Российский университет дружбы народов

Факультет физико-математических и естественных наук

Направление прикладной математики и информатики

**Отчёт о лабораторной работе № 2. Управление версиями.**

Данько Д. И.

НПМбд-01-21

# Цель работы

– Изучить идеологию и применение средств контроля версий.

– Освоить умения по работе 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

# Выполнение

1. Настроим аккаунт на Github.

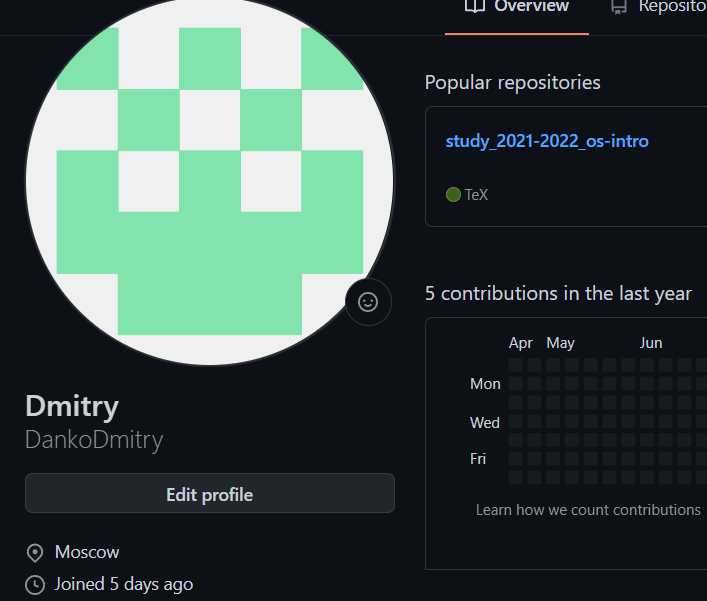


Рисунок 2.1

1. Установка Git-flow.



Рисунок 2.2

1. Создание SSH ключа.

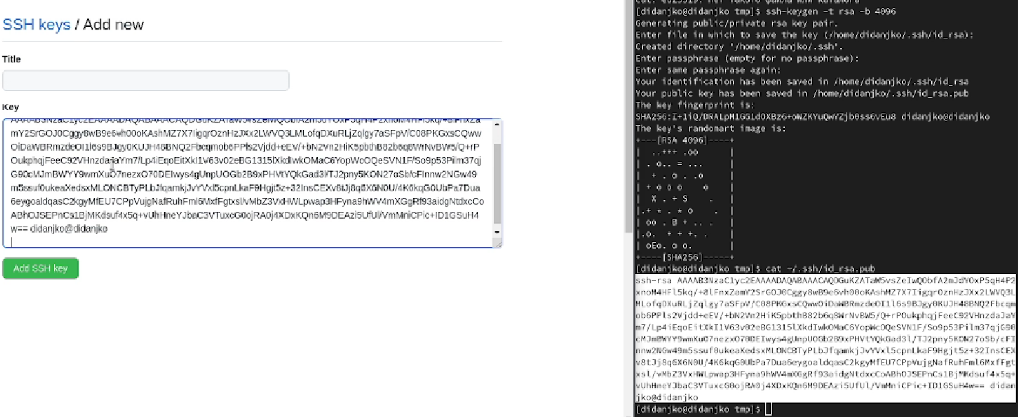


Рисунок 2.3

1. Создание PGP ключа.

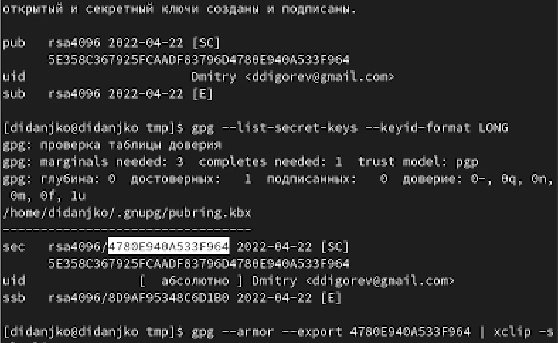


Рисунок 2.4

1. Логинимся на Git.

