**Part I**

1. Создайте пустой репозиторий на сервисе github.com (или gitlab.com, или bitbucket.com).

laba2

1. Выполните инструкцию по созданию первого коммита на странице репозитория, созданного на предыдещем шаге.

mkdir laba2

cd laba2

echo "# laba2" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/Irocchka/laba2.git

git push -u origin main

1. Создайте файл hello\_world.cpp в локальной копии репозитория (который должен был появиться на шаге 2). Реализуйте программу **Hello world** на языке C++ используя плохой стиль кода. Например, после заголовочных файлов вставьте строку using namespace std;.

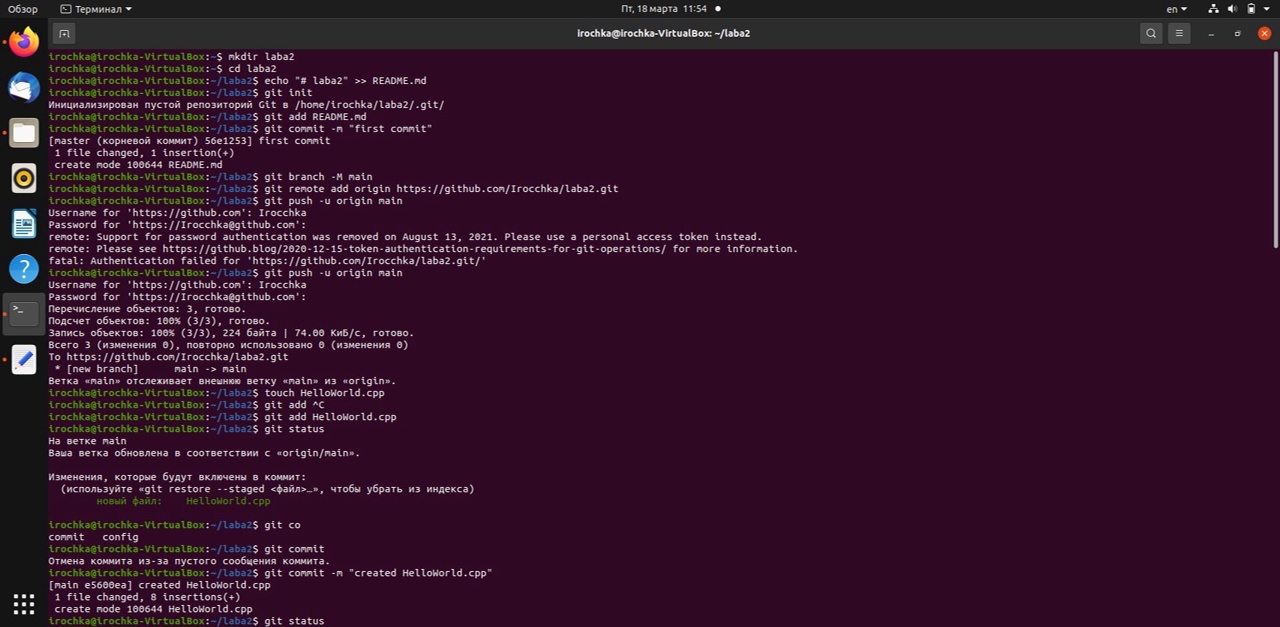
touch HelloWorld.cpp

1. Добавьте этот файл в локальную копию репозитория.

git add HelloWorld.cpp

1. Закоммитьте изменения с *осмысленным* сообщением.

git commit –m “created HelloWorld.cpp”

