|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет  имени Н. Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н. Э. Баумана)** |

|  |  |
| --- | --- |
| ФАКУЛЬТЕТ | «Информатика и системы управления» (ИУ) |

|  |  |
| --- | --- |
| КАФЕДРА | «Информационная безопасность» (ИУ8) |

Лабораторная работа № 2

ПО КУРСУ

«Технологии и методы программирования»

на тему «Изучение систем контроля версий на примере Git»

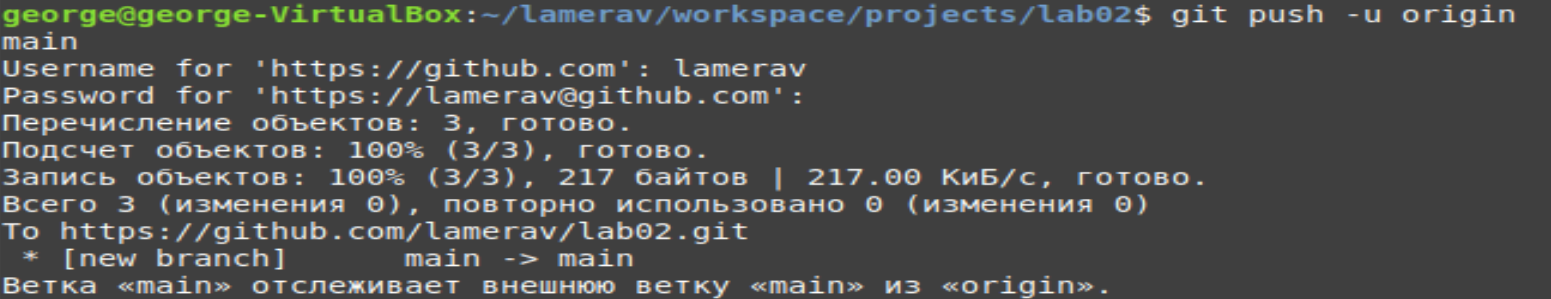
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ8-24 |  |  |  | Г. М. Белов |
|  | (Группа) |  |  |  | (И. О. Фамилия) |
|  |  |  |  |  |  |
| Преподаватель: |  |  |  |  | А. В. Кодык |
|  |  |  |  |  | (И.О. Фамилия) |

2022 г.

**Part 1**

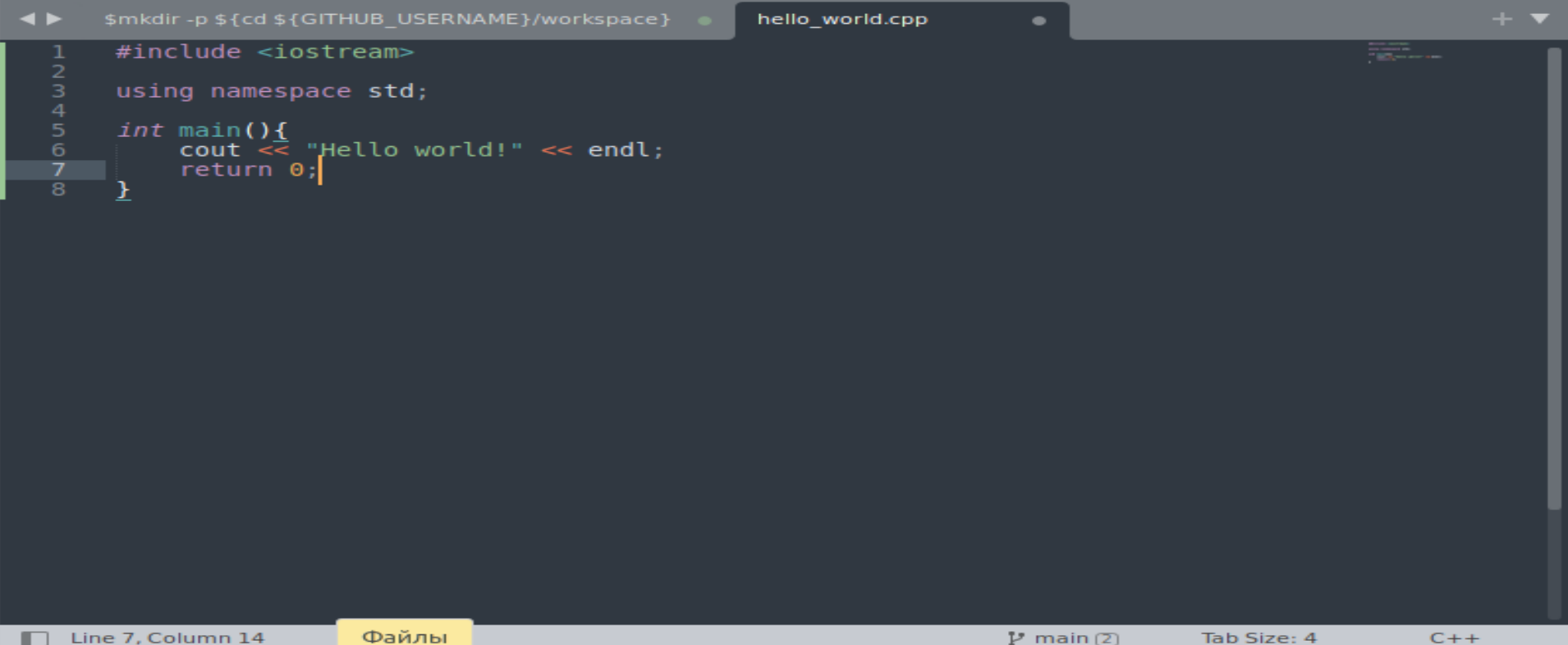
1. Создание репозитория на сервисе github.com

