Отчет по лабораторной работе 2

Управление версиями

Максим Сергеевич Белов

Содержание

[Цель работы 1](#_Toc102076990)

[Задание 1](#_Toc102076991)

[Теоретическое введение 2](#_Toc102076992)

[Выполнение лабораторной работы 2](#_Toc102076993)

[Переходим к базовой настройке git: 3](#_Toc102076994)

[Создание ssh ключа: 4](#_Toc102076995)

[Создание gpg ключа: 5](#_Toc102076996)

[Добавление GPG ключа в GitHub: 5](#_Toc102076997)

[Настройка автоматических подписей коммитов git: 6](#_Toc102076998)

[Настройка gh: 7](#_Toc102076999)

[Сознание репозитория курса на основе шаблона и настройка каталога курса: 7](#_Toc102077000)

[Выводы 8](#_Toc102077001)

# 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.

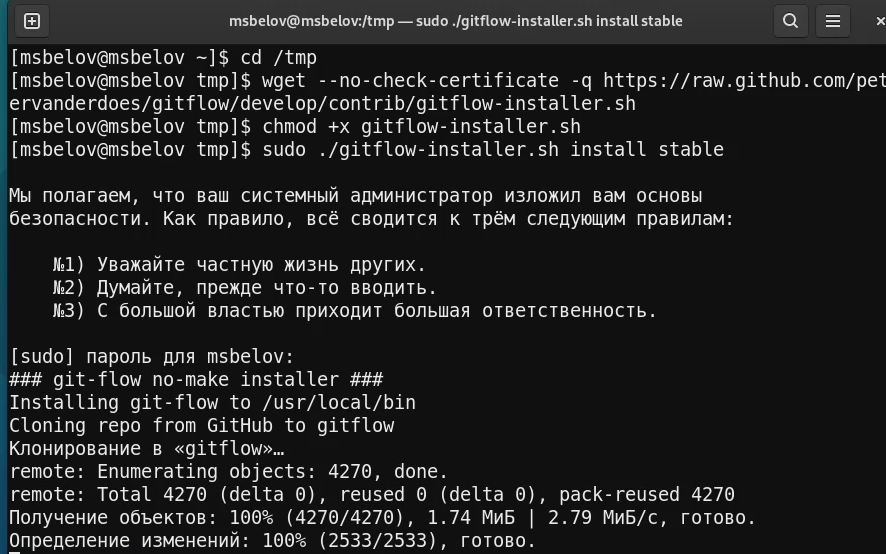
– Создать локальный каталог для выполнения заданий по предмету.

# Теоретическое введение

| Команды | Описание |
| --- | --- |
| git init | Создание основного дерева репозитория |
| git pull | Получение обновлений (изменений) текущего дерева из центрального репозитория |
| git push | Отправка всех произведённых изменений локального дерева в центральный репозиторий программ |
| git status | Просмотр списка изменённых файлов в текущей директории |
| git diff | Просмотр текущих изменений |
| git add . | Добавить все изменённые и/или созданные файлы и/или каталоги |
| git rm имена\_файлов | Удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории) |
| git commit | Сохранить добавленные изменения с внесением комментария через встроенный редактор |
| git checkout имя\_ветки | Переключение на некоторую ветку |

