

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

дисциплина: *Архитектура компьютера*

Студент: Панявкина Ирина

Группа: НКАбд-04-24

МОСКВА

2024 г.

Содержание

1	Цель работы	2
2	Задание.....	2
3	Теоретическое введение	2
4	Выполнение лабораторной работы.....	3
4.1	Настройка GitHub	3
4.2	Базовая настройка Git.....	4
4.3	Создание SSH-ключа	5
4.4	Создание рабочего пространства и репозитория курса на основе шаблона.....	8
4.5	Создание репозитория курса на основе шаблона	9
4.6	Настройка каталога курса.....	11
4.7	Выполнение заданий для самостоятельной работы	13
5	Выводы	17
6	Список литературы	17

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 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. 1). Далее я заполнила основные данные учетной записи.



Irina Panyavkina

irinapanyavkina · she/her

Edit profile

📍 Moscow

🕒 18:15 (UTC +03:00)

✉ renrrurenrru@gmail.com

Рис. 1: Заполнение данных учетной записи GitHub

Аккаунт создан (рис. 2).

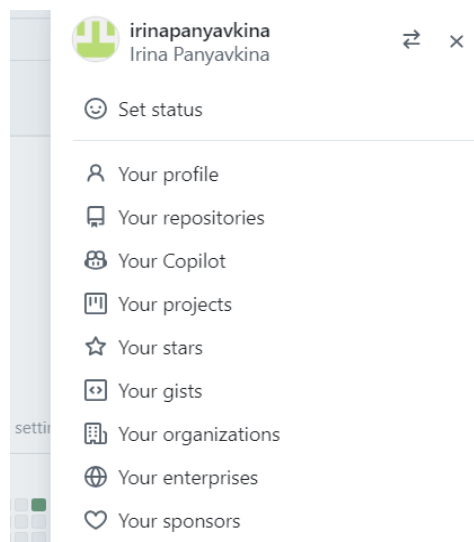
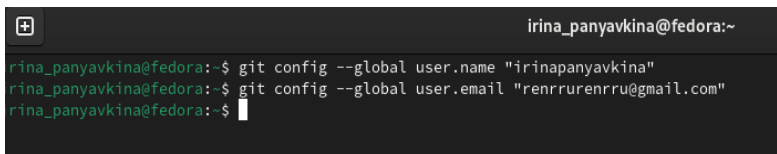


Рис. 2: Аккаунт GitHub

4.2 Базовая настройка Git

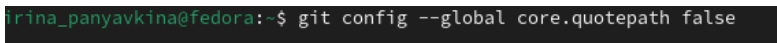
Открываю виртуальную машину, затем открываю терминал и делаю предварительную конфигурацию git. Ввожу команду `git config --global user.name ""`, указывая свое имя и команду `git config --global user.email "work@mail"`, указывая в ней электронную почту владельца, то есть мою (рис. 3).

A terminal window with a dark background. The title bar shows a window icon and the text 'irina_panyavkina@fedora:~'. The terminal contains three lines of text: 'irina_panyavkina@fedora:~\$ git config --global user.name "irinapanyavkina"', 'irina_panyavkina@fedora:~\$ git config --global user.email "renrrurenrru@gmail.com"', and 'irina_panyavkina@fedora:~\$' followed by a cursor.

```
irina_panyavkina@fedora:~$ git config --global user.name "irinapanyavkina"
irina_panyavkina@fedora:~$ git config --global user.email "renrrurenrru@gmail.com"
irina_panyavkina@fedora:~$
```

Рис. 3: Предварительная конфигурация git

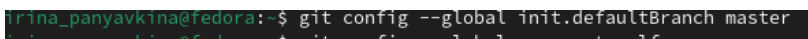
Настраиваю utf-8 в выводе сообщений git для корректного отображения символов (рис. 4).

