Лабораторная работа 2

Сафин Андрей Алексеевич

Содержание

# 1 Цель работы

Приобретение навыков работы с языком разметки markdown.

# 2 Задание

Написать отчет по лабораторной работе 2 в markdown.

# 3 Теоретическое введение

Markdown - язык разметки, позволяющий удобным образом создавать и форматировать текстовые документы с использованием изображений с помощью различного рода синтаксиса.

# 4 Выполнение лабораторной работы

Софт, необходимый для работы с git, был установлен ранее в рамках курса архитектуры компьютеров. Остается создать PGP ключ (рис. [[1](#fig:001)]-[[3](#fig:003)]). Сохранить его на github (рис. [[4](#fig:004)]-[[7](#fig:007)]). Наличие аккаунта и ключей на нем подтверждает рис. [[8](#fig:008)].

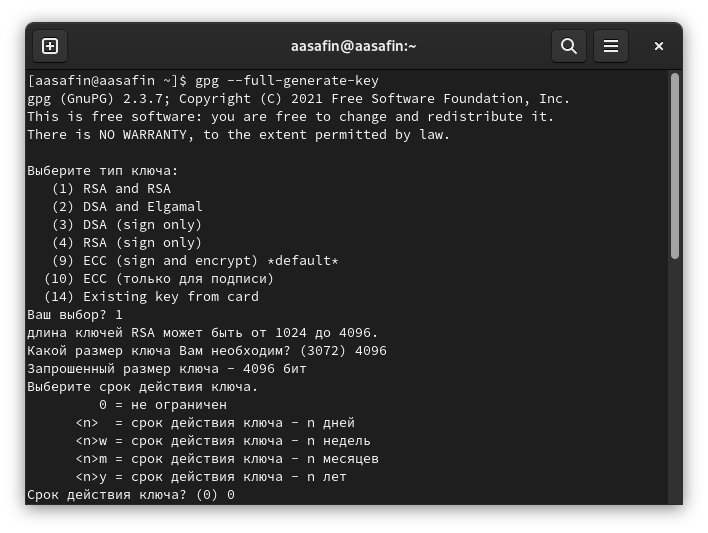


Figure 1: Создание PGP ключа

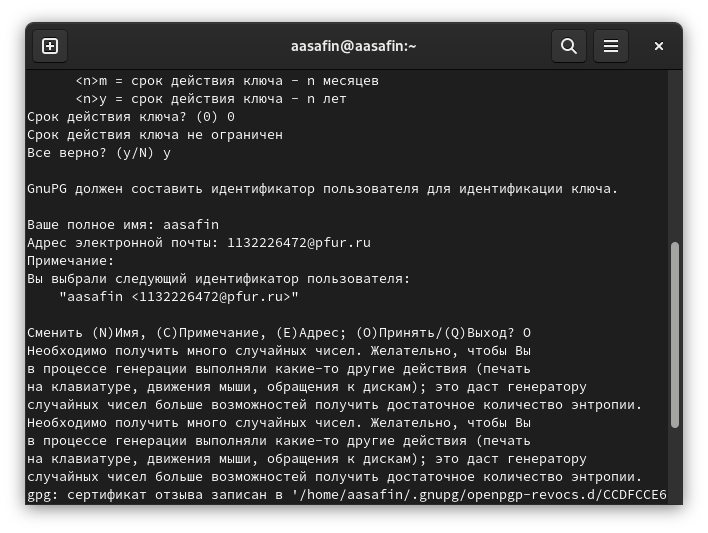


Figure 2: Создание PGP ключа (2)

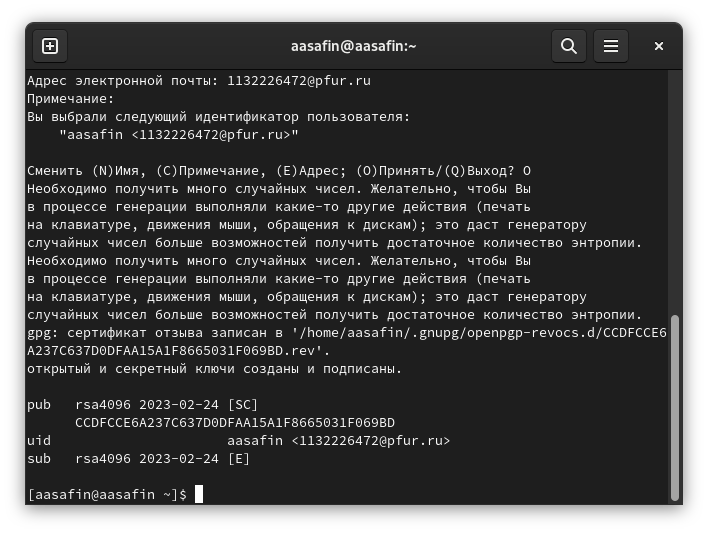


Figure 3: Создание PGP ключа (3)

