

# **Отчёт по лабораторной работе № 2**

**НММбд-03-22**

Старцева Алина Сергеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Установка программного обеспечения . . . . .	6
3.2	Базовая настройка git . . . . .	7
3.3	Создали ключи ssh . . . . .	7
3.4	Создали ключи pgr . . . . .	8
3.5	Настройка github . . . . .	10
3.6	Добавление PGP ключа в GitHub . . . . .	11
3.7	Настройка автоматических подписей коммитов git . . . . .	13
3.8	Настройка gh . . . . .	13
3.9	Создание репозитория курса на основе шаблона . . . . .	13
3.10	Настройка каталога курса . . . . .	14
<b>4</b>	<b>Выводы</b>	<b>16</b>
<b>5</b>	<b>Ответы на контрольные вопросы</b>	<b>17</b>

# Список иллюстраций

3.1	6
3.2	6
3.3	7
3.4	7
3.5	7
3.6	7
3.7	7
3.8	8
3.9	8
3.10	9
3.11	9
3.12	10
3.13	11
3.14	11
3.15	12
3.16	12
3.17	12
3.18	13
3.19	13
3.20	13
3.21	14
3.22	14
3.23	14
3.24	14
3.25	15
3.26	15
3.27	15
5.1	18
5.2	19
5.3	20

# 1 Цель работы

- Изучить идеологию и применение средств контроля версий.
- Освоить умения по работе с git.

## 2 Задание

- Установить и настроить ПО для работы с git.

## 3 Выполнение лабораторной работы

### 3.1 Установка программного обеспечения

Установили git:(рис. [3.1])

```
[astarceva@fedora ~]$ sudo dnf install git
[sudo] пароль для astarceva:
Последняя проверка окончания срока действия метаданных: 1:11:43 назад, Пн 20 фев 2023 23:20:30.
Пакет git-2.39.1-1.fc36.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения.
Выполнено!
```

Рис. 3.1: .

Установили gh:(рис. [3.2])

```
[astarceva@fedora ~]$ dnf install gh
Ошибка: Эту команду нужно запускать с привилегиями суперпользователя (на большинстве систем - под именем пользователя root).
[astarceva@fedora ~]$ sudo dnf install gh
Последняя проверка окончания срока действия метаданных: 1:12:40 назад, Пн 20 фев 2023 23:20:30.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
=====
Установка:
gh          x86_64       2.23.0-1.fc36  updates      8.2 МБ
=====
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 8.2 М
Объем изменений: 41 М
Продолжить? [д/н]: д
Загрузка пакетов:
```

Рис. 3.2: .

## 3.2 Базовая настройка git

Задали имя и email владельца репозитория: (рис. [3.3])

```
выполнено:  
[astarceva@fedora ~]$ git config --global user.name "asstarceva1"  
[astarceva@fedora ~]$ git config --global user.email "1132226503@pfur.ru"
```

Рис. 3.3: .

Настроили utf-8 в выводе сообщений git:(рис. [3.4])

```
[astarceva@fedora ~]$ git config --global core.quotepath false
```

Рис. 3.4: .

Настроили верификацию и подписание коммитов git. Задали имя начальной ветки (будем называть её master).(рис. [3.5])

```
[astarceva@fedora ~]$ git config --global init.defaultBranch maste
```

Рис. 3.5: .

Параметр autocrlf:(рис. [3.6])

```
[astarceva@fedora ~]$ git config --global core.autocrlf input
```

Рис. 3.6: .

Параметр safecrlf: (рис. [3.7])

```
[astarceva@fedora ~]$ git config --global core.safecrlf warn
```

Рис. 3.7: .

## 3.3 Создали ключи ssh

по алгоритму rsa с ключём размером 4096 бит: (рис. [3.8])

```

[astarceva@fedora ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/astarceva/.ssh/id_rsa):
Created directory '/home/astarceva/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/astarceva/.ssh/id_rsa
Your public key has been saved in /home/astarceva/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:TSgdYxapd9H56jmQ9yVL/d4KT99+UFeiJk75hIT6QZU astarceva@fedora
The key's randomart image is:
+---[RSA 4096]-----+
|
|      *+.o . |
|      +..E o . .|
|      .+o..+ o ..|
|      o.o.* + . o|
|      oS=. * . o.|
|      . + + + o|
|      +.+.=.|
|      ++o.=|
|      .o+B|
+-----[SHA256]-----+

```

Рис. 3.8: .

