

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

дисциплина: Архитектура компьютеров
и операционных систем

Студент: Акмурадов Тимур

Группа: НБИбд-02-23

МОСКВА

2023 г

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Настройка GitHub	9
4.2	Базовая настройка Git	10
4.3	Создание SSH-ключа	11
4.4	Создание рабочего пространства и репозитория курса на основе шаблона.....	14
4.5	Создание репозитория курса на основе шаблона	15
4.6	Настройка каталога курса.....	17
4.7	Выполнение заданий для самостоятельной работы	20
5	Выводы	27
6	Список литературы	28

1 Цель работы

Целью данной работы является изучить идеологию и применение средств контроля версий, а также 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 репозитория курса на основе шаблона.
6. Настройка каталога курса.
7. Выполнение заданий для самостоятельной работ

3 Теоретическое введение

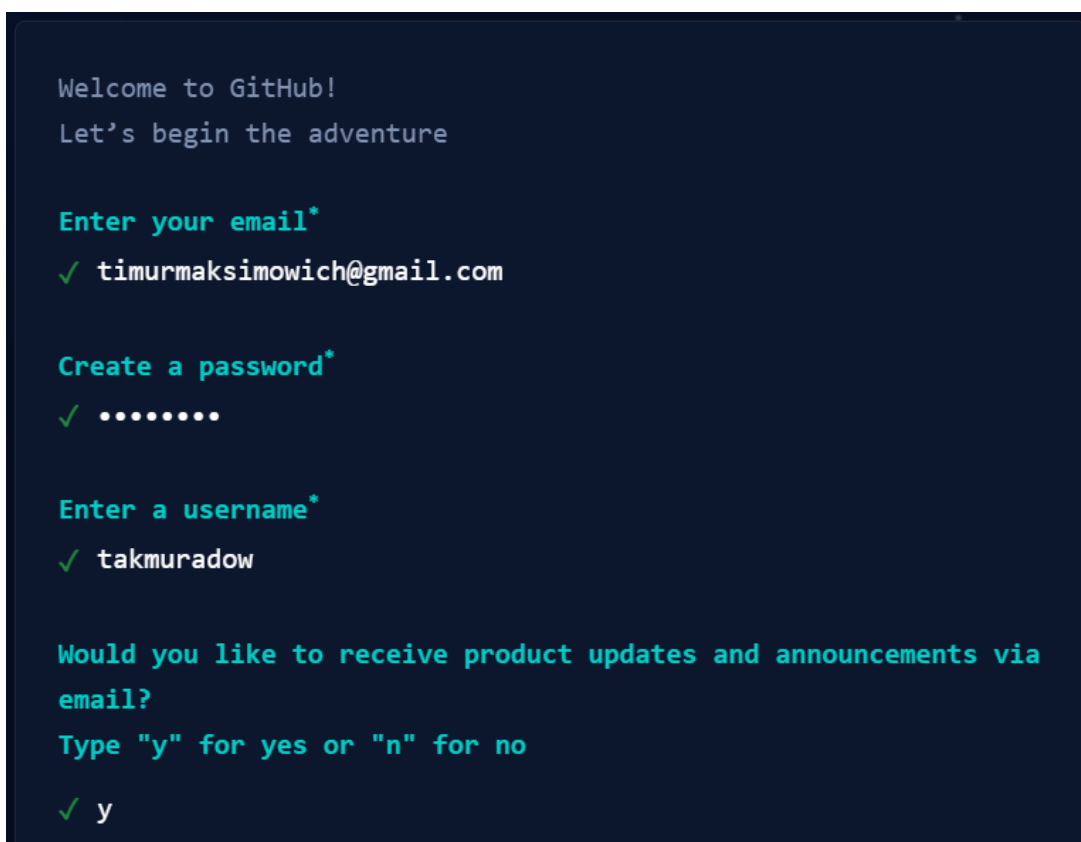
Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля 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 изменения, сделанные разными участниками, 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. 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. 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. Работа пользователя со своей веткой начинается с проверки и получения изменений из центрального репозитория (при этом в локальное дерево до начала этой процедуры не должно было вноситься изменений). Затем можно вносить изменения в локальном дереве и/или ветке. После завершения внесения какого-то изменения в файлы и/или каталоги проекта необходимо разместить их в центральном репозитории

4 Выполнение лабораторной работы

4.1 Настройка GitHub

Создаю учетную запись на сайте GitHub (рис. 4.1). Далее я заполняю основные данные учетной записи и регистрирую аккаунт.



```

Welcome to GitHub!
Let's begin the adventure

Enter your email*
✓ timurmaksimowich@gmail.com

Create a password*
✓ .....

Enter a username*
✓ takmuradow

Would you like to receive product updates and announcements via
email?
Type "y" for yes or "n" for no
✓ y

```

Рис. 4.1: Заполнение данных учетной записи GitHub

Аккаунт создан (рис. 4.2).

