

**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: Гашимова Э.Э.

Группа: НКАбд-02-24

**МОСКВА**

20\_\_24 г.

# **Содержание**

1 Цель работы 3

2 Задание 3

3 Теоретическое введение 3,4

4 Выполнение лабораторной работы 4-17

5 Выводы 18

6 Список литературы 18

## 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. Далее я заполнила основные данные учетной записи.

Аккаунт создан (рис. 2).

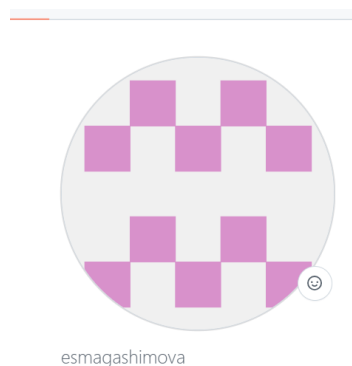


Рис. 2: Аккаунт GitHub

### 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).

```
esmagashimova@fedora:~$ git config --global user.name "<esmagashimova>"
esmagashimova@fedora:~$ git config --global user.emaail "<1132247520@pfur.ru>"
```

Рис. 3: Предварительная конфигурация git

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

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

Рис. 4: Настройка кодировки

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

```
esmagashimova@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.

```
esmagashimova@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.

```
esmagashimova@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/.

```

esmagashimova@fedora:~$ ssh-keygen -C "esmagashimova <1132247520@pfur.ru>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/esmagashimova/.ssh/id_ed25519): ESMA0161_
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ESMA0161_
Your public key has been saved in ESMA0161_.pub
The key fingerprint is:
SHA256:s0KwFVvV2KnGgnUK5BXRiEEyNjbfQU25ztPz9UVTEDQ esmagashimova <1132247520@pfur.ru>
The key's randomart image is:
+--[ED25519 256]--+
|      Bo=+B0o= oEo |
|    o Bo*o.* +   ..|
|    . =+.+ o     .|
|    +. o =      ..|
|    . . S= .    ..|
|    .  o+ o    o|
|    . . . o .o|
|    .      . .|
|                   |
+-----[SHA256]-----+
esmagashimova@fedora:~$ S

```

Рис. 8: Генерация SSH-ключа

Xclip – утилита, позволяющая скопировать любой текст через терминал. Оказывается, в дистрибутиве Linux Kali ее сначала надо установить. Устанавливаю xclip с помощью команды `dnf install xclip` с ключом `-u` от имени суперпользователя, введя в начале команды `sudo` (рис. 9)

```
esmagashimova@fedora:~$ sudo dnf install xclip
[sudo] пароль для esmagashimova:
Последняя проверка окончания срока действия метаданных: 0:02:55 назад, Пт 11 окт 2024 20:33:04.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия                Репозиторий  Размер
=====
Установка:
xclip      x86_64       0.13-21.git11cba61.fc40  fedora       37 k
=====
Результат транзакции
=====
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 37 k
Объем изменений: 62 k
Продолжить? [д/Н]: Да
Загрузка пакетов:
xclip-0.13-21.git11cba61.fc40.x86_64.rpm      18 kB/s | 37 kB      00:01
-----
Общий размер                2.2 kB/s | 37 kB      00:16
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка      : 1/1
Установка       : xclip-0.13-21.git11cba61.fc40.x86_64 1/1
Запуск скриптлета: xclip-0.13-21.git11cba61.fc40.x86_64 1/1

Установлен:
xclip-0.13-21.git11cba61.fc40.x86_64
```

Рис. 9: Установка утилиты xclip

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

