

# СОДЕРЖАНИЕ

[ЧАСТЬ 1 КАК НАЧАТЬ РАБОТУ С GIT 4](#_bookmark0)

1. [Установка Git 4](#_bookmark1)
2. [Настройка Git 4](#_bookmark2)
3. [Начало работы – создание папки и файлов 4](#_bookmark3)
4. [Создание репозитория 5](#_bookmark4)
5. [Добавление файла в репозиторий и добавление первого коммита 5](#_bookmark5)
6. [Индексация изменений 5](#_bookmark6)
7. [Коммиты нескольких изменений 6](#_bookmark7)
8. [Просмотр истории коммитов 7](#_bookmark8)
9. [Получение старых версий 8](#_bookmark9)
10. [Отмена локальных изменений (до индексации) 9](#_bookmark10)
11. [Отмена локальных изменений (после индексации и до коммита) 10](#_bookmark11)
12. [Отмена коммита 10](#_bookmark12)

[ЧАСТЬ 2 УПРАВЛЕНИЕ РЕПОЗИТОРИЯМИ 11](#_bookmark13)

1. [Создание аккаунта на GitHub 11](#_bookmark14)
2. [Создание SSH-ключа для авторизации 12](#_bookmark15)
3. [Создание нового репозитория для своего проекта 12](#_bookmark16)
4. [Связывание локального и удаленного репозитория 13](#_bookmark17)
5. [Создание веток и переключение между ними 14](#_bookmark18)
6. [Слияние веток 14](#_bookmark19)
7. [Выполните цепочку действий в репозитории, Вариант 11 14](#_bookmark20)

[ЧАСТЬ 3 ВЕТВЛЕНИЕ И ОФОРМЛЕНИЕ КОДА 18](#_bookmark21)

1. [Форк репозитория 18](#_bookmark22)
2. [Клонирование репозитория на локальную машину 18](#_bookmark23)
3. [Создание двух веток в main’e 18](#_bookmark24)
4. [Создание двух веток в main’e 19](#_bookmark25)
5. [Слияние ветки branch1 в ветку branch2 с разрешением конфликтов 20](#_bookmark26)
6. [Выгрузка всех изменений во всех ветках в удаленный репозиторий . 21](#_bookmark27)
7. [Выполнение 3-х коммитов в branch1 22](#_bookmark28)
8. [Повторное клонирование репозитория в другую директорию 22](#_bookmark29)
9. [Повторное клонирование репозитория в другую директорию 23](#_bookmark30)
10. [Выгрузка всех изменений в удаленный репозиторий 23](#_bookmark31)
11. [Выгрузка всех изменений в удаленный репозиторий 24](#_bookmark32)
12. [Получение всех изменений в новом репозитории 24](#_bookmark33)

[ОТВЕТЫ НА 8 ВОПРОСОВ 24](#_bookmark34)

[ВЫВОД 26](#_bookmark35)