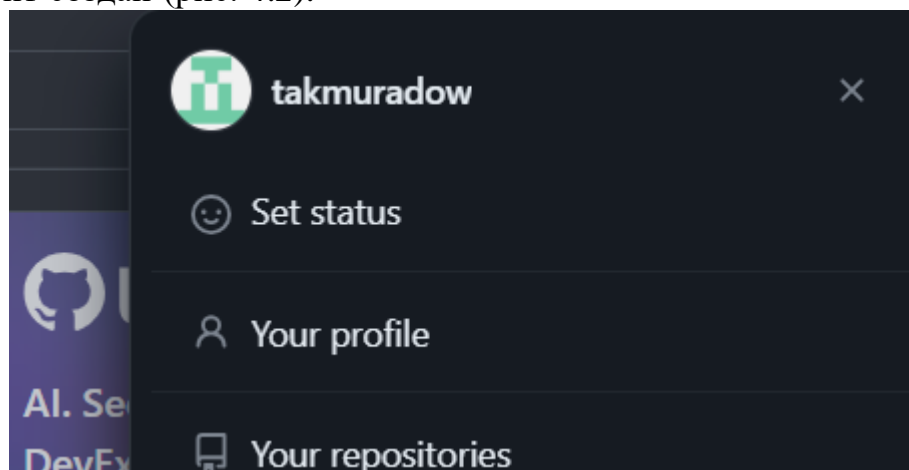
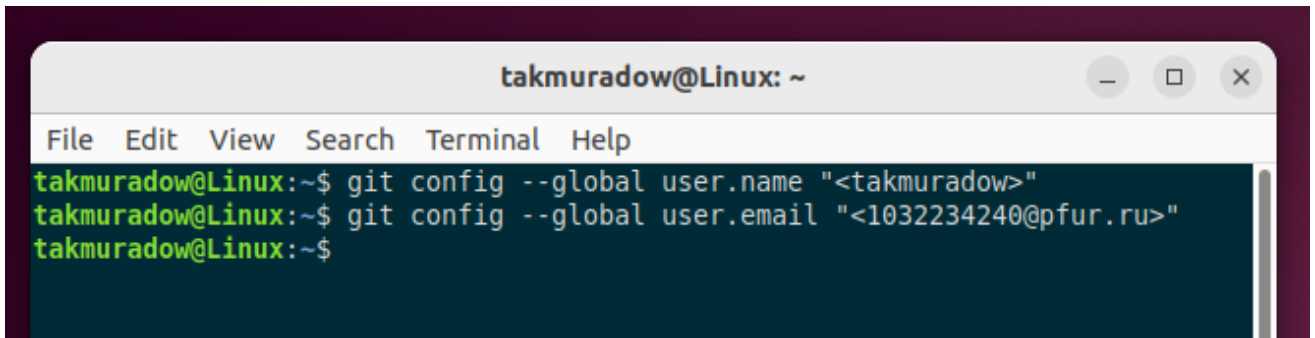


Рис. 4.2: Аккаунт GitHub

4.2 Базовая настройка Git

Запускаю виртуальную машину, затем в терминале задаю предварительную конфигурацию git. Ввожу команду `git config --global user.name ""`, указывая свое имя и команду `git config --global user.email "work@mail"`, указывая в ней электронную почту владельца, то есть мою (рис. 4.3).

A screenshot of a terminal window titled 'takmuradow@Linux: ~'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal shows three lines of commands and their output:

```
takmuradow@Linux:~$ git config --global user.name "<takmuradow>"
takmuradow@Linux:~$ git config --global user.email "<1032234240@pfur.ru>"
takmuradow@Linux:~$
```

Рис. 4.3: Предварительная конфигурация git

Настраиваю utf-8 в выводе сообщений git для корректного отображения символов (рис. 4.4).

```
takmuradow@Linux:~$ git config --global core.quotepath false
```

Рис. 4.4: Настройка кодировки

Задаю имя «master» для начальной ветки (рис. 4.5).

```
takmuradow@Linux:~$ git config --global init.defaultBranch master
```

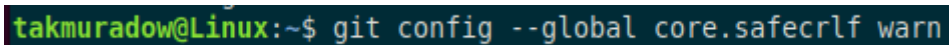
Рис. 4.5: Создание имени для начальной ветки

Задаю параметр `autocrlf` со значением `input`, так как я работаю в системе Linux, чтобы конвертировать CRLF в LF только при коммитах (рис. 4.6). CR и LF – это символы, которые можно использовать для обозначения разрыва строки в текстовых файлах.

```
takmuradow@Linux:~$ git config --global core.autocrlf input
```

Рис. 4.6: Параметр autocrlf

Задаю параметр `safecrlf` со значением `warn`, так Git будет проверять преобразование на обратимость (рис. 4.7). При значении `warn` Git только выведет предупреждение, но будет принимать необратимые конвертации.

A terminal window with a dark background. The prompt is 'takmuradow@Linux:~\$' in green. The command 'git config --global core.safecrlf warn' is entered in white text.

```
takmuradow@Linux:~$ git config --global core.safecrlf warn
```

Рис. 4.7: Параметр `safecrlf`

4.3 Создание SSH-ключа

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый). Для этого ввожу команду `ssh-keygen -C "Имя Фамилия, work@email"`, указывая имя владельца и электронную почту владельца (рис. 4.8). Ключ автоматически сохранится в каталоге `~/.ssh/`.

```
takmuradow@Linux:~$ ssh-keygen -C "takmuradow <1032234240@pfur.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/takmuradow/.ssh/id_rsa): chimp
chimp already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in chimp
Your public key has been saved in chimp.pub
The key fingerprint is:
SHA256:p0M9JsbfZaEi2HoKkiPlmcNW/vQHZH3Rt72u0M1YiU takmuradow <1032234240@pfur.ru>
The key's randomart image is:
+---[RSA 3072]-----+
|
|      E .
|      +
|    +o.  + o.
|   . . .oS.* o + o
|+ *    +oB.+oo .
|. @ o o +o.+o.
|o + + + .. + o.
|   o .. . . +=
+-----[SHA256]-----+
takmuradow@Linux:~$
```

Рис. 4.8: Генерация SSH-ключа

Xclip – утилита, позволяющая скопировать любой текст через терминал. Оказывается, в дистрибутиве Linux Ubuntu ее сначала надо установить. Устанавливаю xclip с помощью команды apt-get install с ключом -y от имени суперпользователя, введя в начале команды sudo (рис. 4.9).

```
takmuradow@Linux:~$ sudo apt-get install -y xclip
[sudo] password for takmuradow:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  xclip
0 upgraded, 1 newly installed, 0 to remove and 24 not upgraded.
Need to get 18,3 kB of archives.
After this operation, 60,4 kB of additional disk space will be used.
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy/universe amd64 xclip amd64 0.13-2 [18,3 kB]
Fetched 18,3 kB in 0s (44,7 kB/s)
Selecting previously unselected package xclip.
(Reading database ... 236127 files and directories currently installed.)
Preparing to unpack .../xclip_0.13-2_amd64.deb ...
Unpacking xclip (0.13-2) ...
Setting up xclip (0.13-2) ...
Processing triggers for man-db (2.10.2-1) ...
takmuradow@Linux:~$
takmuradow@Linux:~$
```

Рис. 4.9: Установка утилиты xclip

Копирую открытый ключ из директории, в которой он был сохранен, с помощью утилиты xclip (рис. 4.10).

```
takmuradow@Linux:~$ cat ~/chimp.pub | xclip -sel clip
```

Рис. 4.10: Копирование содержимого файла

Открываю браузер, захожу на сайт GitHub. Открываю свой профиль и выбираю страницу «SSH and GPG keys». Нажимаю кнопку «New SSH key» (рис. 4.11).

