

# 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: Гасанова Шакира Чингизовна

Группа: НКАбд-05-24

МОСКВА

2024 г.

# Содержание

<b>1 Цель работы .....</b>	<b>3</b>
<b>2 Задание .....</b>	<b>4</b>
<b>3 Теоретическое введение .....</b>	<b>5</b>
<b>4 Выполнение лабораторной работы</b>	
4.1 Настройка GitHub.....	7
4.2 Базовая настройка Git.....	7
4.3 Создание SSH-ключа.....	8
4.4 Создание рабочего пространства и репозитория курса на основе шаблона.....	10
4.5 Создание репозитория курса на основе шаблона.....	10
4.6 Настройка каталога курса.....	13
4.7 Выполнение заданий для самостоятельной работы.....	14
<b>5 Выводы .....</b>	<b>18</b>
<b>6 Источники .....</b>	<b>19</b>

## **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.

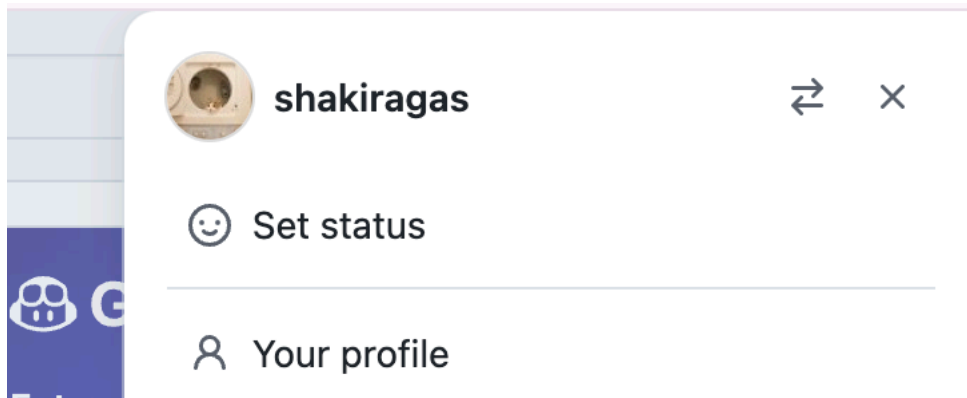


Рис.1 Создание учётной записи

### 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. 2).

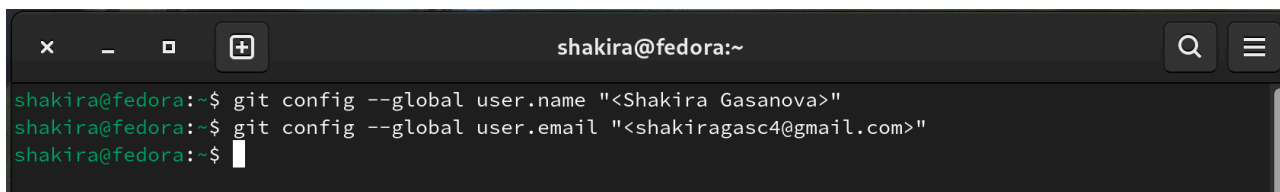


Рис.2 Предварительная конфигурация гит

Настраиваю utf-8 в выводе сообщений git для корректного отображения символов (рис. 3).

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

Рис.3 Настройка кодировки

Задаю имя «master» для начальной ветки и задаю параметр `autocrlf` со значением `input`, так как я работаю в системе Linux, чтобы конвертировать CRLF в LF только при коммитах. CR и LF – это символы, которые можно использовать для обозначения разрыва строки в текстовых файлах (рис. 4).

```
shakira@fedora:~$ git config --global init.defaultBranch master
shakira@fedora:~$ git config --global core.autocrlf input
```

Рис.4 Создание имени для начальной ветки и задание параметра autocrlf

Задаю параметр `safecrlf` со значением `warn`, так Git будет проверять преобразование на обратимость (рис. 5). При значении `warn` Git только выведет предупреждение, но будет принимать необратимые конвертации.

