Лабораторная работа № 2

Первоначальная настройка git

Мальянц Виктория Кареновна

Содержание

# 1 Цель работы

Изучить идеологию и применение средств контроля версий git

# 2 Задание

1. Установка программного обеспечения
2. Базовая настройка git
3. Создание ключа ssh
4. Создание ключа pgp
5. Настройка github
6. Добавление pgp ключа в GitHub
7. Настройка автоматичеких подписей коммитов git
8. Настройка gh
9. Создание репозитория курса на основе шаблона
10. Настройка каталога курса
11. Контрольные вопросы

# 3 Выполнение лабораторной работы

## 3.1 Установка программного обеспечения

Переключаюсь на роль супер-пользователя с помощью sudo -i (рис. 1).

Переключение на роль супер-пользователя с помощью sudo -i

Рис. 1: Переключение на роль супер-пользователя с помощью sudo -i

Устанавливаю git (рис. 2).



Рис. 2: Установка git

Устанавливаю gh (рис. 3).



Рис. 3: Установка gh

## 3.2 Базовая настройка git

Задаю имя и email владельца репозитория (рис. 4).

Имя и email владельца репозитория

Рис. 4: Имя и email владельца репозитория

Настраиваю utf-8 в выводе сообщений git (рис. 5).

Настройка utf-8 в выводе сообщений git

Рис. 5: Настройка utf-8 в выводе сообщений git

Задаю имя начальной ветки (назову ее master) (рис. 6).

Имя начальной ветки

Рис. 6: Имя начальной ветки

Параметр autocrlf (рис. 7).

Параметр autocrlf

Рис. 7: Параметр autocrlf

Параметр safecrlf (рис. 8).

Параметр safecrlf

Рис. 8: Параметр safecrlf

## 3.3 Создание ключа ssh

Создание ключа ssh по алгоритму rsa с ключем размером 4096 бит (рис. 9).

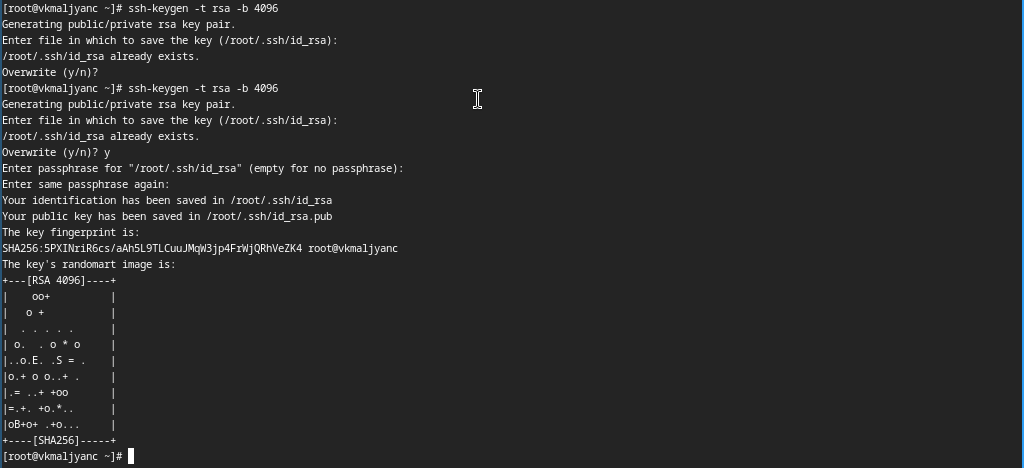


Рис. 9: Создание ключа ssh по алгоритму rsa с ключем размером 4096 бит

Создание ключа ssh по алгоритму ed25519 (рис. 10).

