

# 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: Самойлова С.Д.

Группа: НКАбд-04-24

МОСКВА

2024 г.

## **Содержание**

1. Цель работы...
2. Теоретическое введение...
3. Задание...
4. Задание для самостоятельной работы...
5. Выводы...
6. Список литературы...

## **Цель работы**

Целью работы является изучить идеологию и применение средств контроля версий, а также приобрести практические навыки по работе с системой git

# Теоретическое введение

## 1. Системы контроля версий. Общие понятия

Системы контроля версий (VCS) используются для совместной работы над проектами. Обычно основное дерево проекта хранится в локальном или удалённом репозитории с доступом для участников. VCS фиксирует изменения, объединяет их и позволяет откатываться к предыдущим версиям.

В классических системах применяется централизованная модель с единственным репозиторием, где сервер управляет версиями. Пользователь получает нужную версию файлов, вносит изменения и загружает новую версию, при этом старые версии сохраняются. Сервер может использовать дельта-компрессию, сохраняя только изменения между версиями, что позволяет уменьшить объём хранимых данных.

Системы контроля версий предлагают гибкие функции, такие как поддержка нескольких версий одного файла с сохранением общей истории изменений и индивидуальных ветвей. Они фиксируют информацию о том, кто и когда вносил изменения, что хранится в журнале, доступ к которому можно ограничить. В распределённых системах центральный репозиторий не обязателен. Из классических VCS известны CVS и Subversion, а среди распределённых — Git, Bazaar и Mercurial. Принципы их работы схожи, различия в синтаксисе команд.

## 2. Система контроля версий Git

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.

### 3. Основные команды git

Наиболее часто используемые команды git представлены в таблице

| Команда  | Описание  |
|--|---|
| <code>git init</code>                                  | создание основного дерева репозитория   |
| <code>git pull</code>                                  | получение обновлений (изменений) текущего дерева из центрального репозитория  |
| <code>git push</code>                                  | отправка всех произведённых изменений локального дерева в центральный репозиторий   |
| <code>git status</code>                                | просмотр списка изменённых файлов в текущей директории  |
| <code>git diff</code>                                  | просмотр текущих изменения  |
| <code>git add .</code>                                 | добавить все изменённые и/или созданные файлы и/или каталоги  |
| <code>git add<br/>имена_файлов</code>                  | добавить конкретные изменённые и/или созданные файлы и/или каталоги   |
| <code>git rm<br/>имена_файлов</code>                   | удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории)                               |
| <code>git commit -am<br/>'Описание<br/>коммита'</code> | сохранить все добавленные изменения и все изменённые файлы  |
| <code>git checkout -b<br/>имя_ветки</code>             | создание новой ветки, базирующейся на текущей   |
| <code>git checkout<br/>имя_ветки</code>                | переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой) |
| <code>git push origin<br/>имя_ветки</code>             | отправка изменений конкретной ветки в центральный репозиторий   |
| <code>git merge<br/>--no-ff<br/>имя_ветки</code>       | слияние ветки с текущим деревом   |
| <code>git branch -d<br/>имя_ветки</code>               | удаление локальной уже слитой с основным деревом ветки  |
| <code>git branch -D<br/>имя_ветки</code>               | принудительное удаление локальной ветки   |
| <code>git push origin<br/>:имя_ветки</code>            | удаление ветки с центрального репозитория   |

## Задание

### 1. Базовая настройка git

Существует несколько доступных серверов репозитория с возможностью бесплатного размещения данных. Например, <http://bitbucket.org/>, <https://github.com/> и <https://gitflic.ru>. Для выполнения лабораторных работ предлагается использовать Github.

Создаем учётную запись на сайте <https://github.com/> и заполняем основные данные (рис.1)