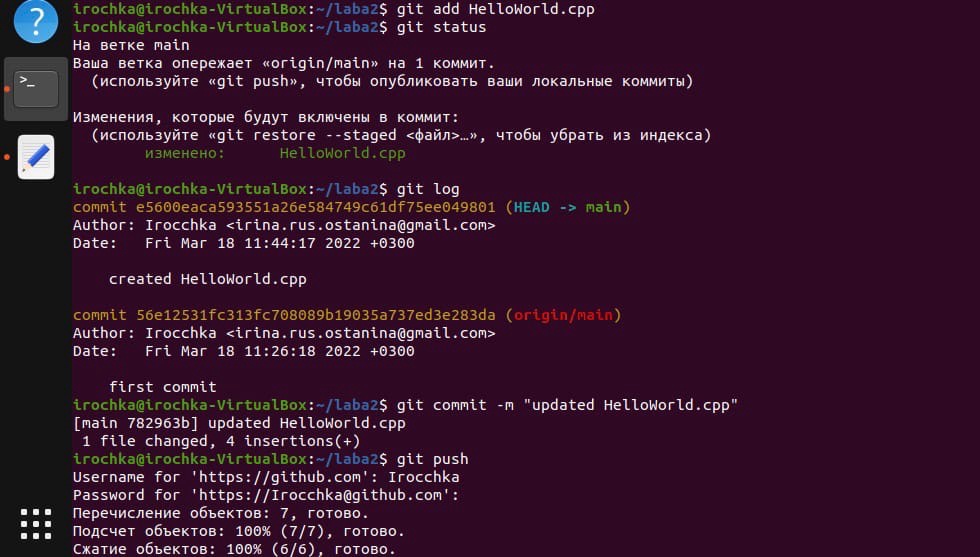


1. Изменитьте исходный код так, чтобы программа через стандартный поток ввода запрашивалось имя пользователя. А в стандартный поток вывода печаталось сообщение Hello world from @name, где @name имя пользователя.
2. Закоммитьте новую версию программы. Почему не надо добавлять файл повторно git add?

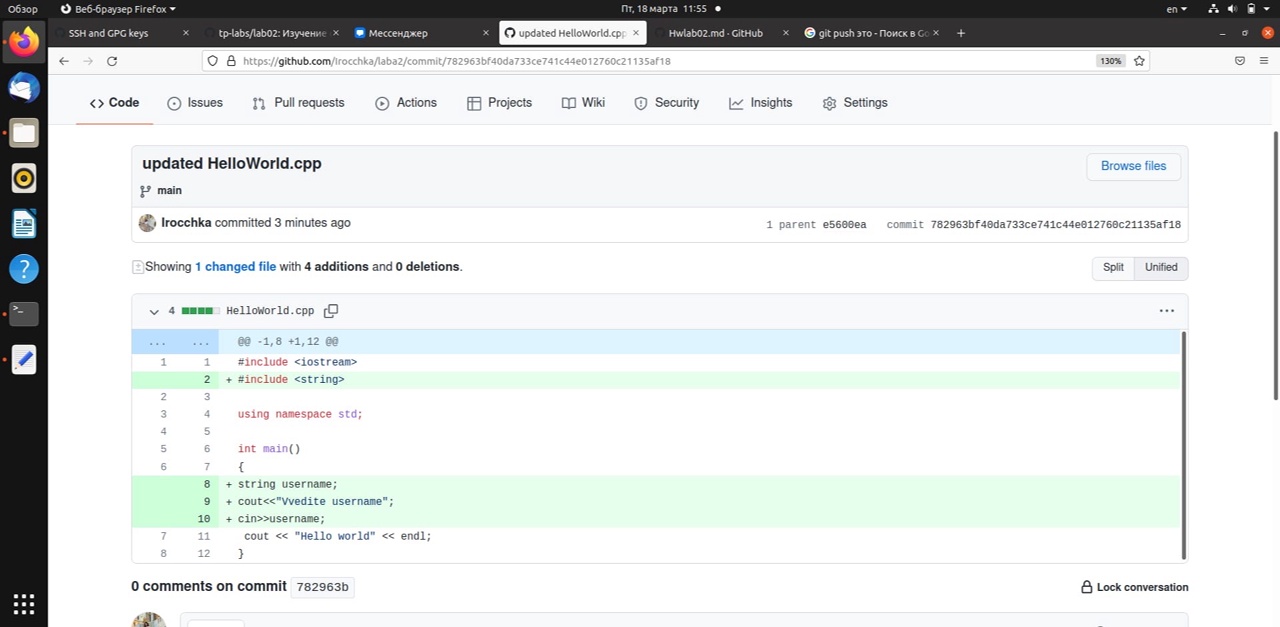
git commit –m “updated HelloWorld.cpp”

1. Запуште изменения в удалёный репозиторий.

git push



1. Проверьте, что история коммитов доступна в удалёный репозитории.



