СИСТЕМА КОНТРОЛЯ ВЕРСИЙ GITОтчет по лабораторной работе №3

По дисциплине «Технологии и методы программирования»

Выполнил

Студент гр. 728-1

\_\_\_\_\_\_\_\_\_\_ Медведев З.М.

\_\_\_\_\_\_\_\_\_\_ 2020

Принял

Преподаватель кафедры КИБЭВС

\_\_\_\_\_\_\_\_\_\_Перминов П.В.

\_\_\_\_\_\_\_\_\_\_ 2020

Оглавление

[1 Введение 3](#_Toc38641181)

[2 Ход работы 4](#_Toc38641182)

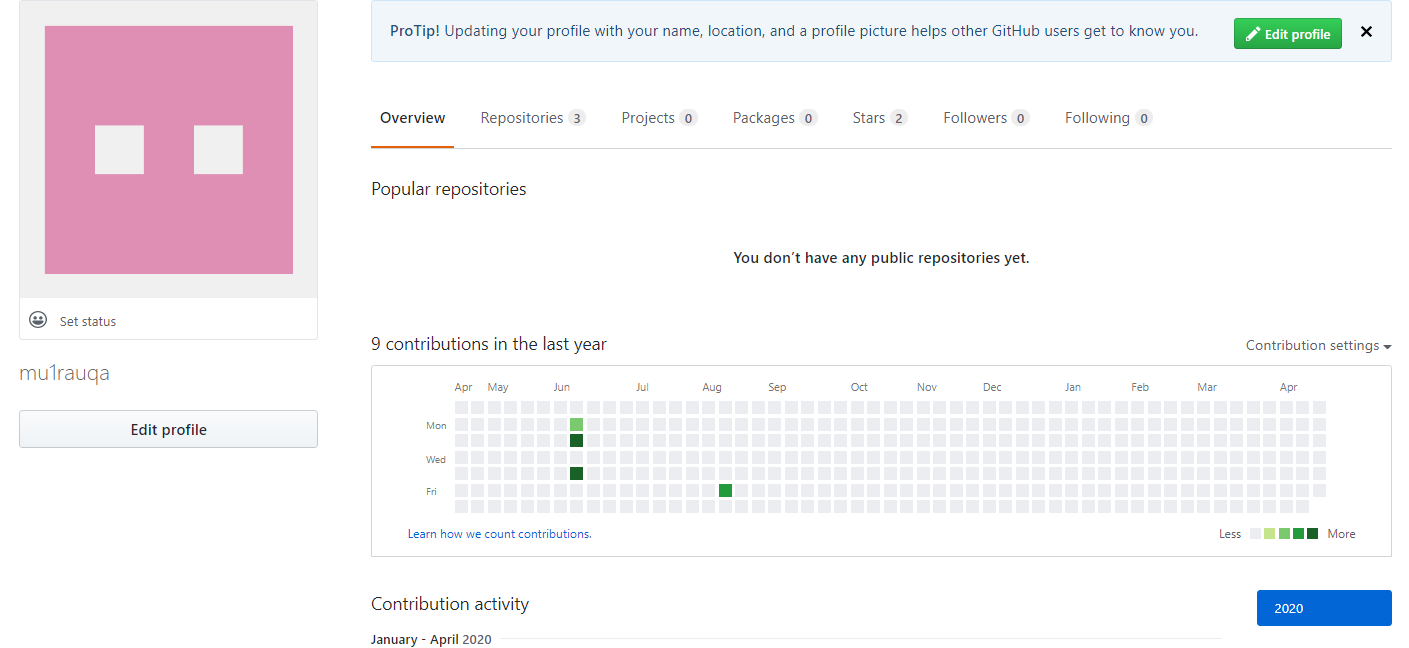
[3 Заключение 12](#_Toc38641183)

# 1 Введение

Целью данной лабораторной работы является изучение функционала системы контроля версий – git и основных команд, а также платформы для хостинга IT-проектов Github.com

# 2 Ход работы

Для работы с git был создан профиль на github (рисунок 2.1)