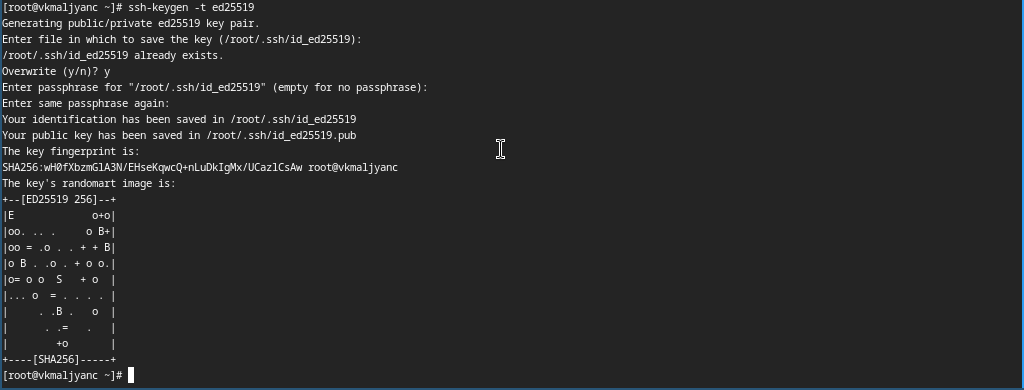


Рис. 10: Создание ключа ssh по алгоритму ed25519

## 3.4 Создание ключа pgp

Генерирую ключ gpg, поэтому ввожу фразу-пароль для его защиты (рис. 11).

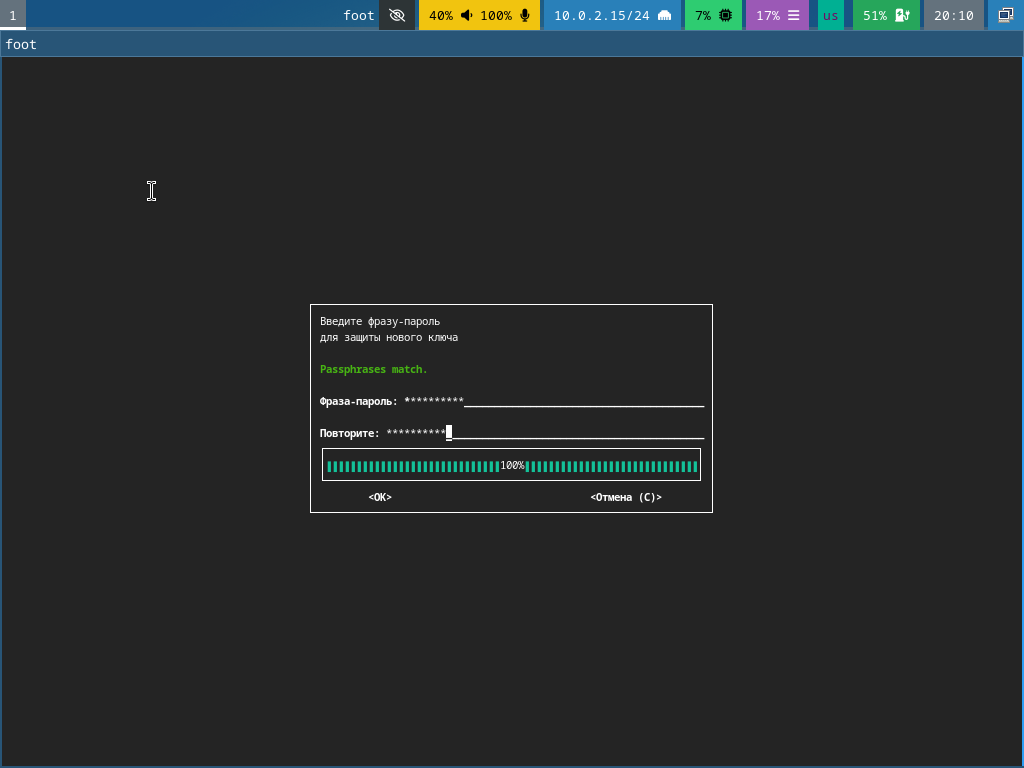


Рис. 11: Фраза-пароль

Генерирую ключ gpg, из предложенных опций выбираю: тип RSA and RSA, размер 4096, 0 (срок действия не истекает никогда) (рис. 12).