2. Так как я для того чтобы разобраться выполнял часть Tutorial, и для примера я просто обновлю файл README.md, и в commit укажу что это первый commit



3. Создаем и изменяем файл hello\_world.cpp





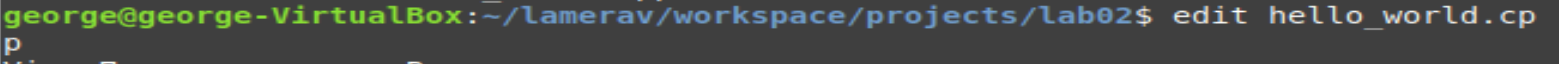
4. Добавляем файл в локальную копию репозитория.



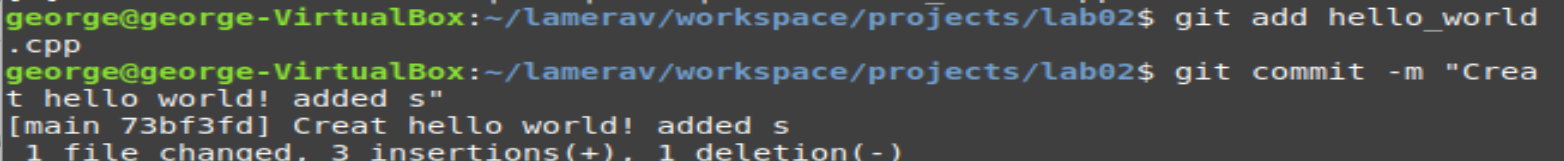
5. Сommit изменений



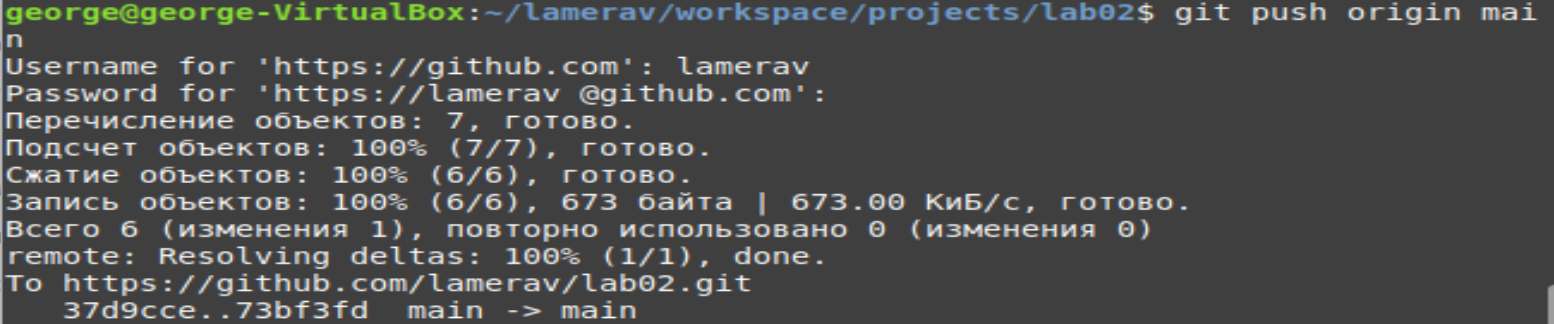
6. Изменение файла hello\_world.cpp



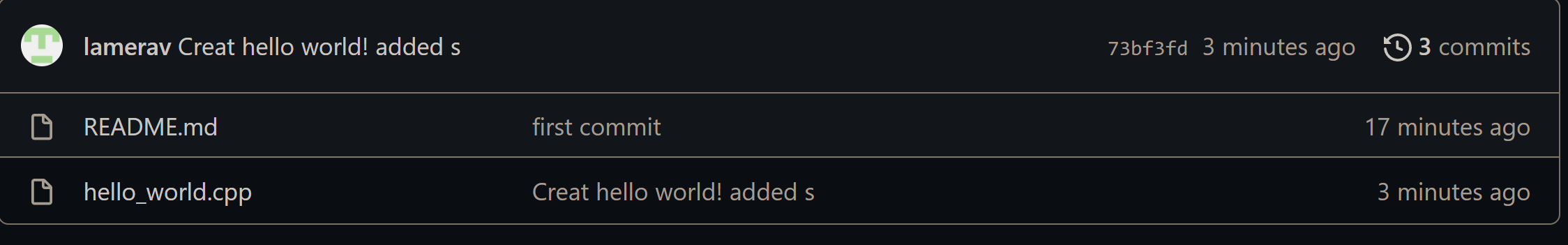
7. Commit измененного файла. Можно не писать git add если использовать для комманды git commit флаг -a который автоматически добавляет все изменения. Иначе у меня выдает ошибку, поэтому я сделал commit классическим способом с git add.



8. Запушим изменения в удаленный репозиторий.