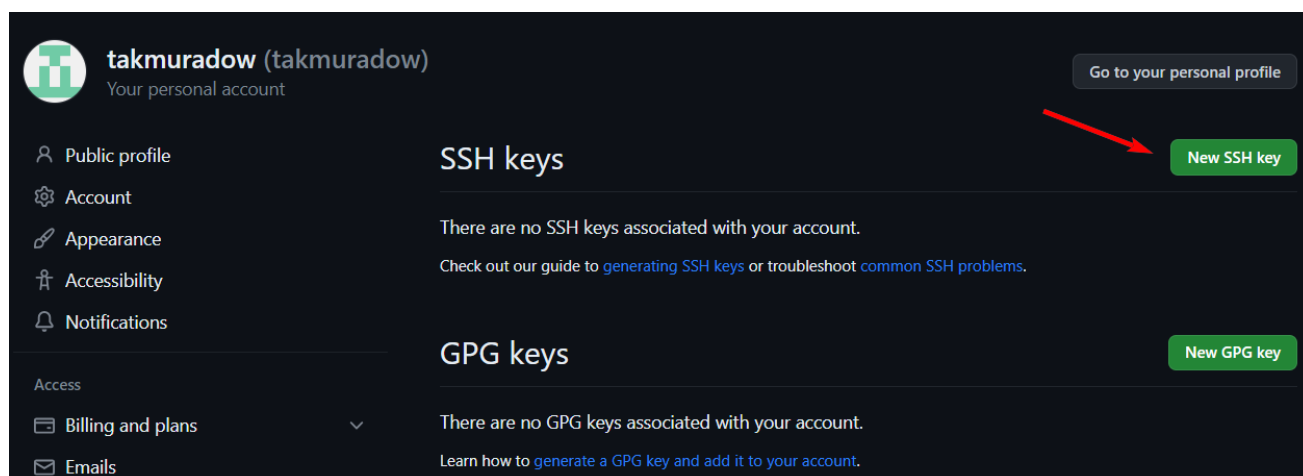


Рис. 4.11: Окно SSH and GPG keys

Вставляю скопированный ключ в поле «Key». В поле Title указываю имя для ключа. Нажимаю «Add SSH-key», чтобы завершить добавление ключа (рис. 4.12).

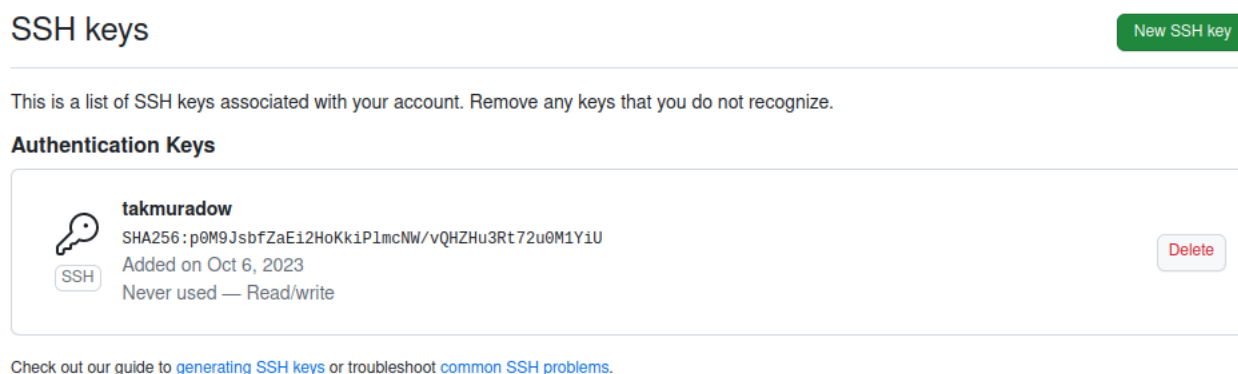


Рис. 4.12: Добавление ключа

4.4 Создание рабочего пространства и репозитория курса на основе шаблона

Закрываю браузер, открываю терминал. Создаю директорию, рабочее пространство, с помощью утилиты `mkdir`, благодаря ключу `-p` создаю все директории после домашней `~/work/study/2023-2024/“Computer architecture”` рекурсивно. Далее проверяю с помощью `ls`, действительно ли были созданы необходимые мне каталоги (рис. 4.13).

```
takmuradow@Linux:~$ mkdir -p ~/work/study/2023-2024/"Computer architecture"
takmuradow@Linux:~$ ls
chimp      Desktop    Downloads  Pictures   snap       Videos
chimp.pub  Documents  Music      Public     Templates  work
```

Рис. 4.13: Создание рабочего пространства

4.5 Создание репозитория курса на основе шаблона

В браузере перехожу на страницу репозитория с шаблоном курса по адресу <https://github.com/yamadharm/course-directory-student-template>. Далее выбираю «Use this template», чтобы использовать этот шаблон для своего репозитория (рис.4.14).

