

Отчёт по лабораторной работе №2

Дисциплина: архитектура компьютера

Маньковская Дарья Станиславовна

Содержание

Список иллюстраций

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 работы.

3 Теоретическое введение

Системы контроля версий (Version Control System, VCS) применяются при работе несколь-

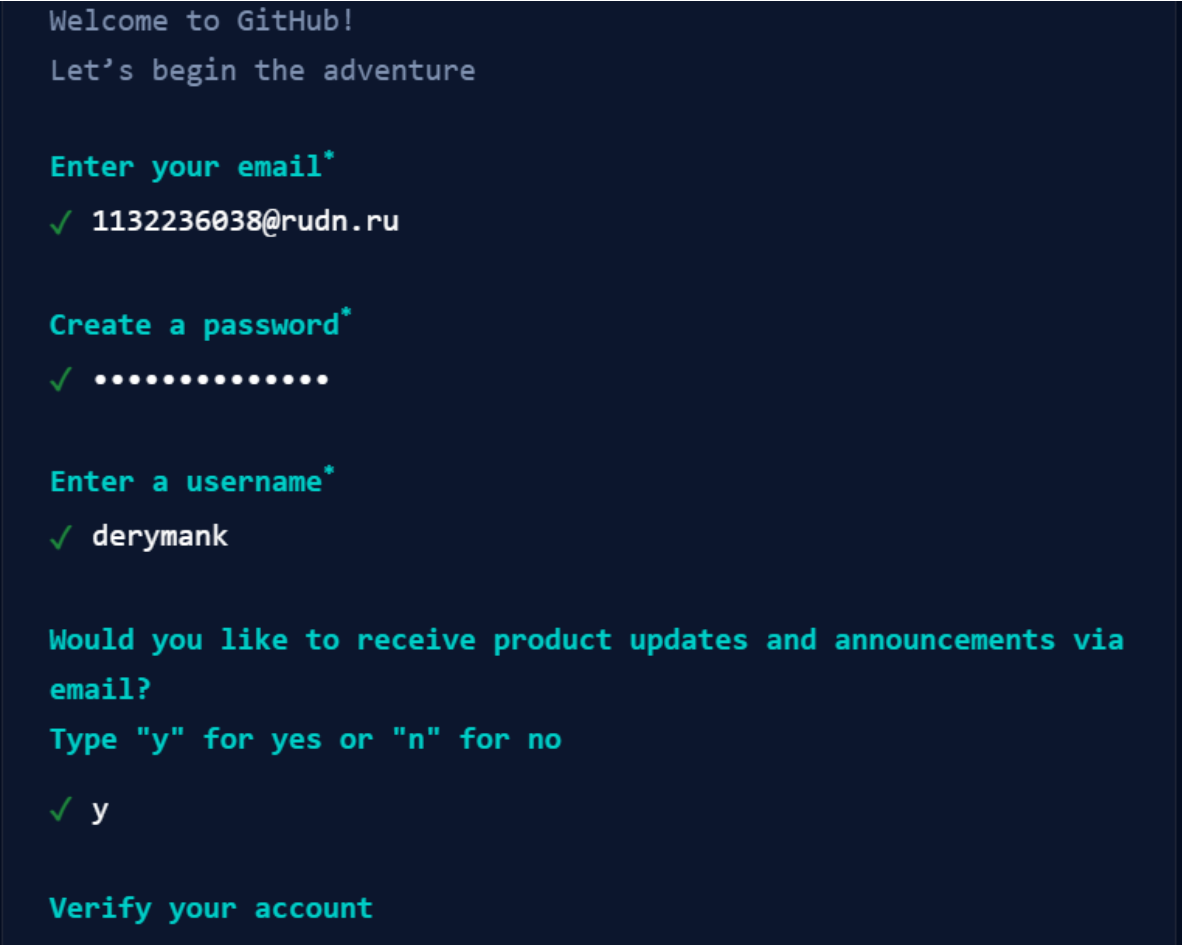
колько проектов. Каждый проект хранится в локальном или удаленном репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведенные разными участниками проекта, производить откат 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 определенных 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 измененных файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объем 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 распределенных системах контроля версий центральный репозиторий не является обязательным. Среди классических VCS наиболее известны CVS, Subversion, а среди распределенных — 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 распределенной системой контроля версий, резервную копию локального хранилища можно сделать простым копированием

или архивацией. Работа 5пользователя 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

Создаю учетную запись на сайте GitHub (рис. 1). Далее я заполнила основные данные учетной записи. (рис. 1).

A screenshot of the GitHub account creation interface. The background is dark blue. The text is in a light blue, monospaced font. It shows the following steps: 1. 'Welcome to GitHub! Let's begin the adventure'. 2. 'Enter your email*' with a green checkmark and the email '1132236038@rudn.ru'. 3. 'Create a password*' with a green checkmark and a series of dots. 4. 'Enter a username*' with a green checkmark and the username 'derymank'. 5. 'Would you like to receive product updates and announcements via email?' with the instruction 'Type "y" for yes or "n" for no' and a green checkmark followed by 'y'. 6. 'Verify your account' at the bottom.

```
Welcome to GitHub!
Let's begin the adventure

Enter your email*
✓ 1132236038@rudn.ru

Create a password*
✓ .....

Enter a username*
✓ derymank

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

Verify your account
```

Рис. 1. Заполнение данных учѐтной записи GitHub

Аккаунт создан (рис. 2).

