Лабораторная работа 1

Git и markdown

Евдокимова Юлия Константиновна, НПИбд-01-18

Содержание

List of Tables

List of Figures

1 Цель работы

Цель работы — установка необходимых программ, закрепление навыков работы с git, освоение языка разметки markdown.

2 Задание

1. Установить git, закрепить базовые команды и выгрузить первые файлы на github.
2. Выполнить отчет в формате Markdown.

3 Выполнение лабораторной работы

3.1 Закрепление навыков работы с Git

3.1.1 Теоретические сведения

Git — распределённая система управления версиями.

Система спроектирована как набор программ, специально разработанных с учётом их использования в сценариях. Это позволяет удобно создавать специализированные системы контроля версий на базе Git или пользовательские интерфейсы.

Ядро Git представляет собой набор утилит командной строки с параметрами. Все настройки хранятся в текстовых файлах конфигурации. Такая реализация делает Git легко портируемым на любую платформу и даёт возможность легко интегрировать Git в другие системы (в частности, создавать графические git-клиенты с любым желаемым интерфейсом).

Репозиторий Git представляет собой каталог файловой системы, в котором находятся файлы конфигурации репозитория, файлы журналов, хранящие операции, выполняемые над репозиторием, индекс, описывающий расположение файлов, и хранилище, содержащее собственно файлы.

По умолчанию репозиторий хранится в подкаталоге с названием «.git» в корневом каталоге рабочей копии дерева файлов, хранящегося в репозитории.

Любой файл в директории существующего репозитория может находиться или не находиться под версионным контролем (отслеживаемые и неотслеживаемые).

Отслеживаемые файлы могут быть в 3-х состояниях: неизменённые, изменённые, проиндексированные (готовые к коммиту).

Основные команды:

git init

Создает новый проект

git add

Добавляет содержимое рабочей директории в индекс (staging area) для последующего коммита.

git status

Показывает состояния файлов в рабочей директории и индексе: какие файлы изменены, но не добавлены в индекс; какие ожидают коммита в индексе.

git commit

Берёт все данные, добавленные в индекс с помощью git add, и сохраняет их слепок во внутренней базе данных, а затем сдвигает указатель текущей ветки на этот слепок.

git reset

Используется в основном для отмены изменений. Она изменяет указатель HEAD и, опционально, состояние индекса.

git clone

Клонирует удаленный репозиторий

3.1.2 Ход выполнения

1. Установка имени и электронной почты

Для начала нам необходимо установить имя пользователя и указать его электронную почту. (рис. 3.1).

2. Установка параметров окончаний строк

Устанавливаю все переводы строк текстовых файлов в главном репозитории одинаковыми, а также проверку обратимости преобразования для текущей настройки core.autocrlf, печать только предупреждения (рис. 3.1).

3. Установка отображения unicode

Чтобы избежать нечитаемых строк, устанавливаю соответствующий флаг (рис. 3.1).