9. История коммитов доступна



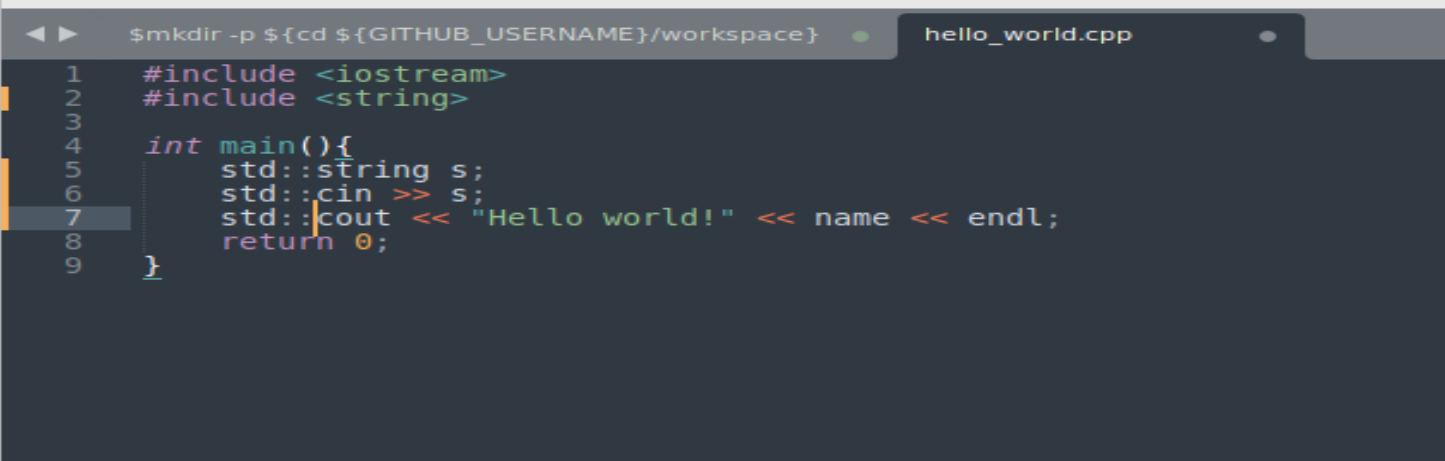
**Part 2**

1. Создание локальной ветки patch1

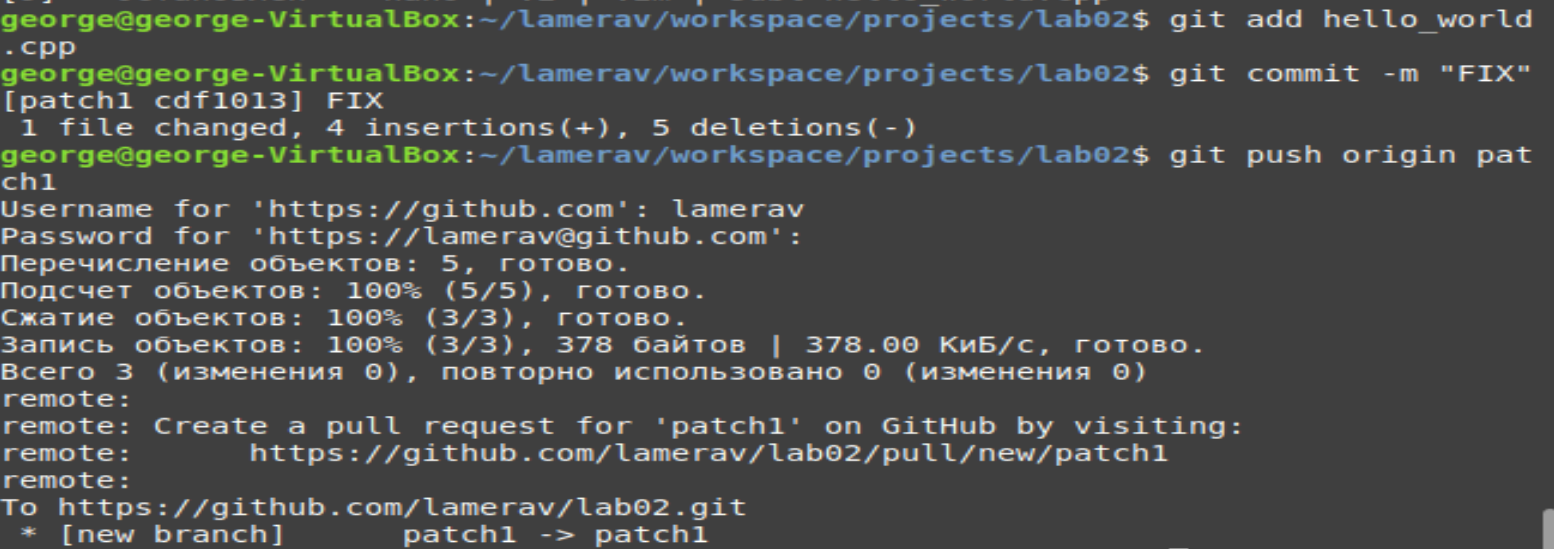


2. В новой ветке исправляем hello\_world.cpp

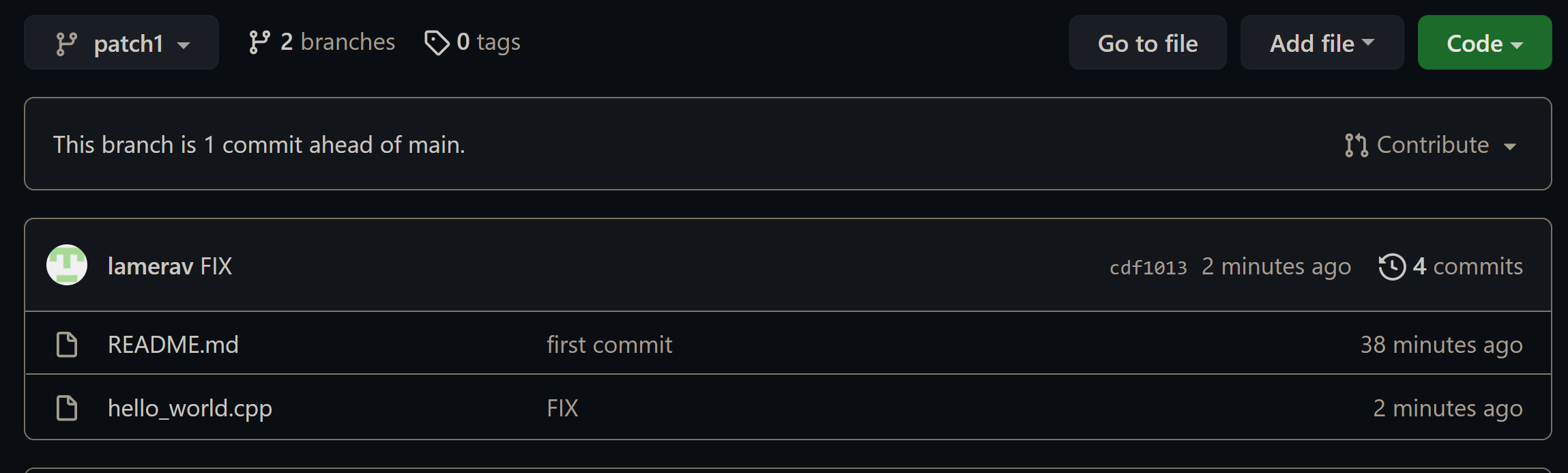




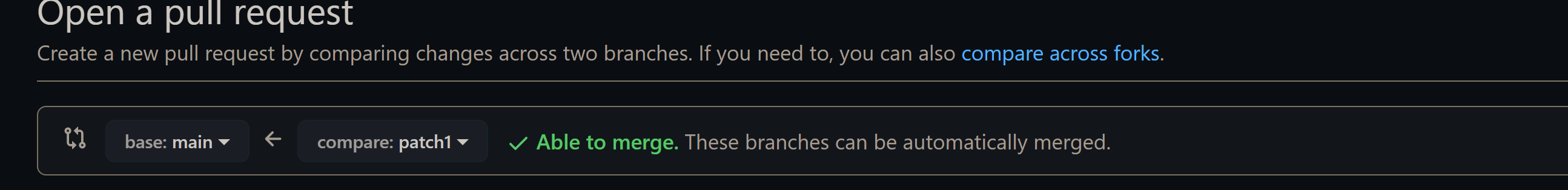
3. Commit, push локальную ветку в удаленный репозиторий



