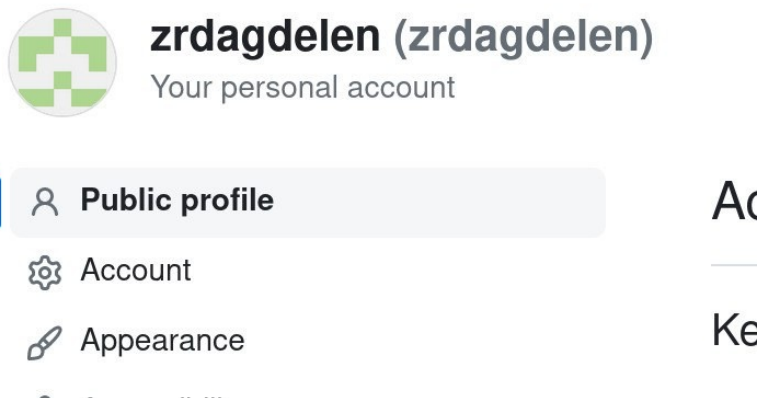


Рис. 1.1 аккаунт GitHub

2. Базовая настройка git

Сначала сделаю предварительную конфигурацию git. Открываю терминал и ввожу следующие команды, указав свое имя/фамилию и email(рис. 2.1):

```
zrdagdelen@zrdagdelen:~$ git config --global user.name "<Zeynap Dagdelen>"
zrdagdelen@zrdagdelen:~$ git config --global user.email "<1132236052@pfur.ru>"
zrdagdelen@zrdagdelen:~$
```

Рис. 2.1 Предварительная конфигурация git

Настраиваю utf-8 в выводе сообщений git для корректного отображения символов и задаю имя начальной ветки (буду называть её master)(рис. 2.2):

```
zrdagdelen@zrdagdelen:~$ git config --global core.quotePath false
zrdagdelen@zrdagdelen:~$ git config --global init.defaultBranch master
```

Рис. 2.2 Настройка кодировки (1 строка) и создание имени для начальной ветки (2 стр)

Задаю параметр `autocrlf` со значением `input`, так как я работаю в системе Linux, чтобы конвертировать CRLF в LF только при коммитах (рис.2.3). CR и LF — это символы, которые можно использовать для обозначения разрыва строки в текстовых файлах.

Задаю параметр `safecrlf` со значением `warn`, потому Git будет проверять преобразования на обратимость (рис.2.3). При значении `warn` Git только выведет предупреждение, но будет принимать необратимые конвертации.

```
zrdagdelen@zrdagdelen:~$ git config --global core.autocrlf input
zrdagdelen@zrdagdelen:~$ git config --global core.safecrlf warn
zrdagdelen@zrdagdelen:~$
```

Рис. 2.3. параметр `autocrlf` (1 стр) и параметр `safecrlf` (2 стр)

3. Создание SSH ключа

Для последующей идентификации пользователя на сервере репозитория мне необходимо сгенерировать пару ключей (приватный и открытый). Потому ввожу команду `ssh-keygen -C "Имя Фамилия <work@mail>"`, указывая имя и почту владельца(рис.).

Благодаря этой команде ключи сохраняться в каталоге `~/.ssh/`.

```
zrdagdelen@zrdagdelen:~$ ssh-keygen -C "Zeynap Dagdelen <1132236052@pfur.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/zrdagdelen/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/zrdagdelen/.ssh/id_rsa
Your public key has been saved in /home/zrdagdelen/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:paTzsdqTghn/lhrpcop9RCN2YjQ0hr2VB1wx+ZR0Dg8 Zeynap Dagdelen <1132236052@pfur.ru>
The key's randomart image is:
+---[RSA 3072]-----+
| o ..+++E..          |
| . + * o.o*          |
| . * o o. +          |
| . * +o.o            |
| o =o.S              |
| . .+ o              |
| *o oo               |
| o+.,.=+             |
| . o+=+o.            |
+---[SHA256]-----+
```

Рис. 3.1. Генерация SSH-ключа

Далее мне необходимо загрузить сгенерированный открытый ключ. Для этого зайти на сайт <http://github.org/> под своей учётной записью и