по алгоритму ed25519: (рис. [3.9])

```

+-----[SHA256]-----+
[astarceva@fedora ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/astarceva/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/astarceva/.ssh/id_ed25519
Your public key has been saved in /home/astarceva/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:HMMmvJ0LNGa+H8/oOxGAKotb/fDM3L11leLzqHkF5aI astarceva@fedora
The key's randomart image is:
+--[ED25519 256]--+
|
|      o. |
|      .+. .|
|      .+.. o |
|      . . .B.. o ..|
|      . + =S+ o..o..|
|      . o o + +E. .o |
|      o B . +.o.o |
|      . * o +*.o+ |
|      o**=. . |
+-----[SHA256]-----+

```

Рис. 3.9: .

## 3.4 Создали ключи рдр

Сгенерировали ключ (рис. [3.10])



Из предложенных опций выбирали: тип RSA and RSA; размер 4096; выбрали срок действия; значение по умолчанию — 0 (срок действия не истекает никогда). GPG запросил личную информацию, которая сохранится в ключе: Имя. Адрес электронной почты. При вводе email убедились, что он соответствует адресу, используемому на GitHub. (рис. [3.11])

```
[astarceva@fedora ~]$ gpg --full-generate-key
gpg (GnuPG) 2.3.7; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: создан каталог '/home/astarceva/.gnupg'
gpg: создан шит с ключами '/home/astarceva/.gnupg/pubring.kbx'
Выберите тип ключа:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (9) ECC (sign and encrypt) *default*
 (10) ECC (только для подписи)
 (14) Existing key from card
Ваш выбор? 1
длина ключей RSA может быть от 1024 до 4096.
Какой размер ключа Вам необходим? (3072) 4096
Запрошенный размер ключа - 4096 бит
Выберите срок действия ключа.
  0 = не ограничен
```

Рис. 3.10: .

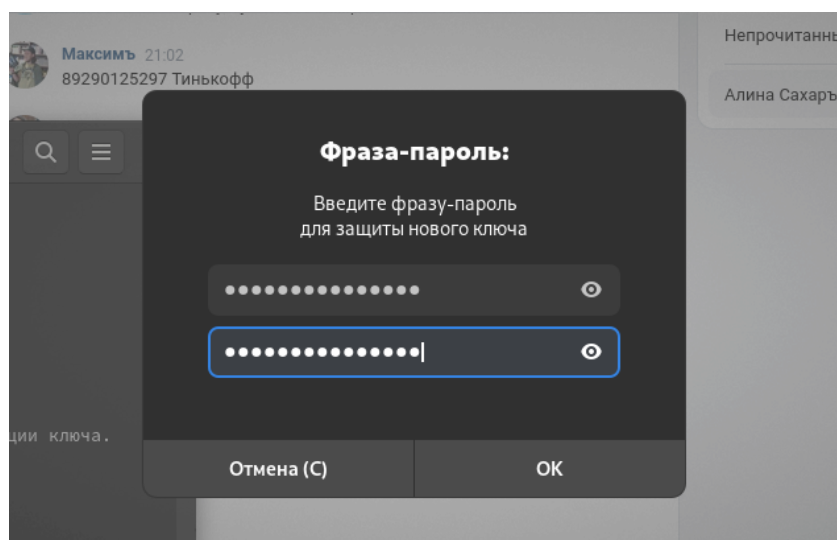


Рис. 3.11: .