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

Рис.5 Параметр safecrlf

### 4.3 Создание SSH-ключа

Для последующей идентификации пользователя на сервере репозитория требуется сгенерировать пару ключей (приватный и публичный). Для этого использую команду `ssh-keygen -C "Имя Фамилия, work@email"`, где указываются имя владельца и его электронная почта. Сгенерированный ключ будет автоматически сохранён в каталоге `~/.ssh/` (рис. 6).

```
shakira@fedora:~$ ssh-keygen -C "Shakira Gasanova <shakiragasc4@gmail.com>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/shakira/.ssh/id_ed25519):
Created directory '/home/shakira/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/shakira/.ssh/id_ed25519
Your public key has been saved in /home/shakira/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:EzIxkDuKSA04Sq7l04jVovGCYA6hNimh2M9XVdUK/M0 Shakira Gasanova <shakiragasc4@gmail.com>
The key's randomart image is:
+--[ED25519 256]--+
| . .oo .o.... |
|=o . o . o . |
|B=+. .o .. o + |
|XB=+. .o.. o E |
| %X.B . .S |
|O.* + . . |
| . . . |
|-----[SHA256]-----+
shakira@fedora:~$
```

Рис.6 Генерация SSH-ключа

Открываю браузер, захожу на сайт GitHub. Открываю свой профиль и выбираю страницу «SSH and GPG keys». Нажимаю кнопку «New SSH key» (рис. 7).

#### SSH keys

New SSH key

There are no SSH keys associated with your account.

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

Рис.7 Окно SSH

Копирую открытый ключ из директории, в которой он был сохранен, с помощью утилиты `xclip` (рис. 8).

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

Рис.8 Копирование ключа



Вставляю скопированный ключ в поле «Key». В поле Title указываю имя для ключа. Нажимаю «Add SSH-key», чтобы завершить добавление ключа (рис. 9).

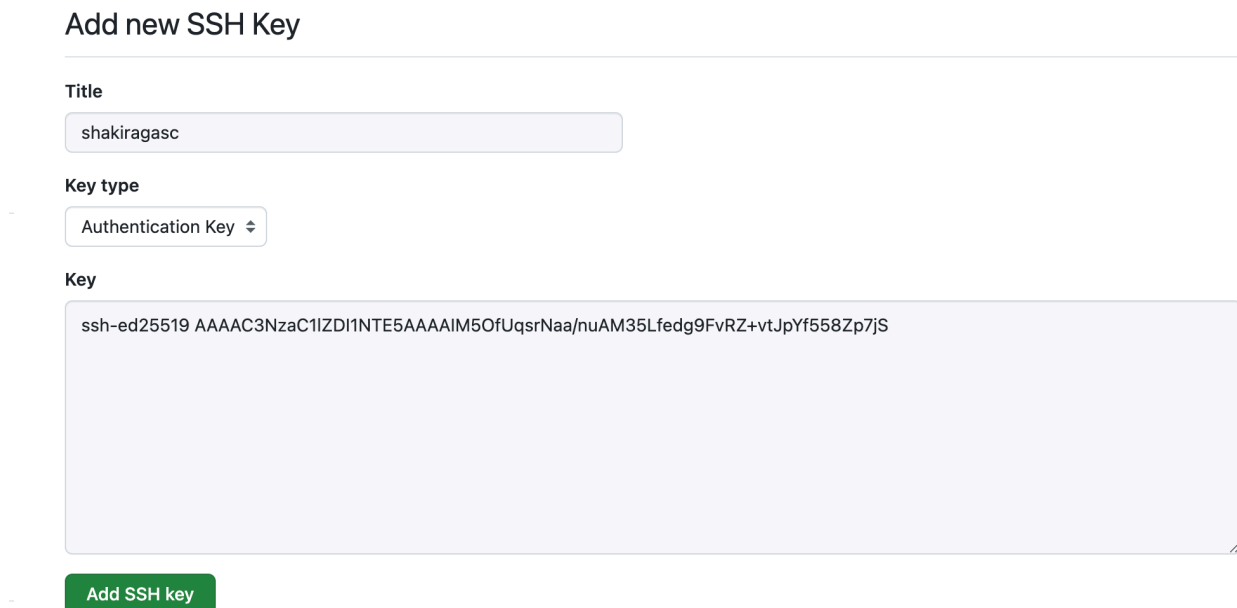


Рис.9 Добавление ключа

Теперь появился ключ (рис. 10).

