Отчёт по лабораторной работе 2

отчет

Апареев Дмитрий Андреевич

Содержание

# 1 Цель работы

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

# 2 Задание

Первоначальна настройка git

# 3 Теоретическое введение

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.

# 4 Выполнение лабораторной работы

Сначала необходимо установить git (рис. [1](#fig:001)).

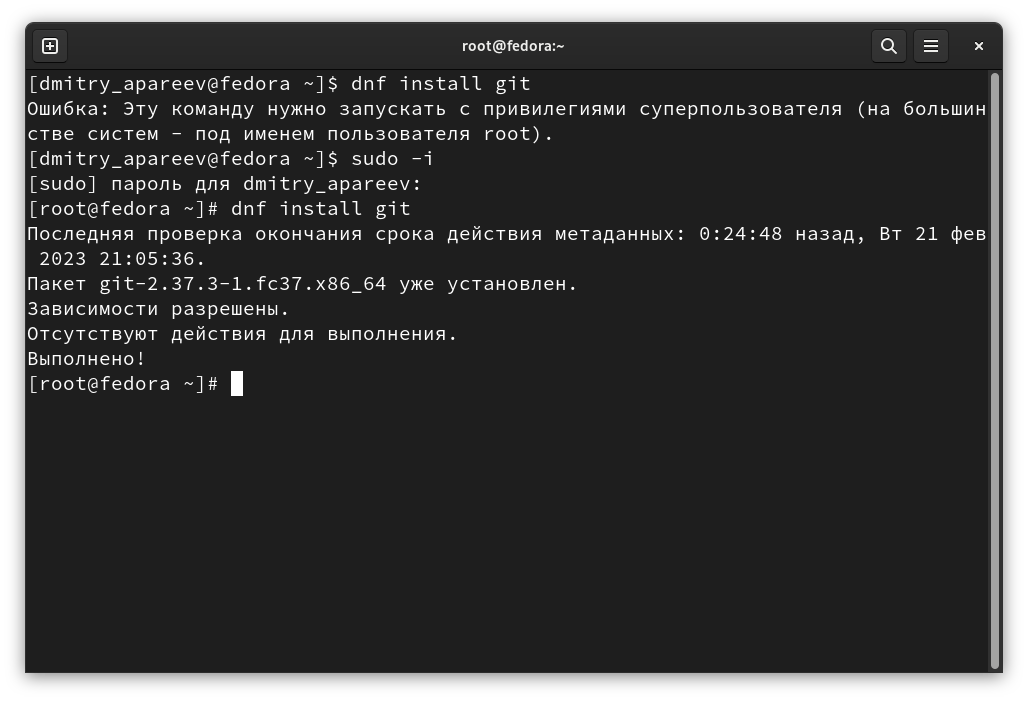


Figure 1: Установка

Установил gh (рис. [2](#fig:002)).

