

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: Бозорова Ануша

Группа: НБИ-01-25

МОСКВА

2025 г.

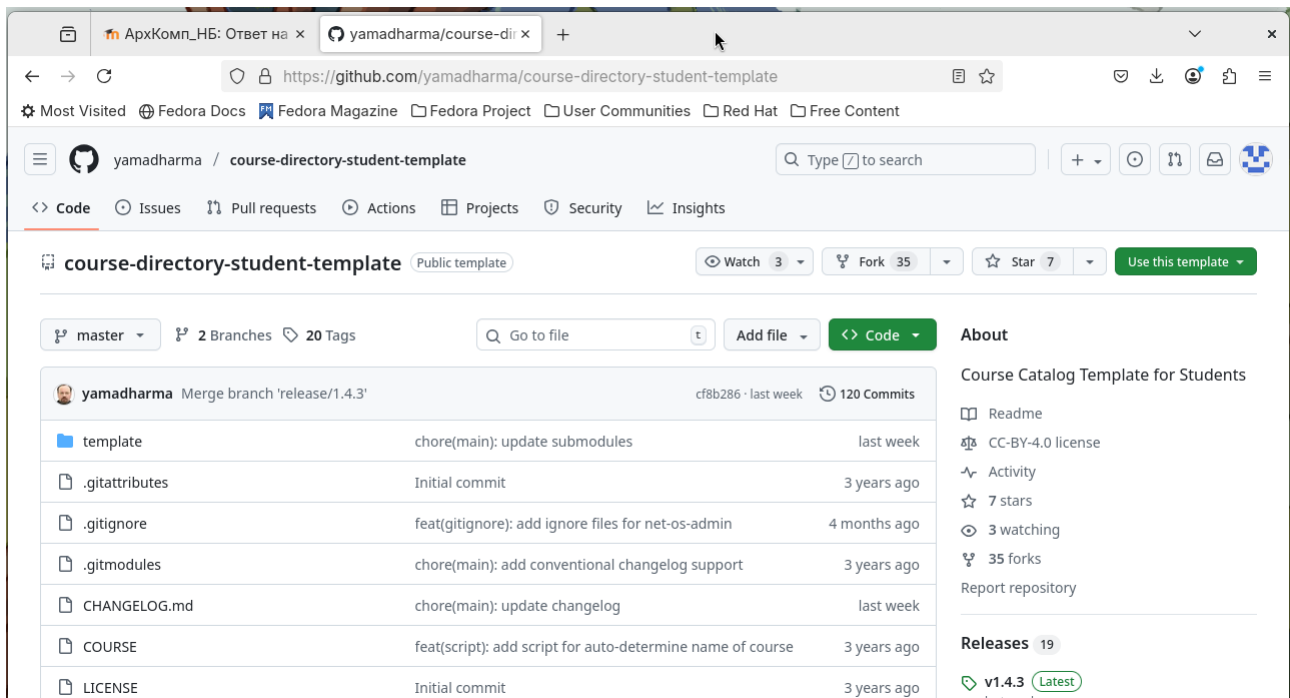
Цель работы

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

Ход работы

Регистрируюсь на гитхабе

Нахожу шаблонный репозиторий и создаю из него свой.




Сначала сделаем предварительную конфигурацию git, создаю пользователя и ставлю параметры.

```
aibozorova@VirtualBox:~$ git config --global user.name "aibozorova"
aibozorova@VirtualBox:~$ git config --global user.email "1132250410@pfur.ru"
aibozorova@VirtualBox:~$ git config --global core.quotepath false
aibozorova@VirtualBox:~$ git config --global init.defaultBranch master
aibozorova@VirtualBox:~$ git config --global core.autocrlf input
aibozorova@VirtualBox:~$ git config --global core.safecrlf warn
aibozorova@VirtualBox:~$
```

Далее создаю ключи для идентификации.

```
aibozorova@VirtualBox:~$  
aibozorova@VirtualBox:~$ ssh-keygen -C "aibozorova 1132250410@pfur.ru"  
Generating public/private ed25519 key pair.  
Enter file in which to save the key (/home/aibozorova/.ssh/id_ed25519):  
Created directory '/home/aibozorova/.ssh'.  
Enter passphrase for "/home/aibozorova/.ssh/id_ed25519" (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/aibozorova/.ssh/id_ed25519  
Your public key has been saved in /home/aibozorova/.ssh/id_ed25519.pub  
The key fingerprint is:  
SHA256:CpScvqyshK1hfrws3i5Ts3cZoZuEmcdYgfVy6vAN23A aibozorova 1132250410@pfur.ru  
The key's randomart image is:  
+--[ED25519 256]--+  
|    o    |  
|    o +.  |  
|   =..o   |  
|    o .+.  |  
|   .X+.ES  |  
|.. 0+BBo   |  
|oo+ B++oo  |  
|+Bo= + o   |  
|++0=o .    |  
+-----[SHA256]-----+
```

