Отчёт по лабораторной работе №2

Операционные системы

Терещенкова М.В.

Содержание

Список иллюстраций

# 1 Цель работы

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

# 2 Задание

1. Создать базовую конфигурацию для работы с git
2. Создать ключ SSH
3. Создать ключ GPG
4. Настроить подписи Git
5. Заргеистрироваться на GitHub
6. Создать локальный каталог для выполнения заданий по предмету.

# 3 Теоретическое введение

**Системы контроля версий. Общие понятия**

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется.

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.

Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

**Примеры использования git**

Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды git с различными опциями.

Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

# 4 Выполнение лабораторной работы

## 4.1 Установка программного обеспечения

Устанавливаю необходимое программное обеспечение git и gh через терминал с помощью команд: **dnf install git** и **dnf install gh**

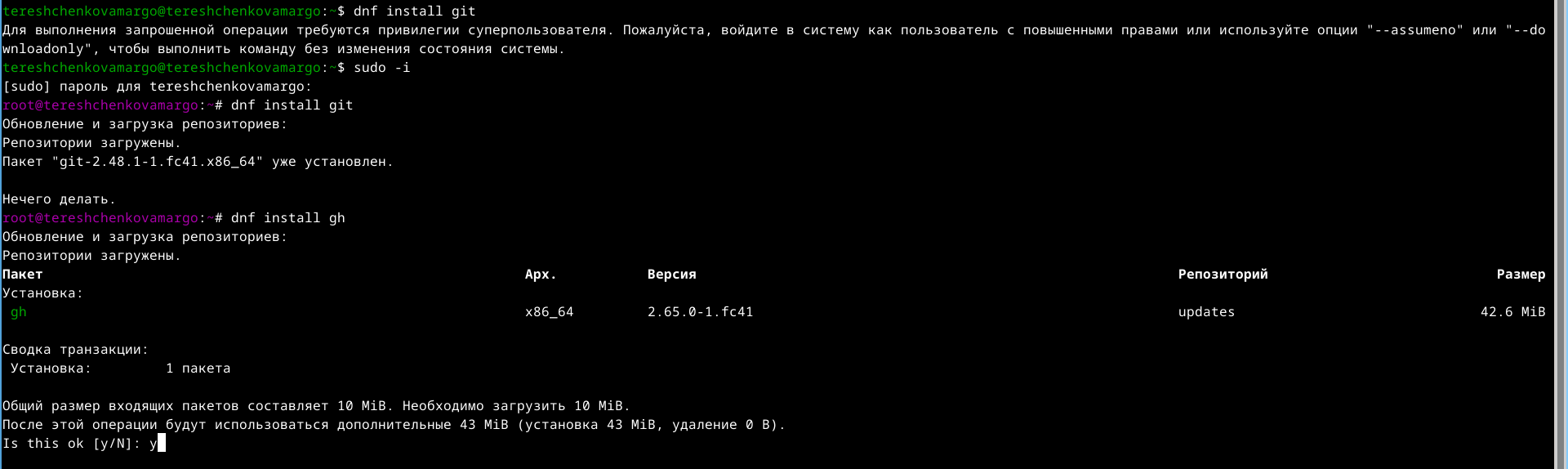


Рис. 1: Установка git и gh

## 4.2 Базовая настройка git

Задаю в качестве имени и email владельца репозитория свои имя, фамилию и электронную почту. Настраиваю utf-8 в выводе сообщений git для их корректного отображения. Настраиваю utf-8 в выводе сообщений git для их корректного отображения.Настраиваю utf-8 в выводе сообщений git для их корректного отображения. Начальной ветке задаю имя master. Задаю параметры autocrlf и safecrlf для корректного отображения конца строки.



Рис. 2: Задаю имя и email владельца репозитория; настраиваю utf-8 в выводе сообщений git; задаю имя начальной ветки и параметры autocrlf и safecrlf

## 4.3 Создание ключа SSH

Создаю ключ ssh размером 4096 бит по алгоритму rsa.