# ЧАСТЬ 1 КАК НАЧАТЬ РАБОТУ С GIT

## Установка Git

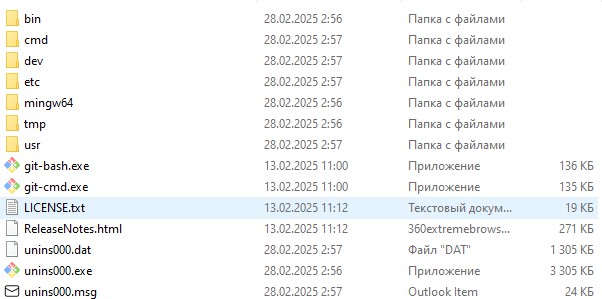


Рисунок 1 – установленный Git

## Настройка Git

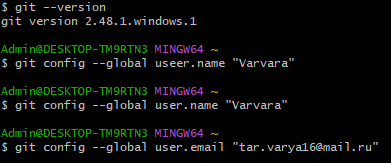


Рисунок 2 – Указание логина и почты для настройки Git

## Начало работы – создание папки и файлов

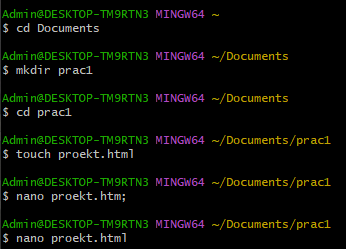


Рисунок 3 – Создание директории и внесение изменений в файлы директории

## Создание репозитория



Рисунок 4 – Инициализация репозитория

## Добавление файла в репозиторий и добавление первого коммита

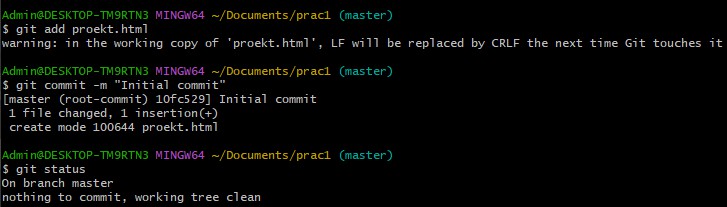


Рисунок 5 – Коммит с комментарием и проверка состояния репозитория

## Индексация изменений

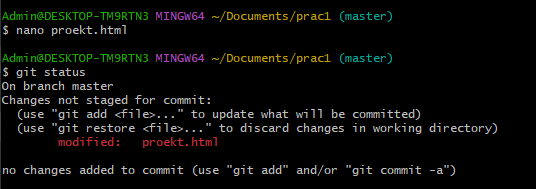


Рисунок 6 – Индексация изменений

## Коммиты нескольких изменений

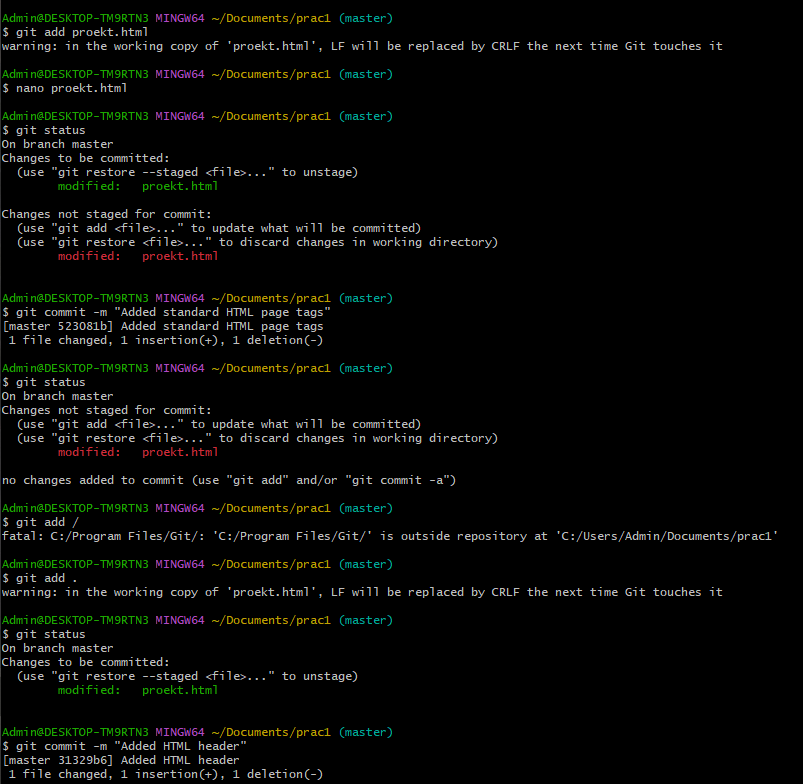


Рисунок 7 – создание двух коммитов с изменениями файла proekt.html

## Просмотр истории коммитов

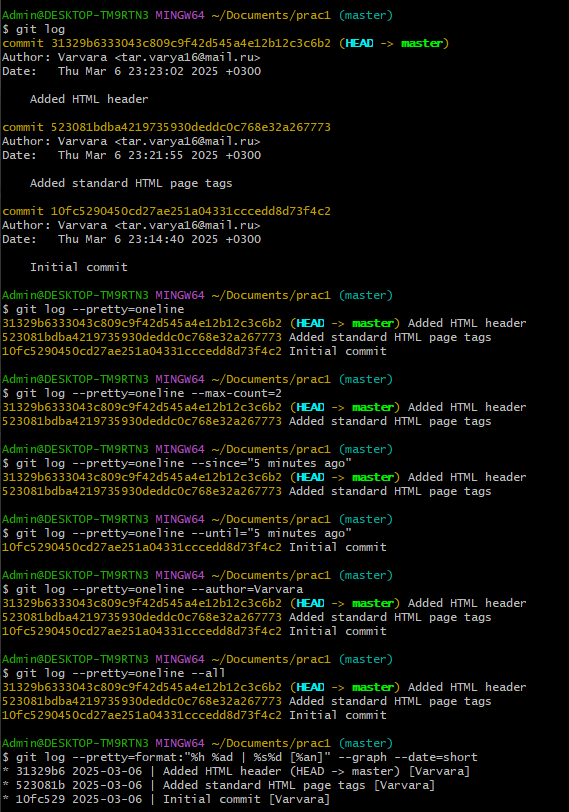


Рисунок 8 – вывод истории коммитов с различными параметрами

## Получение старых версий

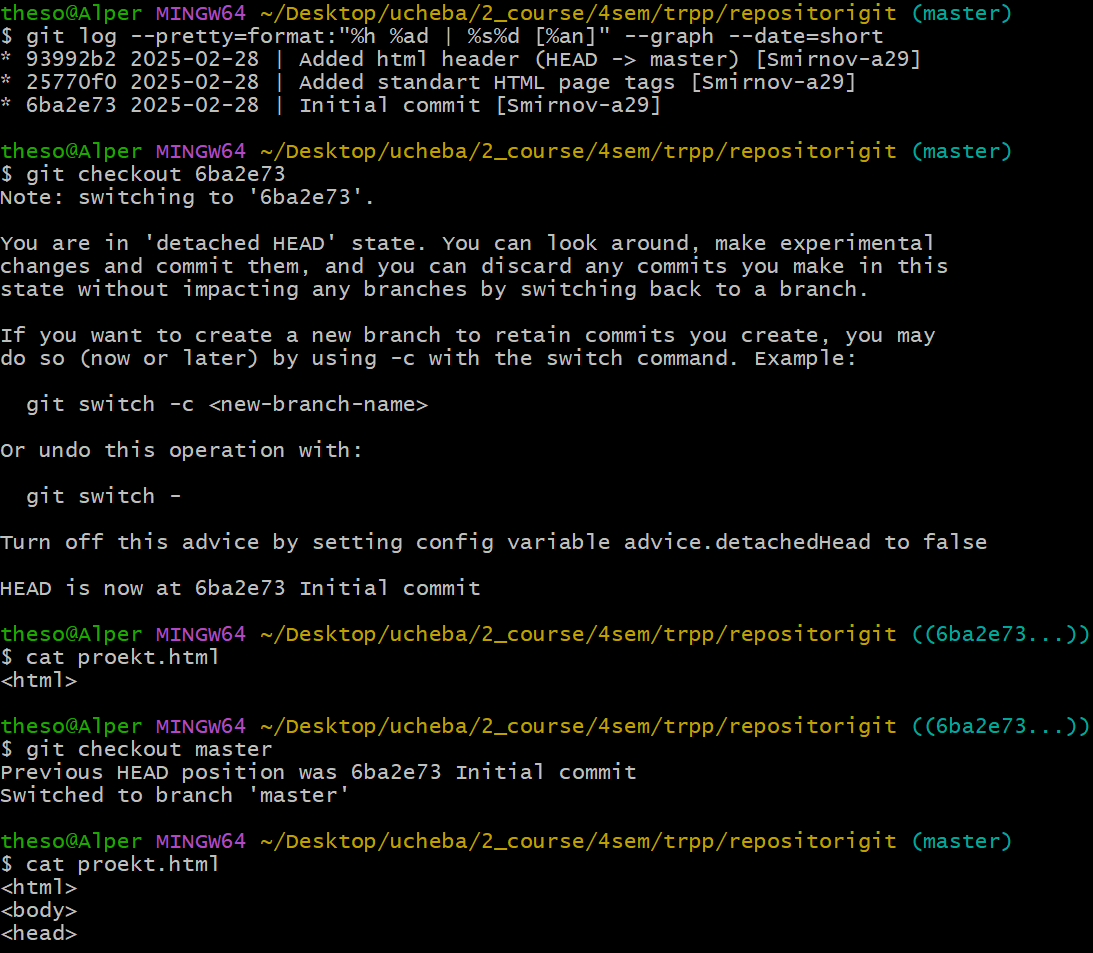


Рисунок 9 – Получение хеша коммита с последующим переключением на него и вывод файла proekt.html. Переключение на ветку master и вывод файла

