Лабораторная работа №2

Первоначальная настройка git

Полякова Юлия Александровна

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

1. Установка git и gh командой dnf install (рис. 1).

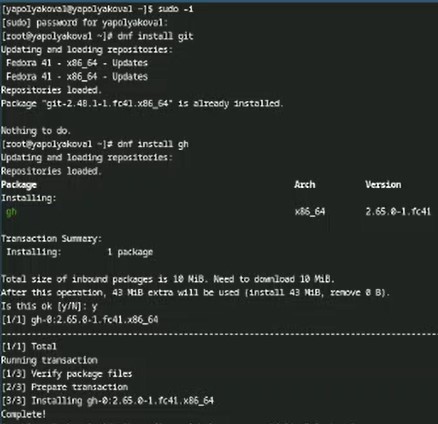


Рис. 1: Установка ПО

1. Базовая настройка git. Задаем имя и email, настраиваем utf-8, задаем имя начальной ветки и параметры autocrlf и safecrlf (рис. 2)

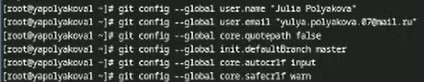


Рис. 2: Базовая настройка

1. Создание ключей ssh по алгоритмам rsa и ed25519 (рис. 3)

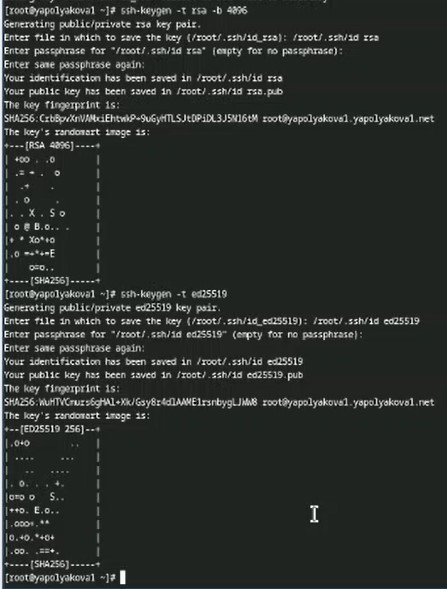


Рис. 3: Создание ключей ssh

1. Создание ключа gpg с типом RSA and RSA, размером 4096 и сроком действия по умолчанию. Также добавляем личную информацию (имя, адрес почты, пустой комментарий) (рис. 4)

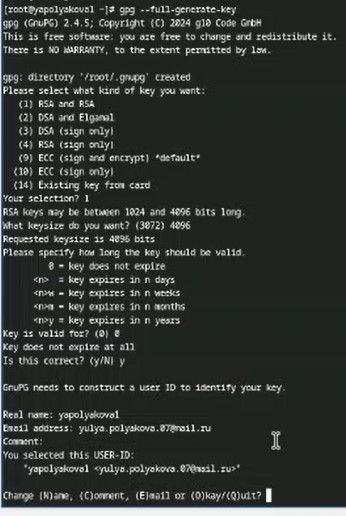


Рис. 4: Создание ключа gpg

1. Учетная запись в GitHub уже была настроена, поэтому переходим к добавлению ключа. Выводим ключ по его отпечатку и копируем в настройки на сайте (рис. 5)