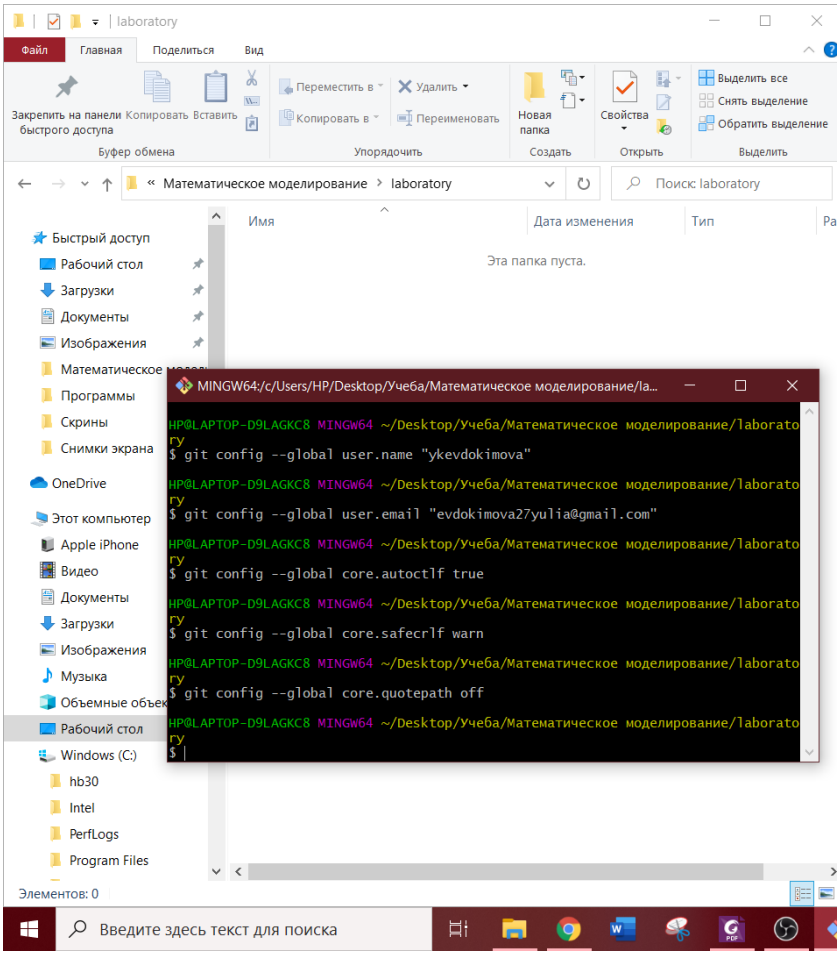

Figure 3.1: Выполнение пунктов 1-3

4. Создание первого файла, репозитория, дальнейшая работа с ними

Создаю папку для первой лабораторной, помещаю в нее пустой файл и открываю его на редактирование (рис. 3.2). После этого создаю git репозиторий из

каталога laboratory, основного каталога с работами по Математическому моделированию (рис. 3.3).