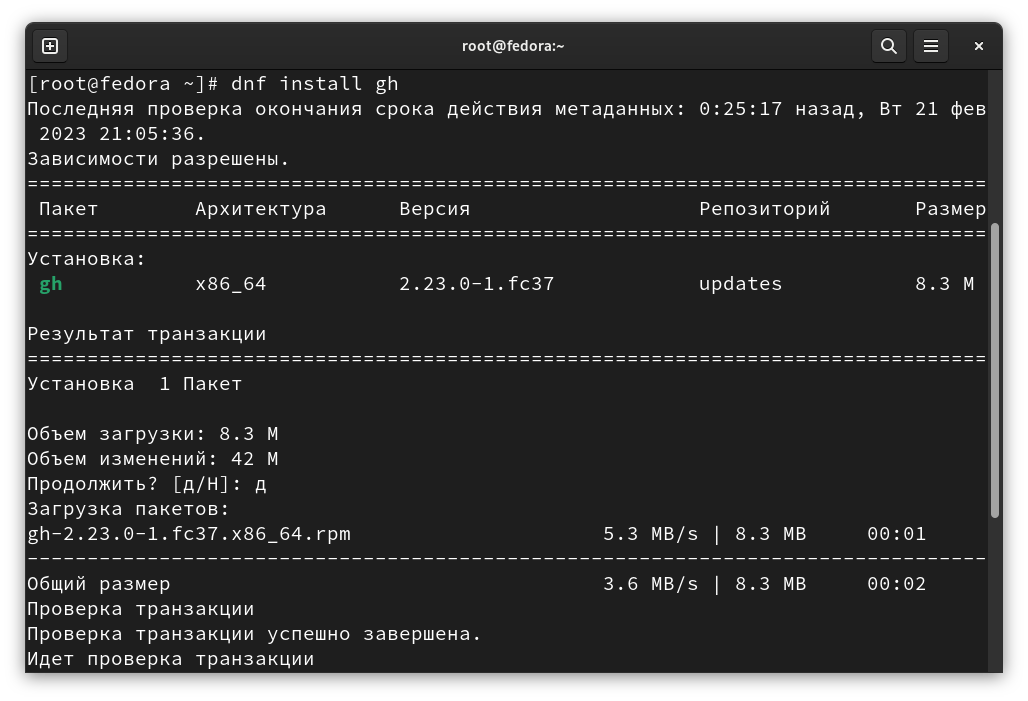


Figure 2: Установка

Задаю имя и email владельца репозитория: (рис. [3](#fig:003)).

Figure 3: задал имя и почту

Figure 3: задал имя и почту

Настроил utf-8 в выводе сообщений git: (рис. [4](#fig:004)).

Figure 4: настройка utf-8

Figure 4: настройка utf-8

Зададим имя начальной ветки (будем называть её master): (рис. [5](#fig:005)).

Figure 5: имя начальной ветки

Figure 5: имя начальной ветки

Параметр autocrlf, Параметр safecrlf: (рис. [6](#fig:006)).

Figure 6: 2 параметра

Figure 6: 2 параметра

по алгоритму rsa с ключём размером 4096 бит: (рис. [7](#fig:007)).

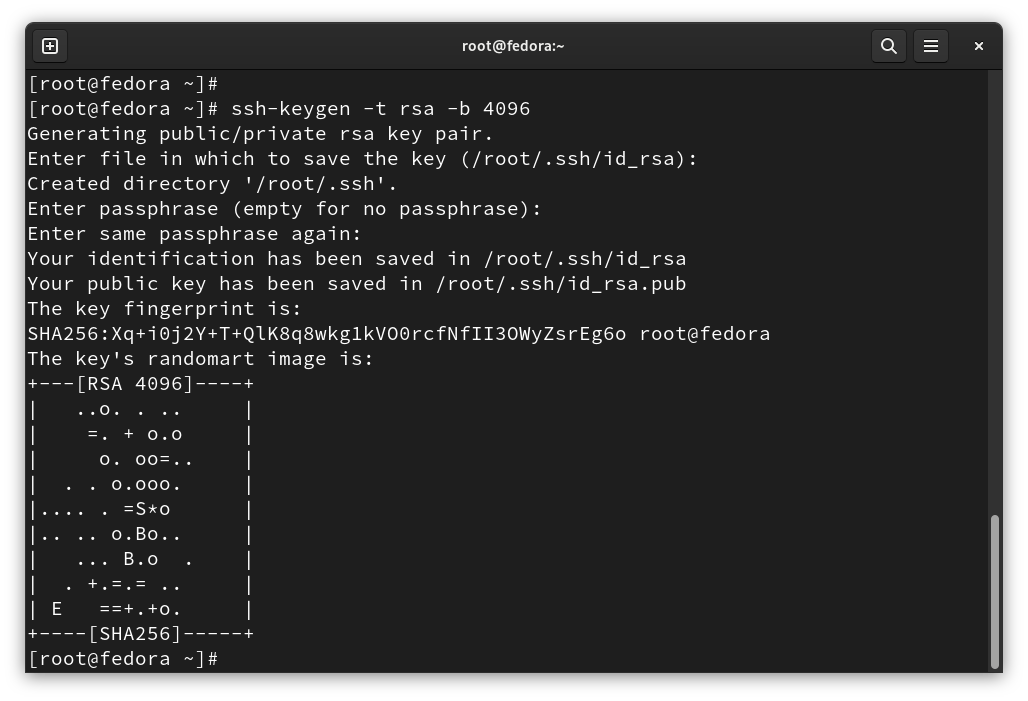


Figure 7: ключ

по алгоритму ed25519: (рис. [8](#fig:008)).