## Отмена локальных изменений (до индексации)

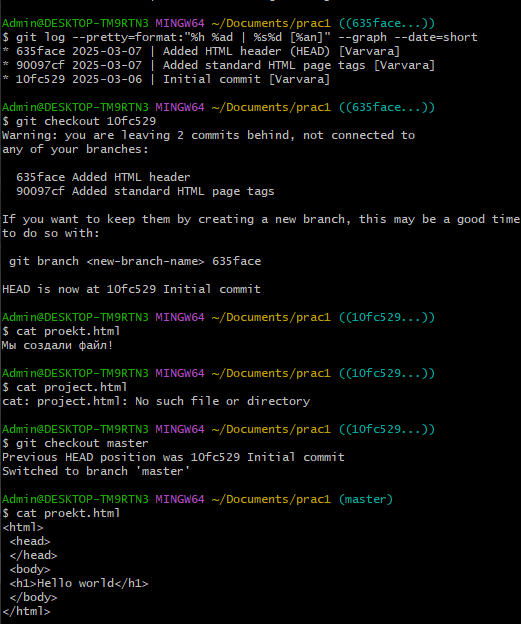


Рисунок 10 – отмена локальных изменений до индексации

## Отмена локальных изменений (после индексации и до коммита)

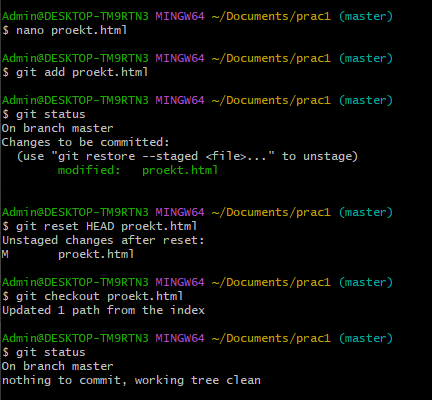


Рисунок 11 – отмена локальных изменений после индексации и до коммита

## Отмена коммита

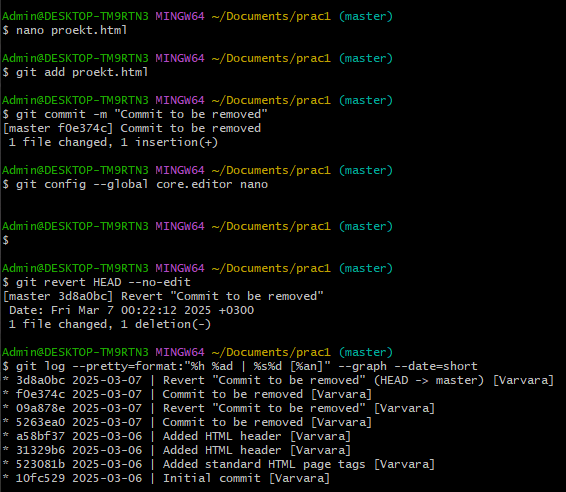


Рисунок 12 – отмена коммита

# ЧАСТЬ 2 УПРАВЛЕНИЕ РЕПОЗИТОРИЯМИ

## Создайте репозиторий на GitHub и на локальной машине, согласно выбранной теме проекта , мой вариант работы 11 :

## V = Oct ((N - 1)/M) + 1 = Oct(22/11) + 1 = 10 + 1 = 11

## Создание аккаунта на GitHub

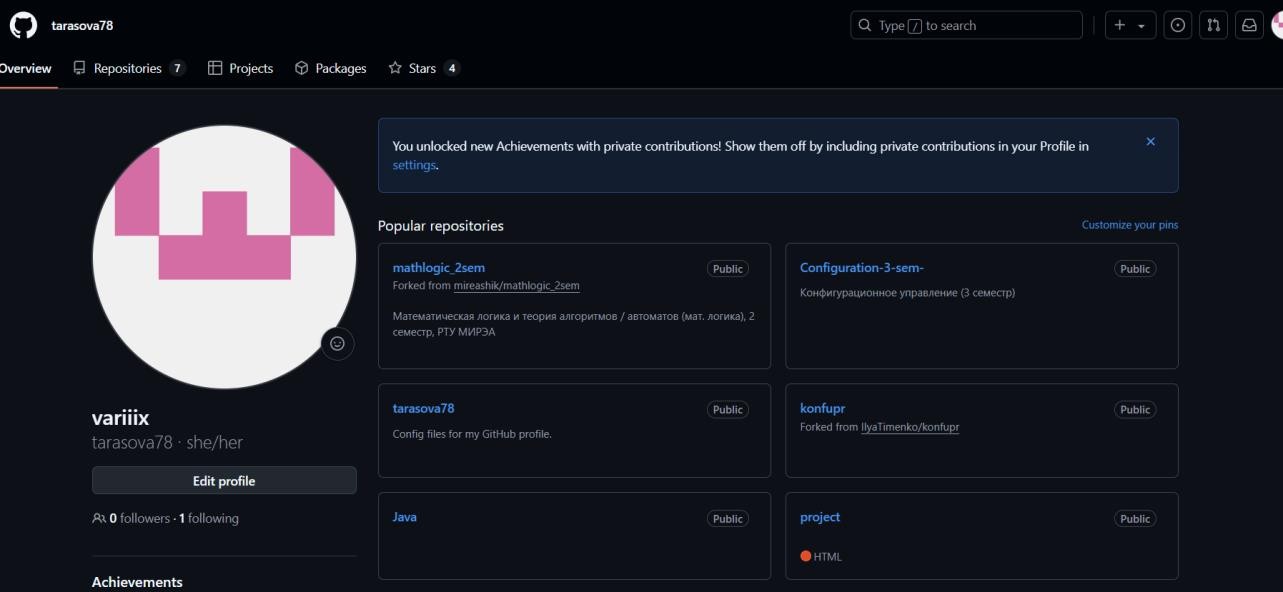


Рисунок 13 – созданный профиль на GitHub

## Создание SSH-ключа для авторизации

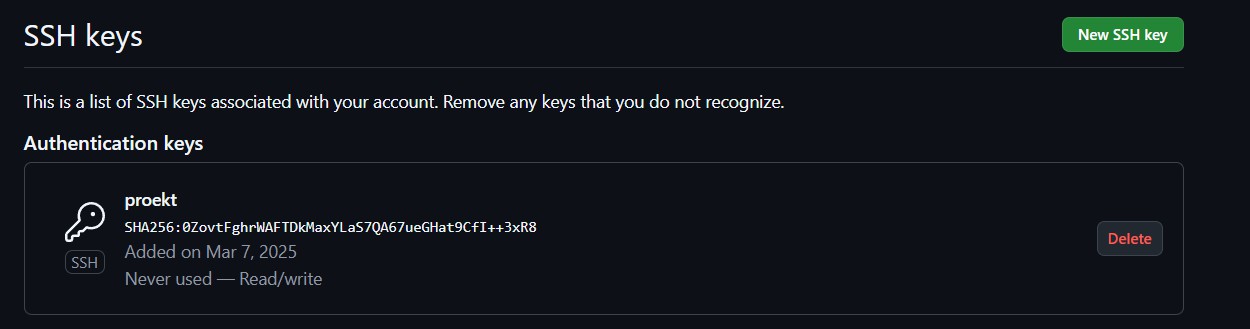


Рисунок 14 – Созданный SSH-ключ

## Создание нового репозитория для своего проекта

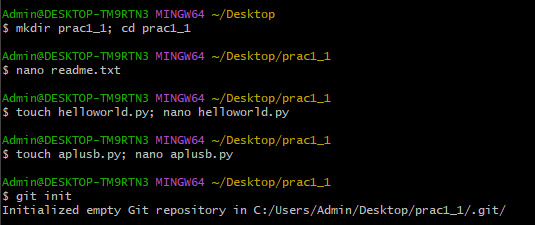