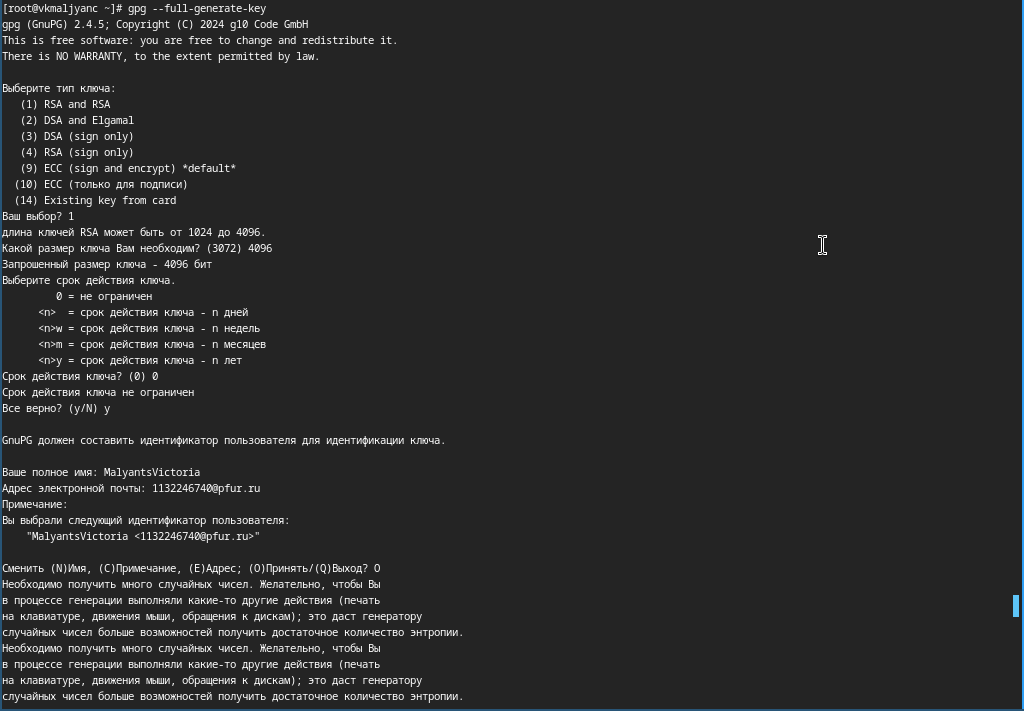


Рис. 12: Генерация ключа gpg

## 3.5 Настройка github

Профиль на GitHub был создан и настроен раннее (рис. 13).

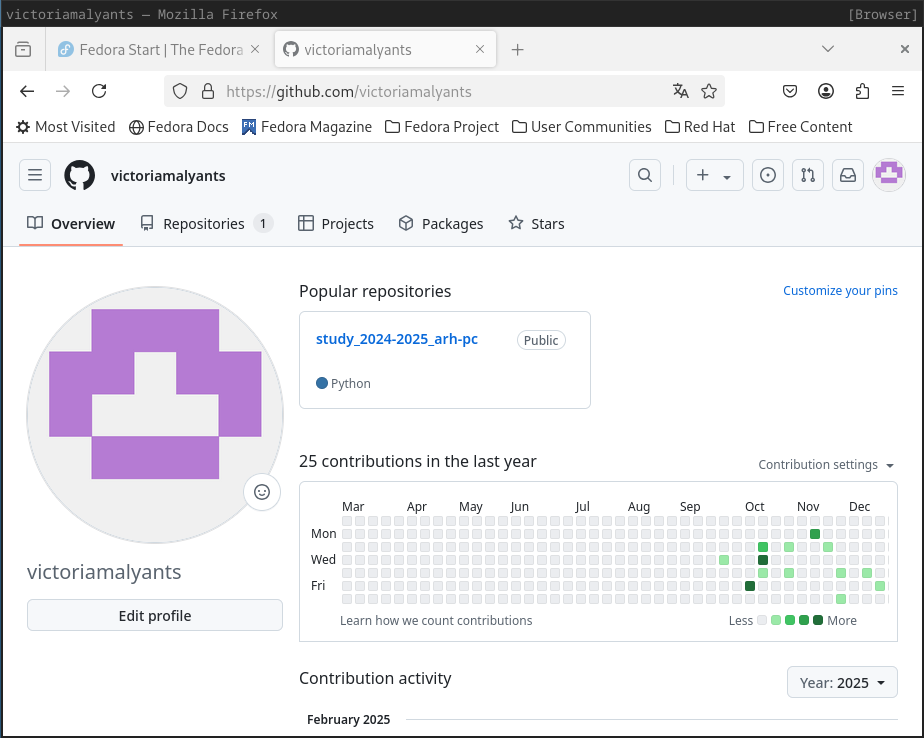


Рис. 13: Профиль GitHub

## 3.6 Добавление pgp ключа в GitHub

Вывожу список ключей и копирую отпечаток приватного ключа (рис. 14).