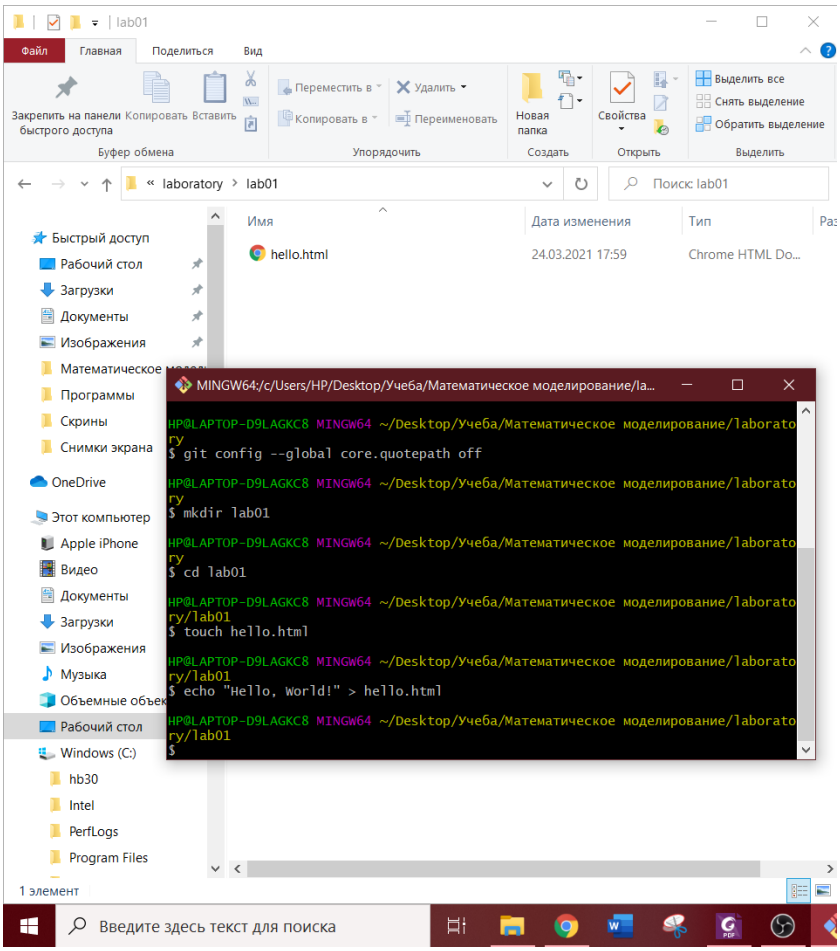

Figure 3.2: Создание файла

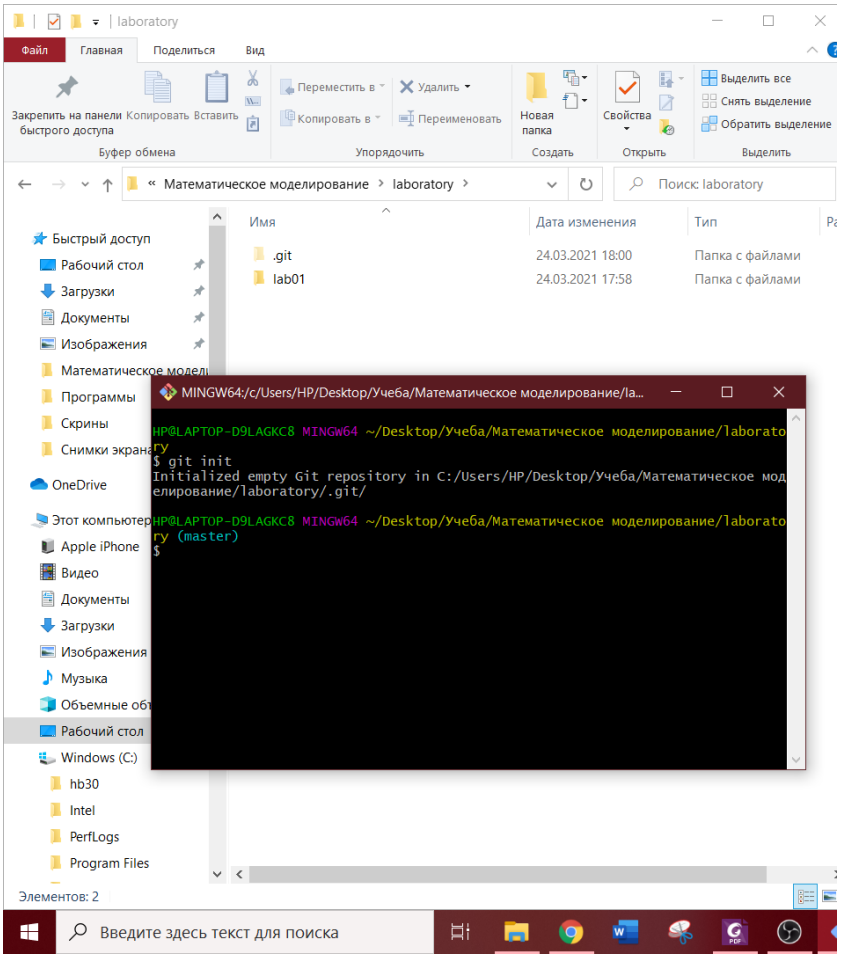

Figure 3.3: Создание репозитория

5. Основная работа с репозиторием

Добавим файл в репозиторий и проверим его состояние. (рис. 3.4).