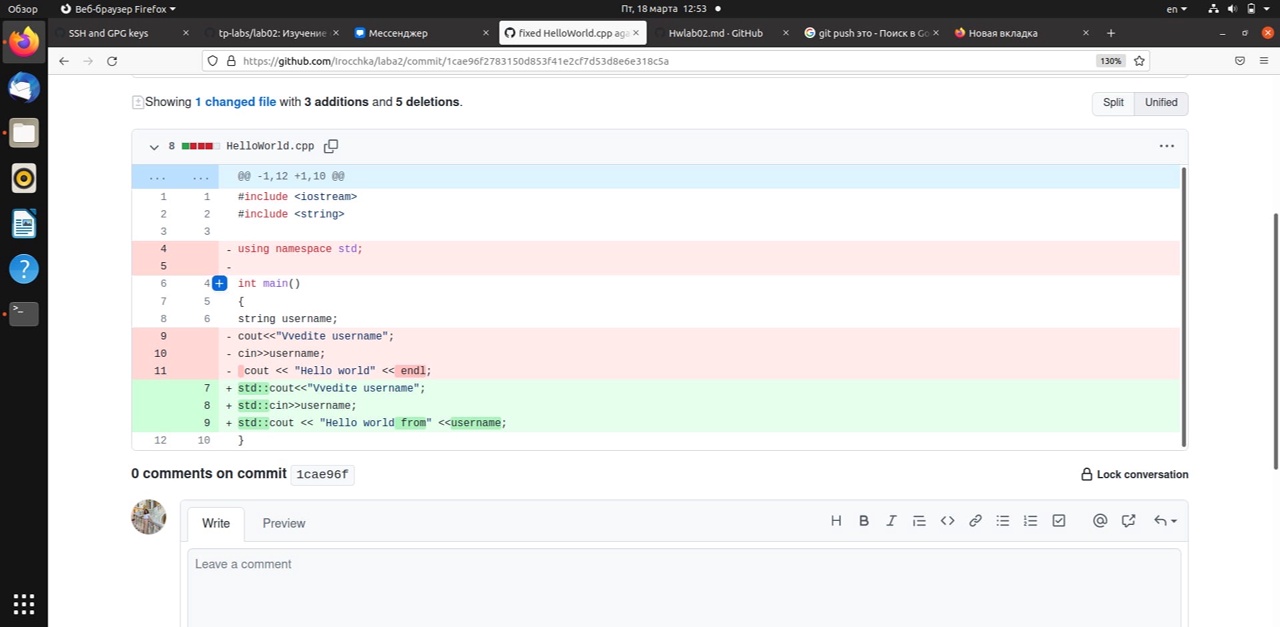
**Part II**

**Note:** *Работать продолжайте с теми же репоззиториями, что и в первой части задания.*

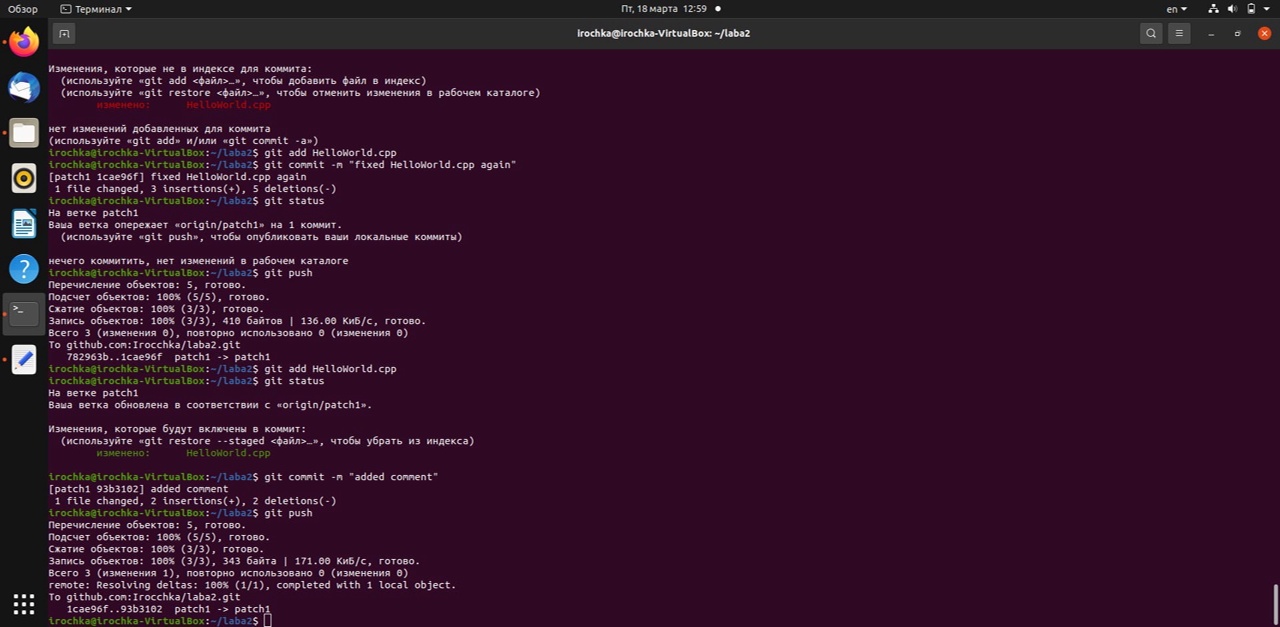
1. В локальной копии репозитория создайте локальную ветку patch1.