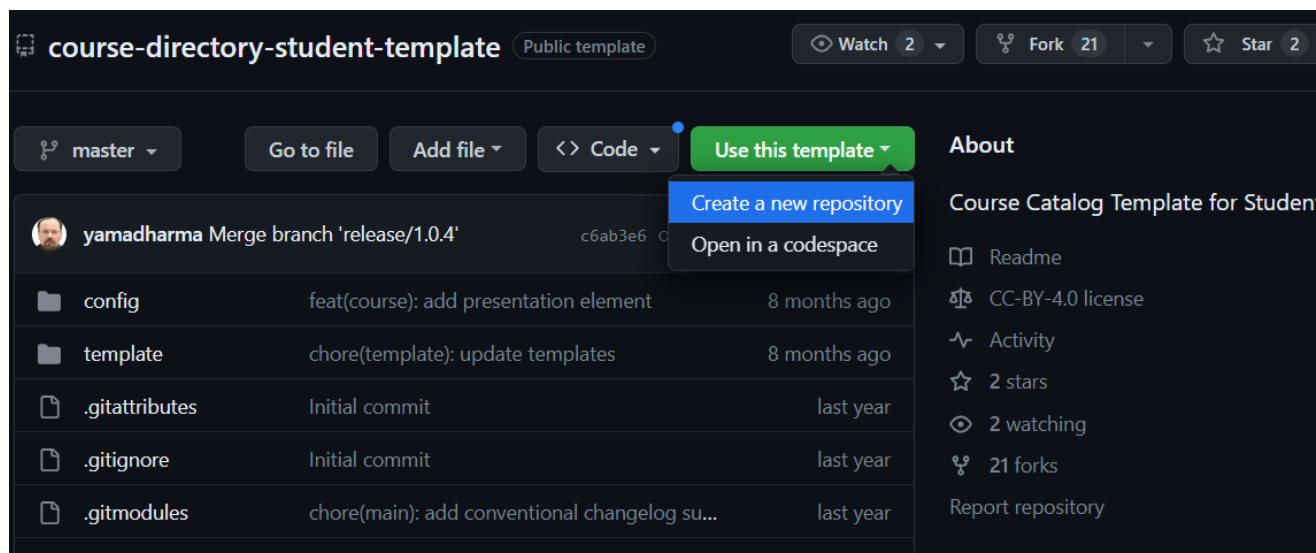


Рис. 4.14: Страница шаблона для репозитория

В открывшемся окне задаю имя репозитория (Repository name): study_2022–2023_arch-рс и создаю репозиторий, нажимаю на кнопку «Create repository» (рис. 4.15).

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 * takmuradow / **Repository name *** study_2023-2024_arch-pc

✔ study_2023-2024_arch-pc is available.

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

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. 4.15: Окно создания репозитория

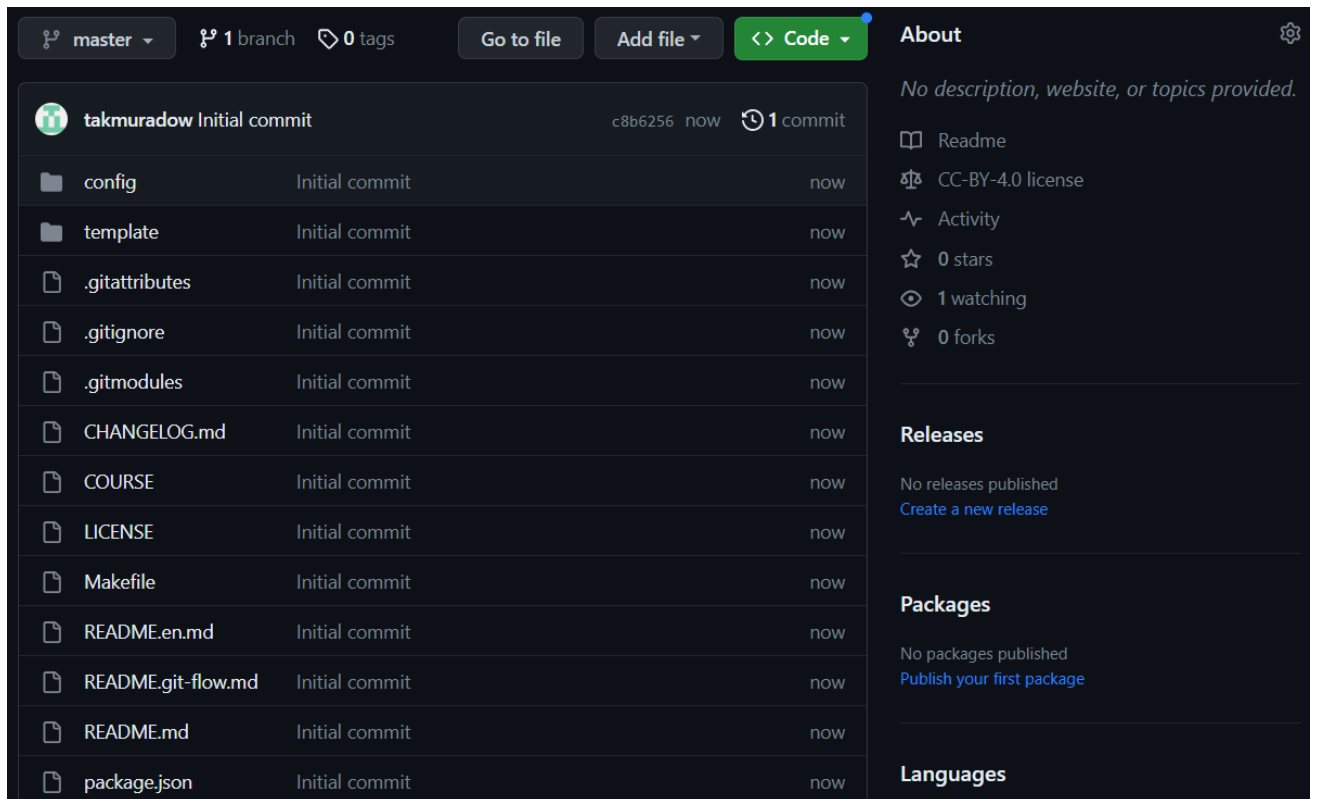


Рис. 4.16: Созданный репозиторий

Через терминал перехожу в созданный каталог курса с помощью утилиты `cd`(рис. 4.17).

```
takmuradow@Linux:~$ cd ~/work/study/2023-2024/"Computer architecture"
takmuradow@Linux:~/work/study/2023-2024/Computer architecture$
```

Рис. 4.17: Перемещение между директориями

Копирую ссылку для клонирования на странице созданного репозитория, сначала перейдя в окно «code», далее выбрав в окне вкладку «SSH» (рис. 4.18).