Рисунок 2.1 - Репозиторий на github

Команда ssh-keygen с ключами –t rsa и –b 4096, генерирует два ключа, один публичный, а другой приватный. Генерация происходит благодаря алгоритму RSA, длиной 4096 бит. После этого использовалась команда ssh-add для добавления обоих ключей в директорию ~/.ssh/id\_rsa (рисунок 2.2)

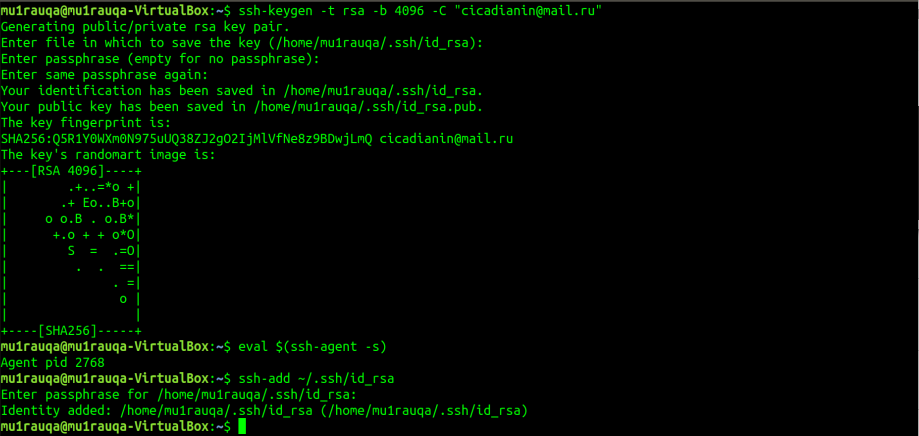


Рисунок 2.2 – Генерация ssh-ключа локально

После проделанных действий в профиль Github был добавлен публичный ключ (рисунок 2.3)

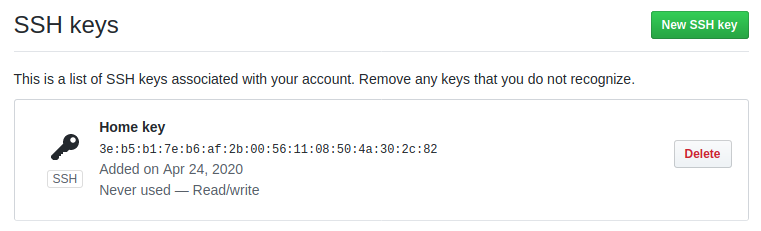


Рисунок 2.3 – Публичные ключи в профиле Github

Для начала необходимо сгенерировать ключи, используя команду gpg --full-generate-key. После ввода будут предложены варианты алгоритмов для создания открытого/закрытого ключа (по умолчанию RSA/RSA). Затем нужно указать длину ключа, далее время жизни ключей (по умолчанию без срока) (рисунок 2.4).

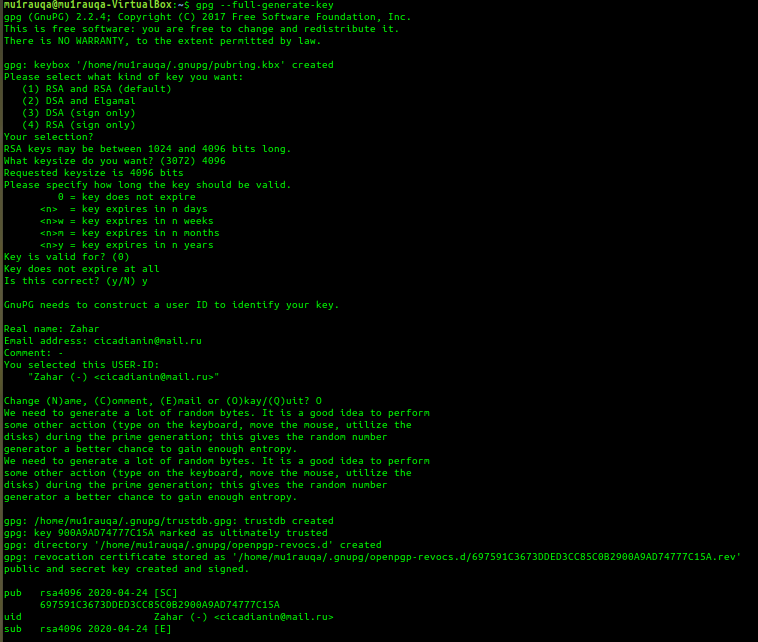


Рисунок 2.4 – Генерация ключей

Следующим шагом, можно будет добавить публичный ключ шифрования в профиль на Github (рисунок 2.5).

