**МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ**

**ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций** **Институт цифрового развития**

**ОТЧЁТ**

**по лабораторной работе №1.3**

Дисциплина: «Основы кроссплатформенного программирования» Тема: «Основы ветвления Git»

|  |  |  |
| --- | --- | --- |
| Выполнил: студент 1 курса | | |
| группы ИВТ-б-о-21-1 | | |
| Богдашов Артём |

Ставрополь 2022

1. Создал репозиторий лицензии MIT. После чего клонировал его на ПК:

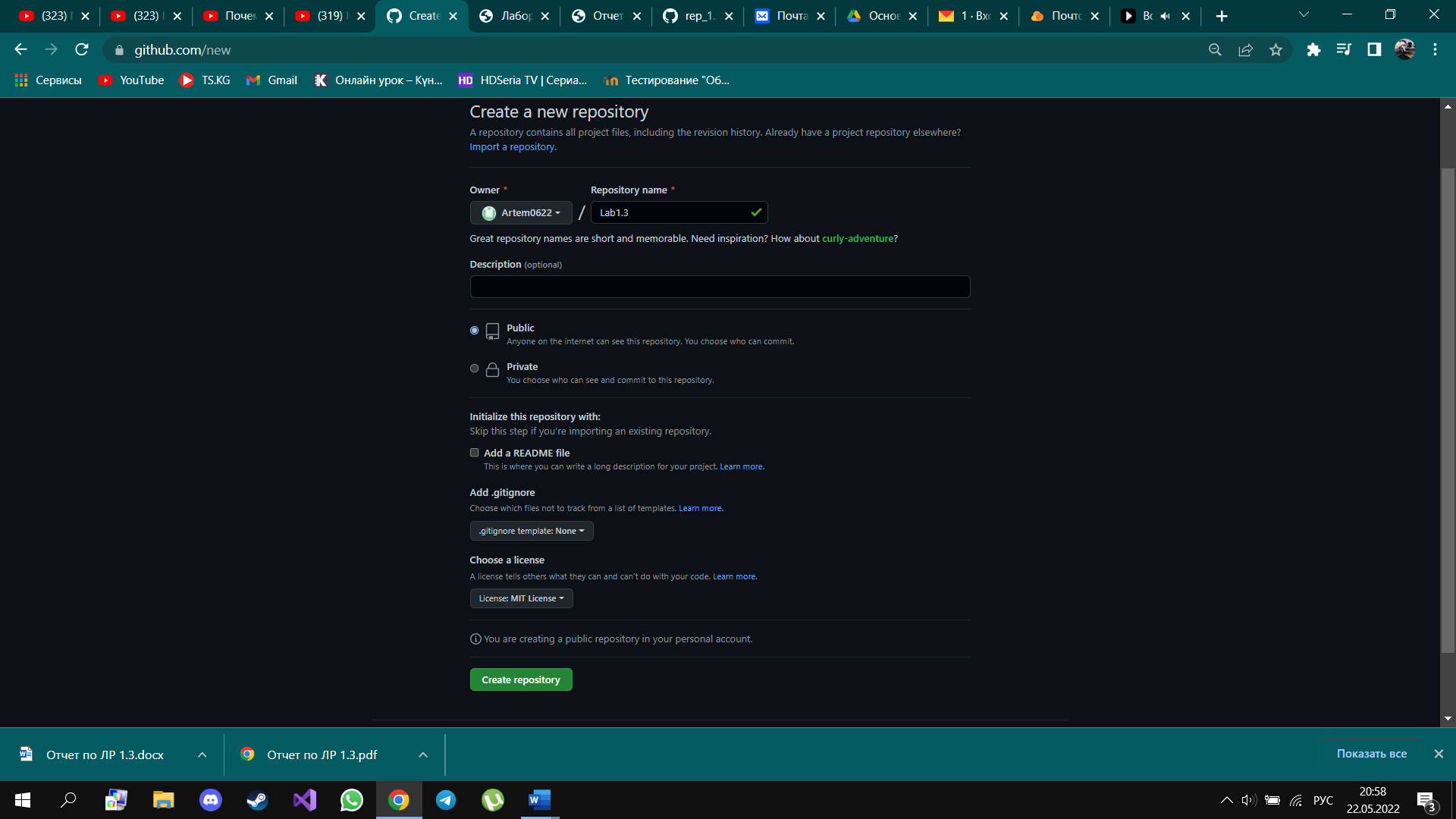


Рисунок 1.1 Создание репозитория

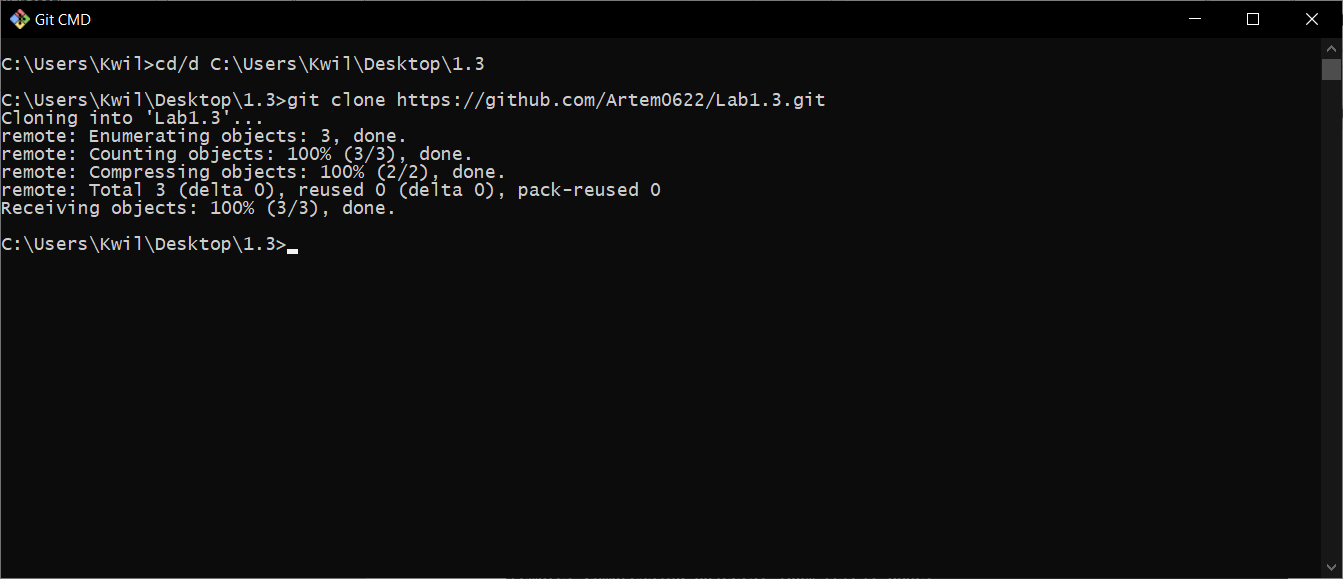


Рисунок 1.2 Клонирование репозитория

2.Создал три файла, проиндексировал 1-й и сделал коммит “add 1.txt file”. Затем проиндексировал 2-й и 3-й и перезаписал коммит с новым комментарием “add 2.txt and 3.txt”:

