## МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ«РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ»**

Факультет физико-математических и естественных наук Кафедра информационных технологий

## ОТЧЕТ

по лабораторной работе 2

**ТЕМА «**Управление версиями**»** по дисциплине» по дисциплине **«**Операционные системы**»**

**Выполнил/лa:\***\*Студент/ка группы\*\* НПИбд-02-21

**Студенческий билет No** 1032205421

**Студент/кa:** Стелина Петрити

# Цель работы

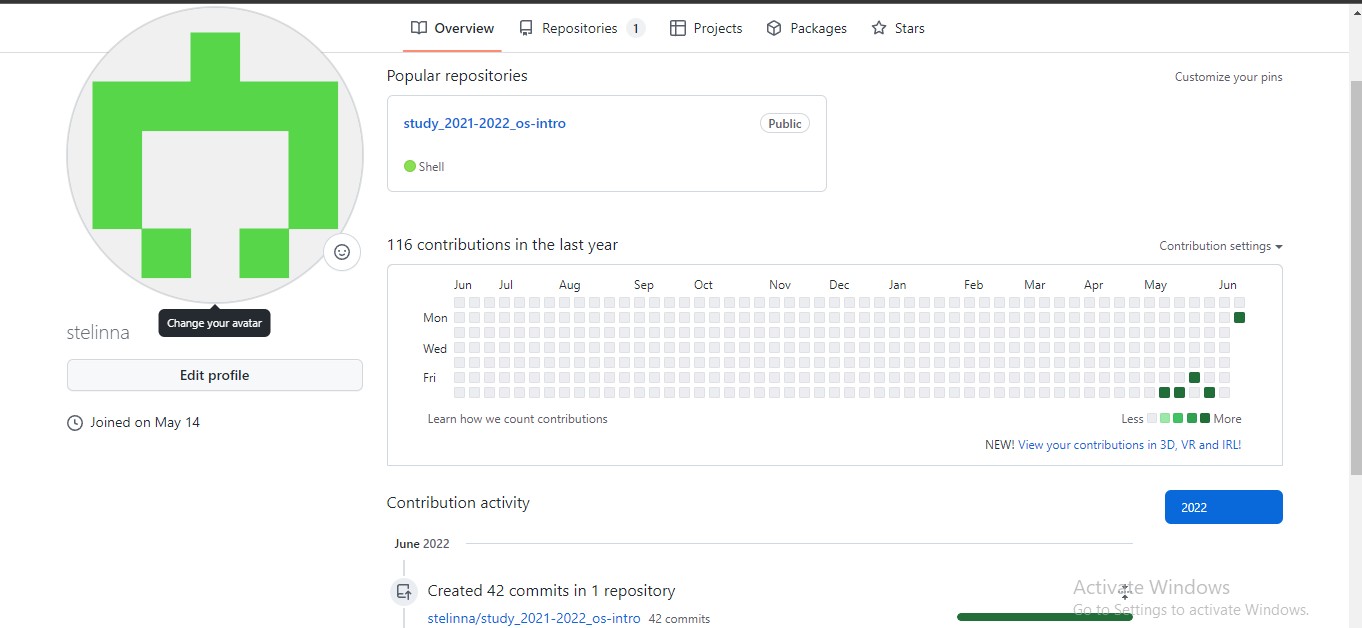
Изучить идеологию и применение средств контроля версий. Освоить умения по работе с git.

# Последовательность выполнения работы

## Задание

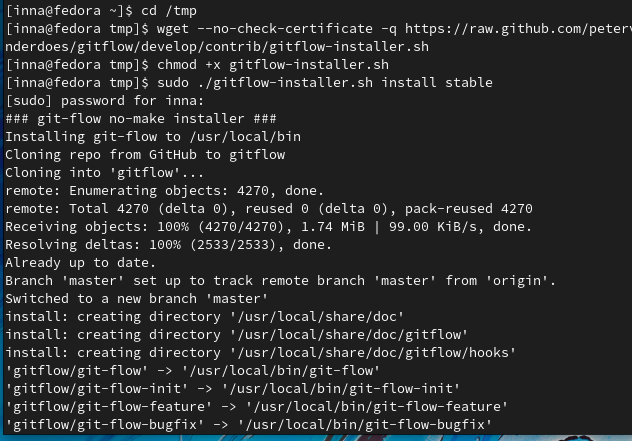
–*Создать базовую конфигурацию для работы с git.*

* *Создать ключ SSH.*
* *Создать ключ PGP.*
* *Настроить подписи git.*
* *Зарегистрироваться на Github.*
* Создать локальный каталог для выполнения заданий по предмету\*
  1. Создадим учётную запись на [https://github.com](https://github.com/).