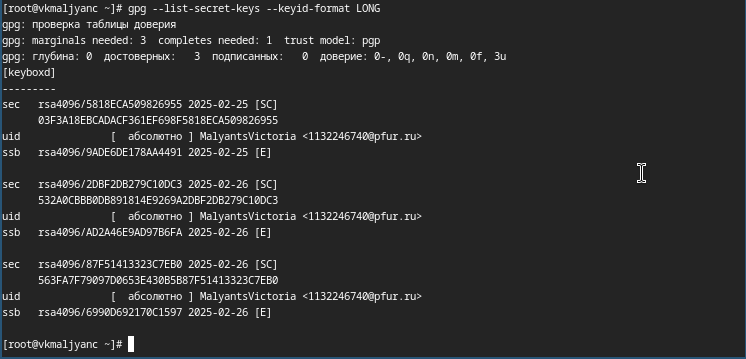


Рис. 14: Список ключей

Копирую сгенерированный ключ gpg (рис. 15).

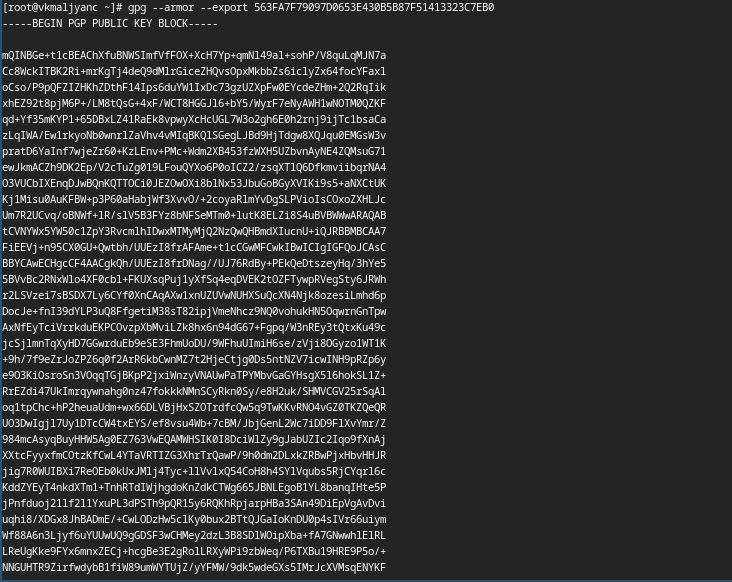


Рис. 15: Копирование сгенерированного ключа gpg

Перехожу в настройки GitHub, нажимаю кнопку New GPG key и вставляю полученный ключ в поле ввода (рис. 16).

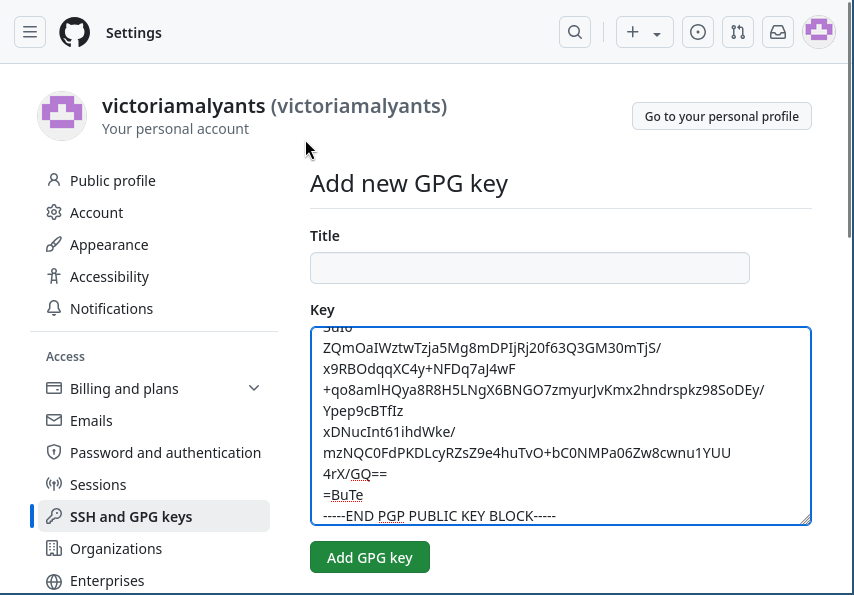


Рис. 16: Создание gpg ключа

Ключ gpg создан (рис. 17).

