Отчёт по лабораторной работе №2

Операционные системы

Серёгина Ирина Андреевна

Содержание

# 1 Цель работы

Цель работы заключается в изучении идеологии и применении средств контроля версий, освоении умения по работе с git.

#Задания 1.Установка ПО 2.Базовая настройка git 3.Создание ssh-ключа 4.Создание pgp-ключа 5.Регистрация на github 6.Добавление gpg-ключа 7.Настройка автоматических подписей коммитов git 8.Настройка gh 9.Создание репозитория курса на основе шаблона 10.Настройка каталога курса

# 2 Выполнение лабораторной работы

## 2.1 Установка ПО

Я захожу в роль супер-пользователя и устанавливаю git, gh (рис. [1](#fig:001)).

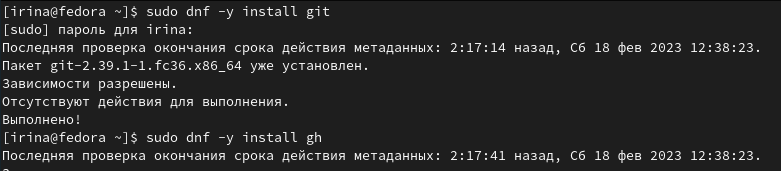


Figure 1: установка git и gh

## 2.2 Базовая настройка git

Задаю имя и электронную почту владельца репозитория (рис. [2](#fig:002)).

Figure 2: установка имени и почты владельца

Figure 2: установка имени и почты владельца

Настраиваю utf-8 в выводе сообщений git (рис. [3](#fig:003)).

Figure 3: Настройка utf-8

Figure 3: Настройка utf-8

Задаю имя начальной ветки master (рис. [4](#fig:004)).

Figure 4: Задаю имя начальной ветки

Figure 4: Задаю имя начальной ветки

Задаю параметры autocrlf, safecrlf (рис. [5](#fig:005)).

Figure 5: Задаю параметры autocrlf, safecrlf

Figure 5: Задаю параметры autocrlf, safecrlf

## 2.3 Создание ssh-ключа

Создаю ключ по алгоритму rsa с ключем размером 4096 бит (рис. [6](#fig:006)).

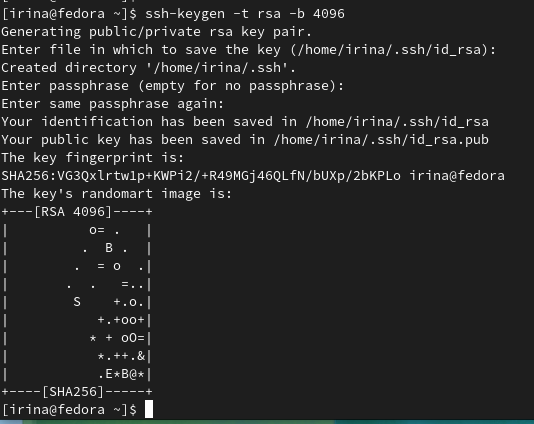


Figure 6: Алгоритм rsa

Создаю ключ по алгоритму ed25519 (рис. [7](#fig:007)).

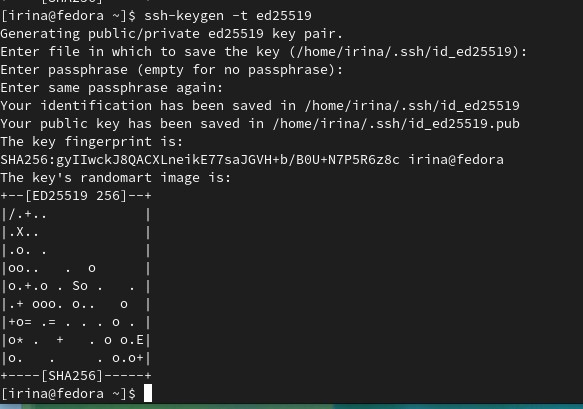


Figure 7: Алгоритм ed25519

## 2.4 Создание pgp-ключа

Генерирую pgp-ключ, из опций выбираю те, которые были представлены в инструкции по лабораторной работе (рис. [8](#fig:008)).