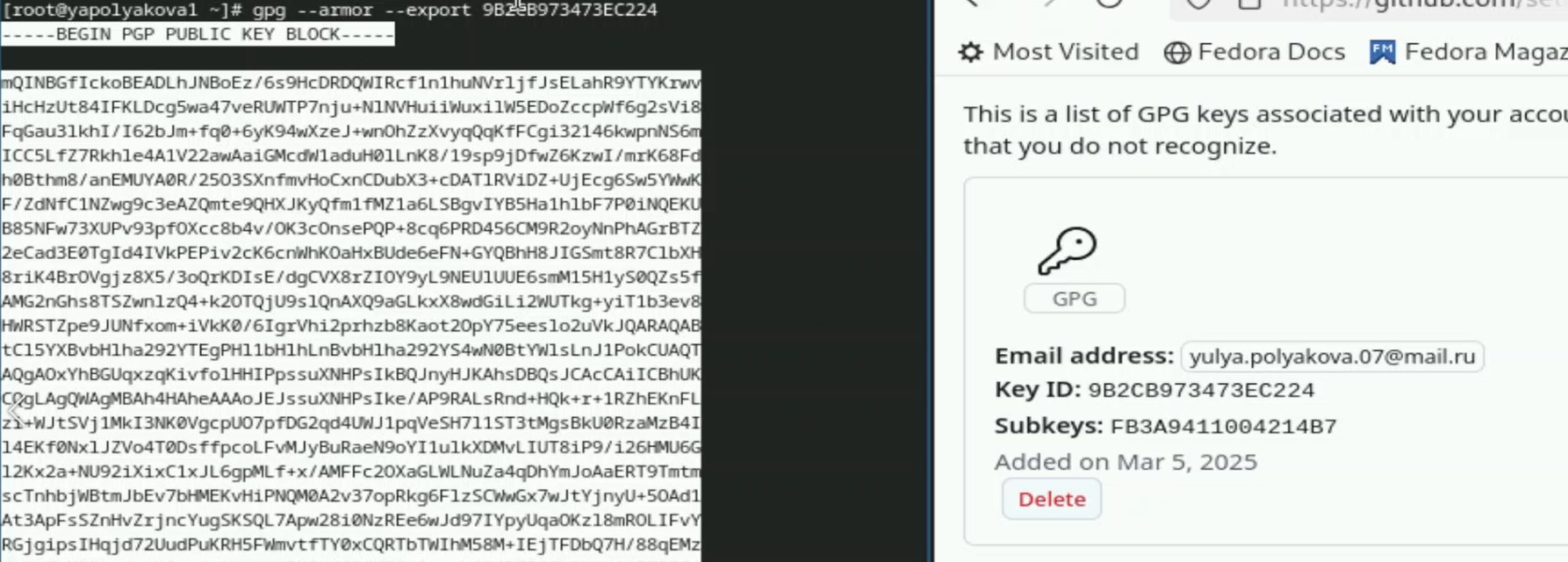


Рис. 5: Добавление ключа gpg

1. Настройка автоматических подписей коммитов git (рис. 6)

Настройка подписей коммитов

Рис. 6: Настройка подписей коммитов

1. Авторизуемся с помощью gh auth login. Создаем папку курса по шаблону, переходим в нее. Создаем в терминале репозиторий по указанному шаблону. Далее клонируем получившийся репозиторий в папку os-intro. На этом моменте возникла сложность с ключом (Permission denied (publickey)), поэтому дополнительно был добавлен на сайт GitHub ключ ssh. После этого был успешно склонирован репозиторий (рис. 7)

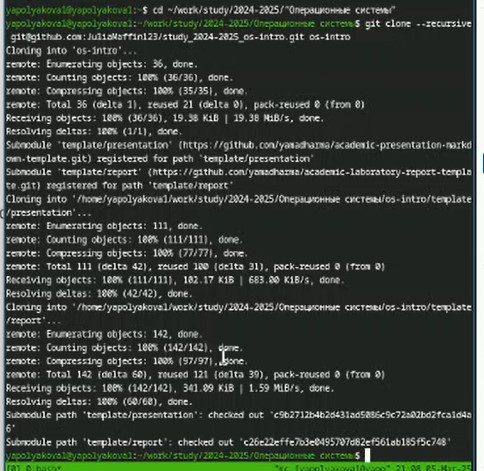


Рис. 7: Клонирование репозитория в папку os-intro

1. Настройка каталога курса (удаление лишних файлов, создание необходимых каталогов командой make) (рис. 8)