## 3.5 Настройка github

Создайте учётную запись на github.com. (рис. [3.12])

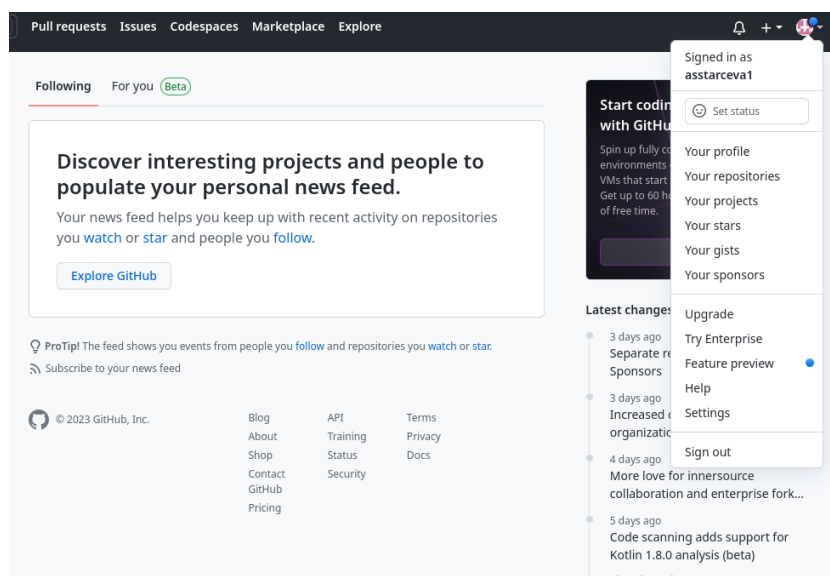


Рис. 3.12: .

Заполните основные данные на github.com. (рис. [3.13])

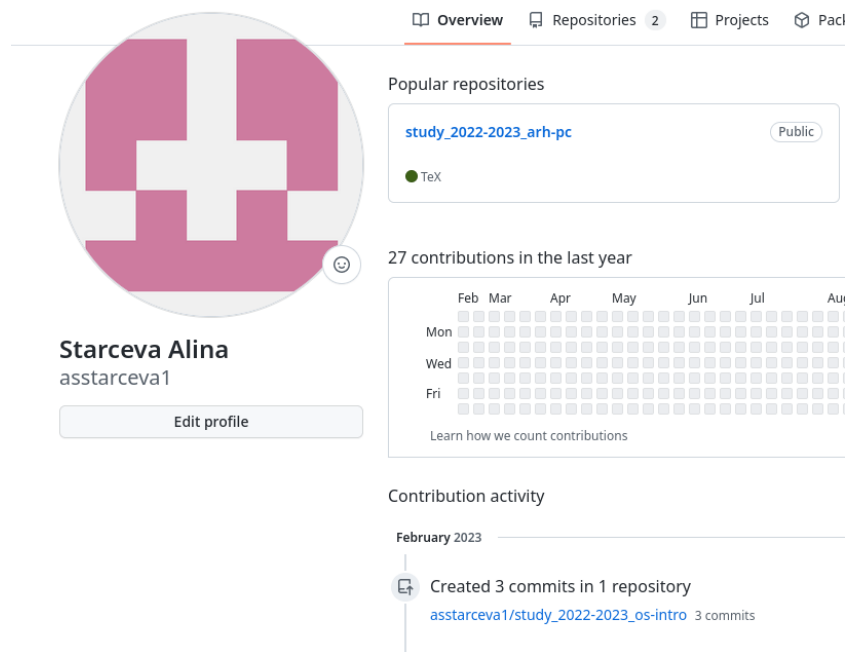


Рис. 3.13: .

## 3.6 Добавление PGP ключа в GitHub

Вывели список ключей и копировали отпечаток приватного ключа: (рис. [3.14])  
Отпечаток ключа — это последовательность байтов, используемая для идентификации более длинного, по сравнению с самим отпечатком ключа.

```
[astarceva@fedora ~]$ gpg --list-secret-keys --keyid-format LONG
gpg: проверка таблицы доверия
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: глубина: 0 достоверных: 1 подписанных: 0 доверие: 0-, 0q, 0n, 0m, 0
, 1u
/home/astarceva/.gnupg/pubring.kbx
-----
sec   rsa4096/53D8B70DF5B860A6 2023-02-20 [SC]
      939546F18E63D7416637BCF253D8B70DF5B860A6
uid           [ абсолютно ] Starceva Alina (rt) <1132226503@pfur.ru>
ssb   rsa4096/BF250B4686AB1838 2023-02-20 [E]
```

Рис. 3.14: .