перейти в меню Setting . После этого выбираю в боковом меню SSH and GPG keys и нажимаю кнопку New SSH key. Скопировав из локальной консоли ключ в буфер обмена при помощи `cat ~/.ssh/id_rsa.pub | xclip -sel clip` (рис. 3.3) вставляю ключ в появившееся на сайте поле и указываю для ключа имя (Title)(рис. 3.4). (Так как в Linux Ubuntu мне сначала нужно установить команду xclip, позволяющую скопировать любой текст через терминал, ввожу `sudo apt install xclip`(рис.3.2))

```
Команда «xclip» не найдена, но может быть установлена с помощью:
sudo apt install xclip
zrdagdelen@zrdagdelen:~$ sudo apt install xclip
[sudo] пароль для zrdagdelen:
Попробуйте ещё раз.
[sudo] пароль для zrdagdelen:
Чтение списков пакетов... Готово
```

Рис. 3.2. Установка команды xclip

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

Рис. 3.3 Копирование ключа в буфер обмена

Add new SSH Key

Title

zrdagdelen2209

Key type

Authentication Key

Key

```
AAAAB3NzaC1yc2EAAAADAQABAAQgQDRD3oRTToAahvS5fe03vuVLfPQW9PANhArxTgYScptDrku8/vxFdoMbpeHF
E99ldlovZf/quARALZYbukOnqluGCwIFkaKkeXvBEwI70jEDM/TULYF/n1xB3iT9V6Q8dL
/LxN7+CFFfGTUq3tZRf8jRRhJLyoNVcTX3hxWtR56uMuwVuHMR6xgLGzaSSn8BuG6HzzR4C557WRn7ruR+enZLfqd0
hKc7zxEGT5ot/oSs7y1RXGXw7GjSDzO8Zed5+nXLiBb53/2nBrPnTaqUZP9oi1D
/aCn8yFZrOqGOgM6OZnAyLwjBITHWH/Ak7cjwaiVQqnT3D0qsRTvW03xihNeTW
/bFtJM1kSis1NHc8GboTLdtayKmWvWns6bCRYiODQ+bsZtx4mMbaj9t2dno4vVsdY8aHkGUyTOJTnYuPRvY78GhNfKR
HVb0kq6W6d/fpEwKLbDFAEwlgUJlix6BmU8EYd485B/gyJKkFLXu5SKyzSujSyQhS3QWe6Kmqz+dq8c= Zeynap
Dagdelen <1132236052@pfur.ru>
```

Add SSH key

Рис. 3.4 Вставка ключа

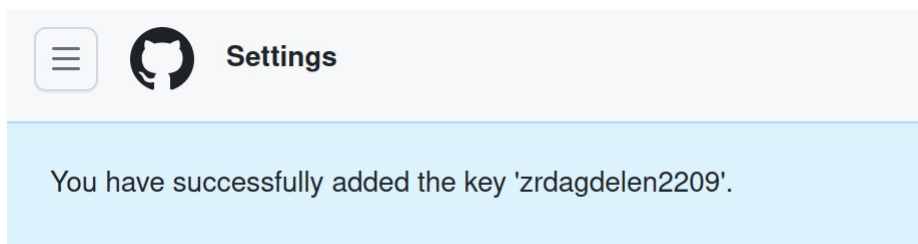


Рис. 3.5. Успешное сохранение ключа

4. Создание рабочего пространства и репозитория курса на основе шаблона

Открываю терминал и создаю каталог для предмета «Архитектура компьютеров» при помощи команды `mkdir` и ключу `-p` (рис. 4.1).

```
zrdagdelen@zrdagdelen:~$ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
zrdagdelen@zrdagdelen:~$ ls
snap  Видео  Загрузки  Музыка  'Рабочий стол'
work  Документы  Изображения  Общедоступные  Шаблоны
```

Рис. 4.1 Рекурсивное создание рабочего пространства + проверка работы команды

5. Создание репозитория курса на основе шаблона

Репозиторий на основе шаблона можно создать через web-интерфейс github. Перехожу на страницу репозитория с шаблоном курса.

Далее выбираю «Use this template» (рис.5.1).

