

Отчёт по лабораторной работе 2

Система контроля версий Git

Зиборова Вероника Николаевна НММбд-02-24

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	12
4	Вопросы для самопроверки	13

Список иллюстраций

2.1	шаблонный репозиторий	6
2.2	параметры git	7
2.3	ssh ключ	7
2.4	импорт ключа	8
2.5	подготовка каталога	8
2.6	Make создает нужные папки	9
2.7	git push	10
2.8	Репозиторий	10
2.9	git push	11
2.10	Репозиторий	11

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

Регистрируюсь на гитхабе. Нахожу шаблонный репозиторий и создаю из него свой.

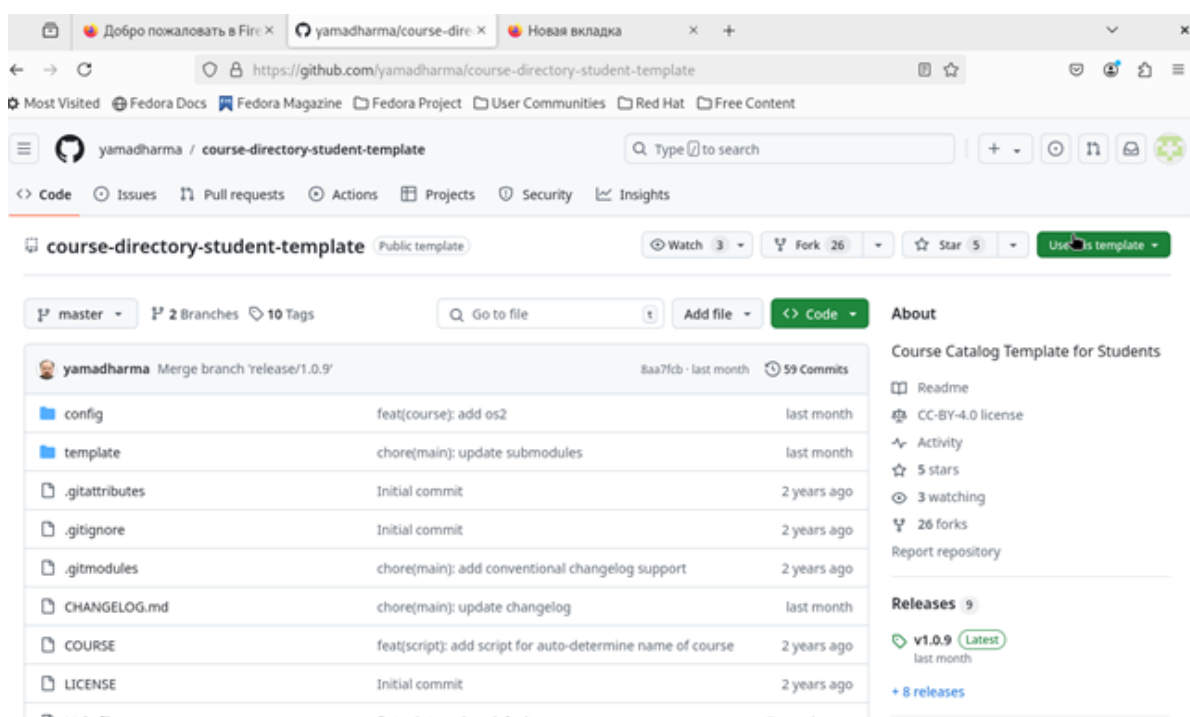
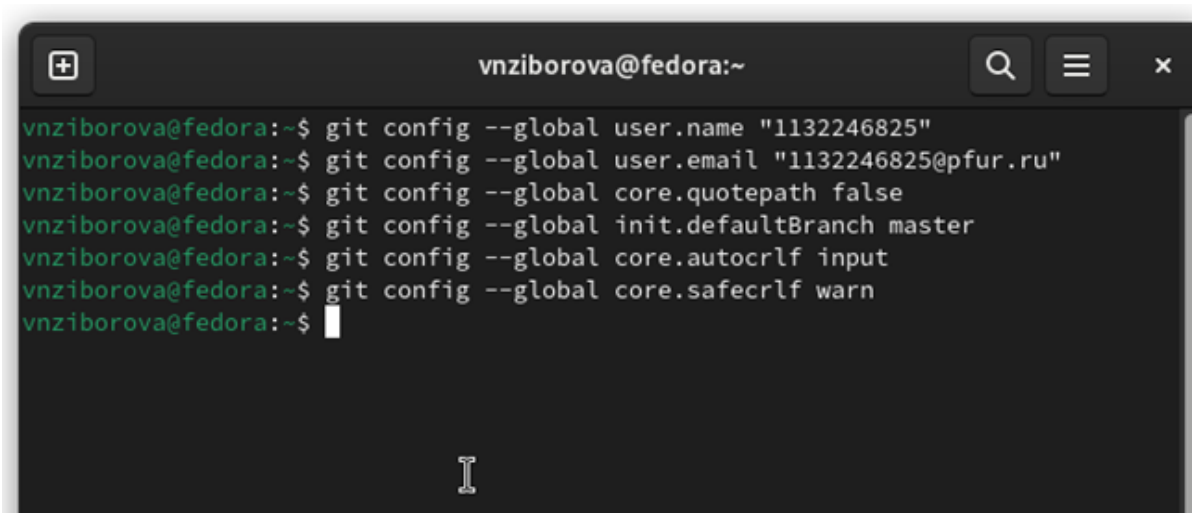