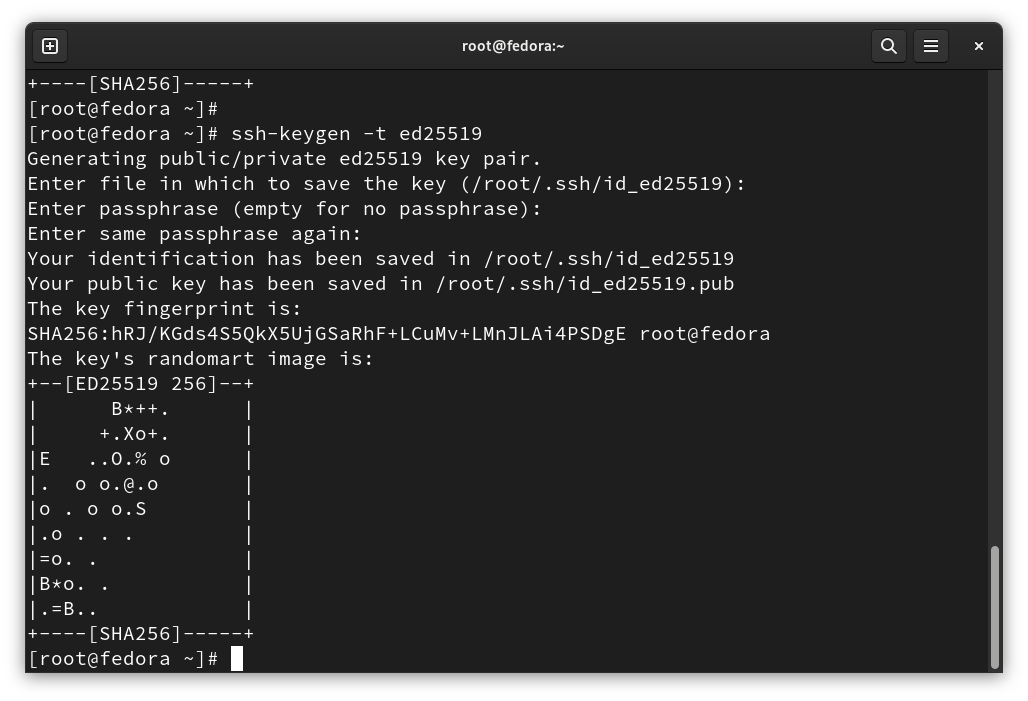


Figure 8: ключ

Генерирую ключ pgp (рис. [9](#fig:009)).