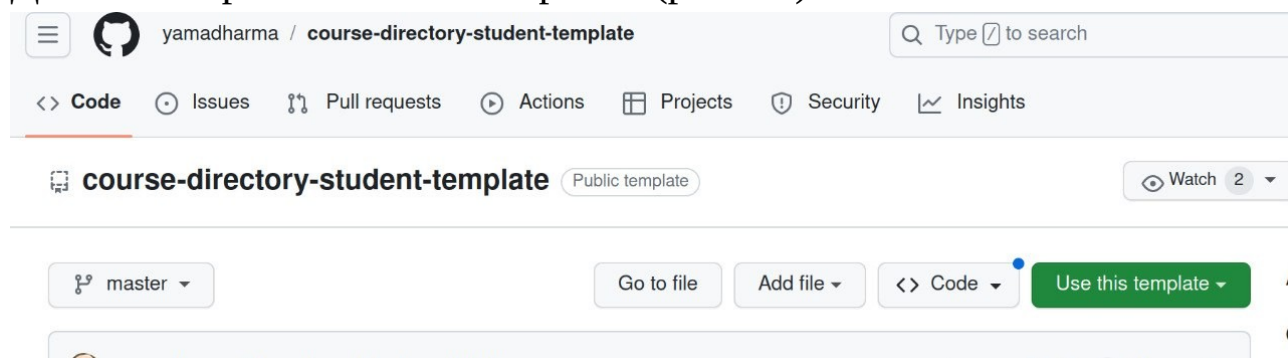


Рис.5.1. Страница репозитории с шаблоном курса.


В открывшемся окне задаю имя репозитория (Repository name) `study_2023–2024_arh-рс` и создаю репозиторий (с помощью кнопки Create repository from template) (рис. 5.2)

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 *

 zrdagdelen ▾

Repository name *

/ study_2023-2024_arh-pc

⚠ Your new repository will be created as **study_2023-2024_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 **psychic-garbanzo** ?

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.5.2. Создание репозитории

Открываю терминал и перехожу в каталог курса с помощью команды `cd ~/work/study/2023-2024/'Архитектура компьютера'`. Клонировую созданный репозиторий с помощью команды `git clone --recursive **ссылка для клонирования** arch-pc` (рис.5.3), ссылку для клонирования скопировав на странице созданного репозитория Code -> SSH(рис.5.4). Проверяем(рис.5.5)

```
zrdagdelen@zrdagdelen:~$ cd ~/work/study/2023-2024/'Архитектура компьютера'
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера$ git clone -
-recursive git@github.com:zrdagdelen/study_2023-2024_arh-pc.git arch-pc
Клонирование в «arch-pc»...
remote: Enumerating objects: 27, done.
```