A terminal window with a dark background. The title bar shows the text 'irina_panyavkina@fedora:~'. The terminal contains one line of text: 'irina_panyavkina@fedora:~\$ git config --global core.quotePath false'.

```
irina_panyavkina@fedora:~$ git config --global core.quotePath false
```

Рис. 4: Настройка кодировки

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

A terminal window with a dark background. The title bar shows the text 'irina_panyavkina@fedora:~'. The terminal contains one line of text: 'irina_panyavkina@fedora:~\$ git config --global init.defaultBranch master'.

```
irina_panyavkina@fedora:~$ git config --global init.defaultBranch master
```

Рис. 5: Создание имени для начальной ветки

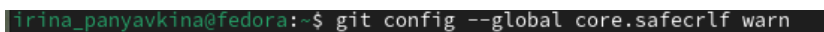
Задаю параметр autocrlf со значением input, так как я работаю в системе Linux, чтобы конвертировать CRLF в LF только при коммитах (рис. 6). CR и LF – это символы, которые можно использовать для обозначения разрыва строки в текстовых файлах.

A terminal window with a dark background. The title bar shows the text 'irina_panyavkina@fedora:~'. The terminal contains one line of text: 'irina_panyavkina@fedora:~\$ git config --global core.autocrlf input'.

```
irina_panyavkina@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.

A terminal window with a dark background. The title bar shows the text 'irina_panyavkina@fedora:~'. The terminal contains one line of text: 'irina_panyavkina@fedora:~\$ git config --global core.safecrlf warn'.

```
irina_panyavkina@fedora:~$ git config --global core.safecrlf warn
```

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

```

irina_panyavkina@fedora:~$ ssh-keygen -C "irinapanyavkina <renrrurenrru@gmail.com>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/irina_panyavkina/.ssh/id_ed25519): 

```

```

Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/irina_panyavkina/.ssh/id_ed25519):
Created directory '/home/irina_panyavkina/.ssh'.
Enter passphrase (empty for no passphrase):

```

```

Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/irina_panyavkina/.ssh/id_ed25519
Your public key has been saved in /home/irina_panyavkina/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:v72gHKpQ3Wr/Wq+E3AL4upEafEYhfdjRlwG2ZRUKL44 irinapanyavkina <renrrurenrru@gmail.com>
The key's randomart image is:
+--[ED25519 256]--+
|    .+.+++.    |
|   . 0 0.*0.   |
|  . + 0 0 0    |
|   . = + .     |
|   + E S       |
|  . 0 0 + +    |
| + = + = *     |
| * + + * =     |
|  . +0. =0+.+. |
+-----[SHA256]-----+
irina_panyavkina@fedora:~$

```

Рис. 8: Последовательная генерация SSH-ключа

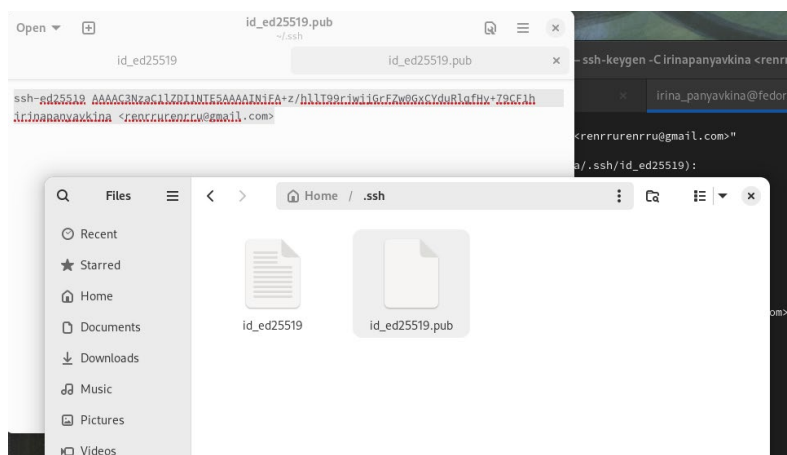


Рис. 8.1: Появление файла публичного SSH-ключа

Xclip – утилита, позволяющая скопировать любой текст через терминал. Оказывается, в дистрибутиве Linux Kali ее сначала надо установить. Устанавливаю xclip (рис. 9).