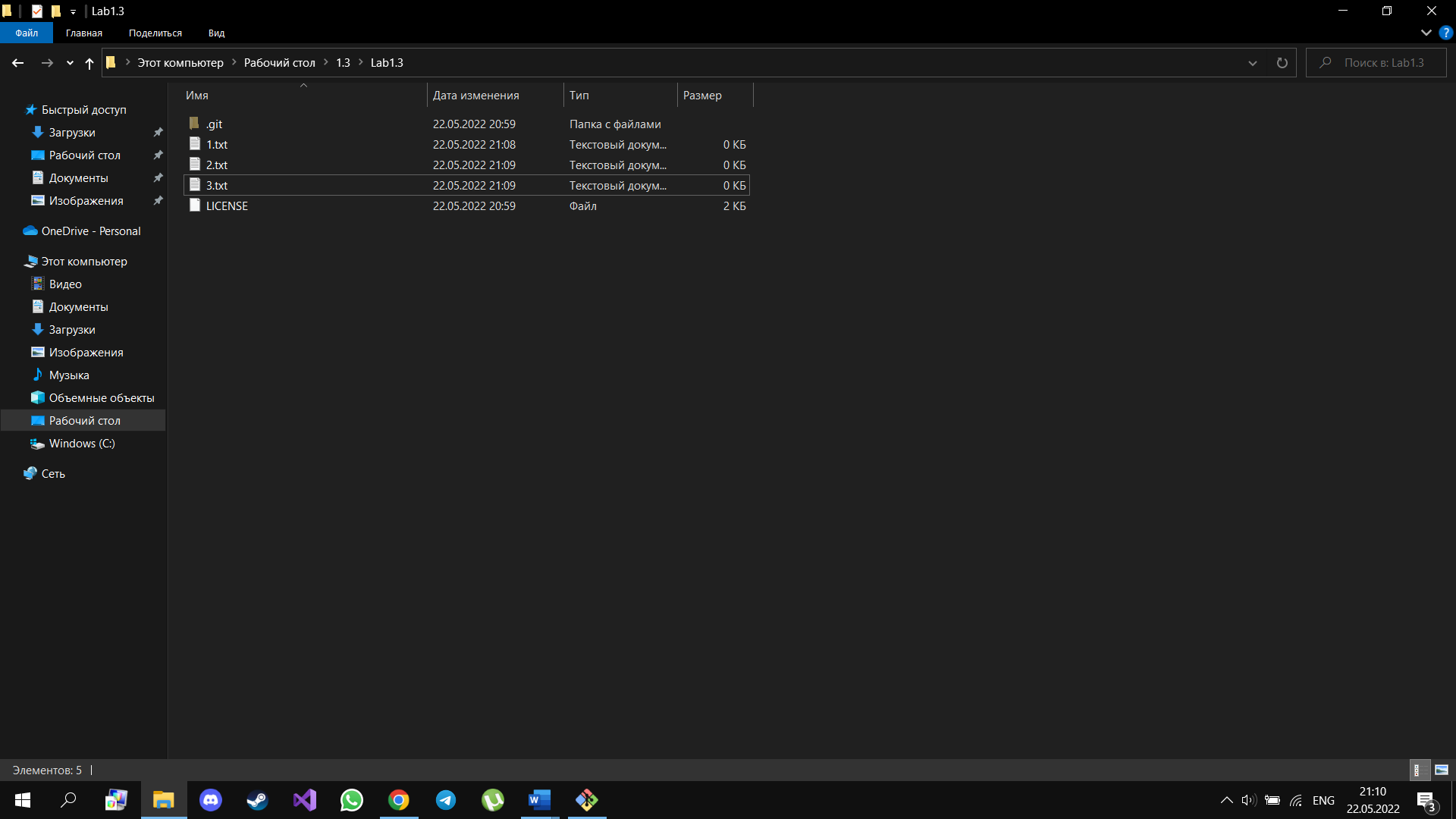


Рисунок 2.1 Созданные файлы в репозитории

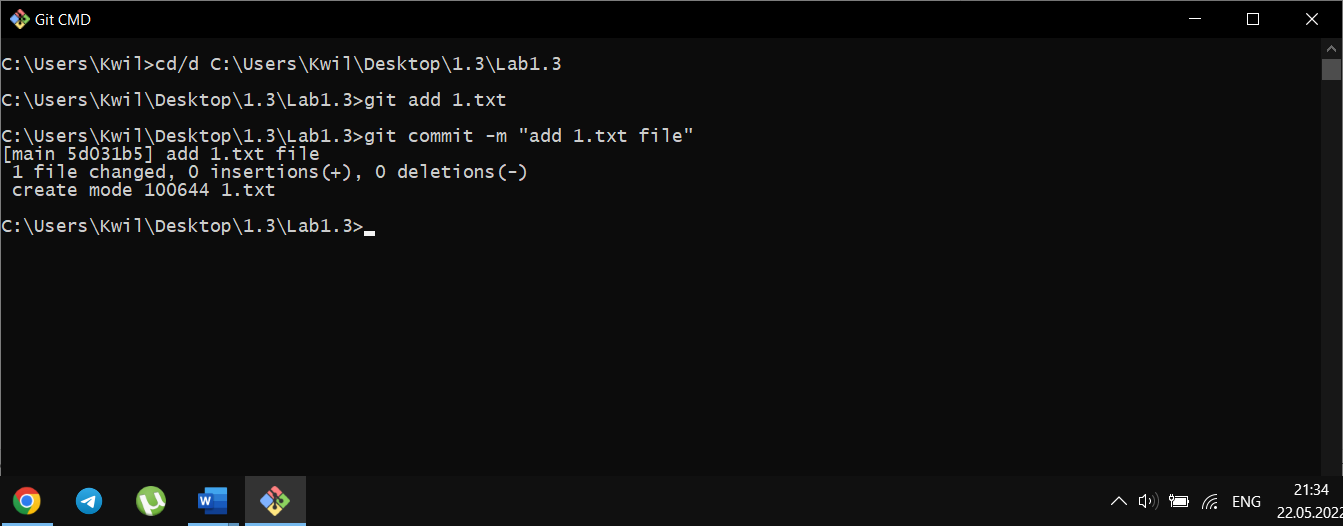


Рисунок 2.2 1-й коммит

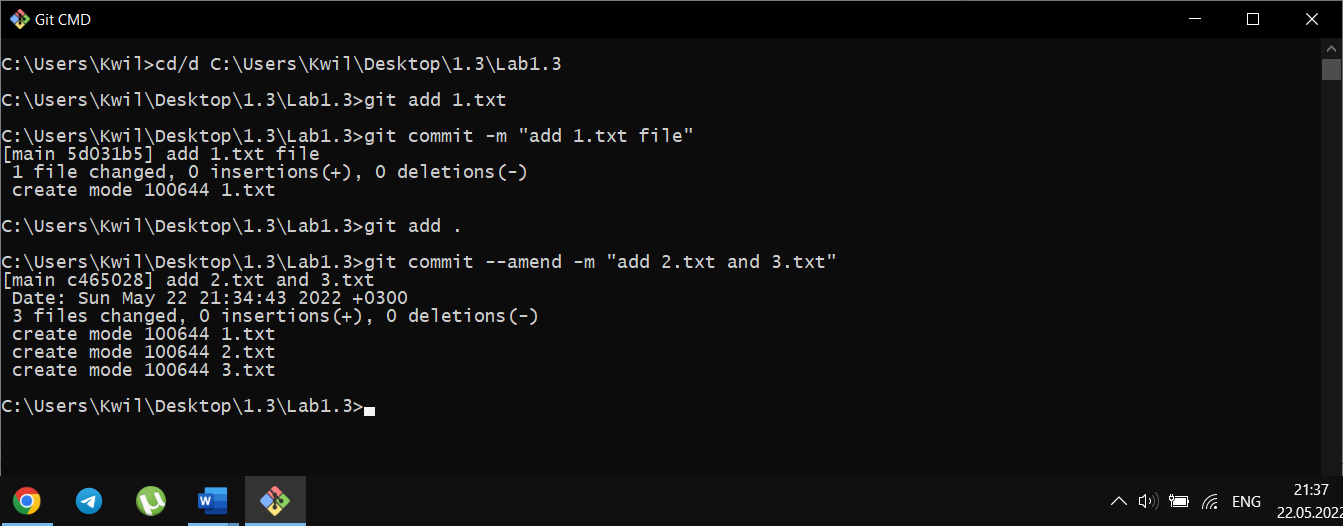


Рисунок 2.3 Изменение 1-го коммита

3.Создал новую ветку my\_first\_branch, перешел на нее и создал в ней новый файл in\_branch.txt, закоммитил изменения. Вернулся на ветку master, создал и сразу перешел на ветку new\_branch, сделал изменения в файле 1.txt и закоммитил изменения.