Рис. 2.1: шаблонный репозиторий

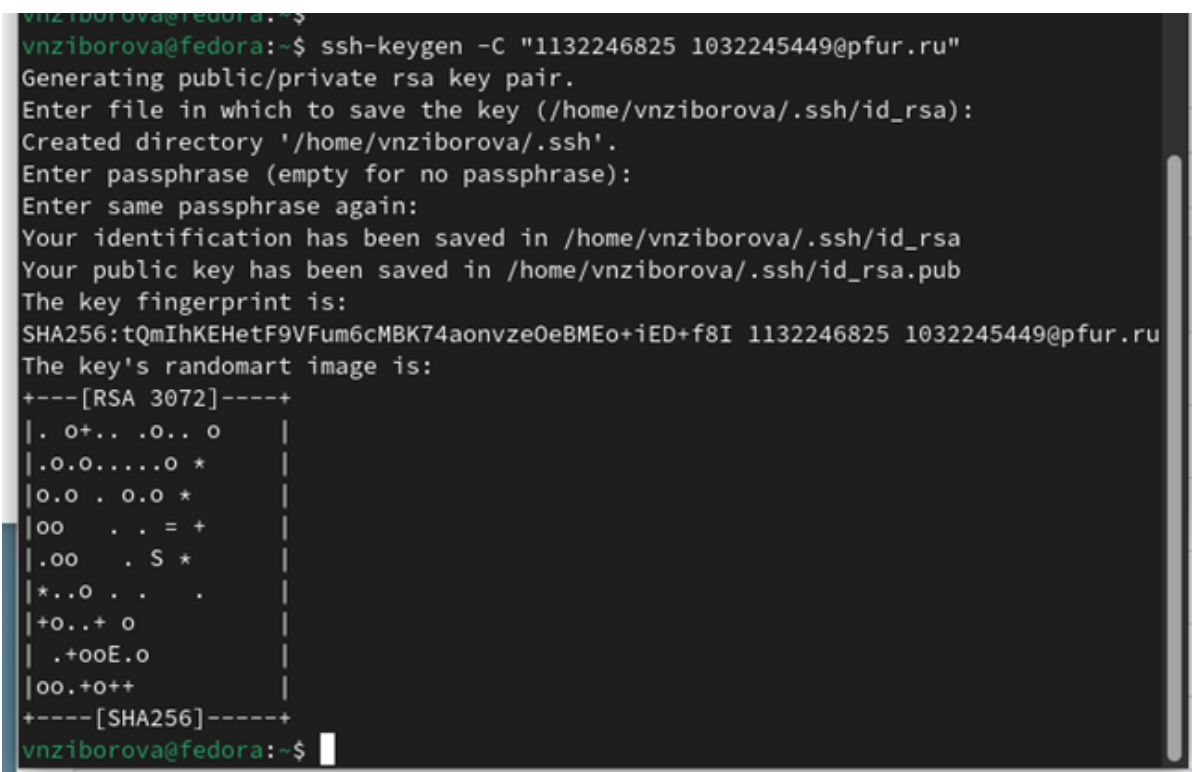
Сначала сделаем предварительную конфигурацию git, создаю пользователя и ставлю параметры.