```
esmagashimova@fedora:~$ cat ~/ESMA0161_.pub
```

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).

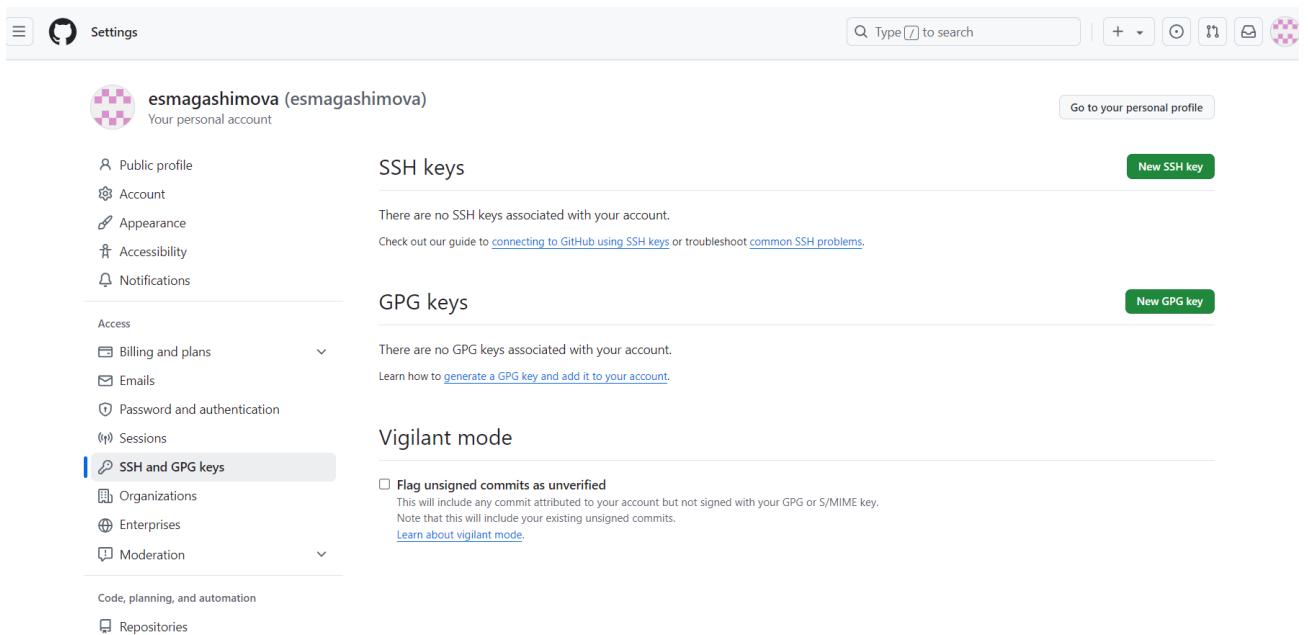


Рис. 11: Окно SSH and GPG keys

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

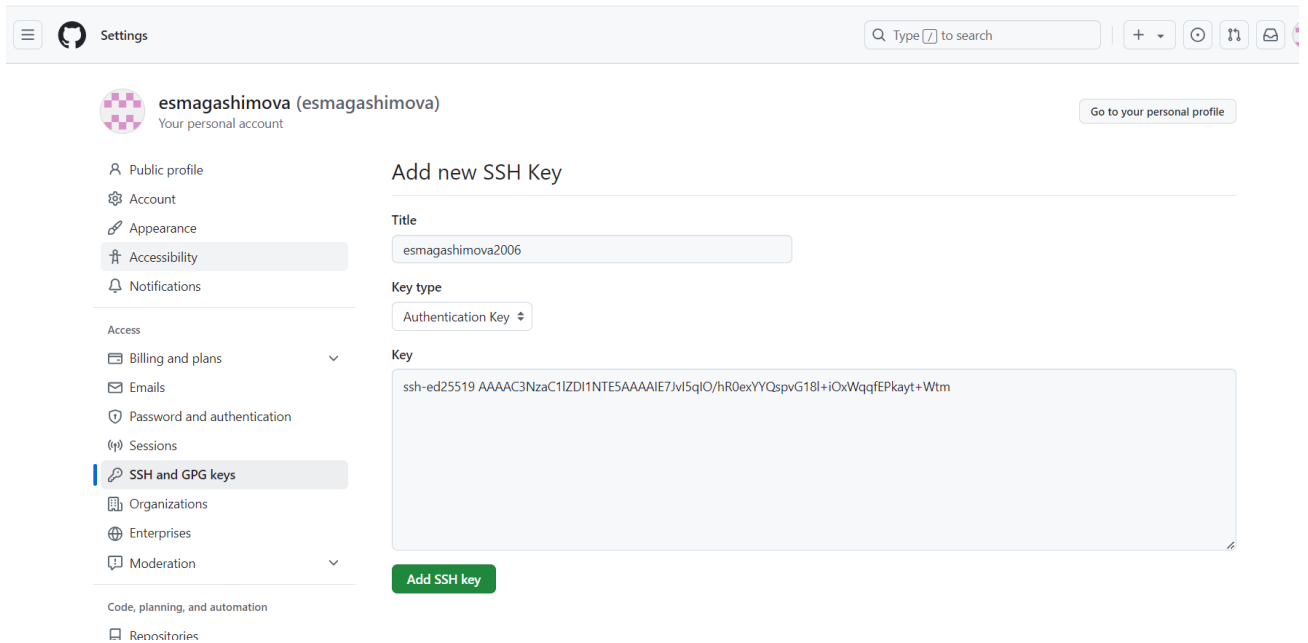


Рис. 12: Добавление ключа

## 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/2022-2023/“Архитектура компьютера” рекурсивно. Далее проверяю с помощью ls, действительно ли были созданы необходимые мне каталоги (рис. 13).