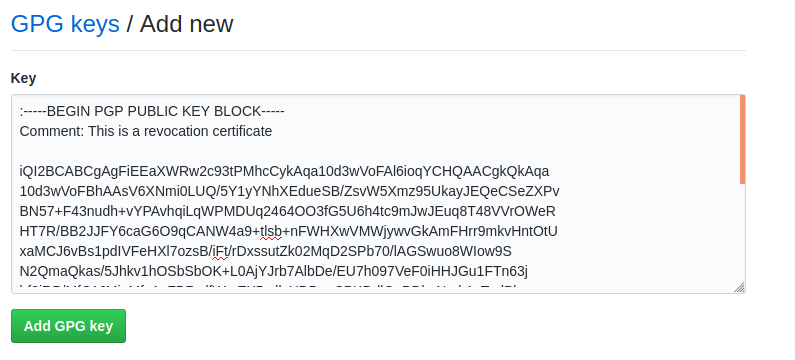


Рисунок 2.5 – Добавление ключа gpg в Github профиль

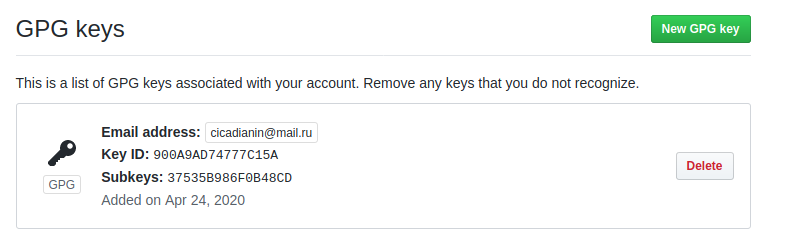


Рисунок 2.6 – Добавленный gpg-ключ в профиле Github

Затем был создан публичный репозиторий с файлом README.md, файлом .gitignore для языка C и файлом лицензии BSD 3-Clause (рисунок 2.7) и (рисунок 2.8)