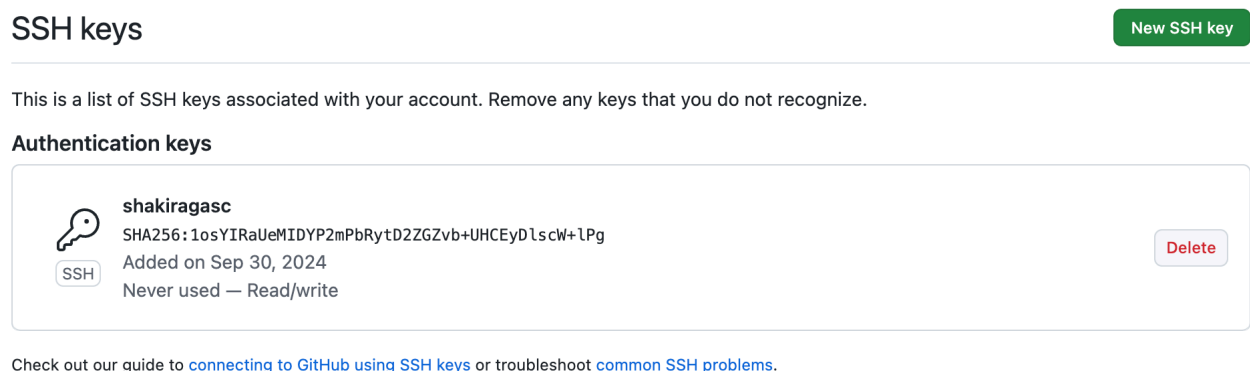


Рис.10 Созданный SSH-ключ

## 4.4 Создание рабочего пространства и репозитория курса на основе шаблона

Закрываю браузер и открываю терминал. С помощью утилиты `mkdir` создаю рабочую директорию. Использую ключ `-p` для рекурсивного создания всех необходимых каталогов по пути `~/work/study/2024-2025/“Архитектура компьютера”`. Затем проверяю с помощью команды `ls`, были ли успешно

созданы нужные каталоги (рис. 11).

```
shakira@fedora:~$ mkdir -p work/study/2024-2025/"Архитектура компьютера"
shakira@fedora:~$ ls
newdir    parentdir  Видео      Загрузки   Музыка     'Рабочий стол'
parentdir work       Документы  Изображения Общедоступные Шаблоны
shakira@fedora:~$
```

Рис.11 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. 12).

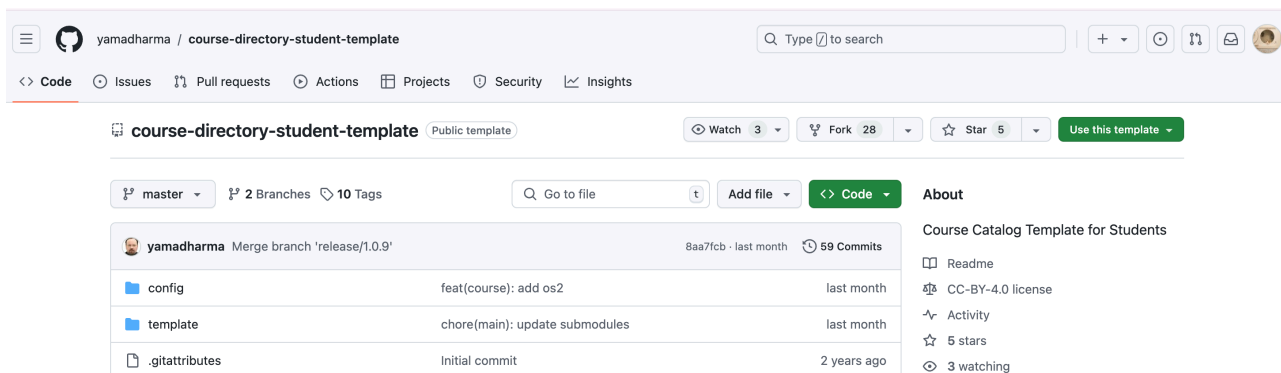


Рис.12 Шаблон для репозитория

В открывшемся окне задаю имя репозитория (Repository name): study\_2024-2025\_arh-рс и создаю репозиторий, нажимаю на кнопку «Create repository from template» (рис. 13).