```
esmagashimova@fedora:~$ mkdir -p work/study/2024-2025/"Архитектура компьютера"
esmagashimova@fedora:~$ mkdir -p work/study/2024-2025/"Архитектура компьютера"
esmagashimova@fedora:~$ ды
bash: ды: команда не найдена...
esmagashimova@fedora:~$ ls
ESMA0161_      work      Документы  Изображения  Общедоступные  Шаблоны
ESMA0161_.pub  Видео    Загрузки   Музыка        'Рабочий стол'
```

Рис. 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. 14).

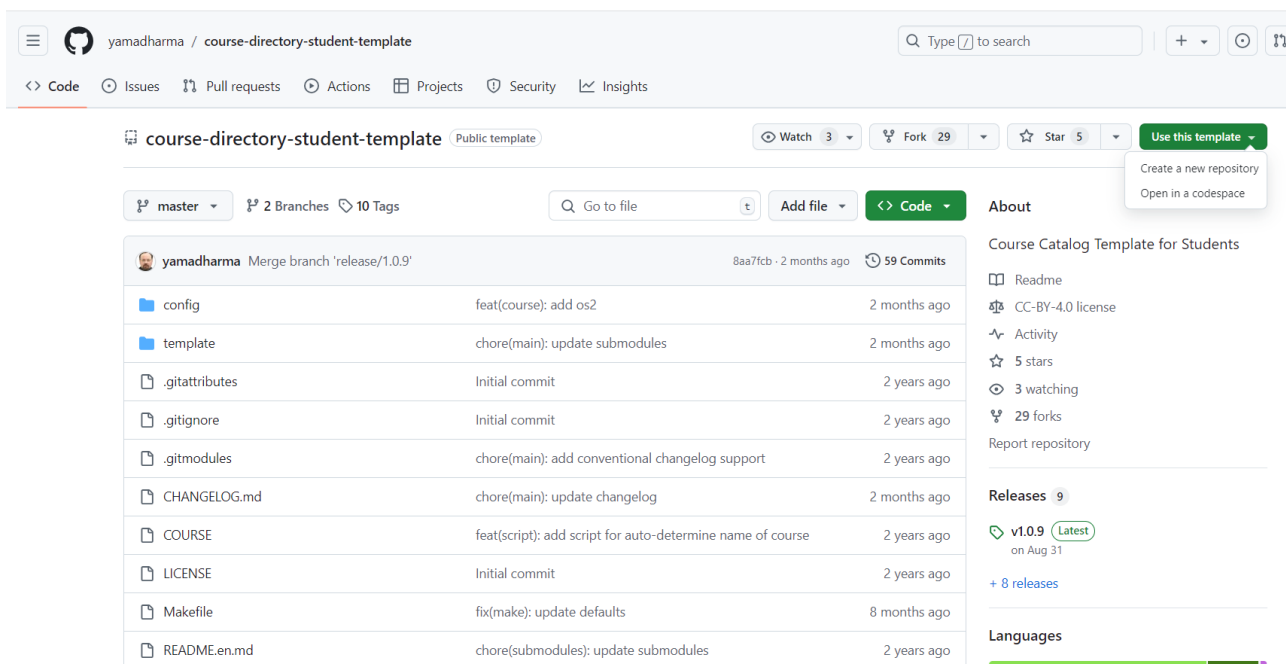


Рис. 14: Страница шаблона для репозитория

В открывшемся окне задаю имя репозитория (Repository name): study\_2022-2023\_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

yamadharma/course-directory-student-template

Start your repository with a template repository's contents.

☐ Include all branches

Copy all branches from yamadharma/course-directory-student-template and not just the default branch.

Owner \*

esmagashimova

Repository name \*

study\_2024-2025\_arh-pc

✔ Your new repository will be created as study\_2024-2025\_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 fictional-octo-palm-tree ?

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.

Рис. 15: Окно создания репозитория

Репозиторий создан (рис. 16).

study\_2024-2025\_arh-pc Public

generated from [yamadharm/course-directory-student-template](#)

Pin Unwatch 1

master 1 Branch 0 Tags

Go to file t

Add file

<> Code

Initial commit d3158c4 · now 1 Commit

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

Рис. 16: Созданный репозиторий

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