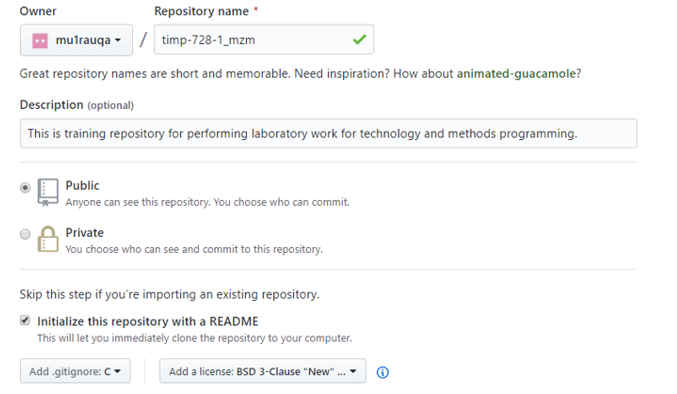


Рисунок 2.7 – Создание удаленного репозитория

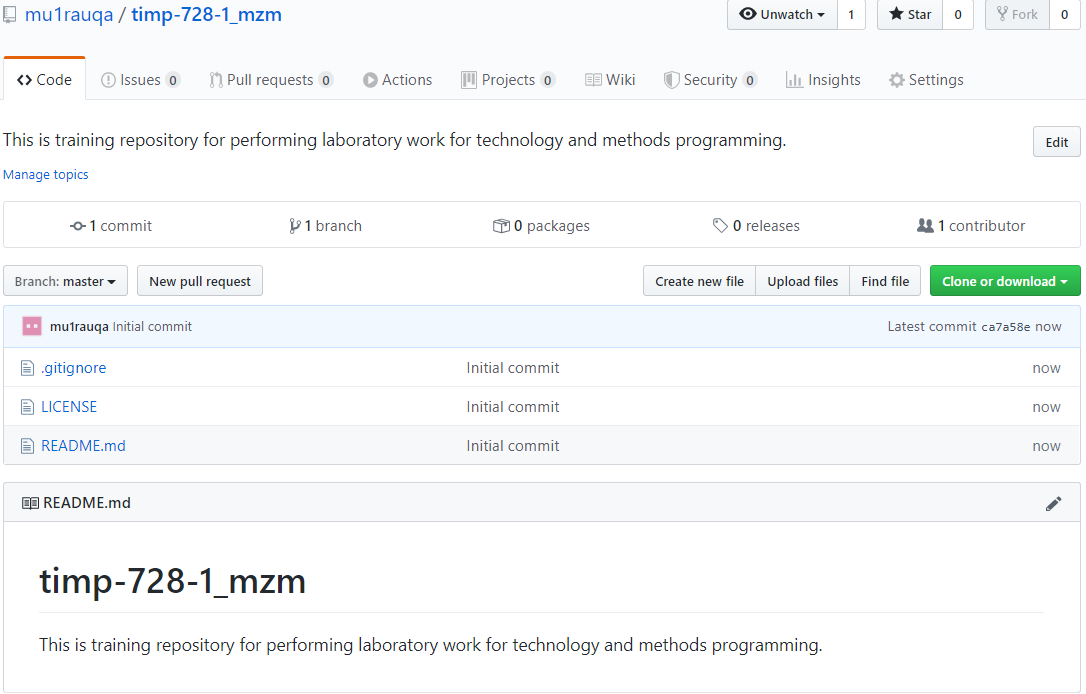


Рисунок 2.8 – Созданный удаленный публичный репозиторий

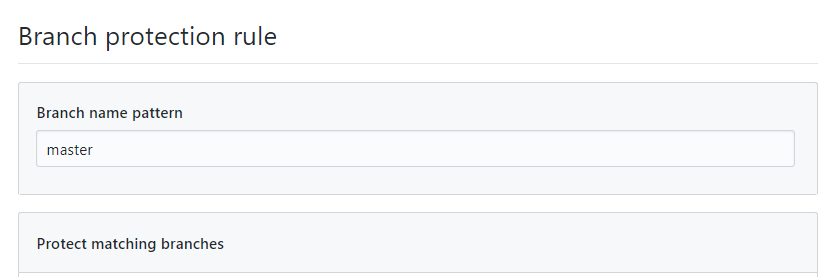


Рисунок 2.9 – Добавление ветки master в protected

Затем была выполнена команда git clone (рисунок 2.10)

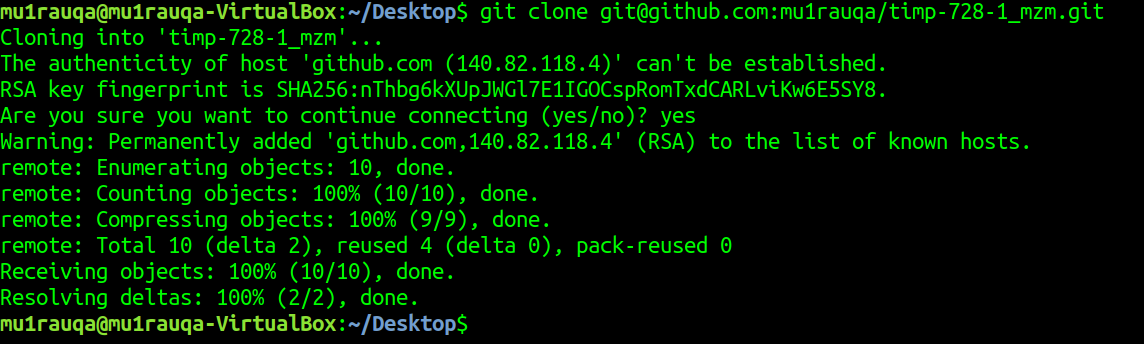


Рисунок 2.10 – Клонирование репозитория

Далее был выполнен переход в директорию timp-728-1\_mzm, выведен список файлов, при помощи команды ls –a, и выведена история коммитов (рисунок 2.11).