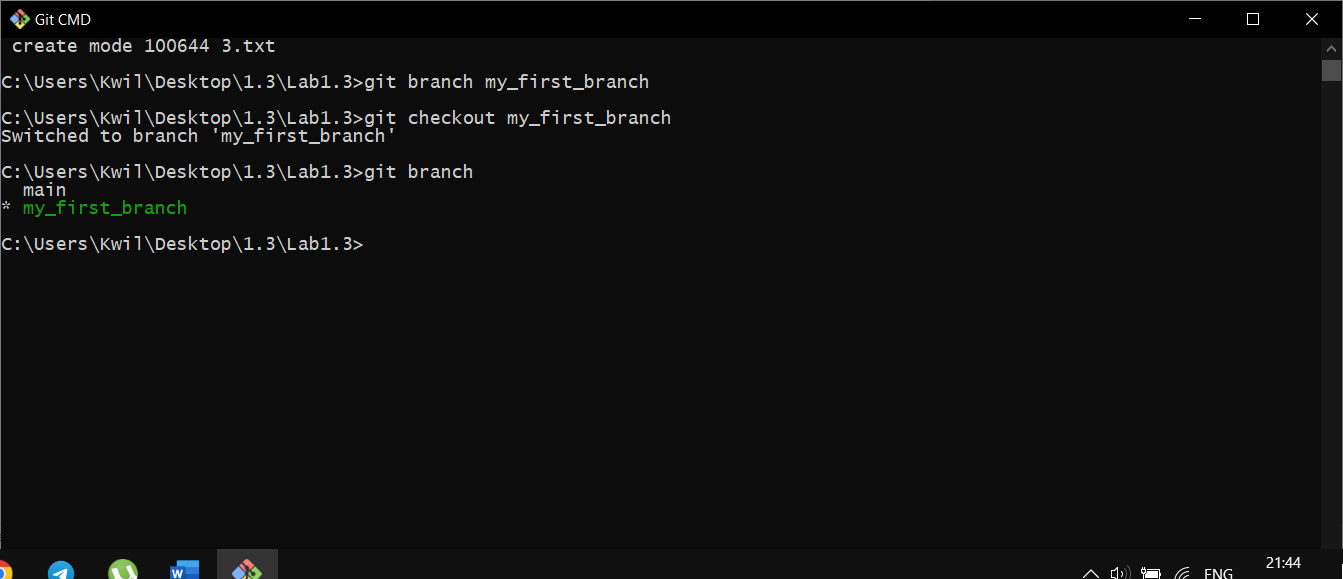


Рисунок 3.1 Создание и переход на новую ветку

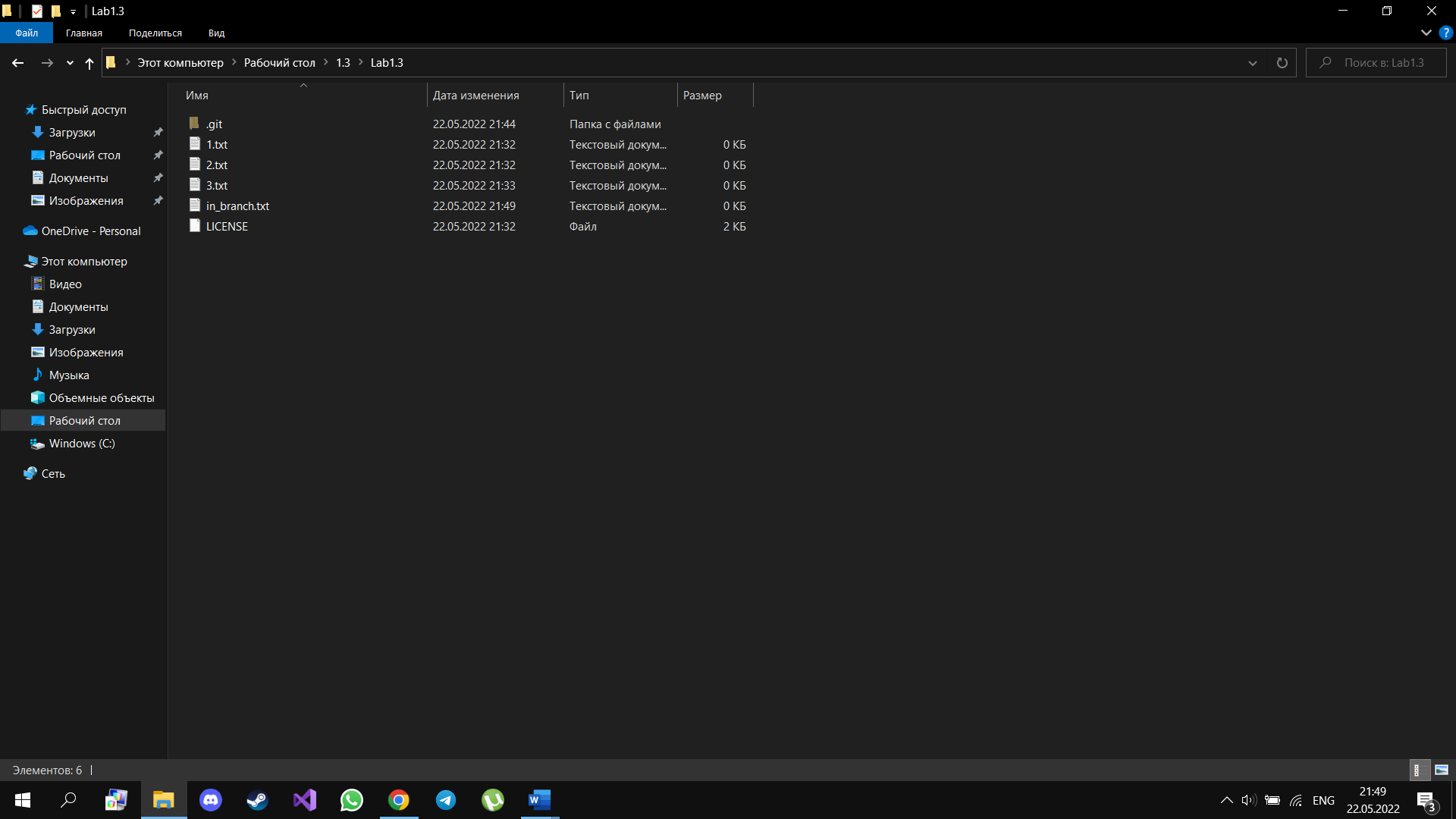


Рисунок 3.2 Добавление файла в репозиторий

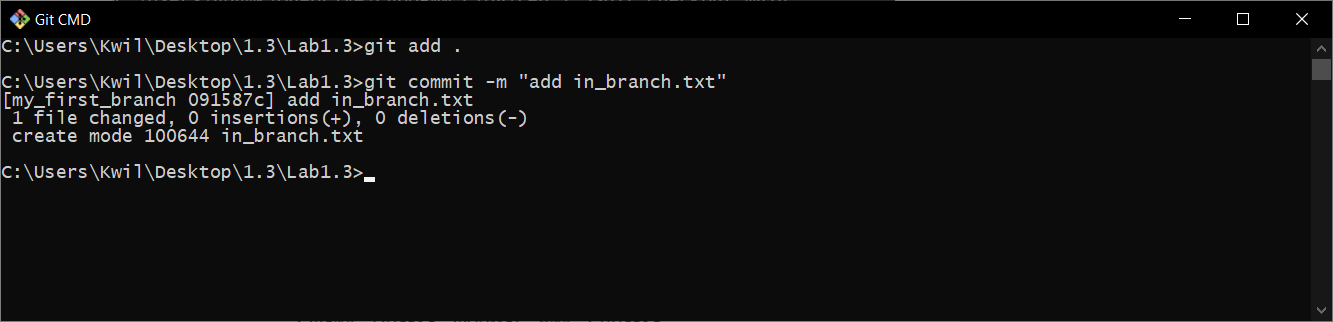


Рисунок 3.3 Коммит изменений

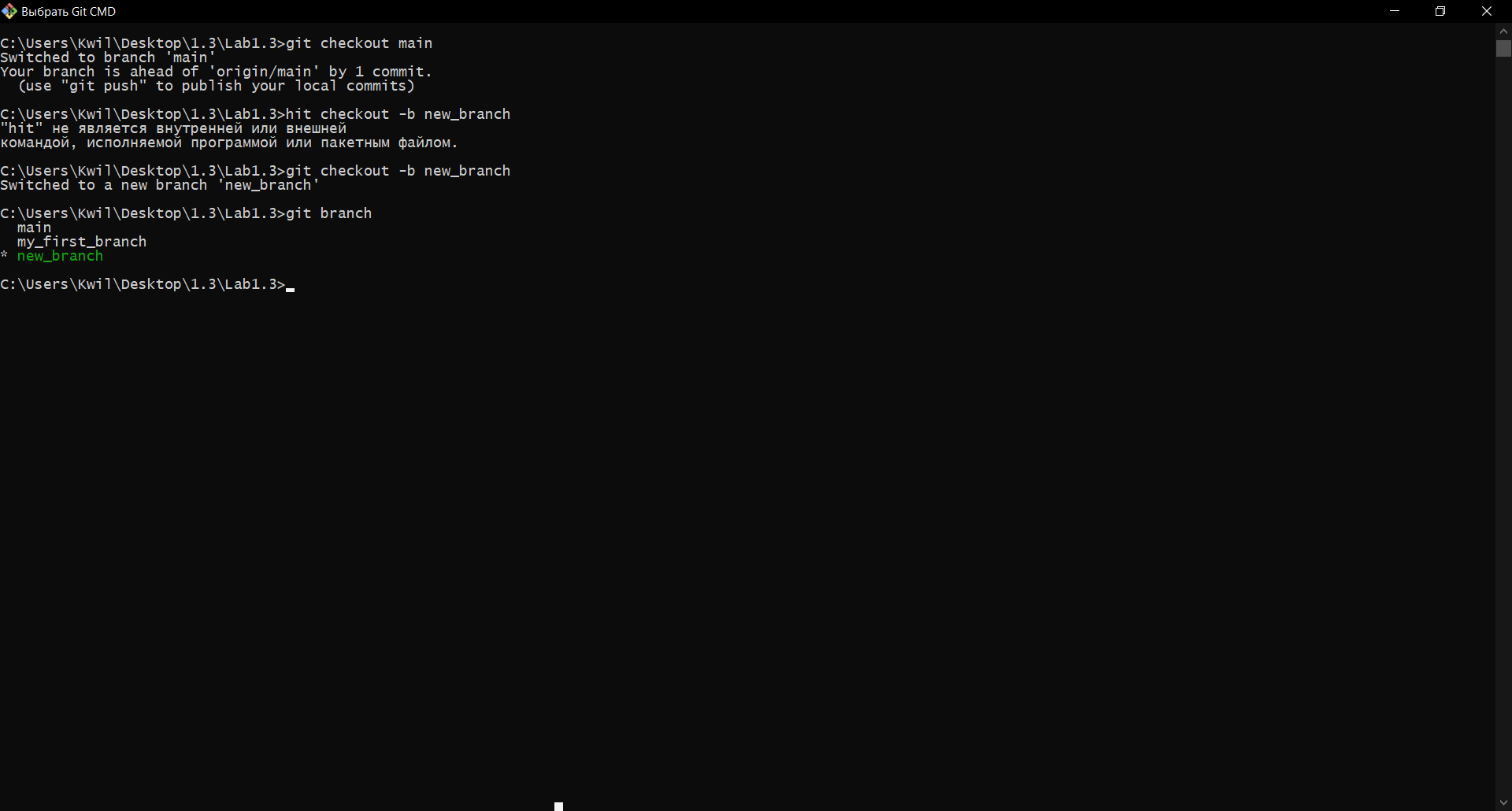


Рисунок 3.4 Переход на ветку main и создание + переход на новую ветку

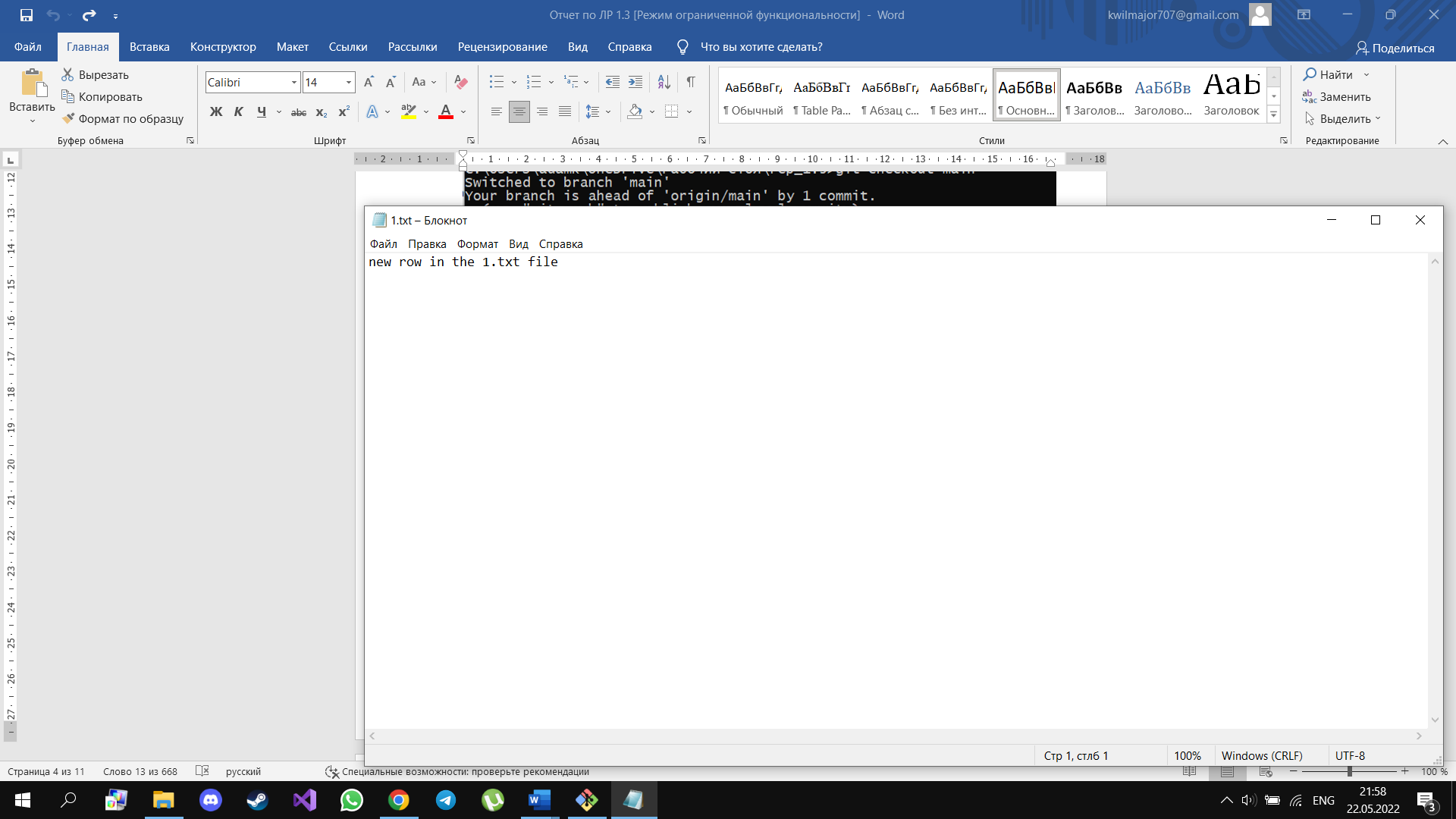


Рисунок 3.5 Изменения в файле 1.txt



Рисунок 3.6 Коммит изменений

4.Перешел на ветку main и слил ветки main и my\_first\_branch, после этого слил ветки main и new\_branch. Удалил все ветки кроме main:

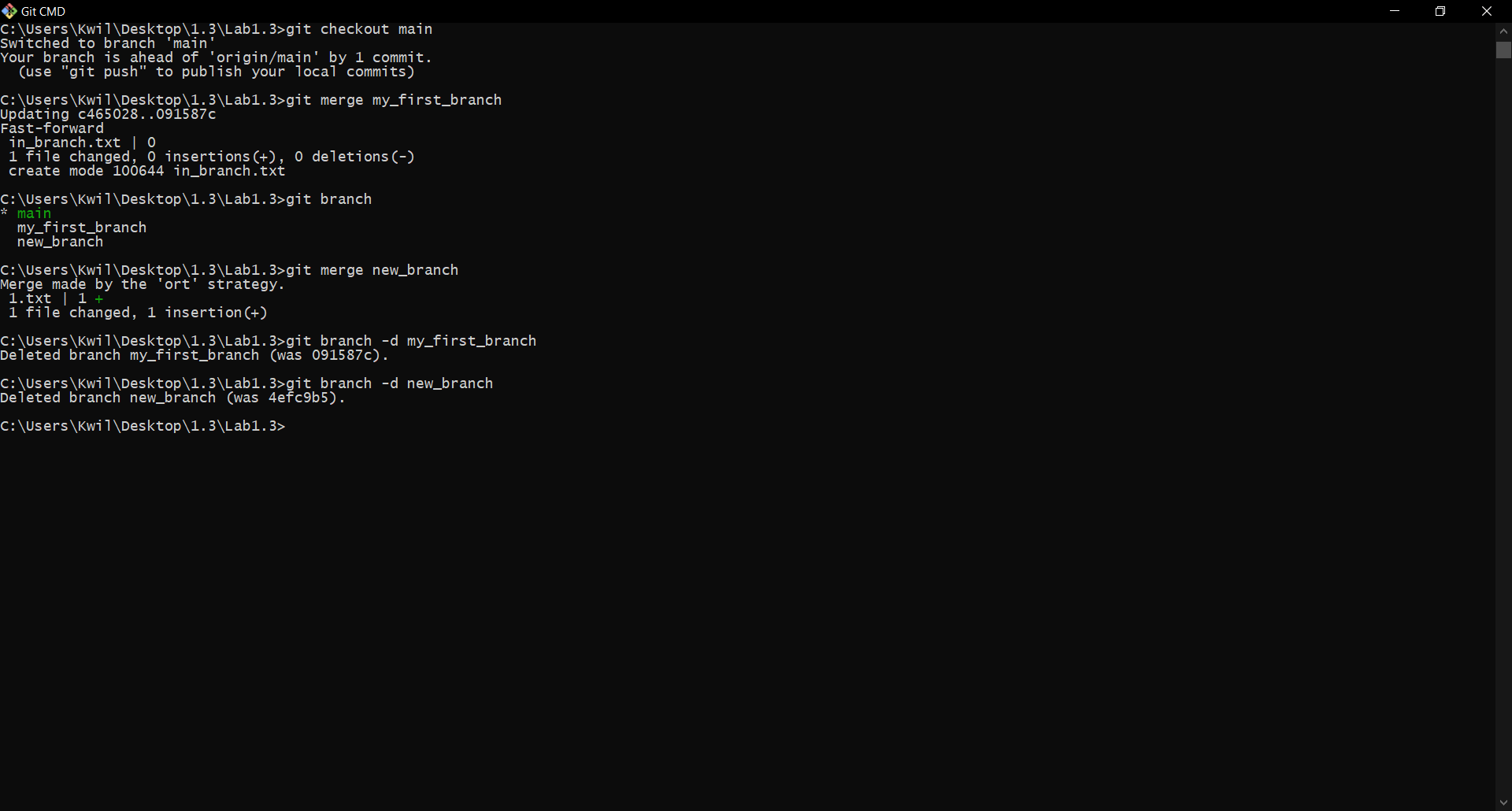


Рисунок 4.1 Слияние и удаления веток

5.Создал ветки branch\_1 и branch\_2. Перешел на ветку branch\_1 и изменил файлы 1.txt, 3.txt, после чего закоммитил изменения. Перешел на ветку branch\_2, изменил файлы 1.txt, 3.txt, после чего закоммитил изменения. После этого слил изменения ветки branch\_2 в ветку branch\_1:

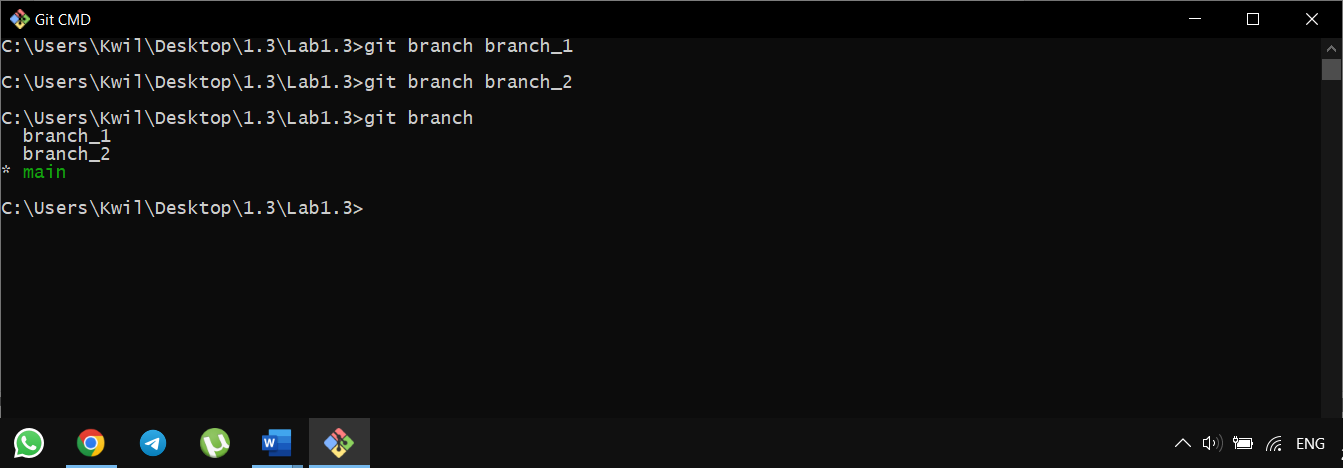


Рисунок 5.1 Создание веток branch\_1 и branch\_2

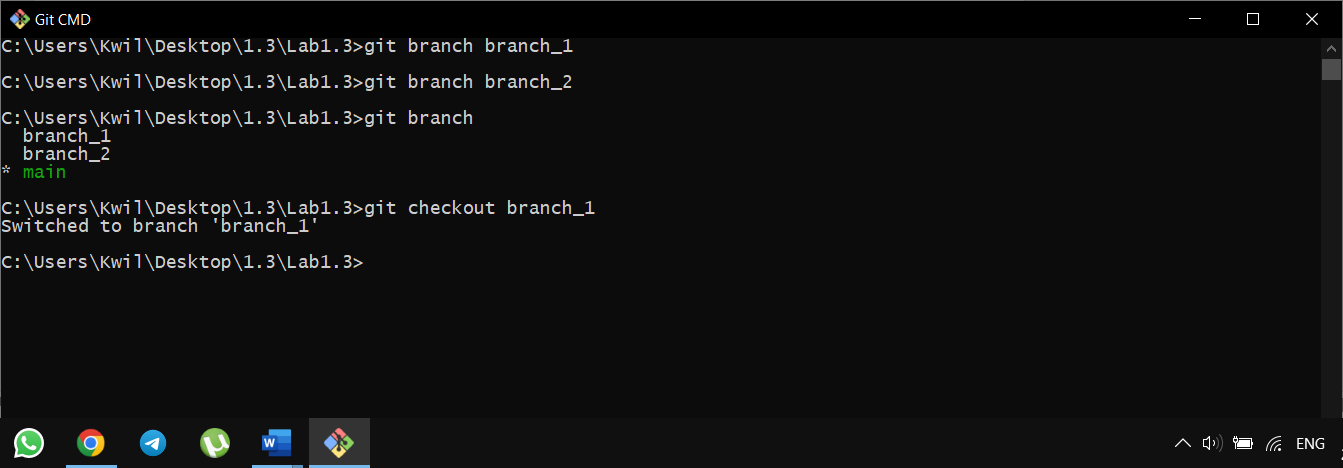
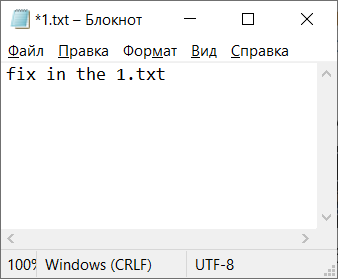


