Лабораторная работа №2

Управление версиями

Демидова Екатерина Алексеевна

Содержание

# 1 Цель работы

Целью работы является изучение идеологии и применения средств контроля версий, а также освоение умения по работе с git.

# 2 Задание

– Создать базовую конфигурацию для работы с git. – Создать ключ SSH. – Создать ключ PGP. – Настроить подписи git. – Зарегистрироваться на Github. – Создать локальный каталог для выполнения заданий по предмету.

# 3 Теоретическое введение

Cистема контроля версий (Version Control System, VCS) – это система, записывающая изменения в файл или набор файлов в течение времени и позволяющая вернуться позже к определенной версии [1]. Среди классических VCS наиболее известны CVS, Subversion, а среди распределён- ных — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд. Среди классических VCS наиболее известны CVS, Subversion, а среди распределён- ных — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд. | При выполнении работы была сипользована VCS Github [2]. История Git. В 2005 году компания, разрабатывающая систему контроля версий BitKeeper, порвала отношения с сообществом разработчиков ядра Linux. После этого сообщество приняло решение о разработке своей собственной системы контроля версий. Основными ценностями новой системы стали: полная децентрализация, скорость, простая архитектура, хорошая поддержка нелинейной разработки [1].

# 4 Выполнение лабораторной работы

1. Создаю учетную запись на https://github.com. и заполняю основные данные (рис. 1)

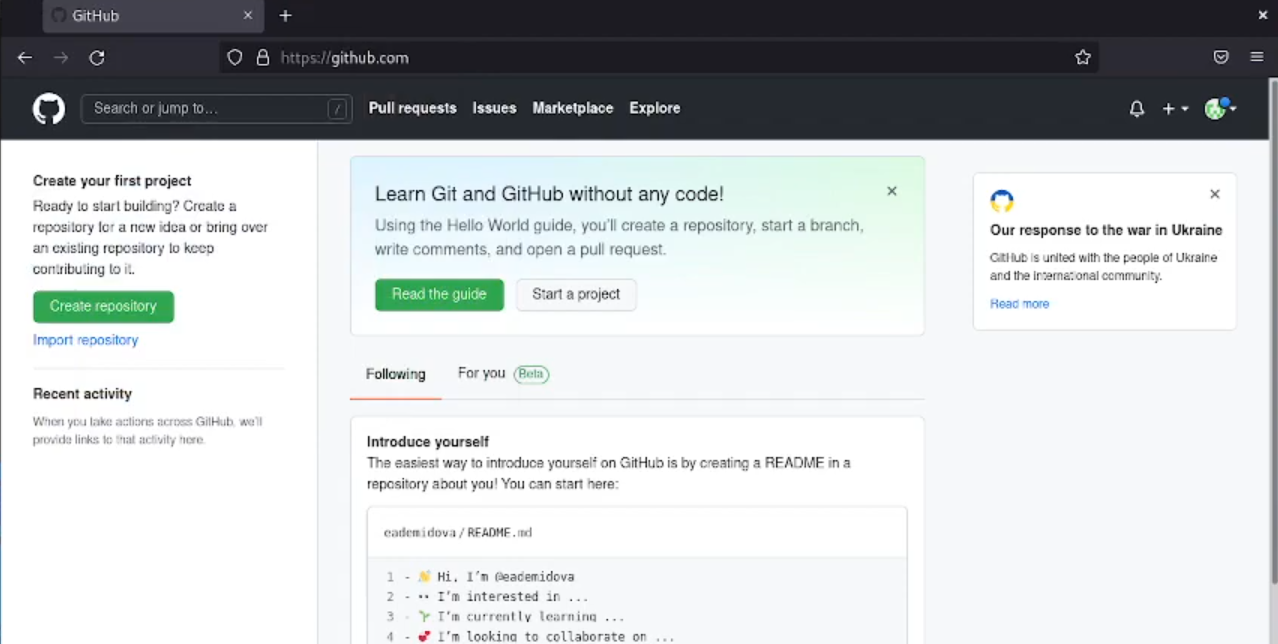


Рис. 1: Учетная запись

1. Устанавливаю программное обеспечение, а именно git-flow и gh (рис. 2)