4. Ветка patch1



5. Созданный pull-request ветки patch1 → main



6. Добавляем комментарии в hello\_world.cpp

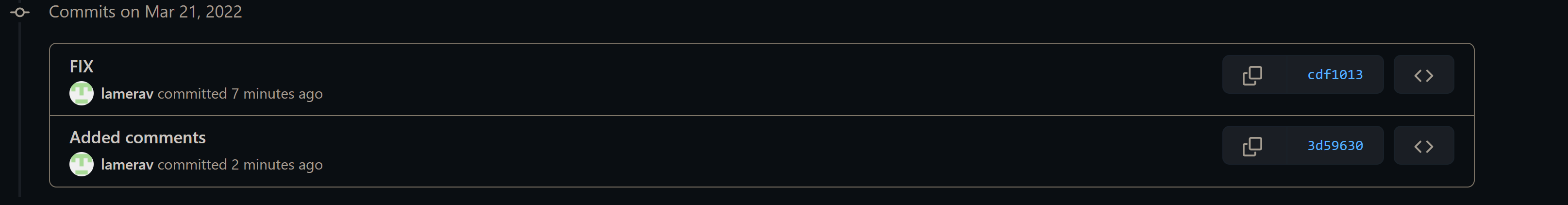




7. Commit, push



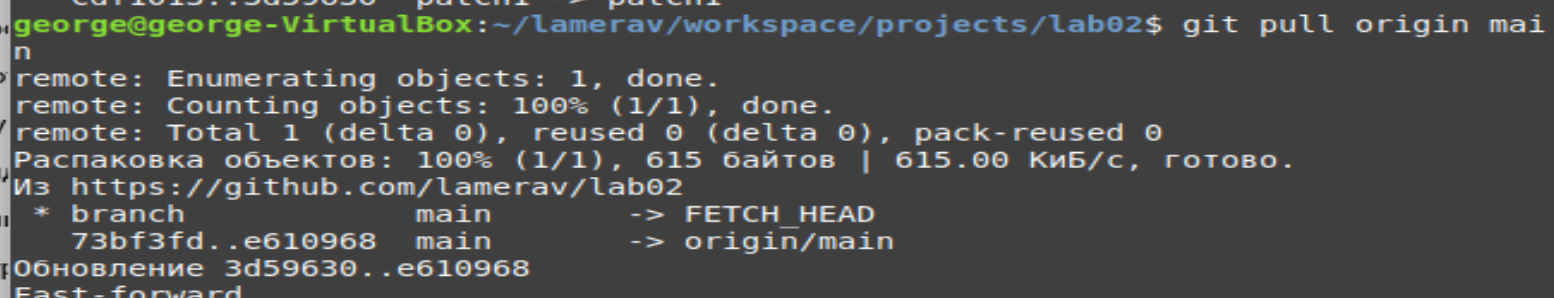
