Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Рязанский государственный радиотехнический университет

имени В.Ф. Уткина»

Кафедра «Электронные вычислительные машины»

Отчет о лабораторной работе №4

**"** **Git. Совместная работа."**

по дисциплине

«Утилиты разработки программного обеспечения»

Выполнили:

Ст. гр. 245

Бекренев В.О.

Луковкин И.В.

Бригада №2 Проверил:

Доцент каф. ЭВМ

Ефимов А.И.

Рязань, 2023 г.

**Цель работы:** получение навыков работы с системой контроля версий Git, используя GitLab (ветви, разрешение конфликтов).

**Теоретическая часть**

Ветка – это направления разработки, которое существует независимо от другого направления, однако имеющие с ним общую историю, если заглянуть немного в прошлое. Ветка всегда берет начало как копия чего-либо и двигается от этого момента создавая свою собственную историю.

Ветвление Git очень легковесно. Операция создания ветки выполняется почти мгновенно, переключение между ветками - также быстро. В отличии от многих других СКВ, Git поощряет процесс работы, при котором ветвление и слияние выполняется часто, даже по несколько раз в день.

Ветка (branch) в Git — это легко перемещаемый указатель на один из коммитов. Имя основной ветки по умолчанию в Git — master.

Когда вы делаете коммиты, то получаете основную ветку, указывающую на ваш последний коммит. Каждый коммит автоматически двигает этот указатель вперед.

**Практическая часть**

В ходе работы вместо GitLab была использован похожая по принципу работы СКВ – GitHub.

1. Один из вас создаёт репозиторий, добавляет туда файла fixme.cpp (находится в этой же директории ЛР\_4).

На сайте Github.com был создан репозиторий “lab\_4”, в который предварительно был внесен файл “test.rtf” (рисунок 1).

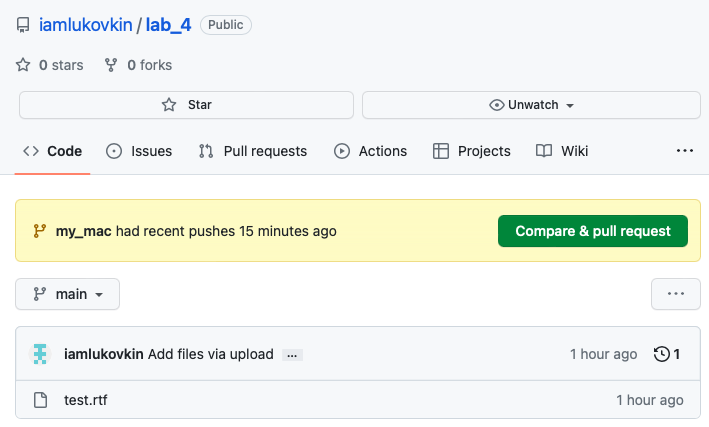


Рисунок 1 – Создание репозитория и добавление в него файла

1. Участник, создавший репозиторий, добавляет второго в коллабоаторы проекта через GitLab.

На сайте, где создавался репозиторий, в поле *Settings* были настроены участники работы с репозиторием (рисунок 2).