Рис.5.3. Перемещение по каталогам и клонирование репозиторий

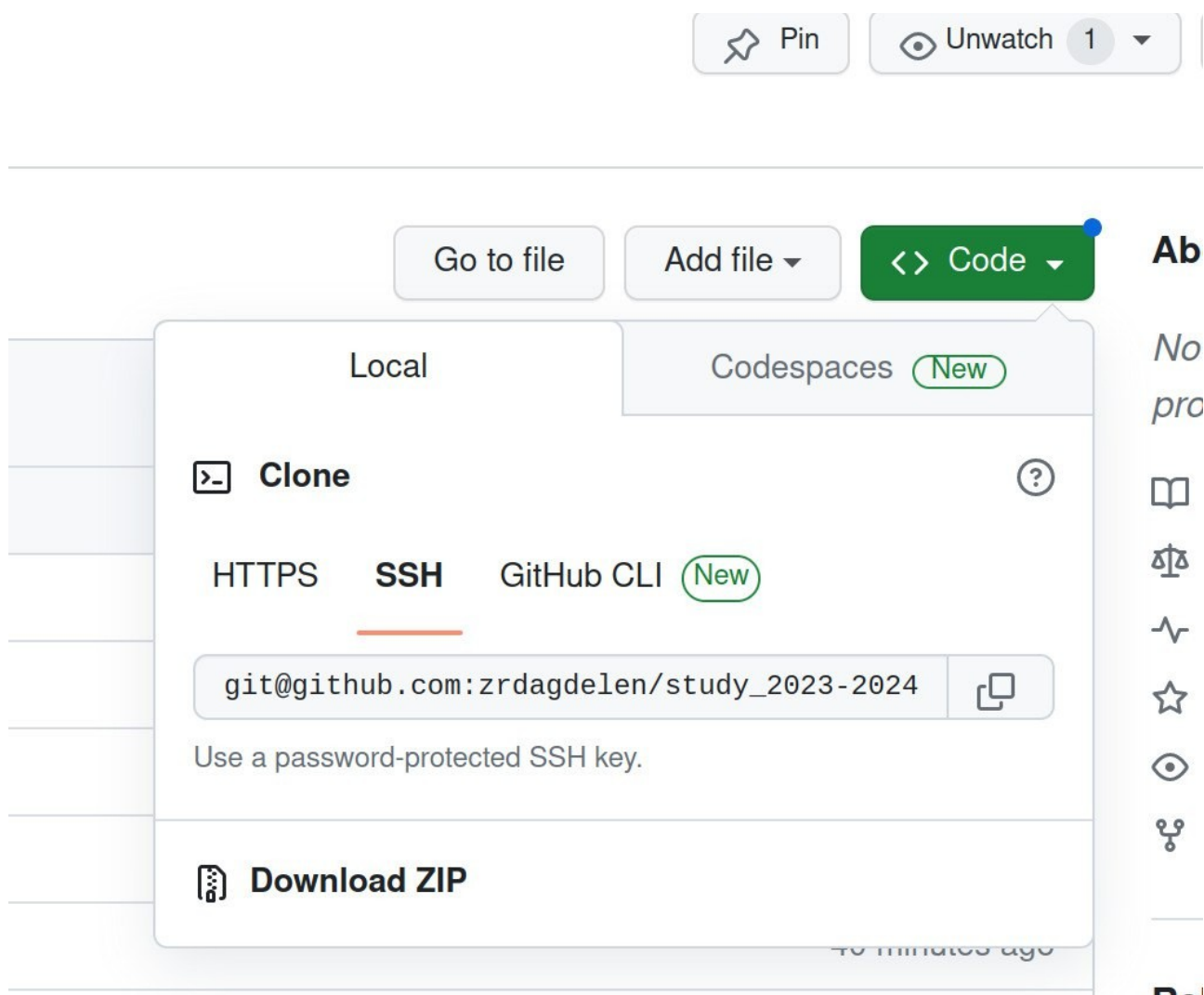


Рис.5.4. Ссылка для клонирования

```
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера$ ls  
arch-pc  
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера$
```

Рис.5.5. Проверка работы команды

6. Настройка каталога курса

Перехожу в каталог курса с помощью `cd` и удаляю файл `package.json` с помощью команды `rm`, проверяем (рис. 6.1)


```

zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера$ cd arch-pc
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ rm
package.json
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ ls
CHANGELOG.md  COURSE  Makefile  README.git-flow.md  template
config        LICENSE  README.en.md  README.md
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc$

```

Рис.6.1. Удаление файла+проверка

Создаю необходимые каталоги (рис.6.2) и отправляю файлы на сервер (рис.6.3)

```

zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура ком
пьютера/arch-pc$ echo arch-pc > COURSE
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура ком
пьютера/arch-pc$ make
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура ком
пьютера/arch-pc$ ls
CHANGELOG.md  labs      prepare  README.git-flow.md
config        LICENSE  presentation  README.md
COURSE        Makefile  README.en.md  template
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура ком
пьютера/arch-pc$

```

Рис.6.2. Создание каталогов и проверка работы команд

```

пьютера/arch-pc$ git add .
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура ком
пьютера/arch-pc$ git commit -am 'feat(main): make course str
ucture'
[master eb9990f] 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.j
pg
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

```

Рис.6.3. Отправка файлов на сервер

```

create mode 100644 presentation/report/report.md
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.13 КиБ | 2.67 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:zrdagdelen/study_2023-2024_arh-pc.git
c6bb3a6..eb9990f master -> master
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc$

```

Рис.6.4. Отправка файлов на сервер

Проверяю правильность создания иерархии рабочего пространства в локальном репозитории и на странице github(рис.6.5)