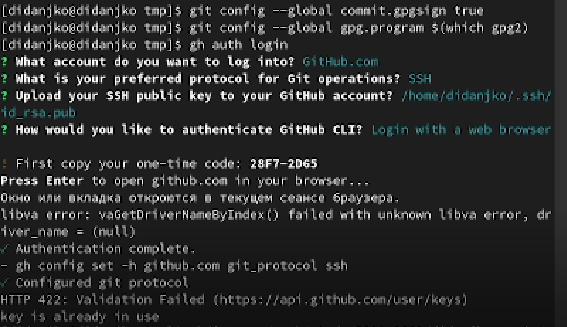


Рисунок 2.5

1. Копируем репозиторий.

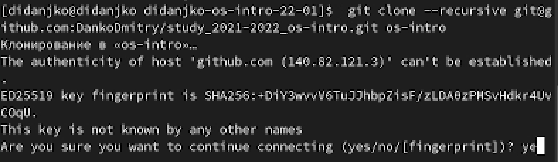


Рисунок 2.6

1. Удаляем не нужные файлы и создаём необходимые катологи.
2. Отправляем файлы на сервер.

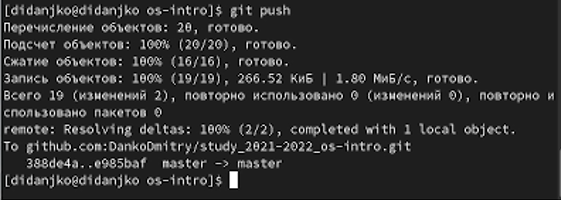


Рисунок 2.7

# Список команд

cd /tmp

wget --no-check-certificate -q https://raw.github.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh

chmod +x gitflow-installer.sh

sudo ./gitflow-installer.sh install stable

sudo dnf install gh

git config --global user.name didanjko

git config --global user.email ddigorev@gmail.com

git config --global core.quotepath false

git config --global init.defaultBranch master

git config --global core.autocrlf input

git config --global core.safecrlf warn

ssh-keygen -t rsa -b 4096

gpg --full-generate-key

gpg --list-secret-keys --keyid-format LONG

gpg --armor --export <…………….> | xclip -sel clip

git config --global user.signingkey <…………………>

git config --global commit.gpgsign true

git config --global gpg.program $(which gpg2)

gh auth login

mkdir -p ~/work/study/2021-2022/study\_2021-2022\_os-intro

cd ~/work/study/2021-2022/study\_2021-2022\_os-intro

gh repo create study\_2021-2022\_os-intro --template=yamadharma/course-directory-student-template --public

git clone --recursive git@github.com:/DankoDmitry/study\_2021-2022\_os-intro.git os-intro

cd ~/work/study/2021-2022/study\_2021-2022\_os-intro/os-intro

rm package.json

make COURSE=os-intro

git add .

git commit -am 'feat(main): make course structure'

git push

# Контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом.

1. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

Хранилище – место хранения всех данных, основного дерева, веток, коммитов.

Коммит – синоним/копия основного дерева.

История – история всех изменений.

Рабочая копия - текущее состояние файлов проекта, основанное на версии из хранилища.

1. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

Централизованные:

• Простота использования.

• Вся история — всегда в едином общем хранилище.

• Нужно подключение к сети.

• Резервное копирование нужно только одному хранилищу.

• Удобство разделения прав доступа к хранилищу.

• Почти все изменения навсегда попадают в общее хранилище.

Распределенные:

• Двухфазный commit: 1) запись в локальную историю;

2) пересылка изменений другим.

• Подключение к сети не нужно.

• Локальные хранилища могут служить резервными копиями.

• Локальное хранилище контролирует его владелец,

• но общее — администратор.

• Возможна правка локальной истории перед отправкой на сервер.

1. Опишите действия с VCS при единоличной работе с хранилищем.

Сделать рабочую копию - ветку.

Внести изменения.

Слить с основным репозиторием – деревом.

1. Опишите порядок работы с общим хранилищем VCS.

Сделать рабочую копию - ветку.

Внести изменения.

Слить с основным репозиторием – деревом.

1. Каковы основные задачи, решаемые инструментальным средством git?

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

7. Назовите и дайте краткую характеристику командам git.

8. Приведите примеры использования при работе с локальным и удалённым репозиториями.

9. Что такое и зачем могут быть нужны ветви (branches)?

10. Как и зачем можно игнорировать некоторые файлы при commit?

# Выводы

# Выполняя данную лабораторную работу, несколько разобрались с работой Git и Github в частности, создали свой репозиторий по образцу.