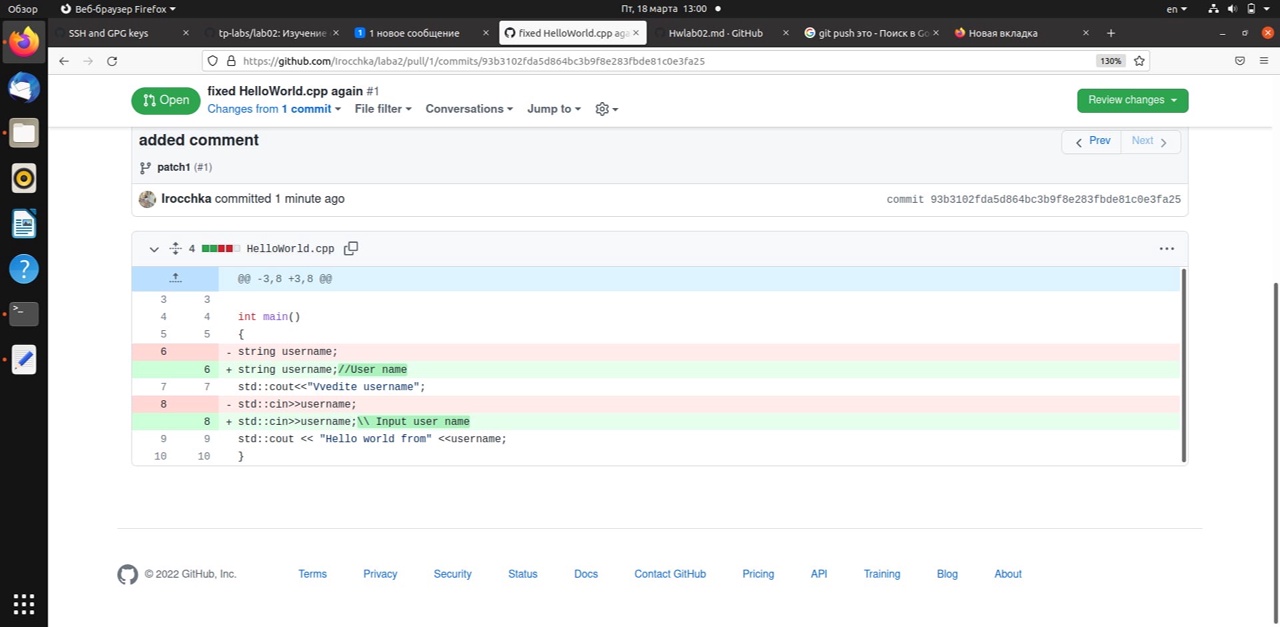
git checkout -b patch1

1. Внесите изменения в ветке patch1 по исправлению кода и избавления от using namespace std;.
2. **commit**, **push** локальную ветку в удалённый репозиторий.



