Лабораторная работа №1

Научное программирование

Леонтьева Ксения Андреевна | НПМмд-02-23

Содержание

# 1 Цель работы

Изучить идеалогию и применение средств контроля версий, а также освоить умения по работе с git.

# 2 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. 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.

# 3 Выполнение лабораторной работы

Для начала была создана учетная запись и заполнены основные данные на сайте https://github.com (рис. fig. 1).

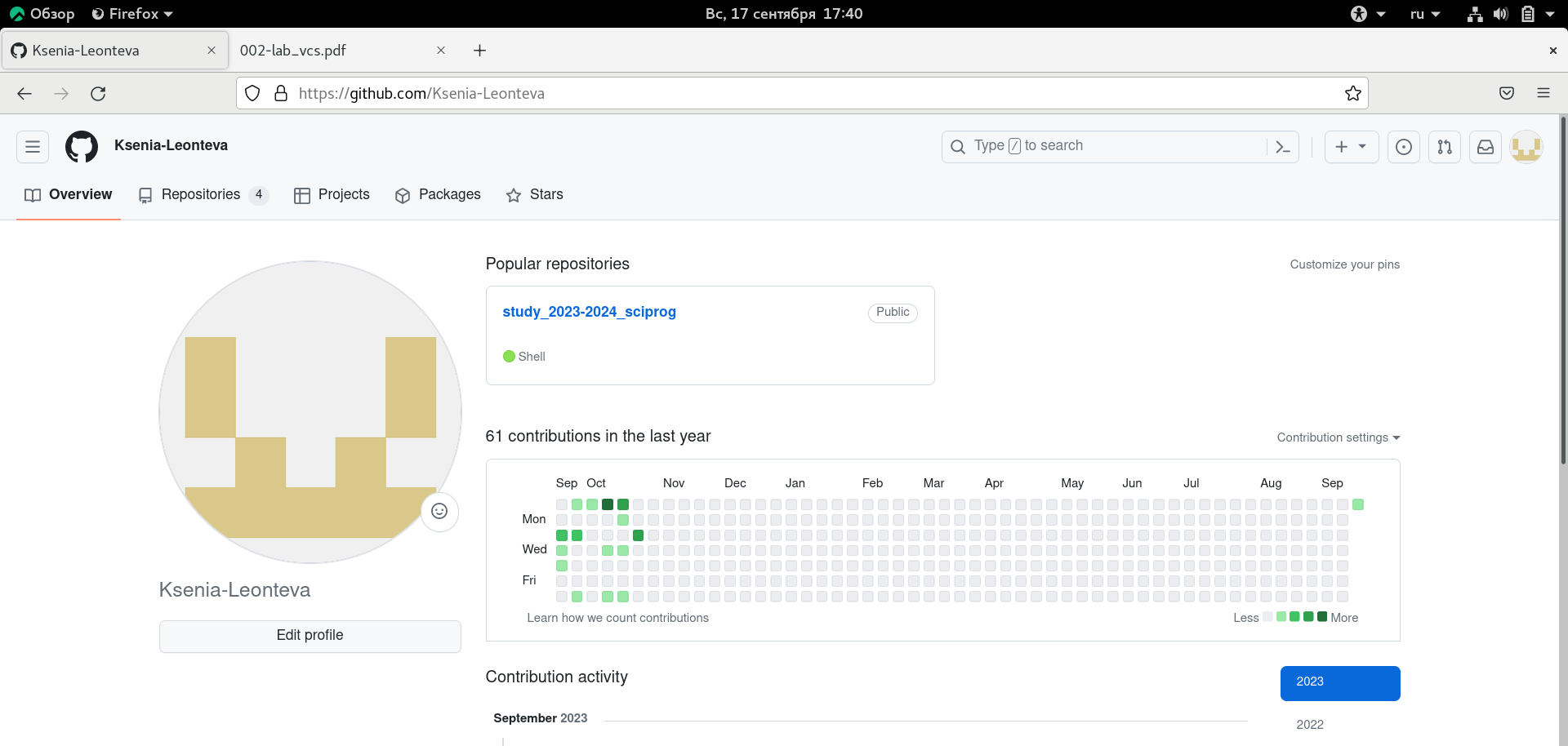


Рис. 1: Учетная запись на GitHub

Вручную установили программное обеспечение git-flow (рис. fig. 2).