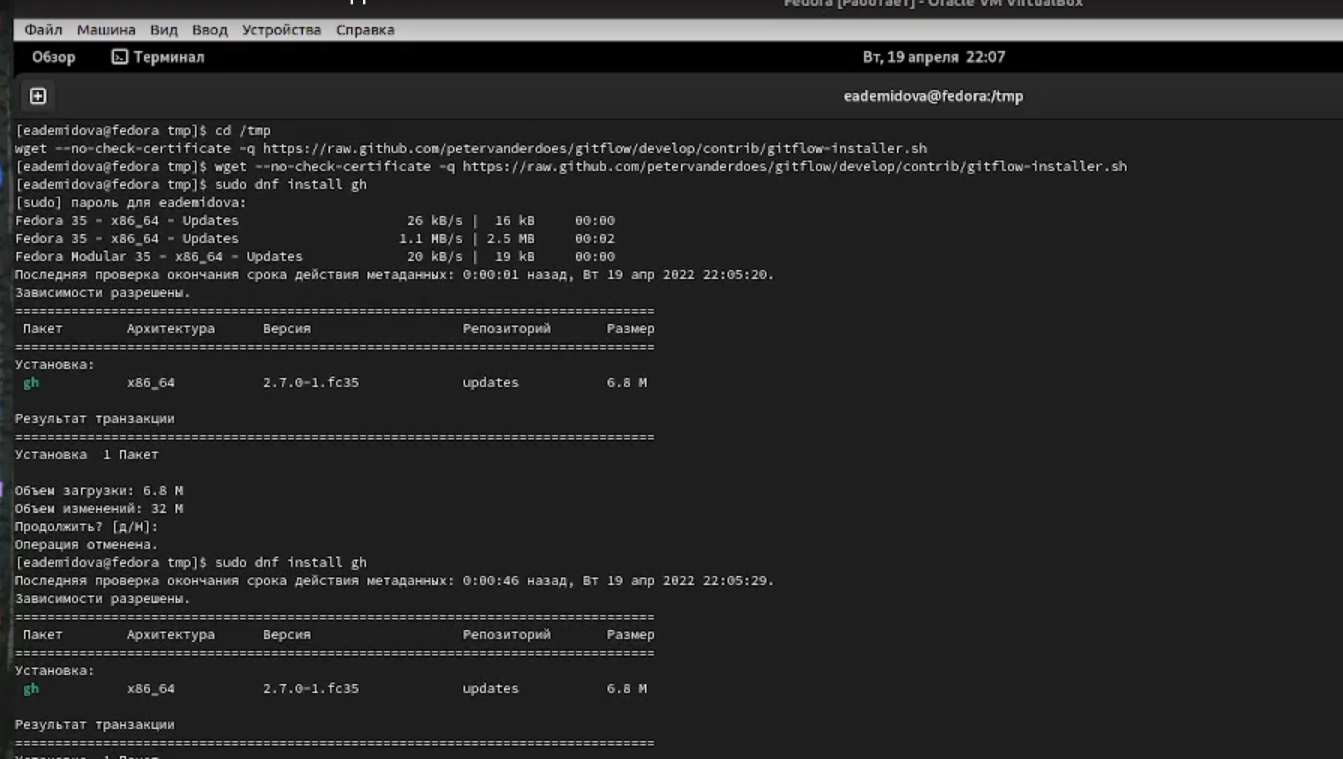


Рис. 2: Усатновка ПО

1. Провожу базовую настройку git. Задаю имя и email владельца репозитория, настраиваю utf-8 в выводе сообщений git,верификацию и подписание коммитов git (рис. 3)

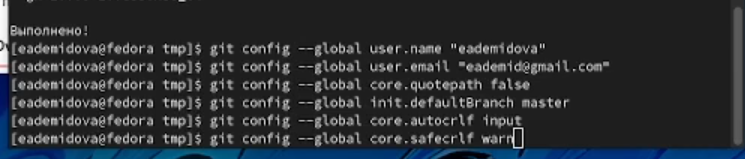


Рис. 3: Базовая настройка git

1. Создаю ключи ssh по алгоритму rsa и ed25519 (рис. 4)