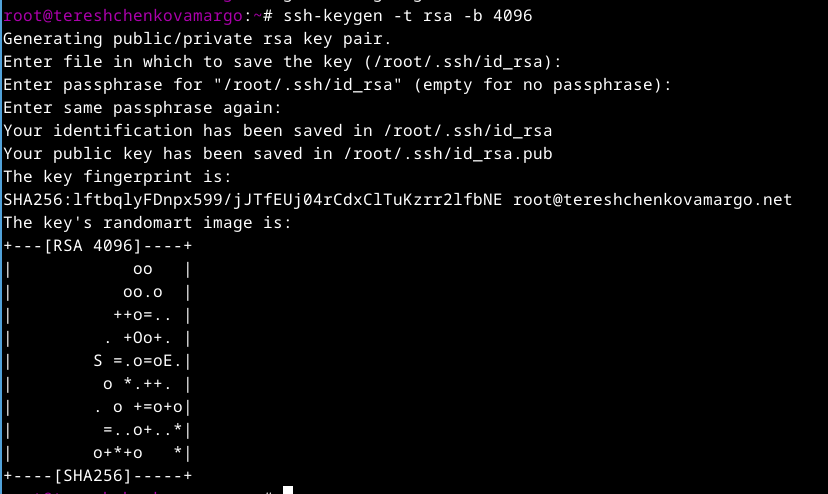


Рис. 3: Создание ключа ssh по алгоритму rsa

Создаю ключ ssh по алгоритму ed25519

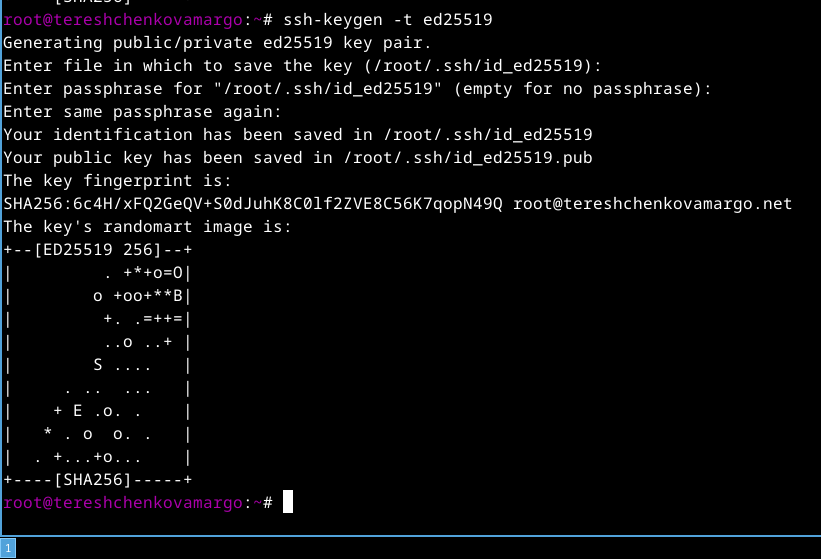


Рис. 4: Создание ключа ssh по алгоритму ed25519

## 4.4 Создание ключей pgp

Генерируем ключ.Из предложенных опций выбираю:тип RSA and RSA; размер 4096;выбераю срок действия; значение по умолчанию — 0 (срок действия не истекает никогда). Также ввожу своё имя и корпоративную почту.