```
esmagashimova@fedora:~$ cd ~/work/study/2024-2025/'Архитектура компьютера'
esmagashimova@fedora:~/work/study/2024-2025/Архитектура компьютера$ S
```

Рис. 17: Перемещение между директориями

Клонирую созданный репозиторий с помощью команды `git clone --recursive git@github.com:/study_2022-2023_arh-pc.git arch-pc` (рис. 18).

```
esmagashimova@fedora:~/work/study/2024-2025/Архитектура компьютера$ git clone --recursive git@github.com:esmagashimova/study_2024-2025_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:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdKr4UvC0qU.
```

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).

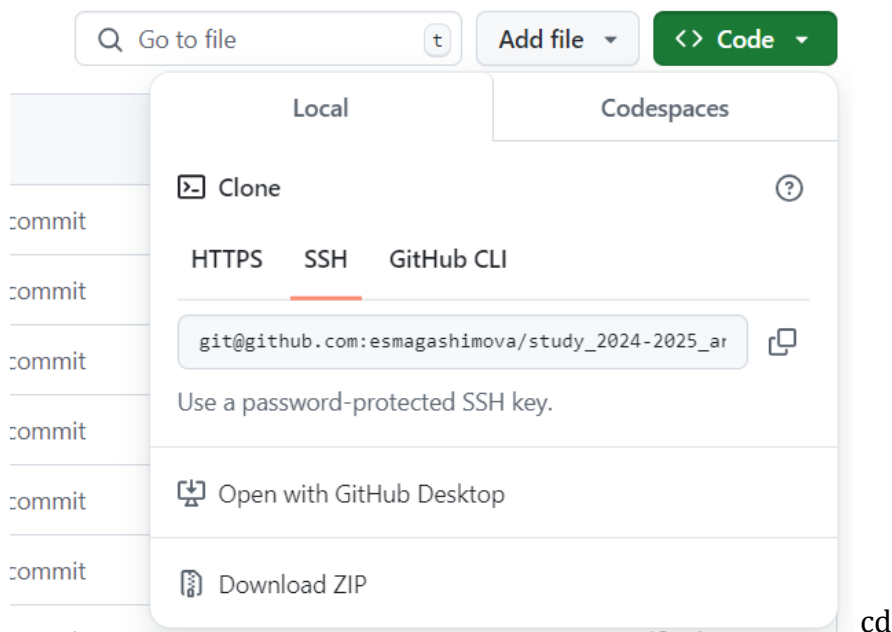


Рис. 19: Окно с ссылкой для копирования репозитория

## 4.6 Настройка каталога курса

Перехожу в каталог `arch-pc` с помощью утилиты `cd` и удаляю лишние файлы с помощью утилиты `rm` (рис. 21).

