Федеральное агентство связи

Ордена Трудового Красного Знамени федеральное государственное

бюджетное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра информатики

Лабораторная работа №2

по дисциплине: «Технология разработки программного обеспечения»

на тему: «Методы ветвления в системе контроля версий»

Выполнил студент

группы БФИ1902

Соцков И.Н

Москва, 2020 г.

Оглавление

[1. Цель работы 3](#_Toc51026064)

[2. Задание 3](#_Toc51026065)

[3. Ход работы 3](#_Toc51026066)

[Вывод 7](#_Toc51026067)

[Список использованных источников 8](#_Toc51026068)

**1. Цель работы**

Цель данной лабораторной работы — изучить подходы к командной разработке в системе контроля версий Git.

**2. Задание**

Задание на лабораторную работу — провести ряд следующих операций в предыдущей работе либо вновь созданном репозитории:

1. Настроить игнорирование git'ом файлов определенного расширения с использованием glob-шаблонов

2. Вывести лог коммитов в измененном пользовательском формате

3. Переименовать ветку

4. Провести в ветке работу и слить с веткой master, удалить слитую ветку

5. Настроить удаленный репозиторий: получить изменения из репозитория и отправить изменения

6. Организовать совместную работу над проектом в соответствии с выбранной идеологией

**3. Ход работы**

Для работы создадим новый репозиторий. Чтобы настроить игнорирование Git'ом файлов определенного расширения необходимо создать файл «.gitignore». Процесс создания файла и добавление его в репозиторий представлен на рисунке 1.

