

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: Армихос Гонзалез Карла

Группа: НКАбд-02-24

МОСКВА

2024 г.

Содержание

1. Цель работы.....	4
2. Задание.....	5
3. Теоретическое задание.....	6
4. Порядок выполнения лабораторной работы.....	8
4.1 Базовая настройка git.....	8
4.2 Создание SSH ключа.....	8
4.3 Создание рабочего пространства и репозитория курса на основе шаблона.....	10
4.4 Настройка каталог курса.....	12
5. Задание для самостоятельной работы.....	13
6. Список литературы.....	14

Список таблиц

рис. 4.1.1 Настройка Git.....	8
рис. 4.1.2 Настроим utf-8.....	8
рис. 4.1.3 Конфигурация.....	8
рис. 4.2.1 Создание SSH ключа.....	9
рис. 4.2.2 Добавить созданный ключ.....	9
рис. 4.3.1 Создать каталог.....	10
рис. 4.3.2 Создать новый репозиторий.....	10
рис. 4.3.3 Ключ SHH.....	11
рис. 4.3.4 Клонировать созданный репозиторий.....	12
рис. 4.4.1 Удалить package.json.....	12
рис. 4.4.2 COURSE.....	12
рис. 4.4.3 make.....	12
рис. 4.4.4 Отправьте файлы.....	12
рис. 5.1 labs>lab2>report.....	13
рис. 5.2 Загрузите файлы на github.....	13

1. Цель работы

Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.

2. Задание

- Настроить Git.
- Создайте открытый ключ и загрузите его в Git.
- Создайте репозиторий и используйте его ключ
- Создайте каталог с иерархией, в котором укажите наш учебный год и предмет.
- Загружать наши отчеты в созданный репозиторий

3. Теоретическое задание

Системы контроля версий (VCS) — это инструменты, предназначенные для управления изменениями в файлах и документах. Они позволяют отслеживать изменения в коде, координировать работу нескольких разработчиков, а также обеспечивают возможность восстановления предыдущих версий файлов. VCS особенно полезны в программной разработке, где требуется постоянное обновление и модификация кода.

Основные понятия, связанные с VCS, включают хранилище, `commit`, историю и рабочую копию. *Хранилище* — это место, где сохраняются все версии проекта. **Commit** — это зафиксированное изменение в коде, которое сохраняет состояние файлов в хранилище. **История** — это последовательность всех коммитов, которая позволяет отслеживать изменения и возвращаться к предыдущим версиям. **Рабочая копия** — это локальная версия проекта, с которой разработчик работает, внося изменения перед их фиксацией в хранилище.

Системы контроля версий можно разделить на централизованные и децентрализованные. В централизованных VCS, таких, как SVN (Subversion), все изменения хранятся на центральном сервере, и разработчики работают с единственной версией проекта. В децентрализованных VCS, таких, как Git, каждый разработчик имеет свою полную копию репозитория, что позволяет работать автономно и синхронизировать изменения позже.

При единоличной работе с хранилищем VCS разработчик вносит изменения в рабочую копию, выполняет команду `commit`, чтобы сохранить эти изменения в хранилище, и, если необходимо, просматривает историю изменений для отслеживания прогресса или возврата к предыдущим версиям.

При работе с общим хранилищем VCS порядок действий включает клонирование репозитория, создание веток для новых функций или исправлений, внесение изменений в рабочую копию, выполнение коммитов и отправку изменений в центральное хранилище. Также важна синхронизация изменений с другими разработчиками, чтобы избежать конфликтов.

Основные задачи, решаемые инструментальным средством Git, включают управление версиями, совместная работа над проектами, ведение истории изменений и возможность отката к предыдущим состояниям кода. Git также

поддерживает ветвление и слияние, что позволяет параллельно разрабатывать новые функции.

Некоторые основные команды Git включают:

- `git init`: создание нового репозитория.
- `git clone`: копирование удаленного репозитория на локальную машину.
- `git add`: добавление изменений в индекс перед коммитом.
- `git commit`: фиксация изменений в хранилище.
- `git push`: отправка локальных изменений в удалённый репозиторий.
- `git pull`: получение изменений из удалённого репозитория.
- `git branch`: управление ветками.