8. Изменения есть



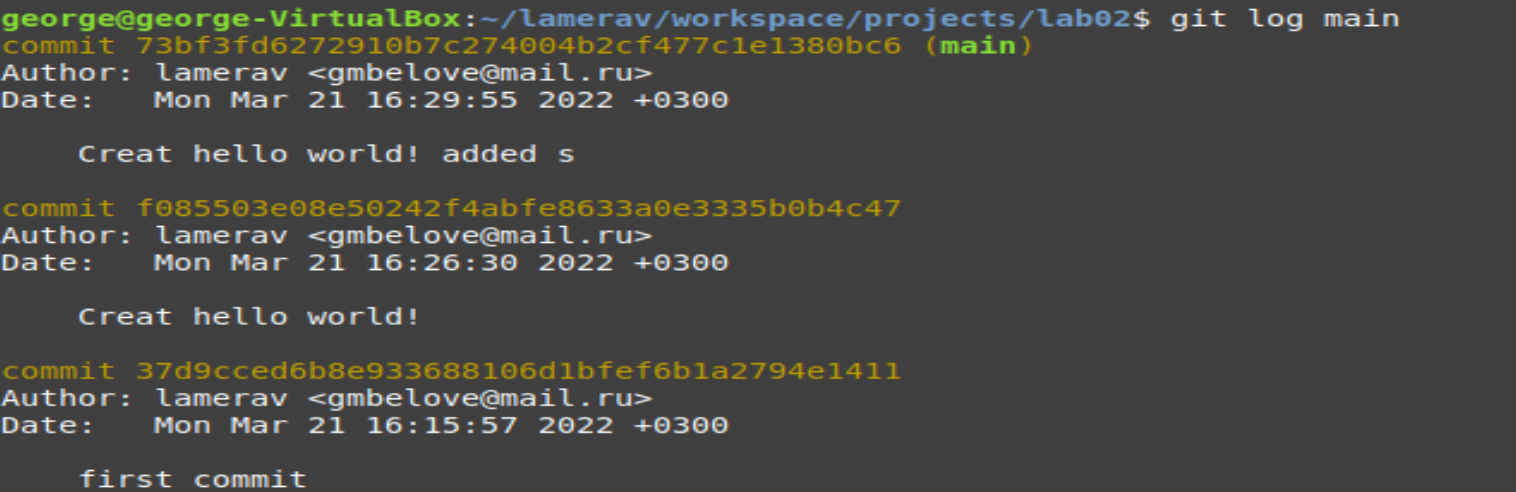
9. Слияние и удаление ветки patch1



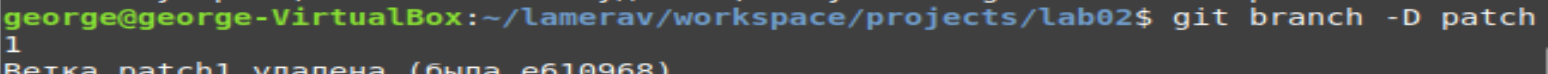
10. Pull



11. История ветки main



12. Удаление локально ветки patch1