Рисунок 5.2 Переход на ветку branch\_1



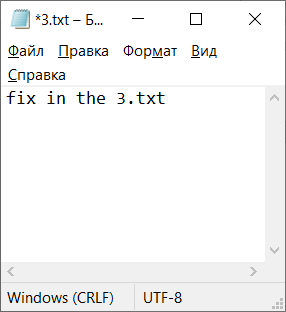


Рисунок 5.3 Изменения в файлах 1.txt и 3.txt



Рисунок 5.4 Коммит изменений

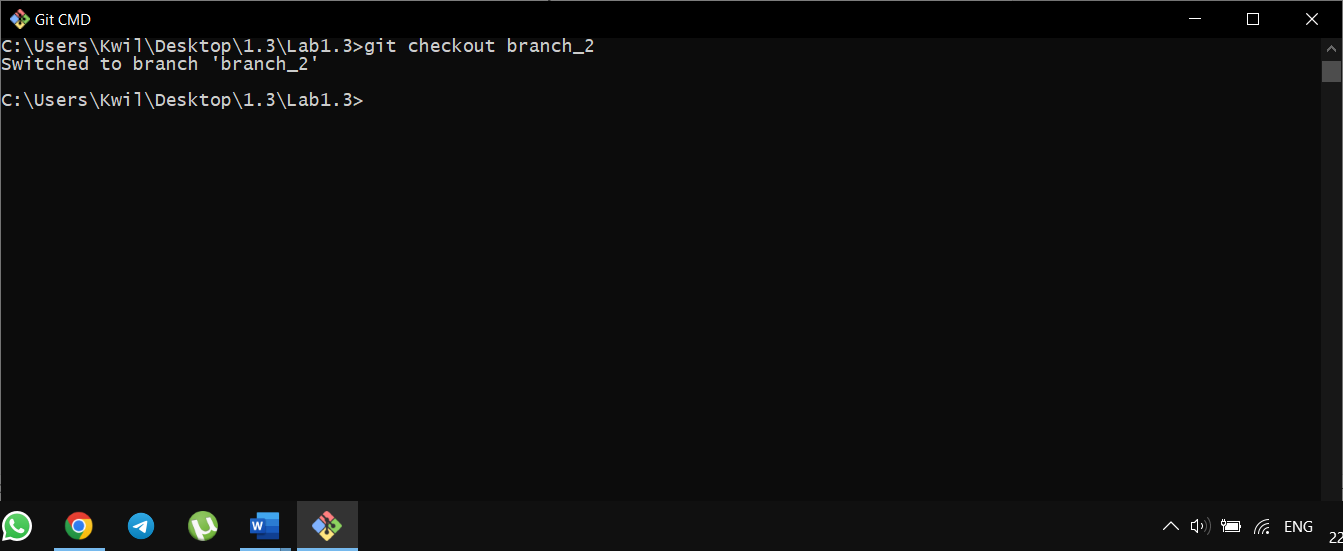


Рисунок 5.5 Переход на ветку branch\_2

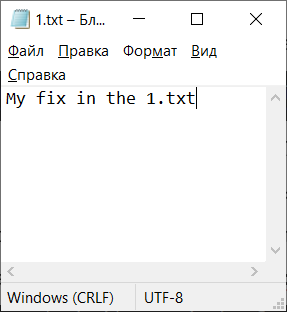
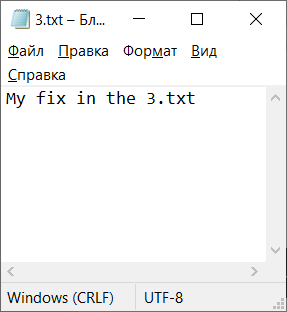
 

Рисунок 5.6 Изменения в файлах 1.txt и 3.txt

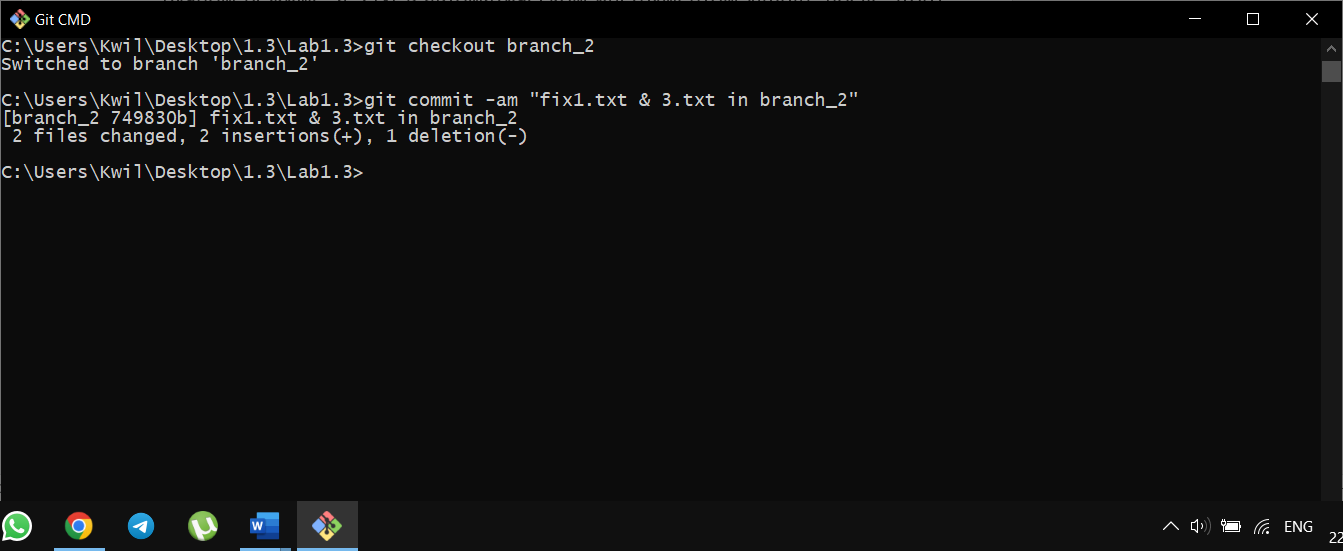


Рисунок 5.7 Коммит изменений

6.Выполнил слияние новых веток и решил конфликт файла 1.txt в ручном режиме, а 3.txt в автоматическом инструментом vimdiff, после этого отправил ветку branch\_1 в GitHub:

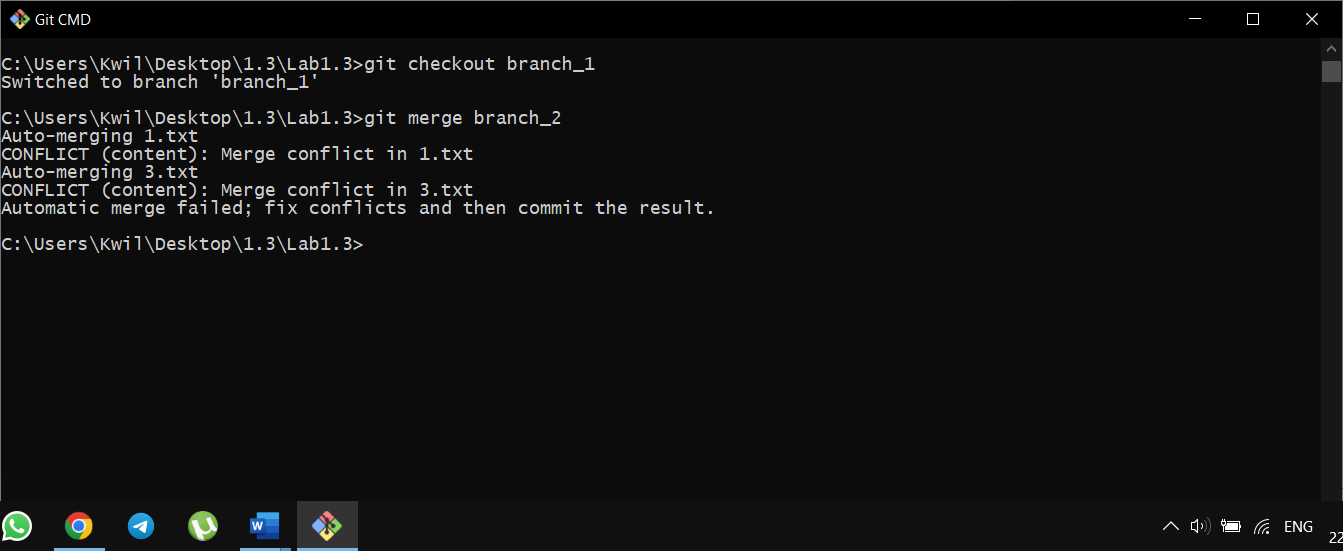


Рисунок 6.1 Слияние веток

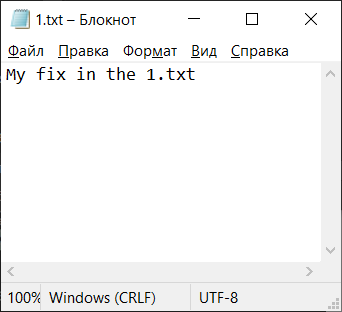


Рисунок 6.2 Решение конфликта в ручном режиме

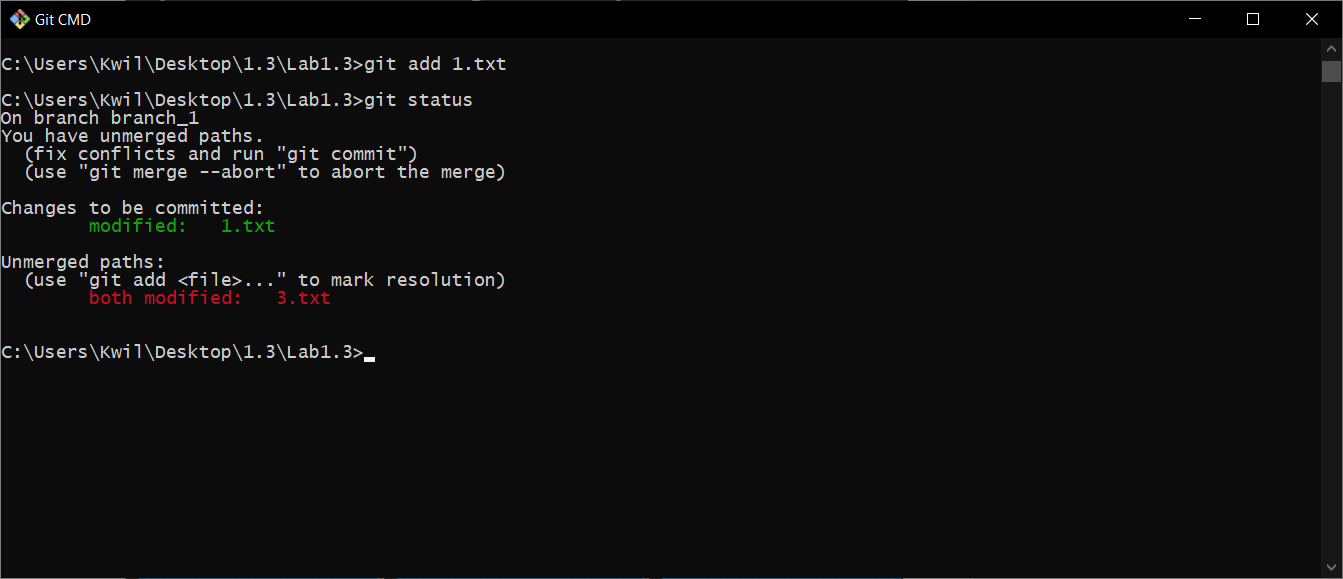


Рисунок 6.3 Добавление 1.txt в индекс, так как конфликт решен

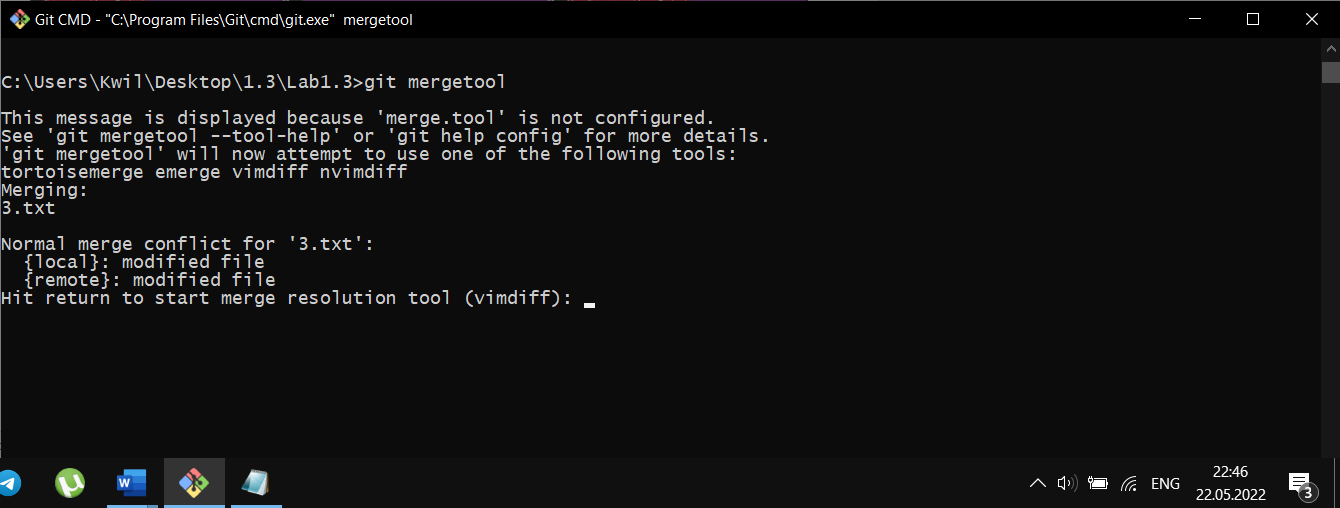


Рисунок 6.4 Команда mergetool

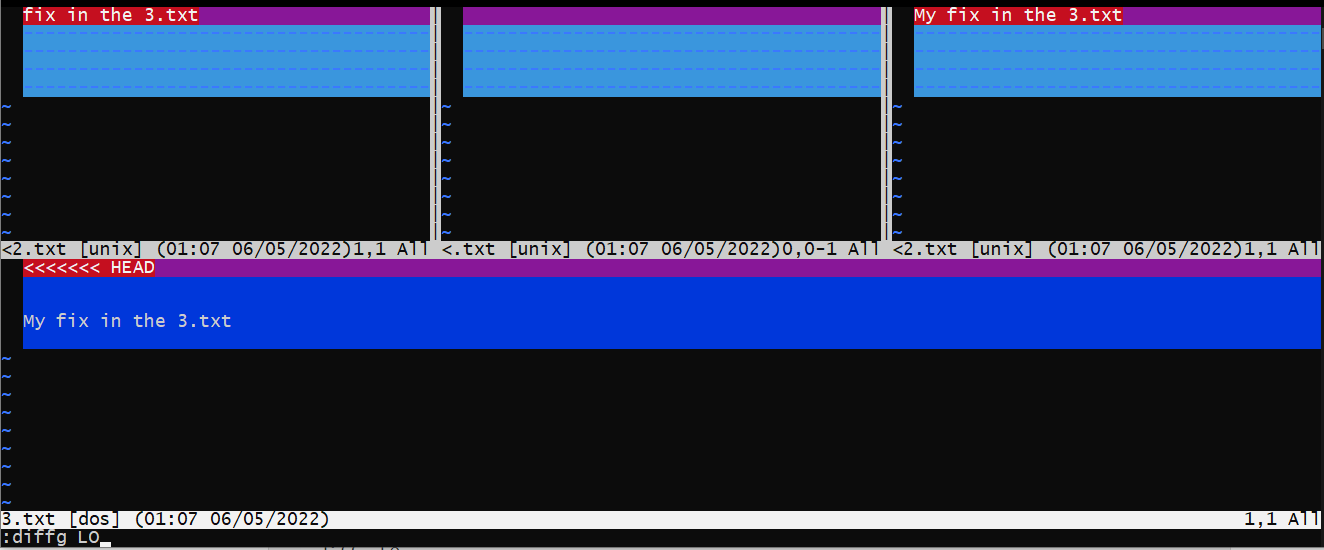


Рисунок 6.5 Команда :diffg LO

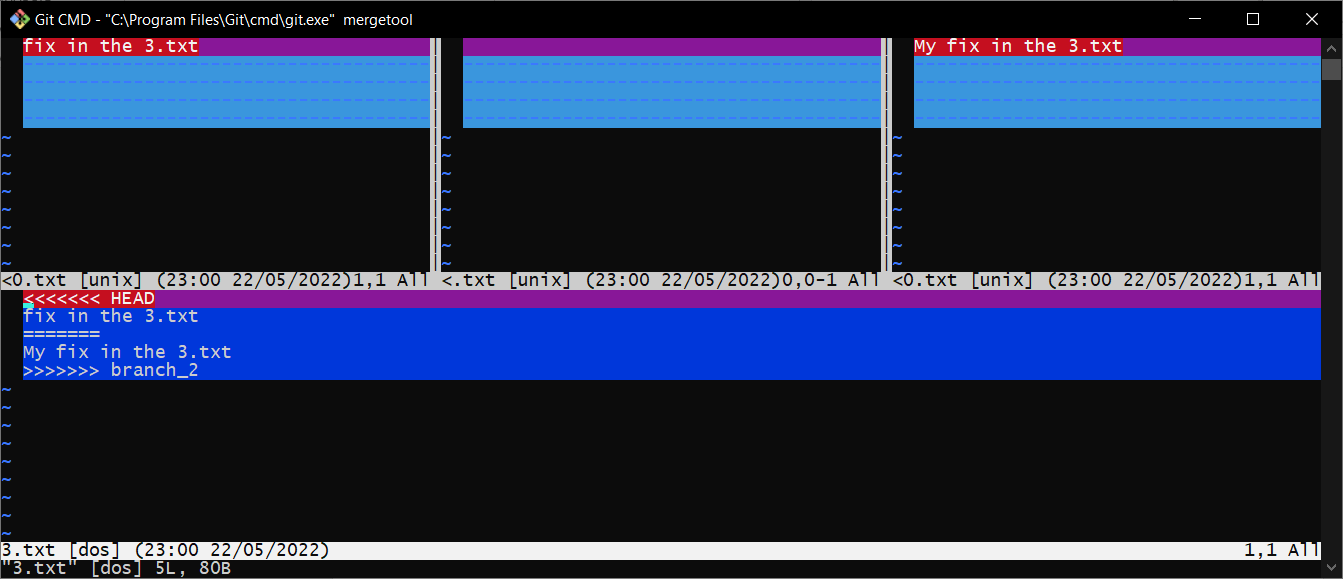