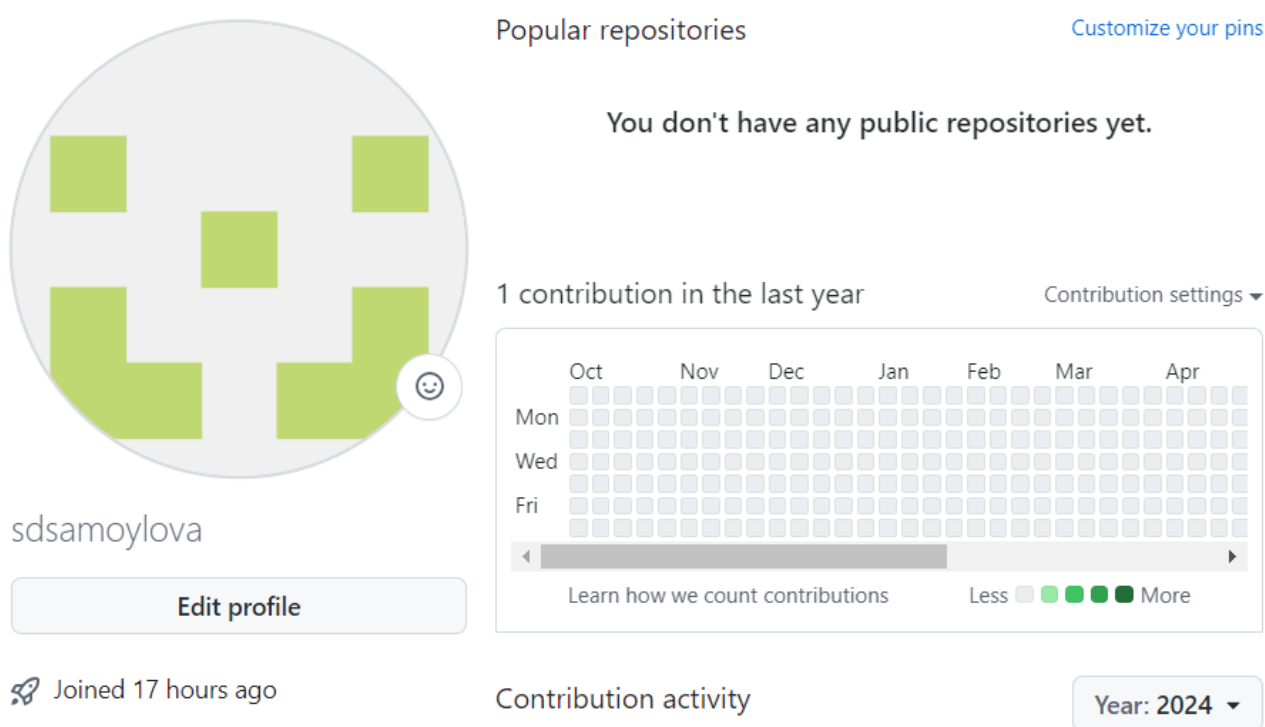


рис.1

Сделаем предварительную конфигурацию git. Откроем терминал и введем следующие команды, указав имя и email владельца репозитория (рис.2)

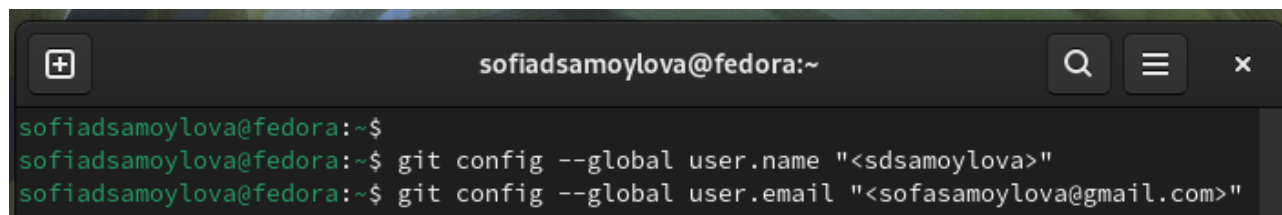
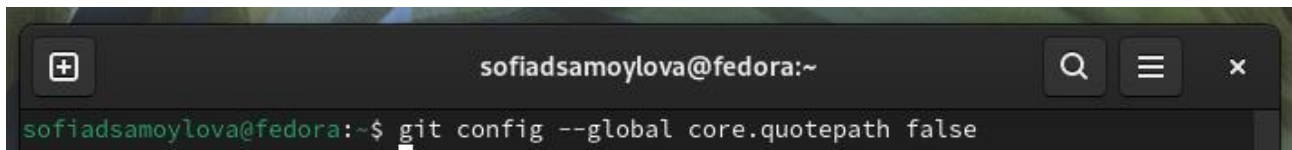


рис.2

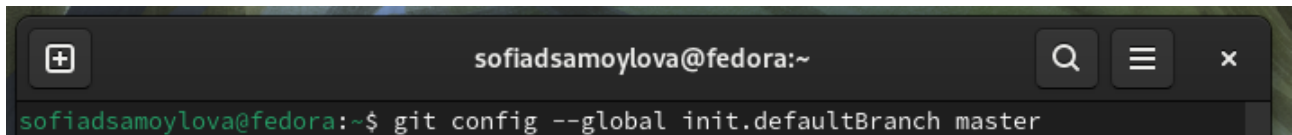
Настроим utf-8 в выводе сообщений git (рис.3)

A terminal window with a dark background. The title bar shows the username 'sofiadsamoylova@fedora:~'. The command 'git config --global core.quotePath false' has been entered and executed, with the prompt returning to the shell.