## 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 \*

 shakiragas ▾

Repository name \*

/ study\_2024-2025\_arh-т

✔ study\_2024-2025\_arh-pc is available.

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

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.13 Окно создания репозитория



## Репозиторий создан (рис.14).












 **study\_2024-2025\_arh-pc** Public

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


 Pin  Unwatch 1 ▾  Fork 0 ▾  Star 0 ▾


 master ▾  1 Branch  0 Tags   Add file  Code ▾


 **shakiragas** Initial commit efee339 · 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
 Makefile	Initial commit	now
 README.en.md	Initial commit	now
 README.git-flow.md	Initial commit	now


No description, website, or topics provided.


 Readme

 CC-BY-4.0 license

 Activity

 0 stars

 1 watching

 0 forks

**Releases**

No releases published

[Create a new release](#)

**Packages**

No packages published

[Publish your first package](#)

**Languages**

Рис.14 Созданный репозиторий

Через терминал перехожу в созданный каталог курса с помощью утилиты `cd` (рис. 15).

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

Рис.15 Переход в каталог курса

Клонирую созданный репозиторий с помощью команды `git clone -recursive git@github.com:/study_2024-2025_arh-pc.git` `arch-pc` (рис. 16).

```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера$ git clone --recursive git@github.com:shakiragas/study_2024-2025_arh-pc.git
Клонирование в «study_2024-2025_arh-pc»...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 33 (delta 1), reused 18 (delta 0), pack-reused 0 (from 0)
Получение объектов: 100% (33/33), 18.82 КиБ | 9.41 МиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/shakira/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-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)
Получение объектов: 100% (111/111), 102.17 КиБ | 1.08 МиБ/с, готово.
Определение изменений: 100% (42/42), готово.
Клонирование в «/home/shakira/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-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)
Получение объектов: 100% (142/142), 341.09 КиБ | 2.26 МиБ/с, готово.
Определение изменений: 100% (60/60), готово.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера$
```

Рис.16 Клонирование репозитория

Копирую ссылку для клонирования на странице созданного репозитория, сначала перейдя в окно «code», далее выбрав в окне вкладку «SSH» (рис. 17).

