

Лабораторная работа №2

Управление версиями

Бабенко Артём Сергеевич

Содержание

Цель работы	3
Теоретическое введение	4
Выполнение лабораторной работы	5
Выводы	10
Контрольные вопросы	11

Цель работы

Изучить идеологию и применение средств контроля версий. Освоить умения по работе с git.

Теоретическое введение

Markdown — это облегченный язык разметки с синтаксисом форматирования обычного текста. Документация поддерживает разметку Markdown в соответствии с CommonMark и ее синтаксический анализ через подсистему Markdig. Документация также поддерживает пользовательские расширения Markdown, которые предоставляют более обширный контент на сайте документации.

Для записи Markdown можно использовать любой текстовый редактор, но мы рекомендуем Visual Studio Code с расширением Docs Authoring Pack. Расширение Docs Authoring Pack содержит средства редактирования и функции предварительного просмотра, позволяющие увидеть, как будут выглядеть статьи на сайте Документации.

Выполнение лабораторной работы

1. Устанавливаю git-flow в Fedora Linux (рис.1).

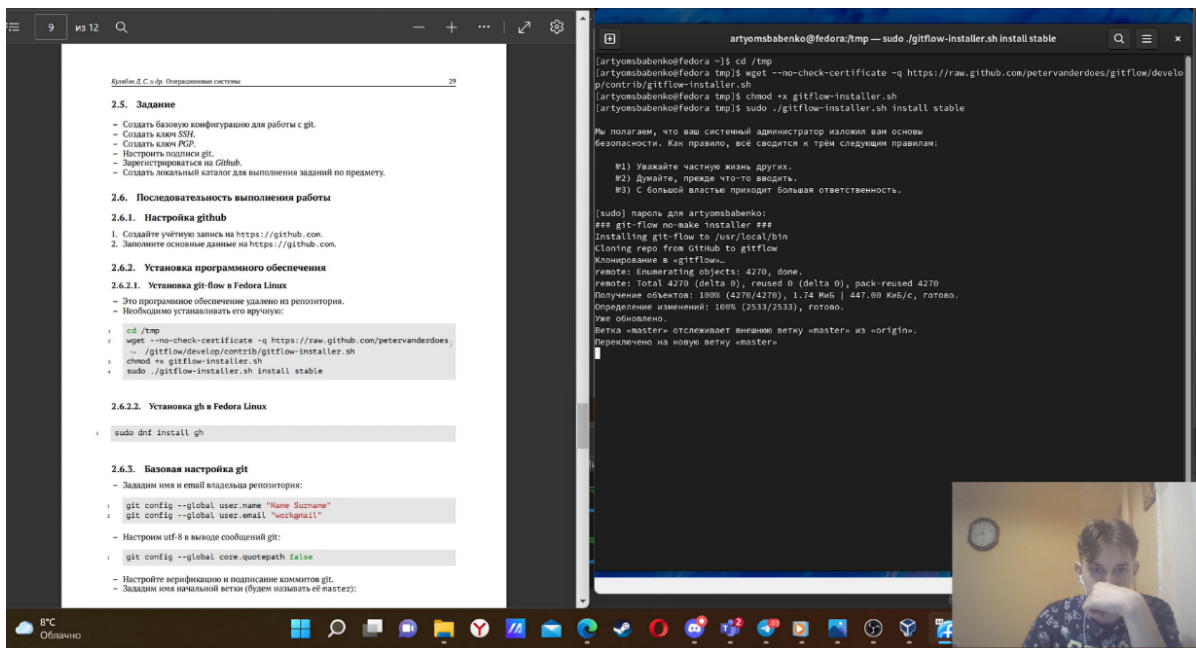


Рис.1. Установка git-flow

2. Делаю базовую настройку git (рис.2).