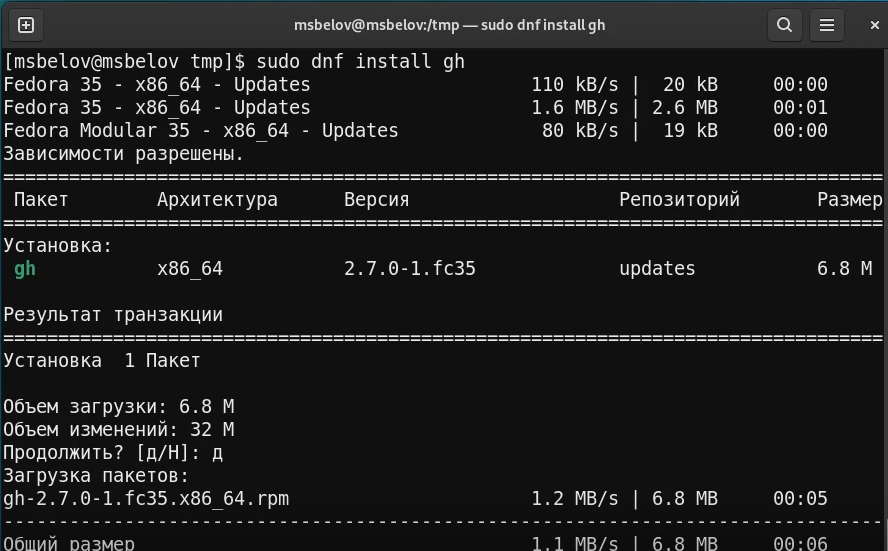
# Выполнение лабораторной работы

Аккаунт github у меня уже имелся поэтому я сразу перешёл к установке git-flow:



Установка git-flow в Fedora Linux

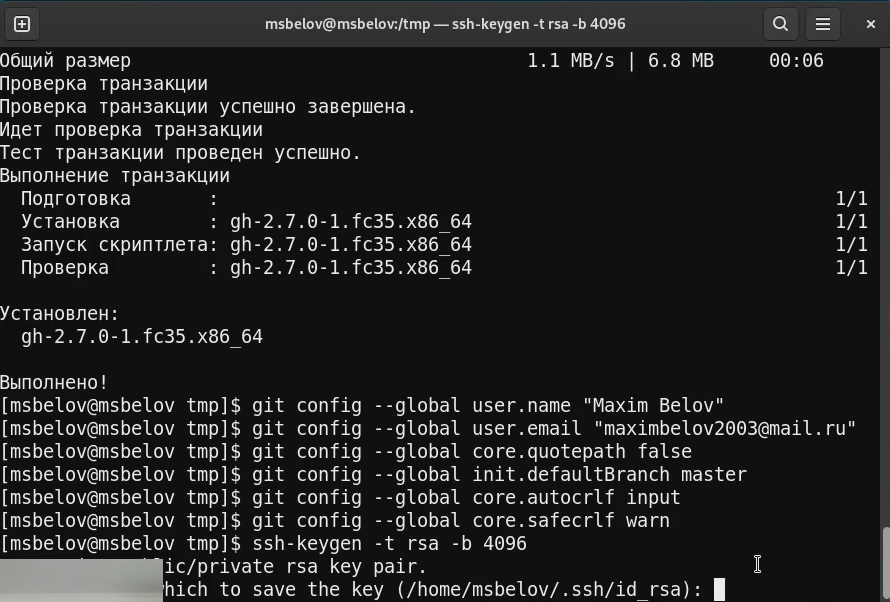
Устанавливаем gh в Fedora Linux с помощью команды sudo dnf install gh:



Установка gh

## Переходим к базовой настройке git:

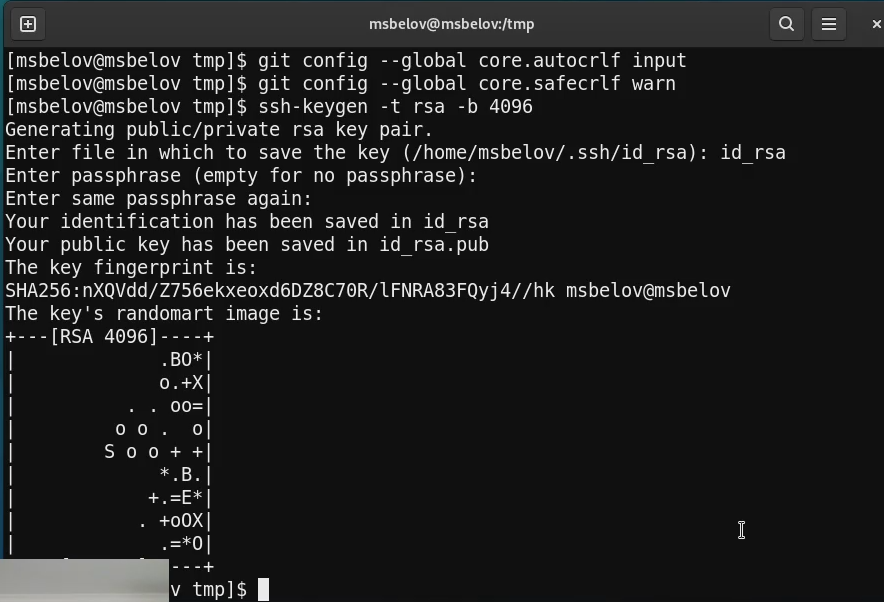
* Зададим имя и email владельца репозитория
* Настроим utf-8 в выводе сообщений git
* Настроим верификацию и подписание коммитов git.
* Зададим имя начальной ветки (будем называть её master)
* Параметр autocrlf
* Параметр safecrlf



Базовая настройка git

## Создание ssh ключа:

Создадим ключ ssh по алгоритму rsa (ssh-keygen -t rsa -b 4096)



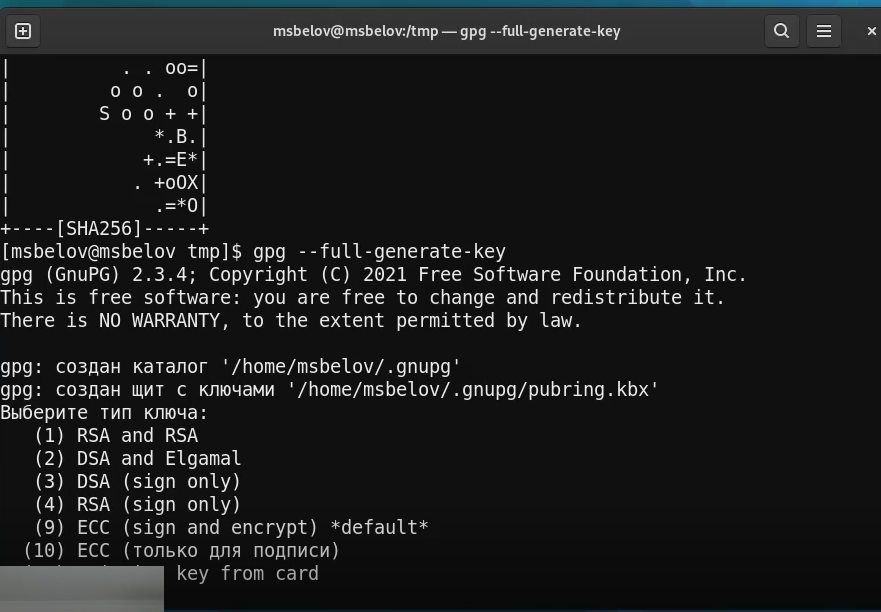
Создание ключа ssh

## Создание gpg ключа:

Генерируем ключ с помощью команды gpg --full-generate-key

Из предложенных опций выбираем: - тип RSA and RSA; - размер 4096; - срок действия 0 (не истекает)