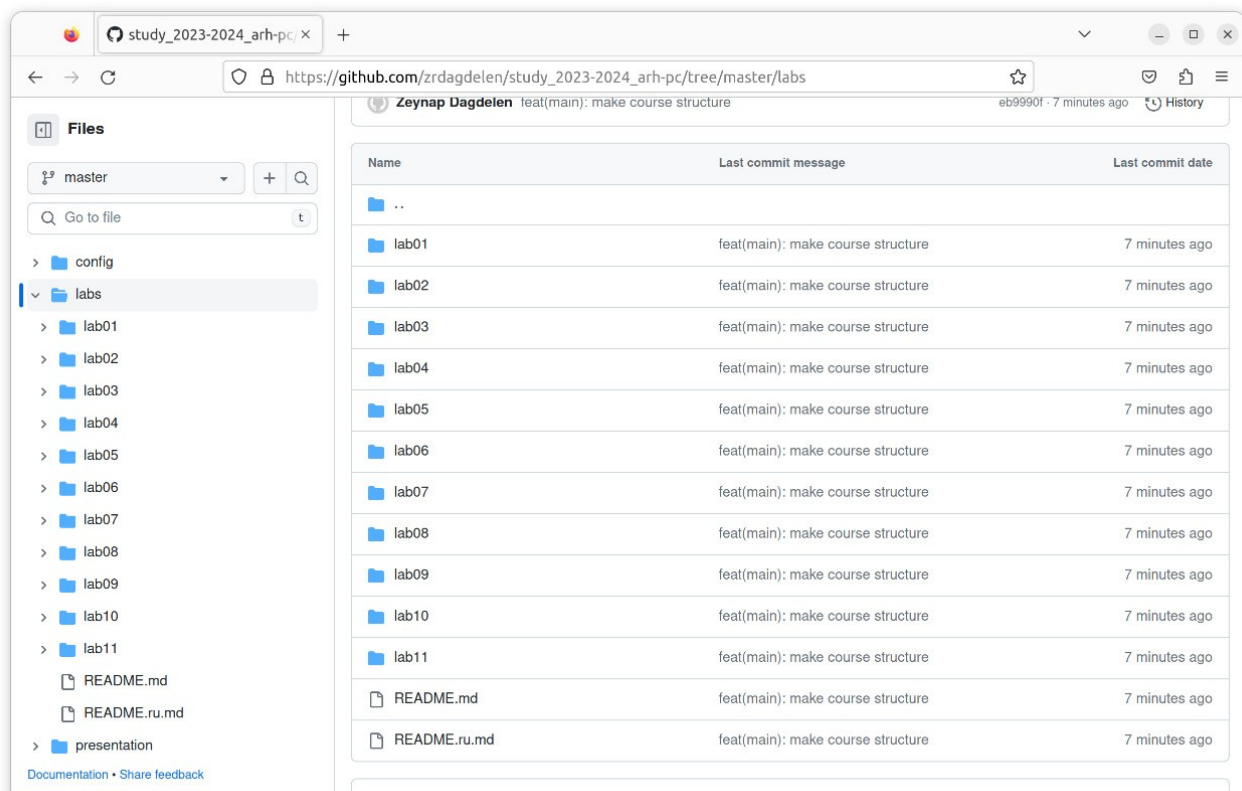


Рис.6.5. Проверка

7. Выполнение самостоятельной работы

С помощью команды `cd` перемещаюсь в подкаталог `labs/lab02/report` и создаю текстовый файл с форматом `odt` (чтобы потом `chfpe ;t` работать в текстовом редакторе LibreOffice Writer)(рис.7.1)

```
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ cd labs/lab02/report
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$ touch л02_Дагделен_отчет_арх.odt
```

Рис.7.1. Перемещение и создание файла `odt`

Захожу в домашнюю папку, открываю нужный подкаталог и нахожу только что созданный файл, пишу туда любое слово и сохраняю, чтобы файл стал формата `odt` и при последующем его открытии он сразу же открывался в приложении(рис.7.2-7.3)