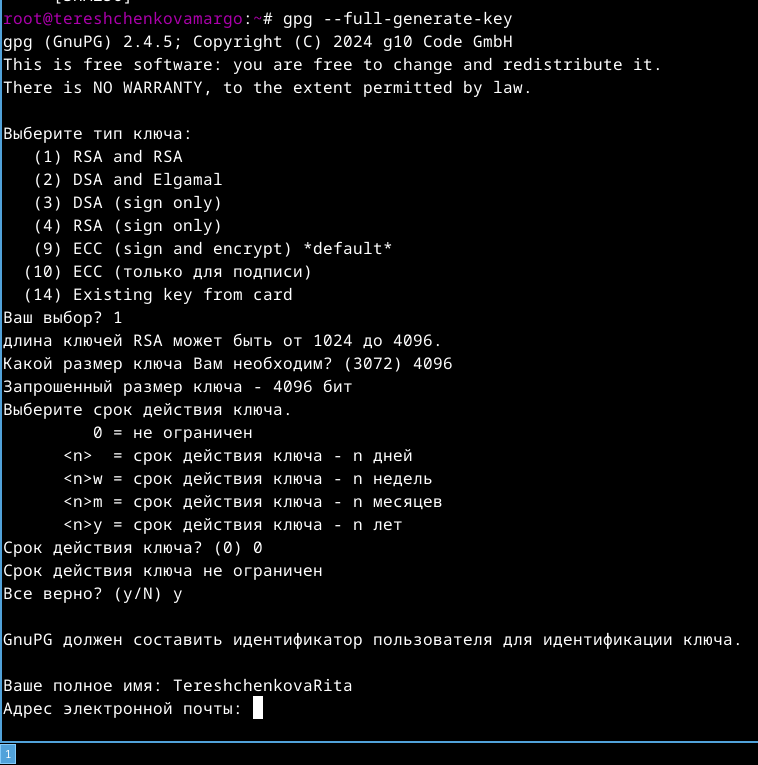


Рис. 5: Создание ключа pgp

## 4.5 Настройка github.

У меня уже настроен github.

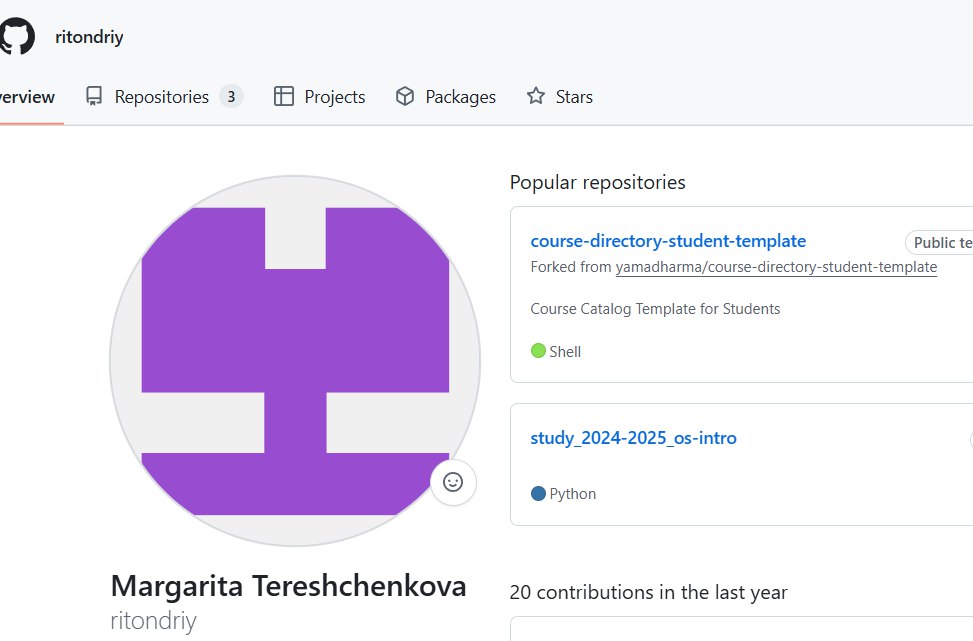


Рис. 6: Мой профиль в github

## 4.6 Добавление PGP ключа в GitHub

Выводим список ключей и копируем отпечаток приватного ключа с помощью команды: **gpg –list-secret-keys –keyid-format LONG**