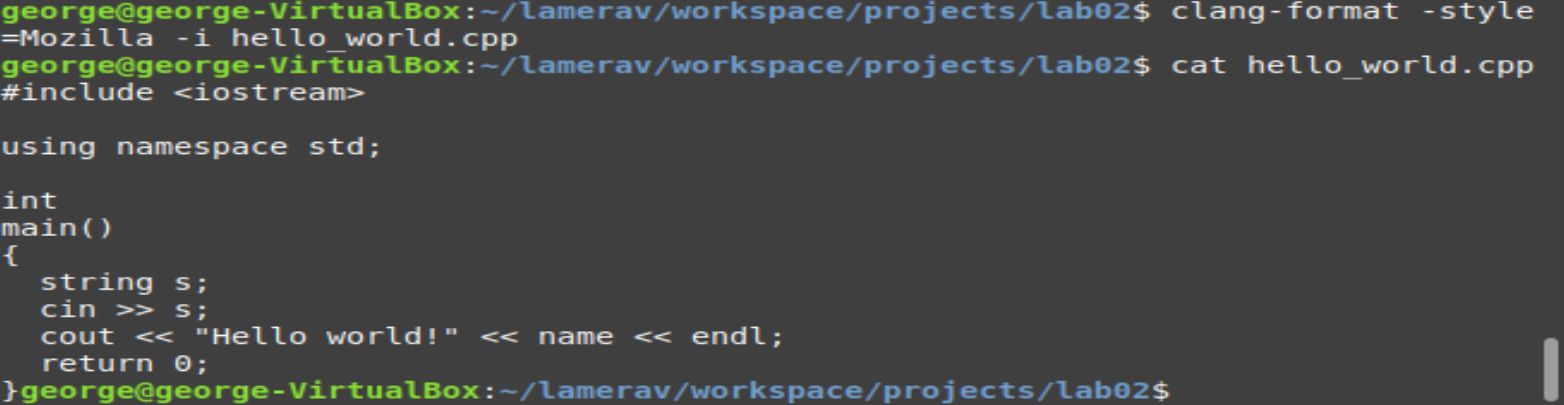


**Part 3**

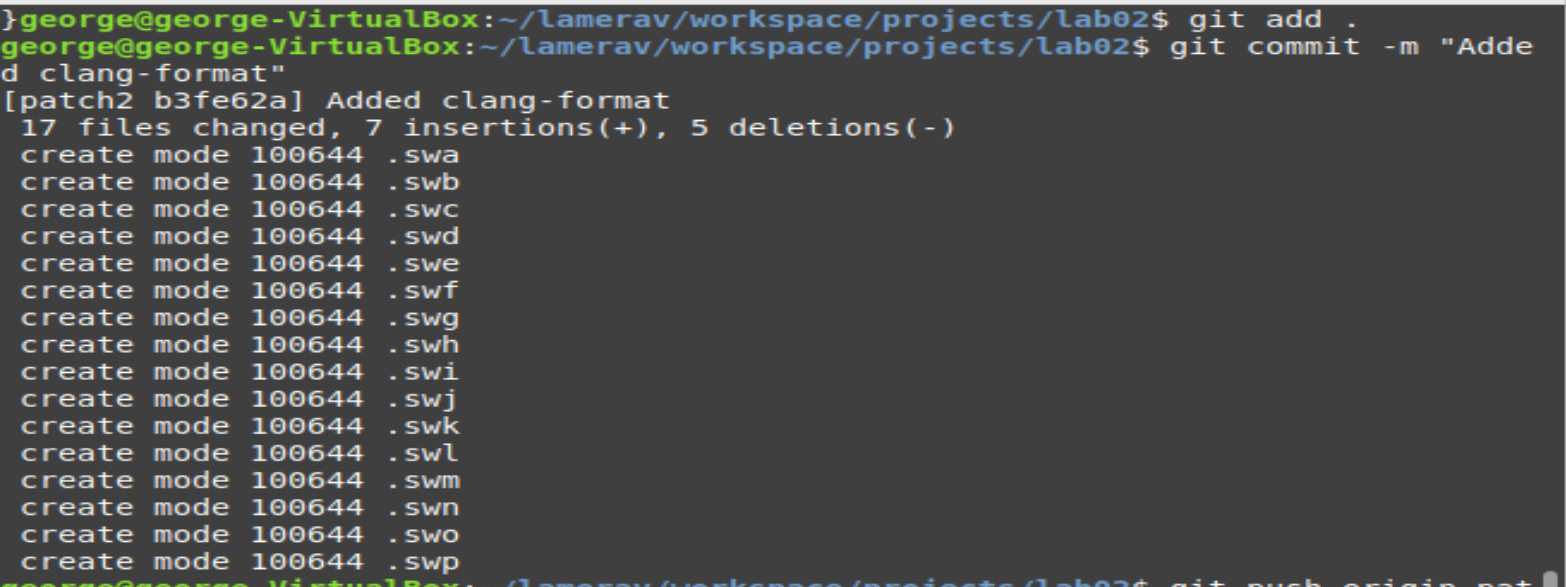
1. Создание ветки patch2

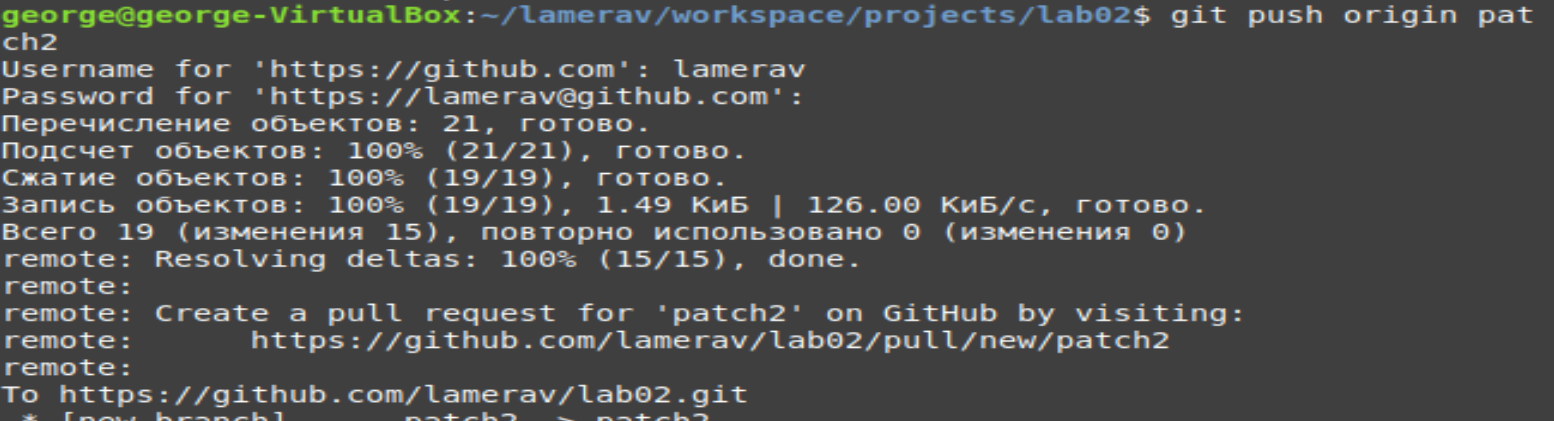


2. Изменение стиля кода с помощью clang-format