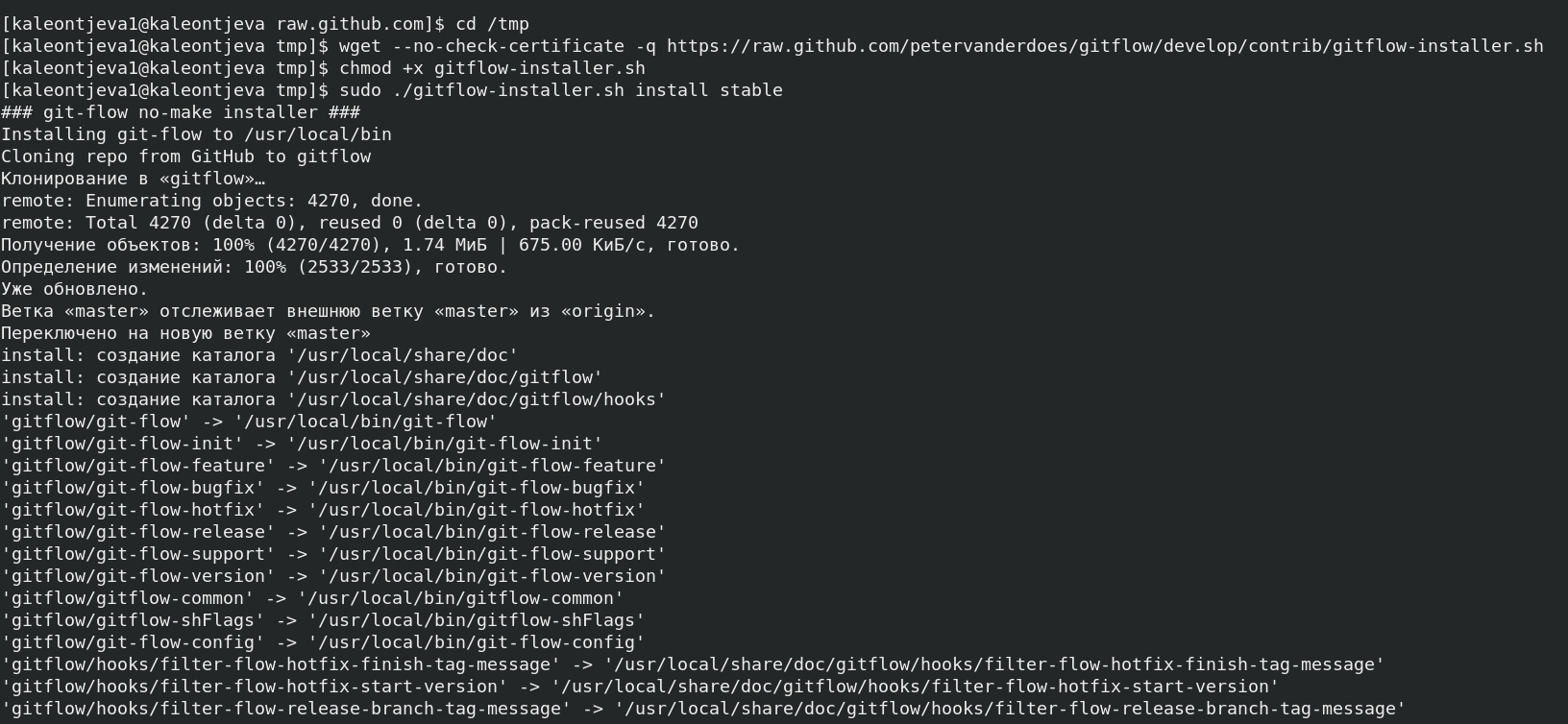


Рис. 2: Установка git-flow

Осуществили базовую настройку git (рис. fig. 3), а именно:

* задали имя и email владельца репозитория,
* настроили utf-8 в выводе сообщений git,
* настроили верификацию и подписание коммитов git,
* задали имя начальной ветки (master),
* параметр autocrlf,
* параметр safecrlf.

Также создали ключи ssh (рис. fig. 3 и рис. fig. 4):

* по алгоритму rsa с ключом размером 4096 бит,
* по алгоритму ed25519.