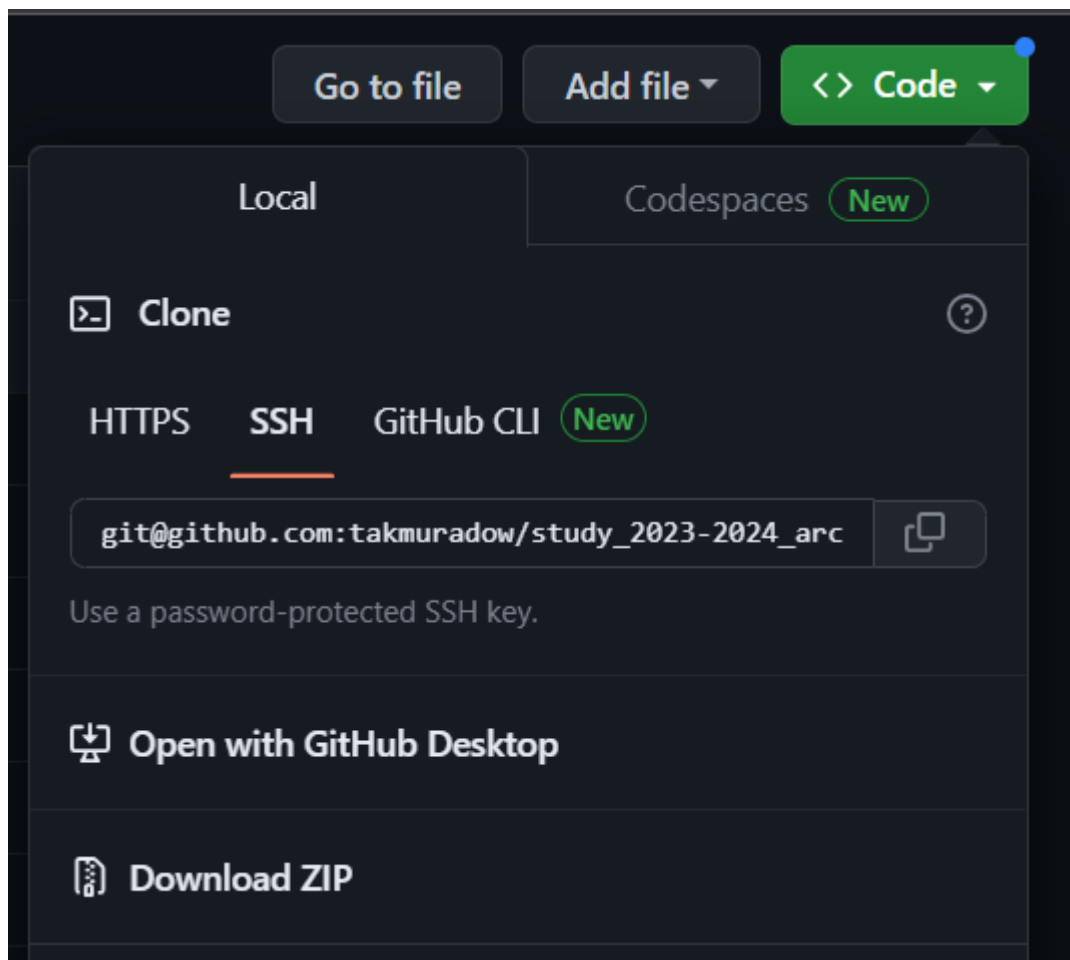


Рис. 4.18: Окно с ссылкой для копирования репозитория

Клонирую созданный репозиторий с помощью команды

`git clone --recursive git@github.com:/study_2023–2024_arh-pc.git arch-pc`

(рис. 4.19).

```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture$ git clone --recursive git@github.com:takmuradow/study_2023-2024_arch-pc.git arch-pc
Cloning into '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
Receiving objects: 100% (27/27), 16.93 KiB | 4.23 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markdown-template.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharma/academic-laboratory-report-template.git) registered for path 'template/report'
Cloning into '/home/takmuradow/work/study/2023-2024/Computer architecture/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
Receiving objects: 100% (82/82), 92.90 KiB | 1.02 MiB/s, done.
Resolving deltas: 100% (28/28), done.
Cloning into '/home/takmuradow/work/study/2023-2024/Computer architecture/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
Receiving objects: 100% (101/101), 327.25 KiB | 1.87 MiB/s, done.
Resolving deltas: 100% (40/40), done.
Submodule path 'template/presentation': checked out 'b1be3800ee91f5809264cb755d316174540b753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a83917b82e3aef11a33b1e3b2'
takmuradow@Linux:~/work/study/2023-2024/Computer architecture$
```

Рис. 4.19: Клонирование репозитория

4.6 Настройка каталога курса

Перехожу в каталог arch-pc с помощью утилиты cd (рис. 4.20).

```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture$ cd ~/work/study/2023-2024/"Computer architecture"/arch-pc
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc$
```

Рис. 4.20: Перемещение между директориями

Удаляю лишние файлы с помощью утилиты rm (рис. 4.21).

```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc$ rm package.json
```

Рис. 4.21: Удаление файлов

Создаю необходимые каталоги (рис. 4.22).

```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc$ echo arch-pc > COURSE
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc$ make
```

Рис. 4.22: Создание каталогов

Отправляю созданные каталоги с локального репозитория на сервер: добавляю все созданные каталоги с помощью `git add`, комментирую и сохраняю изменения на сервере как добавление курса с помощью `git commit` (рис. 4.23).

```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc$ git add .
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc$ git commit -am 'feat(main): make course structure'
[master 824e843] 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/kulvabov.jpg
```

Рис. 4.23: Добавление и сохранение изменений на сервере

Отправляю все на сервер с помощью `push` (рис. 4.24).

```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc$ git push
Enumerating objects: 37, done.
Counting objects: 100% (37/37), done.
Delta compression using up to 6 threads
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 342.13 KiB | 2.65 MiB/s, done.
Total 35 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:takmuradow/study_2023-2024_arch-pc.git
c8b6256..824e843 master -> master
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc$
```

Рис. 4.24: Выгрузка изменений на сервер

Проверяю правильность выполнения работы на самом сайте GitHub
(рис. 4.25).