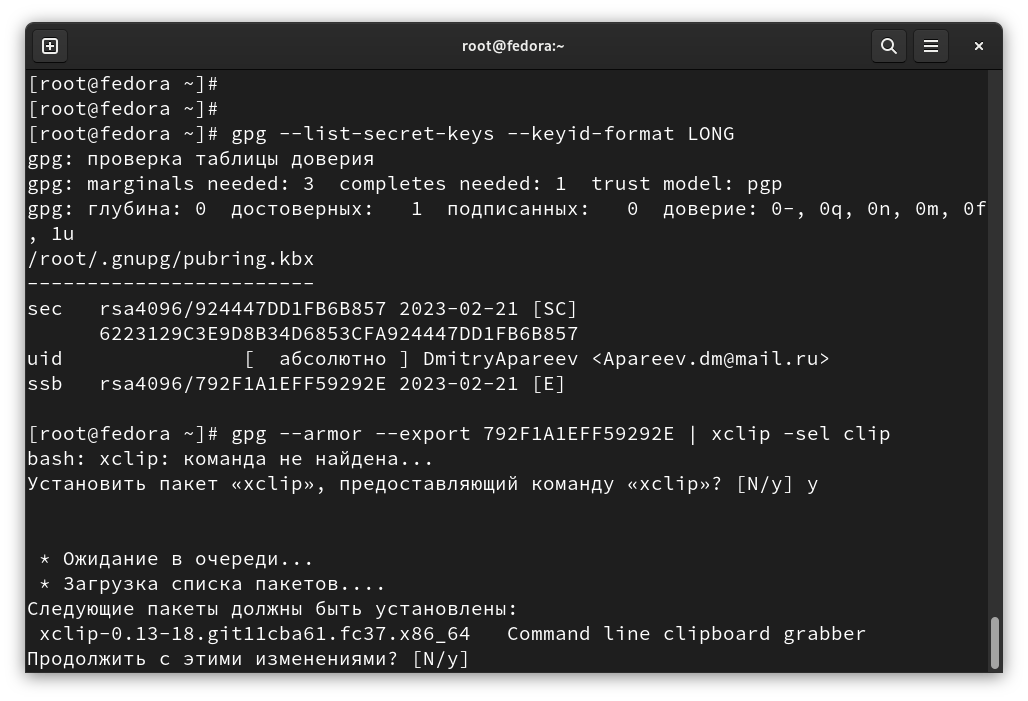


Figure 9: ключ

Вставляю копию ключа на гитхаб (рис. [10](#fig:010)).