```

irina_panyavkina@fedora:~$ cat ~/.ssh/id_ed25519.pub | xclip -sel clip
bash: xclip: command not found...
Install package 'xclip' to provide command 'xclip'? [N/y] y

* Waiting in queue...
* Loading list of packages....
The following packages have to be installed:
xclip-0.13-21.git11cba61.fc40.x86_64    Command line clipboard grabber
Proceed with changes? [N/y] y

* Waiting in queue...
* Waiting for authentication...
* Waiting in queue...
* Downloading packages...
* Requesting data...
* Testing changes...
* Installing packages...

irina_panyavkina@fedora:~$ cat ~/.ssh/id_ed25519.pub | xclip -sel clip
irina_panyavkina@fedora:~$

```

Рис. 9: Установка утилиты xclip

Копирую открытый ключ из директории, в которой он был сохранен, с помощью утилиты xclip (рис. 10).

```

irina_panyavkina@fedora:~$ cat ~/.ssh/id_ed25519.pub | xclip -sel clip
irina_panyavkina@fedora:~$

```

Рис. 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. 11).

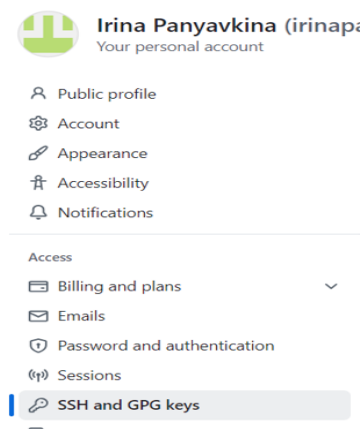


Рис. 11: Окно SSH and GPG keys

Вставляю скопированный ключ в поле «Key». В поле Title указываю имя для ключа. Нажимаю «Add SSH-key», чтобы завершить добавление ключа (рис. 12).

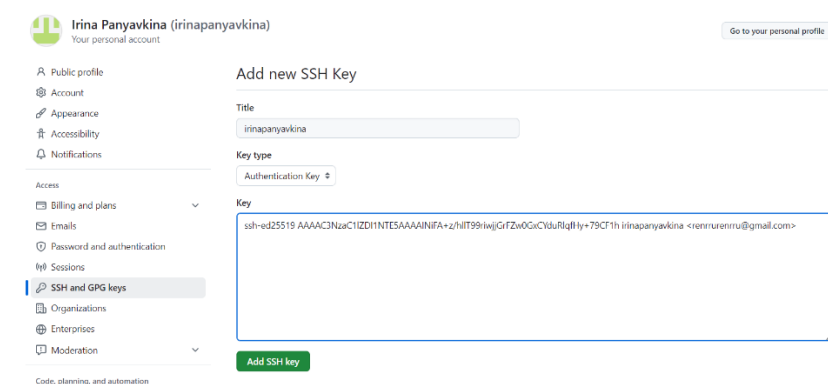


Рис. 12.1: Добавление ключа

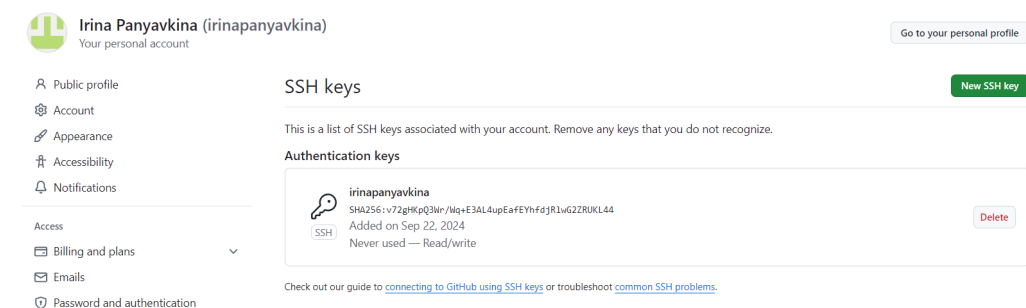


Рис. 12.2: Отображение публичного SSH-ключа на Github

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/2024-2025/` “Архитектура компьютера” рекурсивно. (рис. 13).

