Управление версиями

13.06.2022

# 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 лабораторной работы

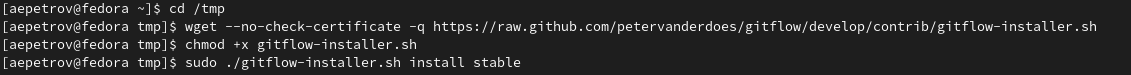
## Создание учетной записи на github.com

* Создание учентой записи в моем случае не требуется, поэтому перейдем к следующему заданию.

## Установка Программного обесепечения

### Установка git-flow

* Установим git-flow согласно указаниям (рис. [-@fig:001])



1.Установка git-flow в терминале

### Установка gh

* Установим gh(рис. [-@fig:002])

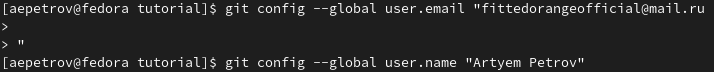
2.Установка gh

2.Установка gh

## Базовая настройка git

### Зададим имя и email(рис. [-@fig:003]):

1. git config --global user.name "Artyem Petrov"  
2. git config --global user.email"fittedorangeofficial@mail.ru"



3.Установление почты и имени для git

### Настроим utf-8 в выводе сообщений git(рис. [-@fig:004]):

1. git config --global core.quotepath false

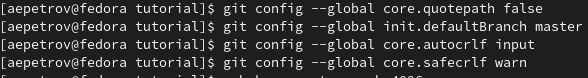
### Настроим верификацию и подписание коммитов(1) и задаим имя начальной ветке(2)(рис. [-@fig:004]):

* Настроим верификацию и подписание коммитов:

1. git config --global init.defaultBranch master

* Параметры autocrlf и safecrlf:

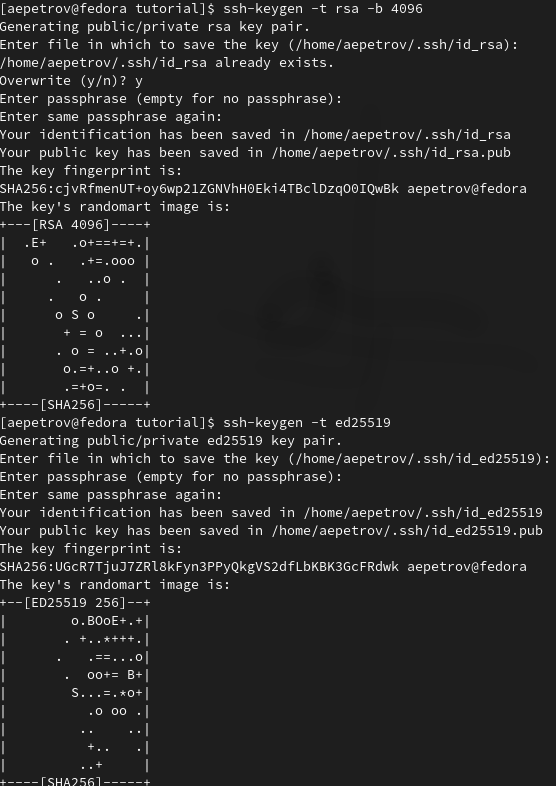
1. git config --global core.autocrlf input  
2. git config --global core.safecrlf warn