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

рис.3

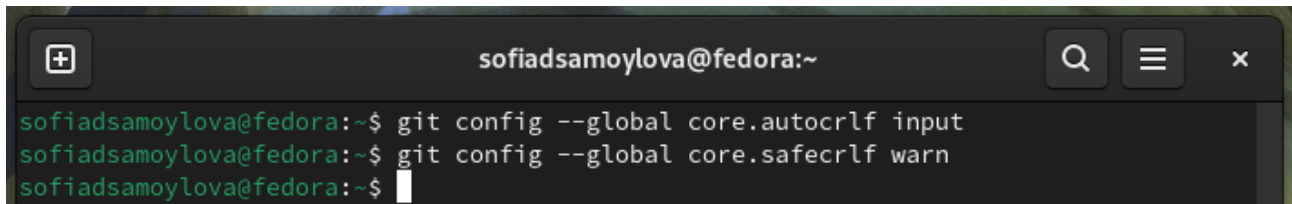
Зададим имя начальной ветки (будем называть её master) (рис.4)

A terminal window with a dark background. The title bar shows the username 'sofiadsamoylova@fedora:~'. The command 'git config --global init.defaultBranch master' has been entered and executed, with the prompt returning to the shell.

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

рис.4

Параметр autocrlf и safecrlf (рис.5)

A terminal window with a dark background. The title bar shows the username 'sofiadsamoylova@fedora:~'. Two commands have been entered and executed: 'git config --global core.autocrlf input' and 'git config --global core.safecrlf warn'. The prompt returns to the shell.

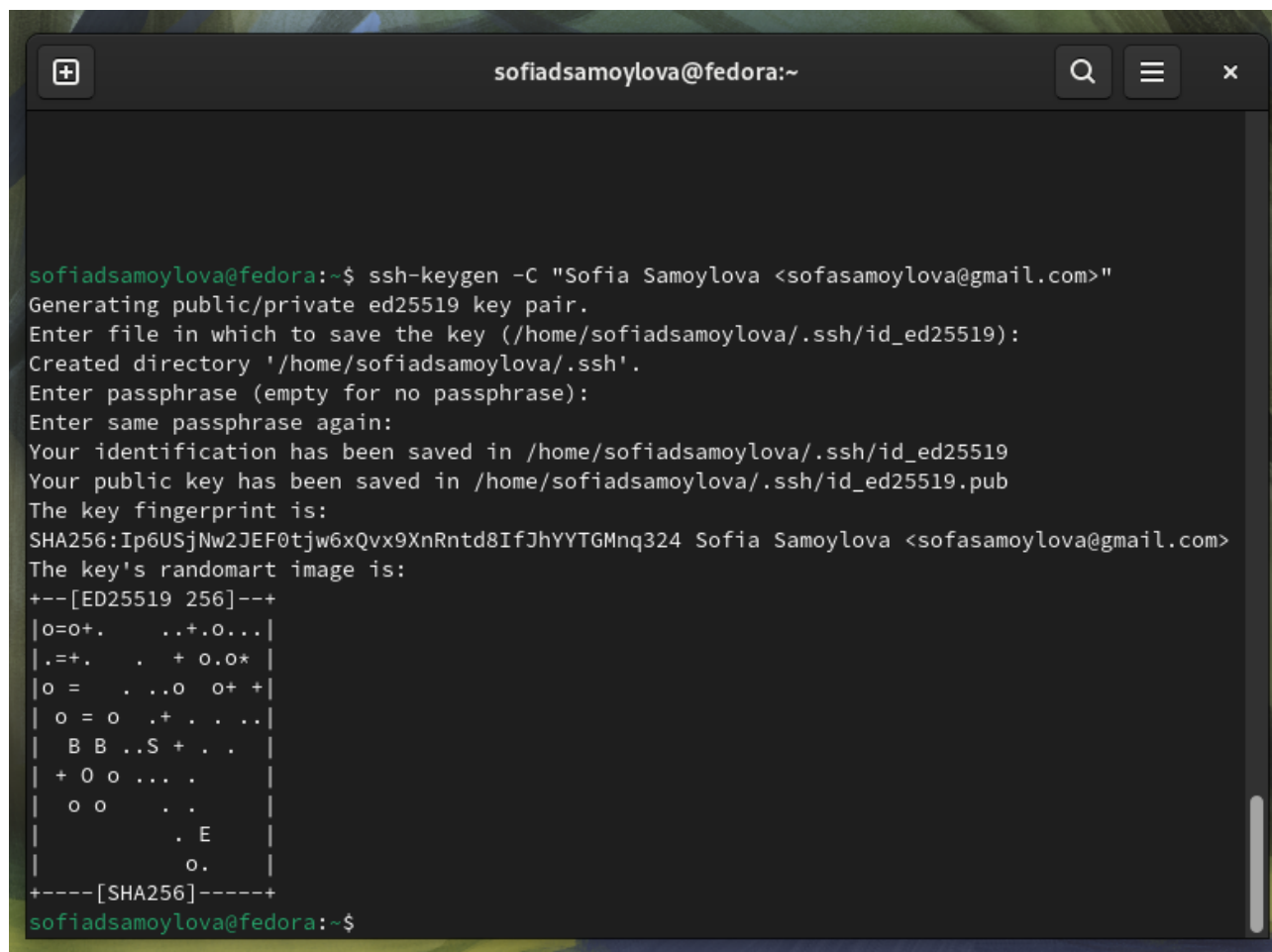
```
sofiadsamoylova@fedora:~$ git config --global core.autocrlf input
sofiadsamoylova@fedora:~$ git config --global core.safecrlf warn
sofiadsamoylova@fedora:~$
```