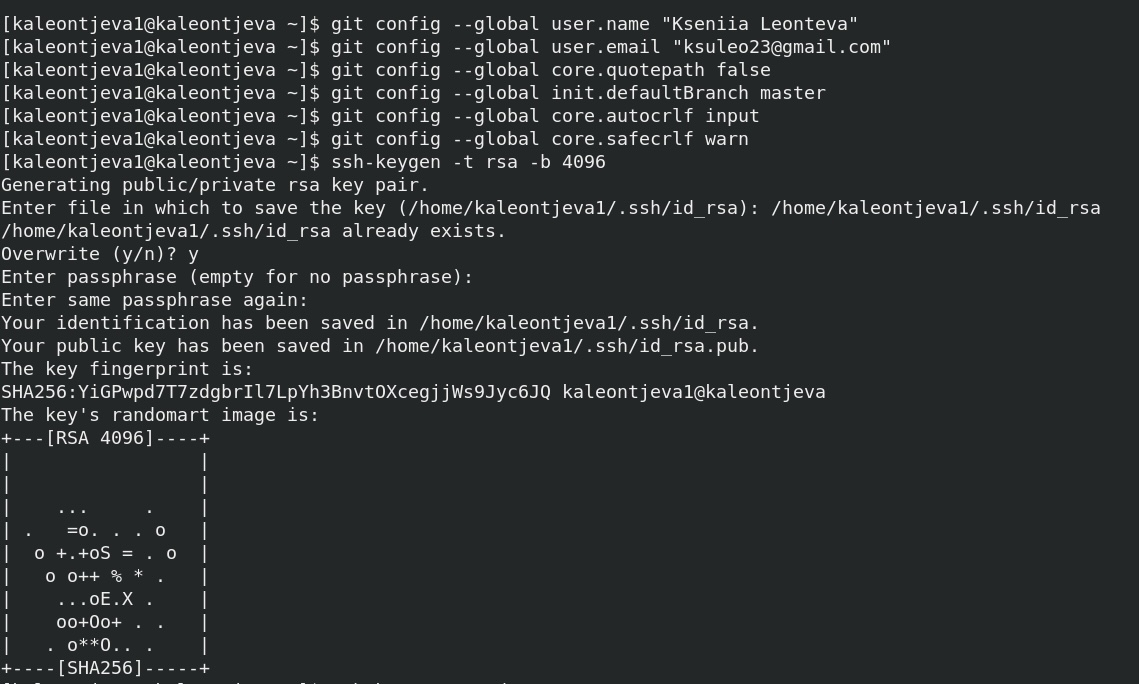


Рис. 3: Базовая настройка git и создание ключа ssh по алгоритму rsa

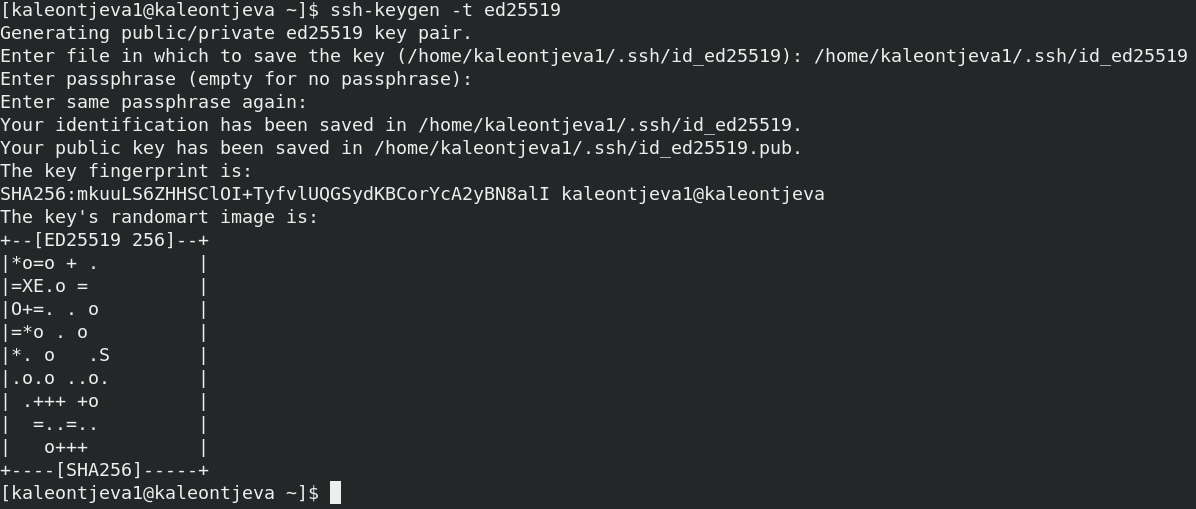


Рис. 4: Создание ключа ssh по алгоритму ed25519

Сгенерировали ключ pgp и выбрали из предложенных опций необходимые (рис. fig. 5 и рис. fig. 6).