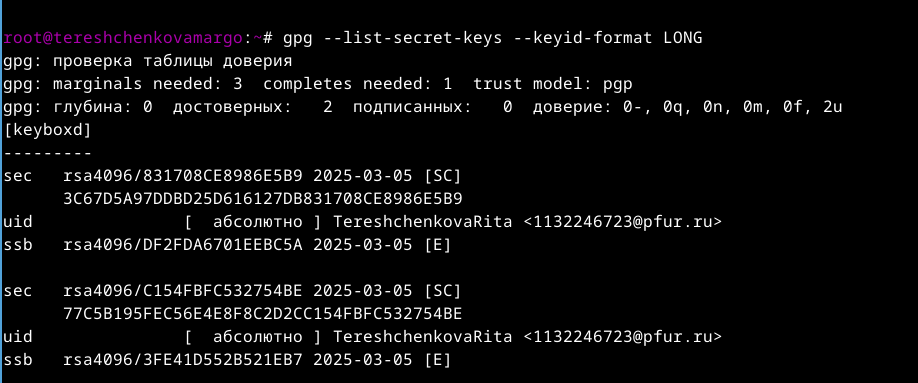


Рис. 7: Вывод списка ключей

Cкопировала свой сгенерированный PGP ключ в буфер обмена: **gpg –armor –export | xclip -sel clip**

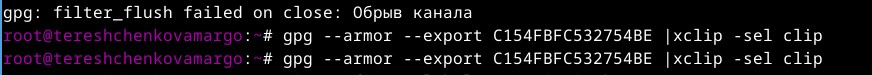


Рис. 8: Команда копирования

Перехожу в настройки GitHub, нажимаю на кнопку New GPG key и вставляю полученный ключ в поле ввода.