рис.5

## 2. Создание SSH ключа

Для последующей идентификации пользователя на сервере репозитория необходимо

сгенерировать пару ключей (приватный и открытый) (рис.6)

A terminal window titled 'sofiadsamoylova@fedora:~' with search, menu, and close buttons in the title bar. The terminal shows the execution of the 'ssh-keygen' command with the following output:

```
sofiadsamoylova@fedora:~$ ssh-keygen -C "Sofia Samoylova <sofasamoylova@gmail.com>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/sofiadsamoylova/.ssh/id_ed25519):
Created directory '/home/sofiadsamoylova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/sofiadsamoylova/.ssh/id_ed25519
Your public key has been saved in /home/sofiadsamoylova/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:Ip6USjNw2JEF0tjw6xQvx9XnRntd8IfJhYYTGMnq324 Sofia Samoylova <sofasamoylova@gmail.com>
The key's randomart image is:
+--[ED25519 256]--+
|o=o+.      ..+.o...|
|.=+.      .  + o.o* |
|o =      . ..o  o+ +|
| o = o  .+. . . . |
|  B B ..S + . .  |
| + 0 o ... .    |
|  o o      . .    |
|                . E |
|                o.  |
+-----[SHA256]-----+
sofiadsamoylova@fedora:~$
```

рис.6

Xclip – утилита, позволяющая скопировать любой текст через терминал. Оказывается, в дистрибутиве Linux Kali ее сначала надо установить. Устанавливаю xclip с помощью команды apt-get install с ключом -у от имени суперпользователя, введя в начале команды sudo (рис. 7).



```
sofiadsamoylova@fedora:~$ xclip
bash: xclip: команда не найдена...
Установить пакет «xclip», предоставляющий команду «xclip»? [N/y] y

* Ожидание в очереди...
* Загрузка списка пакетов....
Следующие пакеты должны быть установлены:
xclip-0.13-21.git11cba61.fc40.x86_64  Command line clipboard grabber
Продолжить с этими изменениями? [N/y] y

* Ожидание в очереди...
* Ожидание аутентификации...
* Ожидание в очереди...
* Загрузка пакетов...
* Запрос данных...
* Проверка изменений...
* Установка пакетов...

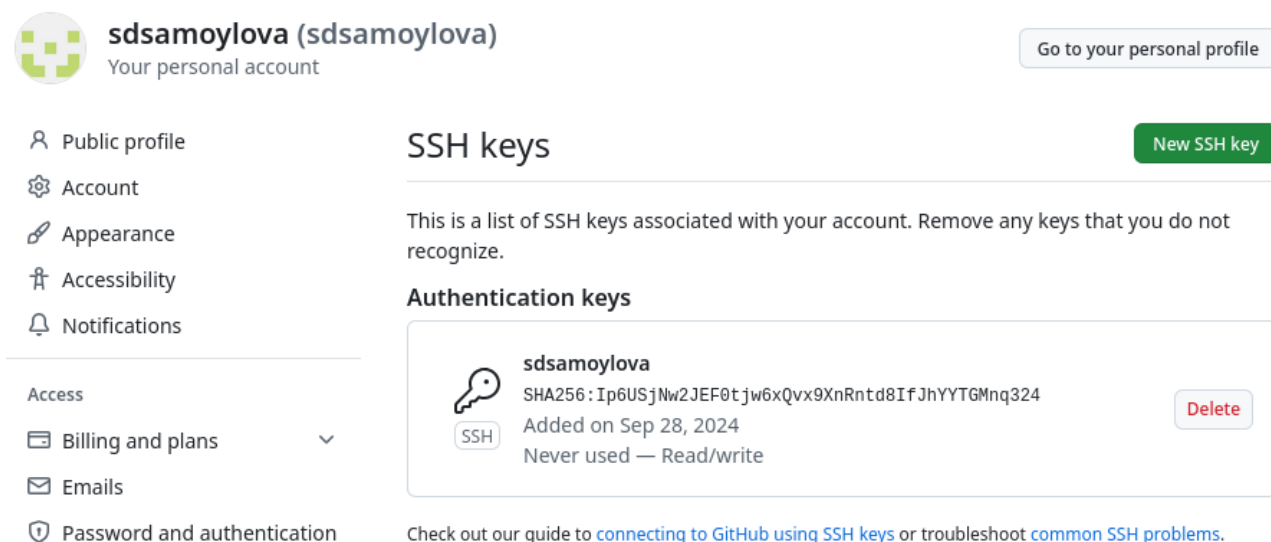
sofiadsamoylova@fedora:~$
```