```
irina_panyavkina@fedora:~$ mkdir -p ~/work/study/2024-2025/"Архитектура компьютера"
irina_panyavkina@fedora:~$
```

Рис. 13: Создание рабочего пространства

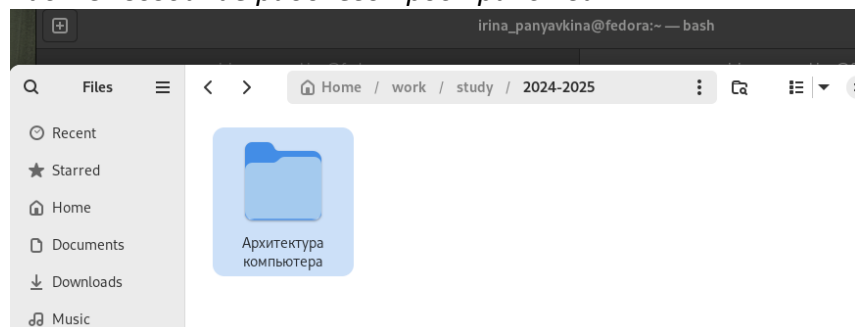


Рис. 13.2 Успешное создание новой директории

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. 14).

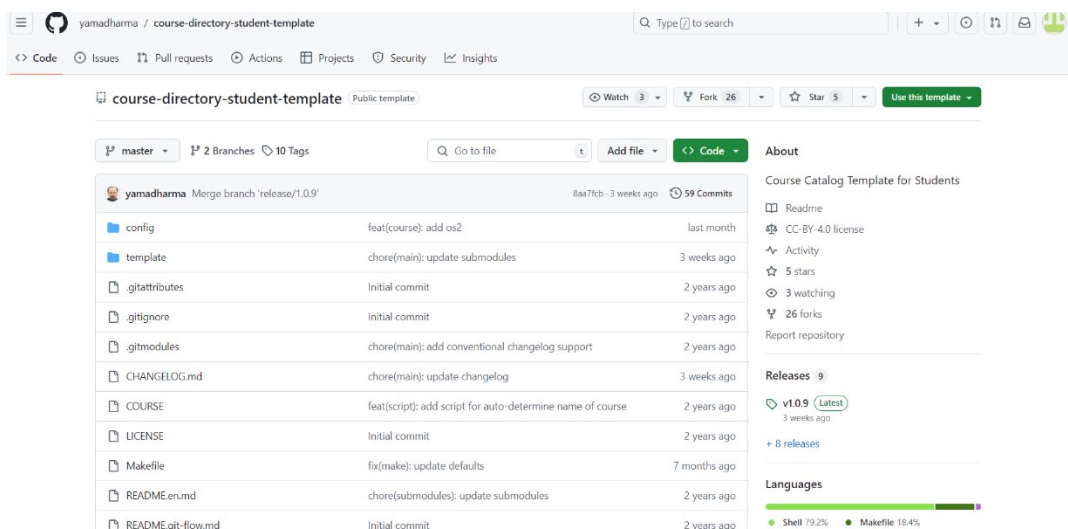


Рис. 14: Страница шаблона для репозитория

В открывшемся окне задаю имя репозитория (Repository name): study_2024–2025_arh-pc 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 ().*

Repository template
yamadharm/course-directory-student-template
Start your repository with a template repository's contents.

☒ **Include all branches**
Copy all branches from yamadharm/course-directory-student-template and not just the default branch.

Owner * irinapanyavkina / **Repository name *** study_2024-2025_arhpc
✔ Your new repository will be created as study_2024-2025_arhpc.
The repository name can only contain ASCII letters, digits, and the characters -, ., and _.

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

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.

Рис. 15: Окно создания репозитория

Репозиторий создан (рис. 16).

<div> <div> <div></div> <div>irinapanyavkina</div> </div> <div>Initial commit</div> </div> <div>40d2759 · now · 1 Commit</div>	
config	Initial commit · now
template	Initial commit · now
.gitattributes	Initial commit · now
.gitignore	Initial commit · now
.gitmodules	Initial commit · now
CHANGELOG.md	Initial commit · now
COURSE	Initial commit · now
LICENSE	Initial commit · now
Makefile	Initial commit · now
README.en.md	Initial commit · now
README.git-flow.md	Initial commit · now
README.md	Initial commit · now
package.json	Initial commit · now
<div> <div> <div></div> <div>README</div> </div> <div>CC-BY-4.0 license</div> <div> <div></div> <div></div> </div> </div>	

Рис. 16: Созданный репозиторий

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