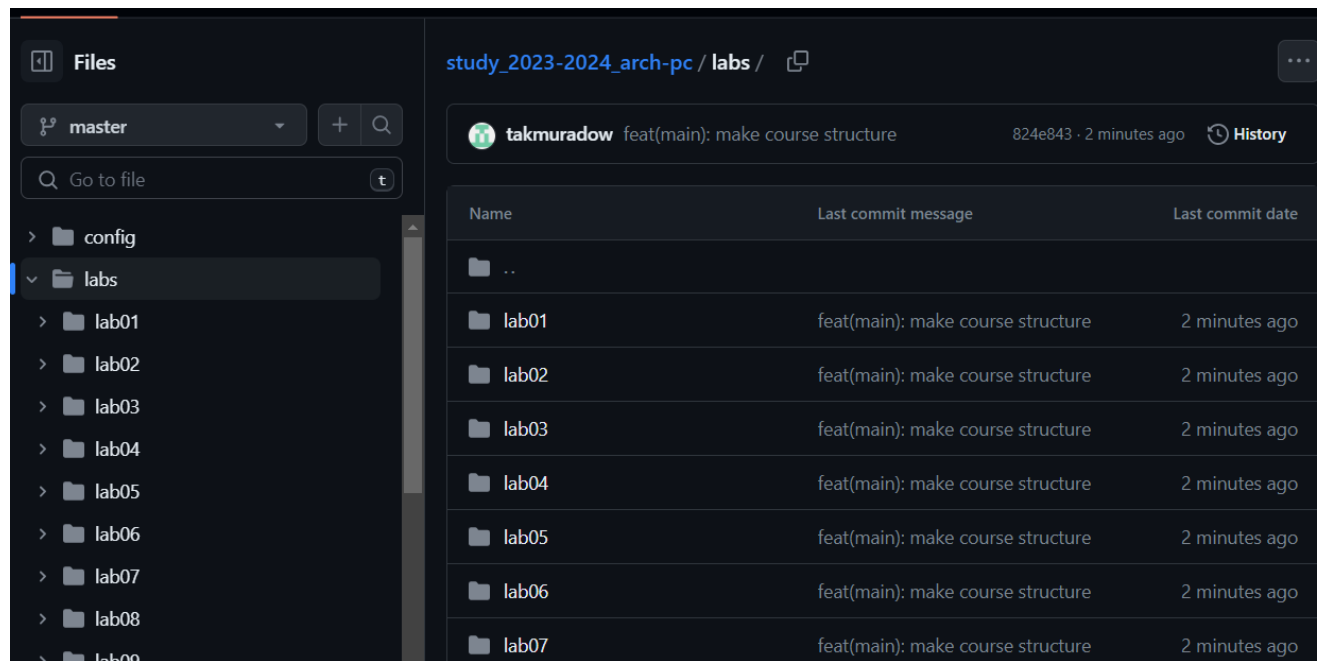


Рис. 4.25: Страница репозитория

4.7 Выполнение заданий для самостоятельной работы

1. Перехожу в директорию `labs/lab02/report` с помощью утилиты `cd`.
Создаю в каталоге файл для отчета по третьей лабораторной работе с помощью утилиты `touch` (рис. 4.26).

```
takmuradow@Linux:~/work/study/2023-2024/Computer_architecture/arch-pc/labs/lab02/report$ touch L02_Akmuradow_Report
```

Рис. 4.26: Создание файла

Оформить отчет я смогу в текстовом процессоре LibreOffice Writer, найдя его в меню приложений (рис. 4.27).

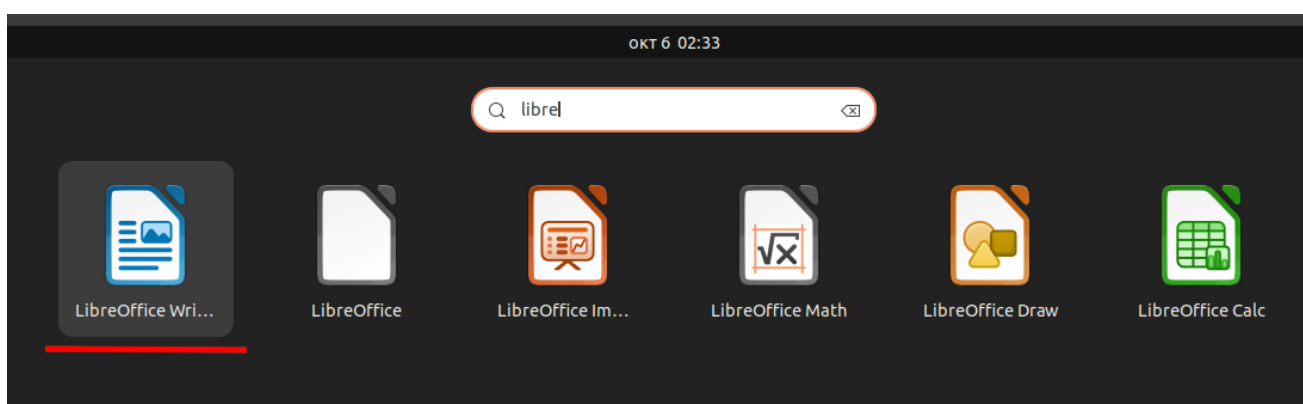


Рис. 4.27: Меню приложений

После открытия текстового редактора открываю в нем созданный файл и могу начать в нем работу над отчетом (рис. 4.28).