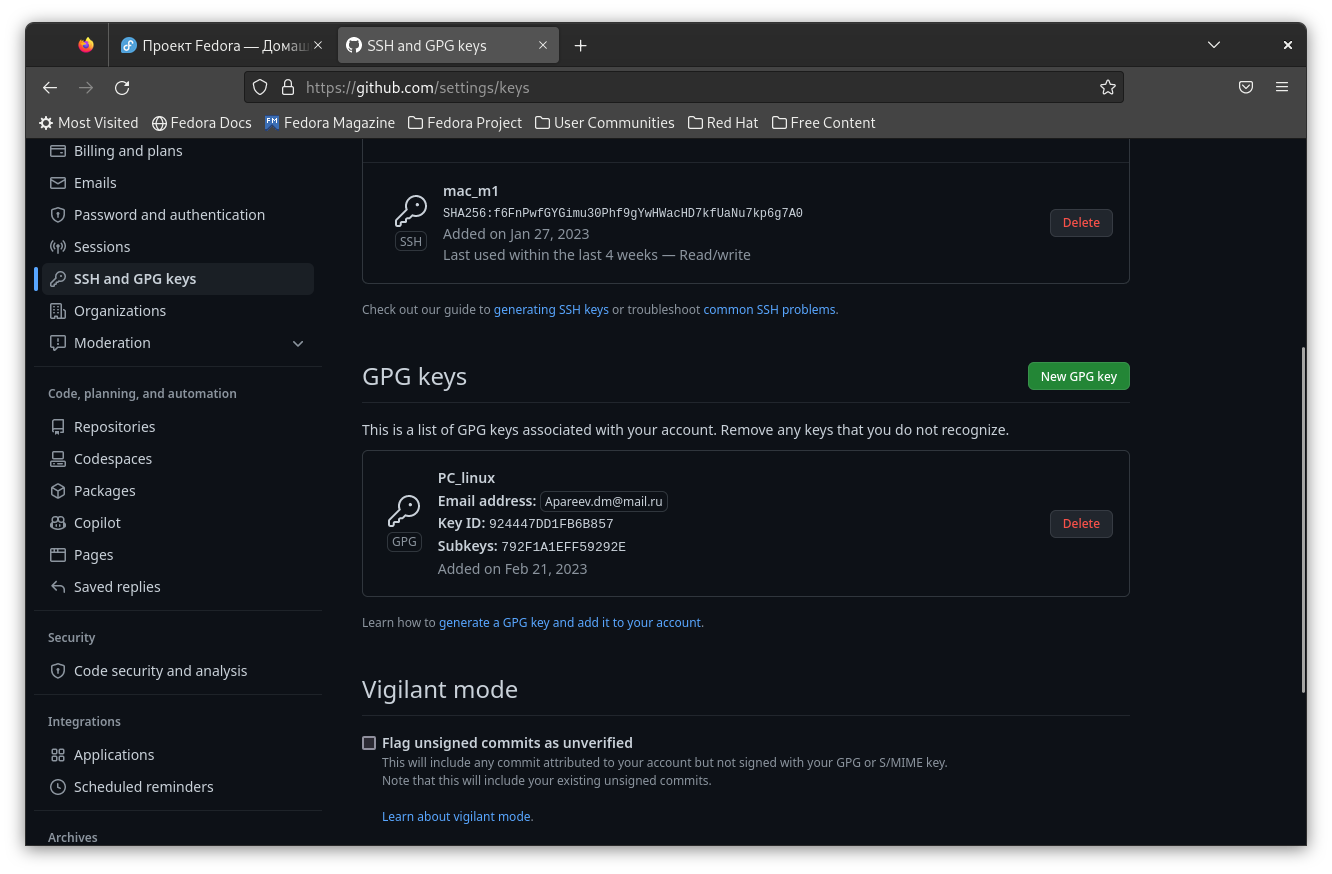


Figure 10: вставка ключа на гитхаб

Используя введёный email, укажите Git применять его при подписи коммитов: (рис. [11](#fig:011)).

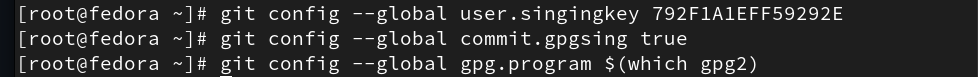


Figure 11: настройка подписей

Настройка gh: (рис. [12](#fig:012)). (рис. [13](#fig:013)).