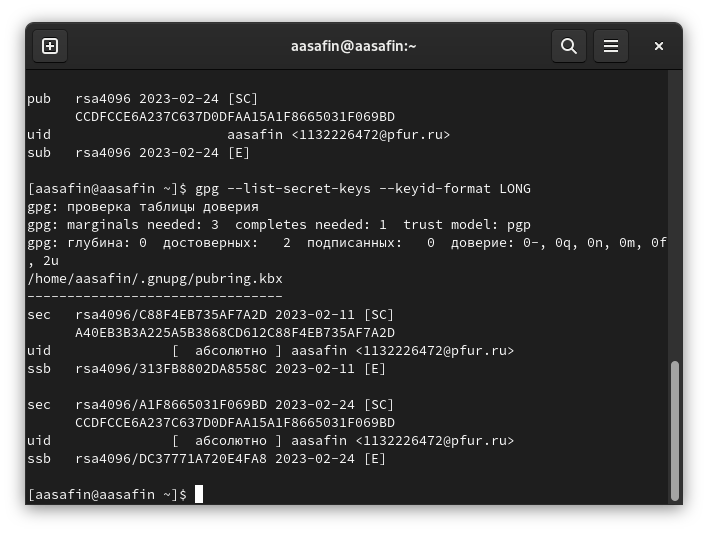


Figure 4: Вывод PGP ключа

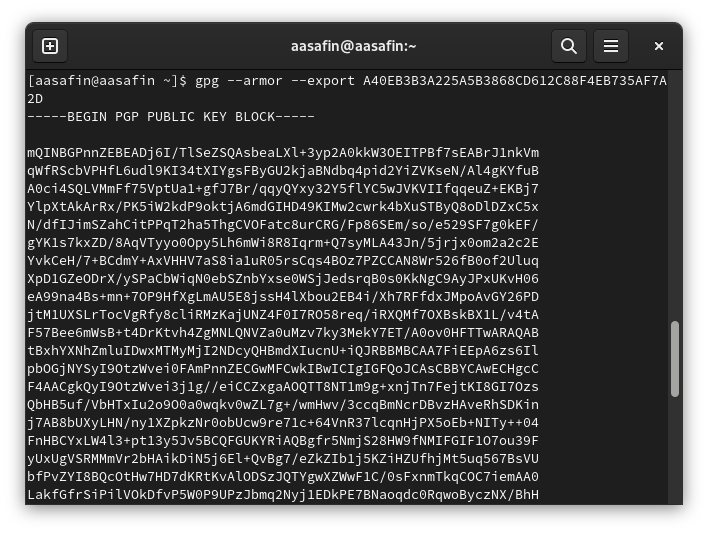


Figure 5: Экспорт PGP ключа

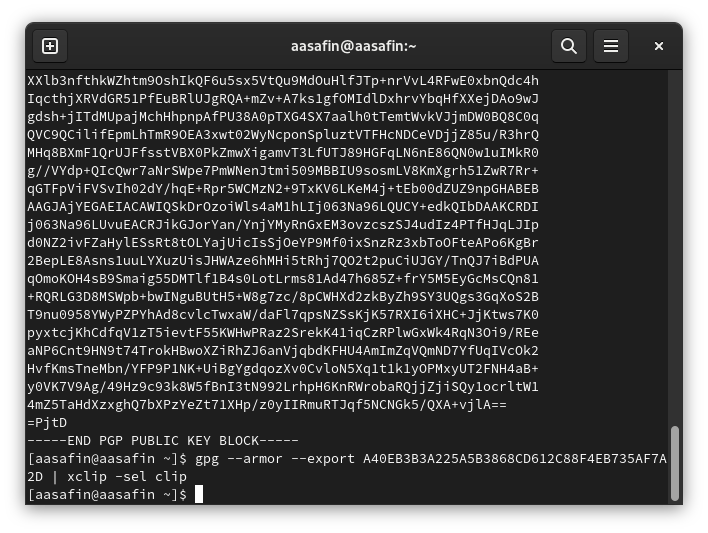


Figure 6: Экспорт PGP ключа (2)