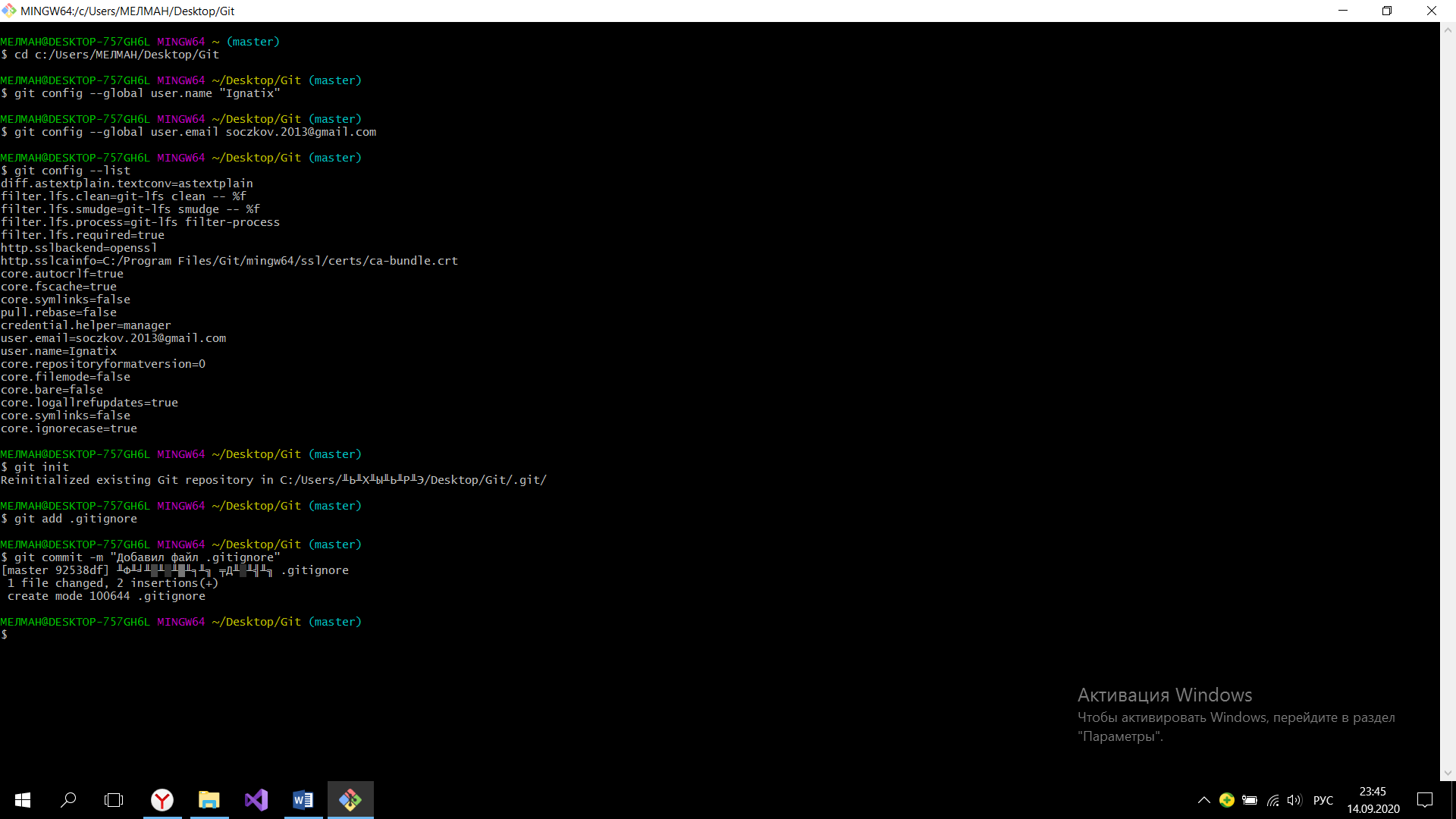


Рисунок 1 — Создание файла «.gitignore»

Внутри этого файла укажем, какой формат файлов будет игнорироваться Git'ом. Например возьмем файлы формата «.txt» и «.jpg». Теперь добавление или изменение файлов формата «.png» и «.jpg» будут игнорироваться Git'ом. Содержимое файла «.gitignore» представлено на рисунке 2.