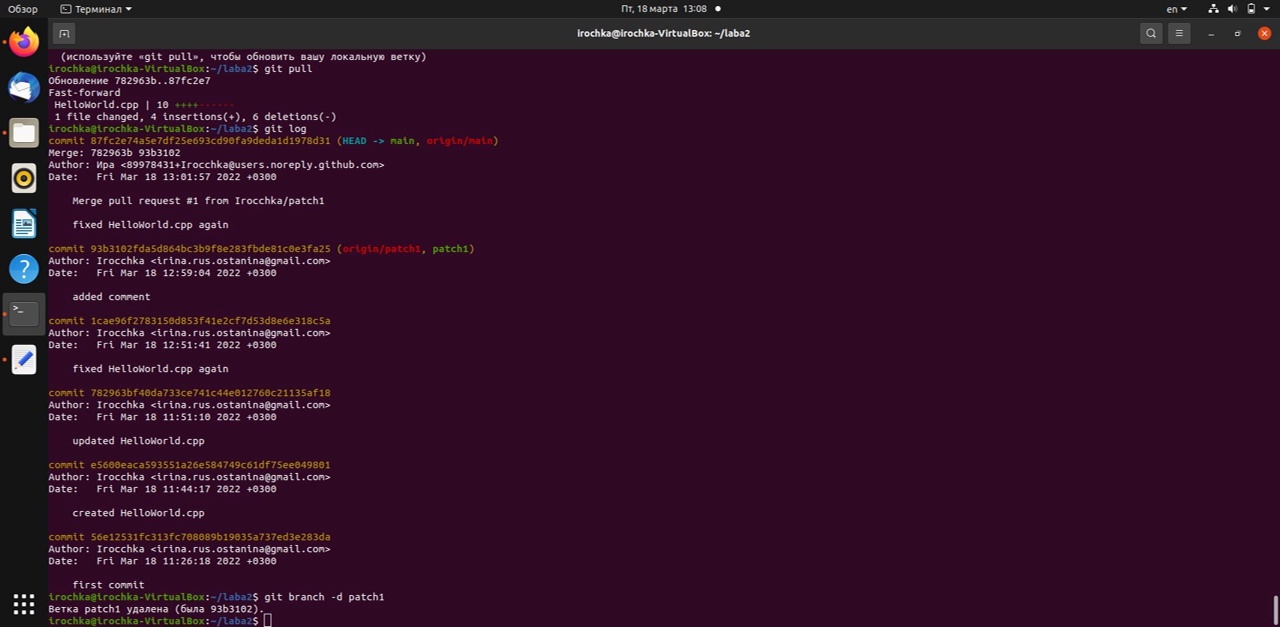
1. Проверьте, что ветка patch1 доступна в удалёный репозитории.
2. Создайте pull-request patch1 -> master.
3. В локальной копии в ветке patch1 добавьте в исходный код комментарии.
4. **commit**, **push**.





1. Проверьте, что новые изменения есть в созданном на **шаге 5** pull-request
2. В удалённый репозитории выполните слияние PR patch1 -> master и удалите ветку patch1 в удаленном репозитории.
3. Локально выполните **pull**.
4. С помощью команды **git log** просмотрите историю в локальной версии ветки master.
5. Удалите локальную ветку patch1.