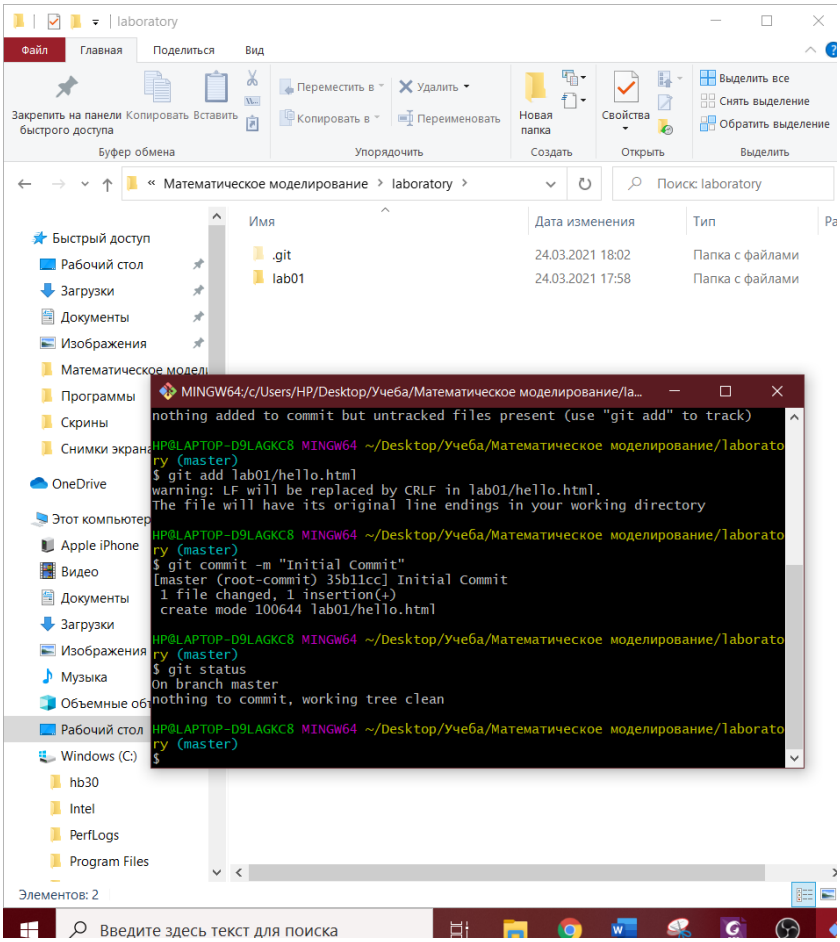

Figure 3.4: Работа с репозиторием

Изменим созданную ранее страницу (рис. 3.5).

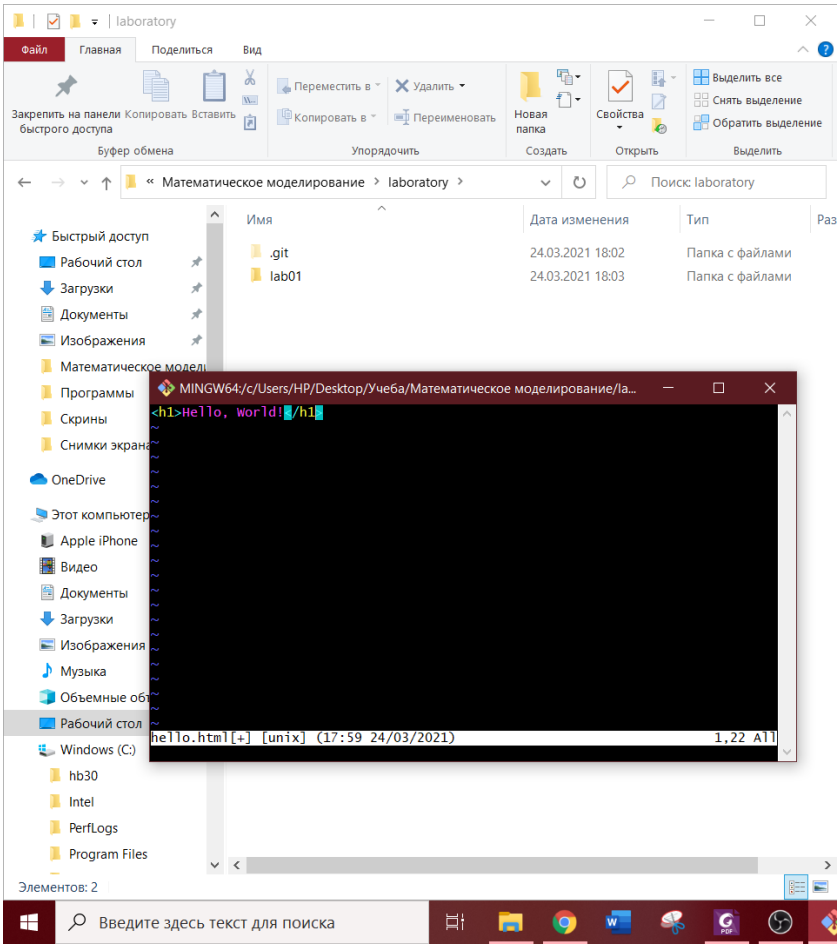

Figure 3.5: Изменение hello.html

После проверки состояния рабочего каталога командой git status видим, что файл hello.html был изменен, но при этом эти изменения еще не зафиксированы в репозитории. Также обратим внимание на то, что сообщение о состоянии дает подсказку о том, что нужно делать дальше (рис. 3.6).