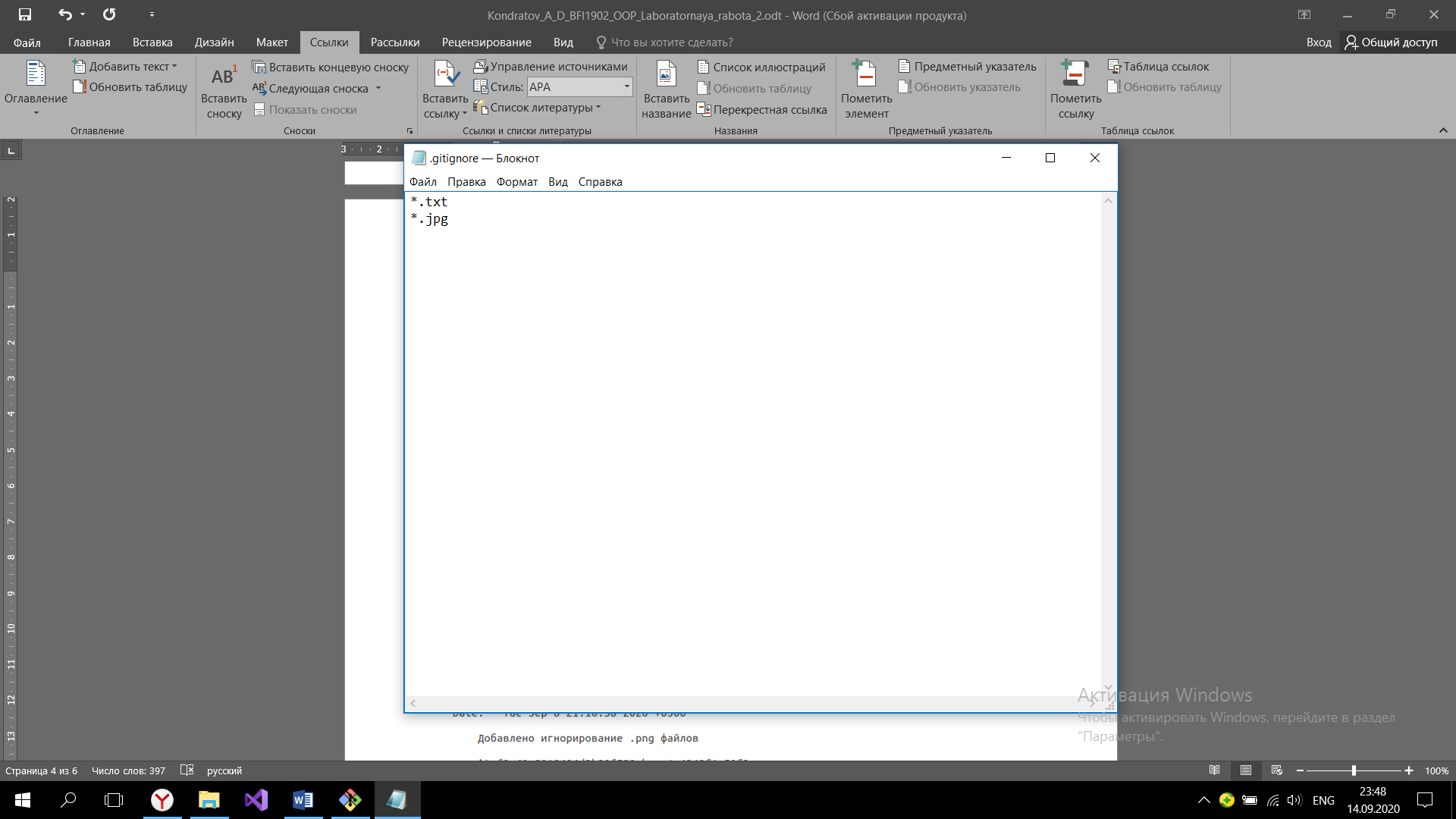


Рисунок 2 — Содержимое файла «.gitignore»

Вывести лог коммитов можно в текстовый файл при помощи команды «git log > log.txt». Содержимое файла «log.txt» представлено на рисунке 3.