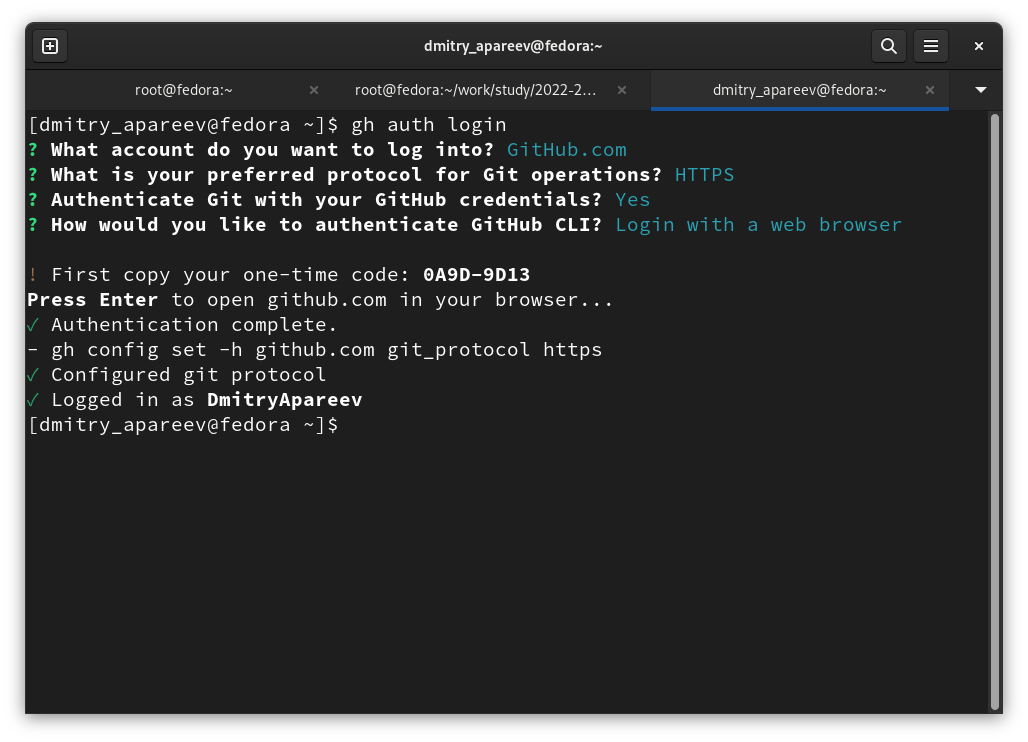


Figure 12: настройка gh

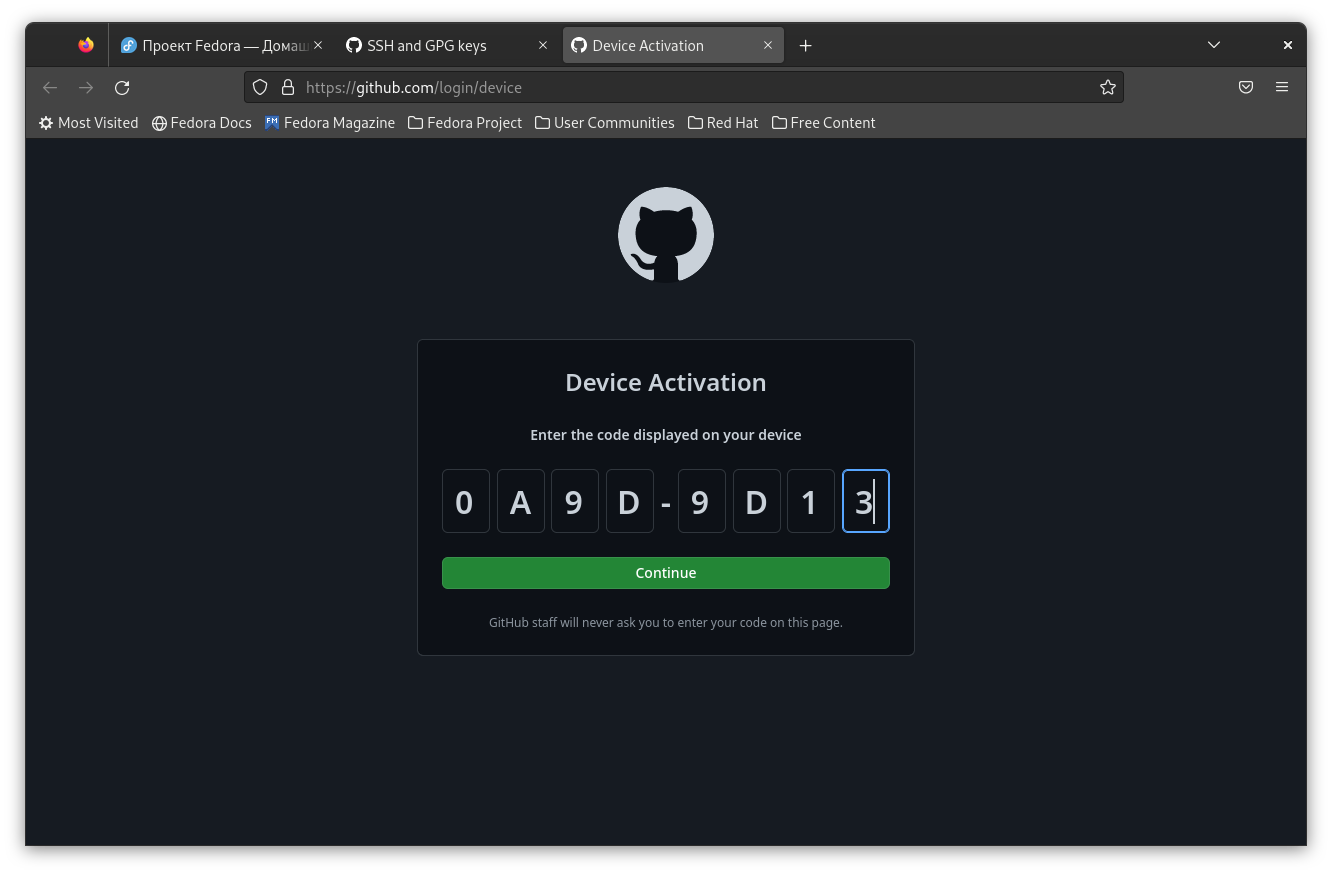


Figure 13: настройка gh

Создаю папку Операционные системы и перехожу в нее, затем создаю репозиторий на гитхаб (рис. [14](#fig:014)).

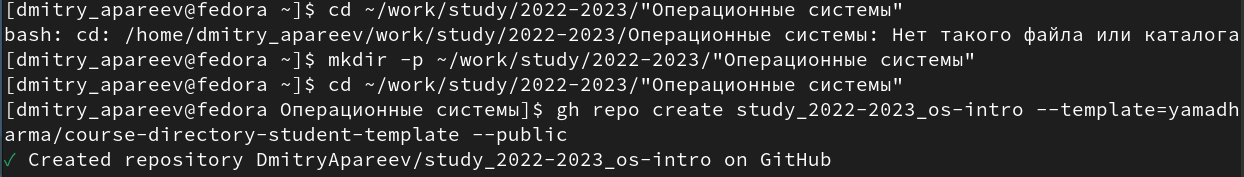


Figure 14: создание курса

Клонирую шаблон репозитория к себе (рис. [15](#fig:015)).