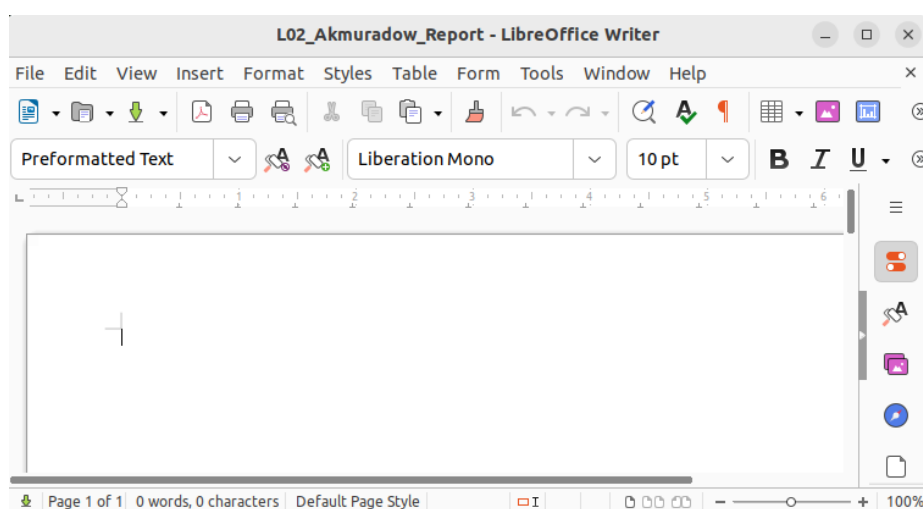


Рис. 4.28: Работа с отчетом в текстовом редакторе

2. Перехожу из подкаталога lab02/report в подкаталог lab01/report с помощью утилиты cd (рис. 4.29).

```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab02/report$ cd ..
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab02$ cd ..
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/labs$ cd lab01
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab01$ cd report
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab01/report$
```

Рис. 4.29: Перемещение между директориями

Проверяю местонахождение файлов с отчетами по первой лабораторной работе. Она должна быть в подкаталоге домашней директории «Downloads», для проверки использую команду ls (рис. 4.30).

```
takmuradow@Linux:~/Downloads$ ls
L01_Akmuradow_Report.pdf
takmuradow@Linux:~/Downloads$
```

Рис. 4.30: Проверка местонахождения файлов

Копирую первую лабораторную с помощью утилиты cp и проверяю правильность выполнения команды cp с помощью ls (рис. 4.31).

```
takmuradow@Linux:~/Downloads$ cp ~/Downloads/L01_Akmuradow_Report.pdf /home/takmuradow/work/study/2023-2024/"Computer architecture"/
arch-pc/labs/lab01/report
takmuradow@Linux:~/Downloads$ cd
takmuradow@Linux:~$ cd ~/work/study/2023-2024/"Computer architecture"/arch-pc/labs/lab01/report
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab01/report$ ls
bib image L01_Akmuradow_Report.pdf Makefile pandoc report.md
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab01/report$
```

Рис. 4.31: Копирование файла

Добавляю файл L01_Akmuradow_Report.pdf (рис. 4.32).

```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab01/report
$ git add L01_Akmuradow_Report.pdf
```

Рис. 4.32: Добавление файла на сервер

Сохраняю изменения на сервере командой `git commit -m "..."`, поясняя, что добавил файлы.

То же самое делаю для отчета по третьей лабораторной работе: перехожу в директорию `labs/lab02/report` с помощью `cd`, добавляю с помощью `git add` нужный файл, сохраняю изменения с помощью `git commit` (рис. 4.33).

```
takmuradow@Linux:~$ cd ~/work/study/2023-2024/"Computer architecture"/arch-pc/labs/lab02/report
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab02/report
$ ls
bib image L02_Akmuradow_Report Makefile pandoc report.md
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab02/report
$ git add L02_Akmuradow_Report
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab02/report
$ git commit -m "Add existing files"
[master 1e8ac55] Add existing files
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 labs/lab01/report/L01_Akmuradow_Report.pdf
 create mode 100644 labs/lab02/report/L02_Akmuradow_Report
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab02/report
$
```

Рис. 4.33: Подкаталоги и файлы в репозитории

Отправляю в центральный репозиторий сохраненные изменения командой `git push -f origin master` (рис. 4.34).

```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab02/report
$ git push -f origin master
Enumerating objects: 14, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 6 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 1.39 MiB | 2.20 MiB/s, done.
Total 8 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:takmuradow/study_2023-2024_arch-pc.git
 824e843..1e8ac55 master -> master
```

Рис. 4.34 Отправка в центральный репозиторий сохраненных изменений

Проверяю на сайте GitHub правильность выполнения заданий. Вижу, что пояснение к совершенным действиям отображается (рис. 4.35).