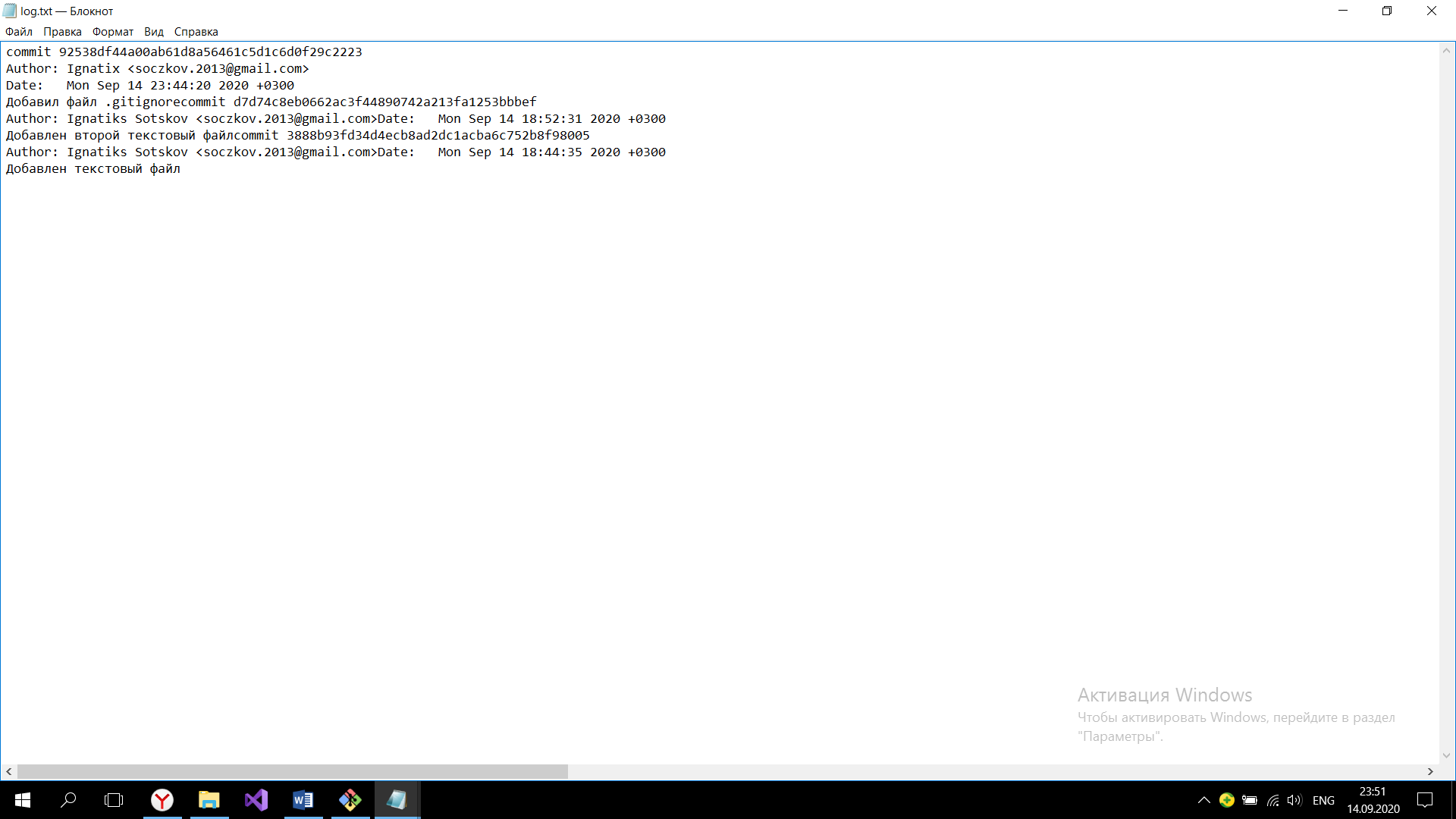


Рисунок 3 — Содержимое файла «log.txt»

Теперь создадим новую ветку и переименуем её. Ветку назовем «OOPBranch», а переименовывать будем в «OOP2Branch». Сделаем это при помощи команды «git branch -m OOPBranch OOP2Branch». Теперь, перейдя в ветку «OOP2Branch», добавим файл «file.txt». Далее запишем в «file.txt» информацию, в моем случае я записал это: «как же тяжело разбираться с СКВ Git». Затем переименовываем ветку как и планировали и вносим изменение в файл «file.txt», записывая туда сообщение «я люблю ООП!». Дальше произведем коммит и сольем ветку с веткой master. И в итоге удалим слитую ветку при помощи команды «git branch -d OOP2Branch». Выше описанные действия представлены на рисунке 4.