```
esmagashimova@fedora:~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc$ rm package.json
(evdvorkina@evdvorkina) - [~/../study/2022-2023/Архитектура компьютера/arch-pc]
$ rm package.json
```

Рис. 21: Удаление файлов

Создаю необходимые каталоги и 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).

```
esmagashimova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc$ git add .
esmagashimova@fedora:~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc$ git commit -m 'feat(main): make course structure'
[master e570140] feat(main): make course structure
223 files changed, 53681 insertions(+), 14 deletions(-)
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
```

Рис. 23: Добавление и сохранение изменений на сервере

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

```
esmagashimova@fedora:~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc$ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 341.40 КиБ | 2.09 МиБ/с, готово.
Total 35 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:esmagashimova/study_2024-2025_arh-pc.git
 d3158c4..e570140 master -> master
esmagashimova@fedora:~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc$
```

Рис. 24: Выгрузка изменений на сервер

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

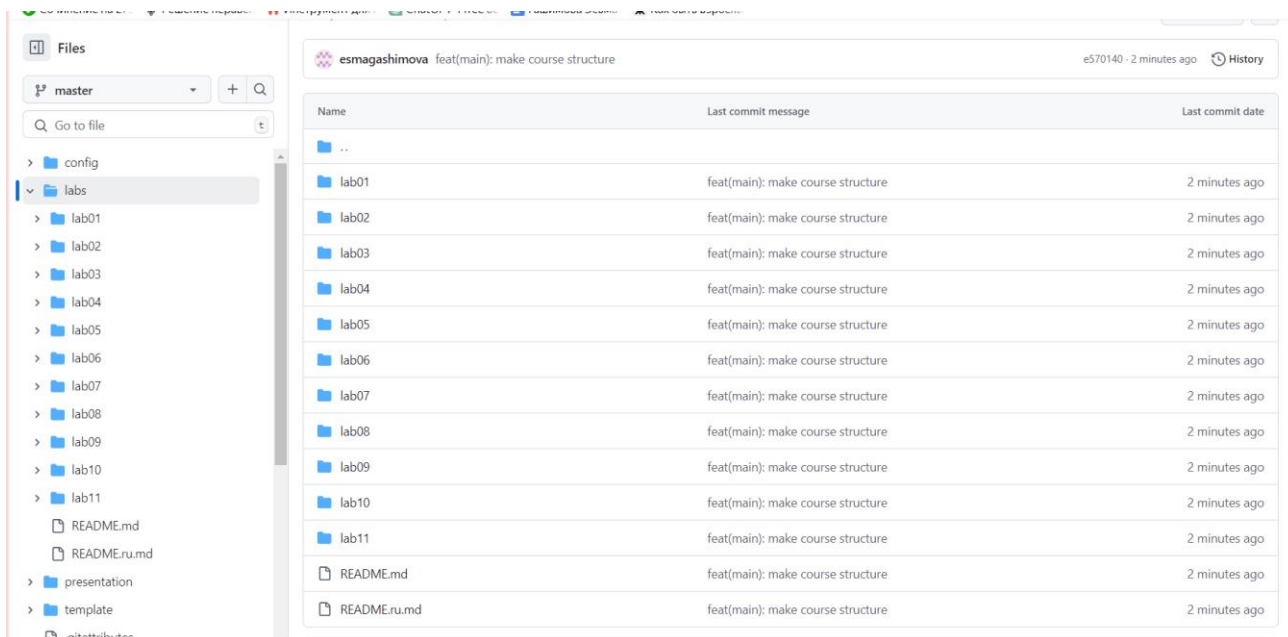


Рис. 25: Страница репозитория

#### 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. 26).

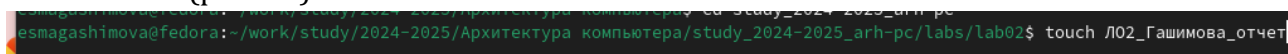


Рис. 26: Создание файла

Оформить отчет я смогу в текстовом процессоре LibreOffice Writer, найдя его в меню приложений. После открытия текстового процессора открываю в нем созданный файл и могу начать в нем работу над отчетом (рис. 28).

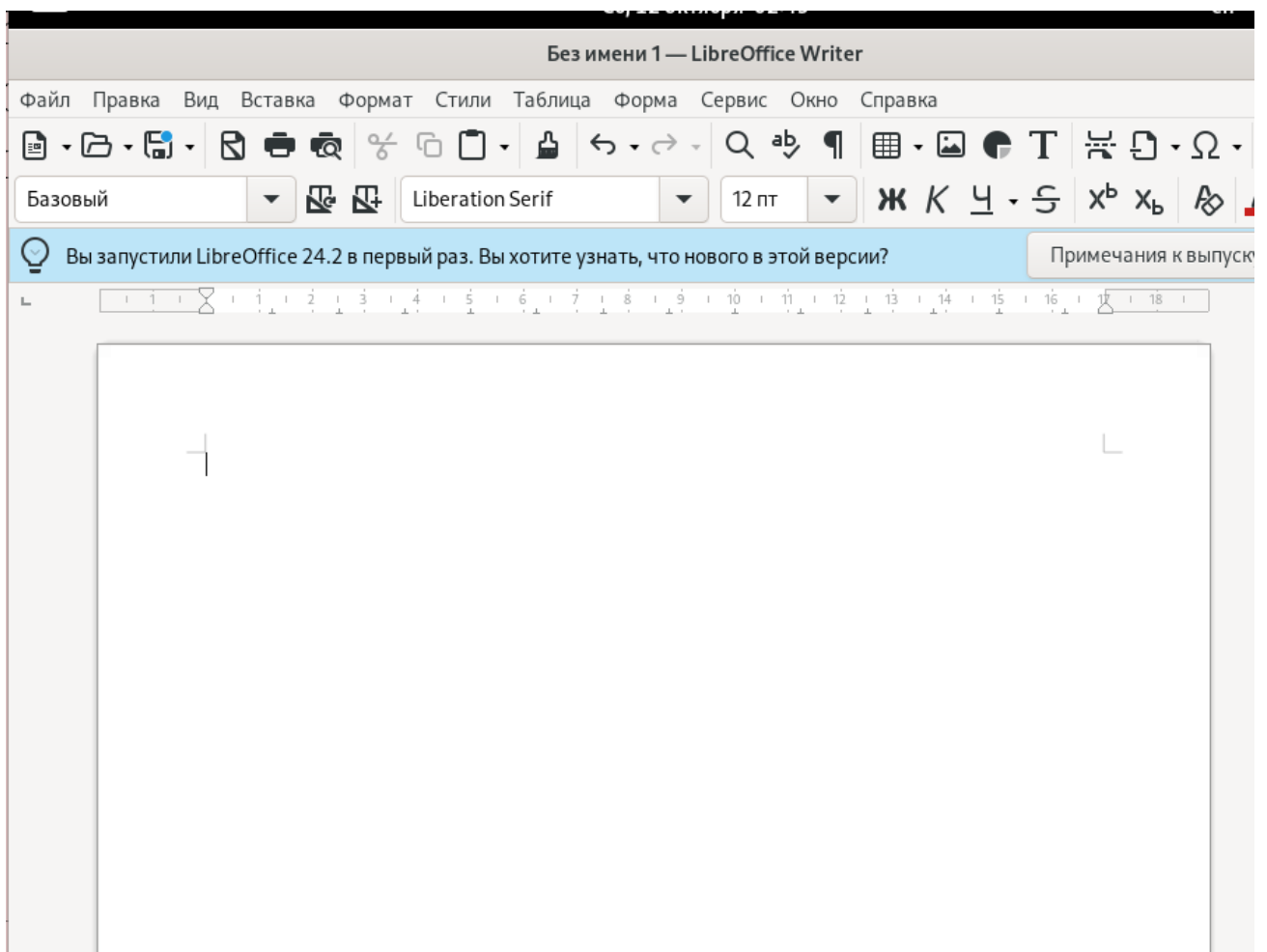


Рис. 28: Работа с отчетом в текстовом процессоре

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