A terminal window titled 'vnziborova@fedora:~' with search and menu icons in the top right. It contains a series of git configuration commands and their outputs.

```
vnziborova@fedora:~$ git config --global user.name "1132246825"
vnziborova@fedora:~$ git config --global user.email "1132246825@pfur.ru"
vnziborova@fedora:~$ git config --global core.quotepath false
vnziborova@fedora:~$ git config --global init.defaultBranch master
vnziborova@fedora:~$ git config --global core.autocrlf input
vnziborova@fedora:~$ git config --global core.safecrlf warn
vnziborova@fedora:~$
```

Рис. 2.2: параметры git

Далее создаю ключи для идентификации.

A terminal window showing the execution of the ssh-keygen command. It prompts for a file name, passphrase, and displays the key fingerprint and a randomart image.

```
vnziborova@fedora:~$ ssh-keygen -C "1132246825 1032245449@pfur.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vnziborova/.ssh/id_rsa):
Created directory '/home/vnziborova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vnziborova/.ssh/id_rsa
Your public key has been saved in /home/vnziborova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:tQmIhKEHetF9VFum6cMBK74aonvze0eBMEo+iED+f8I 1132246825 1032245449@pfur.ru
The key's randomart image is:
+---[RSA 3072]-----+
|. o+.. .o.. o |
|.o.o.....o * |
|o.o . o.o * |
|oo . . = + |
|.oo . S * |
|*..o . . . |
|+o..+ o |
| .+ooE.o |
|oo.+o++ |
+-----[SHA256]-----+
vnziborova@fedora:~$
```