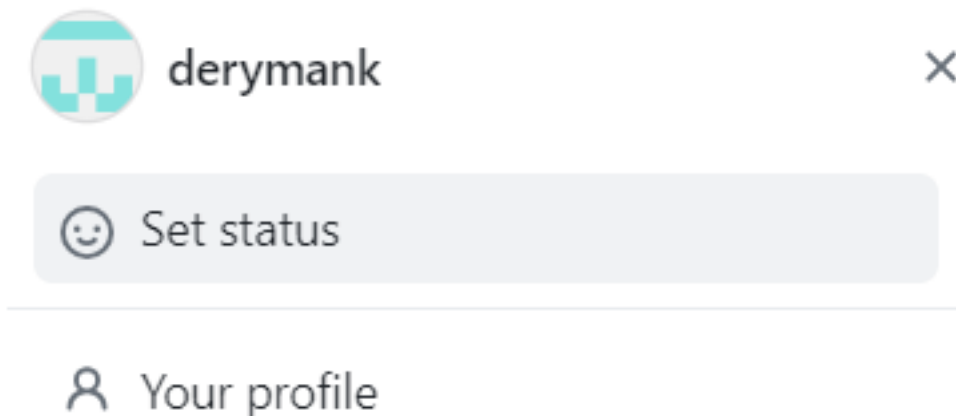


Рис. 2. Аккаунт GitHub

Открываю виртуальную машину, затем открываю терминал и делаю предварительную конфигурацию git. Ввожу команду `git config --global user.name` `XX`, указывая свое имя и команду `git config --global user.email` `XXwork@mailXX`, указывая в ней мою почту (рис.3).

```
[derymank@fedora ~]$ git config --global user.name "<Darya Mankovskaya>"
[derymank@fedora ~]$ git confog --global user.email "<1132236038@rudn.ru>"
git: «confog» не является командой git. Смотрите «git --help».

Самые похожие команды:
    config
[derymank@fedora ~]$ git config --global user.email "<1132236038@rudn.ru>"
```

Рис. 3. Предварительная конфигурация git

Настраиваю utf-8 в выводе сообщений git для корректного отображения символов (рис. 4).

```
[derymank@fedora ~]$ git config --global core.quotepath false
[derymank@fedora ~]$
```

Рис. 4. Настройка кодировки

Задаю имя «master» для начальной ветки (рис. 5).

```
[derymank@fedora ~]$ git config --global init.defaultBranch master
[derymank@fedora ~]$
```

Рис. 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. 6). CR и LF – это сим-

волы, которые можно использовать для обозначения разрыва строки в текстовых файлах.

```
[derymank@fedora ~]$ git config --global core.autocrlf input
```

Рис. 6. Параметр autocrlf

Задаю параметр safecrlf со значением warn, так Git будет проверять преобразование на обратимость (рис. 7). При значении warn Git только выведет предупреждение, но будет принимать необратимые конвертации.

```
[derymank@fedora ~]$ git config --global core.safecrlf warn
```

Рис. 7. Параметр safecrlf

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. 8). Ключ автоматически сохранится в каталоге `~/.ssh/`.

```
[derymank@fedora ~]$ ssh-keygen -C "Darya Mankovskaya <1132236038@rudn.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/derymank/.ssh/id_rsa):
Created directory '/home/derymank/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/derymank/.ssh/id_rsa
Your public key has been saved in /home/derymank/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:BcW+rgKeDu+pGnhPYxrW+TdS3cr+gqQZ1Yp7lNeERbg Darya Mankovskaya <1132236038@rudn.ru>
The key's randomart image is:
+---[RSA 3072]-----+
|      .o.o.      |
|      .o .       |
|      o.+        |
|      ..E .      |
|      oS+ =      |
| .   ...o * + o  |
| o =.*o 0 = .    |
| + 0o+* = =     |
| ..o+* .+.+.o.  |
+---[SHA256]-----+
```

Рис. 8. Генерация SSH-ключа

Xclip – утилита, позволяющая скопировать любой текст через терминал.