git branch –d patch1



**Part III**

**Note:** *Работать продолжайте с теми же репоззиториями, что и в первой части задания.*

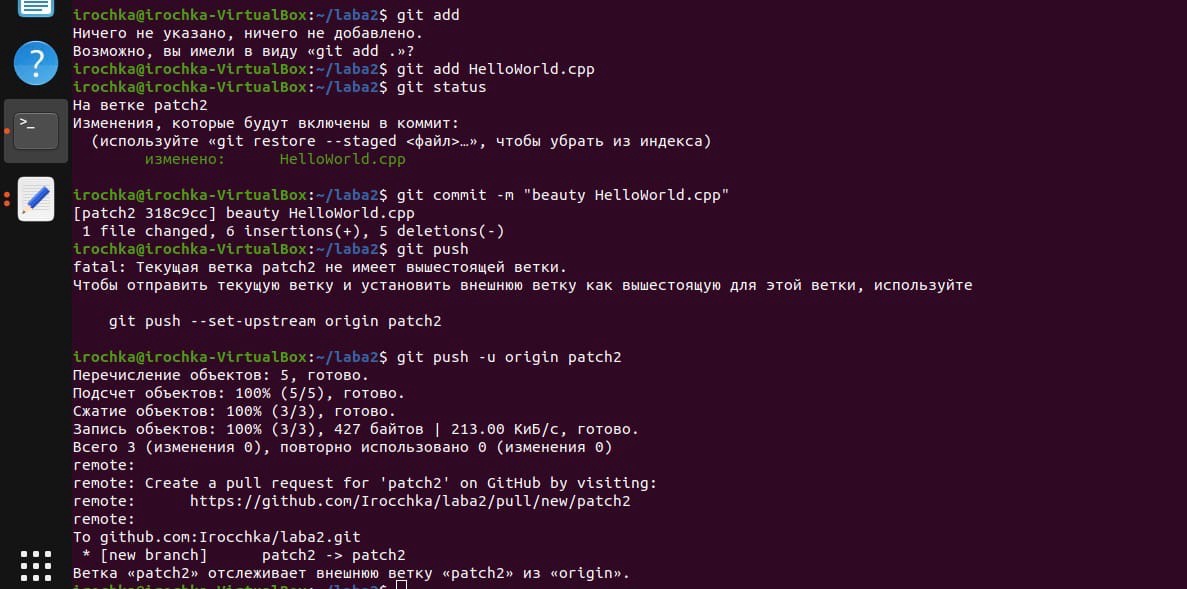
1. Создайте новую локальную ветку patch2.

git checkout -b patch2

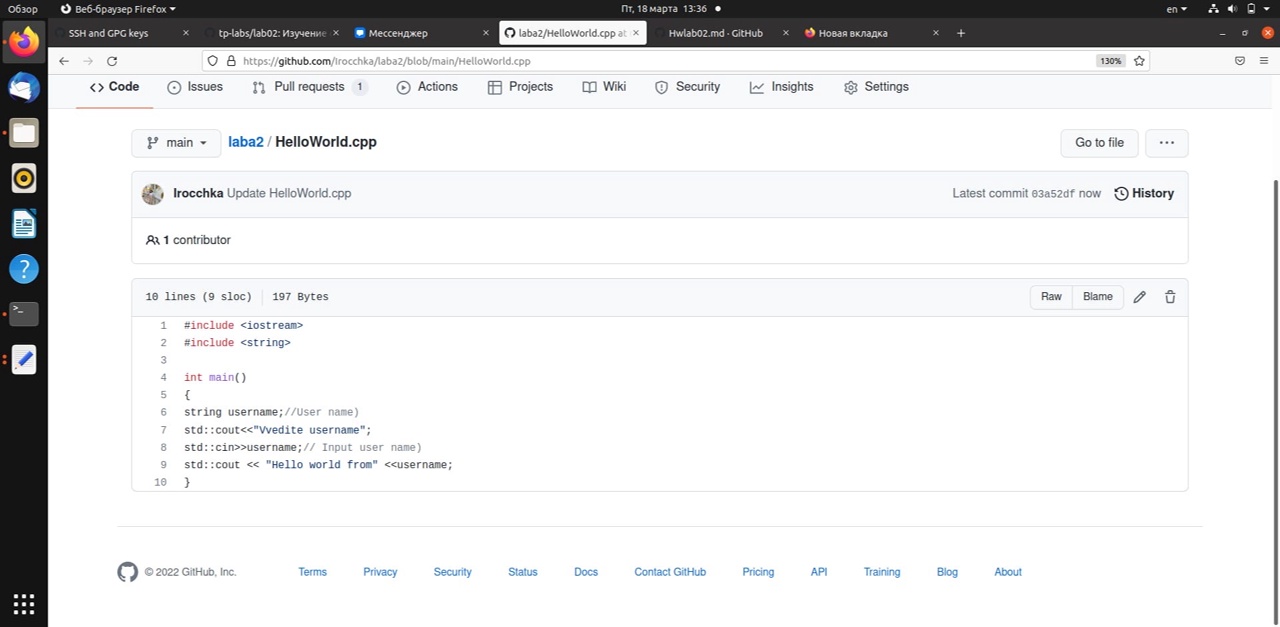
1. Измените *code style* с помощью утилиты **[clang-format](http://clang.llvm.org/docs/ClangFormat.html)**. Например, используя опцию -style=Mozilla.