рис.7

Ключи сохраняются в каталоге ~/.ssh/.

Далее необходимо загрузить сгенерённый открытый ключ. Для этого заходим на сайт <http://github.org/> под своей учётной записью и переходим в меню Setting. После этого выбрать в боковом меню SSH and GPG keys и нажать кнопку New SSH key, скопировав из локальной консоли ключ в буфер обмена и вставляем ключ в появившееся на сайте поле и указываем для ключа имя (Title) (рис.8)

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



The screenshot shows the GitHub 'Settings' page for the user 'sdsamoylova'. On the left is a sidebar with navigation links: Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, and Password and authentication. The main content area is titled 'SSH keys' and includes a 'New SSH key' button. Below this is a list of authentication keys. One key is shown for the user 'sdsamoylova', with a SHA256 fingerprint, added on Sep 28, 2024, and marked as 'Never used'. A 'Delete' button is next to the key. At the bottom, there is a link to a guide on connecting to GitHub using SSH keys.

рис.8

### 3. Создание рабочего пространства и репозитория курса на основе шаблона

При выполнении лабораторных работ следует придерживаться структуры рабочего пространства. Рабочее пространство по предмету располагается в следующей иерархии:

~/work/study/

└── <учебный год>/

└── <название предмета>/

└── <код предмета>/

Например, для 2023–2024 учебного года и предмета «Архитектура компьютера» (код предмета arch-pc) структура каталогов примет следующий вид:

~/work/study/

└── 2023–2024/

└── А р х и т е к т у р а к о м п ь ю т е р а /

└── arch-pc/

└── labs/

└── lab01/

└── lab02/

└── lab03...

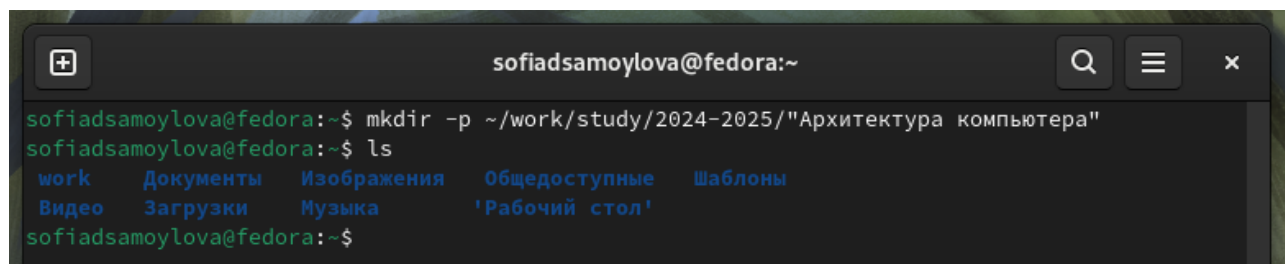
- Каталог для лабораторных работ имеет вид labs.
- Каталоги для лабораторных работ имеют вид lab<номер>, например: lab01, lab02 и т.д.

Название проекта на хостинге git имеет вид: study\_<учебный год>\_<код предмета>

Например, для 2023–2024 учебного года и предмета «Архитектура компьютера» (код предмета arch-pc) название проекта примет следующий вид: study\_2023–2024\_arch-pc

Открываем терминал и создаем каталог для предмета «Архитектура компьютера»:mkdir -p ~/work/study/2024-2025/"Архитектура компьютера"

(рис.9)