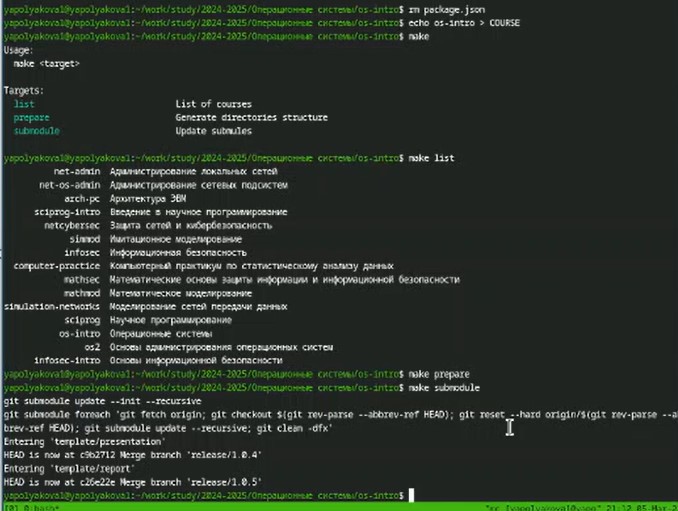


Рис. 8: Настройка каталога курса

1. Успешное отправление файлов на сервер командами git add . (добавить все файлы), git commit (создание коммита) и git push (непосредственно отправка файлов на сервер) (рис. 9)

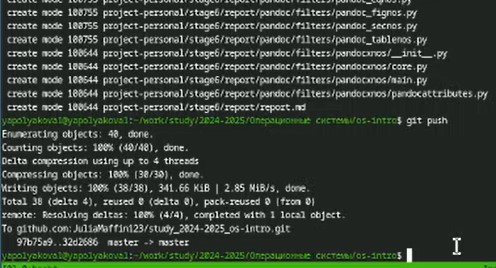


Рис. 9: Отправление файлов на сервер

# 3 Контрольные вопросы

1. VCS - это инструмент, который помогает разработчикам отслеживать изменения в коде, управлять версиями файлов и координировать работу в команде. Задачи: отслеживание изменений, решение конфликтов, поддержка параллельной разработки, управление версиями, совместная работа и улучшение координации.
2. Хранилище (репозиторий) - место хранения всех версий и служебной информации. Commit - состояние проекта на определенный момент времени или логическая группа измениений, которые пользователь потом отправляет в хранилище. История - сохраниение полной истории изменений. Рабочая копия - текущее состояние файлов проекта, полученных из хранилища и возможно измененных.
3. Централизованные имеют одно основное хранилище всего проекта. Каждый пользователь копирует себе изменения оттуда, а потом загружает свои. Пример: Subversion, CVS, TFS, VAULT, AccuRev. Децентрализованные имеют у каждого пользователя свой вариант репозитория, возможно даже не один. Есть возможность добавлять и забирать изменения из любого репозитория. Пример: Git, Mercurial, Bazaar.
4. Получение нужной версии файлов, размещение новой версии в хранилище, обновление рабочей копии, отслеживание и разрешение конфликтов.
5. То же самое, только с более тщательным контролем, “чистыми коммитами”, советуясь с другими пользователями.
6. Отслеживание изменений, добавление новых функций в проект, объединение разных версий кода в один, откат к предыдущей версии, исправление конфликтов.
7. Команды можно разделить на работу с деревом, работу с изменениями, слияние и удаление веток.
8. Например, студент создает удаленный репозиторий, клонирует его себе, пишет программу, делает commit и отправляет изменения в локальном репозитории на удаленный. Его друг внес изменения на удаленном репозитории, а студент в следующий раз их получил с помощью pull
9. С помощью ветвей можно разрабатывать новый функционал независимо от основного кода. Можно делать откат и переключаться между версиями, не боясь что-то потерять.
10. Игнорировать можно не добавляя их add-ом, в случае, если эти файлы не нужны на удаленном репозитории.

# 4 Выводы

Была изучена идеология и применение средств контроля версий. Были освоены умения по работе с git.