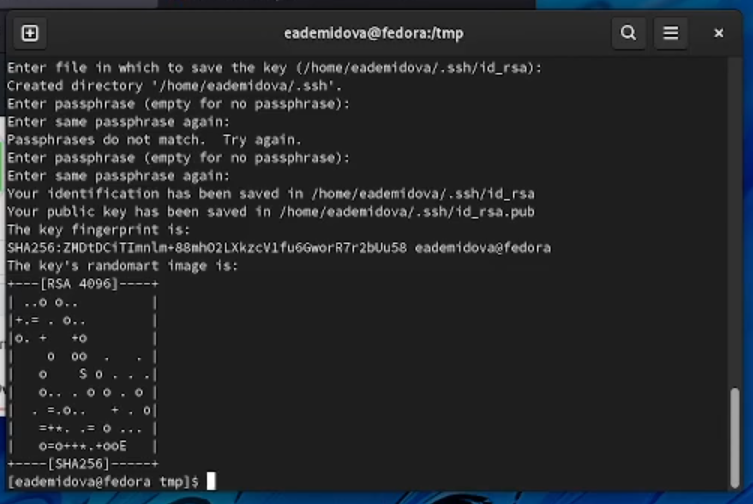


Рис. 4: Создание ключей ssh

1. Создаю ключи pgp, выбирая необходимые опции (рис. 5)