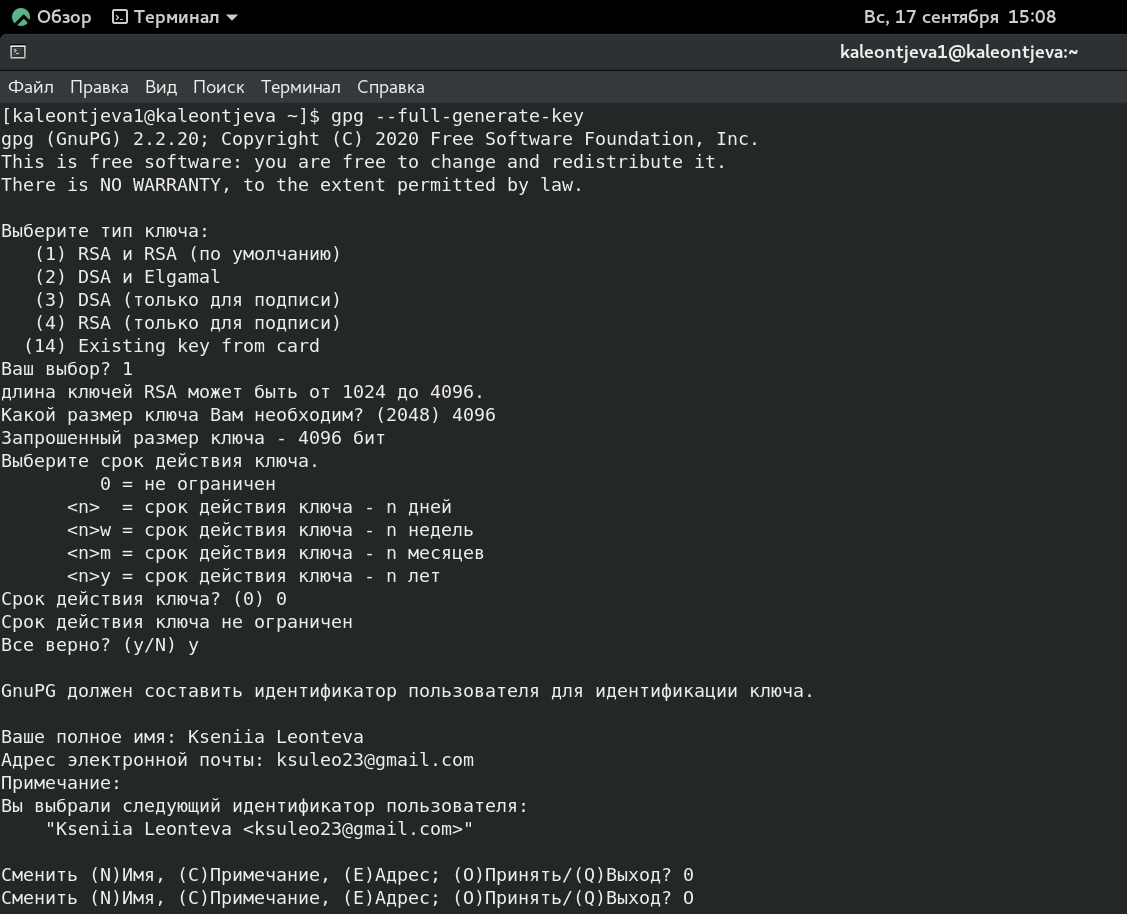


Рис. 5: Генерация ключа pgp

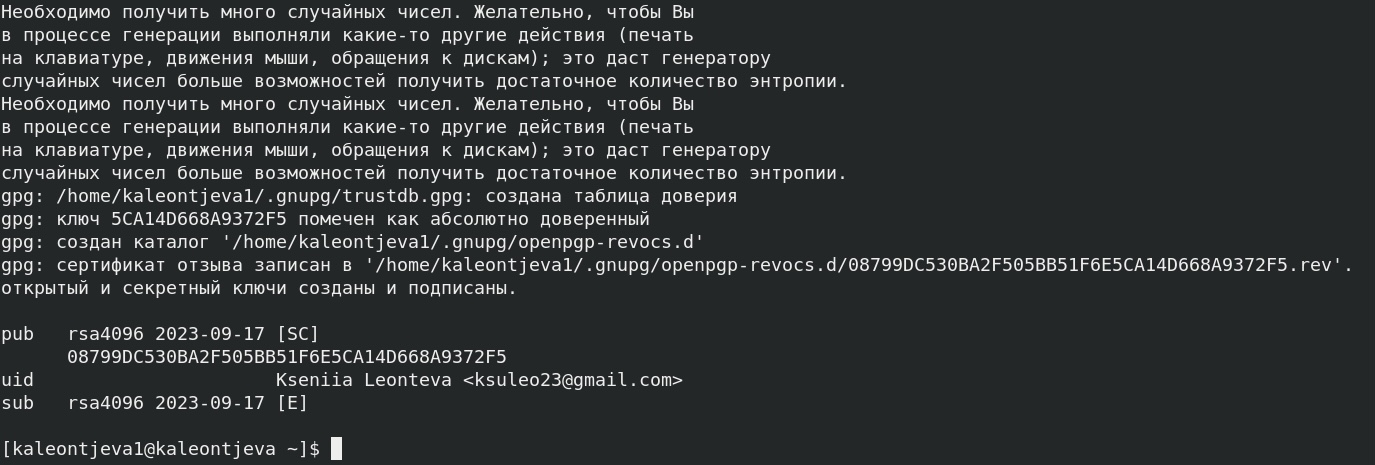


Рис. 6: Генерация ключа pgp

Выводим список ключей и копируем отпечаток приватного ключа (рис. fig. 7).

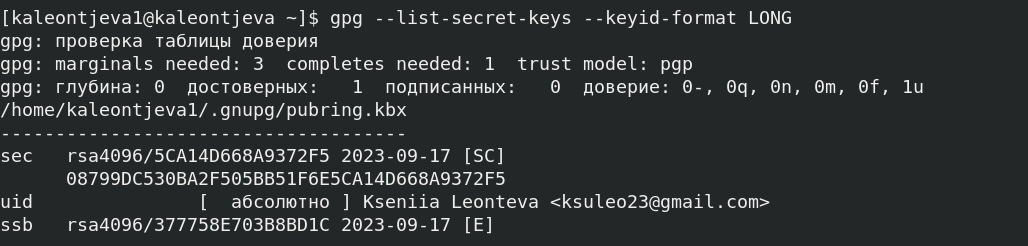


Рис. 7: Вывод списка ключей

Копируем сгенерированный PGP ключ в буфер обмена (рис. fig. 8).

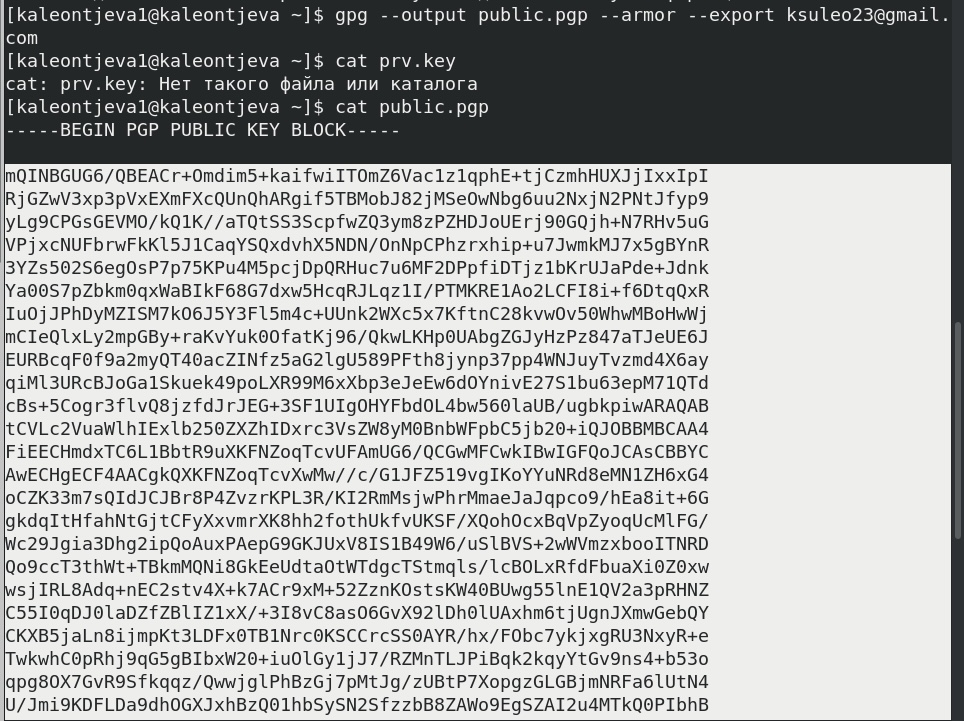


Рис. 8: Копирование сгенерированного ключа PGP

Затем переходим в настройки GitHub, в раздел keys и, нажав на кнопку New GPG key, вставляем полученный ключ в поле ввода (рис. fig. 9).