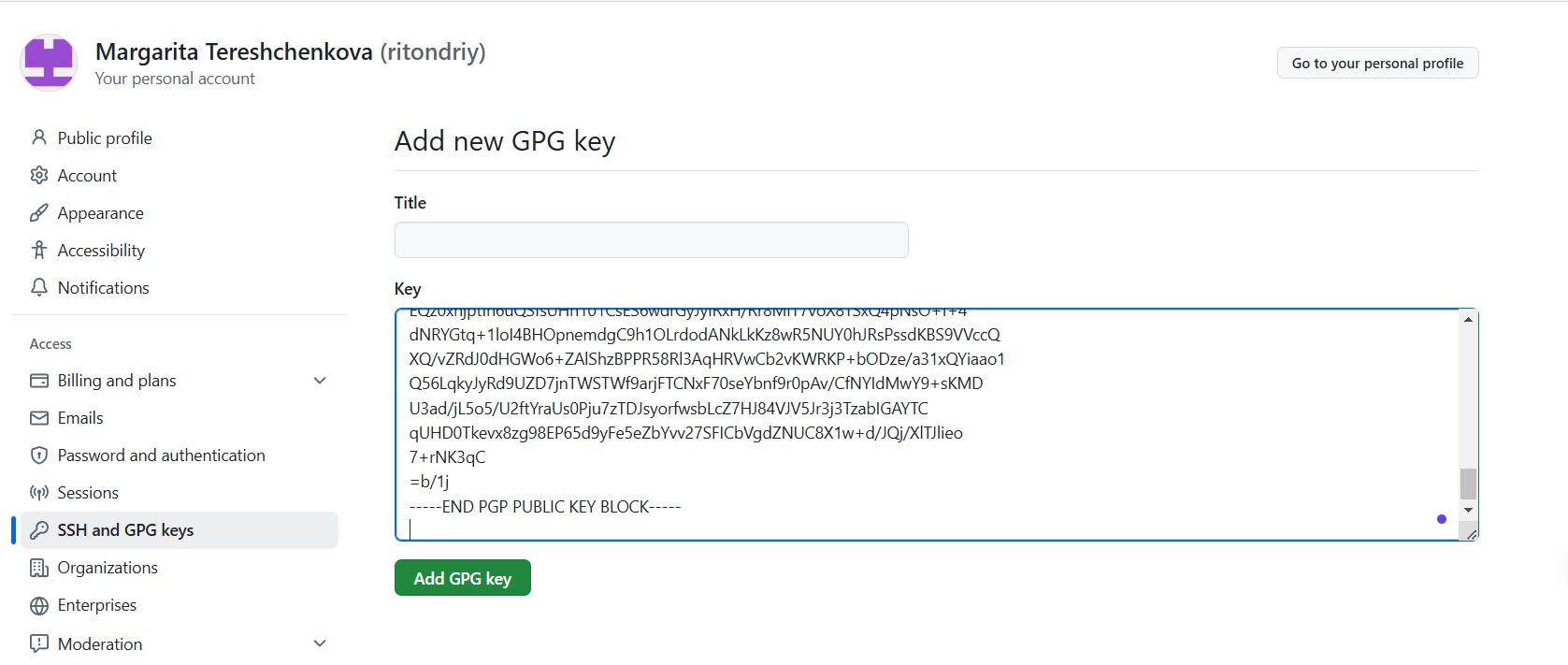


Рис. 9: New GPG key

## 4.7 Настройка автоматических подписей коммитов git

Используя введёный email, указываю Git применять его при подписи коммитов.

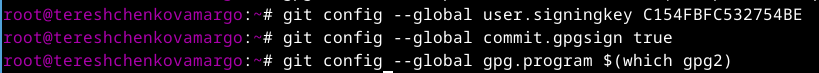


Рис. 10: Настройка автоматических подписей коммитов git

## 4.8 Настройка gh

Авторизовываюсь с помощью команды: **gh auth login** и отвечаю на вопросы утилиты.



Рис. 11: Авторизация gh

Ввожу код.