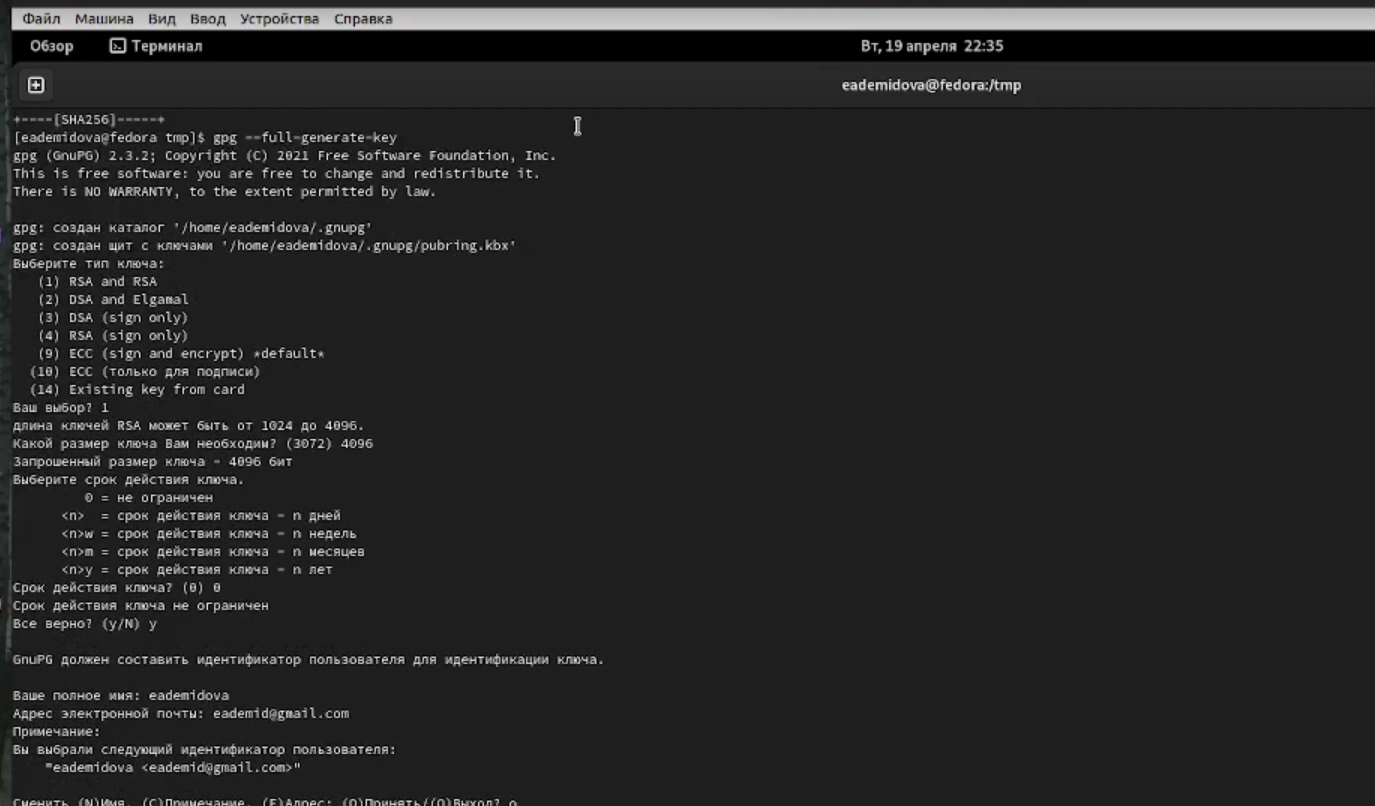


Рис. 5: Создание ключей pgp

1. Добавляю PGP ключ в GitHub. Для этого вводим список ключей и копируем отпечаток приватного ключа, а затем добавляем его в специальное поле ввода в настройках GitHub (рис. 6)



Рис. 6: Добавление ключа pgp

1. Настраиваем автоматические подписи коммитов git. Используя введёный email, укажим Git применять его при подписи коммитов (рис. 7)

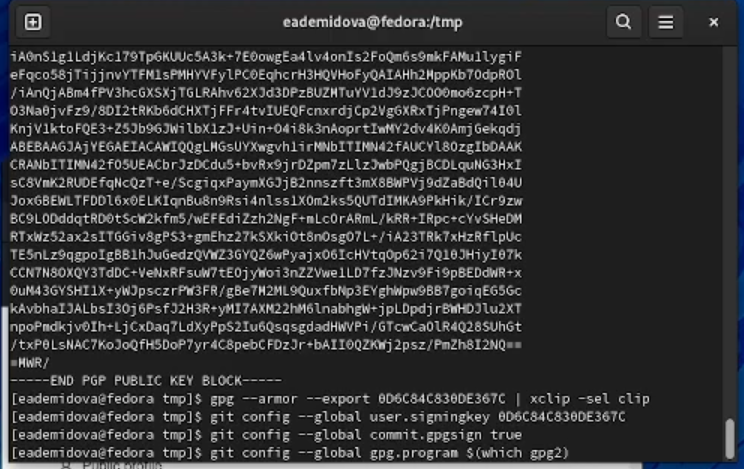


Рис. 7: Настройка коммитов

1. Настроим gh. Авторизуемся и ответм на несколько наводящих вопросов (рис. 8)

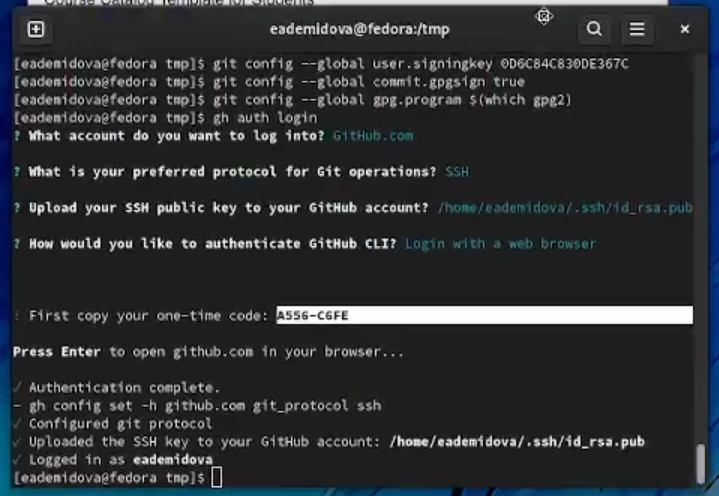


Рис. 8: Настройка gh

1. Создадим репозиторий курса на основе данного шаблона (рис. 9)