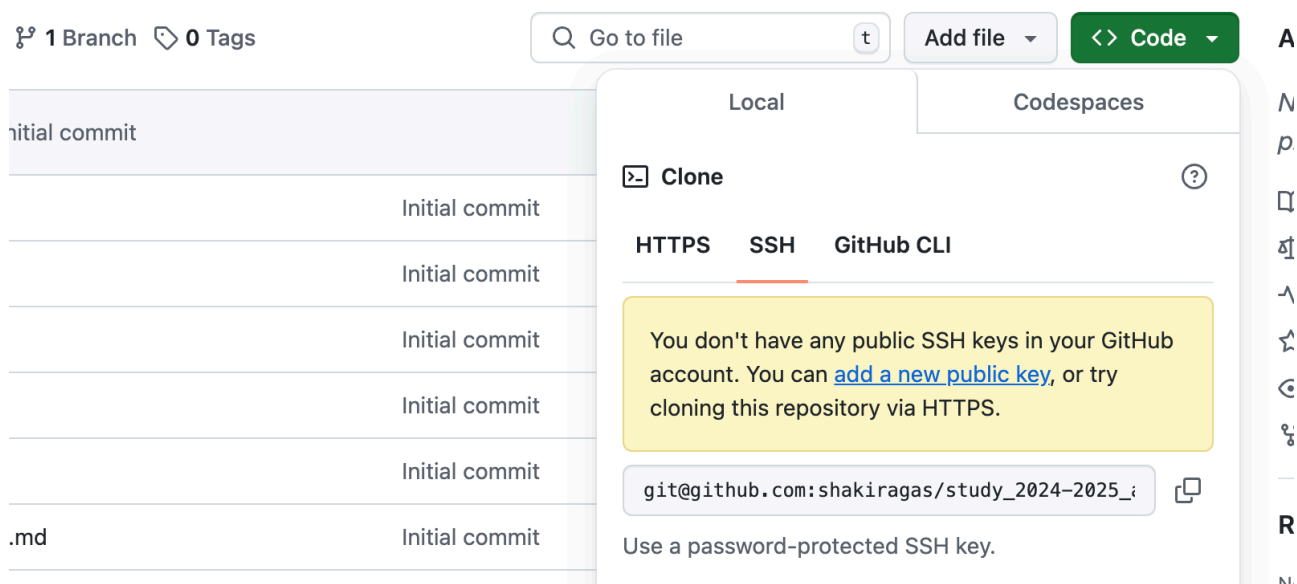


Рис.17 Копирование репозитория

## 4.6 Настройка каталога курса

Перехожу в каталог `arch-pc` с помощью утилиты `cd` (рис. 18).

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

Рис.18 Переход в каталог

Удаляю лишние файлы с помощью утилиты `rm` (рис. 19).

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

Рис.19 Удаление файлов

Создаю необходимые каталоги (рис. 20).

```
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc$ echo arch-pc > COURSE
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc$ make
Usage:
  make <target>

Targets:
  list          List of courses
  prepare      Generate directories structure
  submodule    Update submodules
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc$
```

Рис.20 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` и отправляю всё на сервер с помощью `git push` (рис. 21).

```
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc$ git add .
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc$ git commit -am 'feat(main): make course structure'
[master 5e504c8] feat(main): make course structure
 2 files changed, 1 insertion(+), 14 deletions(-)
 delete mode 100644 package.json
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc$ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 292 байта | 292.00 КиБ/с, готово.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:shakiragas/study_2024-2025_arh-pc.git
 efee339..5e504c8 master -> master
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc$
```

Рис.21 Добавление и сохранение изменений на сервере и опривка

Проверяю правильность выполнения работы на сайте GitHub (рис. 22).

..		
lab01	feat(main): make course structure	2 minutes ago
lab02	feat(main): make course structure	2 minutes ago
lab03	feat(main): make course structure	2 minutes ago
lab04	feat(main): make course structure	2 minutes ago
lab05	feat(main): make course structure	2 minutes ago
lab06	feat(main): make course structure	2 minutes ago
lab07	feat(main): make course structure	2 minutes ago
lab08	feat(main): make course structure	2 minutes ago
lab09	feat(main): make course structure	2 minutes ago
lab10	feat(main): make course structure	2 minutes ago
lab11	feat(main): make course structure	2 minutes ago

Рис.22 Результат

## 4.7 Выполнение заданий для самостоятельной работы

1 Задание. 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. 23).

```
shakira@fedora: ~/work/study/2024-2025/Архитектура_компьютера/arh-pc$ cd ~/work/study/2024-2025/Архитектура_компьютера/arh-pc/labs/lab02/report
shakira@fedora: ~/work/study/2024-2025/Архитектура_компьютера/arh-pc/labs/lab02/report$ touch Л02_Гасанова_отчет
shakira@fedora: ~/work/study/2024-2025/Архитектура_компьютера/arh-pc/labs/lab02/report$
```

Рис.23 Создание файла

Оформить отчет я смогу в текстовом процессоре LibreOffice Writer, найдя его в меню приложений (рис. 24).