Рисунок 15 – создание репозитория с несколькими файлами

## Связывание локального и удаленного репозитория

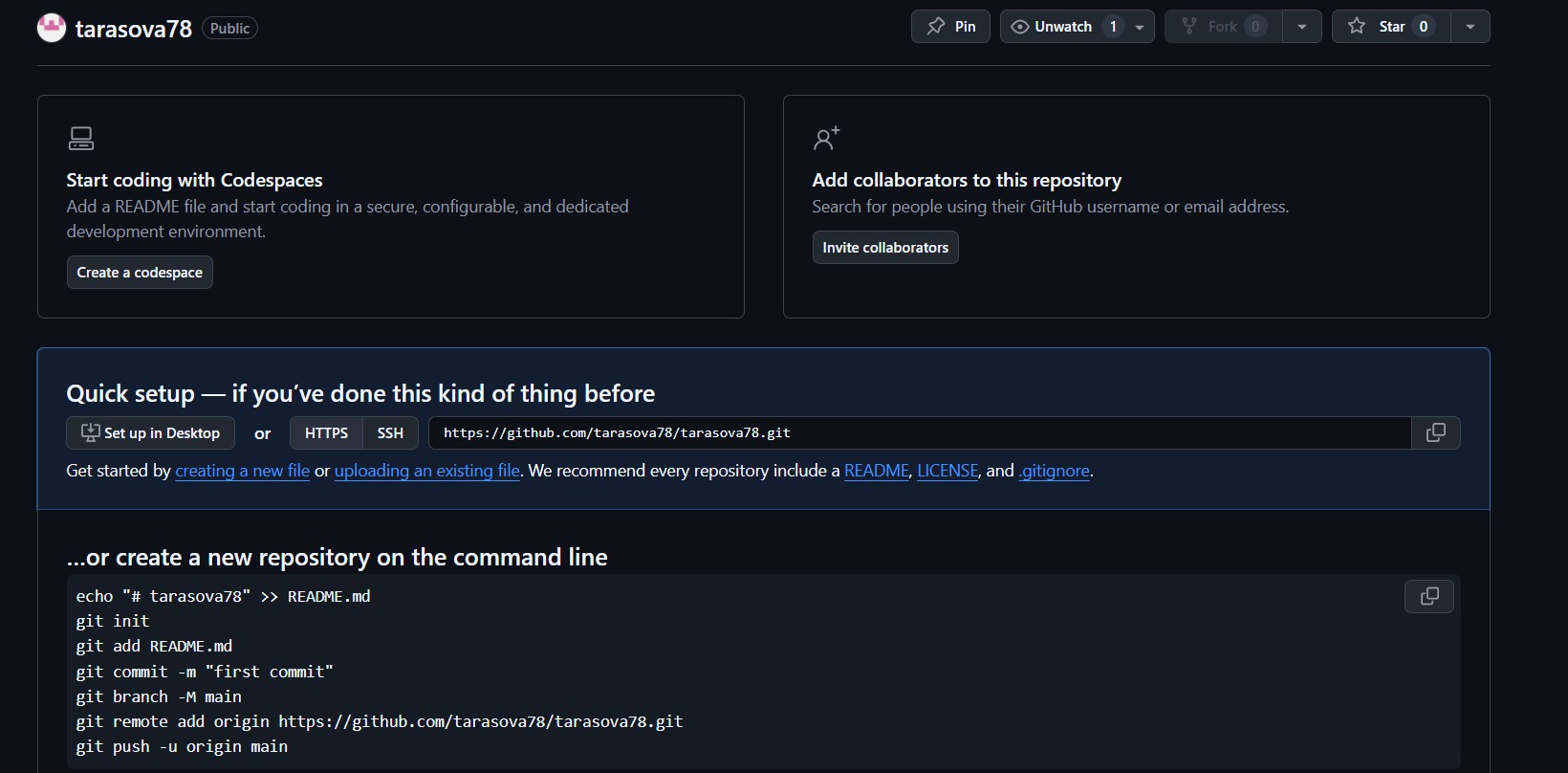


Рисунок 16 – созданный репозиторий на GitHub



Рисунок 17 – связывание локального с удаленным репозиторием



Рисунок 18 – Коммит файлов в ветке master

## Создание веток и переключение между ними

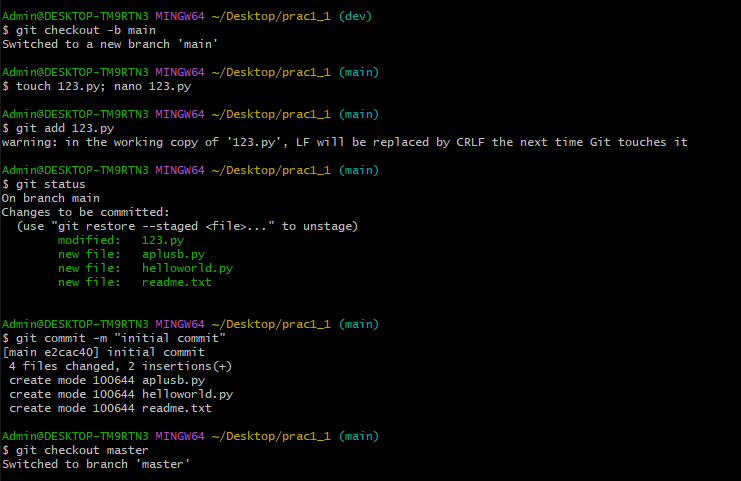


Рисунок 19 – работа в новой ветке

## Слияние веток

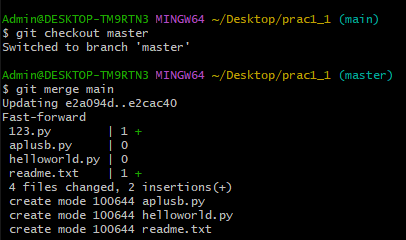


Рисунок 20 – слияние веток dev и master

## Выполните цепочку действий в репозитории, Вариант 11

* 1. Клонируйте непустой удаленный репозиторий на локальную машину

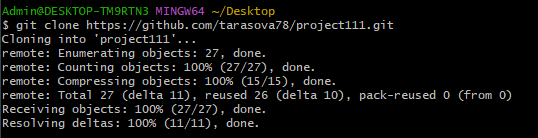


Рисунок 21 – клонирование непустого удаленного репозитория

* 1. Создайте новую ветку и выведите список всех веток

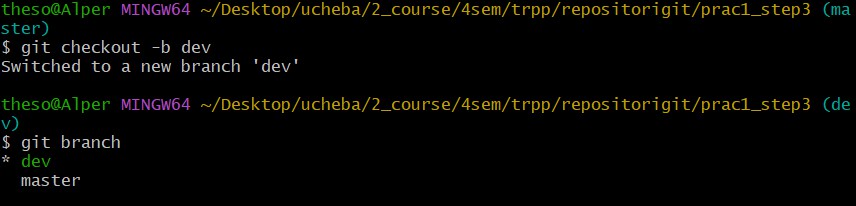
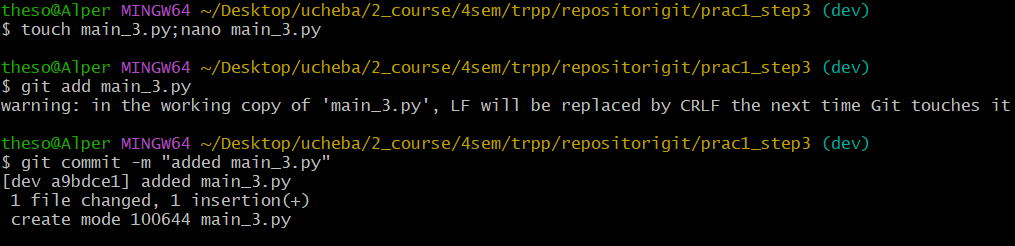
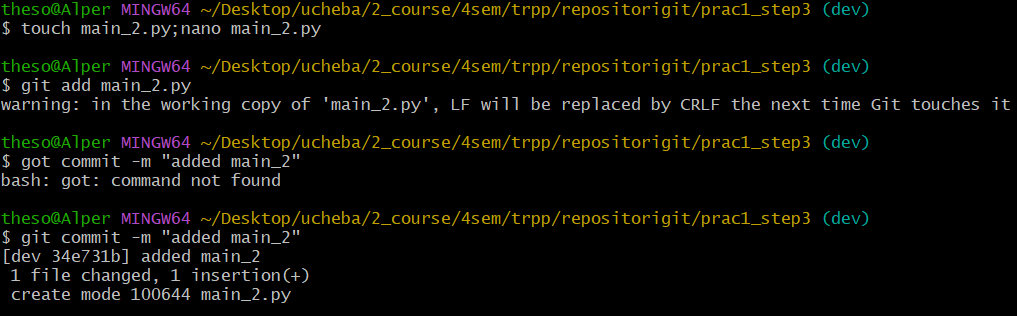
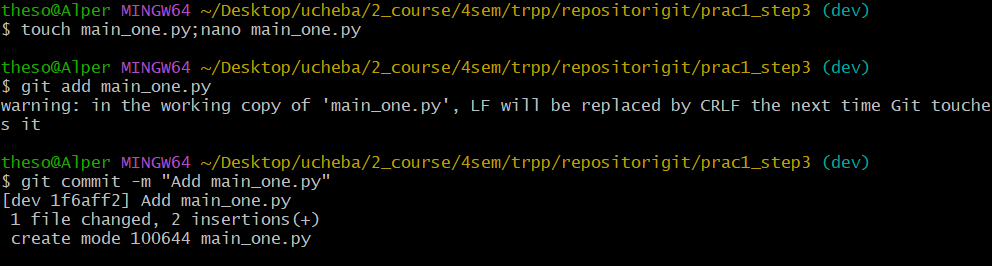