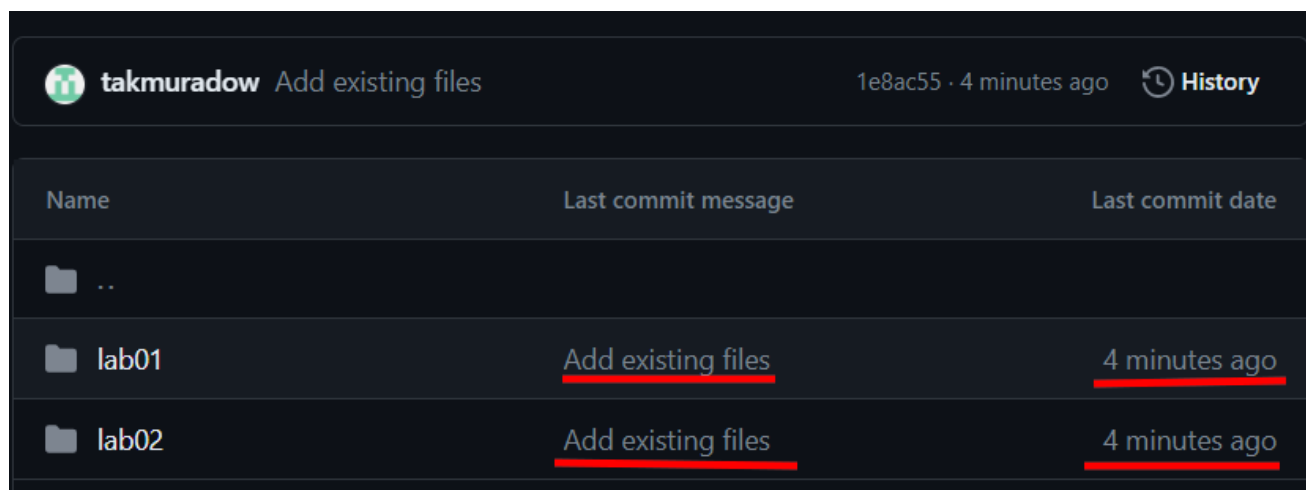


Рис. 4.35: Страница каталога в репозитории

При просмотре изменений так же вижу, что были добавлены файлы с отчетами по лабораторным работам (рис. 4.36).

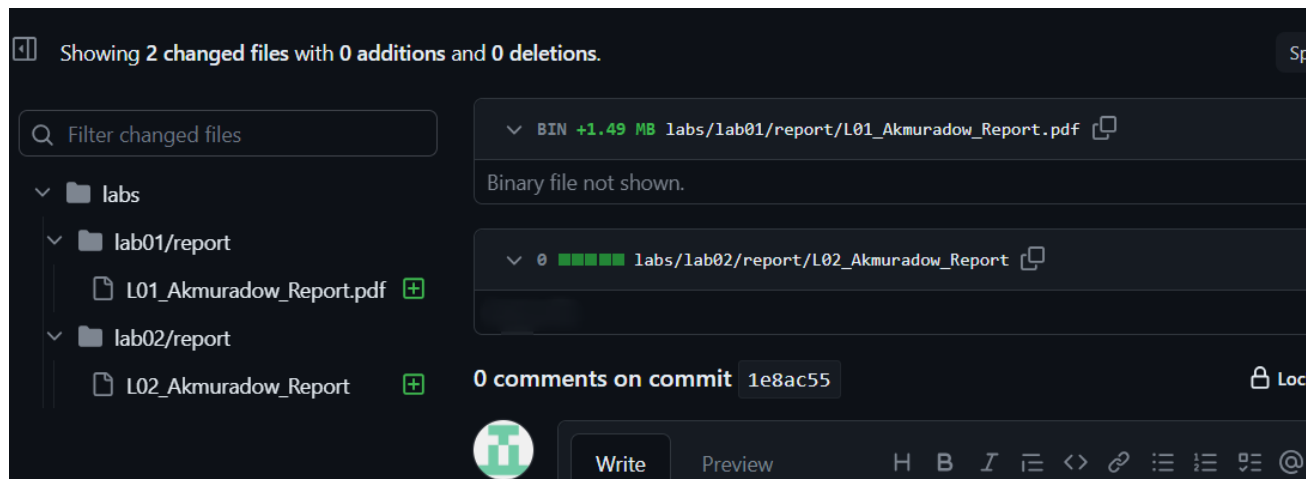
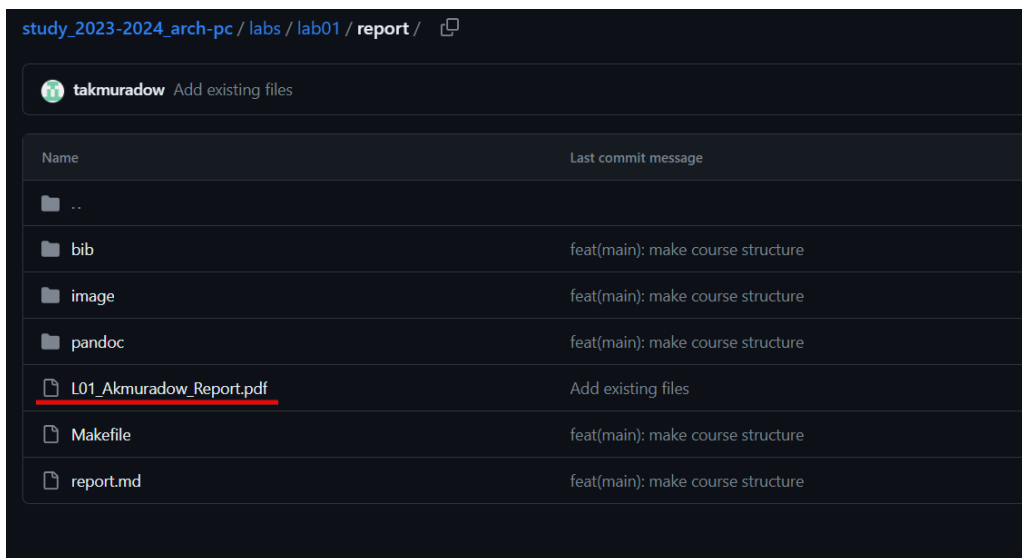


Рис. 4.36: Страница последних изменений в репозитории

Вижу, что отчеты по лабораторным работам находятся в соответствующих каталогах репозитория



study_2023-2024_arch-pc / labs / lab01 / report /	
takmuradow Add existing files	
Name	Last commit message
..	
bib	feat(main): make course structure
image	feat(main): make course structure
pandoc	feat(main): make course structure
<u>L01_Akmuradow_Report.pdf</u>	Add existing files
Makefile	feat(main): make course structure
report.md	feat(main): make course structure

Рис. 4.37: Каталог lab01/report

5 Выводы

При выполнении данной лабораторной работы я изучил идеологию и применение средств контроля версий, а также приобрел практические навыки по работе с системой git.

6 Список литературы

1. Архитектура ЭВМ
2. Git - gitattributes Документация