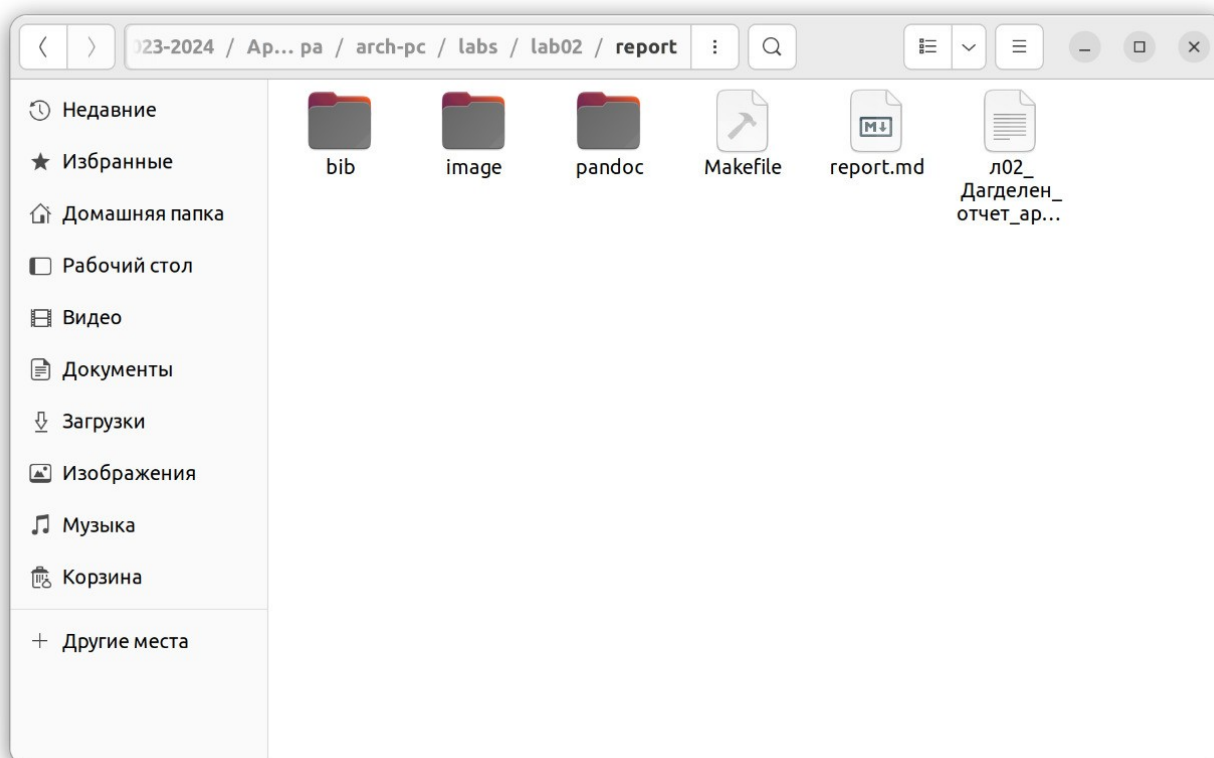


Рис.7.2. Созданный файл в домашней папке

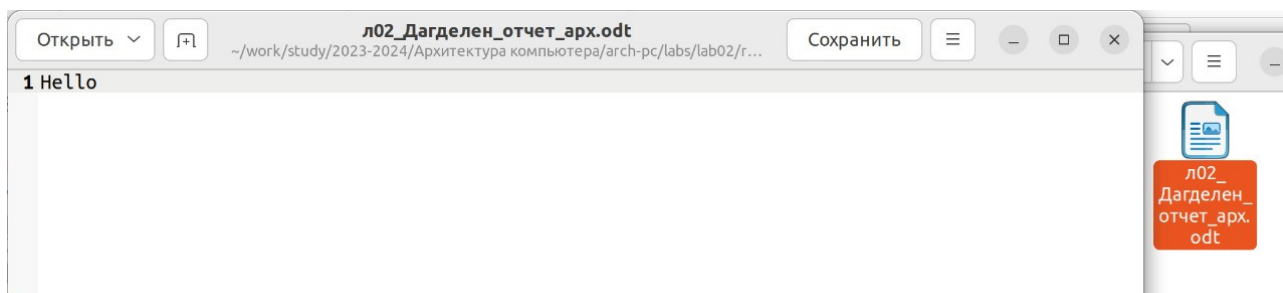


Рис.7.3. Файл odt

С помощью `cd` перемещаюсь между каталогами и перехожу в нужный (рис.7.4) для того, чтобы поместить туда лабораторную работу №1, которая находится у меня в загрузках(рис.7.5).

```
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$ cd ..
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02$ cd ..
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs$ cd lab01
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01$ cd report
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$
```