Устанавливаю xclip с помощью команды `sudo dnf install xclip xsel` (рис. 9).

```
[derymank@fedora ~]$ sudo dnf install xclip xsel
Последняя проверка окончания срока действия метаданных: 0:16:35 назад, Чт 28 сен 2023 00:59:01.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия                Репозиторий  Размер
=====
Установка:
xclip      x86_64       0.13-19.git11cba61.fc38  fedora       37 k
xsel       x86_64       1.2.0-32.fc38           fedora       28 k
=====
Результат транзакции
=====
Установка 2 Пакета

Объем загрузки: 64 k
Объем изменений: 106 k
Продолжить? [д/Н]: д
Загрузка пакетов:
(1/2): xclip-0.13-19.git11cba61.fc38.x86_64.rpm 711 kB/s | 37 kB    00:00
(2/2): xsel-1.2.0-32.fc38.x86_64.rpm           513 kB/s | 28 kB    00:00
-----
Общий размер                65 kB/s | 64 kB    00:00
```


Рис. 9. Установка утилиты xclip

Вывожу ключ с помощью команды `cat ~/.ssh/id_rsa.pub` и копирую его (рис. 10).

```
[derymank@fedora ~]$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQCliKwKzTLZ5moiZSb40Z9M7Hj9l5Z/LZimfTkSnnw
h/vqYVdsJlM83cL2lUQyH/yIOpb3e5UxfD5bp55YwzimHNnGb2yxIVLREusIYmPfmdlrdC+ZiZiIUcvm
LLmYd9jGPha9mV4/5a89kMnu+EsY6V2EIs32VwUNr4BDsGpw3+iZwQ1ElpNRcS/Q6ofoGVPgakBlHykF
hAVvQirb43uUTbnu10aGG+jN6N8QoEkHxvX5B8XtGY4CDcY254+qq2R5/8W/PrU+9lqd9HGcIPmtjDPK
02XX1BXm8nlYJiZi8Fu6yppmVPFujGo1qjFdzMxlbUX6QJxvK60Lk5g2Ixnz6YjnQk5Sv6v2ISEBRynn
KWB36Sk/BAeZx9rsT8AleWSz8fk0CeTkZd+KHEMTUFQK8rJ/WWHxr4WEFrpydRVd0gtvleoZFOWrirPY
AH8Vj8HmOzqNAen1bI+OHSe9sBE8UixV5J13aWbYwM68XkHQWE2bWWNdpkixPjIhCYL0rF0= Darya M
ankovskaya <1132236038@rudn.ru>
```

Рис. 10. Копирование ключа

Открываю браузер, захожу на сайт GitHub. Открываю свой профиль и выбираю страницу «SSH and GPG keys». Нажимаю кнопку «New SSH key» (рис. 11).



derymank (derymank)
Your personal account

Go to your personal profile

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

Code, planning, and automation

Repositories

Codespaces

SSH keys

New SSH key

There are no SSH keys associated with your account.

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

GPG keys

New GPG key

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

Vigilant mode

☐ **Flag unsigned commits as unverified**

This will include any commit attributed to your account but not signed with your GPG or S/MIME key.
Note that this will include your existing unsigned commits.

[Learn about vigilant mode.](#)

Рис. 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. 12).

Add new SSH Key

Title

derymank23

Key type

Authentication Key

Key

```
/Q6ofoGVPgakBlHykFnAVvQirb43uUTbnu10aGG+jN6N8QoEkHxvX5B8XtGY4CDcY254+q  
q2R5  
/8W/PrU+9lqd9HGcIPmtjDPKO2XX1BXm8nlyJiZi8Fu6yppmVPFujGo1qjFdzMxlbUX6QJxvK  
6OLk5g2Ixnz6YjnQk5Sv6v2ISEBRynnKWB36Sk  
/BAeZx9rsT8AleWSz8fkOCeTkZd+KHEMTUfQK8rJ  
/WWHxr4WEFrpYdRVdOgtvleoZFOWrirPYAH8Vj8HmOzqNAen1bI+OHSe9sBE8UixV5J13a  
WbYwM68XkhQWE2bWWNdpkixPjIhCYL0rF0= Darya Mankovskaya  
<1132236038@rudn.ru>
```

Add SSH key

Рис. 12. Добавление ключа

Закрываю браузер, открываю 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`, действительно ли были созданы необходимые мне каталоги (рис. 13).

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

Рис. 13. Пример использования ключей утилиты

В браузере перехожу на страницу репозитория с шаблоном курса по адресу <https://github.com/yamadharma/course-directory-student-template>. Далее выбираю

«Use this template», чтобы использовать этот шаблон для своего репозитория (рис. 14).