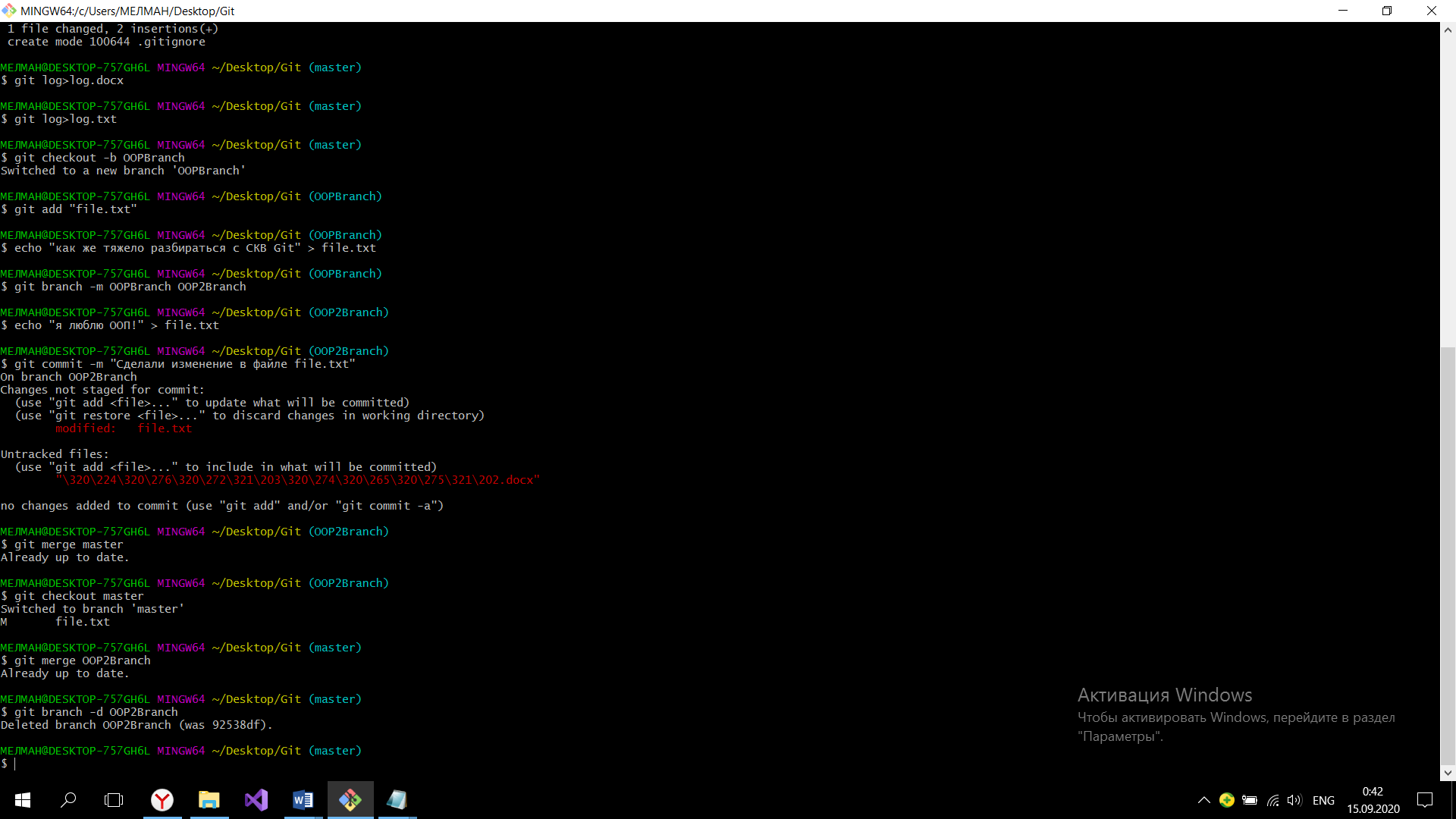


Рисунок 4 – Операции над веткой

После этого создадим удаленный репозиторий под названием «oop» на сайте «[https://github.com](https://github.com/)». Мы можем связать локальный и удаленный репозитории при помощи команды «git remote add origin https://github.com/Имя\_пользователя/oop.git», после чего отправить содержимое локального репозитория на удаленный при помощи команды «git push -u origin master». Выше описанные действия показаны на рисунке 5.