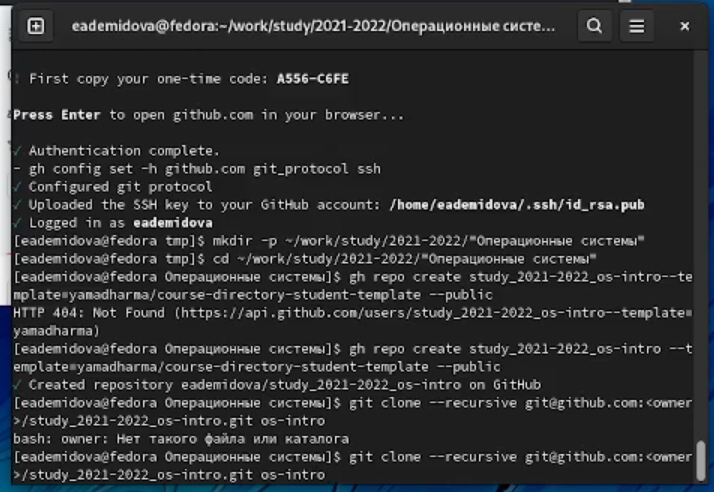


Рис. 9: Создание репозитория курса

1. Перейдем в катаог курса, удалим лишние файлы, создадим необходиые каталоги и отправим файлы на сервер (рис. 10)

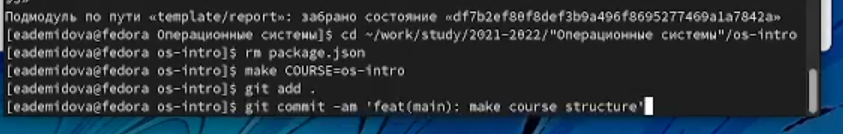


Рис. 10: Настройка каталога курса

Репозиторий git был успешно создан (рис. 11)