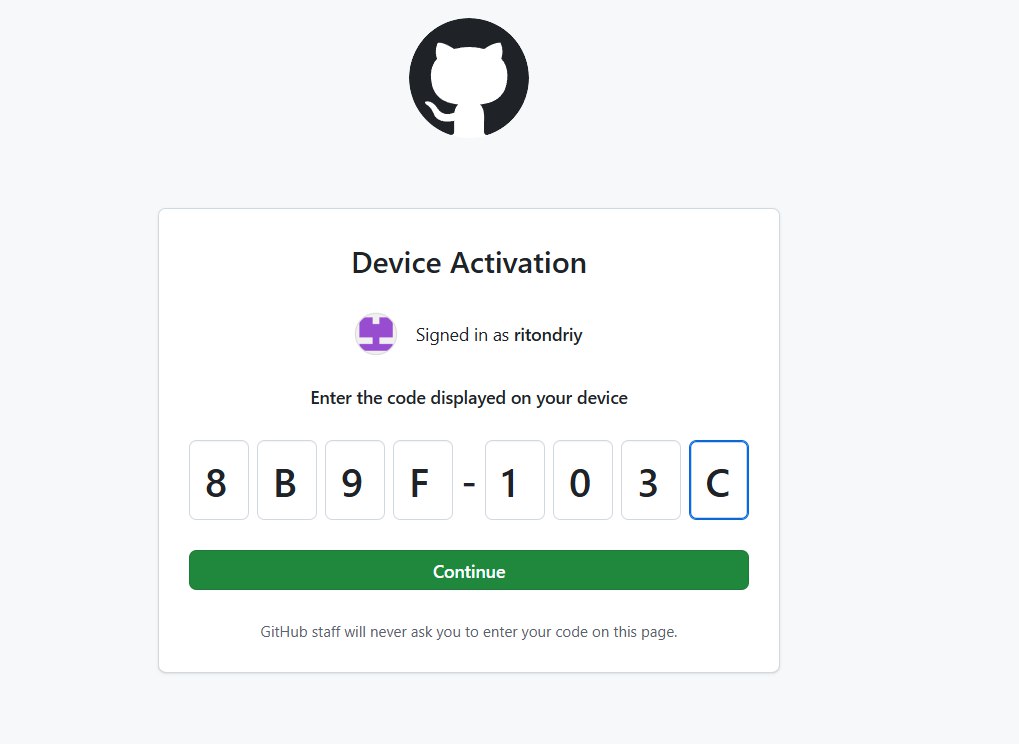


Рис. 12: one-time code

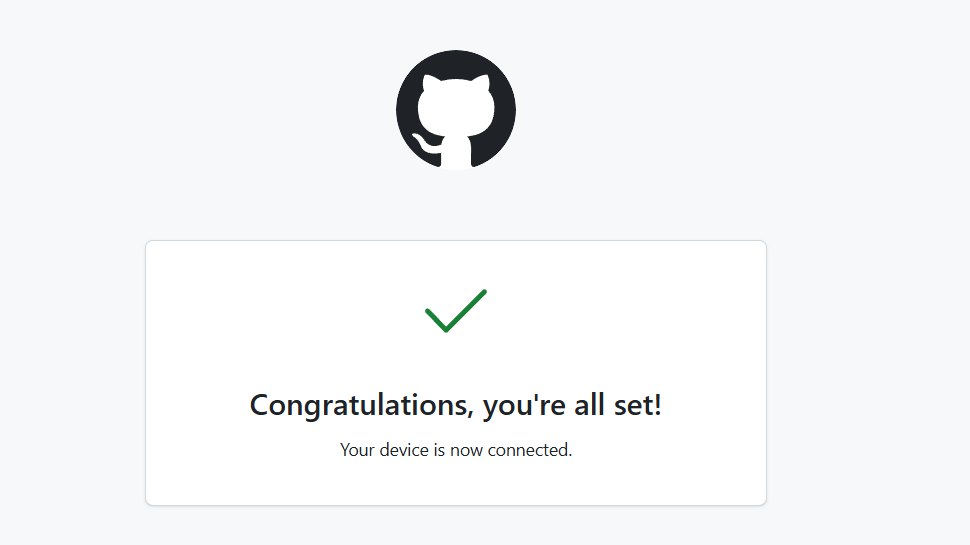


Рис. 13: Успешная авторизация

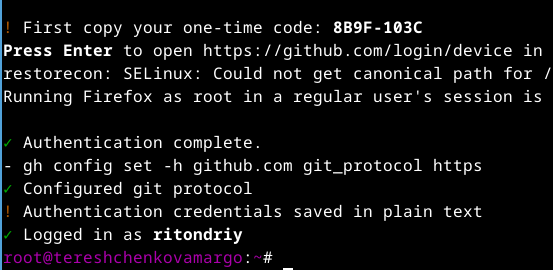


Рис. 14: Успешная авторизация.2

## 4.9 Создание репозитория курса на основе шаблона

Сначала создаю директорию с помощью утилиты mkdir и флага -p, который позволяет установить каталоги на всем указанном пути. После этого с помощью утилиты cd перехожу в только что созданную директорию “Операционные системы”. Далее в терминале ввожу команду gh repo create study\_2024-2025\_os-intro–template yamadharma/course-directory-student-trmplate –public, чтобы создать репозиторий на основе шаблона репозитория. После этого клонирую репозиторий к себе в директорию, я указываю ссылку с протоколом https, а не ssh, потомучто при авторизации в gh выбрала протокол https.