```

irina_panyavkina@fedora:~$ cd /home/irina_panyavkina/work/study/2024-2025/"Архитектура компьютера"
irina_panyavkina@fedora:~/work/study/2024-2025/Архитектура компьютера$

```

Рис. 17: Перемещение между директориями

Клонирую созданный репозиторий с помощью команды `git clone --recursive git@github.com:/study_2024-2025_arh-pc.git` arch-pc (рис. 18).

```

irina_panyavkina@fedora:~/work/study/2024-2025/Архитектура компьютера$ git clone --recursive git@github.com:irinapanyavkina/study_2024-2025_arh-pc.git arch-pc
Cloning into 'arch-pc'...
remote: Enumerating objects: 34, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 34 (delta 1), reused 18 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (34/34), 19.58 KiB | 19.58 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/irina_panyavkina/work/study/2024-2025/Архитектура компьютера/arch-pc/template/presentation'...
remote: Enumerating objects: 111, done.
remote: Counting objects: 100% (111/111), done.
remote: Compressing objects: 100% (77/77), done.
remote: Total 111 (delta 42), reused 100 (delta 31), pack-reused 0 (from 0)
Receiving objects: 100% (111/111), 102.17 KiB | 977.00 KiB/s, done.
Resolving deltas: 100% (42/42), done.
Cloning into '/home/irina_panyavkina/work/study/2024-2025/Архитектура компьютера/arch-pc/template/report'...
remote: Enumerating objects: 142, done.
remote: Counting objects: 100% (142/142), done.
remote: Compressing objects: 100% (97/97), done.
remote: Total 142 (delta 60), reused 121 (delta 39), pack-reused 0 (from 0)
Receiving objects: 100% (142/142), 341.09 KiB | 1.37 MiB/s, done.
Resolving deltas: 100% (60/60), done.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
irina_panyavkina@fedora:~/work/study/2024-2025/Архитектура компьютера$

```

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).

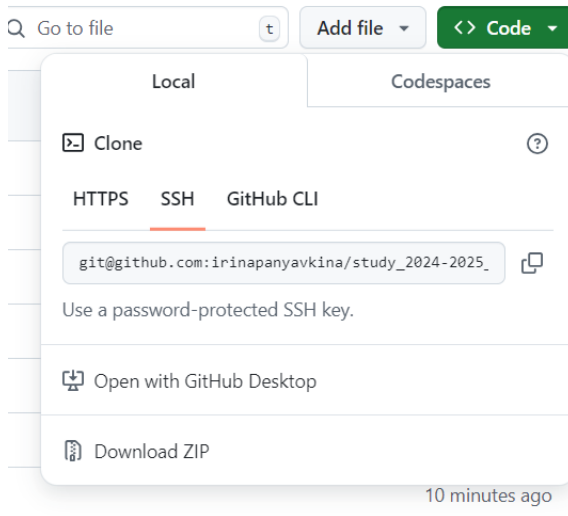


Рис. 19: Окно с ссылкой для копирования репозитория

4.6 Настройка каталога курса

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