```
esmagashimova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab02$ cd ..
esmagashimova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs$ cd ..
esmagashimova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc$ cd lab01/
bash: cd: lab01/: Нет такого файла или каталога
esmagashimova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc$ cd lab01
bash: cd: lab01: Нет такого файла или каталога
esmagashimova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc$ cd labs
esmagashimova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs$ cd lab01
esmagashimova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab01$ cd report
esmagashimova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab01/report$
```

Рис. 29: Перемещение между директориями

Проверяю местонахождение файлов с отчетами по первой и второй лабораторным работам. Они должны быть в подкаталоге домашней директории «Загрузки», для проверки использую команду ls (рис. 30).

1

```
esmagashimova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab01/report$ ls ~/Загрузки
```

*Рис. 30: Проверка местонахождения файлов*

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

```
imova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab01/report$ cp ~/Загрузки/Л01_Гашимова_отче
work/study/2024-2025/Архитектура\ компьютера/study_2024-2025_arh-pc/labs/lab01/report
imova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab01/report$ ls
ge Makefile pandoc report.md Л01_Гашимова_отчет.pdf
imova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab01/report$
```

*Рис. 31: Копирование файла*

3. Добавляю с помощью команды `git add` в коммит созданные файлы: Л01\_Гашимова\_отчет (рис. 34).

```
imova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab01/report$ git add Л01_Гашимова_отчет.pdf
```

*Рис. 34: Добавление файла на сервер*

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

```
imova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab01/report$ cd ../; cd ..
imova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs$ cd lab01/report/
```

*Рис. 35: Перемещение между директориями*

Добавляю файл Л01\_Гашимова\_отчет (рис. 36).

```
imova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab01/report$ git add Л01_Гашимова_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).

```
imova@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab01/report$ git commit -m "Add existing file"
[36cf] Add existing file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Л01_Гашимова_отчет.pdf
```

*Рис. 37: Подкаталоги и файлы в репозитории*

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

```
va@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab01/report$ git push -f origin master
е объектов: 10, готово.
ектов: 100% (10/10), готово.
ктов: 100% (6/6), готово.
ктов: 100% (6/6), 1.58 МиБ | 1.36 МиБ/с, готово.
ta 3), reused 0 (delta 0), pack-reused 0 (from 0)
olving deltas: 100% (3/3), completed with 3 local objects.
Push: esmagashimova/study_2024-2025_arh-pc.git
65186cf master -> master
```