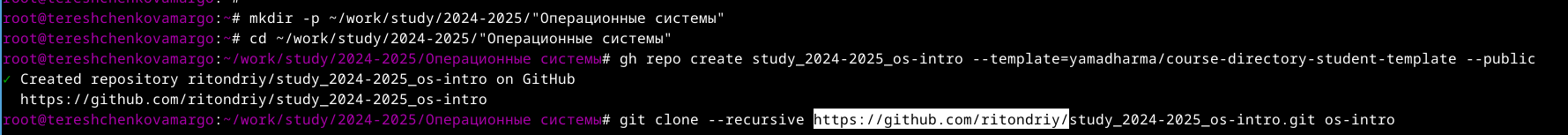


Рис. 15: Создание репозитория

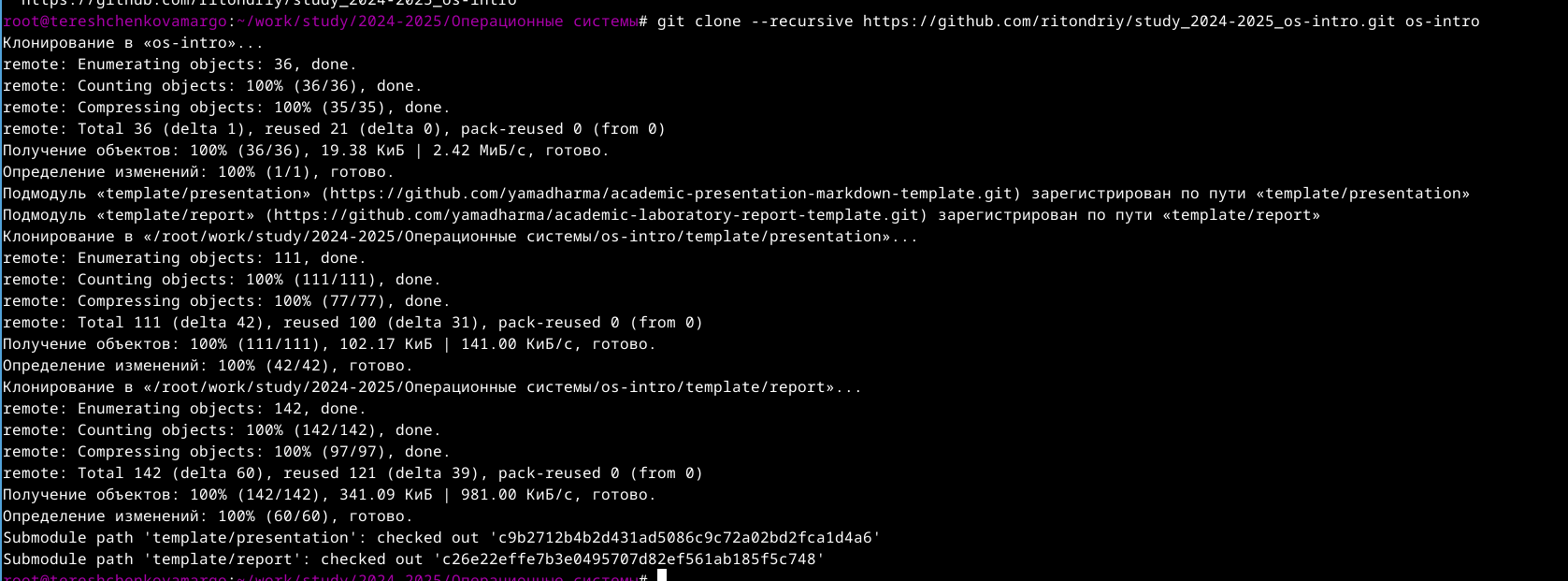


Рис. 16: Создание репозитория.2

Перехожу в каталог курса с помощью утилиты cd, проверяю содержание каталога с помощью утилиты ls.

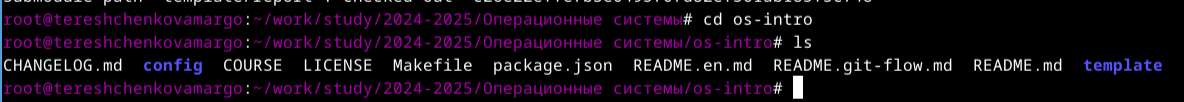


Рис. 17: Перемещение между директориями

## 4.10 Настройка каталога курса

Удаляю лишние файлы с помощью утилиты **rm**, далее создаю необходимые каталоги используя makefile и проверяю корректность выполнения с помощью утилиты **ls**.