Рисунок 22 – создание новой ветки и вывод списка всех веток

* 1. Произведите 3 коммита в новой ветке



Рисунки 23, 24, 25 – 3 коммита в новой ветке

* 1. Выгрузите изменения в удаленный репозиторий

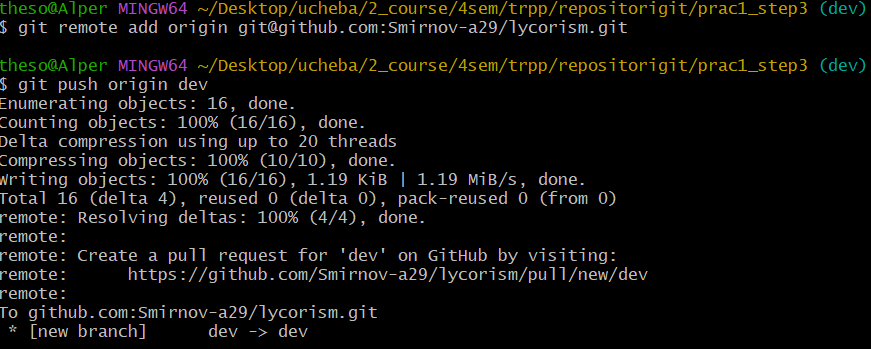


Рисунок 26 – выгрузка изменений

* 1. Склеить 3 коммита в 1

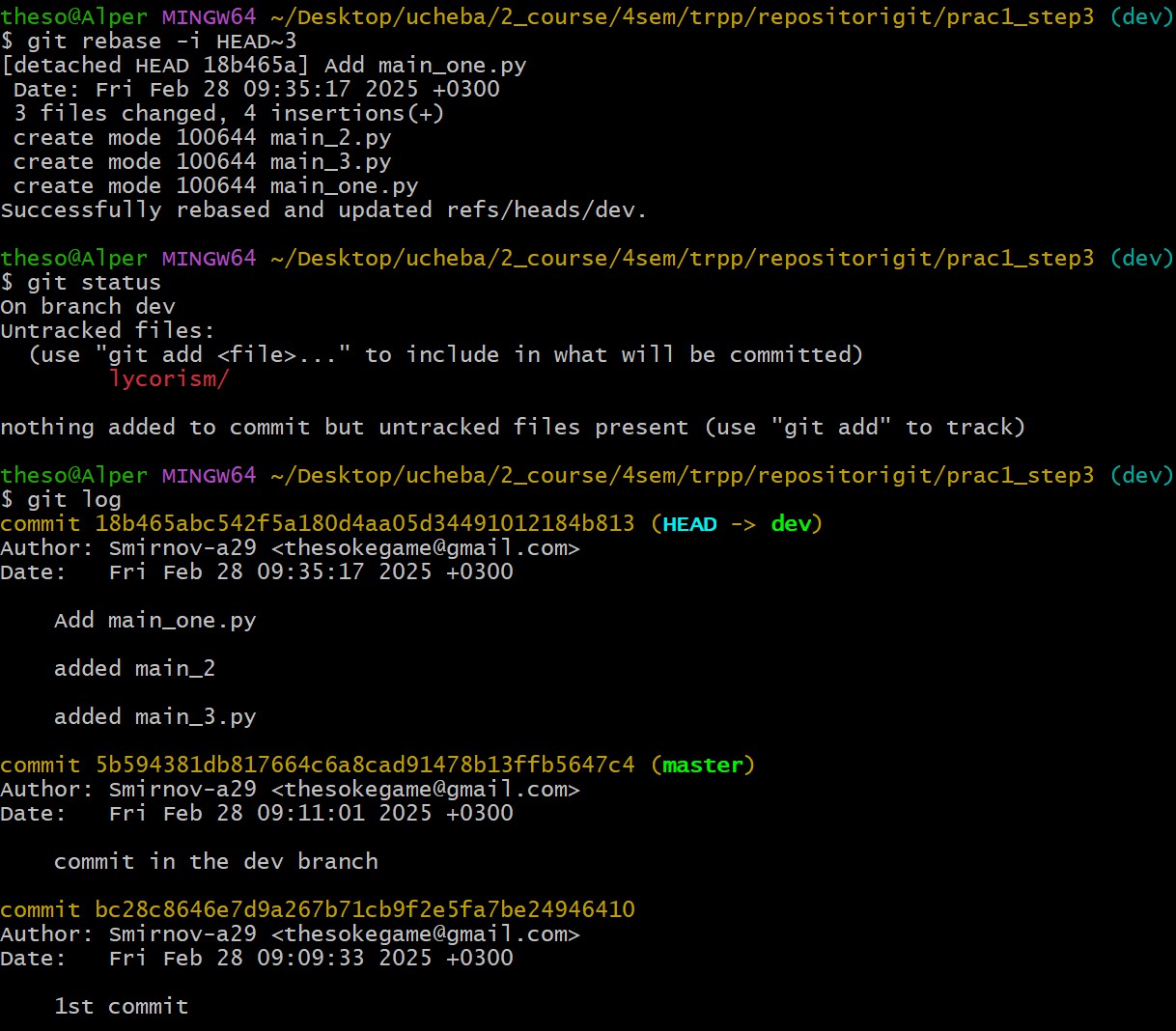


Рисунок 27 – откат на 2 коммита

Рисунок 28 – откат на 2 коммита в удаленном репозитории

* 1. Выведите в консоли различия между веткой main и dev

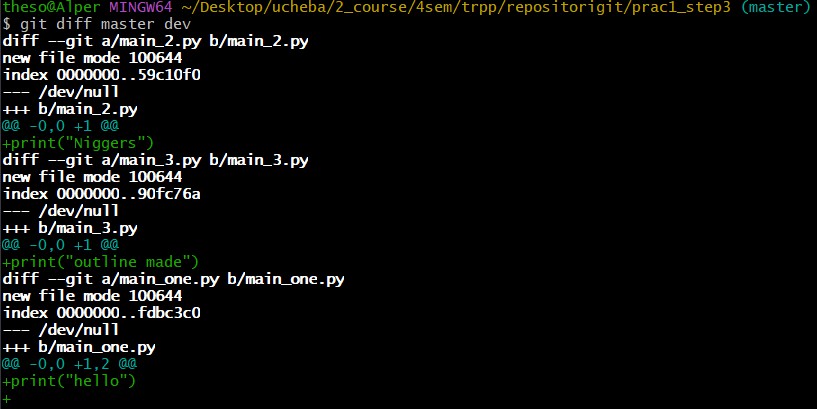