Рис. 2.3: ssh ключ

И добавляю ключ в профиль на гитхабе

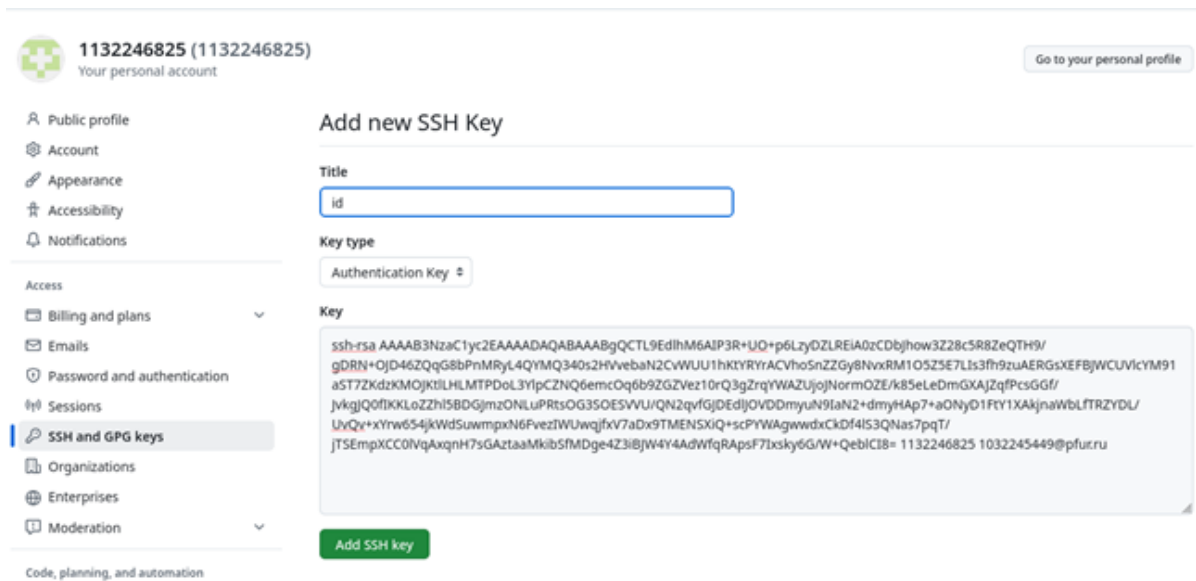


Рис. 2.4: импорт ключа

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

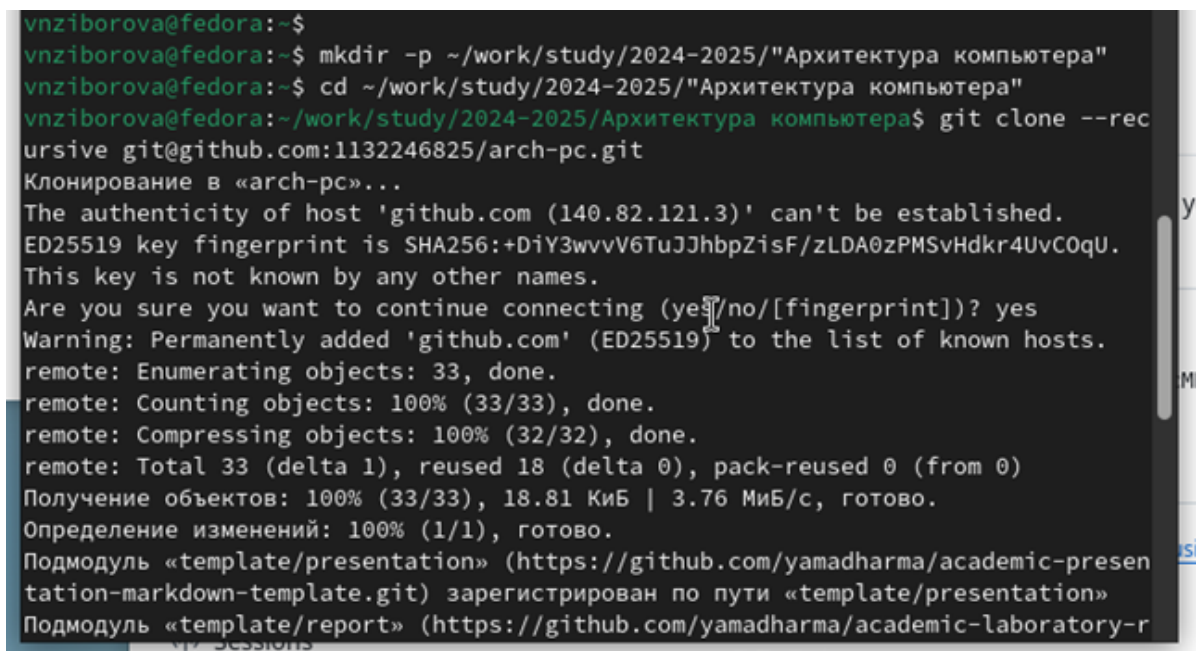


Рис. 2.5: подготовка каталога

Создаю курс и структуру папок


```
vnziborova@fedora:~/work/study/2024-2025/Архитектура компьютера$  
vnziborova@fedora:~/work/study/2024-2025/Архитектура компьютера$ cd ~/work/study/  
/2024-2025/"Архитектура компьютера"/arch-pc  
vnziborova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ rm pack  
age.json  
vnziborova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ echo ar  
ch-pc > COURSE  
vnziborova@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  
  
vnziborova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make pr  
epare  
vnziborova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ ls  
CHANGELOG.md  COURSE  LICENSE  prepare  README.en.md  README.md  
config        labs    Makefile  presentation  README.git-flow.md  template  
vnziborova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 2.6: Make создает нужные папки

Отправляю в гитхаб

```
create mode 100644 presentation/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 presentation/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 presentation/report/pandoc/filters/pandoc_fignos.py
create mode 100755 presentation/report/pandoc/filters/pandoc_secnos.py
create mode 100755 presentation/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/core.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/main.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 presentation/report/report.md
vnziborova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 341.26 КиБ | 3.02 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:1132246825/arch-pc.git
 6f2d7cb..f2498c7 master -> master
vnziborova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 2.7: git push