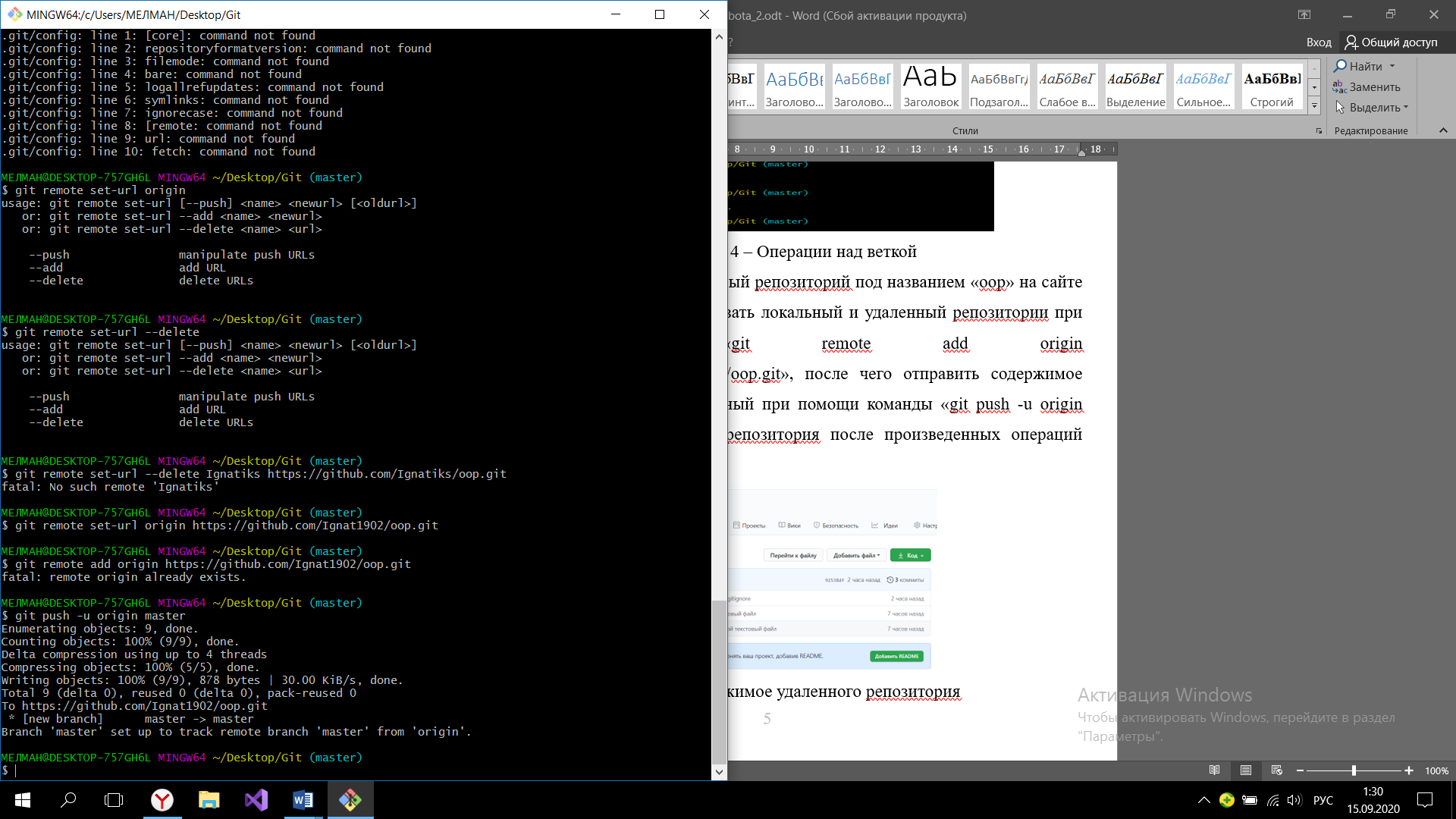


Рисунок 5 – Создание удаленного репозитория

Содержимое удаленного репозитория после произведенных операций представлено на рисунке 6.