3. Commit, push, pull-request





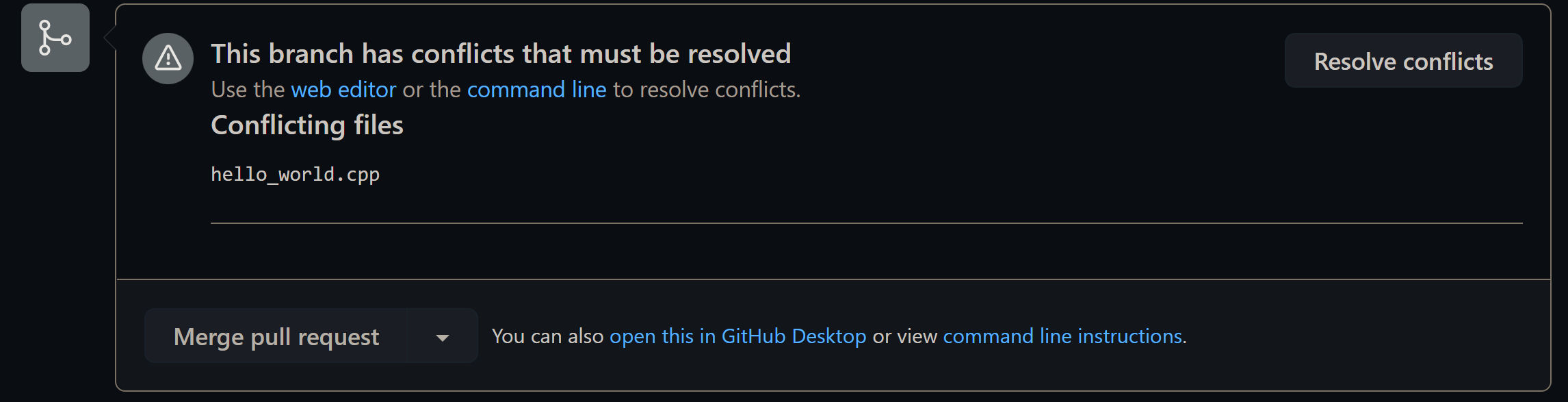
git



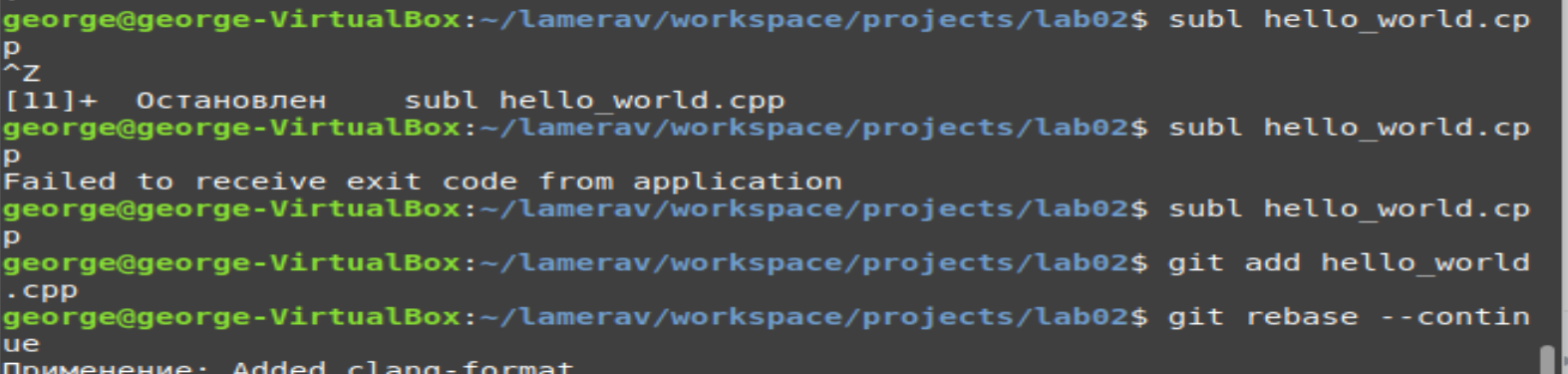
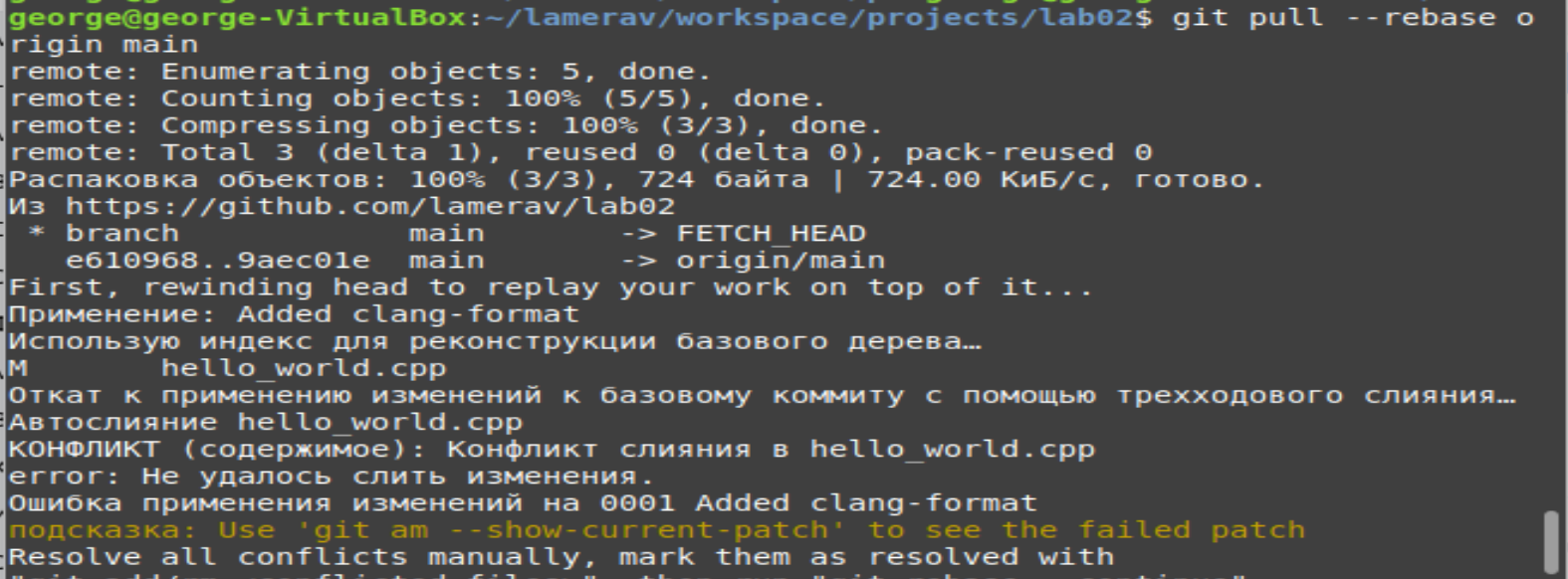
4. Изменяем (добавляем) комментарий