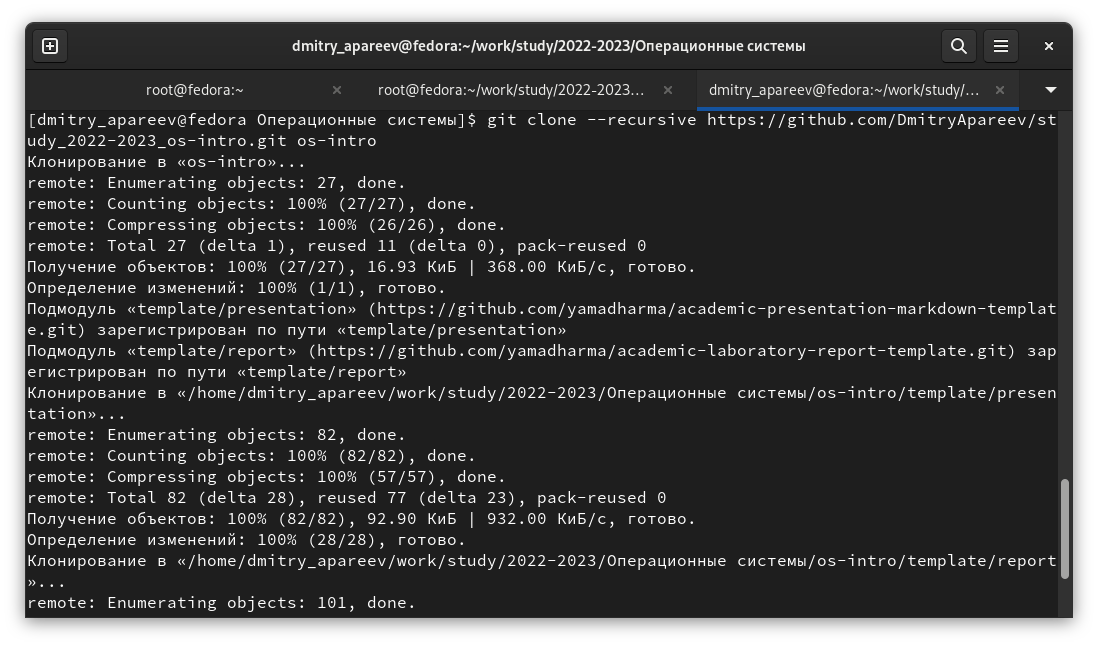


Figure 15: копирование

Перехожу в папку os-intro, удаляю лишние файлы (рис. [16](#fig:016)). (рис. [17](#fig:017)).

Figure 16: переход

Figure 16: переход

Figure 17: удаление файлов

Figure 17: удаление файлов

Создаю необходимые каталоги (рис. [18](#fig:018)).

Figure 18: создание каталога

Figure 18: создание каталога

Добавляю и отправляю файлы на гитхаб (рис. [19](#fig:019)). (рис. [20](#fig:020)).