Примеры использования Git при работе с локальными и удалёнными репозиториями включают:

- Для локального репозитория: `git init`, затем (`git add .`) и `git commit -m "Первый коммит"` для сохранения изменений.
- Для удалённого репозитория: `git clone <URL>`, чтобы скопировать проект, после чего можно вносить изменения, а затем использовать `git push` для отправки своих коммитов на сервер.

4. Порядок выполнения лабораторной работы

4.1 Базовая настройка git

Мы открываем терминал и вводим следующие команды 1) `git config --global user.name "<Имя, фамилия>"` 2) `git config --global user.email "<work@mail>"`, указав имя и адрес электронной почты владельца репозитория (рис. 4.1.1)

```
gkarmikhos@fedora:~$ git config --global user.name "<ArmijosKarla>"
gkarmikhos@fedora:~$ git config --global user.email "karla.armijos2003@gmail.com"
```

рис. 4.1.1 Настройка Git

Настроим utf-8 в выводе сообщений git (рис. 4.1.2)

```
gkarmikhos@fedora:~$ git config --global user.email "karla.armijos2003@gmail.com"
gkarmikhos@fedora:~$ git config --global core.quotePath false
```

рис. 4.1.2 Настроим utf-8

Зададим имя начальной ветки, Параметр `autocrlf`, Параметр `safecrlf` (рис. 4.1.3)

```
gkarmikhos@fedora:~$ git config --global init.defaultBranch master
gkarmikhos@fedora:~$ git config global core.autocrlf input
fatal: not in a git directory
gkarmikhos@fedora:~$ git config --global core.autocrlf input
gkarmikhos@fedora:~$ git config --global core.safecrlf warn
```

рис. 4.1.3 Конфигурация

4.2 Создание SSH ключа

Создать открытый ключ с помощью команды `ssh-keygen -C "Имя <работа@почта>"` (рис. 4.2.1)


```

gkarmikhos@fedora:~$ ssh-keygen -C "Karla Armijos <1032244775@pfur.ru>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/gkarmikhos/.ssh/id_ed25519):
Created directory '/home/gkarmikhos/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/gkarmikhos/.ssh/id_ed25519
Your public key has been saved in /home/gkarmikhos/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:ym9KeZ/83+369EeLjphNgR0a+sKyAS9d0bLCwaBAQeo Karla Armijos <1032244775@pfur.ru>
The key's randomart image is:
+--[ED25519 256]--+
|.+.|
|o.|
|+ o    o = .|
|.E + . =So o|
|. =.=oo . .|
| + *o.. . .o|
| o.+ooo=... ++|
| ..o.o++ooo+B|
+-----[SHA256]-----+
gkarmikhos@fedora:~$ cat /home/gkarmikhos/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAI02XGg87A4XtzZfZSp0uMk2yuGZepk3UkeApi83Dv7E3 Karla Armijos <
1032244775@pfur.ru>
gkarmikhos@fedora:~$

```

рис. 4.2.1 Создание SSH ключа

Мы копируем созданный нами ключ в Git, сначала нам нужно войти в конфигурацию Git, а в разделе New SSH key мы вставляем наш ключ, добавляем заголовок (рис. 4.2.2)

The screenshot shows the GitHub 'Add new SSH Key' page. On the left, a sidebar lists various settings, with 'SSH and GPG keys' highlighted. The main content area contains a form with the following fields:

- Title:** A text input field containing 'keywww'.
- Key type:** A dropdown menu set to 'Authentication Key'.
- Key:** A large text area containing the public key: `ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAI02XGg87A4XtzZfZSp0uMk2yuGZepk3UkeApi83Dv7E3 Karla Armijos <1032244775@pfur.ru>`.

At the bottom of the form is a green button labeled 'Add SSH key'.

рис. 4.2.2 Добавить созданный ключ

4.3 Создание рабочего пространства и репозитория курса на основе шаблона

Создать каталог с разными иерархиями `mkdir -p`
~/work/study/2024-2025/"Архитектура компьютера" (рис. 4.3.1)

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

рис. 4.3.1 Создать каталог

Создайте репозиторий на основе шаблона через веб-интерфейс github, задайте имя репозитория (рис. 4.3.2)

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 * ArmijosKarla / **Repository name *** study_2024-2025_arh-pc

✔ study_2024-2025_arh-pc is available.