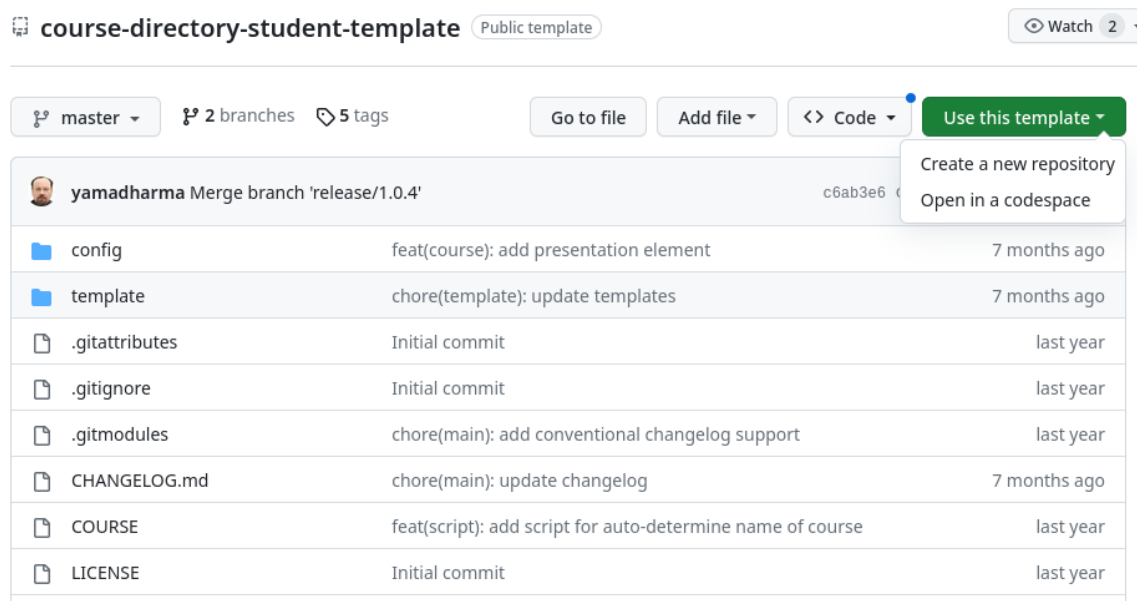


Рис. 14. Страница шаблона для репозитория

В открывшемся окне задаю имя репозитория (Repository name): study_2023–2024_arh-рс и создаю репозиторий, нажимаю на кнопку «Create repository from template» (рис. 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 *

 derymank ▾

Repository name *

/ study_2023-2024_arh-pc

✔ study_2023-2024_arh-pc is available.

Great repository names are short and memorable. Need inspiration? How about **musical-octo-invention** ?

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. 15. Окно создания репозитория

Репозиторий создан (рис. 16).