Скопировали сгенерированный PGP ключ в буфер обмена: (рис. [3.15])

```
[astarceva@fedora ~]$ gpg --armor --export | xclip -sel clip
```

Рис. 3.15: .

Перешли в настройки GitHub, нажали на кнопку New GPG key и вставили полученный ключ в поле ввода. (рис. [3.16], [3.17])

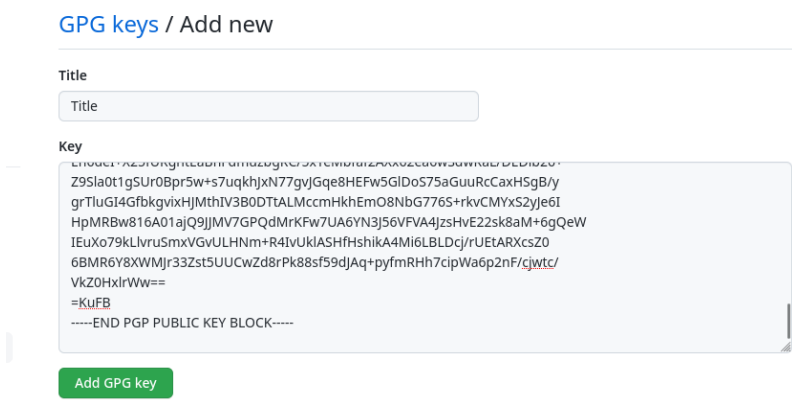


Рис. 3.16: .

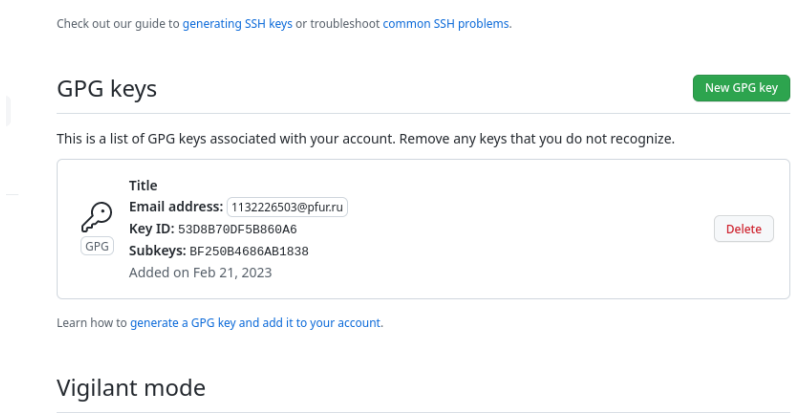


Рис. 3.17: .

## 3.7 Настройка автоматических подписей коммитов git

Используя введённый email, указали Git применять его при подписи коммитов: (рис. [3.18])

```
[astarceva@fedora ~]$ git config --global user.signingkey  
[astarceva@fedora ~]$ git config --global commit.gpgsign true  
[astarceva@fedora ~]$ git config --global gpg.program $(which gpg2)
```

Рис. 3.18: .

## 3.8 Настройка gh

Авторизовались в gh. (рис. [3.19]) Утилита задали несколько наводящих вопросов.

```
[astarceva@fedora ~]$ gh auth login  
? What account do you want to log into? GitHub.com  
? What is your preferred protocol for Git operations? SSH  
? Upload your SSH public key to your GitHub account? /home/astarceva/.ssh/id_rsa  
  .pub  
? Title for your SSH key: Title  
? How would you like to authenticate GitHub CLI? Login with a web browser  
! First copy your one-time code: E72C-8709  
Press Enter to open github.com in your browser...
```

Рис. 3.19: .

## 3.9 Создание репозитория курса на основе шаблона

Создали шаблон рабочего пространства. (рис. [3.20], [3.21], [3.22])

```
[astarceva@fedora ~]$ mkdir -p ~/work/study/2022-2023/"Операционные системы"  
[astarceva@fedora ~]$ cd ~/work/study/2022-2023/"Операционные системы"
```

Рис. 3.20: .