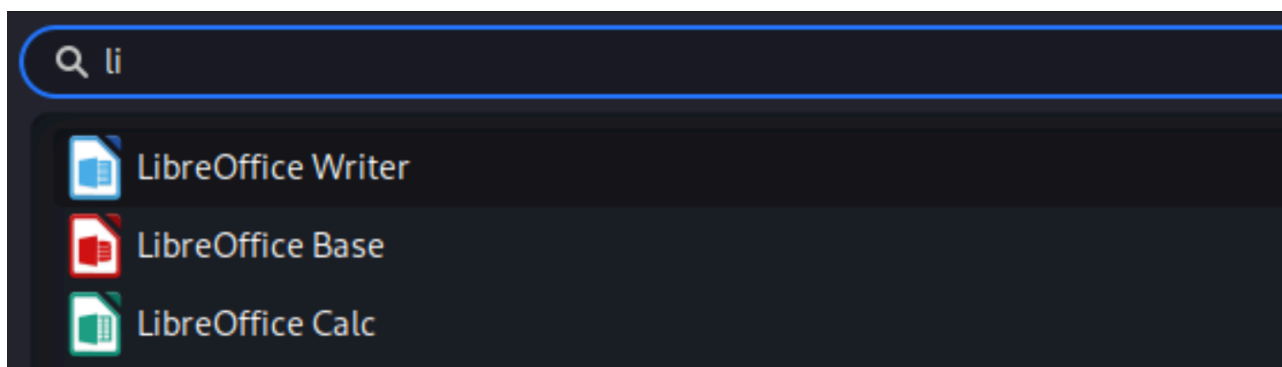


Рис.24 Поиск приложений

После открытия текстового процессора открываю в нем созданный файл и могу начать в нем работу над отчетом (рис. 25).

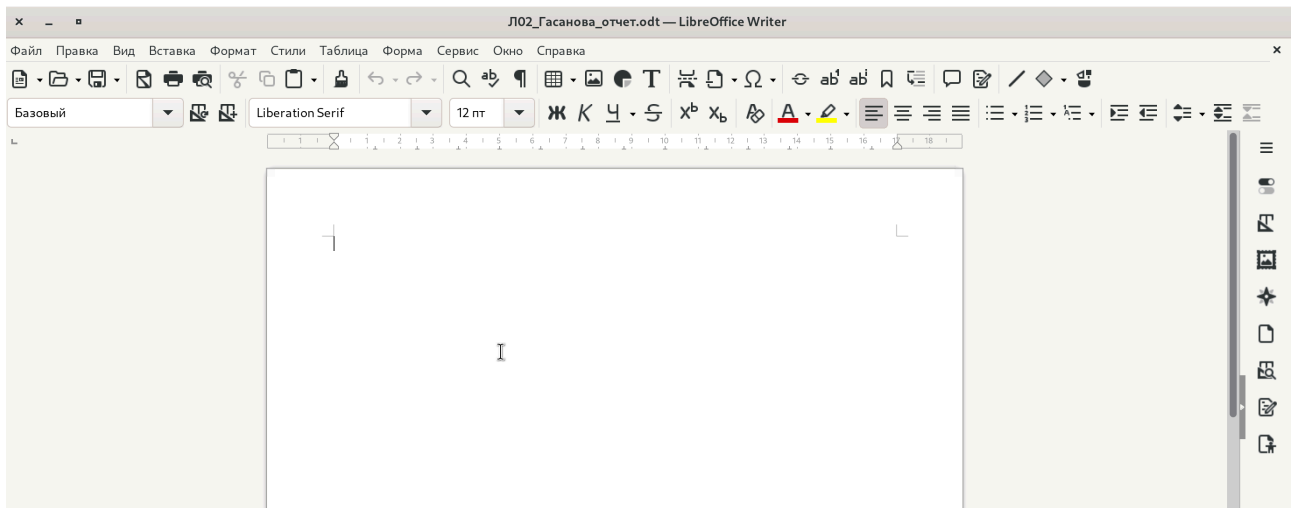


Рис.25 Текстовый редактор

2 Задание. Перехожу из подкаталога lab02/report в подкаталог lab01/report с помощью утилиты cd (рис. 26).

```
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab02/report$ cd ..
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab02$ cd ..
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs$ cd lab01/
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab01$ cd report/
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab01/report$
```