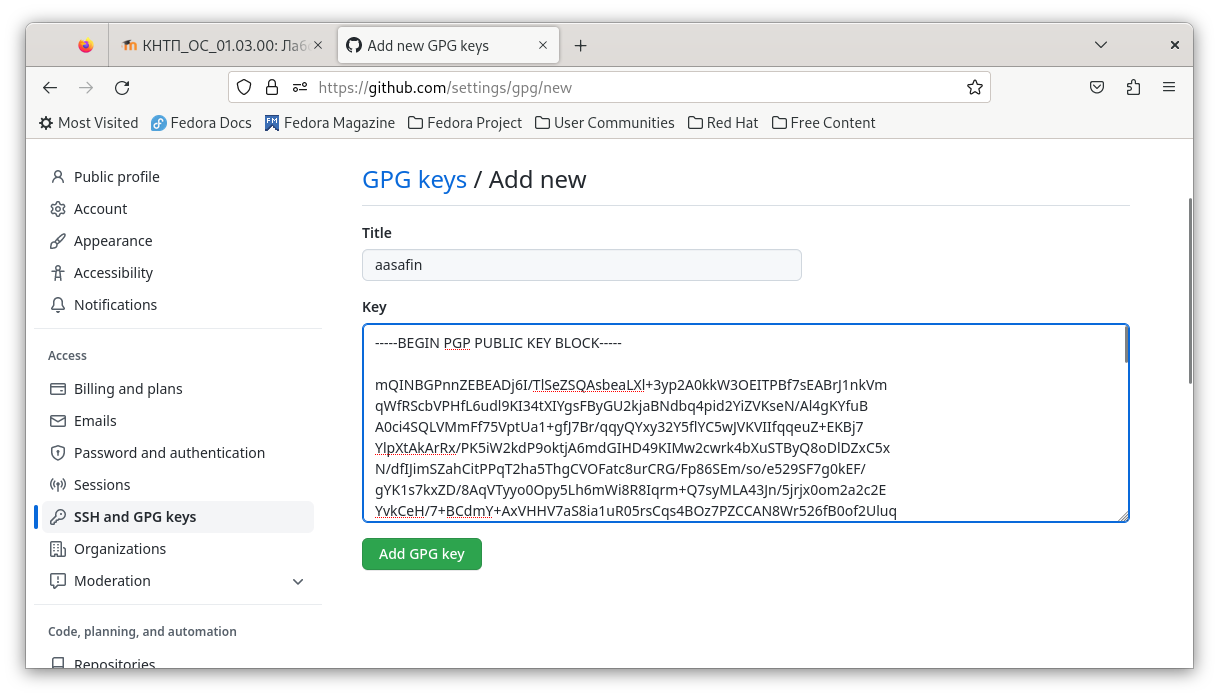


Figure 7: Сохранение PGP ключа на github

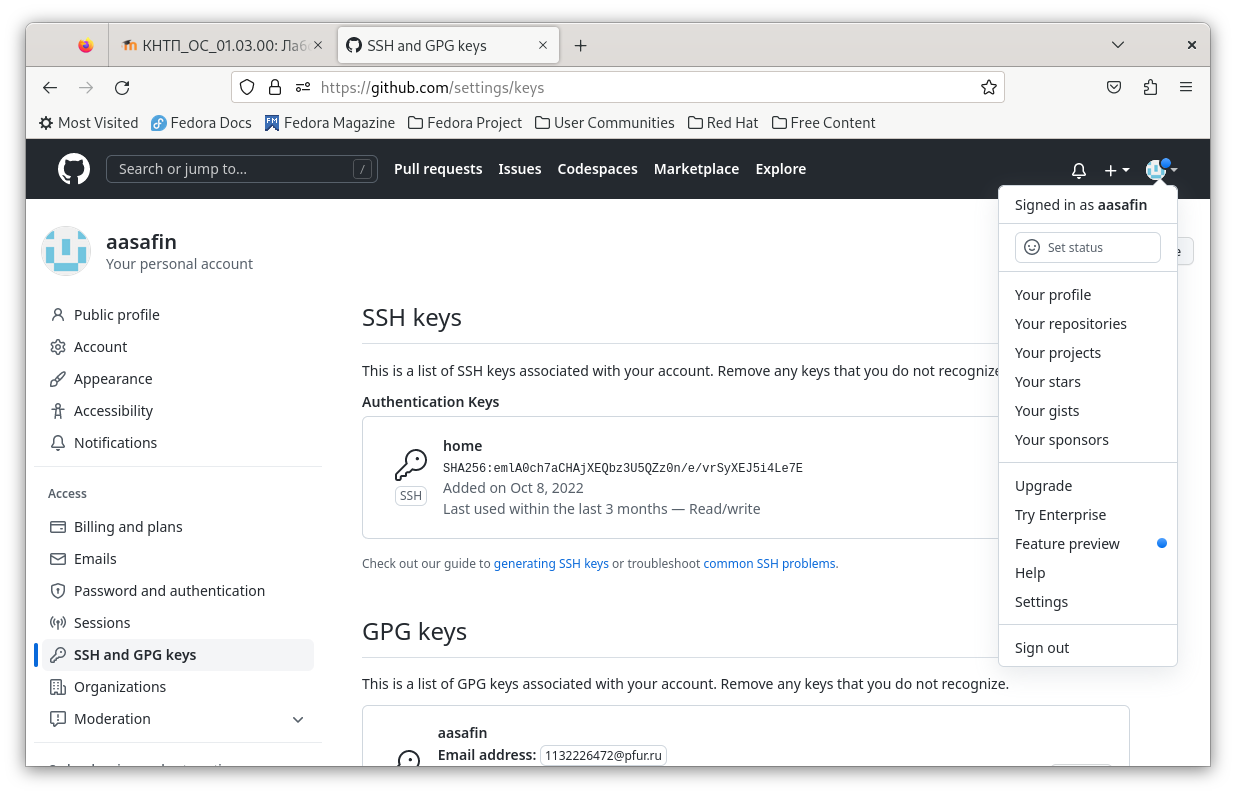


Figure 8: Демонстрация наличия аккаунта github и сохраненных ключей на нем

Помимо этого требуется настроить автоматические подписи (рис. [[9](#fig:009)]). Клонировать репозиторий на виртуальную машину (рис. [[10](#fig:010)]-[[11](#fig:011)]). И настроить структуру файлов (рис. [[12](#fig:012)]-[[13](#fig:013)]).