Рис. 38: Отправка в центральный репозиторий сохраненных изменений

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

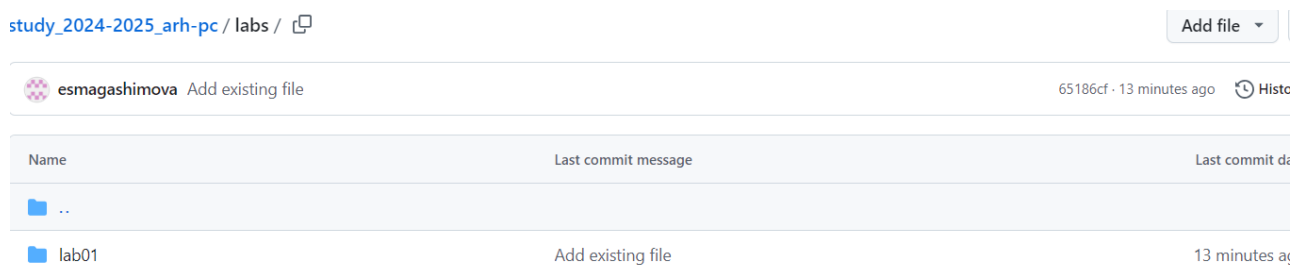


Рис. 39: Страница каталога в репозитории

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



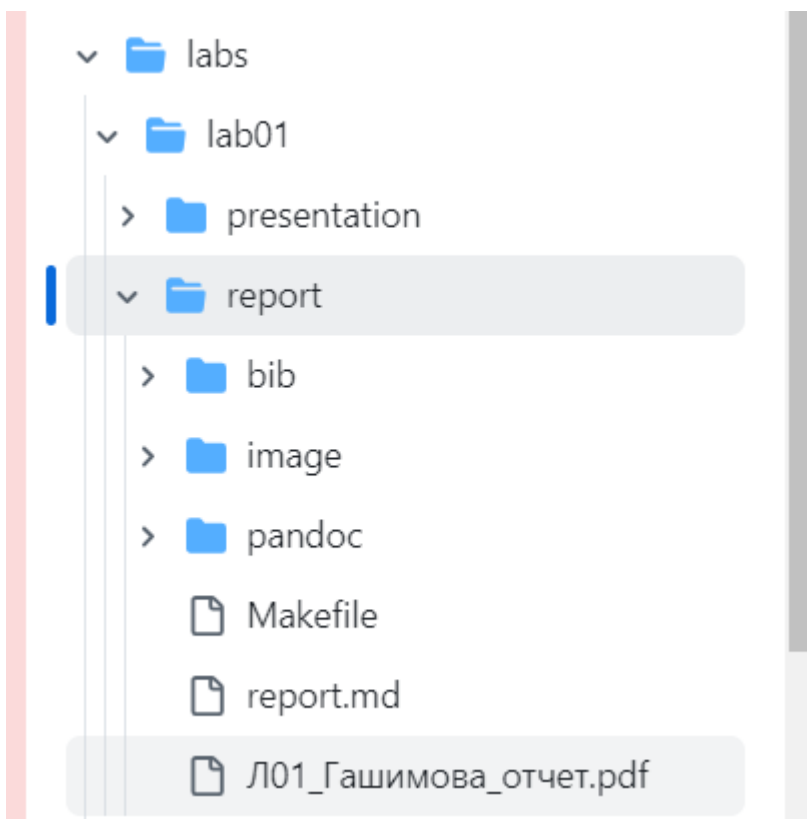


Рис. 40: Страница последних изменений в репозитории

Видно, что отчеты по лабораторным работам находятся в соответствующих каталогах репозитория: отчет по первой - в lab01/report (рис. 41)

[study\\_2024-2025\\_arh-pc](#) / [labs](#) / [lab01](#) / report /


 esmagashimova Add existing file	
Name	Last commit message
..	
bib	feat(main): make course structure
image	feat(main): make course structure
pandoc	feat(main): make course structure
Makefile	feat(main): make course structure
report.md	feat(main): make course structure
Л01_Гашимова_отчет.pdf	Add existing file

Рис. 41: Каталог lab01/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](#)