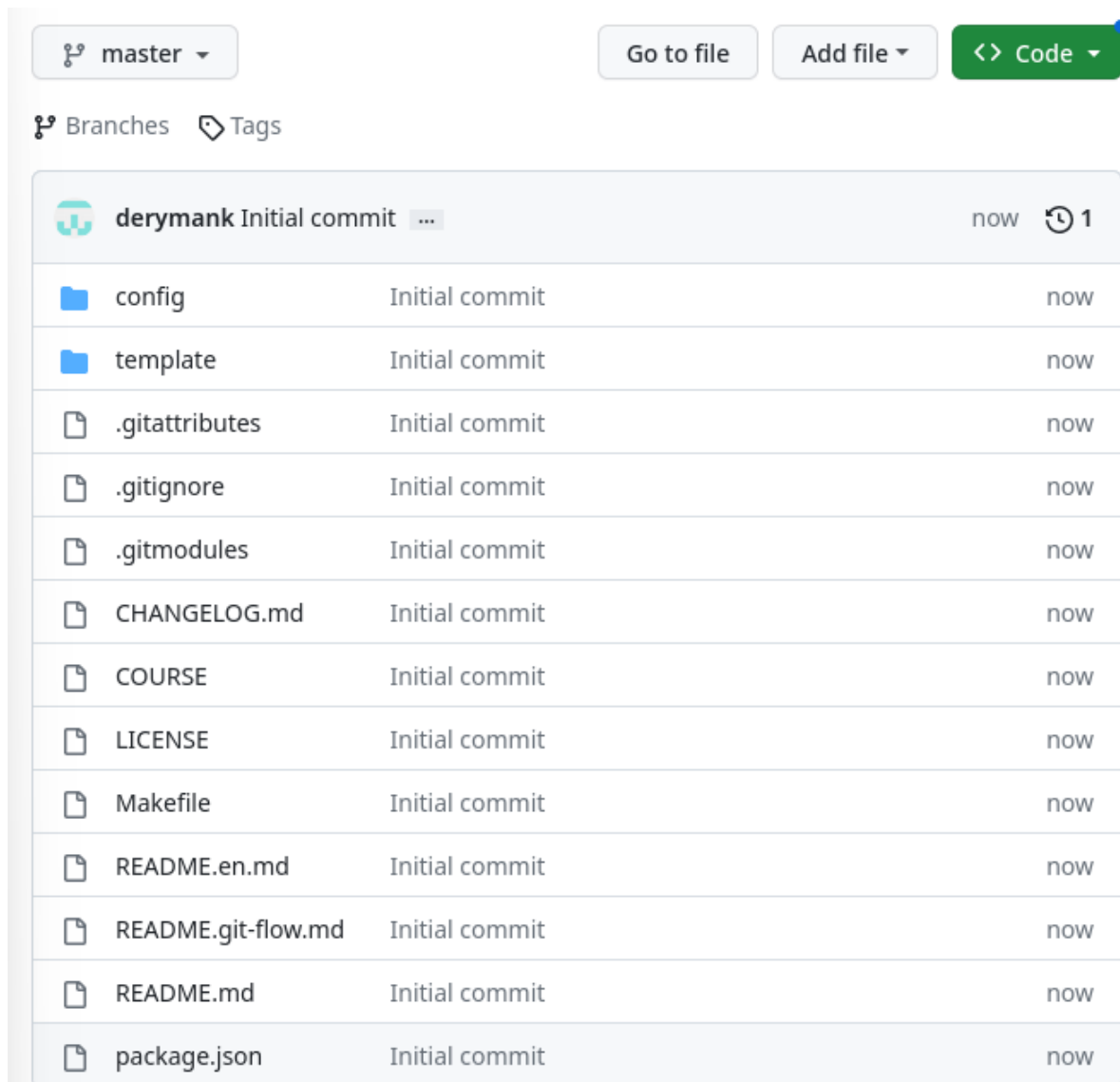


Рис. 16. Созданный репозиторий

Через терминал перехожу в созданный каталог курса с помощью утилиты `cd` (рис. 17).

```
[derymank@fedora ~]$ cd ~/work/study/2023-2024/'Архитектура компьютера'
[derymank@fedora Архитектура компьютера]$
```

Рис. 17. Перемещение между директориями

Клонирую созданный репозиторий с помощью команды `git clone -recursive git@github.com:/study_2023-2024_arh-pc.git arch-pc` (рис. 18).

```
[derymank@fedora Архитектура компьютера]$ git clone --recursive git@github.com:derymank/study_2023-2024_arh-pc.git arch-pc
Клонирование в «arch-pc»...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/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 | 2.12 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/derymank/work/study/2023-2024/Архитектура компьютера/arch-pc/template/presentation»...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
```

Рис. 18. Клонирование репозитория

Копирую ссылку для клонирования на странице созданного репозитория, сначала перейдя в окно «code», далее выбрав в окне вкладку «SSH» (рис. 19).

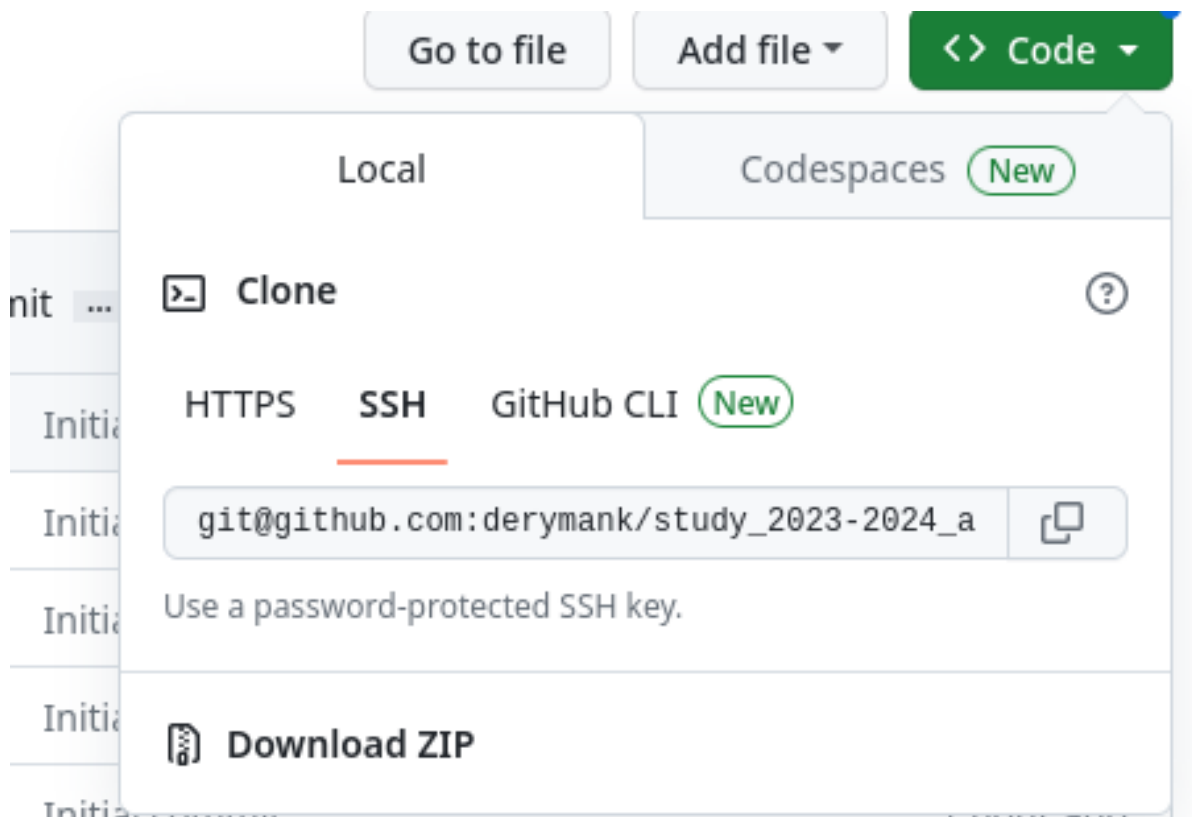


Рис. 19. Окно с ссылкой для копирования репозитория

Перехожу в каталог arch-pc с помощью утилиты cd (рис. 20).

```
derymank@fedora ~]$ cd ~/work/study/2023-2024/Архитектура\ компьютера/arch-pc
derymank@fedora arch-pc]$
```