И добавляю ключ в профиль на гитхабе

**aibozorova (aibozorova)**
Your personal account

[Go to your personal profile](#)

[Public profile](#)
[Account](#)
[Appearance](#)
[Accessibility](#)
[Notifications](#)


[Access](#)
[Billing and licensing](#)
[Emails](#)
[Password and authentication](#)

SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

**1**

SHA256:CpScvqyshK1hfrws3i5Ts3cZoZuEmcdYgfVy6vAN23A
Added on Oct 9, 2025
Never used — Read/write

[Delete](#)

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

Теперь я создаю рабочий каталог и клонирую туда репозиторий с гитхаба.

```
aibozorova@VirtualBox:~$ mkdir -p ~/work/study/2025-2026/"Архитектура компьютера"
aibozorova@VirtualBox:~$ cd ~/work/study/2025-2026/"Архитектура компьютера"
aibozorova@VirtualBox:~/work/study/2025-2026/Архитектура компьютера$ git clone --recursive git@github.com:aibozorova/study_2025_2026_arch-pc.git
Cloning into 'study_2025_2026_arch-pc'...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 38, done.
remote: Counting objects: 100% (38/38), done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 38 (delta 1), reused 28 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (38/38), 23.53 KiB | 23.53 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharm/academic-presentation-markdown-template.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharm/academic-laboratory-report-template.git) registered for path 'template/report'
```

Создаю курс и структуру папок

```
aibozorova@VirtualBox:~/work/study/2025-2026/Архитектура компьютера/study_2025_2026_arch-pc$ ls
CHANGELOG.md LICENSE package.json README.git-flow.md template
COURSE Makefile README.en.md README.md
aibozorova@VirtualBox:~/work/study/2025-2026/Архитектура компьютера/study_2025_2026_arch-pc$ echo arch-pc > COURSE
aibozorova@VirtualBox:~/work/study/2025-2026/Архитектура компьютера/study_2025_2026_arch-pc$ make prepare
aibozorova@VirtualBox:~/work/study/2025-2026/Архитектура компьютера/study_2025_2026_arch-pc$ ls
COURSE LICENSE package.json presentation README.git-flow.md template
labs Makefile prepare README.en.md README.md
aibozorova@VirtualBox:~/work/study/2025-2026/Архитектура компьютера/study_2025_2026_arch-pc$
```

Отправляю в гитхаб

```
create mode 100644 presentation/report/_resources/cst/gost-r-7-0-3-2008-numeric.cst
create mode 100644 presentation/report/arch-pc--presentation--report.qmd
create mode 100644 presentation/report/bib/cite.bib
create mode 100644 presentation/report/image/solvay.jpg
aibozorova@VirtualBox:~/work/study/2025-2026/Архитектура компьютера/study_2025_2026_arch-pc$ git push
Enumerating objects: 74, done.
Counting objects: 100% (74/74), done.
Delta compression using up to 4 threads
Compressing objects: 100% (58/58), done.
Writing objects: 100% (71/71), 701.26 KiB | 4.61 MiB/s, done.
Total 71 (delta 25), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (25/25), completed with 1 local object.
To github.com:aibozorova/study_2025_2026_arch-pc.git
0f4c456..93ce287 master -> master
aibozorova@VirtualBox:~/work/study/2025-2026/Архитектура компьютера/study_2025_2026_arch-pc$
```