4.Установление почты и имени для git

## Создание ключей ssh по алгоритмам rsa & ed25519(рис. [-@fig:005])

1. ssh-keygen -t rsa -b 4096  
2. ssh-keygen -t ed25519



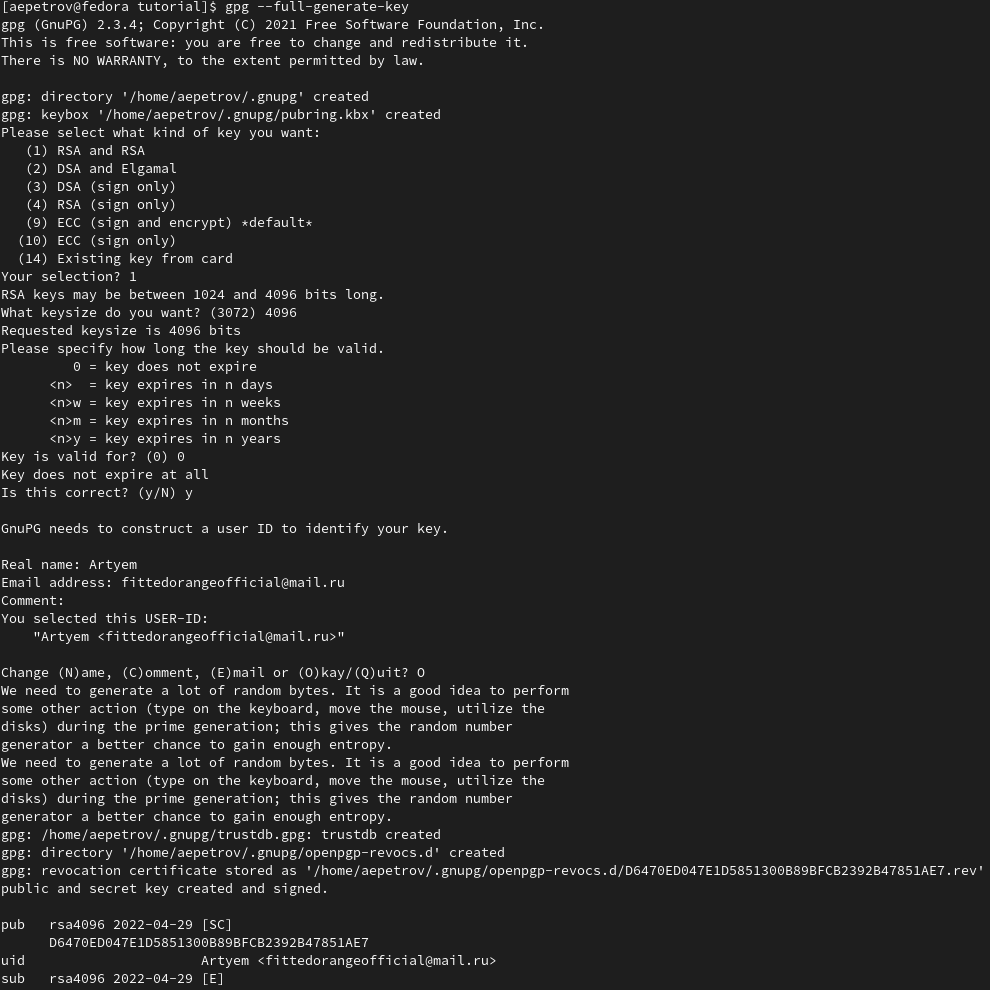
5.Генерация ключей по вышеописанным алгоритмам

## Создание ключа pgp(рис. [-@fig:006])

* Генерируем ключ:

1. gpg --full-generate-key

* Выбираем следующим образом: RSA & RSA, 4096, 0, Artyem, fittedorangeofficial@mail.ru.



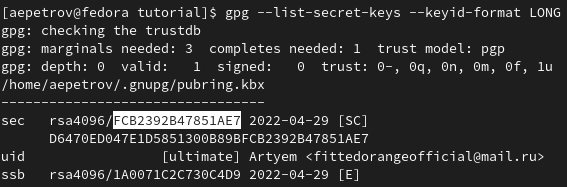
6.Создание ключа pgp

## Добавление PGP ключа в GitHub

* Выводи список ключей и копируем отпечаток приватного ключа(рис. [-@fig:007]):

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

* Отпечаток ключа — это последовательность байтов, используемая для идентификации более длинного, по сравнению с самим отпечатком ключа.
* Формат строки: sec Алгоритм/Отпечаток\_ключа Дата\_создания [Флаги] [Годен\_до] ID\_ключа Ключен выделен на рис. [-@fig:007]:



7.Нахождение ключа pgp

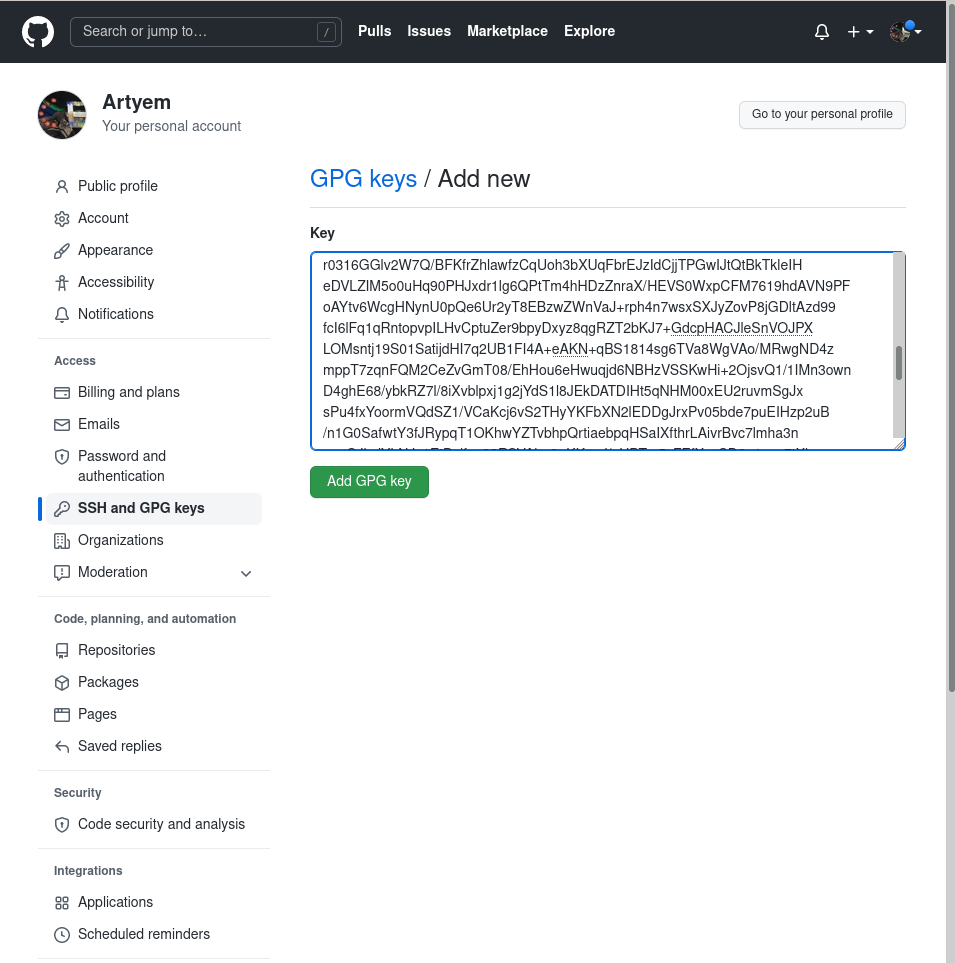
* Cкопируйте ваш сгенерированный PGP ключ в буфер обмена(рис. [-@fig:008]):

1. gpg --armor --export <PGP Fingerprint> | xclip -sel clip

8.Копирование ключа pgp

8.Копирование ключа pgp

– Перейдем в настройки GitHub (https://github.com/settings/keys), нажмем на кнопку New GPG key и вставьте полученный ключ в поле ввода(рис. [-@fig:009])

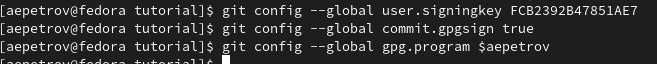


9.Копирование ключа pgp

## Настройка автоматических подписей коммитов git

* Используя введёный email, укажим Git применять его при подписи коммитов(рис. [-@fig:010]):

1. git config --global user.signingkey <PGP Fingerprint>  
2. git config --global commit.gpgsign true  
3. git config --global gpg.program $(which gpg2)



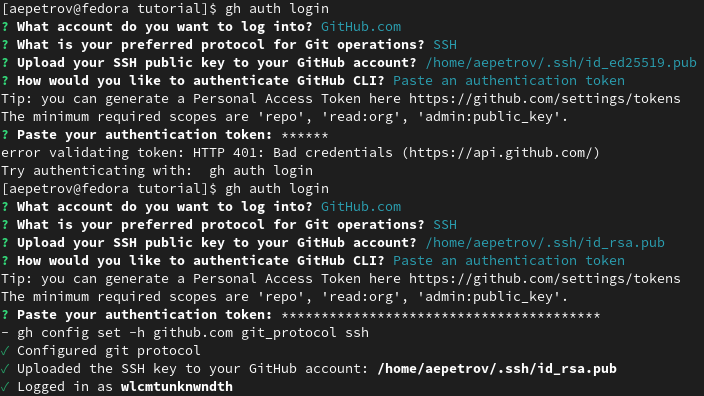
10.Настройка автоматических подписей коммитов git

## Настройка gh.

* Для начала авторизуемся(рис. [-@fig:011]):

gh auth login

* Утилита задаст несколько вопрос и попросит токен в конце аутентификации, который можно создать в настройках разработчика(https://github.com/settings/tokens)[link]. Если токен введен правильно, то вы авторизируетесь.