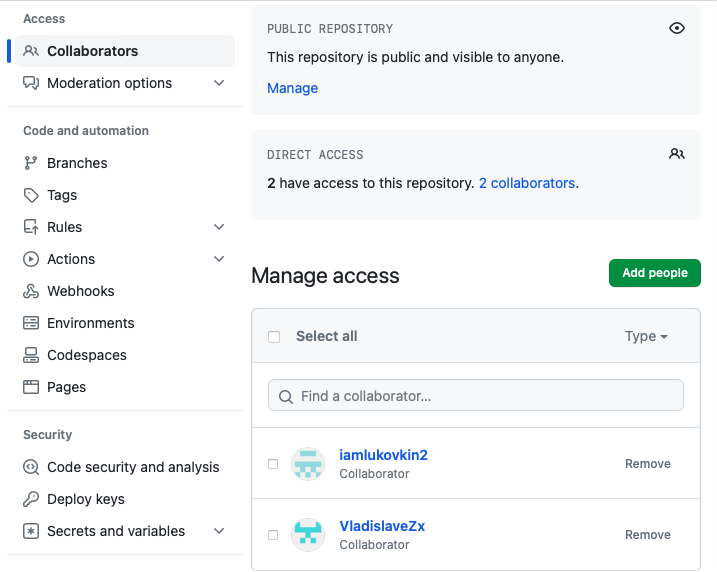


Рисунок 2 – Добавление редакторов

1. Первый и второй участник клонируют проект на компьютеры.

Для работы с репозиторием с имитацией работы с двух компьютеров на виртуальной машине был создан второй редактор. Здесь же был склонирован репозиторий и создана ветка *test\_win*. Вторая копия была произведена на основной компьютер с веткой *test\_mac* (рисунок 3).

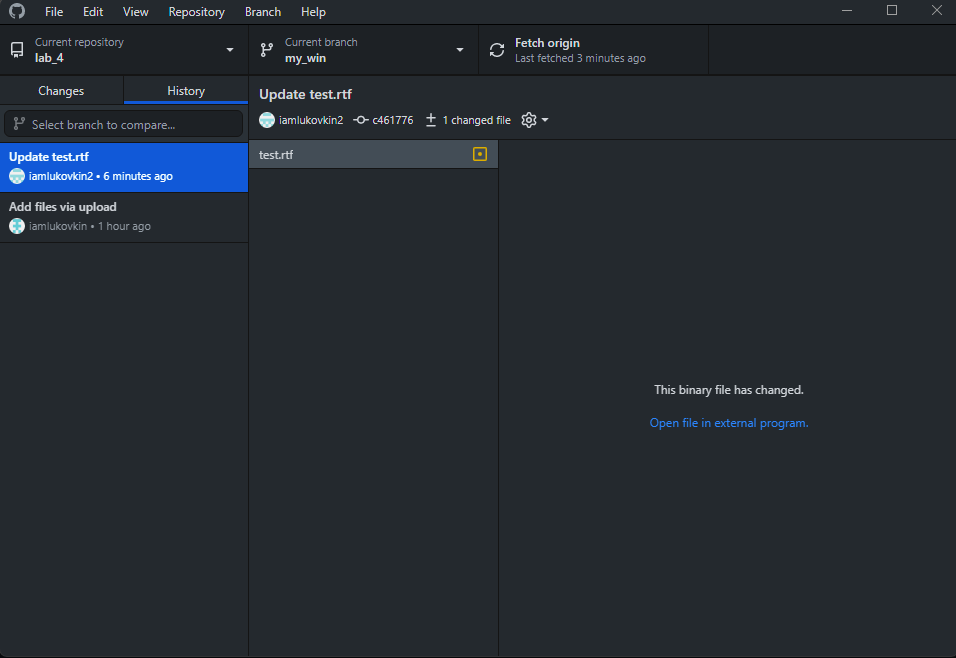
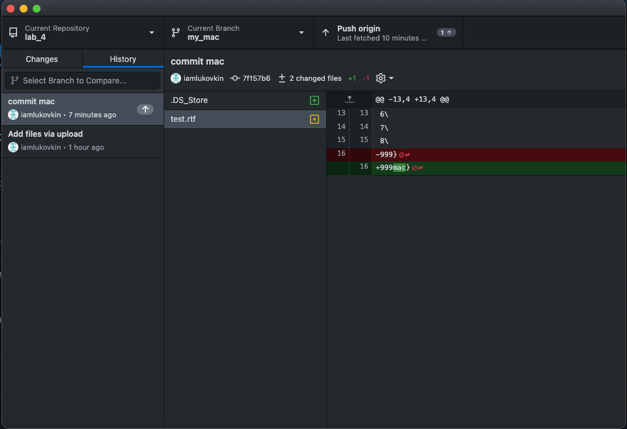


Рисунок 3 – Клонирование репозитория и создание веток

5. Поправьте файл "fixme.cpp". Сделайте так, чтобы ваши изменения покрывали одни и те же участки кода.