clang-format -style=Mozilla –i HelloWordl.cpp

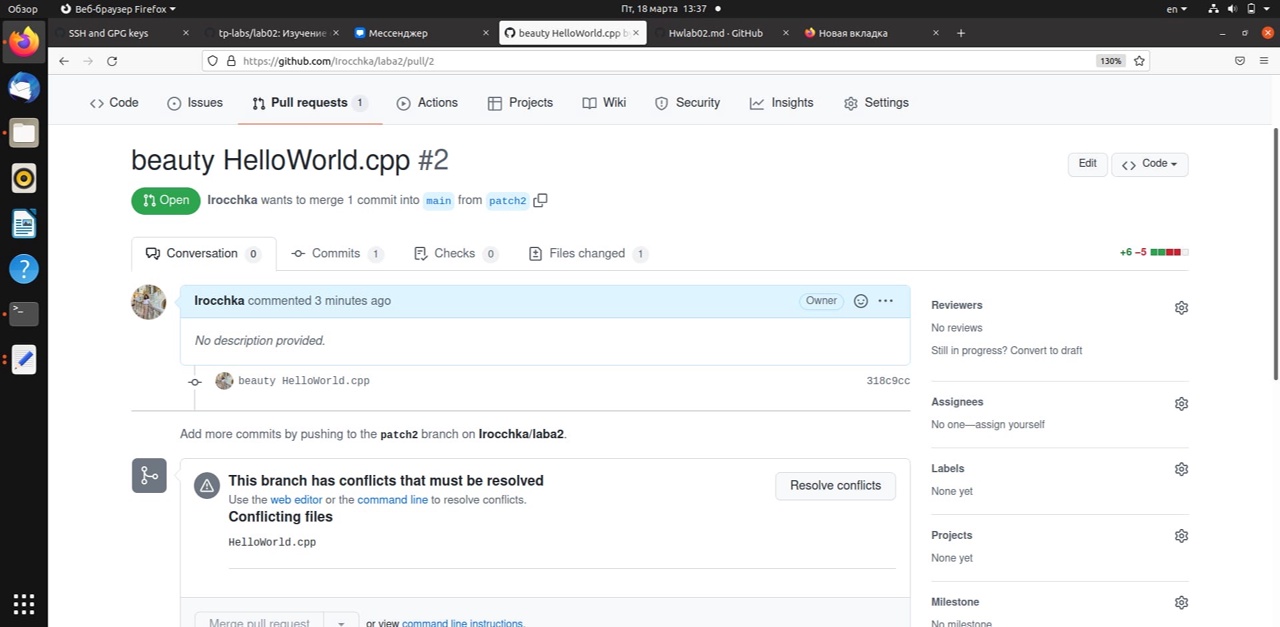
1. **commit**, **push**, создайте pull-request patch2 -> master.