Рисунок 2.11 – История коммитов

Затем была создана ветка feature/practice1 (рисунок 2.12)

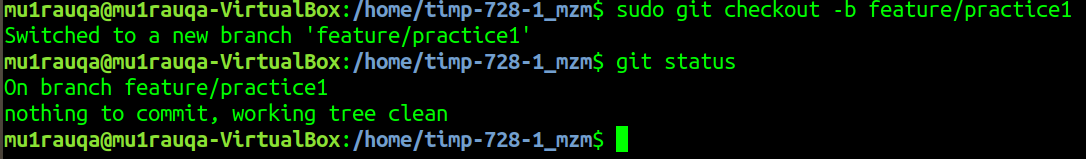


Рисунок 2.12 – Создание ветки и переключение на неё

Используя редактор vim, были внесены изменения в файл README.md (рисунок 2.13).

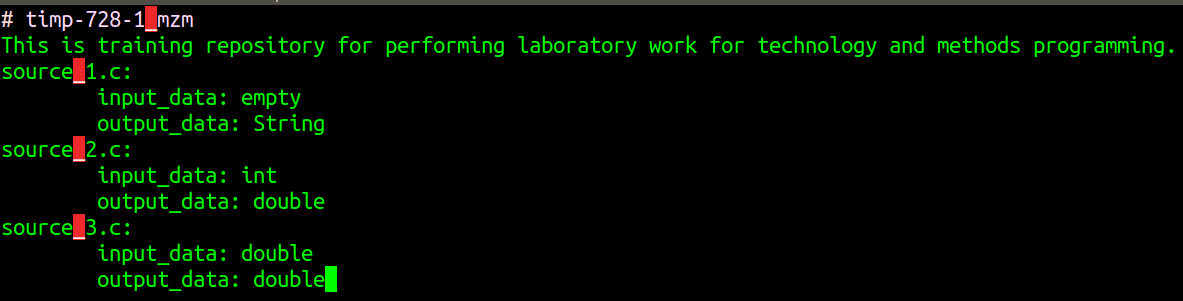
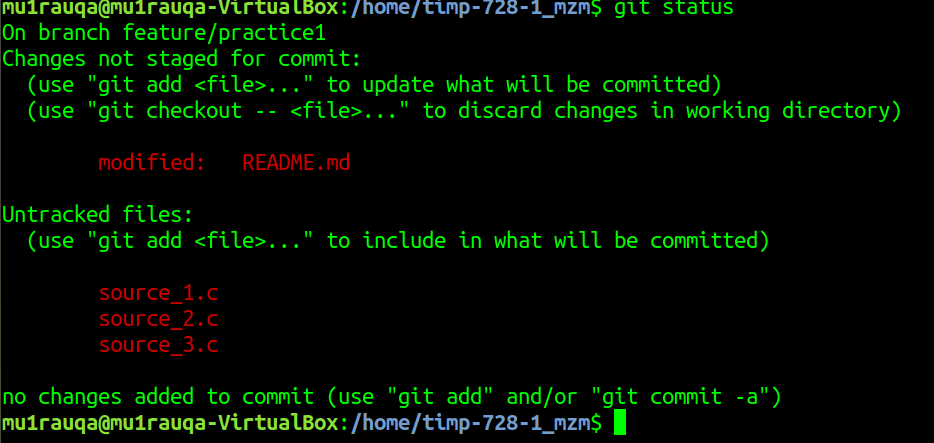