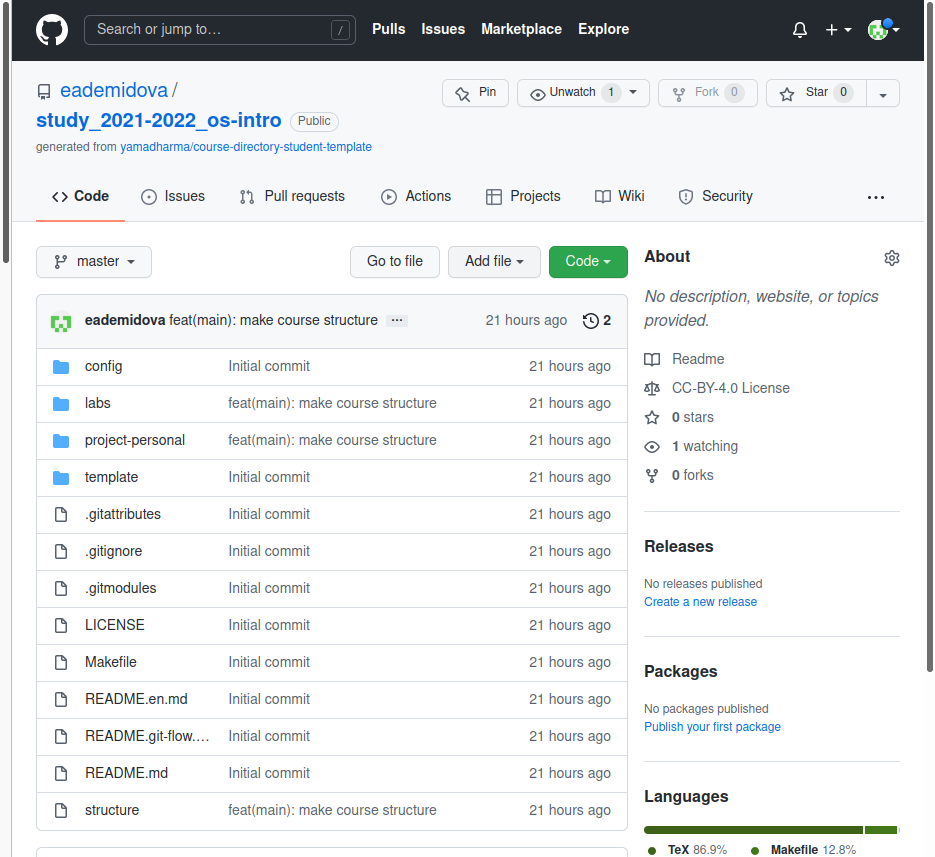


Рис. 11: Репозиторий git

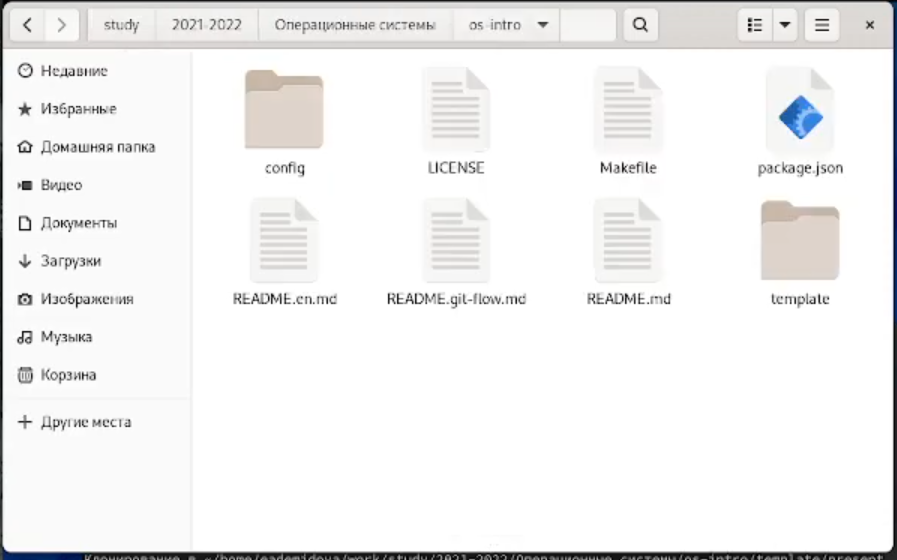


Рис. 12: Репозиторий на локальной машине

# 5 Контрольные вопросы

1. Системы контроля версий -VCS- это программное обеспечение, которое используется для облегчения работы с изменяющейся информацией, обычно - в проектах. Чаще всего используется при разработке, когда над одним проектом работает большое количество людей.
2. Репозиторий в системе контроля версий - Это удаленный репозиторий, в котором хранятся все файлы проекта commit - фиксирует изменения перед загрузкой файлов в систему контроля версий история хранит все изменения в проекте, и при необходимости позволяет откатиться в желаемое место рабочая копия - - это копия проекта на компьютере разработчика. Если другой член команды изменил проект, вам необходимо скачать новую версию проекта на свой компьютер.
3. 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. В децентрализованных системах у каждого из участников проекта есть полная копия проекта на своем компьютере, что делает его менее зависимым от сервера (Git).
4. Для начала необходимо создать и подключить удаленный репозиторий. Затем, поскольку никто, кроме вас, не изменяет проект, по мере изменения проекта отправляйте изменения на сервер, и нет необходимости загружать изменения.
5. 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.
6. Упрощение обмена информацией, ускорение разработки, устранение ошибок и недочетов при разработке.

* git init - инициализирует локальный репозиторий git add \* или add. - добавляет файлы в репозиторий git commit - версия фиксации git pull - загружает текущую версию проекта git push - отправляет измененный проект на сервер git checkout - позволяет переключаться между ветками git status - текущий статус проекта git branch - просмотреть доступные ветки git remote add - добавить удаленный репозиторий
* git push –all (push origin master/любой branch)

1. Ветви функций, также иногда называемые ветвями тем, используются для разработки новых функций, которые должны появиться в текущих или будущих выпусках.
2. Существуют временные и системные файлы, которые загромождают проект и не нужны. путь к ним можно добавить в файл .gitignore, тогда они не будут добавлены в проект

# 6 Выводы

В результате выполнения лабораторной работы была изучена идеология и применение средств контроля версий. А также освоены умения по работе с git, были обретены навыки создания ключей, репозитория, добавления и удаления файлов из репозитория.

# Список литературы

1. Статья о системах контроля версий [Электронный ресурс]. 2020. URL: <https://habr.com/ru/company/otus/blog/521290/>.

2. GitHub [Электронный ресурс]. 2021. URL: <https://github.com/>.