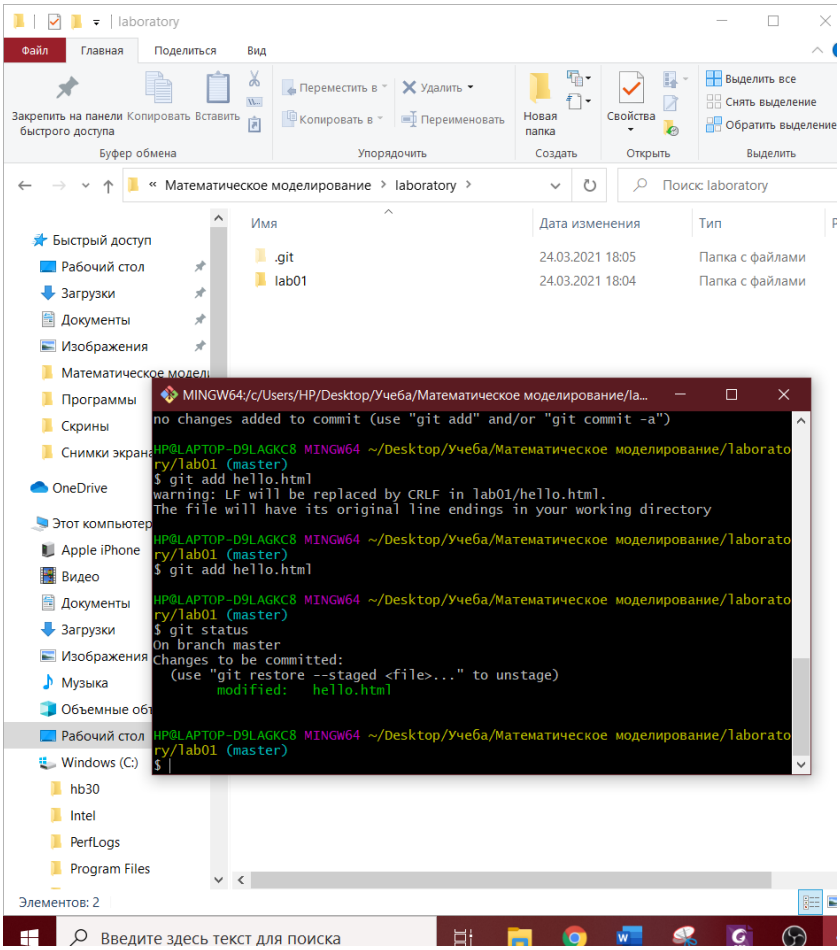

Figure 3.6: Сообщение о подсказке для hello.html

Внесем очередные изменения в наш файл (рис. 3.7).