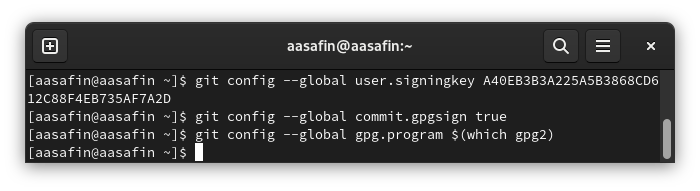


Figure 9: Настройка автоматических подписей коммитов

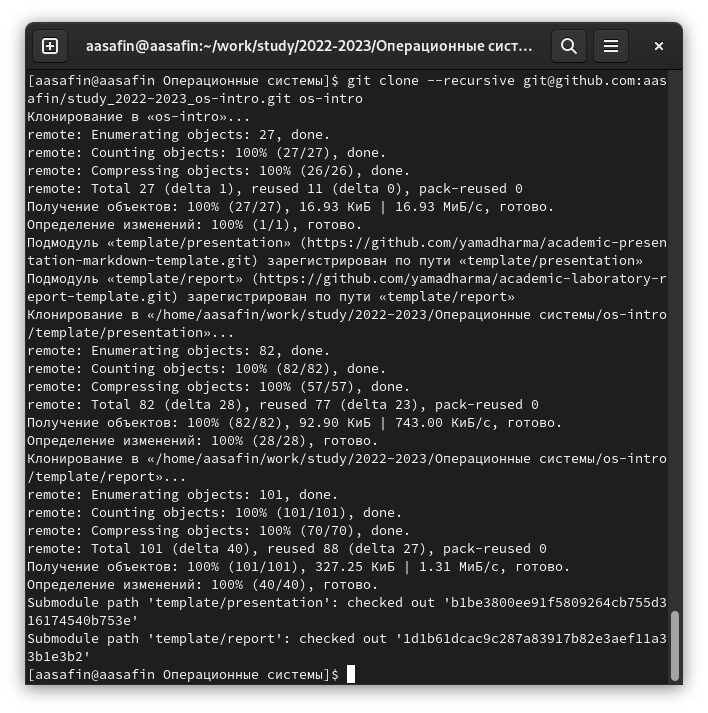


Figure 10: Клонирование репозитория на виртуальную машину

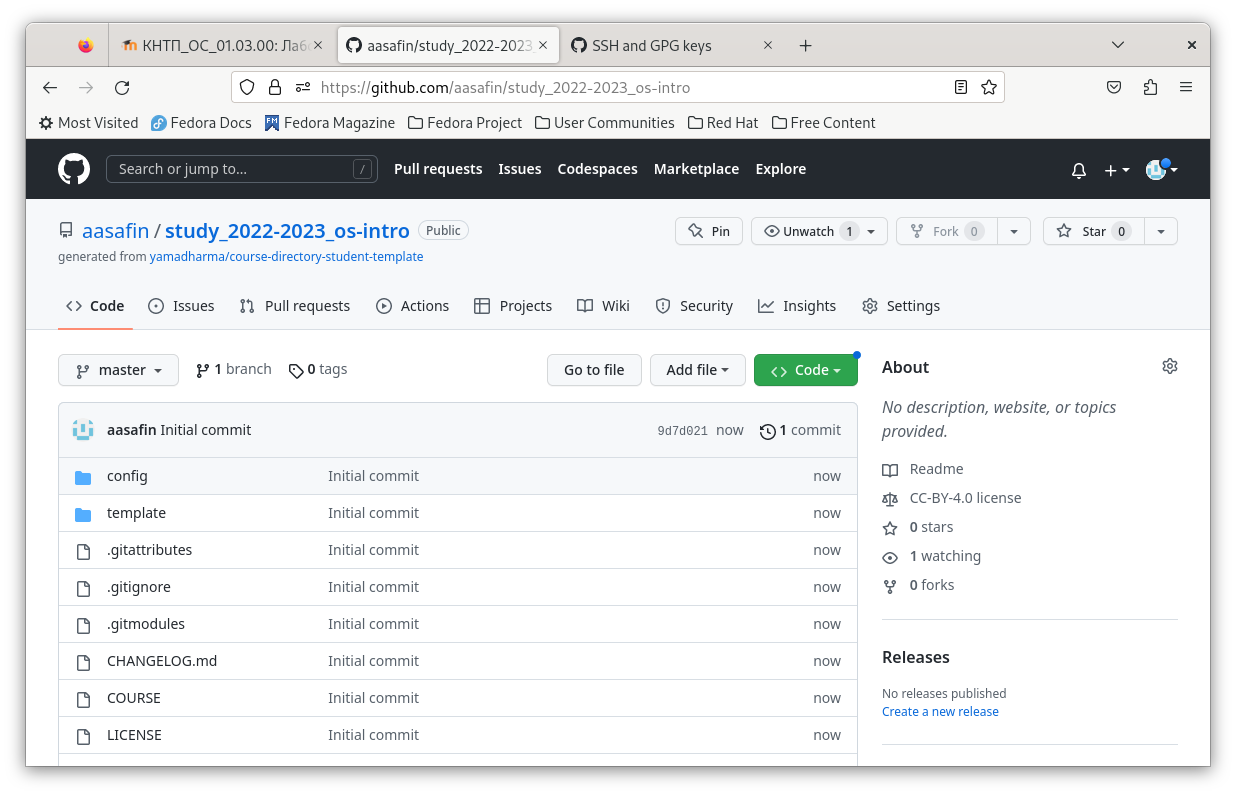


Figure 11: Демонстрация наличия репозитория на аккаунте

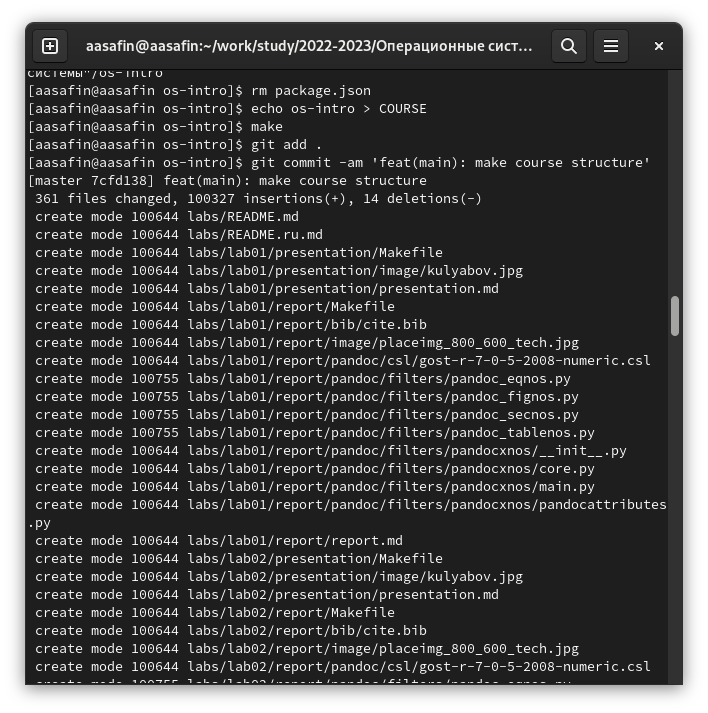


Figure 12: Конечная настройка файловой структуры

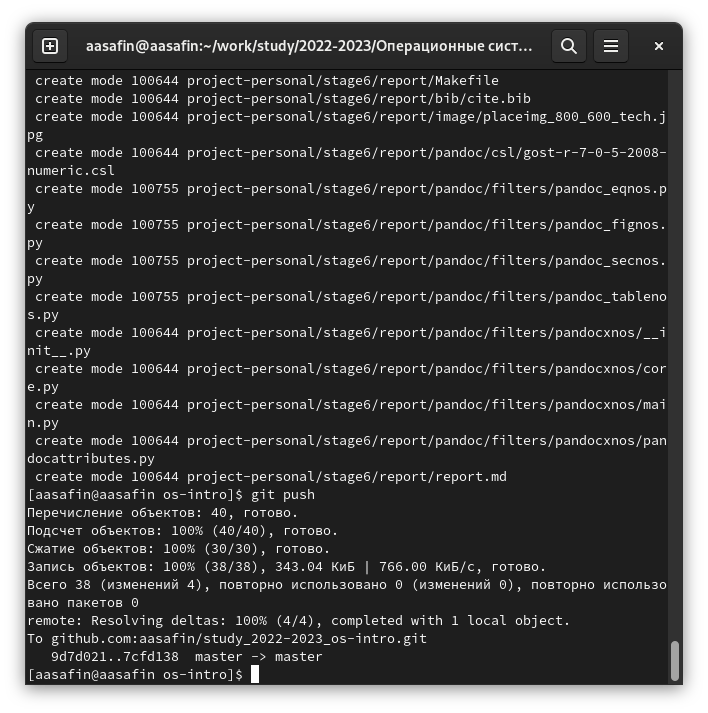


Figure 13: Конечная настройка файловой структуры (2)

# 5 Ответы на контрольные вопросы

1. VCS - система контроля версий. Используется для удобной работы множества людей над одним проектом. Позволяет разделять изменения в зависимости от автора и момента изменения, применять их по отдельности, и возвращаться к более старым версиям проекта.
2. Хранилище - место хранения версий. Commit - фиксация изменений с остороны определенного лица. История - список внесенных изменений, структурированный по времени. Рабочая копия - локальная копия документа, располагающаяся в память устройства пользователя.