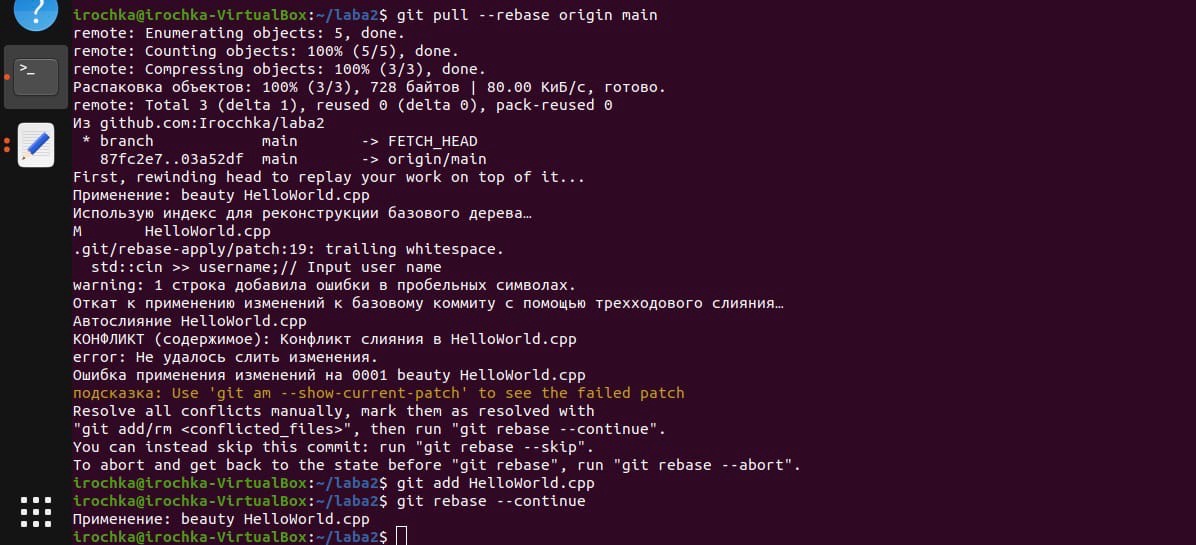
1. В ветке **master** в удаленном репозитории измените комментарии, например, расставьте знаки препинания, переведите комментарии на другой язык.



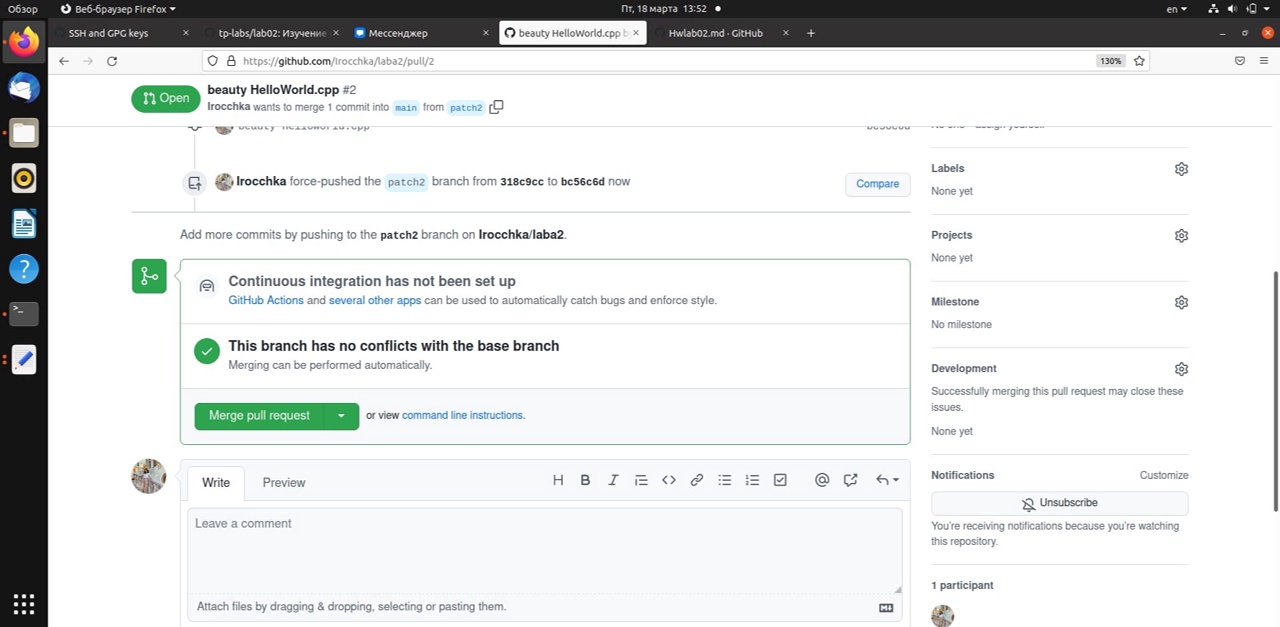
1. Убедитесь, что в pull-request появились *конфликтны*.



1. Для этого локально выполните **pull** + **rebase** (точную последовательность команд, следует узнать самостоятельно). **Исправьте конфликты**.



1. Сделайте *force push* в ветку patch2
2. Убедитель, что в pull-request пропали конфликтны.



1. Вмержите pull-request patch2 -> master.