```
irina_panyavkina@fedora:~/work/study/2024-2025/Архитектура компьютера$ cd ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc
```

Рис. 20: Перемещение между директориями

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

```
irina_panyavkina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ rm package.json
irina_panyavkina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 21: Удаление файлов

Создаю необходимые каталоги (рис. 22).

```
irina_panyavkina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
irina_panyavkina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make prepare
irina_panyavkina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

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).

```

irina_panyavkina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
irina_panyavkina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
[master 34bd1a9] feat(main): make course structure
222 files changed, 53681 insertions(+), 1 deletion(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/.projectile
create mode 100644 labs/lab01/presentation/.texlabroot
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/.projectile
create mode 100644 labs/lab02/presentation/.texlabroot
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

```

Рис. 23: Добавление и сохранение изменений на сервере

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

```

irina_panyavkina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 38, done.
Counting objects: 100% (38/38), done.
Compressing objects: 100% (26/26), done.
Writing objects: 100% (36/36), 341.43 KiB | 2.35 MiB/s, done.
Total 36 (delta 4), reused 34 (delta 3), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:irinapanyavkina/study_2024-2025_arhpc.git
 d30d880..34bd1a9 master -> master
irina_panyavkina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$

```

Рис. 24: Выгрузка изменений на сервер

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

Перехожу из подкаталога lab01/report в подкаталог lab02/report с помощью утилиты cd (рис. 32).

```
(evdvorkina@evdvorkina)-[~/.../arch-pc/labs/lab01/report]
$ cd ..

(evdvorkina@evdvorkina)-[~/.../Архитектура компьютера/arch-pc/labs/lab01]
$ cd ..

(evdvorkina@evdvorkina)-[~/.../2022-2023/Архитектура компьютера/arch-pc/labs]
$ cd lab02/report

(evdvorkina@evdvorkina)-[~/.../arch-pc/labs/lab02/report]
$
```

Рис. 32: Перемещение между директориями

Копирую вторую лабораторную с помощью утилиты cp и проверяю правильность выполнения команды cp с помощью ls (рис. 33).

```
(evdvorkina@evdvorkina)-[~/.../arch-pc/labs/lab02/report]
$ cp ~/Загрузки/Л02_Дворкина_отчет.pdf /home/evdvorkina/work/study/2022-2023/'Архитектура компьютера'/arch-pc/labs/lab02/report

(evdvorkina@evdvorkina)-[~/.../arch-pc/labs/lab02/report]
$ ls
bib image Makefile pandoc report.md Л02_Дворкина_отчет.pdf
```

Рис. 33: Копирование файла

3. Добавляю с помощью команды git add в коммит созданные файлы: Л02_Дворкина отчет (рис. 34). и

```
(evdvorkina@evdvorkina)-[~/.../arch-pc/labs/lab02/report]
$ git add Л02_Дворкина_отчет.pdf
```

Рис. 34: Добавление файла на сервер

Перехожу в директорию, в которой находится отчет по первой лабораторной работе с помощью cd (рис. 35).

```
(evdvorkina@evdvorkina)-[~/.../arch-pc/labs/lab02/report]
$ cd ..; cd ..

(evdvorkina@evdvorkina)-[~/.../2022-2023/Архитектура компьютера/arch-pc/labs]
$ cd lab01/report/
```

Рис. 35: Перемещение между директориями

Добавляю файл Л01_Дворкина_отчет (рис. 36).

```
(evdvorkina@evdvorkina)-[~/.../arch-pc/labs/lab01/report]
$ git add Л01_Дворкина_отчет.pdf
```

Рис. 36: Добавление файла на сервер

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

То же самое делаю для отчета по третьей лабораторной работе: перехожу в директорию labs/lab03/report с помощью cd, добавляю с помощью git add нужный файл, сохраняю изменения с помощью git commit (рис. 37).