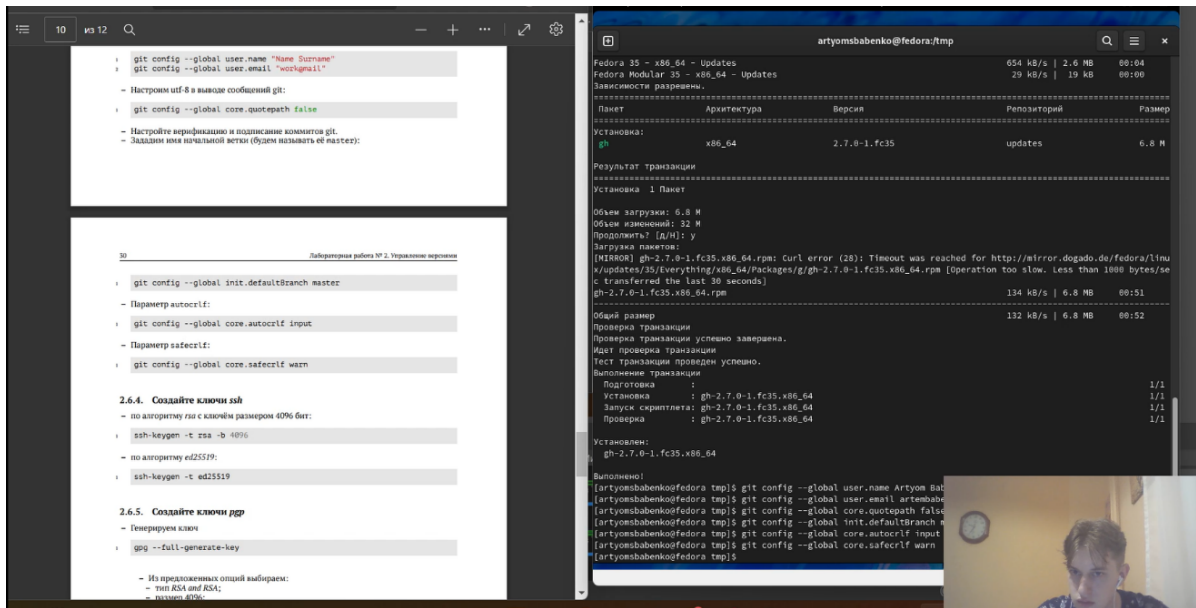


Рис.2. Настройка git

3. Создаю ключ SSH (рис.3.)

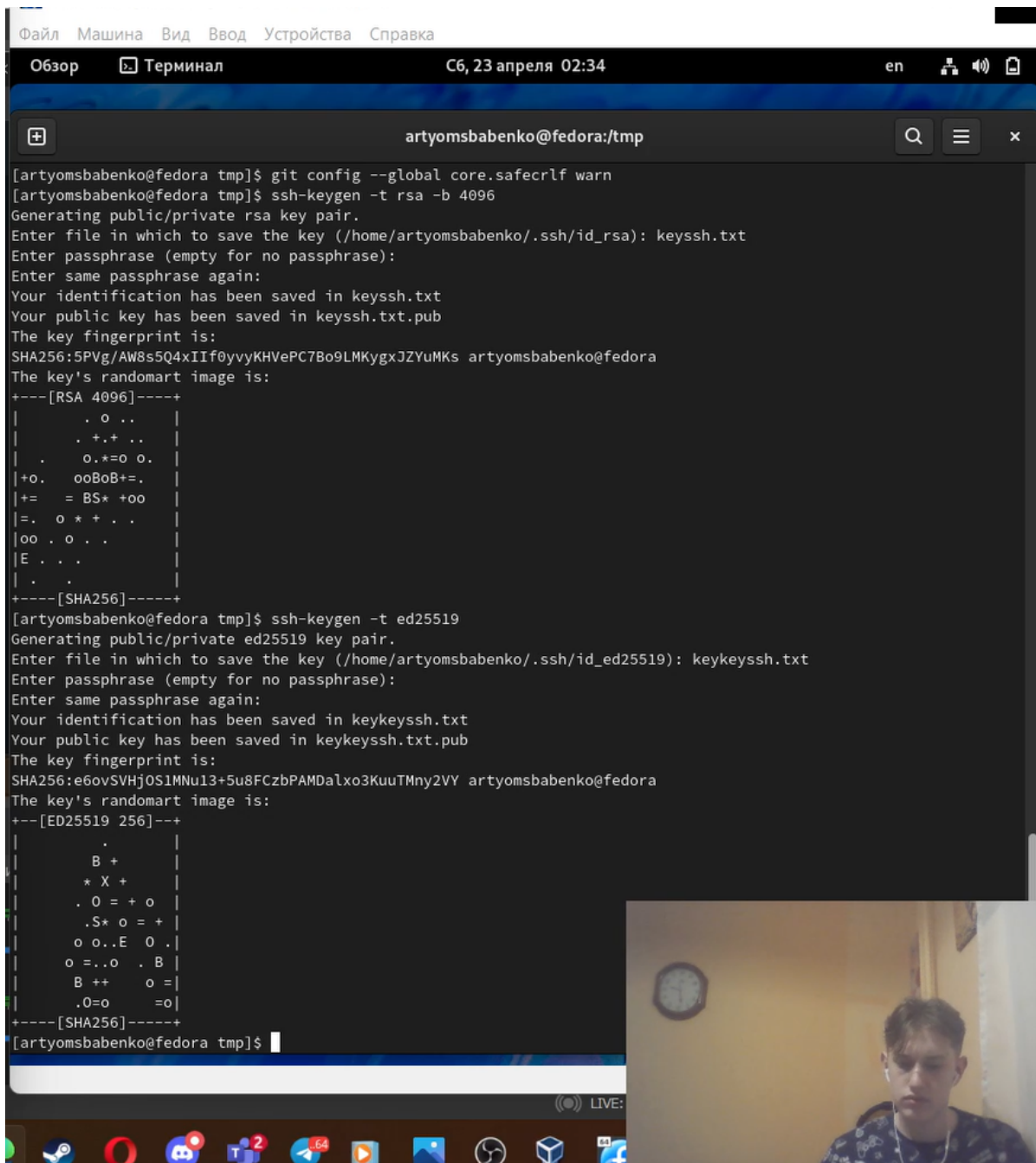


Рис.3. Создание SSH ключа

4. Создаю ключ PGP (рис.4).