Great repository names are short and memorable. Need inspiration? How about [super-octo-enigma](#) ?

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. 4.3.2 Создать новый репозиторий

Ключ SHN из нашего репозитория (рис. 4.3.3)

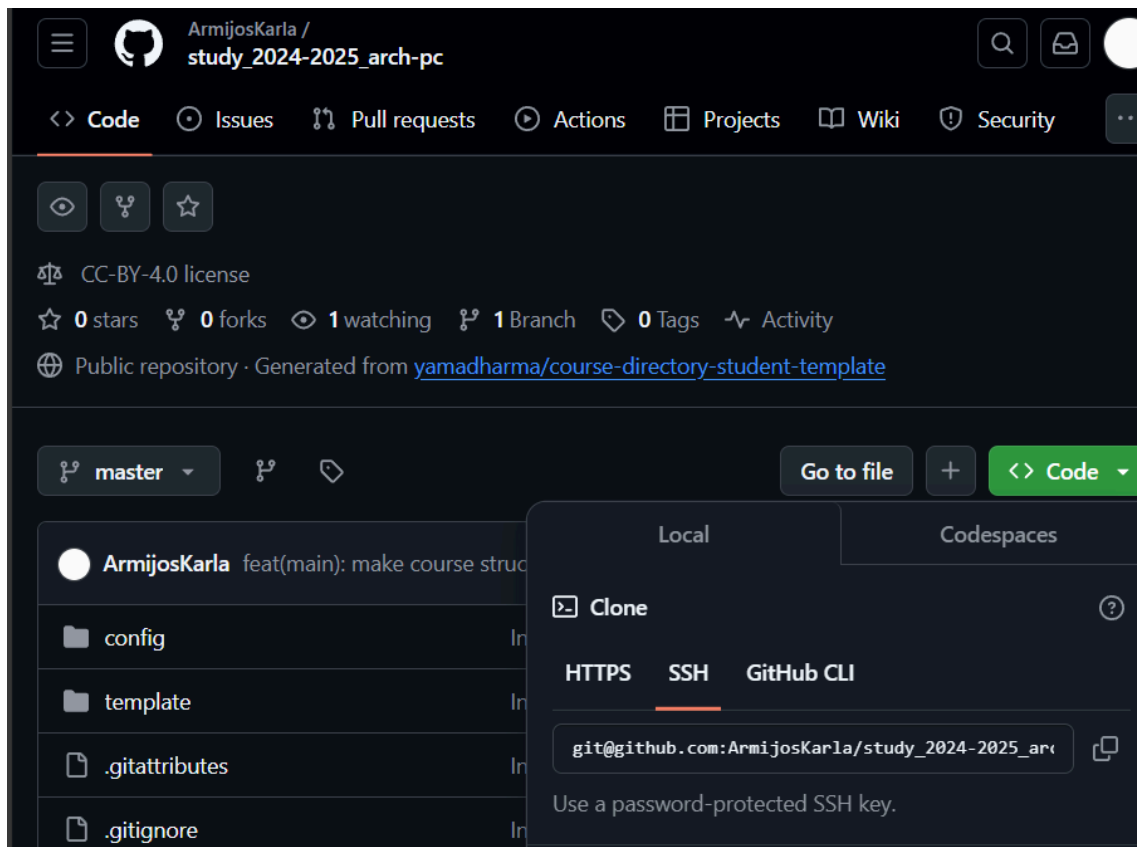


рис. 4.3.3 Ключ SSH

Клонировать папку с помощью сгенерированного ключа нашего репозитория (рис. 4.3.4)

```
gkarmikhos@fedora:~$
gkarmikhos@fedora:~$ mkdir -p ~/work/study/2024-2025/"Архитектура компьютера"
gkarmikhos@fedora:~$ git clone --recursive git@github.com:ArmijosKarla/study_2024-2025_arch-pc.git arch-pc
Cloning into 'arch-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)
Receiving objects: 100% (33/33), 18.82 KiB | 1.57 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markdown-template.git) regi
stered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharma/academic-laboratory-report-template.git) registered for
path 'template/report'
Cloning into '/home/gkarmikhos/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 | 843.00 KiB/s, done.
Resolving deltas: 100% (42/42), done.
Cloning into '/home/gkarmikhos/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 | 1.93 MiB/s, done.
Resolving deltas: 100% (60/60), done.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
gkarmikhos@fedora:~$
```

