Отчёт по лабораторной работе 3

Елизавета Александровна Гайдамака

Содержание

# Цель работы

Целью данной работы является изучение идеологии и применения средств контроля версий и получение умений работы с git.

# 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

# Теоретическое введение

Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение

Хранилище версий - то, где хранятся все документы вместе с историей их изменения и другой служебной информацией. Коммит - зафиксированный набор изменений, который показывает, какие файлы изменились и что именно в них изменилось. История - список всех изменений. Рабочая копия - снимок одной версии проекта

Централизованные VCS: одно основное хранилище всего проекта; каждый пользователь копирует себе необходимые ему файлы из этого репозитория, изменяет и, затем, добавляет свои изменения обратно. Примеры: Subversion, CVS, TFS, VAULT, AccuRev. Децентрализованные VCS: у каждого пользователя свой вариант (возможно не один) репозитория, присутствует возможность добавлять и забирать изменения из любого репозитория. Пример: Git, Mercurial, Bazaar.

Порядок работы с общим хранилищем VCS:

* Клонировать репозиторий себе в гитхаб
* Клонировать репозиторий себе на устройство
* Внести изменения
* Добавить новую версию файлов на сервер

Git решает две задачи: хранить информацию о всех изменениях в коде, начиная с самой первой строчки, а вторая — обеспечение удобства командной работы над кодом.

Краткая характеристика команд git:

* git config - настройки
* git init - создание репозитория
* git add - добавление файлов в индекс
* git commit - коммит изменений
* git status - список измененных файлов
* git push - перенос изменений в главную ветку
* git rm - удаление файлов из индекса

Локальный репозиторий можно загрузить на гитхаб и работать с ним с помощью VCS, т.е. загружать новые версии, не теряя старые.

Ветка в Git это подвижный указатель на один из коммитов. Обычно ветка указывает на последний коммит в цепочке коммитов. Ветки нужны для того, чтобы программисты могли вести совместную работу над проектом и не мешать друг другу при этом.

Чтобы проигнорировать файлы при коммит, надо просто не добавлять их в коммит. Игнорируют те файлы, которые пользователь не хочет отправлять в репозиторий.

# Выполнение лабораторной работы

Cкачиваем и устанавливаем git flow и gh.

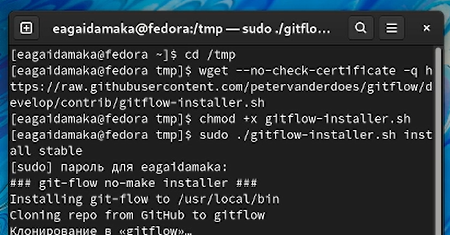


Рис.1



Рис.2

Настраиваем git: задаем имя владельца, настраиваем utf-8 в выводе сообщений git, зададим имя начальной ветки, параметры autpcrlf и safecrlf.

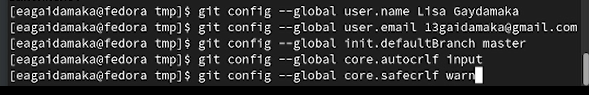


Рис.3

Создаем ключ SSH с помощью команды ssh-keygen

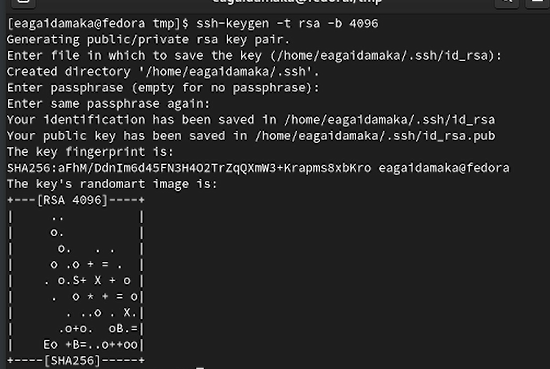


Рис.4

Генерируем pgp ключ и вводим свои данные.

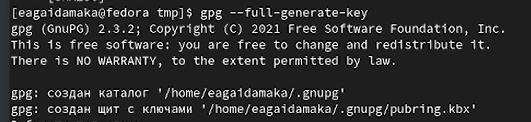


Рис.5

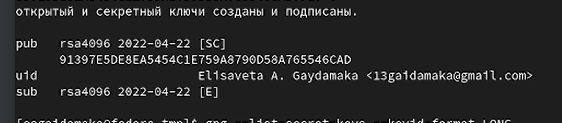


Рис.6

Добавляем ключ в github.