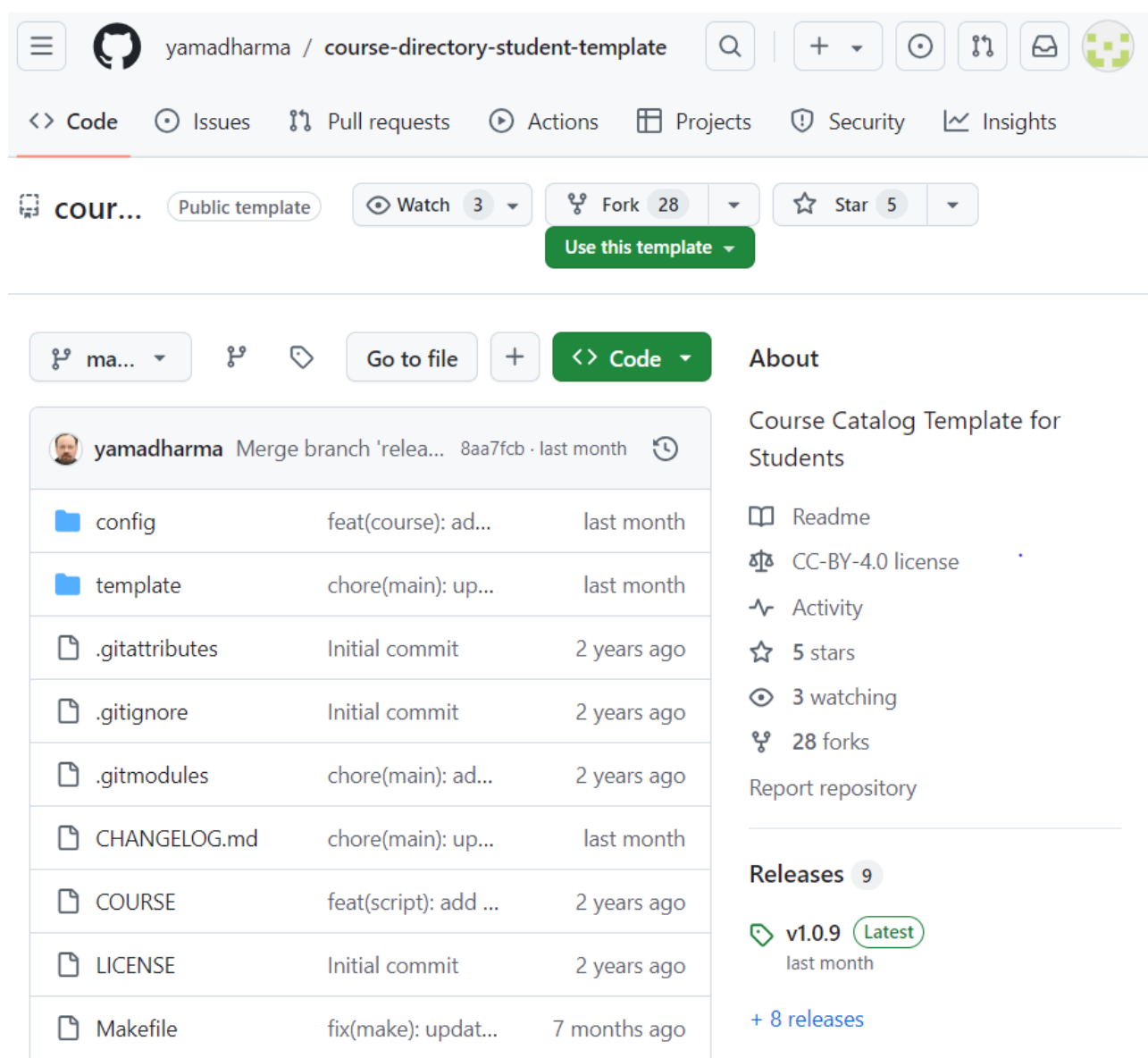
```
sofiadsamoylova@fedora:~  
sofiadsamoylova@fedora:~$ mkdir -p ~/work/study/2024-2025/"Архитектура компьютера"  
sofiadsamoylova@fedora:~$ ls  
work  Документы  Изображения  Общедоступные  Шаблоны  
Видео  Загрузки  Музыка  'Рабочий стол'  
sofiadsamoylova@fedora:~$
```

рис.9

Репозиторий на основе шаблона можно создать через web-интерфейс github.