```
(evdvorkina@evdvorkina)-[~/.../study/2022-2023/Архитектура компьютера/arch-pc]
$ cd labs/lab03/report

(evdvorkina@evdvorkina)-[~/.../arch-pc/labs/lab03/report]
$ git add Л03_Дворкина_отчет

(evdvorkina@evdvorkina)-[~/.../arch-pc/labs/lab03/report]
$ git commit -m "Add existing file"
[master faba3fc] Add existing file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab03/report/Л03_Дворкина_отчет
```

Рис. 37: Подкаталоги и файлы в репозитории

Отправляю в центральный репозиторий сохраненные изменения командой git push -f origin master (рис. 38).

```
(evdvorkina@evdvorkina)-[~/.../arch-pc/labs/lab03/report]
$ git push -f origin master
Перечисление объектов: 22, готово.
Подсчет объектов: 100% (18/18), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (14/14), готово.
Запись объектов: 100% (14/14), 4.20 Миб | 2.60 Миб/с, готово.
Всего 14 (изменений 6), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
To github.com:evdvorkina/study_2022-2023_arh-pc.git
+ 53421f5 ... faba3fc master -> master (forced update)
```

Рис. 38: Отправка в центральный репозиторий сохраненных изменений

Проверяю на сайте GitHub правильность выполнения заданий. Вижу, что пояснение к совершенным действиям отображается (рис. 39).

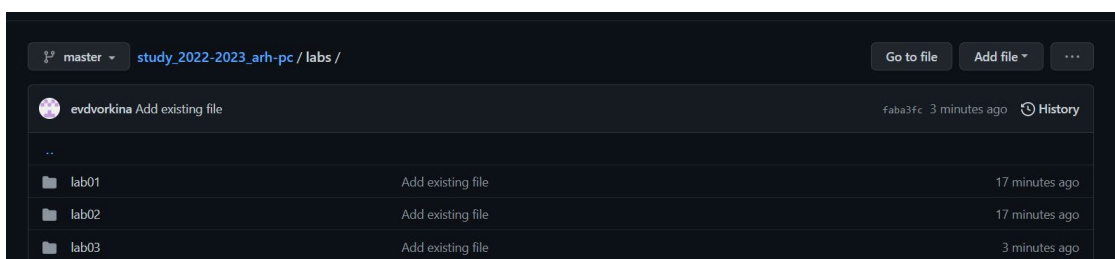


Рис. 39: Страница каталога в репозитории

При просмотре изменений так же вижу, что были добавлены файлы с отчетами по лабораторным работам (рис. 40).

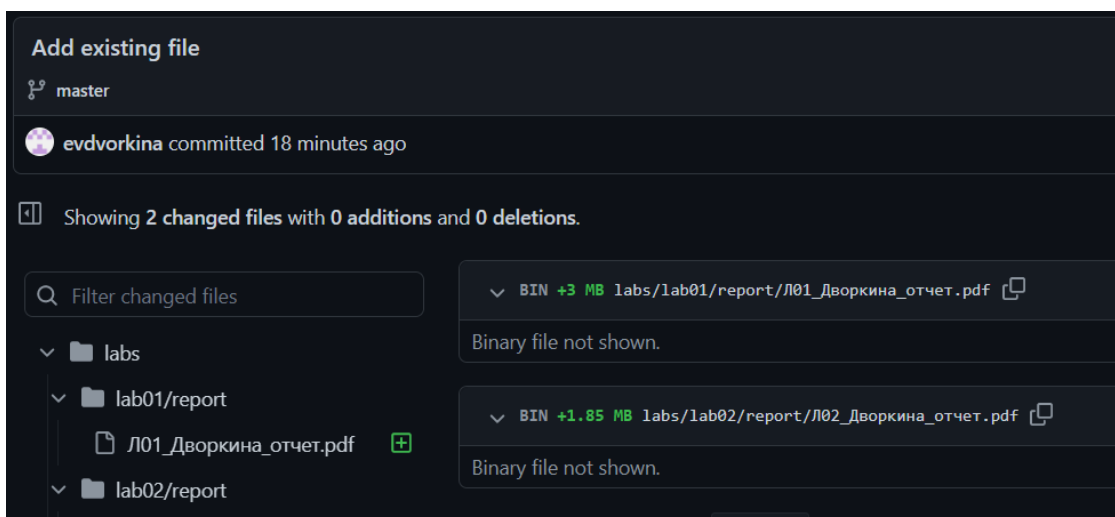


Рис. 40: Страница последних изменений в репозитории

Вижу, что отчеты по лабораторным работам находятся в соответствующих каталогах репозитория: отчет по первой - в lab01/report (рис. 41), по второй – в lab02/report (рис. 42), по третьей в - lab03/report (рис. 43).

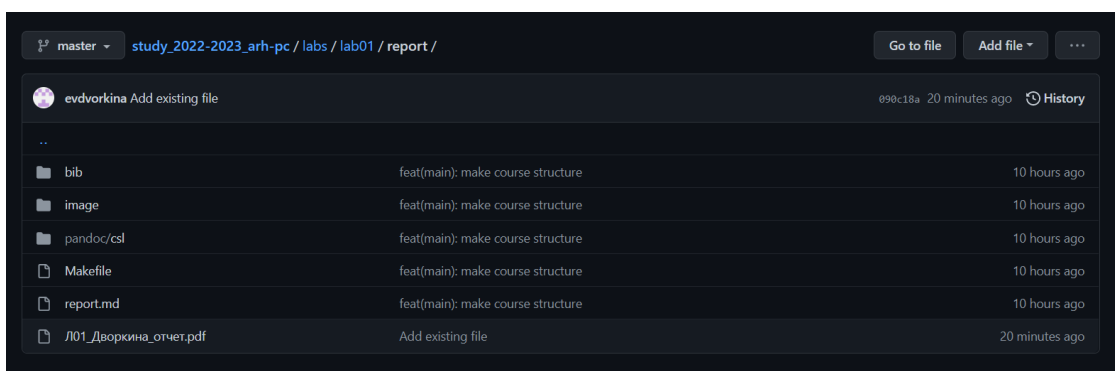


Рис. 41: Каталог lab01/report

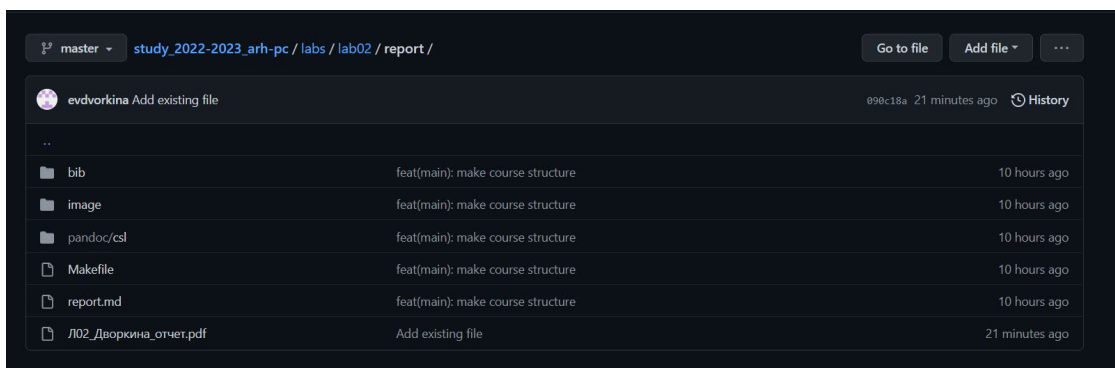


Рис. 42: Каталог lab02/report



Рис. 43: Каталог `lab03/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