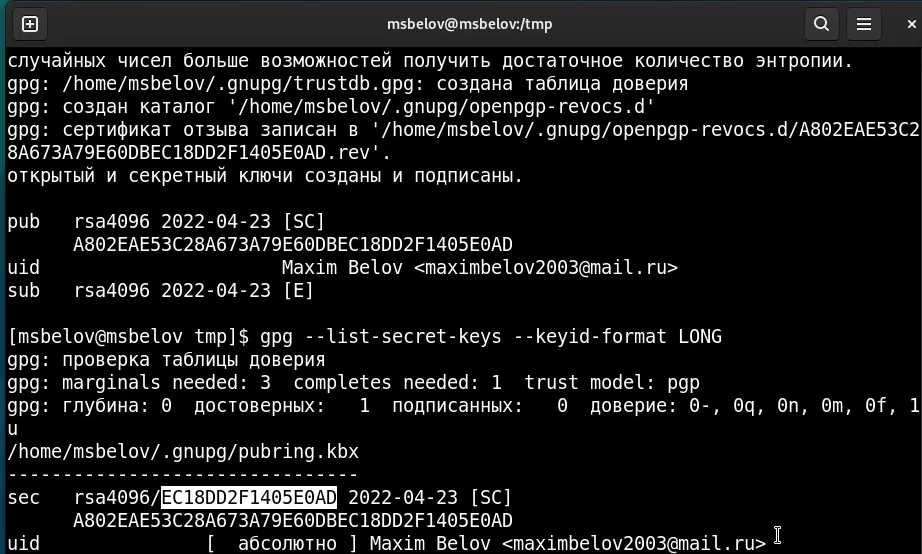
GPG запросит личную информацию, которая сохранится в ключе: - Имя (не менее 5 символов). - Адрес электронной почты. - При вводе email убедитесь, что он соответствует адресу, используемому на GitHub. - Комментарий. Можно ввести что угодно или нажать клавишу ввода, чтобы оставить это поле пустым



Создание ключа gpg

## Добавление GPG ключа в GitHub:

* Выводим список ключей и копируем отпечаток приватного ключа с помощью команды gpg --list-secret-keys --keyid-format LONG
* Генеририруем ключ с помощью команды gpg --armor --export <Отпечаток ключа>
* Перейдем в настройки GitHub, нажмем на кнопку New GPG key и вставим полученный ключ в поле ввода.