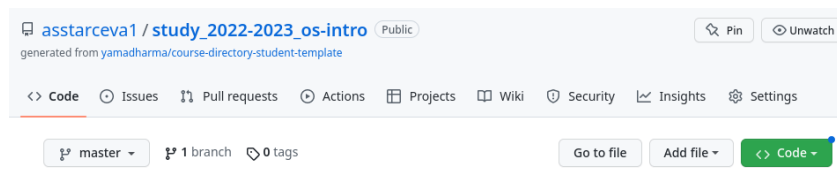


Рис. 3.21: .

```
[astarceva@fedora Операционные системы]$ git clone --recursive git@github.com:asstarceva1/study_2022-2023_os-intro.git
Клонирование в «study_2022-2023_os-intro»...
remote: Enumerating objects: 65, done.
remote: Counting objects: 100% (65/65), done.
remote: Compressing objects: 100% (52/52), done.
remote: Total 65 (delta 5), reused 49 (delta 4), pack-reused 0
Получение объектов: 100% (65/65), 359.28 КиБ | 749.00 КиБ/с, готово.
Определение изменений: 100% (5/5), готово.
Подмодуль «template/presentation» (https://github.com/yamadharm/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharm/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/astarceva/work/study/2022-2023/Операционные системы/study_2022-2023_os-intro/template/presentation»...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
```

Рис. 3.22: .

## 3.10 Настройка каталога курса

Перешли в каталог курса: (рис. [3.23])

```
[astarceva@fedora Операционные системы]$ cd ~/work/study/2022-2023/"Операционные системы"/study_2022-2023_os-intro
```

Рис. 3.23: .

Удалили лишние файлы: (рис. [3.24])

```
[astarceva@fedora study_2022-2023_os-intro]$ rm package.json
```

Рис. 3.24: .

Создали необходимые каталоги: (рис. [3.25])

```
[astarceva@fedora study_2022-2023_os-intro]$ echo os-intro > COURSE
[astarceva@fedora study_2022-2023_os-intro]$ make
```

Рис. 3.25: .

Отправили файлы на сервер: (рис. [3.26], [3.27])

```
[astarceva@fedora study_2022-2023_os-intro]$ git add .
[astarceva@fedora study_2022-2023_os-intro]$ git commit -am 'feat(main): make co
urse structure'
```

Рис. 3.26: .

```
[astarceva@fedora study_2022-2023_os-intro]$ git push
```

Рис. 3.27: .

## 4 Выводы

В ходе выполнения данной лабораторной работы была изучена идеология и применение средств контроля версий и освоены умения по работе с git.



## 5 Ответы на контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются? Система управления версиями (также используется определение «система контроля версий», от англ. Version Control System, VCS или Revision Control System) — программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.
2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия. Хранилище (repository), или репозиторий, — место хранения файлов и их версий, служебной информации. Версия (revision), или ревизия, — состояние всего хранилища или отдельных файлов в момент времени («пункт истории»). Commit («трудовой вклад», не переводится) — процесс создания новой версии; иногда синоним версии. Рабочая копия (working copy) — текущее состояние файлов проекта (любой версии), полученных из хранилища и, возможно, измененных.
3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида. Децентрализованные VCS: У каждого пользователя свой вариант (возможно не один) репозитория. Присутствует возможность добавлять и забирать изменения из любого репозитория (Git, Mercurial, Bazaar).

Централизованные VCS : Одно основное хранилище всего проекта Каждый пользователь копирует себе необходимые ему файлы из этого репозитория, изменяет и, затем, добавляет свои изменения обратно (Subversion, CVS, TFS, VAULT, AccuRev)

4. Опишите действия с VCS при единоличной работе с хранилищем. (рис. [5.1])

## Единоличная работа с VCS

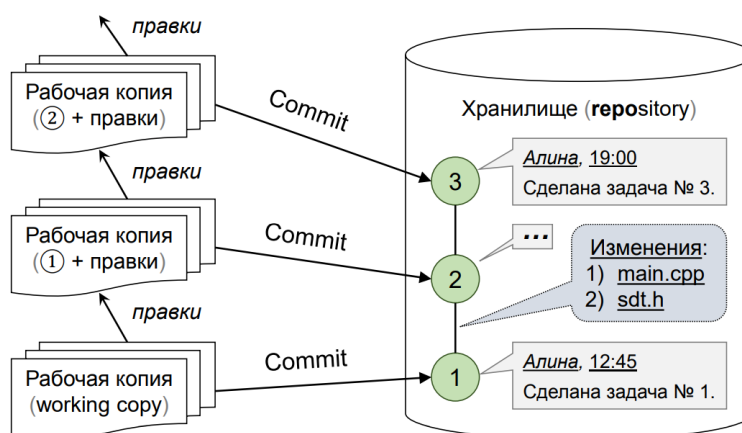


Рис. 5.1: .