Рисунок 29 – вывод различий веток

* 1. Слейте новую ветку с main при помощи merge

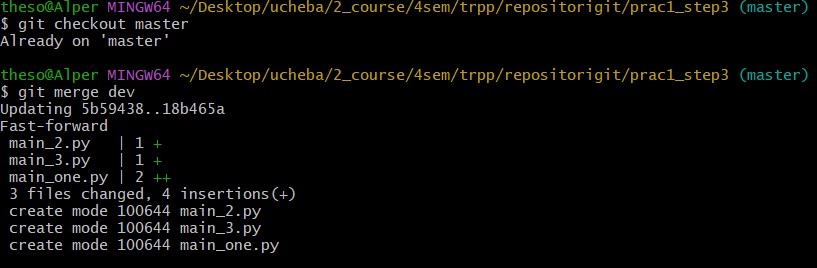


Рисунок 30 – слияние веток

# ЧАСТЬ 3 ВЕТВЛЕНИЕ И ОФОРМЛЕНИЕ КОДА

## Форк репозитория(Мой вариант 2)

## V = Oct ( N - 1 / M ) + 1 = Oct ( 22 - 1 / 10 ) + 1 = 11 + 1 = 12(2-вариант)

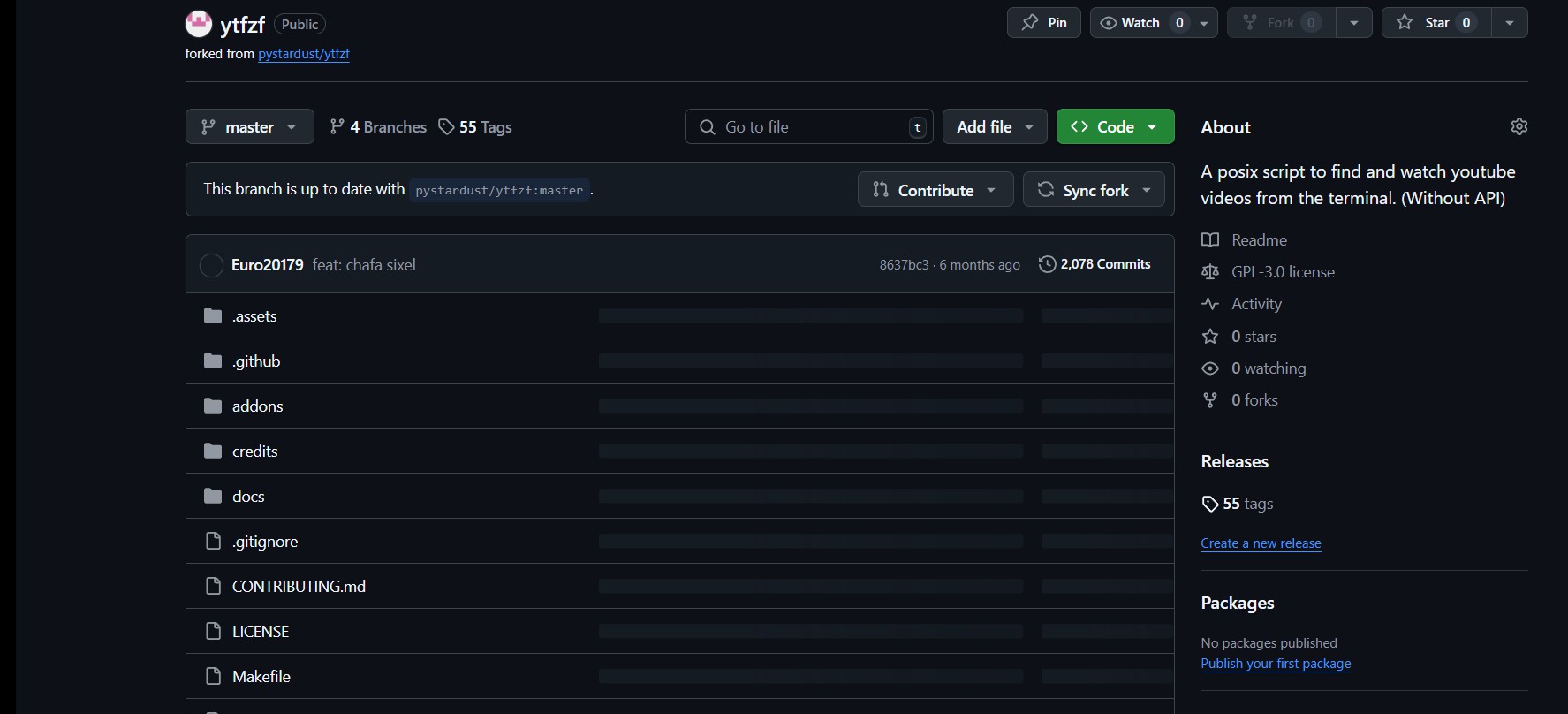


Рисунок 31 – сделанный форк репозитория

## Клонирование репозитория на локальную машину

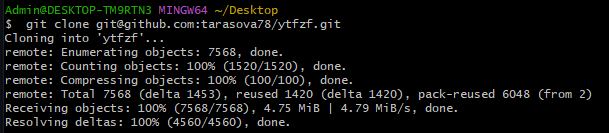


Рисунок 32 – клонирование репозитория