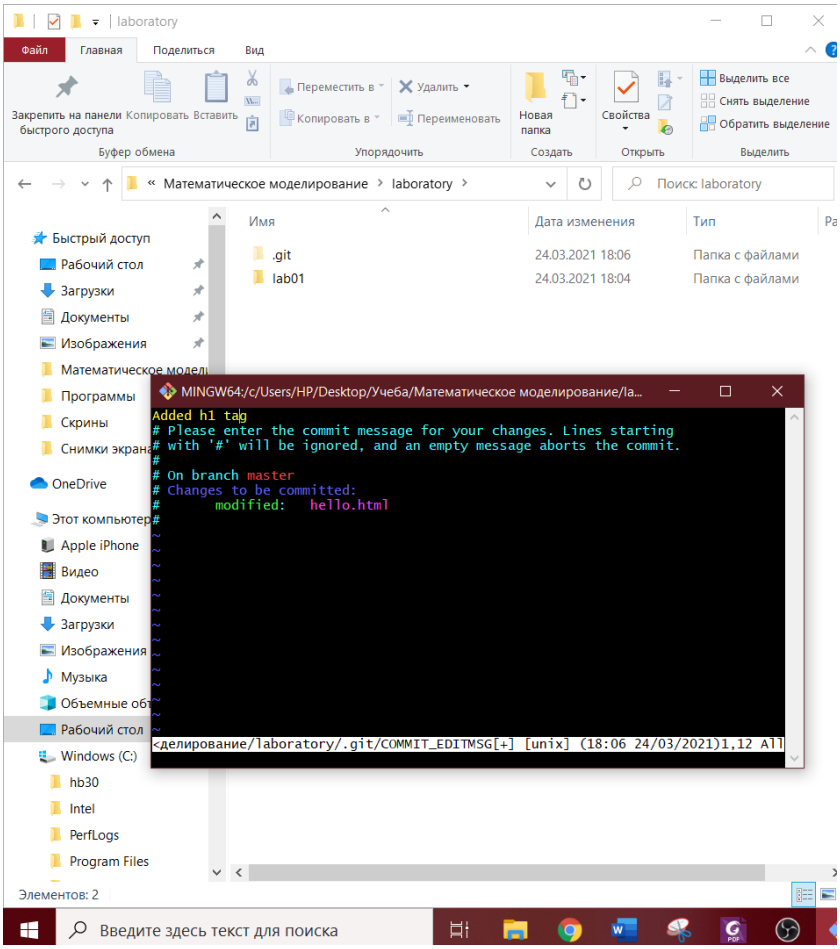

Figure 3.7: Новые изменения hello.html

Теперь выполним команду git add, чтобы проиндексировать изменения. Проверим состояние (рис. 3.8). Изменения файла hello.html были проиндексированы. Это означает, что git теперь знает об изменении, но изменение пока не записано в репозиторий. Следующий коммит будет включать в себя проиндексированные изменения.

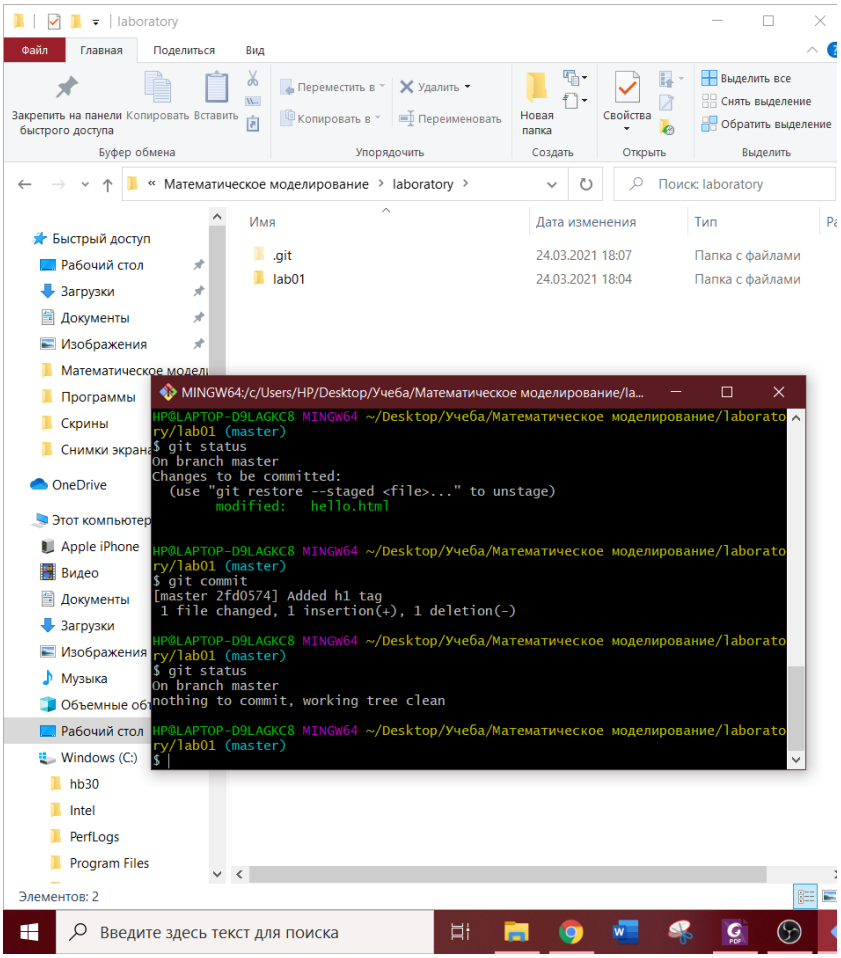

Figure 3.8: Индексация hello.html

Просматриваю список произведенных изменений в разных форматах(рис. 3.9).