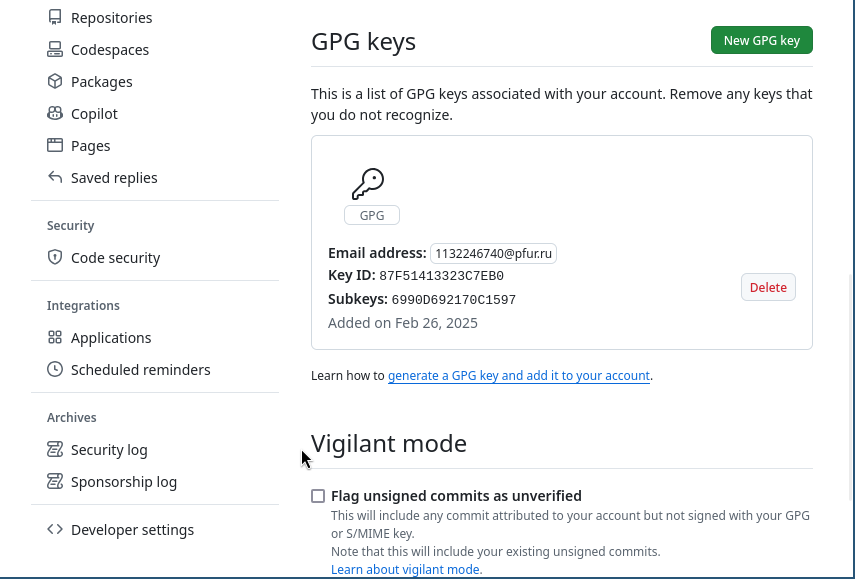


Рис. 17: Ключ gpg создан

## 3.7 Настройка автоматичеких подписей коммитов git

Используя введенный email, указываю Git применять его при подписи коммитов (рис. 18).

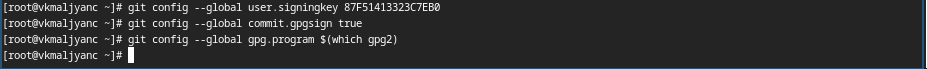


Рис. 18: Настройка для применения подписей коммитов

## 3.8 Настройка gh

Авторизуюсь в gh, отвечаю на наводящие вопросы утилиты (рис. 19).

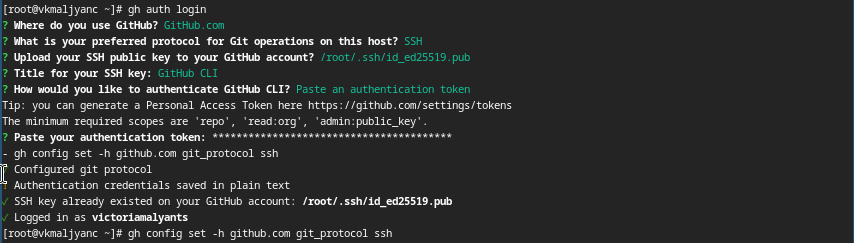


Рис. 19: Авторизация в gh

## 3.9 Создание репозитория курса на основе шаблона

Создаю каталог ~/work/study/2024-2025/“Операционные системы” и перемещаюсь в него (рис. 20).

Создание каталога и перемещение в него

Рис. 20: Создание каталога и перемещение в него

Создаю репозиторий на основе шаблона (рис. 21).