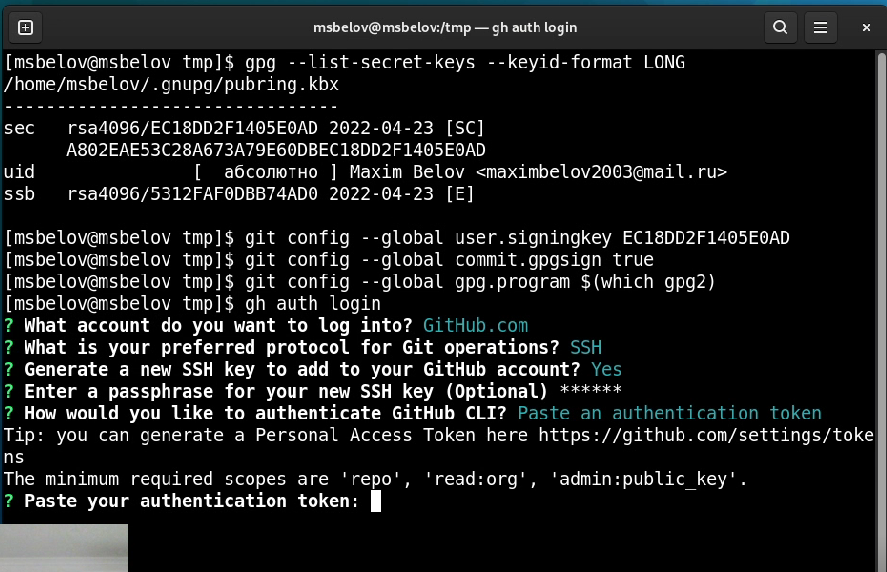


Получение отпечатка приватного ключа

## Настройка автоматических подписей коммитов git:

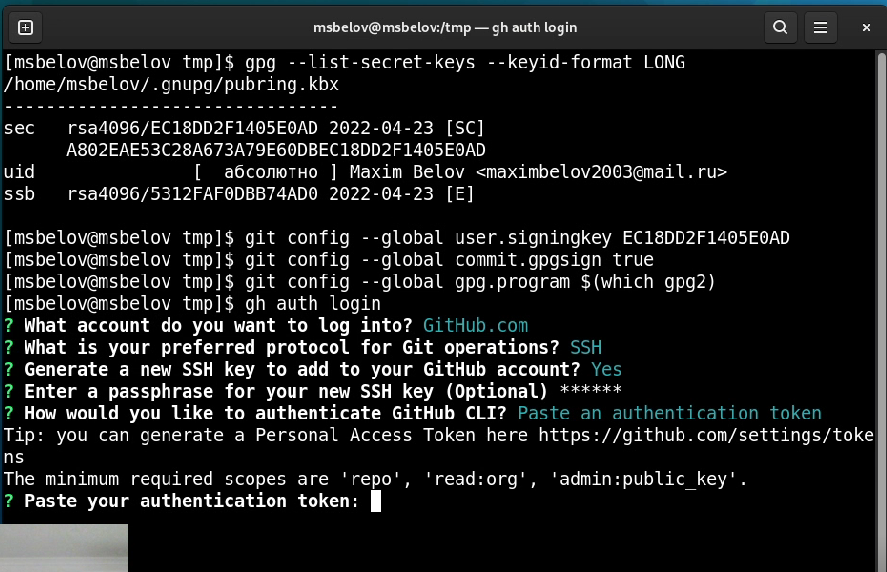
Используя введёный email, укажем Git применять его при подписи коммитов:

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



Настройка автоматических подписей коммитов git

## Настройка gh:



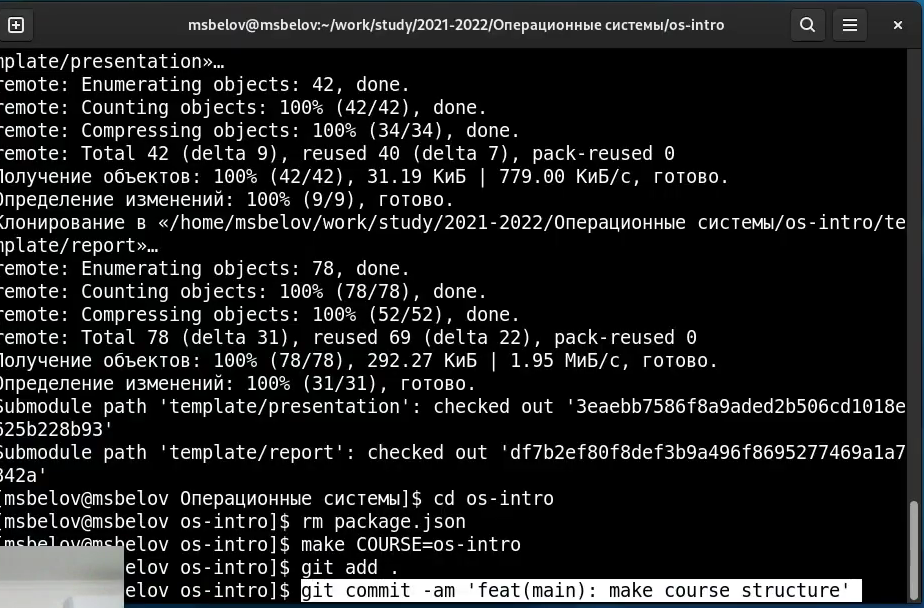
Настройка gh

## Сознание репозитория курса на основе шаблона и настройка каталога курса:

**Создадим шаблон рабочего пространства с помощью команд:**

* mkdir -p ~/work/study/2021-2022/"Операционные системы"
* cd ~/work/study/2021-2022/"Операционные системы"
* gh repo create study\_2021-2022\_os-intro --template=yamadharma/course-directory-student-template --public
* git clone --recursive git@github.com:<owner>/study\_2021-2022\_os-intro.git os-intro

Перейдем в каталог курса, удалим лишние файлы, создадим необходимые каталоги и отправим файлы на сервер:



Настройка каталога курса и отправка файлов на сервер

# Выводы

В ходе работы я изучил идеологию и применение средств контроля версий, а также работу с git.