Переходим на страницу репозитория с шаблоном курса (рис.10)

<https://github.com/yamadharm/course-directory-student-template>



The screenshot shows the GitHub repository page for 'yamadharm / course-directory-student-template'. The repository is a public template with 3 watchers, 28 forks, and 5 stars. A green button 'Use this template' is visible. The repository contains a table of files and folders, including 'config', 'template', '.gitattributes', '.gitignore', '.gitmodules', 'CHANGELOG.md', 'COURSE', 'LICENSE', and 'Makefile'. The 'About' section on the right provides more details about the repository, including the README, CC-BY-4.0 license, and activity. The 'Releases' section shows the latest release 'v1.0.9' as the 'Latest' version, released last month, with a link to '+ 8 releases'.

| File/Folder    | Commit Message        | Commit Date  |
|----------------|-----------------------|--------------|
| config         | feat(course): ad...   | last month   |
| template       | chore(main): up...    | last month   |
| .gitattributes | Initial commit        | 2 years ago  |
| .gitignore     | Initial commit        | 2 years ago  |
| .gitmodules    | chore(main): ad...    | 2 years ago  |
| CHANGELOG.md   | chore(main): up...    | last month   |
| COURSE         | feat(script): add ... | 2 years ago  |
| LICENSE        | Initial commit        | 2 years ago  |
| Makefile       | fix(make): updat...   | 7 months ago |

рис.10

Далее выбираем Use this template.


В открывшемся окне задаем имя репозитория (Repository name) study\_2024–2025\_arch-pc и создаем репозиторий (кнопка Create repository from template).

## 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 \*

 sdsamoylova ▾

Repository name \*

/ study\_2024–2025\_arch-pc

✔ Your new repository will be created as study\_2024-2025\_arch-pc.

The repository name can only contain ASCII letters, digits, and the characters ., -, and \_.

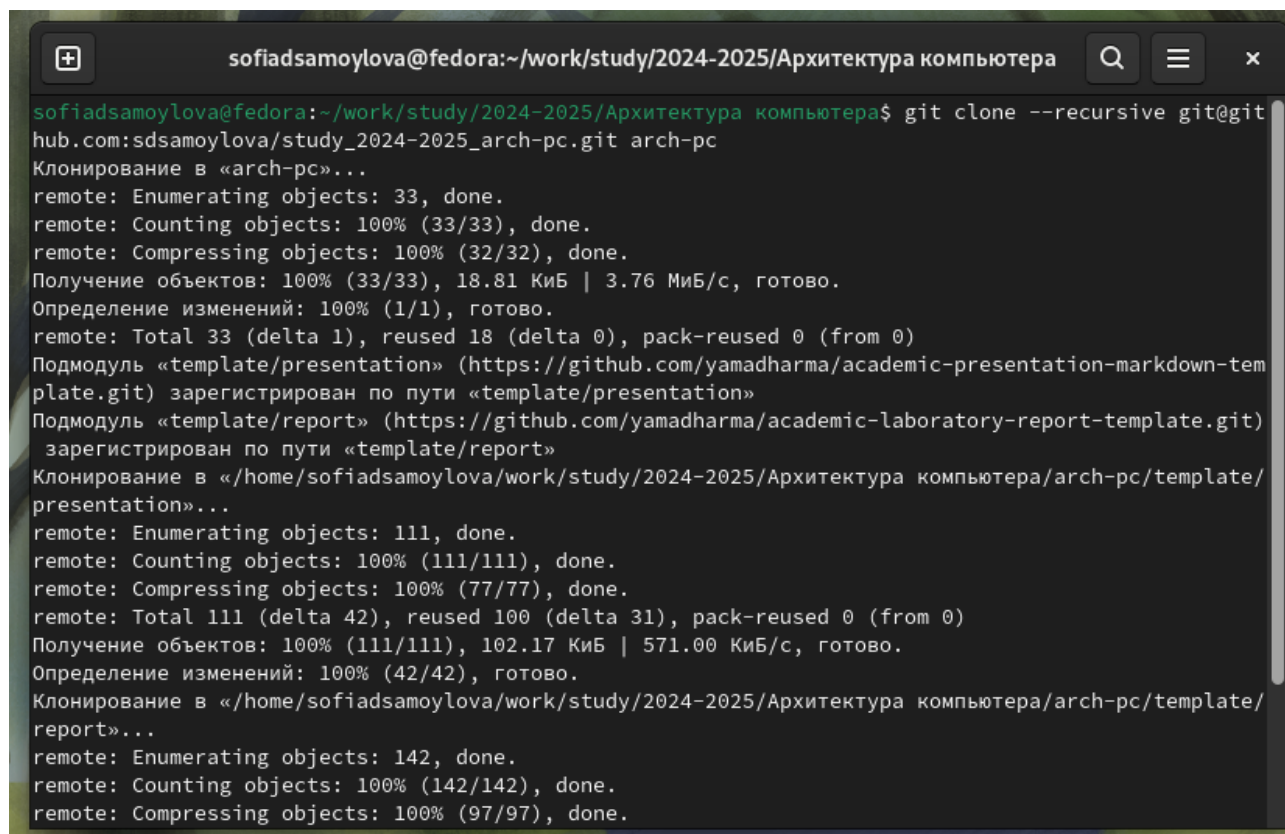
Great repository names are short and memorable. Need inspiration? How about **friendly-spoon** ?