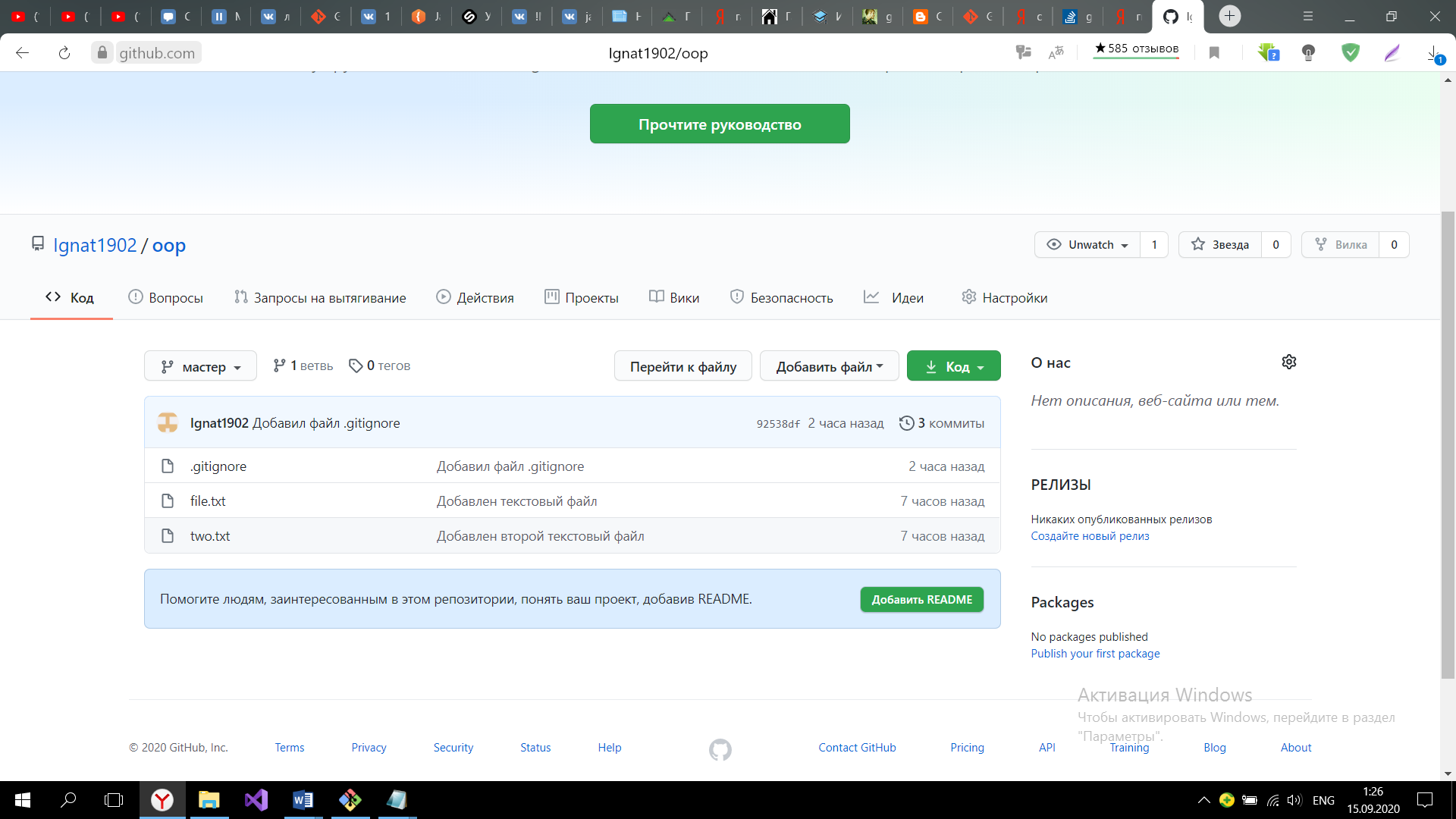


Рисунок 6 — Содержимое удаленного репозитория

Вывод

Таким образом, мы изучили подходы к командной разработке в системе контроля версий Git.

Список использованных источников

1. git-scm.com
2. Scott Chacon, Ben Straub «Pro Git»