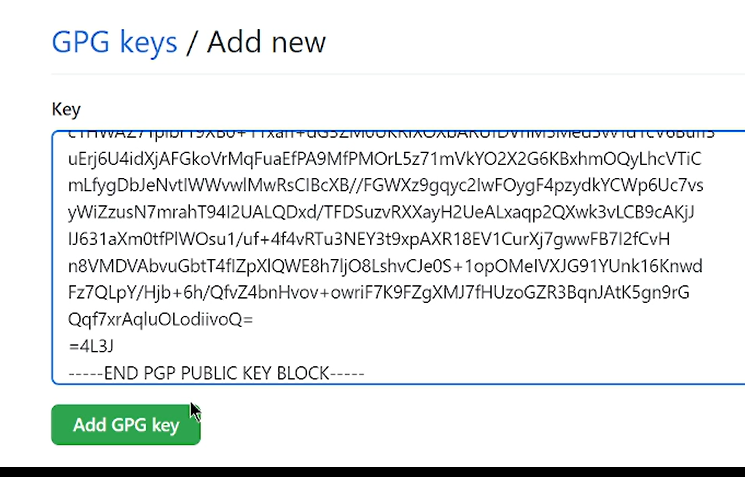


Рис.7

Настраиваем подписи git.

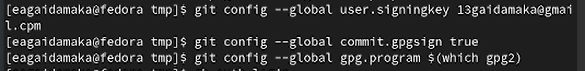


Рис.8

Авторизируемся в гитхабе на устройстве.

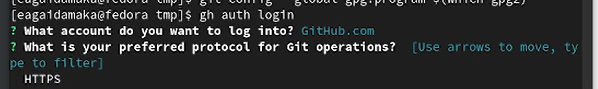


Рис.9

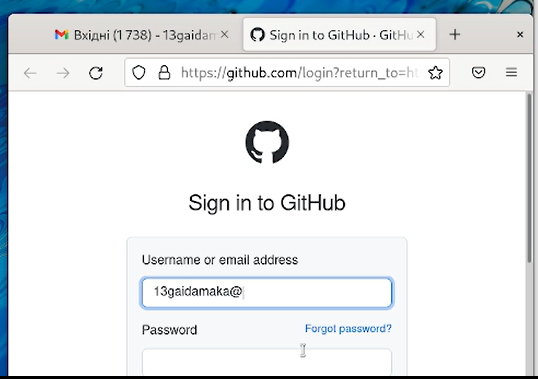


Рис.10

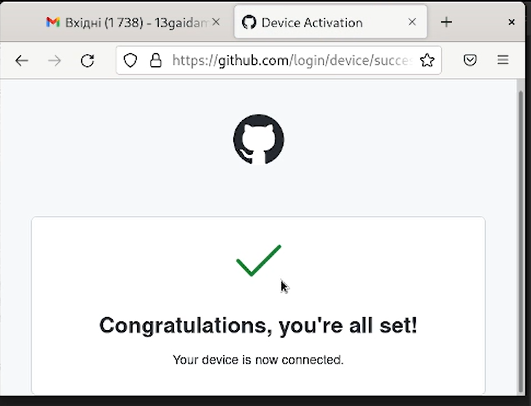


Рис.11

Создаем репозиторий на гитхабе с помощью template. Потом клонируем его себе на компьютер.

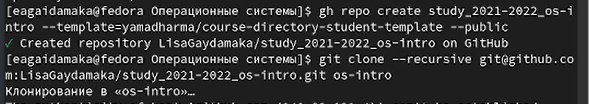


Рис.12

Вносим поправки в репозиторий на компьютере.

Рис.13

Рис.13

Добавляем файлы с поправками в коммит и отправляем на сервер.

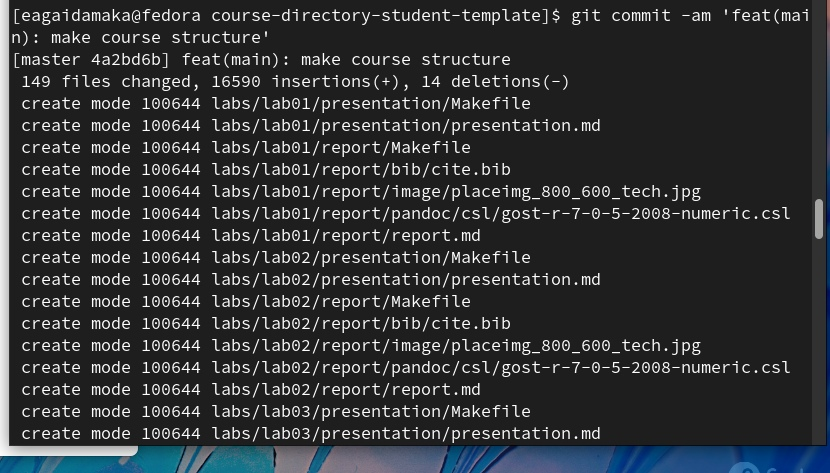


Рис.14

Проверяем, что новая версия загрузилась.

# Выводы

Благодаря данной работе я изучила идеологию и применение средств контроля версий и получила умения работы с git.