Рис.26 Перемещение между директориями

Проверяю расположение файлов с отчетами по первой и второй лабораторным работам. Они должны находиться в подкаталоге «Загрузки» в домашней директории. Для проверки использую команду ls (рис. 27).

```
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab01/report$ ls ~/Загрузки
Л01_Гасанова_отчет.odt Л02_Гасанова_отчет.odt
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab01/report$
```

Рис.27 Проверка

Копирую первую лабораторную с помощью утилиты cp и проверяю правильность выполнения команды cp с помощью ls (рис. 28).

```
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab01/report$ cp ~/Загрузки/Л01_Гасанова_отчет.odt /home/shakira/work/study/2024-2025/"Архитектура компьютера"/arh-pc/labs/lab01/report
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab01/report$ ls
Л01_Гасанова_отчет.odt
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab01/report$
```

Рис.28 Копирование первой лабораторной работы

Возвращаюсь в подкаталог lab02/report с помощью утилиты cd (рис. 29).

```
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab01/report$ cd ..
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab01$ cd ..
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs$ cd lab02/report
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab02/report$
```

Рис.29 Перемещение между директориями

Копирую её с помощью утилиты cp и проверяю правильность выполнения команды с помощью ls (рис. 30).

```
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab02/report$ cp ~/Загрузки/Л02_Гасанова_отчет.odt /home/shakira/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab02/report
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab02/report$ ls
Л02_Гасанова_отчет
```

Рис.30 Копирование второй лабораторной работы

**3 Задание.** Добавляю с помощью команды `git add` в коммит созданные файлы: Л01\_Гасанова\_отчет и Л02\_Гасанова\_отчет. Не забываю перемещаться по директориям, чтобы найти нужные файлы (рис. 31).

```
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc$ cd labs/lab01/report
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab01/report$ git add Л01_Гасанова_отчет.odt
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab01/report$ cd ..
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs$ cd lab02/report
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab02/report$ git add Л02_Гасанова_отчет.odt
fatal: спецификатор пути «Л02_Гасанова_отчет.odt» не соответствует ни одному файлу
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab02/report$ git add Л02_Гасанова_отчет
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab02/report$
```

Рис.31 Добавление файлов на сервер

Сохраняю изменения на сервере командой `git commit -m "..."` и заодно добавляю пояснение (рис. 32).

```
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab02/report$ git commit -m "Add existing file"
Текущая ветка: master
Ваша ветка опережает «origin/master» на 1 коммит.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)

ничего коммитить, нет изменений в рабочем каталоге
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab02/report$ git push
Перечисление объектов: 100% (10/10), готово.
Подсчет объектов: 100% (10/10), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (9/9), 7.24 КиБ | 3.62 МБ/с, готово.
Total 9 (delta 1), reused 1 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:shakiragas/study_2024-2025_arh-pc.git
 3435f84..2858428 master -> master
shakira@fedora: ~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab02/report$
```

Рис.32 Сохранение изменений на сервере

Проверяю на сайте GitHub правильность выполнения заданий и проверяю наличие пояснений (рис. 33, рис. 34).