Рис.7.4. Перемещение между каталогами

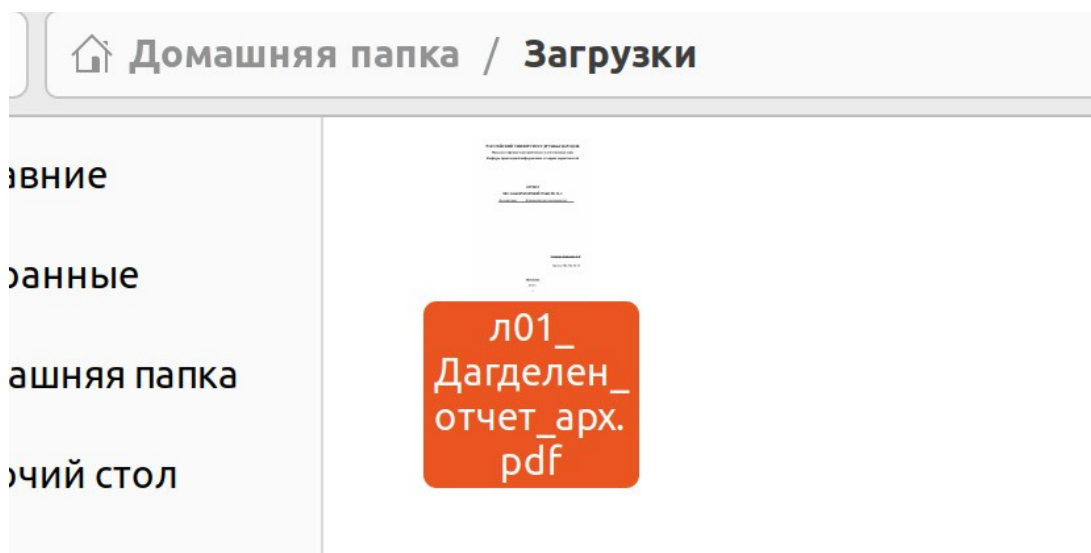


Рис.7.5. Сохраненная лабораторная №1

Копирую лабораторную №1 из загрузок в нужную папку(рис.7.6)

```
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ cp ~/Загрузки/л01_Дагделен_отчет_арх.pdf ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ ls
bib      Makefile  report.md
image    pandoc    л01_Дагделен_отчет_арх.pdf
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$
```

Рис.7.6. Копирование файла из одного места в другое

Добавляю в коммит созданный файл с лабораторным №1(рис.7.7)

```
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ git add л01_Дагделен_отчет_арх.pdf
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$
```

Рис.7.7. Добавление файла на сервер

Перемещаюсь между директориями, добавляю вторую лабораторную, сохраняю изменения с помощью git commit(рис.7.8).

```
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ cd ..
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01$ cd ..
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs$ cd lab02/report
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$ git add л02_Дагделен_отчет_арх.odt
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$ git commit -m 'Add existing file'
[master ecd5387] Add existing file
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 labs/lab01/report/л01_Дагделен_отчет_арх.pdf
 create mode 100644 labs/lab02/report/л02_Дагделен_отчет_арх.odt
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$
```

Рис.7.8. Добавление второй лабораторной и сохранение изменений

Отправляю в центральный репозиторий сохраненные изменения командой `git push -f origin master`(рис.7.9). Проверяю на сайте (рис.7.10, рис.7.11)

```
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$ git push -f origin master
Перечисление объектов: 15, готово.
Подсчет объектов: 100% (13/13), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (9/9), готово.
Запись объектов: 100% (9/9), 2.25 МиБ | 2.52 МиБ/с, готово.
Всего 9 (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:zrdagdelen/study_2023-2024_arh-pc.git
    eb9990f..ecd5387  master -> master
zrdagdelen@zrdagdelen:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$
```