Рис. 20. Перемещение между директориями

Удаляю лишние файлы с помощью утилиты rm (рис. 21).

```
[derymank@fedora arch-pc]$ rm package.json
```

Рис. 21. Удаление файлов

Создаю необходимые каталоги (рис. 22).

```
[derymank@fedora arch-pc]$ echo arch-pc > COURSE
[derymank@fedora arch-pc]$ make
```

Рис. 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. 23).

```
[derymank@fedora arch-pc]$ git add .
[derymank@fedora arch-pc]$ git commit -am 'feat(main): make course structure'
[master 7706fb2] 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
```


Рис. 23. Добавление и сохранение изменений на сервере

Отправляю все на сервер с помощью push (рис. 24).

```
[derymank@fedora arch-pc]$ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.14 КиБ | 2.76 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:derymank/study_2023-2024_arh-pc.git
 66f38cc..7706fb2 master -> master
```

Рис. 24. Выгрузка изменений на сервер

Проверяю правильность выполнения работы сначала на самом сайте GitHub (рис. 25).

 derymank feat(main): make course structure

Name	Last commit message
..	
lab01	feat(main): make course structure
lab02	feat(main): make course structure
lab03	feat(main): make course structure
lab04	feat(main): make course structure
lab05	feat(main): make course structure
lab06	feat(main): make course structure
lab07	feat(main): make course structure
lab08	feat(main): make course structure
lab09	feat(main): make course structure
lab10	feat(main): make course structure
lab11	feat(main): make course structure
README.md	feat(main): make course structure
README.ru.md	feat(main): make course structure

Рис. 25. Страница репозитория

Перехожу в директорию labs/lab02/report с помощью утилиты cd. Создаю в каталоге файл для отчета по второй лабораторной работе с помощью утилиты touch (рис. 26).

```
[derymank@fedora arch-pc]$ cd labs/lab02/report  
[derymank@fedora report]$ touch Л02_Маньковская_отчет
```

Рис. 26. Создание файла

Оформить отчет я смогу в текстовом процессоре LibreOffice Writer, найдя его в меню приложений (рис. 27).

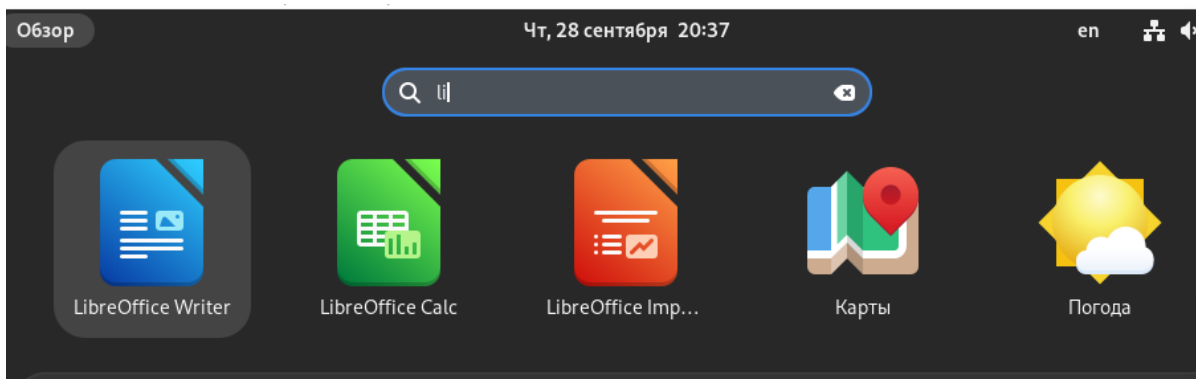


Рис. 27. Меню приложений

После открытия текстового процессора открываю в нем созданный файл и могу начать в нем работу над отчетом (рис. 28).

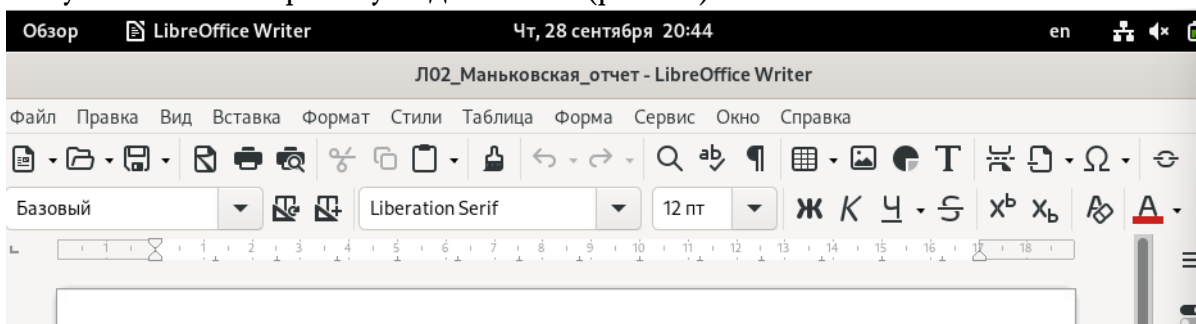


Рис. 28. Работа с отчетом в текстовом процессоре

Перехожу из подкаталога lab02/report в подкаталог lab01/report с помощью утилиты cd (рис. 29).

```
[derymank@fedora report]$ cd ..
[derymank@fedora lab02]$ cd ..
[derymank@fedora labs]$ cd lab01/report/
[derymank@fedora report]$
```