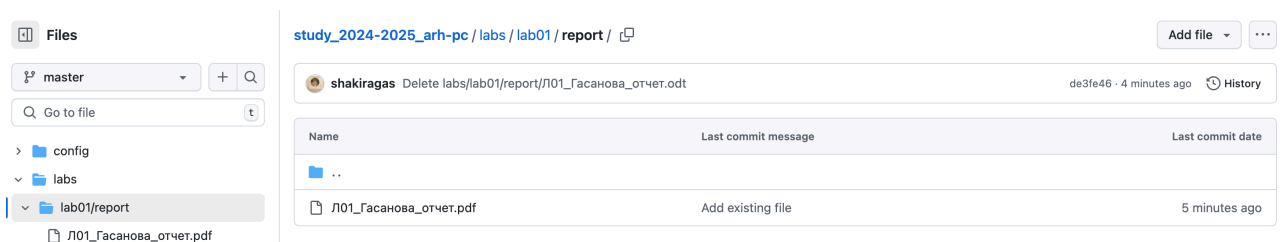


Рис.33 Проверка на сайте

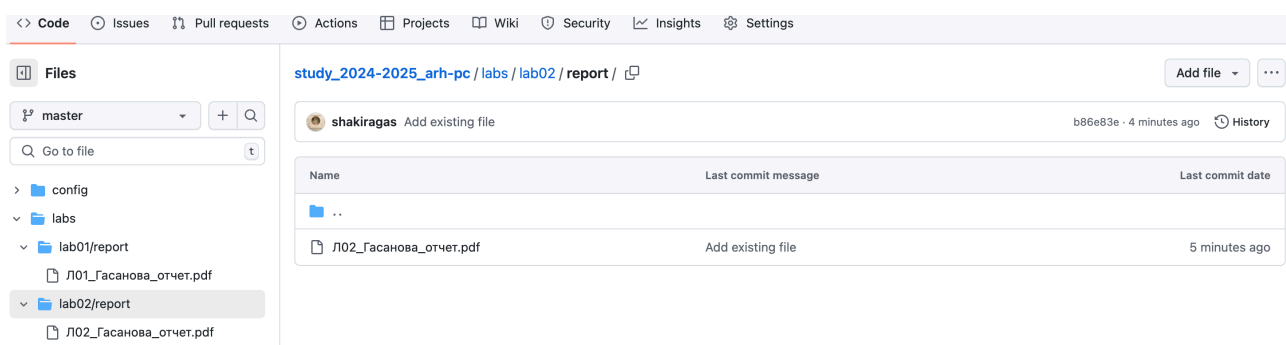


Рис.34 Проверка на сайте



При просмотре изменений замечаю, что были добавлены файлы с отчетами по лабораторным работам (рис. 35, рис. 36).

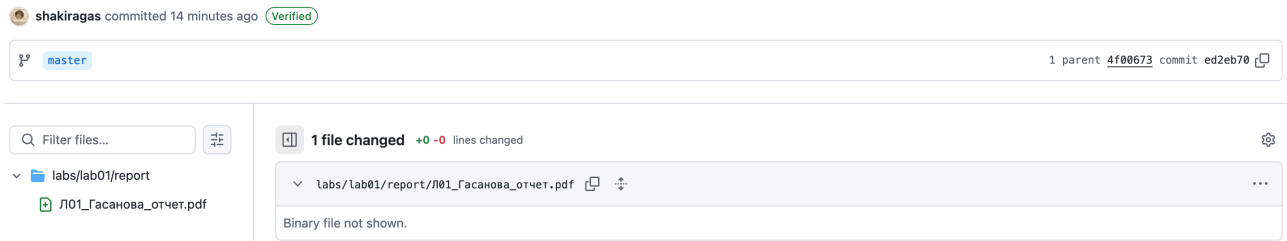


Рис.35 Проверка на сайте

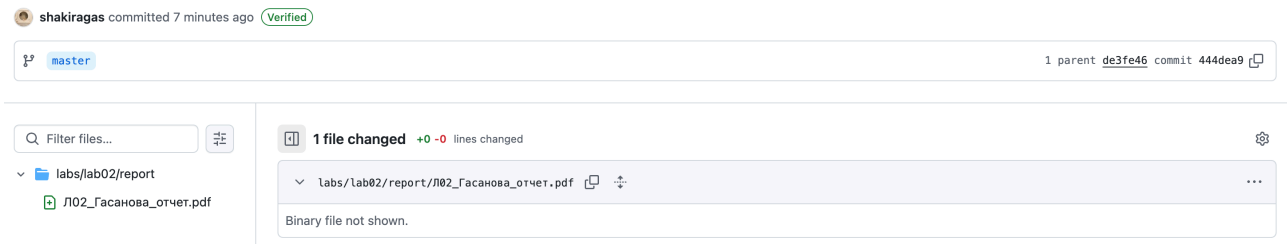


Рис.36 Проверка на сайте

## **5 Выводы**

В ходе выполнения этой лабораторной работы я изучила принципы и применение инструментов контроля версий, а также приобрела практические навыки работы с системой Git.

## **6 Источники**

1. [Архитектура ЭВМ \(rudn.ru\)](http://rudn.ru)