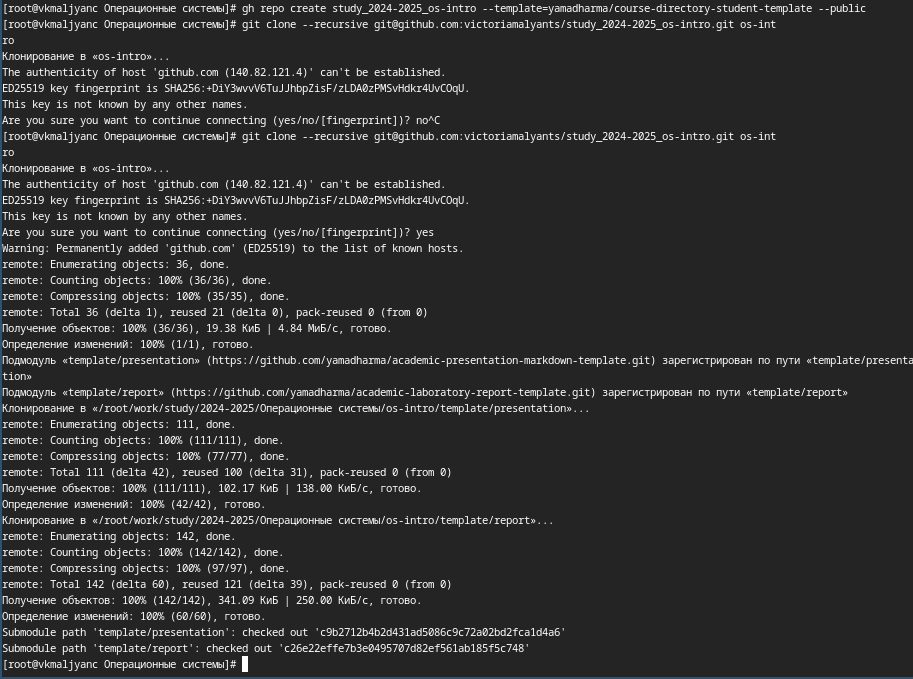


Рис. 21: Создание репозитория на основе шаблона

## 3.10 Настройка каталога курса

Перехожу в каталог курса ~/work/study/2024-2025/“Операционные системы”/os-intro (рис. 22).

Переход в каталог курса

Рис. 22: Переход в каталог курса

Удаляю лишние файлы (рис. 23).

Удаление лишних файлов

Рис. 23: Удаление лишних файлов

Создаю необходимые каталоги (рис. 24).

Создание необходимых каталогов

Рис. 24: Создание необходимых каталогов

Ввожу фразу-пароль для разблокировки ключа gpg (рис. 25).

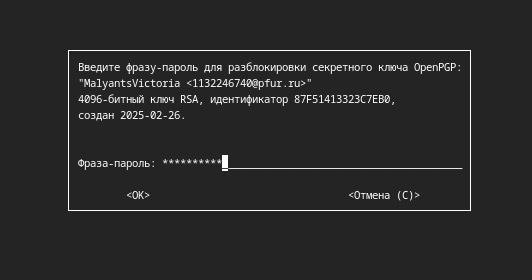


Рис. 25: Фраза-пароль

Создаю каталоги и отправляю файлы на сервер (рис. 26).

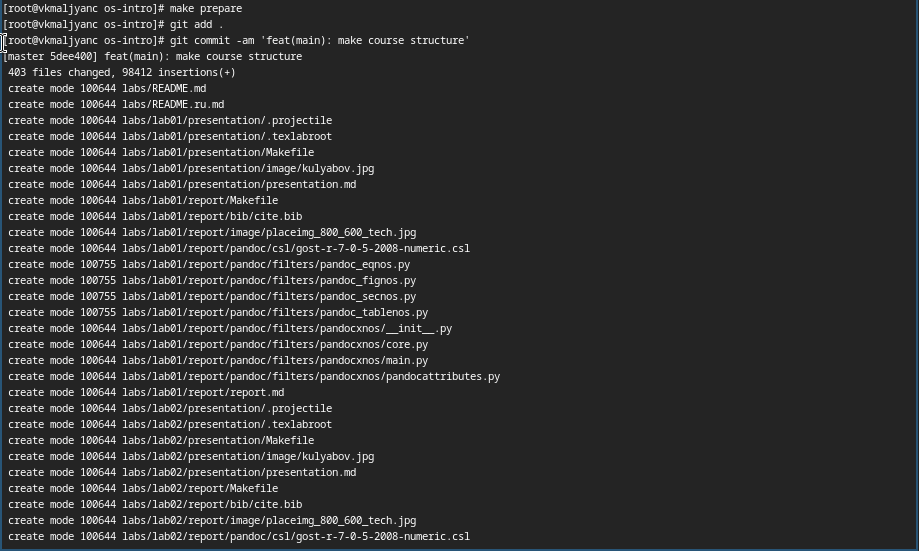


Рис. 26: Создение каталогов и отправление файлов на сервер

Отправляю файлы на сервер (рис. 27) [1].

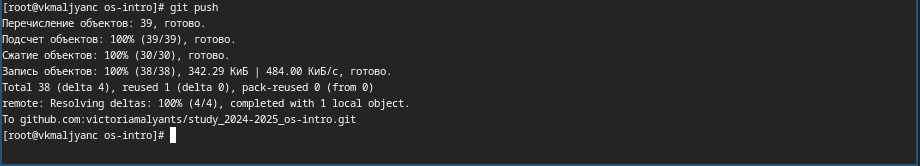


Рис. 27: Отправление файлов на сервер

Каталоги и файлы создались успешно (рис. 28).