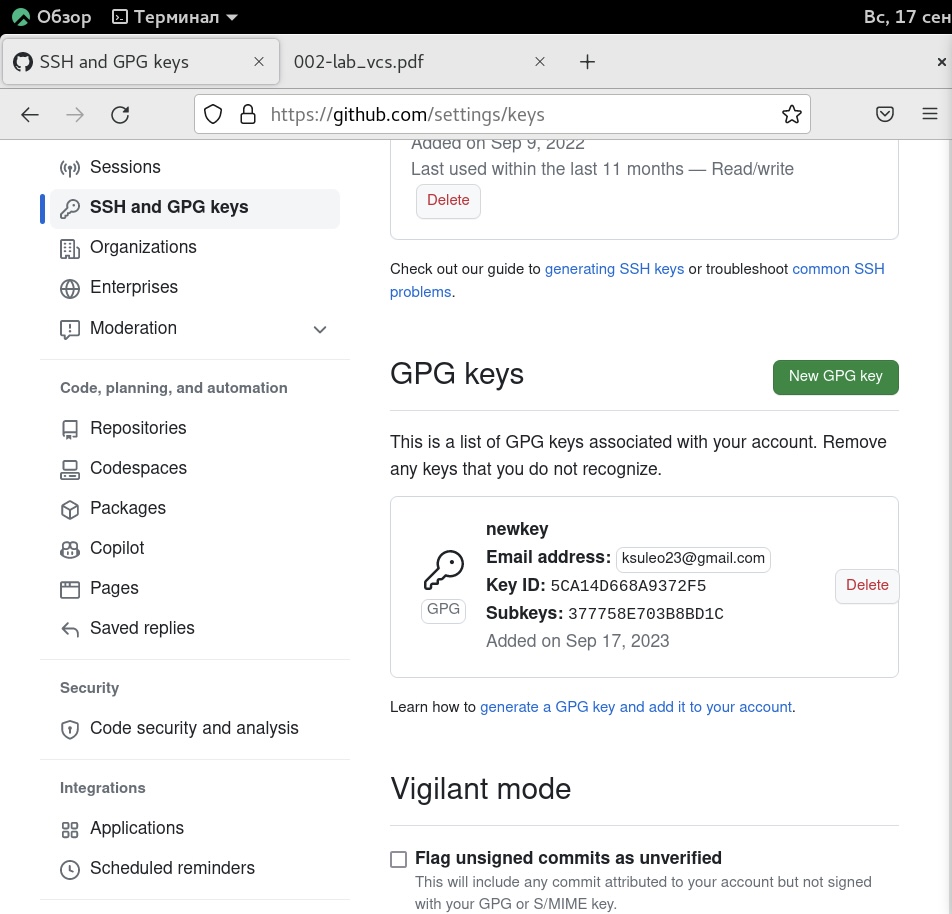


Рис. 9: Добавление gpg-ключа на GitHub

Настраиваем автоматические подписи коммитов git. Используя введенный email, указываем Git поменять его при подписи коммитов (рис. fig. 10).

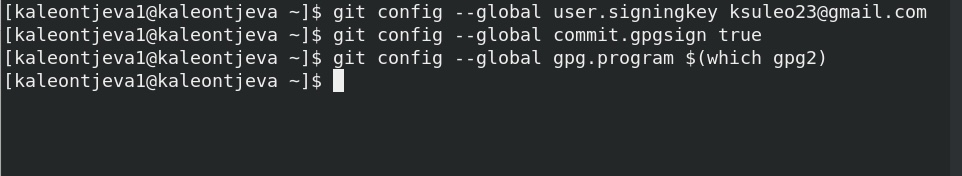


Рис. 10: Настройка автоматических подписей коммитов git

Устанавливаем и настраиваем gh. Отвечаем на наводящие вопросы утилиты и авторизуемся через браузер (рис. fig. 11 - рис. fig. 13).