Загружаю отчеты по работам на гитхаб.

```
aibozorova@VirtualBox:~/work/study/2025-2026/Архитектура компьютера/study_2025_2026_arch-pc$  
aibozorova@VirtualBox:~/work/study/2025-2026/Архитектура компьютера/study_2025_2026_arch-pc$ git add .  
aibozorova@VirtualBox:~/work/study/2025-2026/Архитектура компьютера/study_2025_2026_arch-pc$ git commit -am 'add lab'  
[master 2fad08b] add lab  
2 files changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 labs/lab01/report/Л01_Бозорова_отчет.docx  
create mode 100644 labs/lab01/report/Л01_Бозорова_отчет.pdf  
aibozorova@VirtualBox:~/work/study/2025-2026/Архитектура компьютера/study_2025_2026_arch-pc$ git push  
Enumerating objects: 11, done.  
Counting objects: 100% (11/11), done.  
Delta compression using up to 4 threads  
Compressing objects: 100% (7/7), done.  
Writing objects: 100% (7/7), 952.70 KiB | 6.27 MiB/s, done.  
Total 7 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.  
To github.com:aibozorova/study_2025_2026_arch-pc.git  
93ce287..2fad08b master -> master  
aibozorova@VirtualBox:~/work/study/2025-2026/Архитектура компьютера/study_2025_2026_arch-pc$
```

Вывод:

В ходе выполнения работы изучили работу с GitHub.

Вопросы для самопроверки

- 1) Что такое системы контроля версий (VCS) и для решения каких задач они предназначены?

Системы контроля версий (VCS — Version Control Systems) — это инструменты, позволяющие отслеживать изменения в файлах, управлять версиями проектов, восстанавливать предыдущие состояния и координировать совместную работу над проектом. Основные задачи VCS:

- Хранение истории изменений
- Восстановление прошлых версий
- Совместная работа нескольких разработчиков
- Ветвление и слияние версий проекта
- Снижение рисков потери данных

- 2) Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

Хранилище (репозиторий) — место, где хранятся все версии файлов проекта, включая их историю изменений. Это может быть локальный каталог на компьютере или удалённый сервер.

Commit — фиксация изменений в проекте. Каждый commit сохраняет текущие изменения и добавляет их в историю проекта.

История — последовательность commit'ов, представляющая эволюцию проекта. Она позволяет вернуться к любой версии проекта в прошлом.

Рабочая копия — локальная версия файлов проекта, с которой работает разработчик. Она может быть изменена до создания commit'a.

- 3) Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

Централизованные VCS (CVCS) хранят все данные на одном сервере, к которому обращаются все разработчики. Примеры: SVN, CVS. Основной недостаток — зависимость от центрального сервера.

Децентрализованные VCS (DVCS) хранят полную копию репозитория у каждого разработчика, и обмен изменениями происходит напрямую между локальными копиями. Примеры: Git, Mercurial. Основное преимущество — отсутствие зависимости от центрального сервера и возможность работать автономно.

- 4) Опишите действия с VCS при единоличной работе с хранилищем.

Инициализация репозитория (например, `git init`).

Добавление файлов под контроль версий (`git add`).

Фиксация изменений с созданием commit'ов (`git commit`).

Работа с ветками (опционально).

Просмотр истории изменений (`git log`).

Откат к предыдущей версии (при необходимости, например, с помощью `git checkout` или `git revert`).

- 5) Опишите порядок работы с общим хранилищем VCS.

Клонирование общего репозитория (`git clone`).

Создание новой ветки для своей задачи (`git checkout -b new-feature`).

Внесение изменений и создание commit'ов.

Синхронизация с удалённым репозиторием (`git pull` для получения новых изменений).

Отправка изменений в удалённый репозиторий (`git push`).

Создание pull request'a для объединения изменений в основную ветку.

6) Каковы основные задачи, решаемые инструментальным средством git?

Отслеживание изменений файлов

Создание и управление ветками

Совместная работа над проектом через pull requests и push/pull операций

Разрешение конфликтов при слиянии изменений

Поддержка распределённой модели работы с репозиториями

Восстановление предыдущих версий проекта

7) Назовите и дайте краткую характеристику командам git.

git init — создание нового репозитория

git clone — клонирование удалённого репозитория

git add — добавление изменений в область подготовки

git commit — создание commit'a

git status — проверка состояния репозитория

git log — просмотр истории commit'ов

git pull — получение изменений из удалённого репозитория

git push — отправка изменений в удалённый репозиторий

8) Приведите примеры использования при работе с локальным и удалённым репозиториями.

Локальный репозиторий:

Создание репозитория: git init

Добавление файла: git add file.txt

Фиксация изменений: git commit -m "Add file"

Удалённый репозиторий:

Клонирование: git clone https://example.com/repo.git

Получение изменений: git pull origin main

Отправка изменений: git push origin main