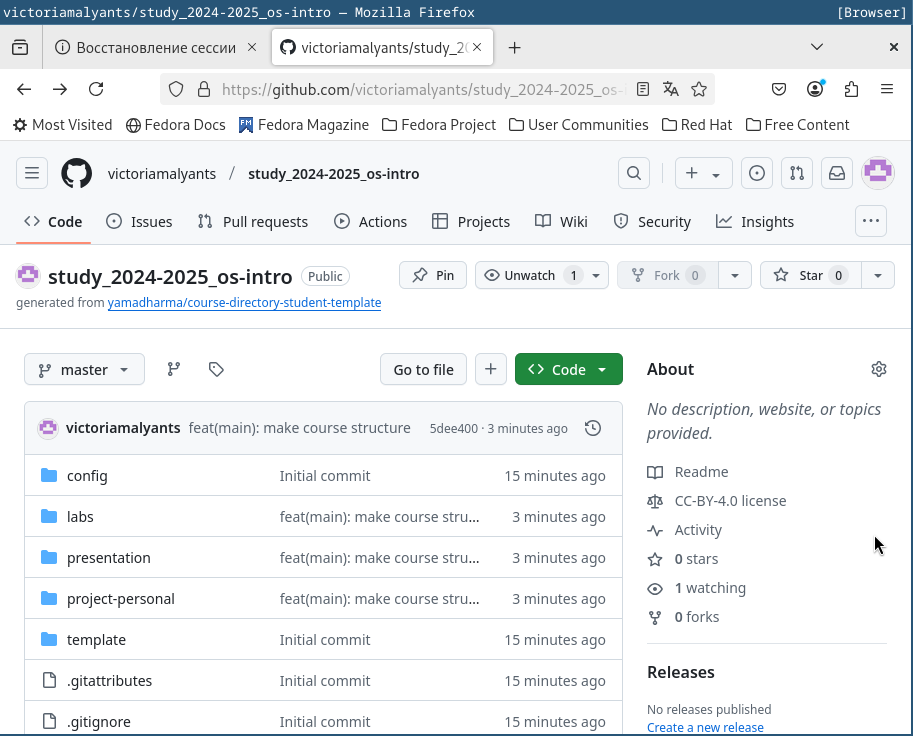


Рис. 28: Каталоги в GitHub

# 4 Выводы

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

# 5 Контрольные вопросы

1. Системы контроля версий (VCS) предназначаются для совместной работы над проектами, а также для отслеживания изменений в файлах.
2. Хранилище - место, где хранятся все версии файлов проекта. Commit - для сохранения изменений в хранилище. История - поседовательность всех коммитов. Рабочая копия - локальная версия файлов.
3. Централизованные VCS - все изменения хранятся на центральном сервере, например: CVS, Subversion. Децентрализованные VCS - каждый разработчик проекта имеет полную копию хранилища, включая всю историю изменений, например: Git, Mercurial.
4. Создание локальнго хранилища, внесение изменений в файлы, использование commit, просмотр изменений и возможность восстановления предыдущих версий
5. Клонирования репозитория, создание новой ветки, внесение изменений в commit, обновление локальной версии, слияние с основной веткой, отправка в хранилище.
6. Отслеживание изменений в файлах, управление версиями и ветвлением, слияение изменений от разных разработчиков, восстановление предыдущих версий файлов и работа с удаленными репозиториями.
7. git init (создание основного дерева репозитория), git pull (получение обновлений текущего дерева из центрального репозитория), git push (отправка всех произведенных изменений локального дерева в центральный репозиторий), git status (просмотр списка измененных файлов в текущей директории), git diff (просмотр текущих изменений), git add (сохранение текущих изменений), git commit (сохранение добавленных изменений), git pust origin (отправка изменений конкретной ветки в центральный репозиторий), git merge –no-ff (слияние ветки с текущим деревом), git branch (управление ветками).
8. Локальный репозиторий: git add ., git commit Удаленный репозиторий: git clone, git pull
9. Ветви - нужны для работы с отдельными частями проекта, не мешая основной части.
10. С помощью файла .gitignore, чтобы игнорировать добавление файлов, которые не требуются в репозитории.

# Список литературы

1. [Лабораторная работа № 2](https://esystem.rudn.ru/mod/page/view.php?id=1224371).