рис. 4.3.4 Клонировать созданный репозиторий

4.4 Настройка каталог курса

Перейдите в каталог курса и удалите лишние файлы (рис. 4.4.1)

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

рис. 4.4.1 Удалить package.json

Создайте необходимые каталоги command course (рис 4.4.2)

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

рис 4.4.2 COURSE

Команд make (рис. 4.4.3)

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

Targets:
  list           List of courses
  prepare       Generate directories structure
  submodule     Update submules
```

рис. 4.4.3 make

Отправьте файлы на сервер: помощью команды `git add . / f=git commit y git push` (рис. 4.4.4)

```
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
[master fdcd82] feat(main): make course structure
 2 files changed, 1 insertion(+), 14 deletions(-)
 delete mode 100644 package.json
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 293 bytes | 97.00 KiB/s, done.
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:ArmijosKarla/study_2024-2025_arch-pc.git
 89adfc4..fdcd82 master -> master
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

рис. 4.4.4 Отправьте файлы

5. Задание для самостоятельной работы

Создайте отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report) (рис. 5.1).

```
gkarmikhos@fedora:~$ cd ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab02/report
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report$
```

рис. 5.1 labs>lab2>report

Загрузите файлы на github (рис 5.2)

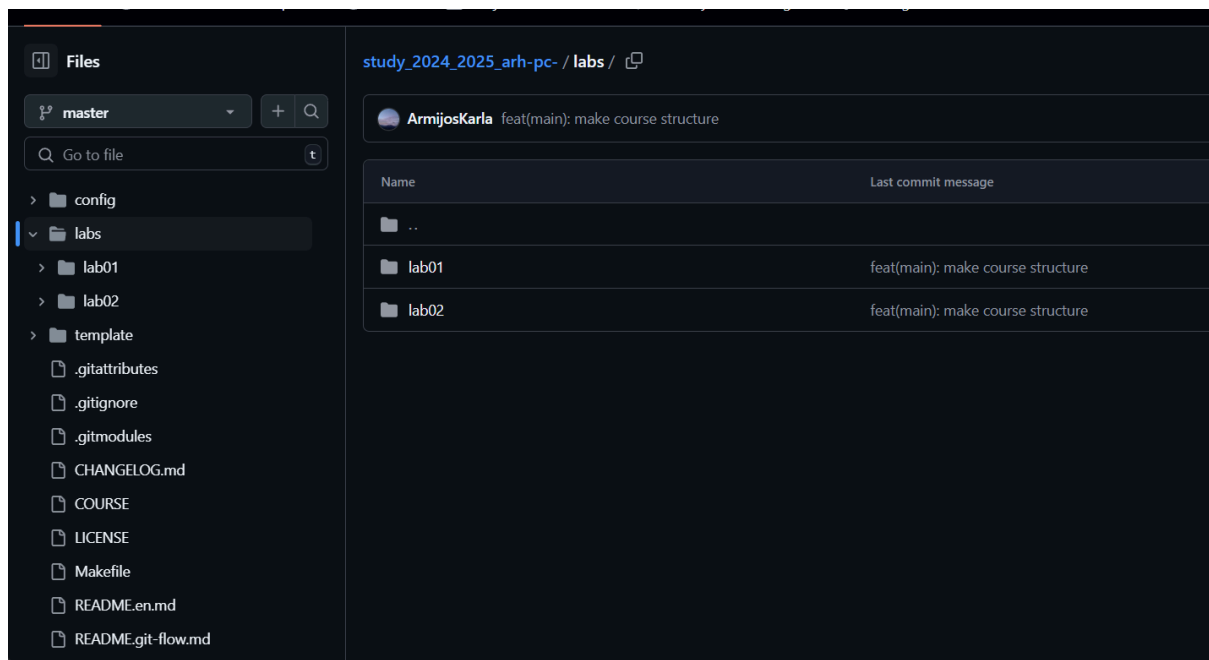


рис. 5.2 Загрузите файлы на github

6. Список литературы

[Архитектура ЭВМ](#)