Рис.7.9. Отправка сохраненных изменений

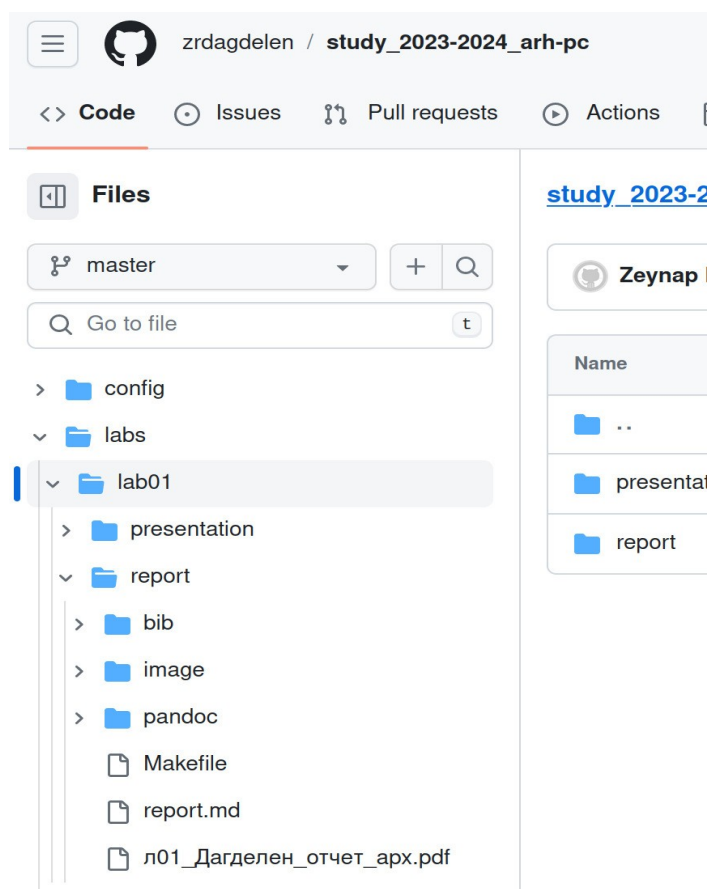


Рис.7.10. Проверка на сайте

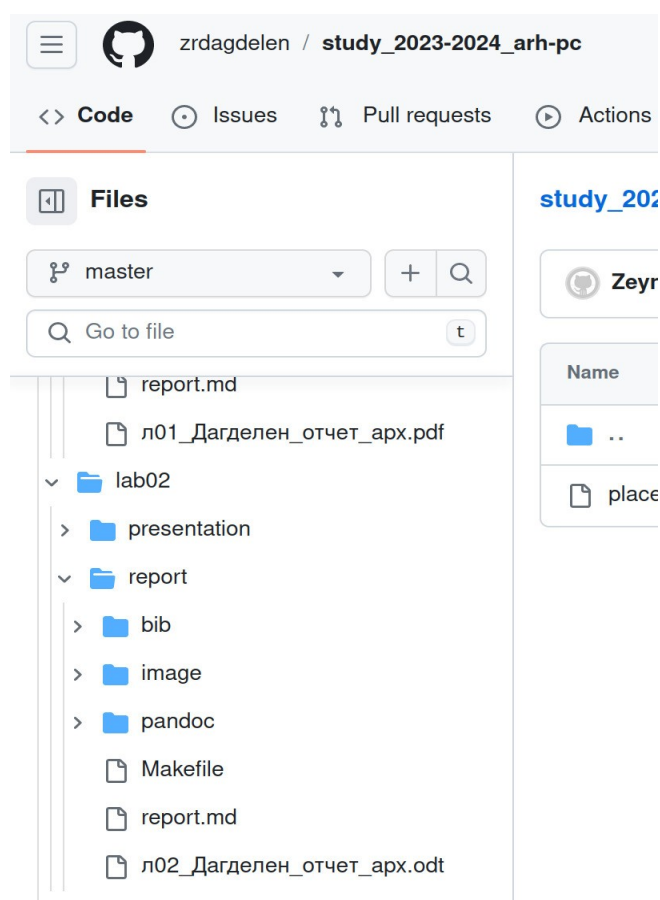


рис.7.11.Проверка на сайте

5 Выводы

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

6 Список литературы

[Архитектура ЭВМ](#)

[GitHub](#)