Commit-ы веток представлены на рисунке 3.

6. Влейте изменения ветки напарника к себе в ветку. Отразите в сообщении к коммиту, то, что это коммит с вливанием веток.

При слиянии веток возник конфликт, о чем сообщает возникшее окно (рисунок 4). Помимо этого, сразу же представлены возможные варианты решения этого конфликта (рисунок 5).

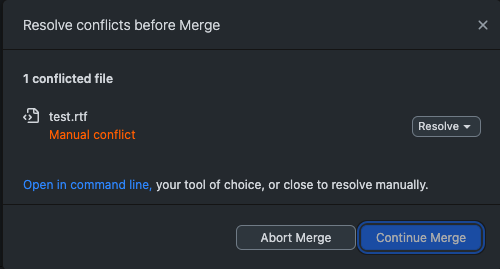


Рисунок 4 – Возникновение конфликта



Рисунок 5 – Варианты решения конфликта

7. Разрешите возникшие конфликты.

При выполнении работы конфликт был решён путем выбора пункта *Use the modified file from my\_win*.

8. Исследуйте, когда конфликт получается, а когда - нет.

Конфликты при слиянии возникают при редактировании пользователями одного и того же объекта файла.

9. Влейте изменения в ветку feature/develop\_\* (предварительно её создав).

Результат слияния представлен на рисунке 6.



Рисунок 6 – Слияние в ветку *develop*

10. Сделайте ещё два любых коммита в своих ветках.

11. Произведите редактирование последнего сообщения коммитов;

Редактирование сообщений коммитов возможно при использовании функции *Amend commit.* Принцип ее работы представлен на рисунке 7.

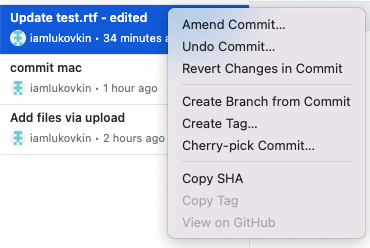
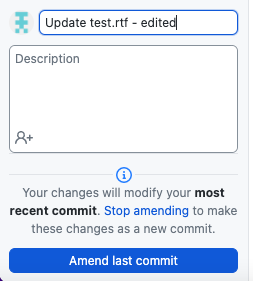


Рисунок 7 – Редактирование сообщений комитов

12. Влейте в свою ветку изменения напарника и затем отмените коммит со слиянием веток.

Проведенные действия можно отменить. Так в работе был отменен последний коммит при использовании функции *Undo Commit* (рисунок 8).

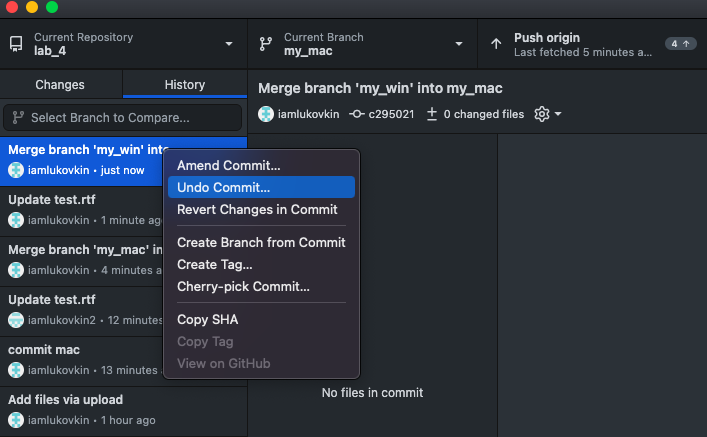


Рисунок 8 – Отмена действия

**Заключение**

В ходе работы нами были изучены методы работы c одной из СКВ из семейства Git. Были достигнуты все поставленные цели.