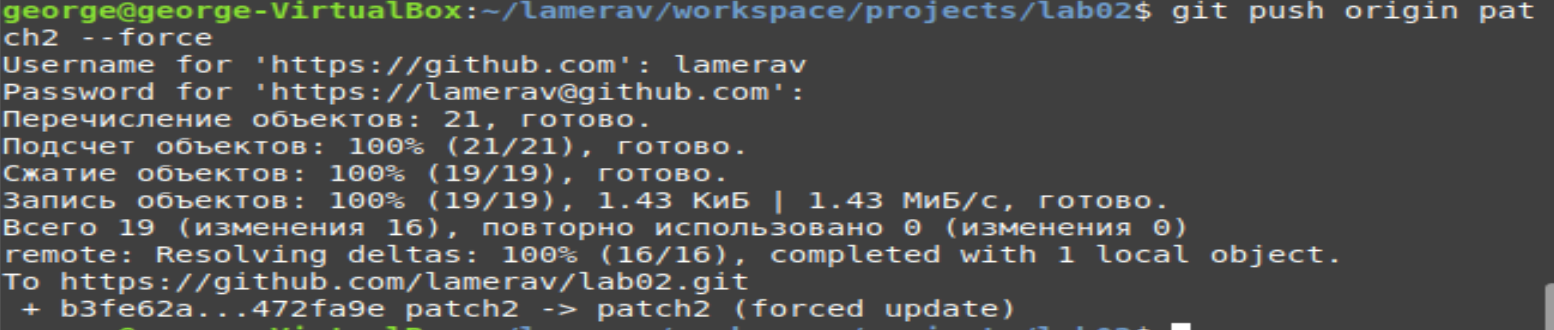
5. Конфликт



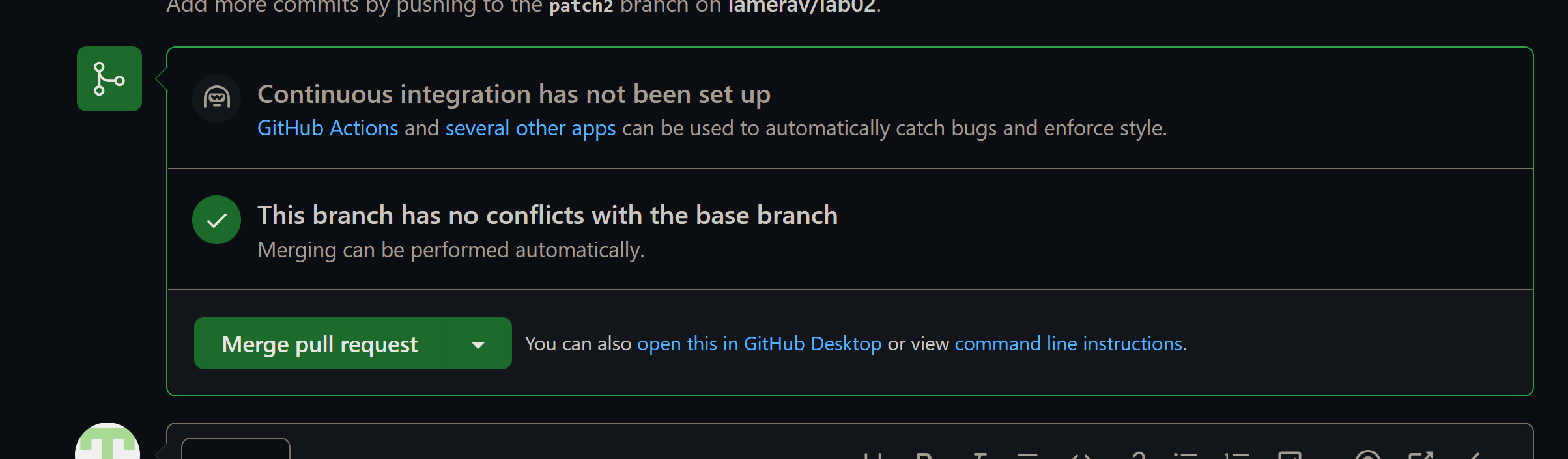
6. Выполняем pull и rebase



7. Force push в patch2



8. Конфликтны пропали



9. merge patch2 → main