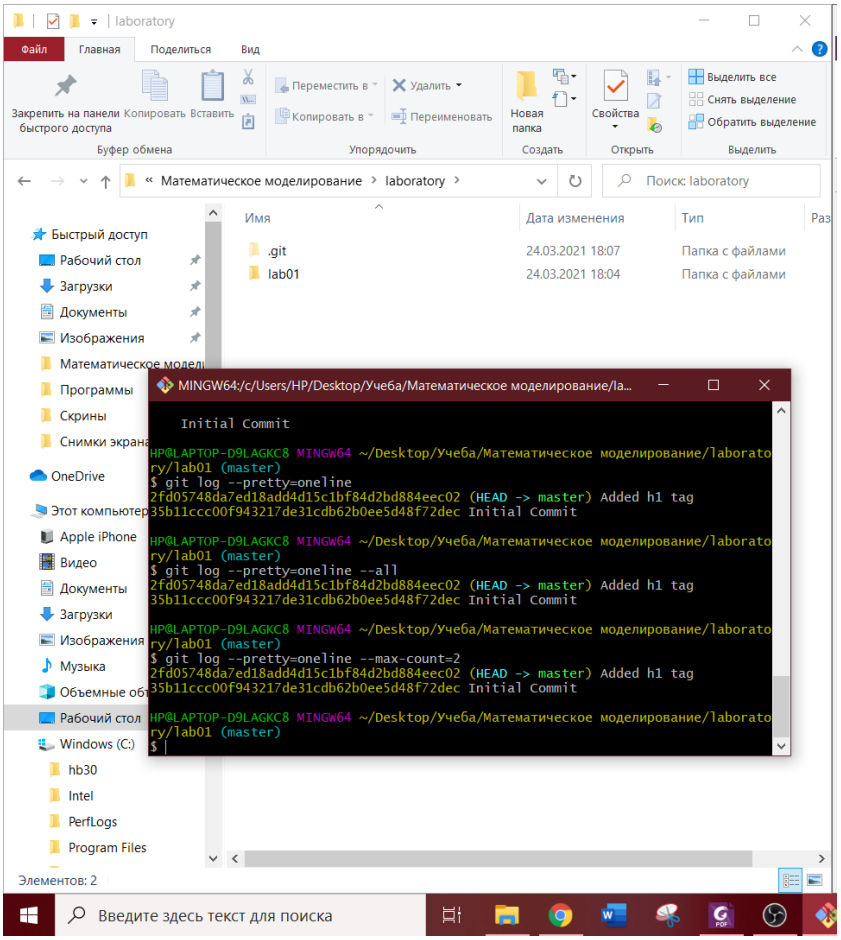

Figure 3.9: Список изменений в различных форматах

6. Подключение удаленного репозитория на GitHub.

Связываю репозиторий с репозиторием на GitHub (рис. 3.10).

В главном каталоге создаю файл README.md и добавляю его в репозиторий, обнавляю данные в удаленном репозитории (рис. 3.11).