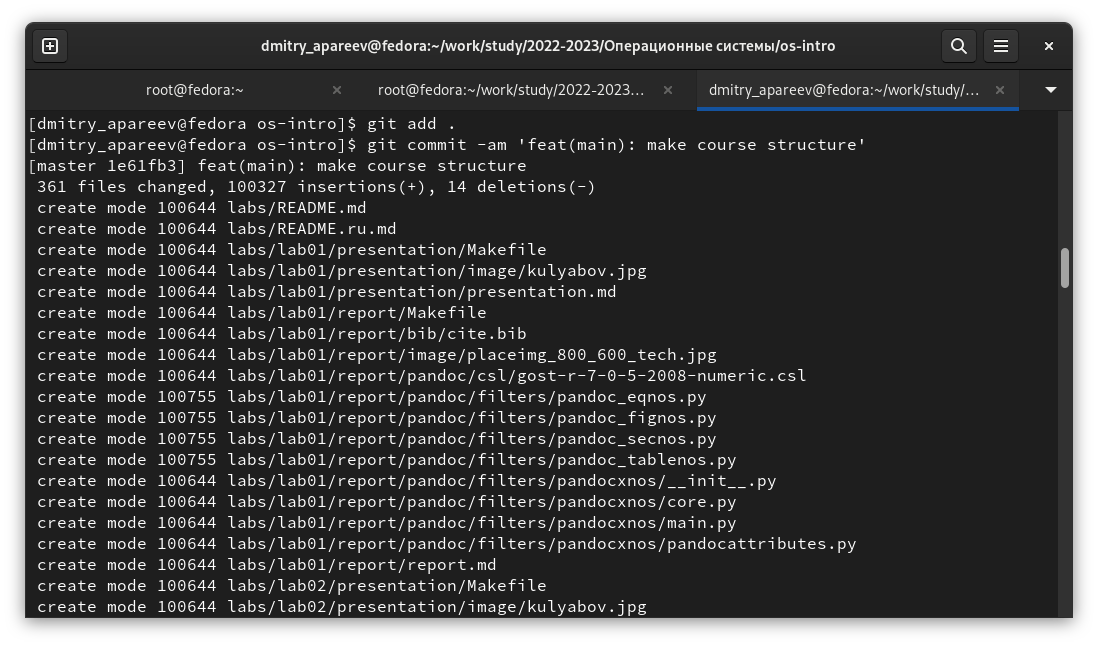


Figure 19: добавление

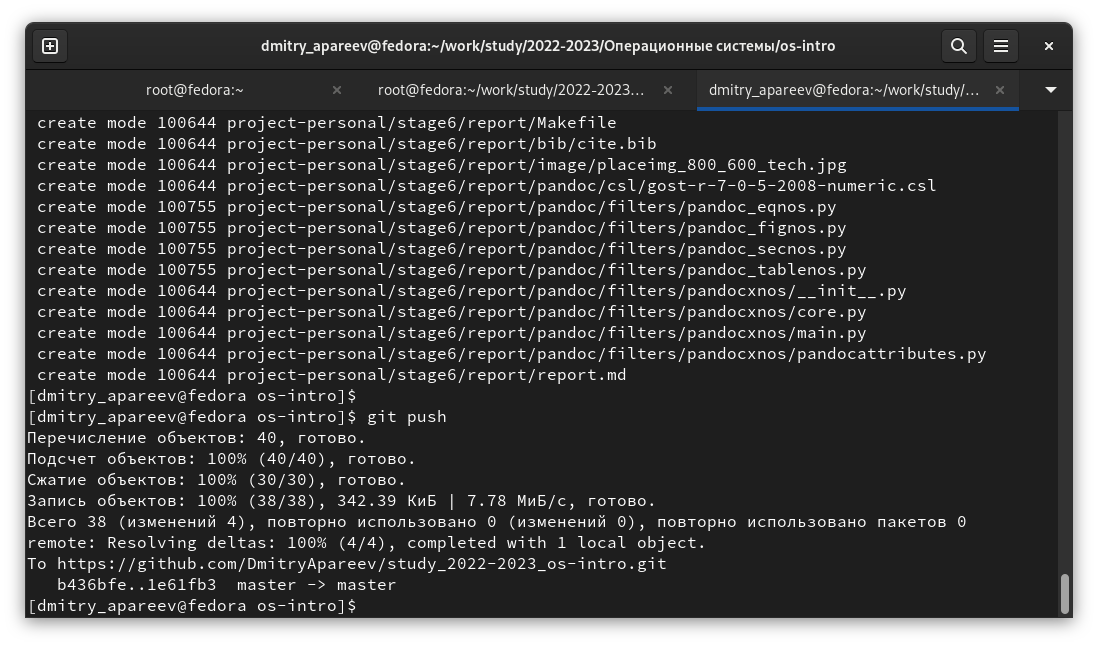


Figure 20: отправка на гитхаб

# 5 Выводы

Выполнив лабораторную работу, я научился прикреплять гитхаб репозиторий и отправлять коммиты