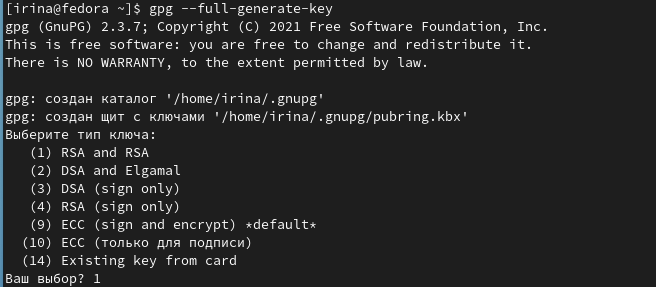


Figure 8: Генерирую pgp-ключ

Защищаю ключ с помощью пароля (рис. [9](#fig:009)).

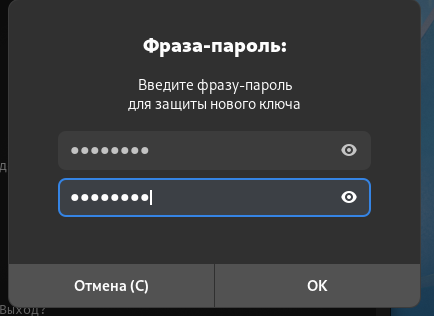


Figure 9: Защищаю pgp-ключ

## 2.5 Регистрация на github

В прошлом семестре я уже создала аккаунт на github, поэтому регистрация не требуется (рис. [10](#fig:010)).

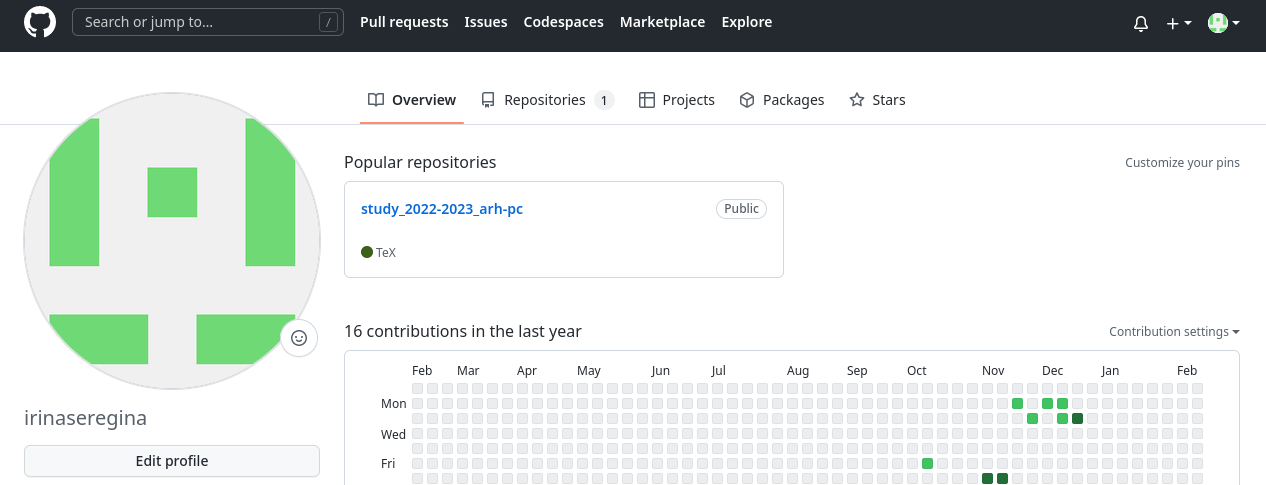


Figure 10: Аккаунт на github

## 2.6 Добавление gpg-ключа

Вывожу список ключей и копирую отпечаток приватного ключа (рис. [11](#fig:011)).

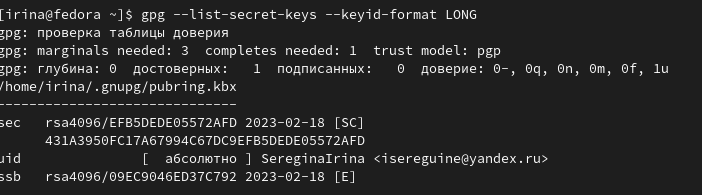


Figure 11: Копирую отпечаток приватного ключа

Копирую мой ключ в буфер обмена с помощью xclip (рис. [12](#fig:012)).

Figure 12: Копирую ключ

Figure 12: Копирую ключ

Перехожу на сайт и добавляю новый ключ, куда вставляю его из буфера обмена (рис. [13](#fig:013)).