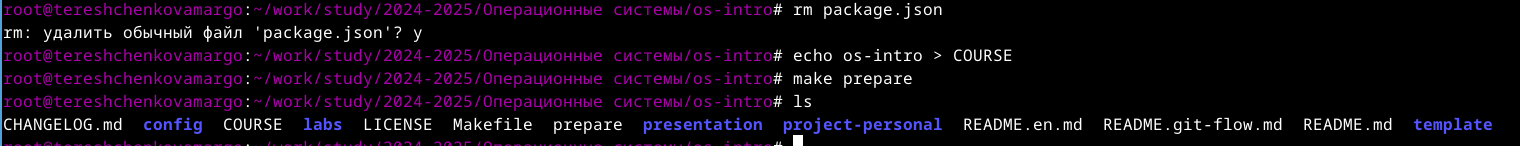


Рис. 18: Удаление файлов и создание каталогов

Добавляю все новые файлы для отправки на сервер (сохраняю добавленные изменения) с помощью команды **git add** и комментирую их с помощью **git commit**.

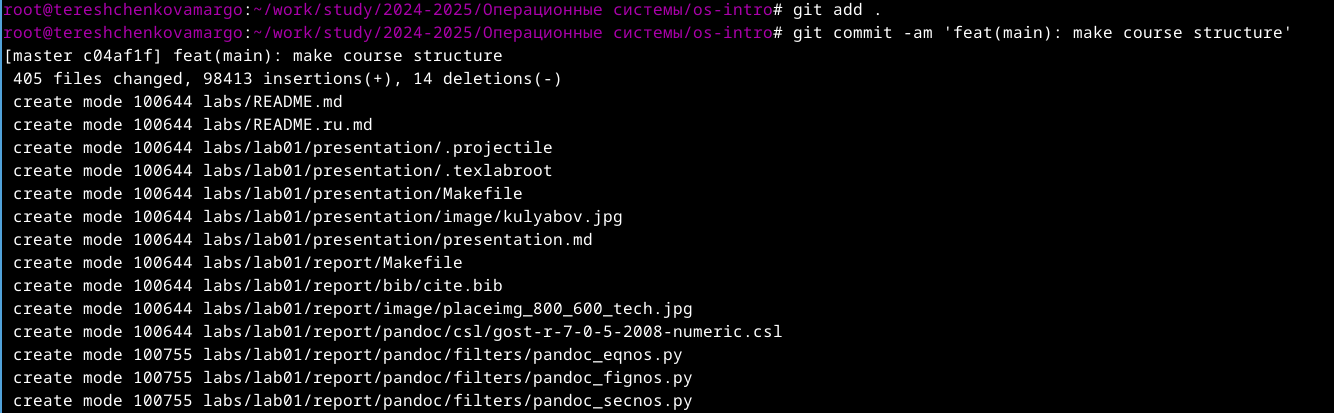


Рис. 19: Отправка файлов на сервер

Отправляю файлы на сервер с помощью **git push**

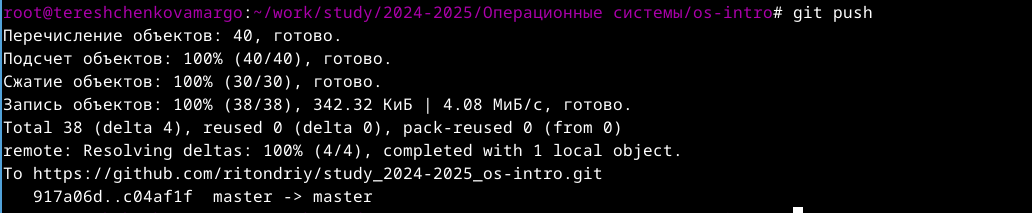


Рис. 20: Отправка файлов на сервер.2

# 5 Выводы

Изучила идеологию и применение средств контроля версий;освоила умения по работе с git.

# Список литературы