## Создание двух веток в main’e

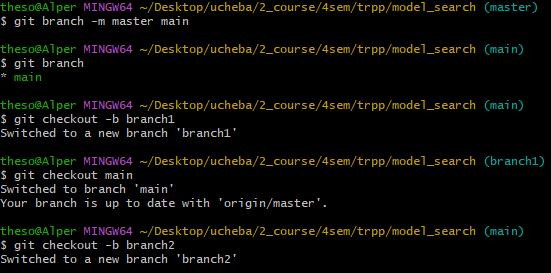


Рисунок 33 – создание двух веток

## Создание двух веток в main

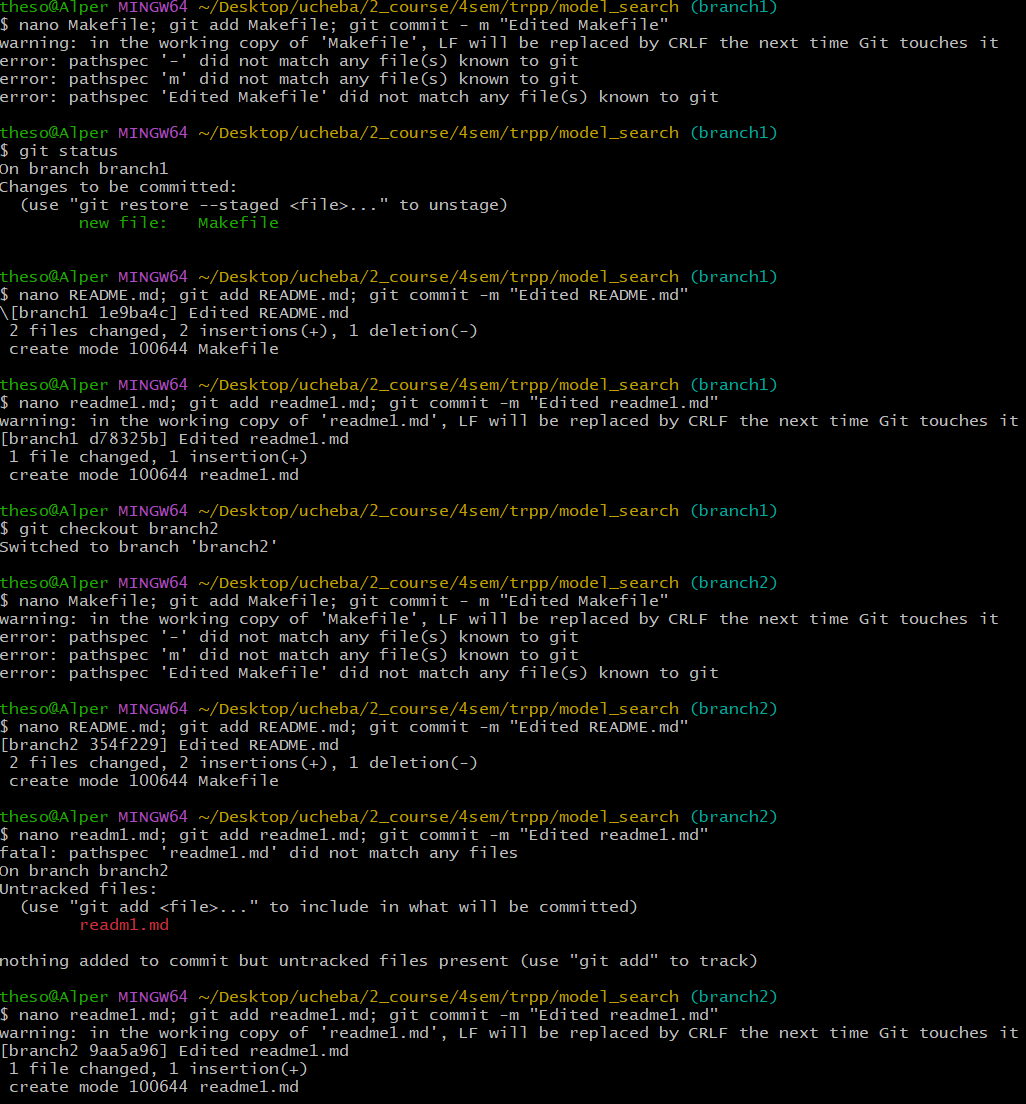


Рисунок 34 – создание 3-х коммитов

## Слияние ветки branch1 в ветку branch2 с разрешением конфликтов

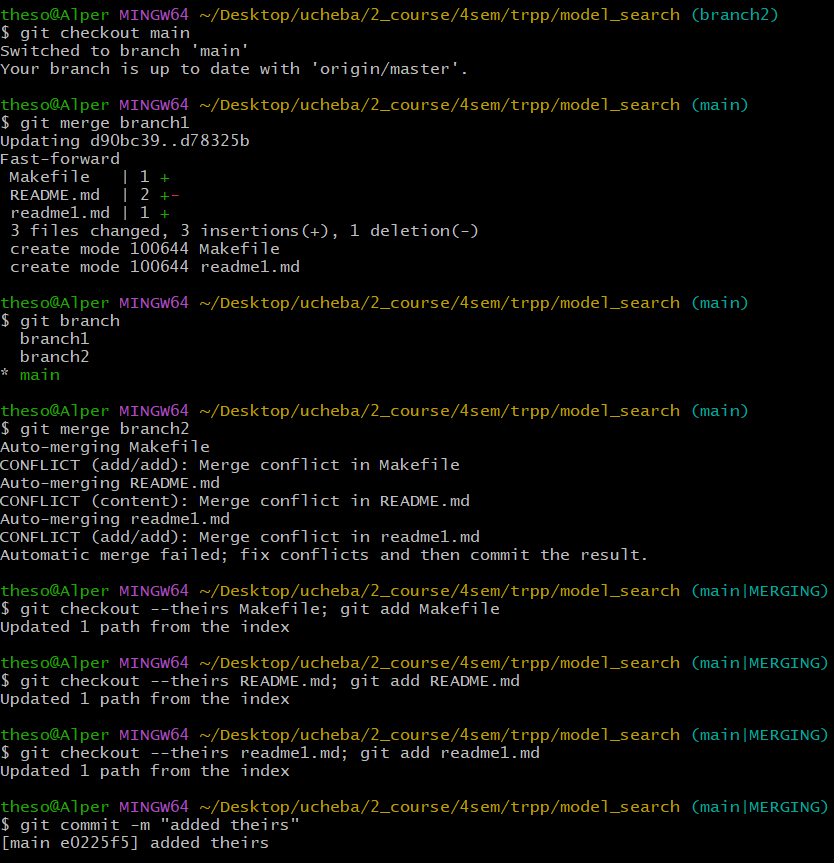


Рисунок 35 – слияние веток branch1 и branch2

## Выгрузка всех изменений во всех ветках в удаленный репозиторий

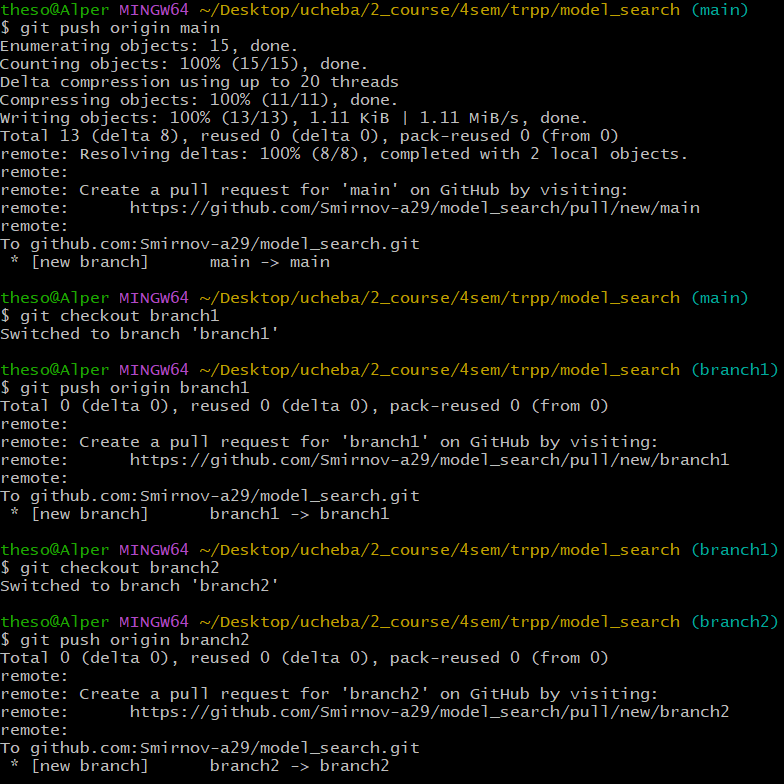