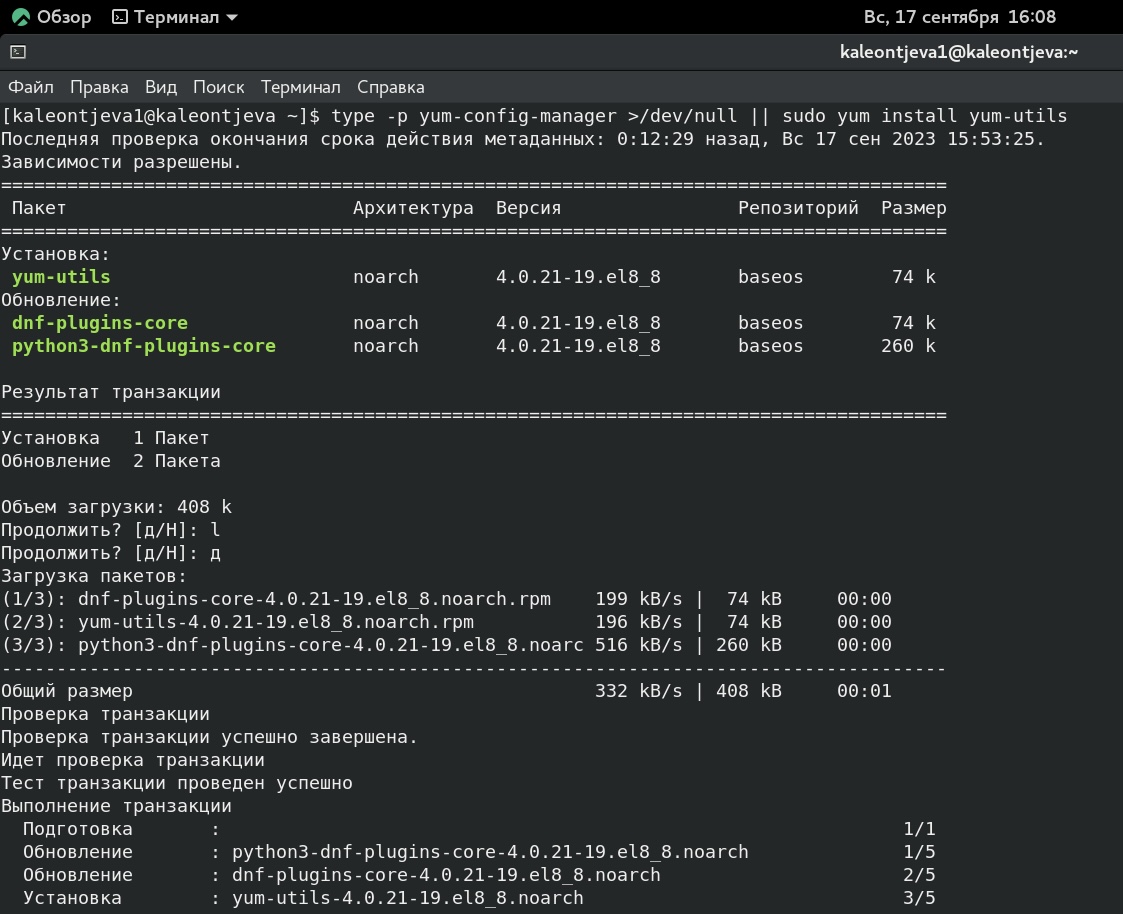


Рис. 11: Установка gh

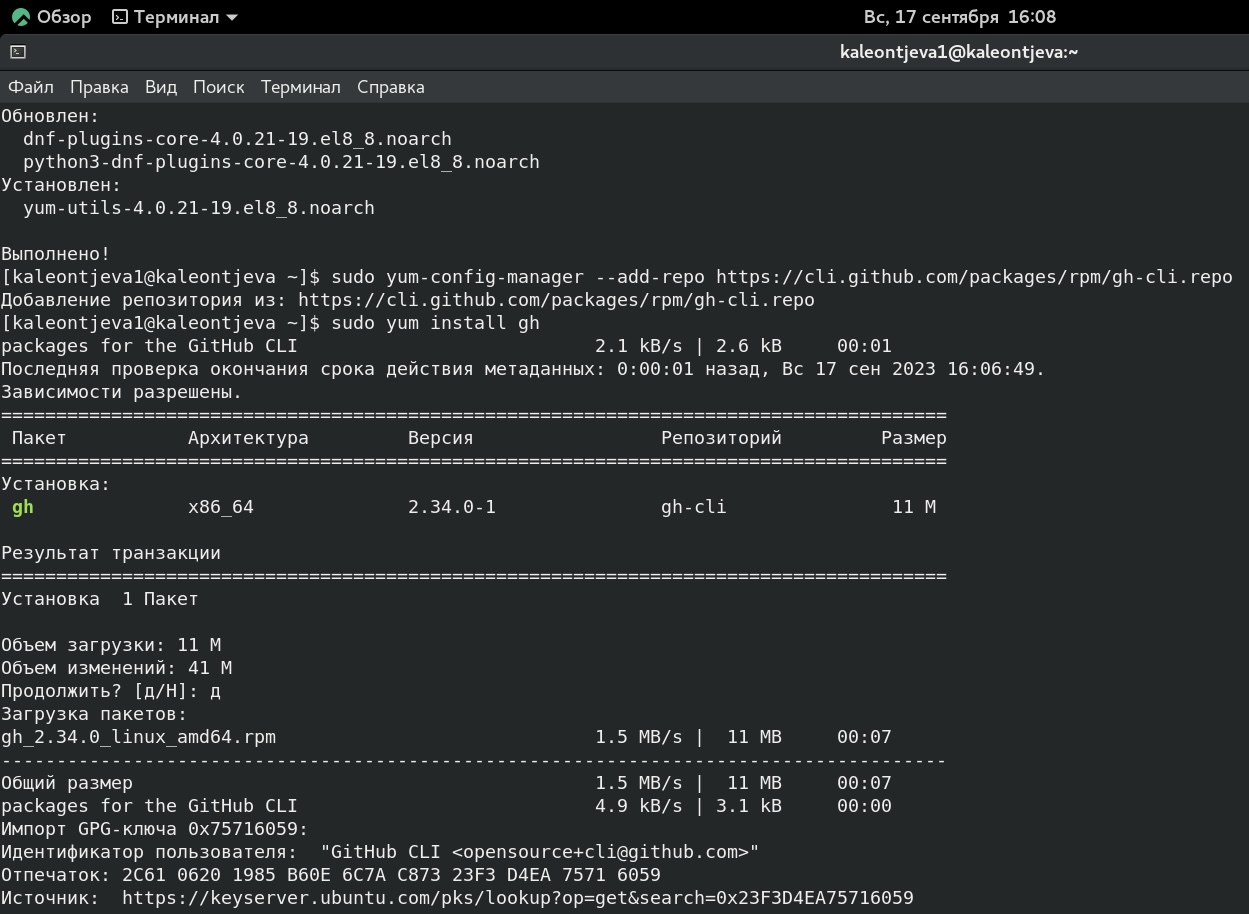


Рис. 12: Установка gh

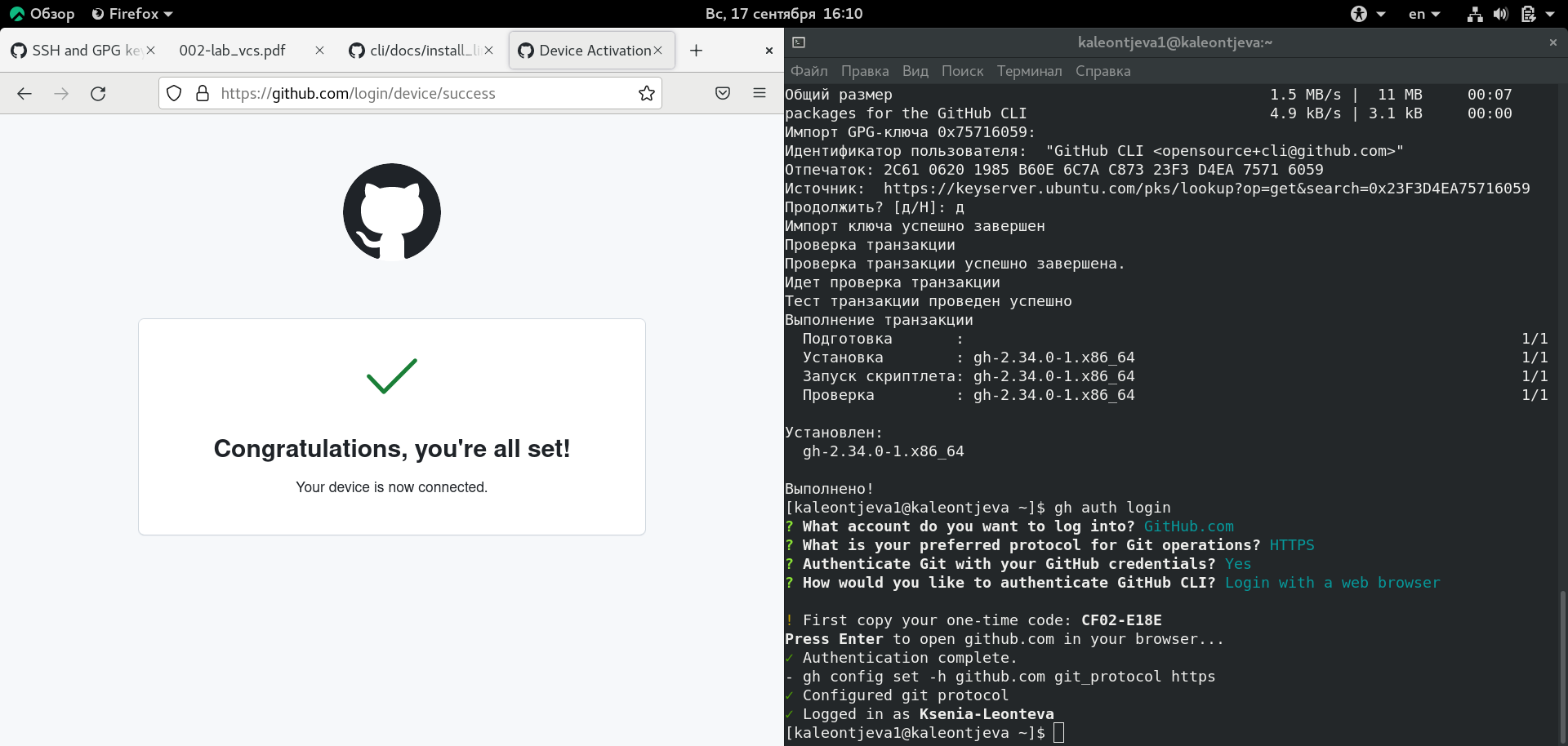


Рис. 13: Настройка gh и авторизация

Создаем репозиторий курса на основе шаблона (рис. fig. 14), параллельно сгенерировав и добавивив на GitHub еще один ssh-ключ (рис. fig. 15 и рис. fig. 16).