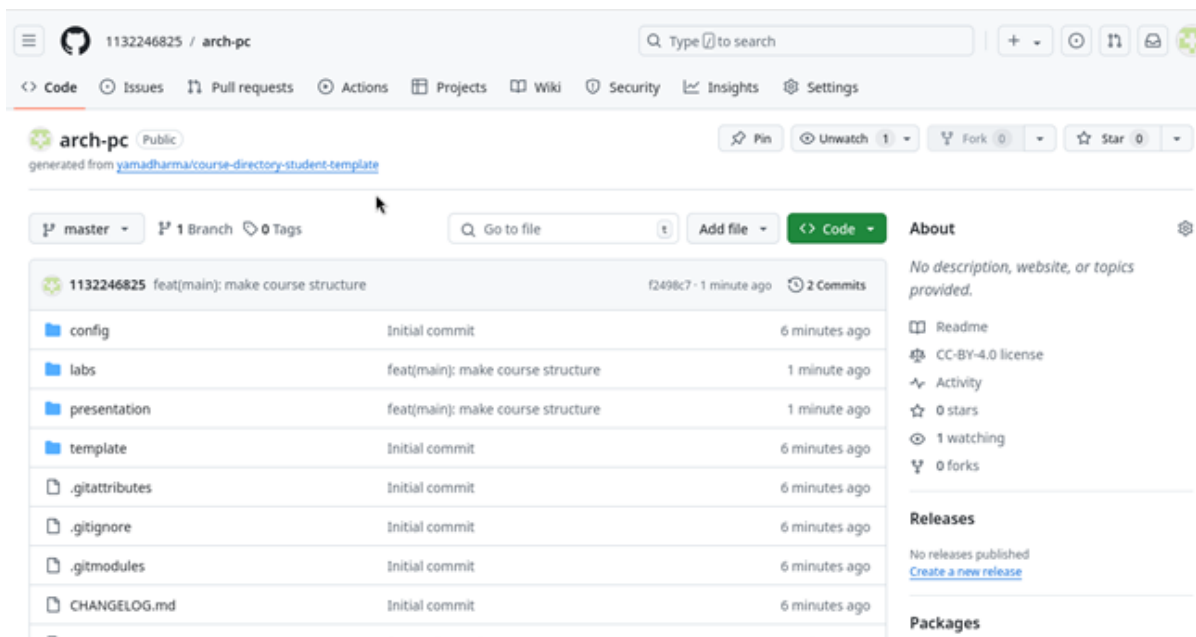


Рис. 2.8: Репозиторий

Загружаю отчеты по работам на гитхаб.

```

vnziborova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
vnziborova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add
.
vnziborova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git com
mit -am lab01
[master a0522b0] lab01
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 labs/lab01/Л01_Зиборова_отчет.pdf..pdf
vnziborova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git pus
h
Перечисление объектов: 8, готово.
Подсчет объектов: 100% (8/8), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 894.90 КиБ | 4.89 МиБ/с, готово.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:1132246825/arch-pc.git
 f2498c7..a0522b0 master -> master
vnziborova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$

```

Рис. 2.9: git push

The screenshot shows the GitHub web interface for a repository named 'arch-pc' under the user '1132246825'. The selected branch is 'lab01'. On the left, a file explorer shows the directory structure: 'config', 'labs', and 'lab01'. Inside 'lab01', there are subdirectories 'presentation' and 'report', and a file 'Л01_Зиборова_отчет.pdf..pdf'. The main area displays a table of files in the 'lab01' branch.

Name	Last commit message	Last commit date
..		
presentation	feat(main): make course structure	2 minutes ago
report	feat(main): make course structure	2 minutes ago
Л01_Зиборова_отчет.pdf..pdf	lab01	now

Рис. 2.10: Репозиторий

3 Выводы

В ходе выполнения работы изучили работу с GitHub.

4 Вопросы для самопроверки

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

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'а.

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`'а 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`