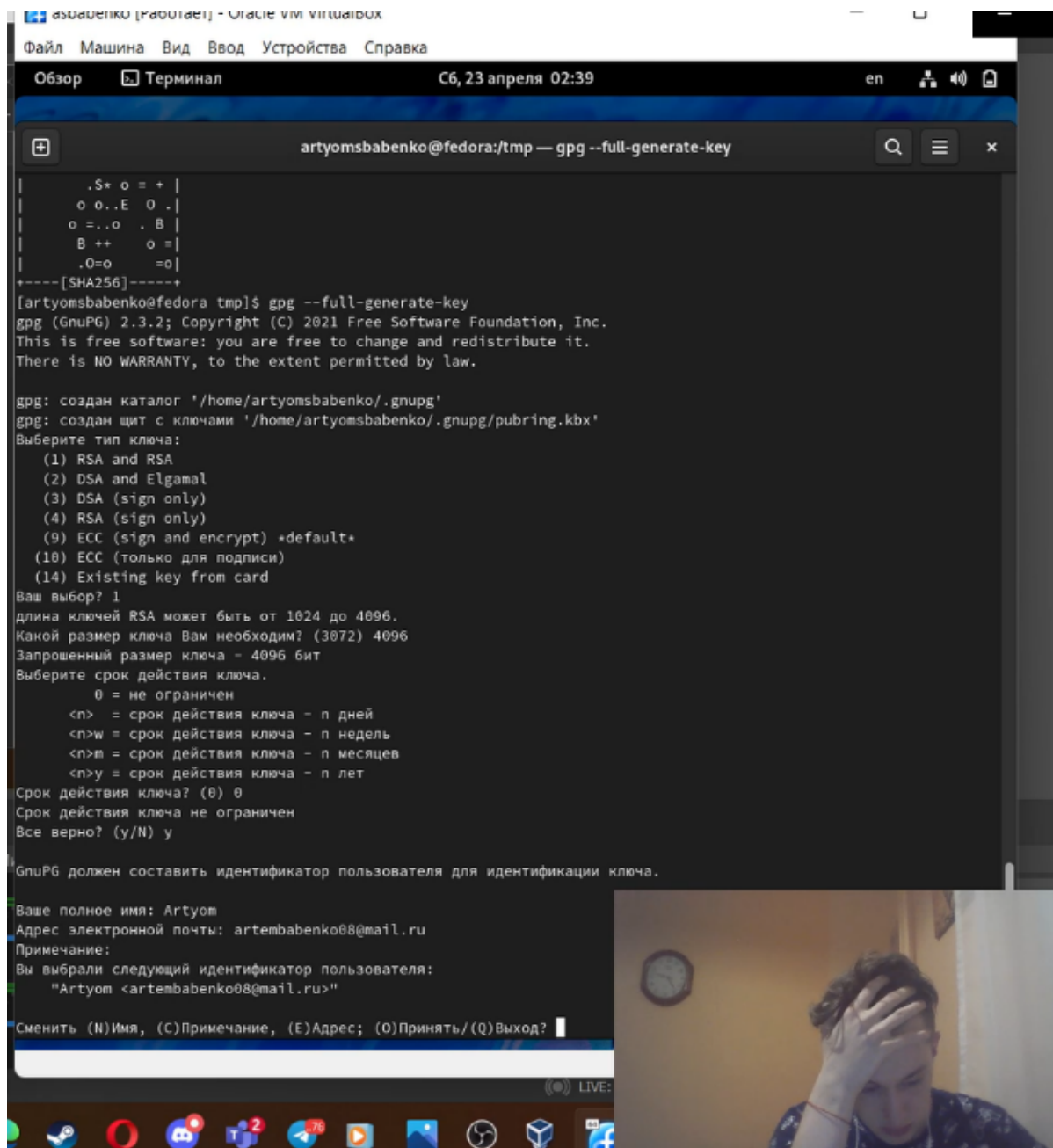


Рис.4. Создание PGP ключа.

5. Настраиваю подписи git (рис.5.)

Выводы

Я изучил идеологию и применение средства контроля версий, а также освоил умения по работе с git.

Контрольные вопросы

1. Система контроля версий (Version Control System, VCS) — программное обеспечение для облегчения работы с изменяющейся информацией. и VCS 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, и многое другое.
2. 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.
3. Ветки нужны, чтобы несколько программистов могли вести работу над одним и тем же проектом или даже файлом одновременно, при этом не мешая друг другу. Кроме того, ветки используются для тестирования экспериментальных функций: чтобы не повредить основному проекту, создается новая ветка специально для экспериментов.