Рисунок 2.13 – Добавление изменений в файл README.md

Рисунок 2.14 – Вывод команды git status

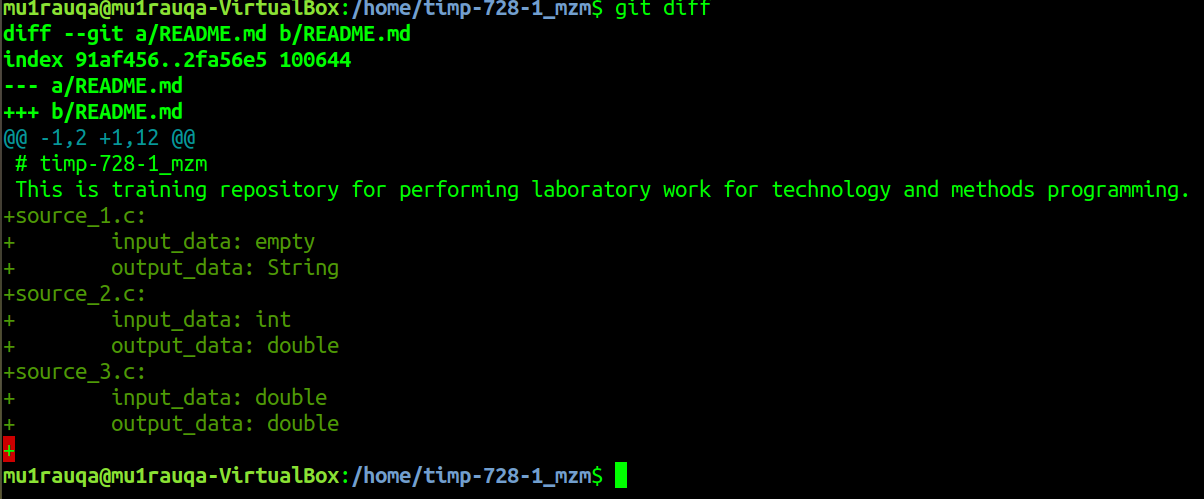


Рисунок 2.15 – Вывод команды git diff

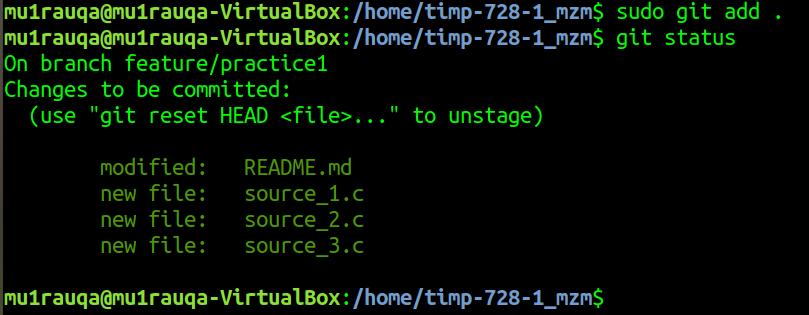


Рисунок 2.16 – Добавление файлов в систему git

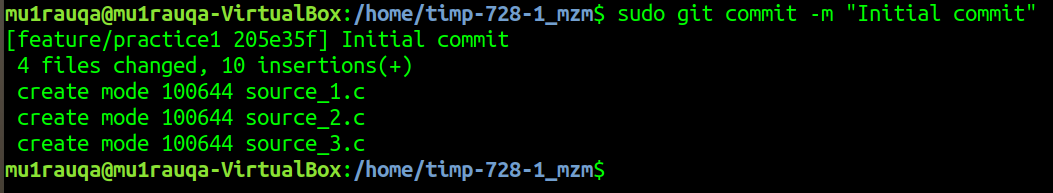


Рисунок 2.17 – Создание инициализирующего коммита

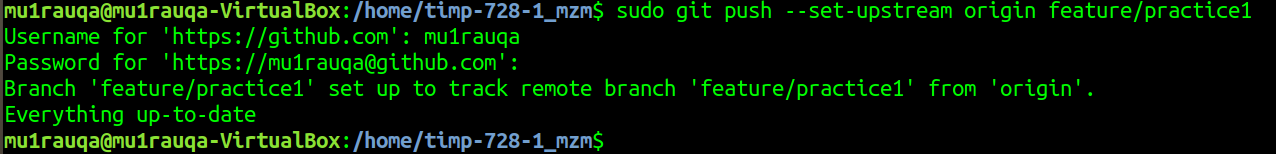


Рисунок 2.18 – Загрузка изменений на удаленный репозиторий