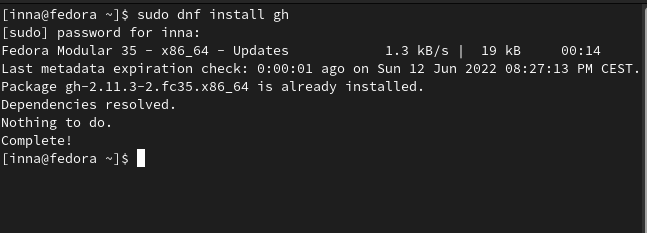
*1.1.Создать учетную запись*

* 1. Установите программное обеспечение git-flow через терминал



*2.2.программное обеспечение git-flow через терминал*

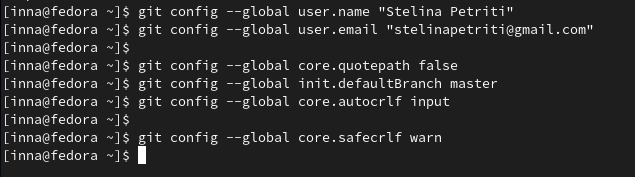
* 1. Установите программное обеспечение gh через терминал



*3.3.программное обеспечение gh через терминал*

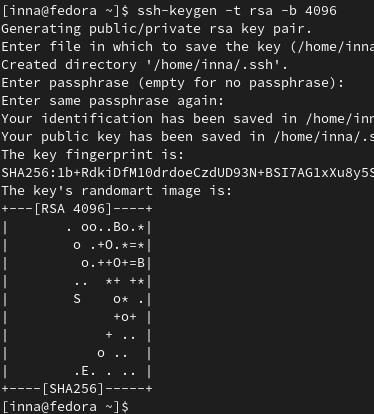
* 1. Давайте сделаем базовую настройку git

1. Укажите имя и адрес электронной почты владельца репозитория.
2. Настройте utf-8 при выводе сообщений git.
3. Настройте проверку и подписание коммитов git.
4. Давайте зададим имя начальной ветви (назовем ее master).
5. Установите параметры autocrlf и safecrlf.

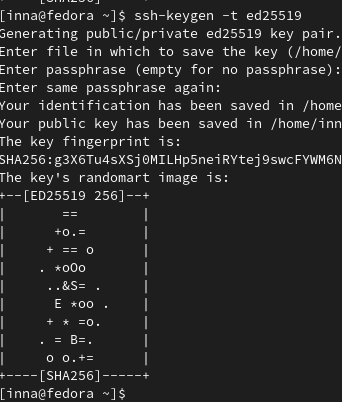


fig(1,2,3,4,5)

* 1. Создайте SSH-ключ
     1. в соответствии с алгоритмом rsa с размером ключа 4096 бит;

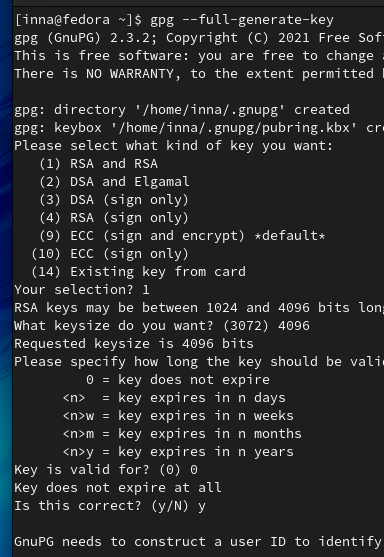


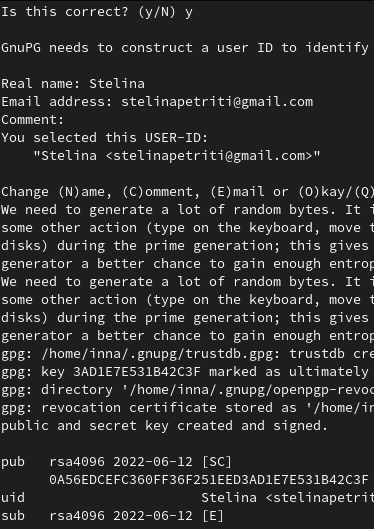
* + - 1. *rsa с размером ключа 4096 бит;*
    1. в соответствии с алгоритмом ed25519.



* + - 1. *с алгоритмом ed25519*

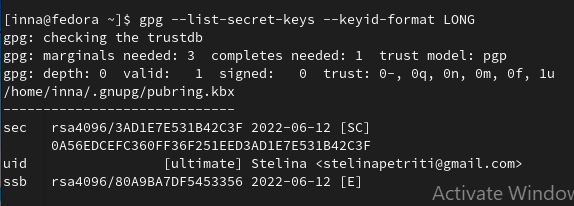
1. Создайте ключ GPG

* Выберите параметры, описанные в лаборатории:
* введите RSA и RSA; размер 4096; — выберите дату истечения срока действия; значение по умолчанию – 0 - GPG запросит личную информацию, которая будет сохранена в ключе. – Имя. – Адрес электронной почты.
  1. *RSA и RSA/ размер 4096/ датa истечения срока действия*