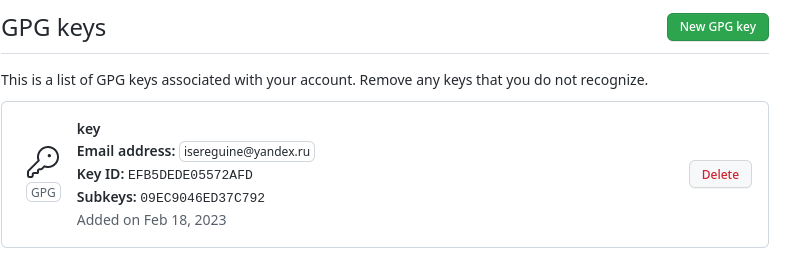


Figure 13: Добавляю ключ

## 2.7 Настройка автоматических подписей коммитов git

Настраиваю коммиты так, чтобы при их подписи использовалась моя почта (рис. [14](#fig:014)).

Figure 14: Настройка коммитов

Figure 14: Настройка коммитов

## 2.8 Настройка gh

Я авторизируюсь, отвечая на насколько вопросов, после этого выбираю продолжить авторизацию в браузере (рис. [15](#fig:015)).

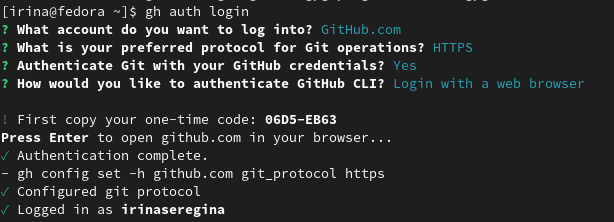


Figure 15: Авторизация

Авторизация прошла успешно (рис. [16](#fig:016)).

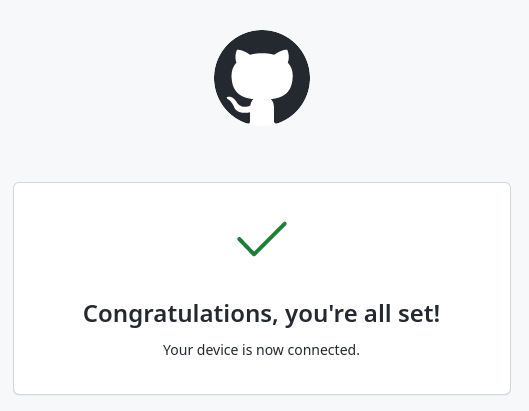


Figure 16: Завершение авторизации

## 2.9 Создание репозитория курса на основе шаблона

Создаю каталог курса с помощью mkdir, после этого перехожу в каталог курса и создаю репозиторий на основе шаблона (рис. [17](#fig:017)).

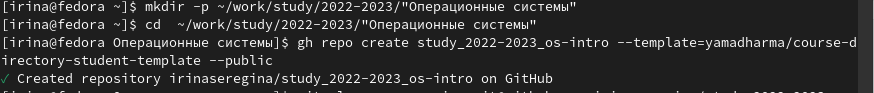


Figure 17: Создание репозитория

Затем клонирую репозиторий себе в директорию (рис. [18](#fig:018)).

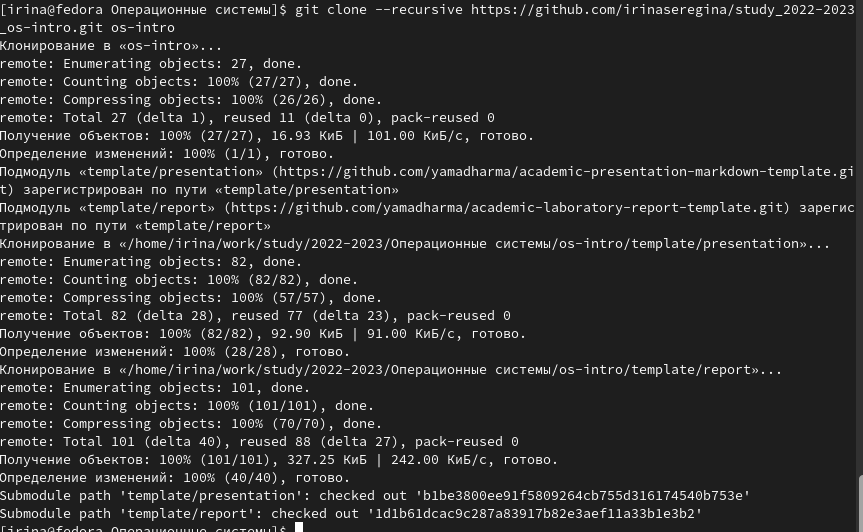


Figure 18: Клонирование репозитория

## 2.10 Настройка каталога курса

Перехожу в каталог курса, смотрю содержимое с помощью ls (рис. [19](#fig:019)).