5. Опишите порядок работы с общим хранилищем VCS. (рис. [5.2])

## Работа с общим хранилищем

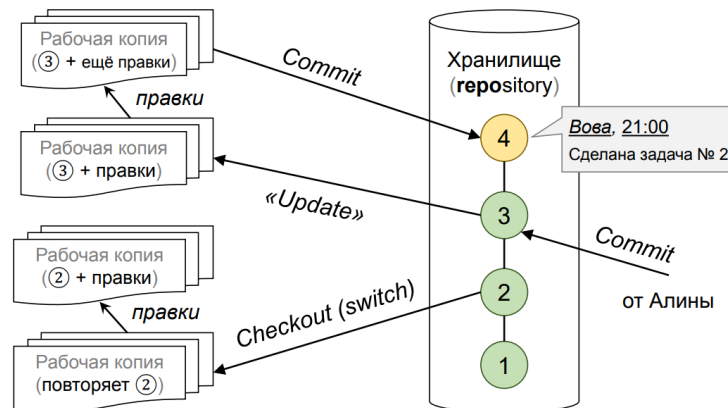


Рис. 5.2: .

6. Каковы основные задачи, решаемые инструментальным средством git? У Git две основных задачи: первая — хранить информацию о всех изменениях в вашем коде, начиная с самой первой строчки, а вторая — обеспечение удобства командной работы над кодом.
7. Назовите и дайте краткую характеристику командам git. git init - создание репозитория git add (имена файлов) - Добавляет файлы в индекс git commit – выполняет коммит проиндексированных файлов в репозиторий git status – показывает какие файлы изменились между текущей стадией и HEAD. Файлы разделяются на 3 категории: новые файлы, измененные файлы, добавленные новые файлы git checkout (sha1 или метка) - получение указанной версии файла git push – отправка изменений в удаленный репозиторий git fetch – получение изменений из удаленного репозитория git clone (remote url) - клонирование удаленного репозитория себе
8. Приведите примеры использования при работе с локальным и удалённым репозиториями. (рис. [5.3])

```
[astarceva@fedora study_2022-2023_os-intro]$ git add .  
[astarceva@fedora study_2022-2023_os-intro]$ git commit -am 'feat(main): make course structure'
```

Рис. 5.3: .

## 9. Что такое и зачем могут быть нужны ветви (branches)?

Ветка (англ. branch) — это последовательность коммитов, в которой ведётся параллельная разработка какого-либо функционала Основная ветка– master Ветки в GIT. Показать все ветки, существующие в репозитории git branch. Создать ветку git branch имя.

Ветки нужны, чтобы несколько программистов могли вести работу над одним и тем же проектом или даже файлом одновременно, при этом не мешая друг другу. Кроме того, ветки используются для тестирования экспериментальных функций: чтобы не повредить основному проекту, создается новая ветка специально для экспериментов.

## 10. Как и зачем можно игнорировать некоторые файлы при commit? Игнорируемые файлы — это, как правило, артефакты сборки и файлы, генерируемые машиной из исходных файлов в вашем репозитории, либо файлы, которые по какой-либо иной причине не должны попадать в коммиты. Вот некоторые распространенные примеры таких файлов:

кэши зависимостей, например содержимое node\_modules или packages; скомпилированный код, например файлы .o, .рус и .class ; каталоги для выходных данных сборки, например bin, out или target; файлы, сгенерированные во время выполнения, например .log, .lock или .tmp; скрытые системные файлы, например .DS\_Store или Thumbs.db; личные файлы конфигурации IDE, например .idea.workspace.xml.