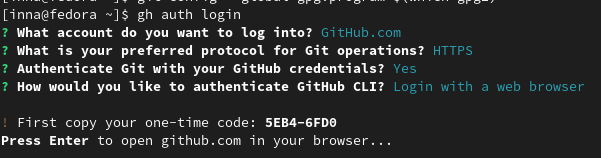
*6.2.GPG*

1. Выведите список ключей и скопируйте отпечаток закрытого ключа



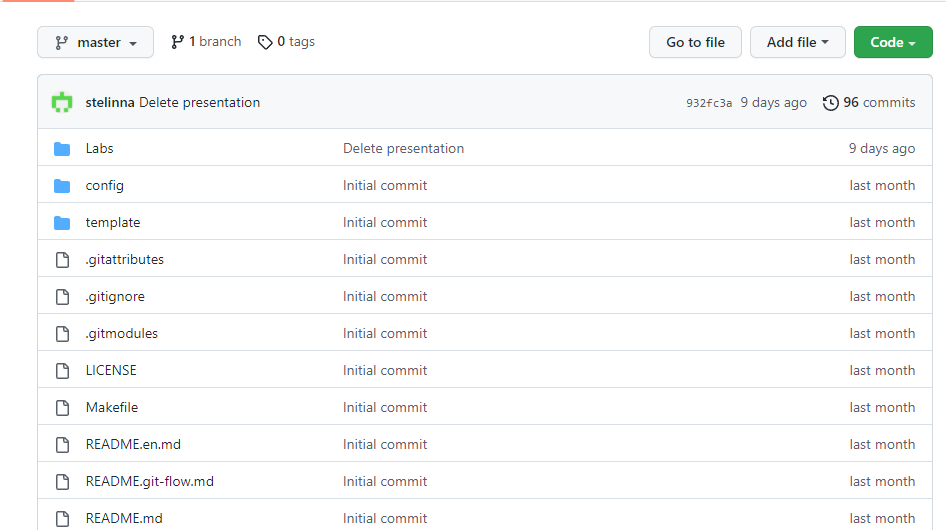
* 1. список ключей/копия отпечаток закрытого ключа

1. Настройте gh. Сначала вам нужно войти в систему, вы можете воспользоваться браузером



*8.1.gh*

1. Создание репозитория и необходимых файлов, которые были необходимы в этой лаборатории



# Выводы

В этой лаборатории мы научились изучать идеологию и применение инструментов контроля версий. Также мы освоили навыки работы с git

# Контрольные вопросы

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

это программные инструменты, помогающие командам разработчиков управлять изменениями в исходном коде с течением времени

## Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

система контроля версий (**VCS**) — это система, регистрирующая изменения в одном или нескольких файлах

Хранилище-система,обеспечивает хранение всех существовавших вариантов файлов

Commit-фиксация изменений. Команде commit можно передать сообщение, описывающее изменения в ревизии.

История-список предыдущих ревизий. Рабочая копия - копия другой ветки.

## Что представляют собой и чем отличаются централизованные и децентрализованные

Существуют СКВ централизованные- имеется один репозиторий, в который собираются изменения со всех рабочих копий разработчиков, и децентрализованные, когда репозиториев много, и они могут обмениваться изменениями между собой. централизованные СКВ - Репозиторий один.

## Опишите действия с VCS при единоличной работе с хранилищем.

Создание хранилища (git init). Push файлов (git push).

Настройка хранилища.

Коммиты нужных файлов (git add , git commit). **5.Опишите порядок работы с общим хранилищем VCS.** Push файлов (git push).

Создание хранилища (git init).

Добавление в коммит различных файлов (git add , git commit), коммит ресурсов.

1. **Каковы основные задачи, решаемые инструментальным средством git?** предоставление возможности отката в случае неработоспособности кода хранение всех версий продукта.

## Назовите и дайте краткую характеристику командам git

git clean- удаления мусора из рабочего каталога. Это результаты сборки проекта или файлы конфликтов слияний.

git rm - используется в Git для удаления файлов из индекса и рабочей копии. git mv — это всего лишь удобный способ переместить файл.

git reset - используется в основном для отмены изменений.

git commit - берёт все данные, добавленные в индекс с помощью git add, и сохраняет их слепок во внутренней базе данных

git add - добавляет содержимое рабочего каталога в индекс (staging area) для последующего коммита.

## Приведите примеры использования при работе с локальным и удалённым репозиториями.

Локальные репозитории - разработчик один. Удаленные - когда работает целая команда.

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

ветви - чтобы разделять код. Если мы делаем новый функционал, то мы создаем новую ветку, а после окончания мы сделали в основную ветку.

## Как и зачем можно игнорировать некоторые файлы при commit?

Игнорируемый файл — файл, явным образом помеченный для Git как файл, который необходимо игнорировать. Игнорируемые файлы отслеживаются в специальном файле

.gitignore , который регистрируется в корневом каталоге репозитория.