3. Централизованная модель - модель, в которой единый сервер обеспечивает хранилище и большую часть функций контроля версий. Пример: CVS. В децентрализованных системах же единый сервер необязателен. Пример: git.
4. Стягивание изменений из репозитория для работы с нужной версией, изменение версии пользователем, коммит изменений с возможными комментариями и задвигание изменений обратно на репозиторий.
5. Описание действий в пункте 4 соответствует порядку работы с общим хранилищем.
6. Основные задачи, решаемые инструментами git - задачи систем контроля версий.
7. pull - стягивание изменений, получение рабочий копии. add - добавление изменений в коммит. commit - фиксация изменений. push - отправление изменений на репозиторий.
8. Например, если имеется проект, включающий создание обширной программы, части кода которой распределены между сотрудниками, может быть использован git для удобства контроля версий на удаленном репозитории. Также может быть достаточно системы контроля, использующей локальный репозиторий, в случае, если работа происходит в пределах одной локации, и не требует выхода в интернет.
9. Ветвь - это версия проекта, существующая независимо от основной версии. Используются ветви для привнесения изменений разными людьми или привнесения изменений разного характера с дальнейшей возможностью различать их откатывать их, наоборот, сливать воедино, или выбирать необходимые и добавлять только их.
10. Игнорировать файлы можно с помощью команды add, использованной к ограниченному количеству файлов, и команды remove. Это делается в случае, если пользователь не хочет сохранения некоторых изменений на репозитории.

# 6 Выводы

Навык работы с языком разметки markdown освоен.