Рисунок 6.6 Команда :wga

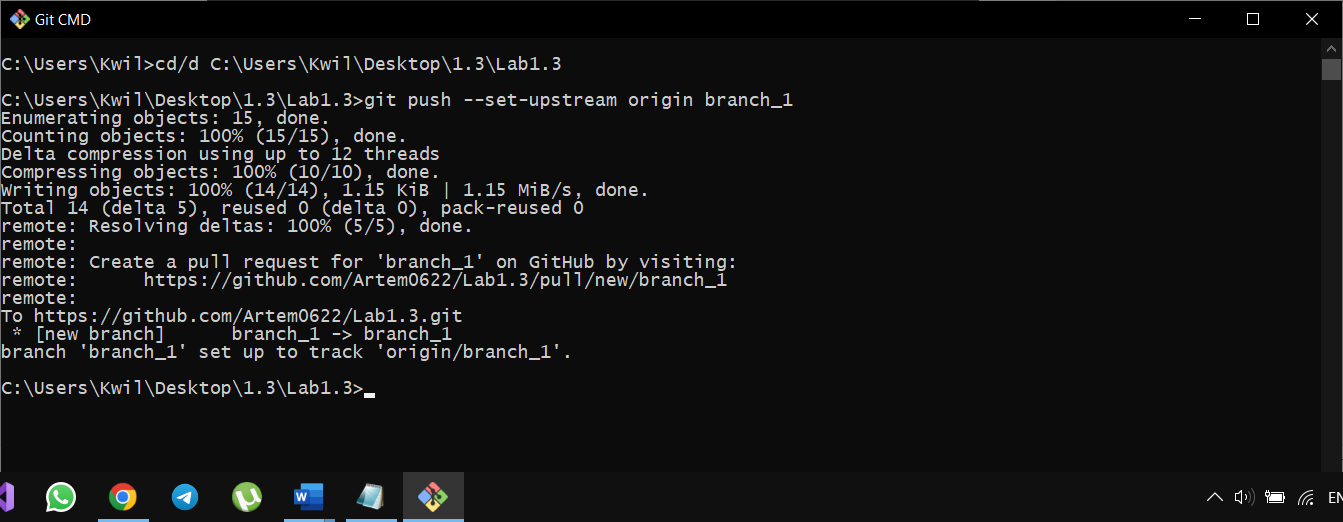


Рисунок 6.7 Пуш ветки на уд. Репозиторий

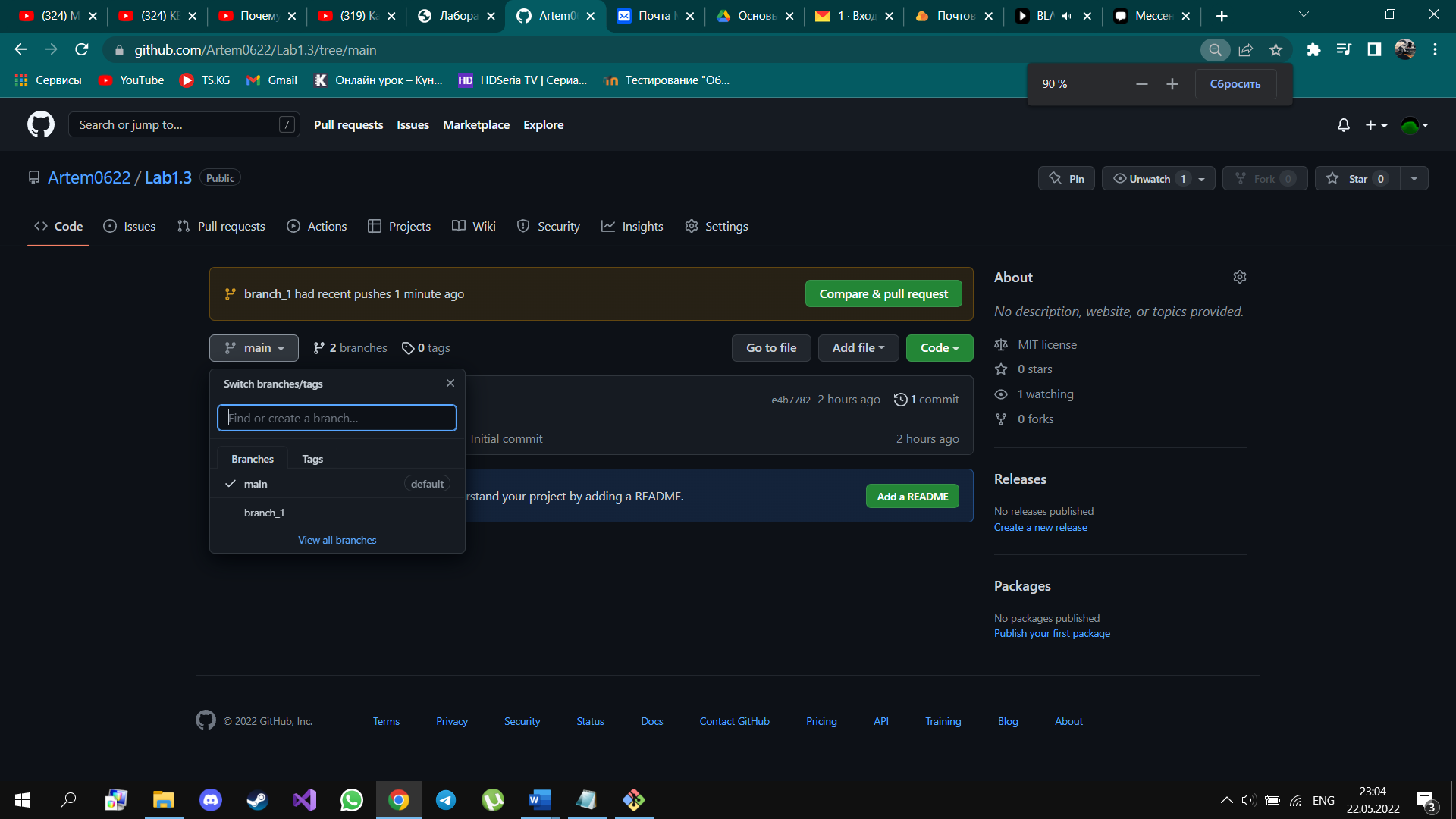


Рисунок 6.8 Изменения на уд. репозитории

7.Создал средствами Git удаленную ветку branch\_4, создал в локальном репозитории ветку отслеживания удаленной ветки branch\_3. Перешел на ветку branch\_3 и добавил в файл 2.txt строку “the final fantasy in the 4.txt file”:

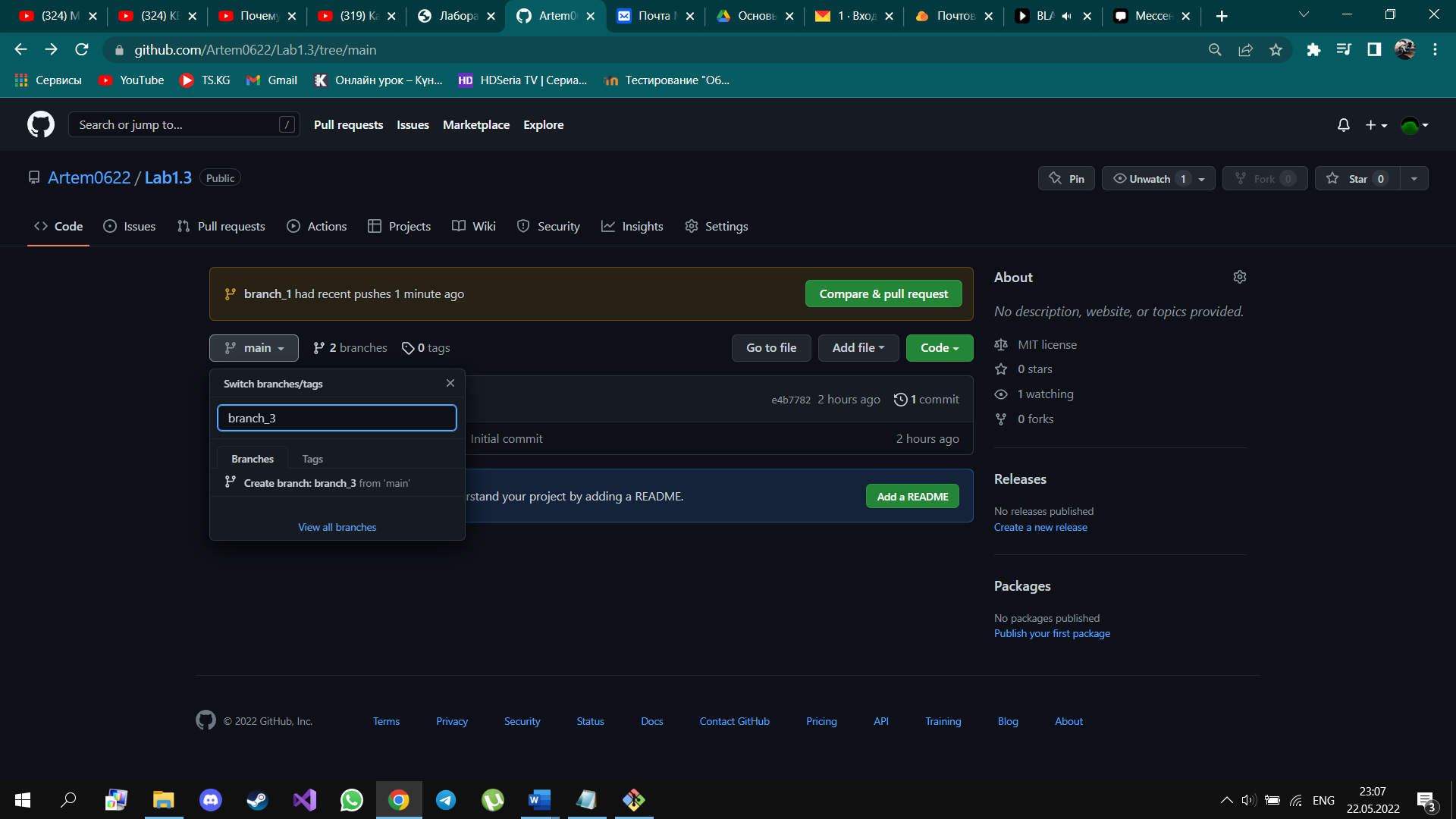


Рисунок 7.1 Создание уд. ветки branch\_3

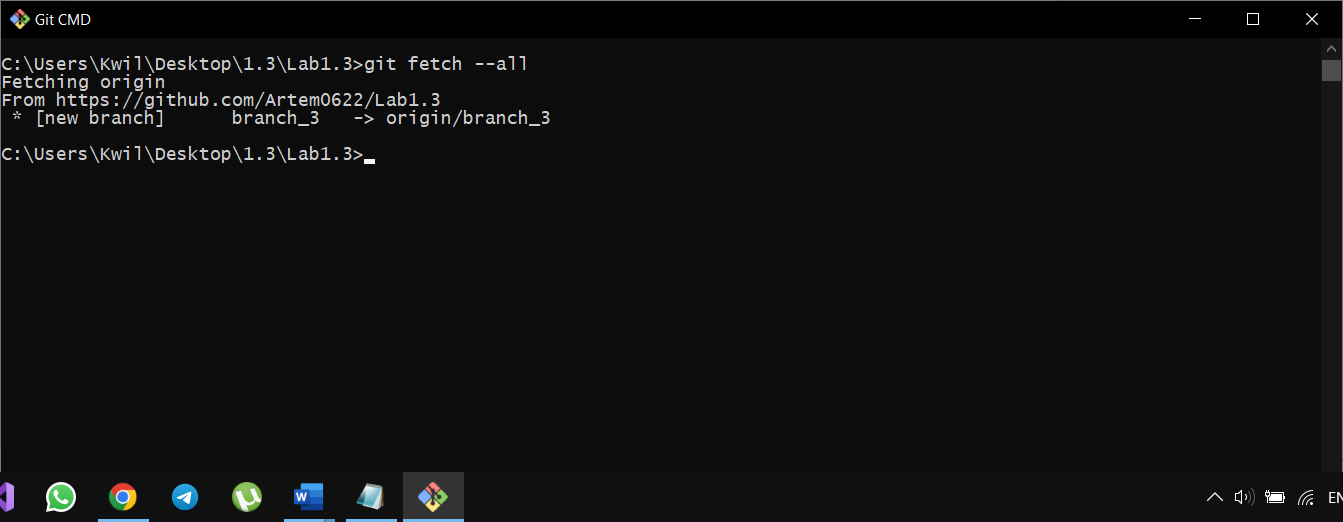


Рисунок 7.2 Загрузка содержимого из уд. репозитория перед отслеживанием

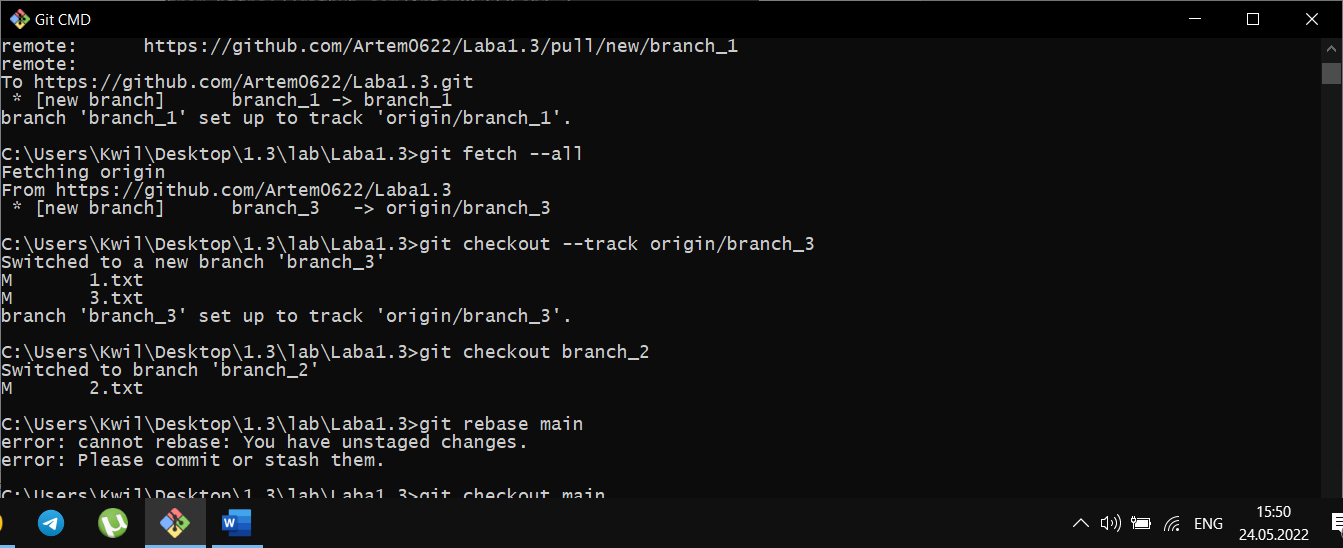


Рисунок 7.3 Создание ветки отслеживания branch\_3

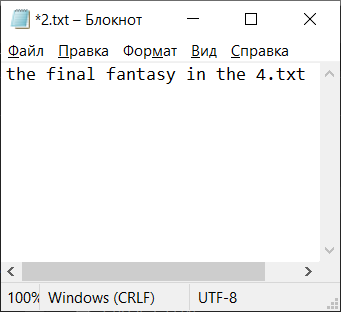


Рисунок 7.4 Добавление в 2.txt строку “the final fantasy in the 4.txt file” на ветке branch\_3

8.Выполнил перемещение ветки main на ветку branch\_2 и отправил изменения веток main и branch\_2 на GitHub:

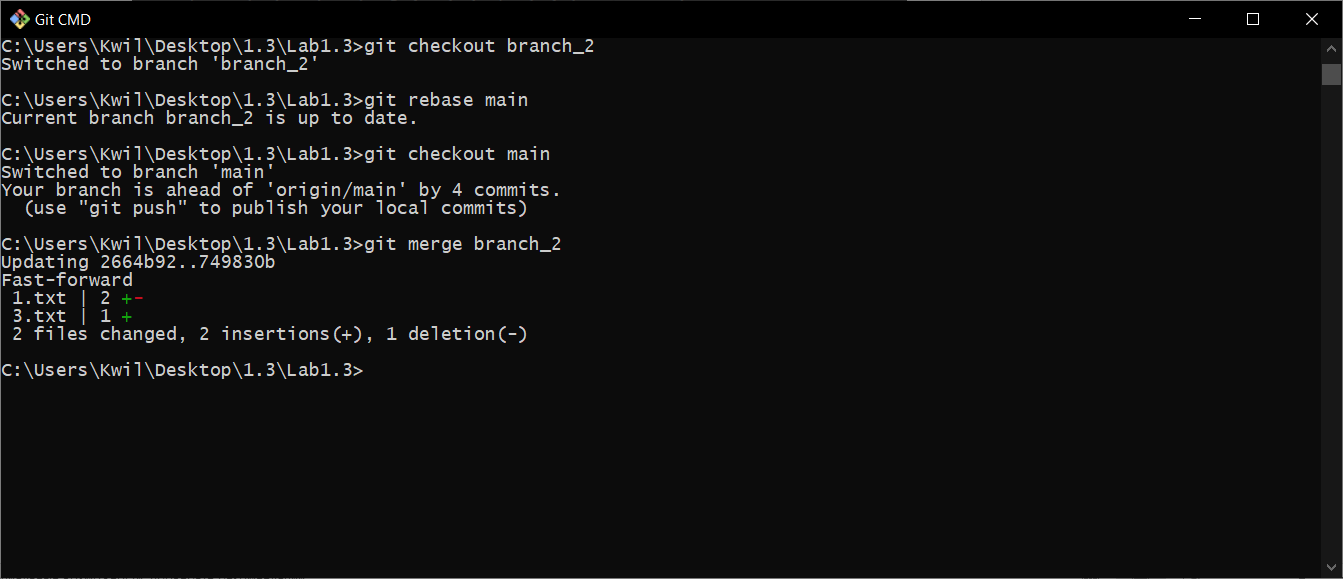


Рисунок 8.1 Результат выполнения команд

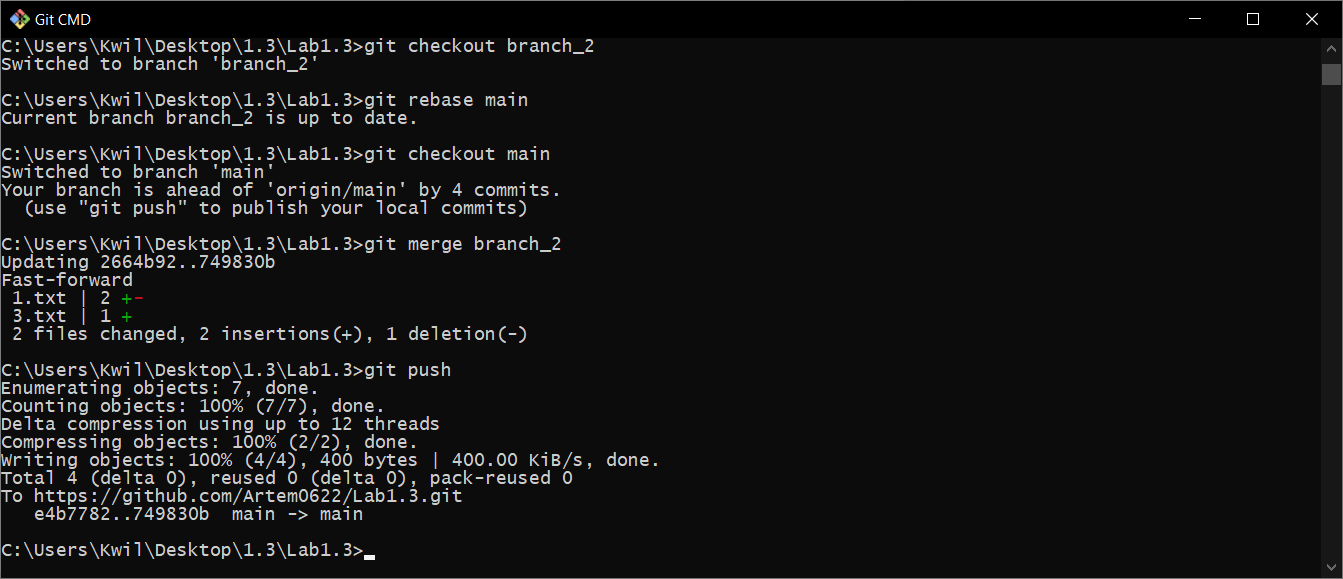


Рисунок 8.2 Пуш веток

**Ответы на контрольные вопросы:**

1. **Что такое ветка?**

Это последовательность коммитов, отклоняющаяся от основной линии разработки.

1. **Что такое HEAD?**

Это указатель, задача которого ссылаться на определённый коммит в репозитории, если точнее: на коммит, который станет родителем для следующего коммита.

1. **Способы создания веток?**

Новую ветку можно создать командой git branch <название\_ветки> или на удаленном репозитории git hub.

1. **Как узнать текущую ветку?**

При помощи команды git branch.

1. **Как переключаться между ветками?**

При помощи команды git checkout <название\_ветки>.

1. **Что такое удаленная ветка?**

Это ветка, находящаяся на удаленном репозитории. Или ссылка на состояние ветки на удаленном репозитории.

1. **Что такое ветка отслеживания?**

Это ветка в локальном репозитории, которая напрямую связана с удаленной веткой на удаленном репозитории.

1. **Как создать ветку отслеживания?**

Командой git checkout --track origin/<название\_ветки>.

1. **Как отправить изменения из локальной ветки в удаленную ветку?**

Командой git push origin <название\_ветки>.

1. **В чем отличие команд git fetch и git pull?**

Git pull – это сочетание команд git fetch (получение изменений с удаленного репозитория) и git merge (объединение веток).

1. **Как удалить локальную и удаленную ветки?**

Используя команду git branch -d <название\_ветки>. Для удаление удаленной ветки существует команда git push origin -d <название\_ветки>.