11.Настройка автоматических подписей коммитов git

## Шаблон для рабочего пространства

* Шаблон находится по ссылке: (https://github.com/yamadharma/course-directory-student-template)

### Создание репозитория на основе шаблона

* Создадим необходимую директорию и перейдем в нее(рис. [-@fig:012]):

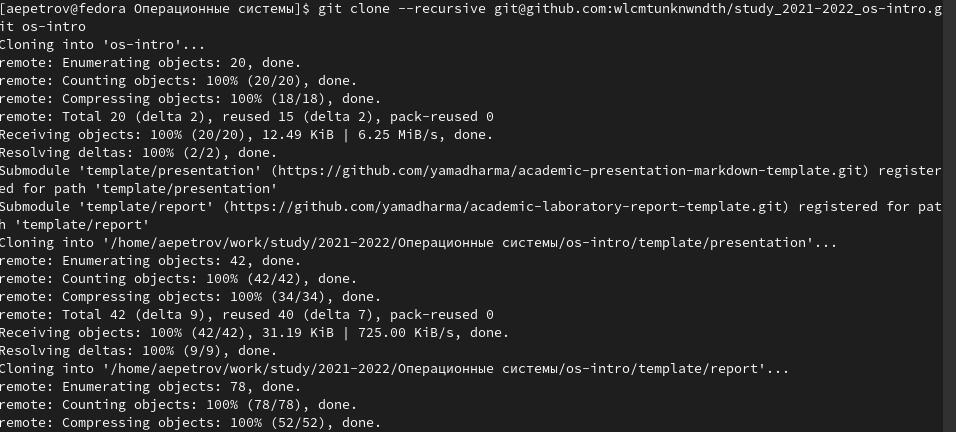
1. mkdir -p ~/work/study/2021-2022/"Операционные системы"  
2. cd ~/work/study/2021-2022/"Операционные системы"

* Создадим репозиторий в нашей директории(рис. [-@fig:012]):

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

* Скопируем содержимое репозитория преподавателя к нам в репозиторий(рис. [-@fig:012]):

1. git clone --recursive  
2. git@github.com:<owner>/study\_2021-2022\_os-intro.git os-intro

### Настройка каталога курса

* Перейдем в каталог курса(рис. [-@fig:013])

cd ~/work/study/2021-2022/"Операционные системы"/os-intro

* Удалим лишние файлы(рис. [-@fig:013])

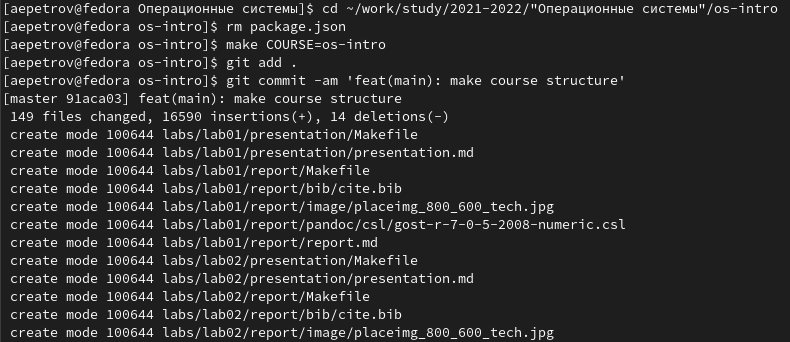
rm package.json

* Создадим необходимые каталоги(рис. [-@fig:013]):

make COURSE=os-intro

* После отправим файлы на сервер(рис. [-@fig:013]):

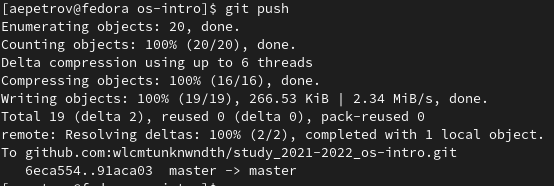
1. git add .  
2. git commit -am 'feat(main): make course structure'



13.Настройка каталога курса

* Окончательно “толкнем” их на сервер(рис. [-@fig:014]):

git push



14.Отправка на сервер

# Выводы

Благодаря данной работе я научился пользоваться системой контроля версий git, github, gitflow и т.д.