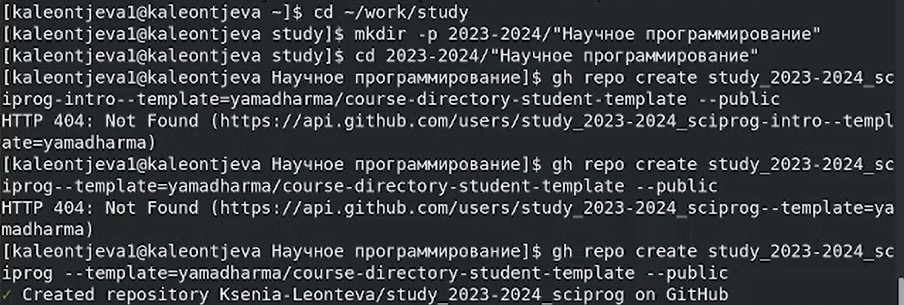


Рис. 14: Создание репозитория курса

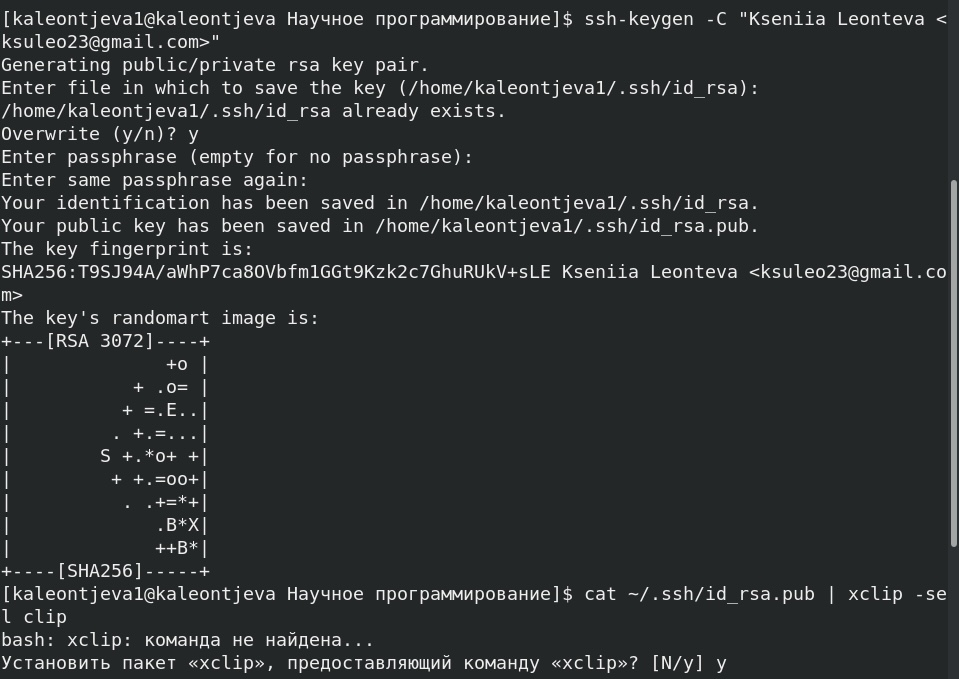


Рис. 15: Генерация ssh-ключа

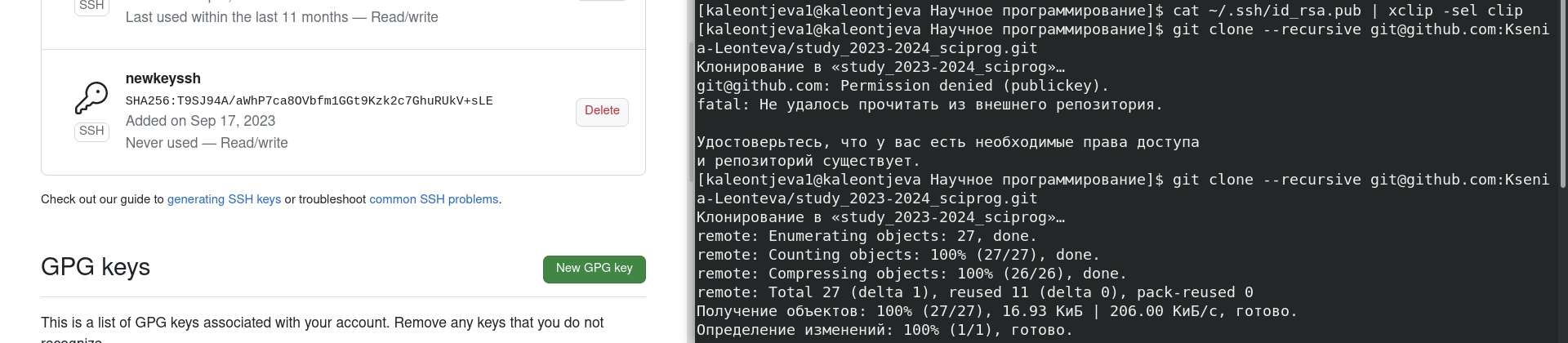


Рис. 16: Добавление ssh-ключа на GitHub и завершение создания репозитория курса

Настраиваем каталог курса (рис. fig. 17 и рис. fig. 18):

* переходим в каталог курса,
* удаляем лишние файлы,
* создаем необходимые каталоги,
* отправляем файлы на сервер.

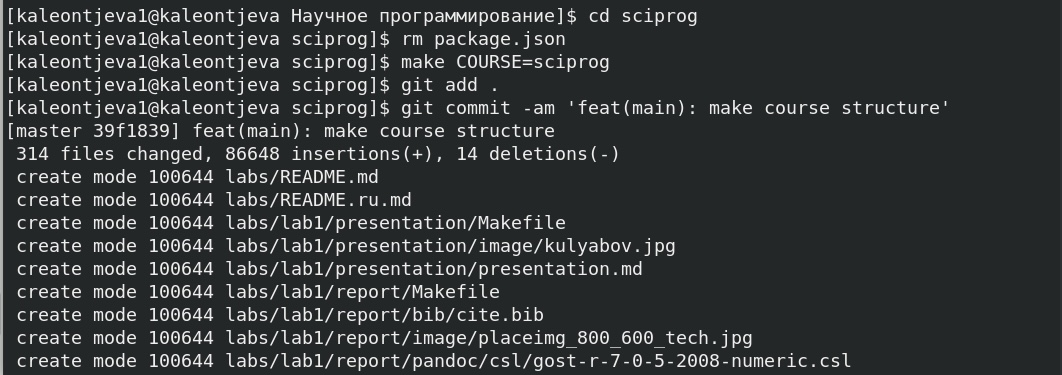


Рис. 17: Настройка каталога курса

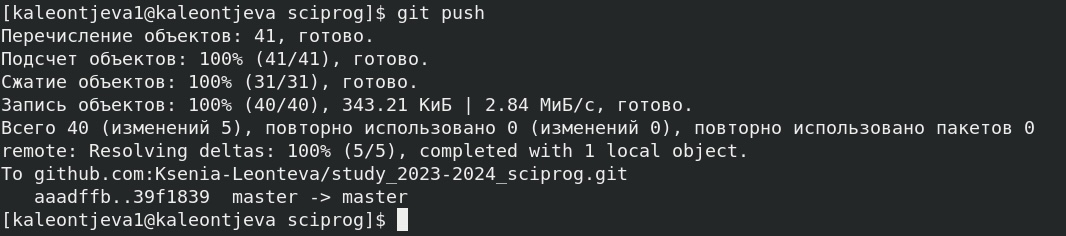


Рис. 18: Настройка каталога курса

# 4 Вывод

В ходе выполнения данной лабораторной работы я изучила идеалогию и применение средств контроля версий, а также освоила умения по работе с git.