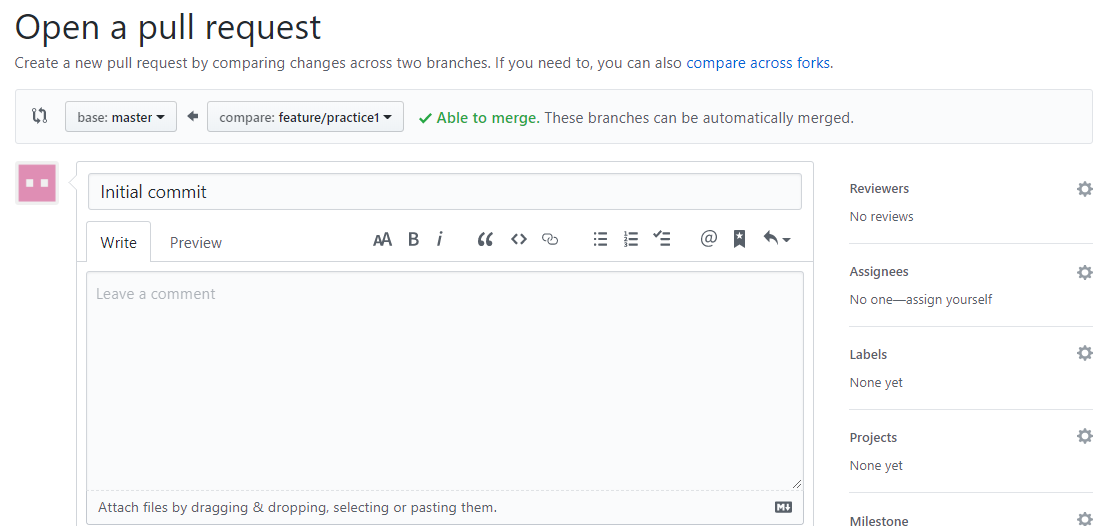


Рисунок 2.19 – Создание pull request из feature/practice1 в master

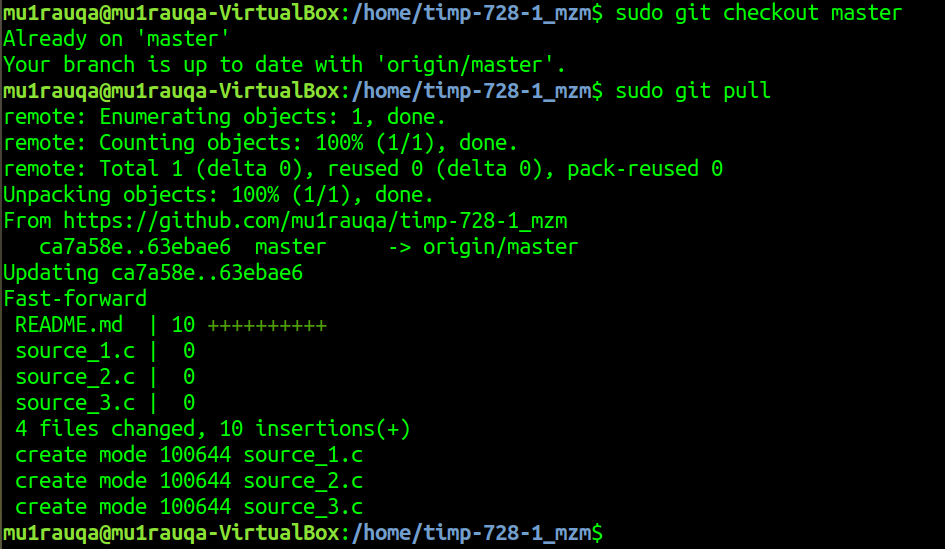


Рисунок 2.20 – Подгрузка изменений из удаленного репозитория в локальный

# 3 Заключение

В ходе выполнения данной лабораторной работы была изучена функциональность системы контроля версий – git и основных команд, а также платформа для хостинга IT-проектов Github.com

Был написан отчет согласно ОС ТУСУР 01-2013