Рис. 29. Перемещение между директориями

Проверяю местонахождение файла с отчетом по первой лабораторной работе. Он должен быть в подкаталоге домашней директории «Загрузки», для проверки использую команду ls (рис. 30).

```
[derymank@fedora report]$ ls ~/Загрузки
Л01_Маньковская_отчет.pdf
```

Рис. 30. Проверка местонахождения файлов

Копирую первую лабораторную с помощью утилиты `cp` и проверяю правильность выполнения команды `cp` с помощью `ls` (рис. 31).

```
[derymank@fedora report]$ cp ~/Загрузки/Л01_Маньковская_отчет.pdf /home/derymank/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab01/report
[derymank@fedora report]$ ls
bib image Makefile pandoc report.md Л01_Маньковская_отчет.pdf
[derymank@fedora report]$
```

Рис. 31. Копирование файла

Перехожу в директорию, в которой находится отчет по первой лабораторной работе с помощью `cd` (рис. 32).

```
[derymank@fedora report]$ cd ../; cd ..
[derymank@fedora labs]$ cd lab01/report/
[derymank@fedora report]$
```

Рис. 32. Перемещение между директориями

Добавляю файл `Л01_Маньковская_отчет` (рис. 33).

```
[derymank@fedora report]$ git add Л01_Маньковская_отчет.pdf
```

Рис. 33. Добавление файла на сервер

Сохраняю изменения на сервере командой `git commit -m [текст]`, поясняя, что добавила файлы. То же самое делаю для отчета по второй 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. 34).

```
[derymank@fedora report]$ cd ~/work/study/2023-2024/Архитектура\ компьютера/arch-pc
[derymank@fedora arch-pc]$ cd labs/lab02/report
[derymank@fedora report]$ git add Л02_Маньковская_отчет
[derymank@fedora report]$ git commit -m "Add existing file"
[master 4ela0ff] Add existing file
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Л01_Маньковская_отчет.pdf
create mode 100644 labs/lab02/report/Л02_Маньковская_отчет
[derymank@fedora report]$
```

Рис. 34. Подкаталоги и файлы в репозитории

Отправляю в центральный репозиторий сохраненные изменения командой

git push -f origin master (рис. 35).

```
[derymank@fedora report]$ git push -f origin master
Перечисление объектов: 14, готово.
Подсчет объектов: 100% (12/12), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (8/8), готово.
Запись объектов: 100% (8/8), 1.21 МиБ | 8.60 МиБ/с, готово.
Всего 8 (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:derymank/study_2023-2024_arh-pc.git
  7706fb2..4e1a0ff  master -> master
[derymank@fedora report]$
```

Рис. 35. Отправка в центральный репозиторий сохраненных изменений

Проверяю на сайте GitHub правильность выполнения заданий. Вижу, что пояснение к совершенным действиям отображается (рис. 36).

[study_2023-2024_arh-pc](#) / [labs](#) /

derymank feat(main): make course structure	
Name	Last commit message
..	
lab01	feat(main): make course structure
lab02	feat(main): make course structure
lab03	feat(main): make course structure
lab04	feat(main): make course structure
lab05	feat(main): make course structure
lab06	feat(main): make course structure
lab07	feat(main): make course structure
lab08	feat(main): make course structure
lab09	feat(main): make course structure
lab10	feat(main): make course structure
lab11	feat(main): make course structure
README.md	feat(main): make course structure
README.ru.md	feat(main): make course structure

Рис. 36. Страница каталога в репозитории

При просмотре изменений так же вижу, что были добавлены файлы с отчетами по лабораторным работам (рис. 37).