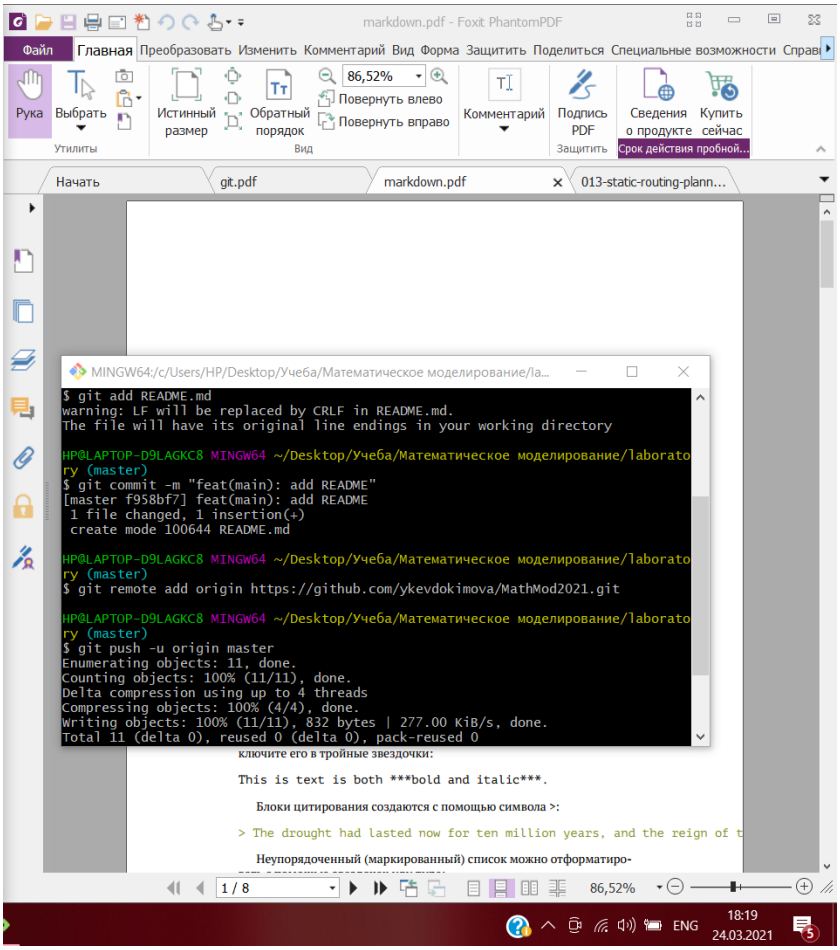

Figure 3.10: Подключение удаленного репозитория

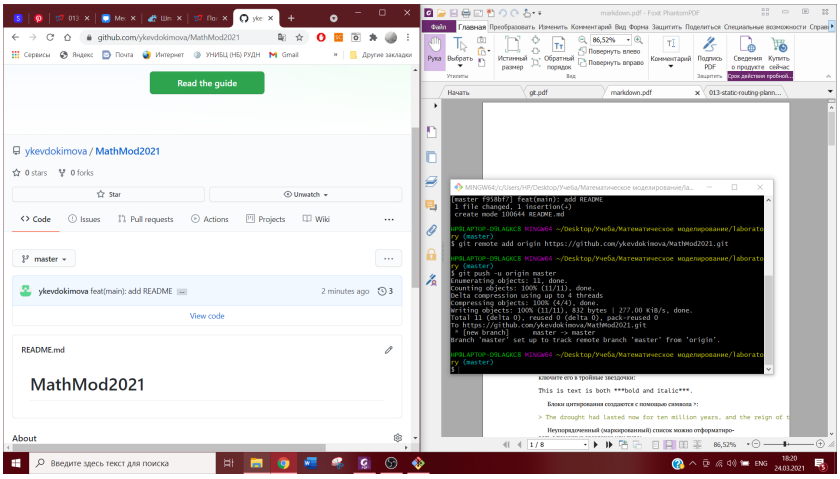

Figure 3.11: Создание файла README.md

7. Демонстрирую итог проделанной работы.

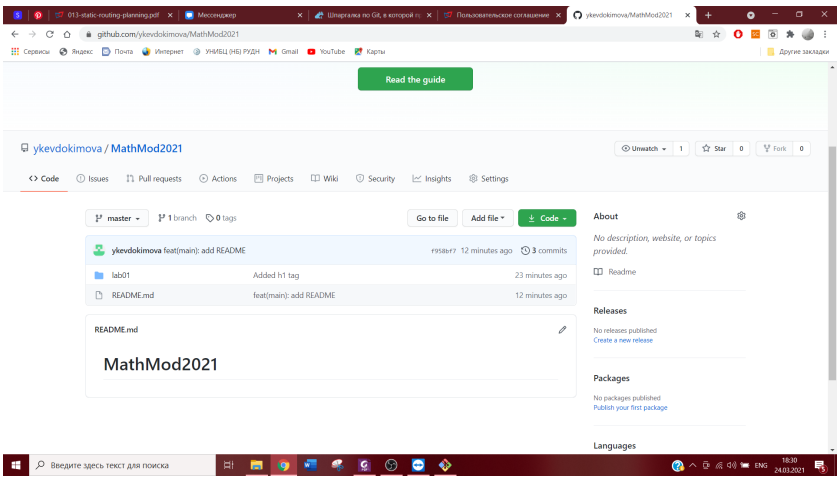

Figure 3.12: Отображение в github

3.2 Знакомство с Markdown

3.2.1 Теоритические сведения

Markdown — облегчённый язык разметки, созданный с целью обозначения форматирования в простом тексте, с максимальным сохранением его читаемости человеком, и пригодный для машинного преобразования в языки для продвинутых публикаций.

4 Вывод

Изучены основные возможности git и markdown.