Description (optional)

рис.11

Открываем терминал, переходим в каталог курса и клонируем созданный репозиторий

(рис.12)

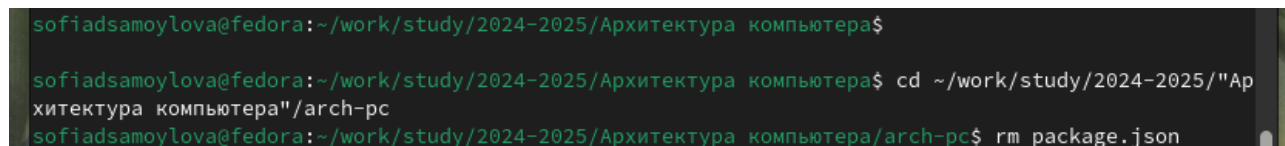


```
sofiadsamoylova@fedora:~/work/study/2024-2025/Архитектура компьютера$ git clone --recursive git@github.com:sdsamoylova/study_2024-2025_arch-pc.git arch-pc
Клонирование в «arch-pc»...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (32/32), done.
Получение объектов: 100% (33/33), 18.81 КиБ | 3.76 МиБ/с, готово.
Определение изменений: 100% (1/1), готово.
remote: Total 33 (delta 1), reused 18 (delta 0), pack-reused 0 (from 0)
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/sofiadsamoylova/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)
Получение объектов: 100% (111/111), 102.17 КиБ | 571.00 КиБ/с, готово.
Определение изменений: 100% (42/42), готово.
Клонирование в «/home/sofiadsamoylova/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.
```

рис.12

#### 4. Настройка каталога курса

Переходим в каталог курса и удаляем лишние файлы (рис.15)



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

рис.13

Создаем необходимые каталоги (рис.14)



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


рис.14

Отправляем файл на сервер (рис.15)


```
sofiadsamoylova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make prepare
sofiadsamoylova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
sofiadsamoylova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -am 'feat(
main): make course structure'
[master 7fdc265] feat(main): make course structure
221 files changed, 53680 insertions(+)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
rename labs/{lab02/report/Л02_Самойлова_отчет => lab01/presentation/.projectile} (100%)
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
```

рис.15

Проверяем правильность создания иерархии рабочего пространства в локальном репозитории и на странице github.

 **sdsamoylova** feat(main): make course structure

7fdc265 · 8 minutes ago





| Name         | Last commit message               | Last commit date |
|--------------|-----------------------------------|------------------|
| ..           |                                   |                  |
| lab01        | feat(main): make course structure | 8 minutes ago    |
| lab02        | feat(main): make course structure | 8 minutes ago    |
| lab03        | feat(main): make course structure | 8 minutes ago    |
| lab04        | feat(main): make course structure | 8 minutes ago    |
| lab05        | feat(main): make course structure | 8 minutes ago    |
| lab06        | feat(main): make course structure | 8 minutes ago    |
| lab07        | feat(main): make course structure | 8 minutes ago    |
| lab08        | feat(main): make course structure | 8 minutes ago    |
| lab09        | feat(main): make course structure | 8 minutes ago    |
| lab10        | feat(main): make course structure | 8 minutes ago    |
| lab11        | feat(main): make course structure | 8 minutes ago    |
| README.md    | feat(main): make course structure | 8 minutes ago    |
| README.ru.md | feat(main): make course structure | 8 minutes ago    |

рис.16

## Задание для самостоятельной работы

Перехожу в директорию labs/lab03/report с помощью утилиты cd. Создаю в каталоге файл для отчета по третьей лабораторной работе с помощью утилиты touch. Я добавила отчет по предыдущей лабораторной работе в соответствующий каталог (рис.17)

 **sdsamoylova** Add files via upload 74ea638 · now 









| Name  | Last commit message               | Last commit date |
|---|-----------------------------------|------------------|
|  ..                        |                                   |                  |
|  bib                       | feat(main): make course structure | 2 hours ago      |
|  image                     | feat(main): make course structure | 2 hours ago      |
|  pandoc                    | feat(main): make course structure | 2 hours ago      |
|  Makefile                  | feat(main): make course structure | 2 hours ago      |
|  report.md               | feat(main): make course structure | 2 hours ago      |
|  Л01_Самойлова_отчет.doc | Add files via upload              | now              |
|  Л01_Самойлова_отчет.pdf | Add files via upload              | now              |

рис.17

## **Выводы**

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



## Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: [http://www.stolyarov.info/books/asm\\_unix](http://www.stolyarov.info/books/asm_unix).
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).