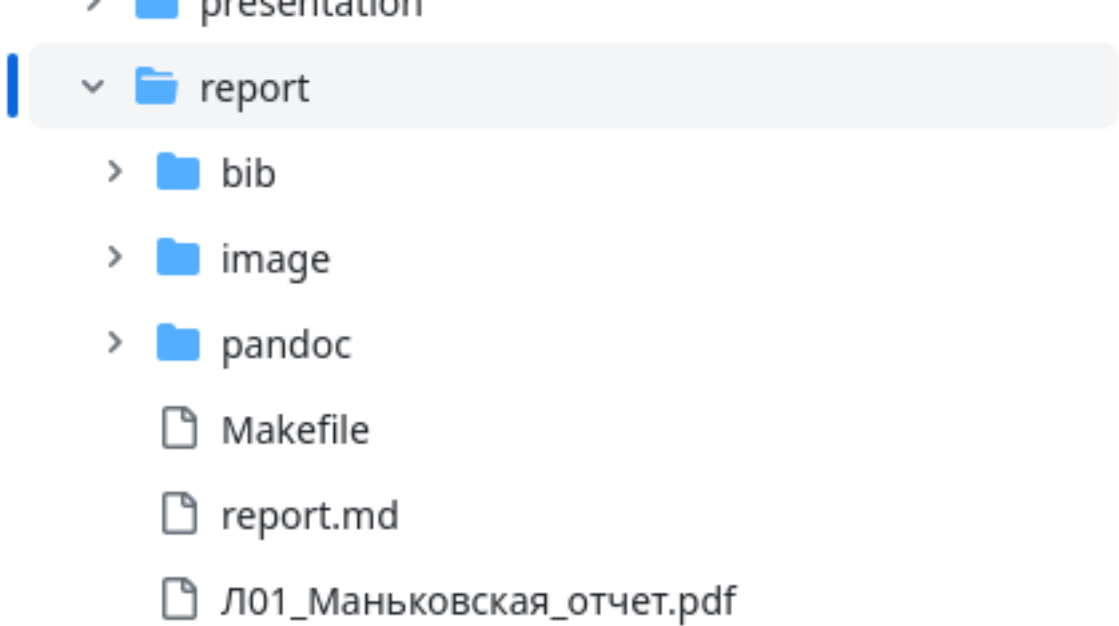


Рис. 37. Страница последних изменений в репозитории

Вижу, что отчеты по лабораторным работам находятся в соответствующих каталогах репозитория: отчет по первой - в lab01/report (рис. 38), по второй – в lab02/report (рис. 39).

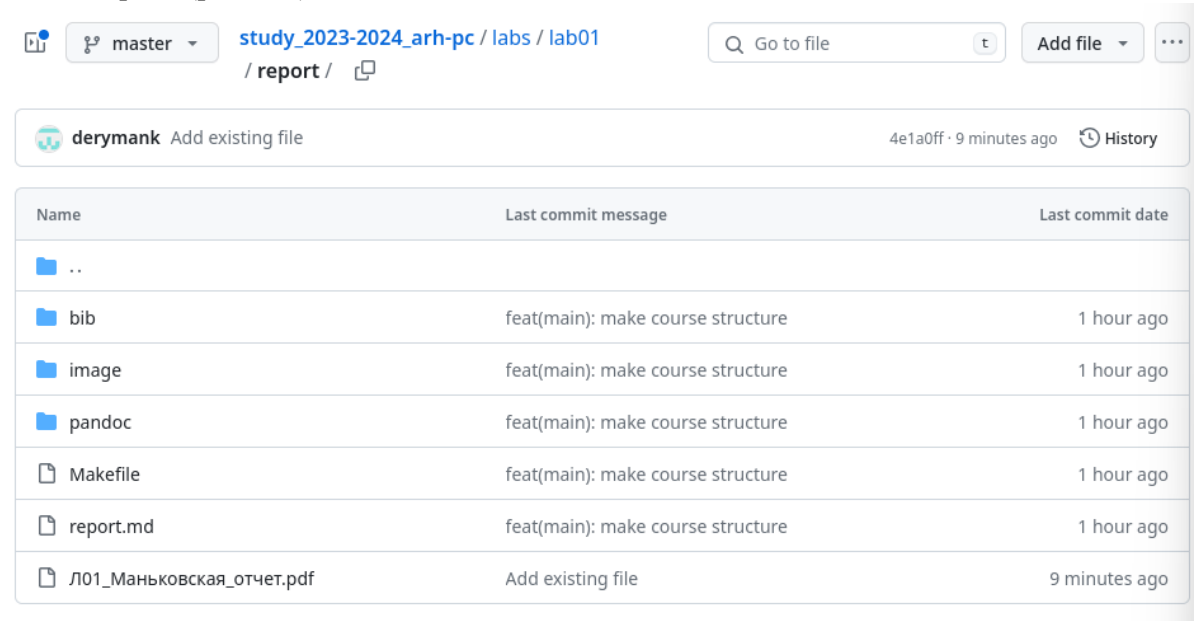



Рис. 38. Каталог lab01/report


 master


study_2023-2024_arh-pc / labs / lab02


Add file

...

/ report /



 derymank Add existing file

4e1a0ff · 10 minutes ago  History








Name	Last commit message	Last commit date
 ..		
 bib	feat(main): make course structure	1 hour ago
 image	feat(main): make course structure	1 hour ago
 pandoc	feat(main): make course structure	1 hour ago
 Makefile	feat(main): make course structure	1 hour ago
 report.md	feat(main): make course structure	1 hour ago
 ЛО2_Маньковская_отчет	Add existing file	10 minutes ago

Рис. 39. Каталог lab02/report

5 Выводы

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

6 Список литературы

1.Архитектура ЭВМ

2.Git – gitattributes Документация