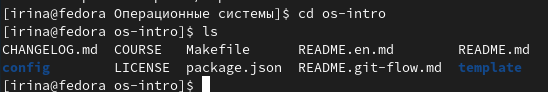


Figure 19: Каталог курса

Удаляю лишние файлы с помощью rm package.json, проверяю, удалились ли они (рис. [20](#fig:020)).

Figure 20: Удаление лишних элементов

Figure 20: Удаление лишних элементов

Создаю необходимые каталоги (рис. [21](#fig:021)).

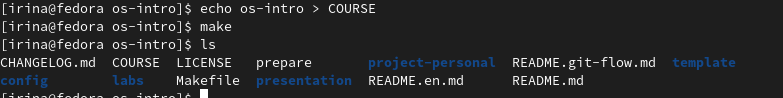


Figure 21: Создание необходимых каталогов

С помощью git add ., git commit добавляю файлы, которые нужно отправить, и добавляю комментарии к ним (рис. [22](#fig:022)).

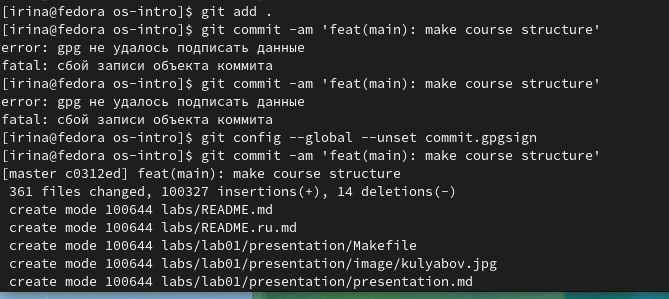


Figure 22: Добавление данных

С помощью git push отправляю данные на сервер (рис. [23](#fig:023)).

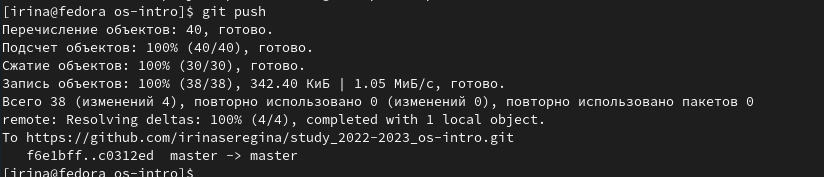


Figure 23: Отправка данных

# 3 Ответы на контрольные вопросы

1. VCS - системы контроля версий, ПО, облегчающее работу с изменяемой информацией, которое позволяет одновременно хранить несколько версий одного элемента, рабоать над информацией совместно и отслеживать чужие поправки.
2. Хранилище - часть, в которой хранятся версии информации и данные о её изменении, коммит - фиксатор изменений, позволяющий их отслеживать, история - то, что позволяет вернуться к более ранней версии данных, хранит историю изменений, рабочая копия - в основном последняя копия документа, основанная на копии из хранилища.
3. Централизованные VCS отличаются от децентрализованных тем, что у них единый общий репозиторий, из которого каждый пользователь может брать информацию, а у децентрализованных каждый имеет свою версию репозитория, имея возможность брать данные из репозиториев других пользователей. К первым относится TFS, ко вторым - Git.
4. Создание и подключение репозитория, а затем постепенная отпрака данных на сервер.
5. Для начала мы берем изначальную версию данных, а после изменения загружаем на сервер новую версию, при этом старая версия не удаляется с сервера.
6. Обеспечение удобной совместной работы и постоянного доступа к данным о состоянии и изменениях в данных.
7. git init - создание основного дерева репозитория git pull - обновление текущего дерева из центрального репозитория git push - отправка изменений локального дерева на сервер git status - просмотр списка измененных файлов в текущей директории git diff - просмотр текущих изменений gt add . - добавление созданных и измененных файлов и каталогов git commit -m “” - сохранение добавленных изменений с комментарием git checkout -b “” - создание ветки с именем git branch -d “” - удвление ветки с именем
8. При использовании команды git pull мы загружаем информацию из удаленного репозитория и обновляем локальный.
9. Ветвление означает создание параллельных веток для более удобной совместной работы над проектом, грубо говоря это история изменений. Изначально существует одна главная ветка. 10.Не все файлы требуют добавления в репозиторий, они могут просто быть бесполезными, поэтому с помощью .gitignore их можно проигнорировать при добавлении информации на сервер.

# 4 Выводы

Я изучила идеологию и применение средств контроля версий, освоила умения по работе с git.

# Список литературы