Рисунок 36 – выгрузка изменений во всех ветках в удаленный репозиторий

## Выполнение 3-х коммитов в branch1

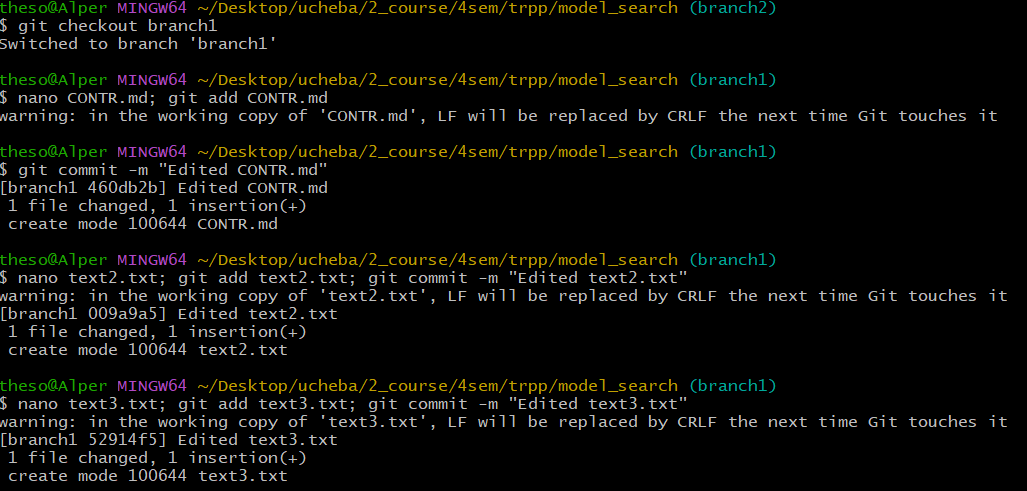


Рисунок 37 – выполнение 3-х коммитов в ветке

## Повторное клонирование репозитория в другую директорию

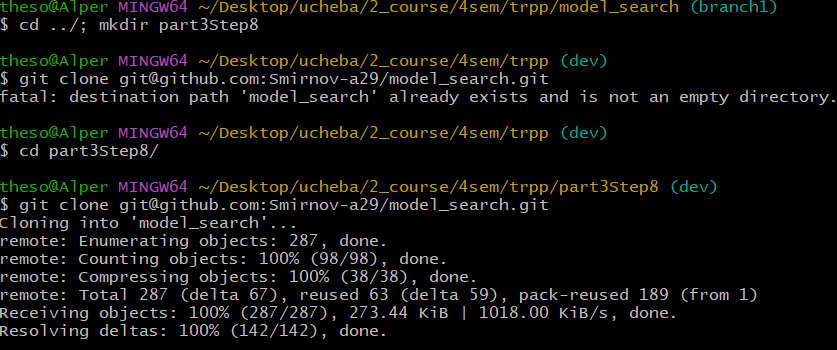


Рисунок 38 – повторное клонирование репозитория

## Повторное клонирование репозитория в другую директорию

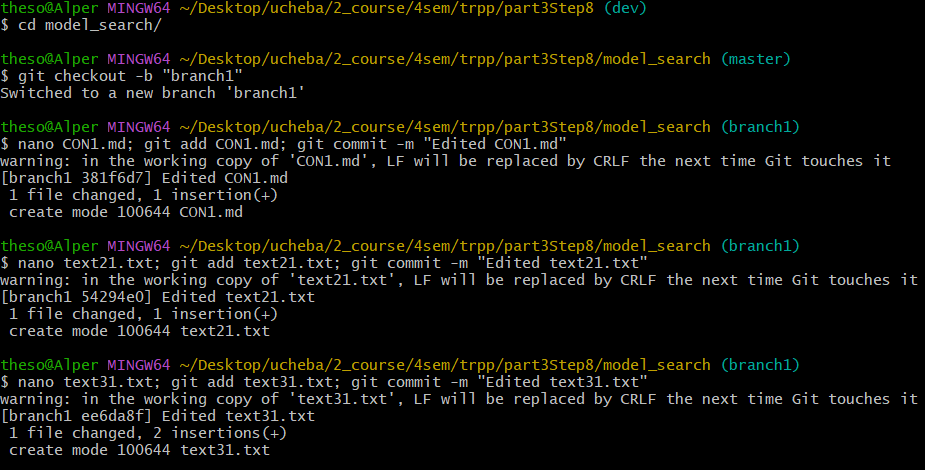


Рисунок 39 – 3 коммита в ветке branch1

## Выгрузка всех изменений в удаленный репозиторий

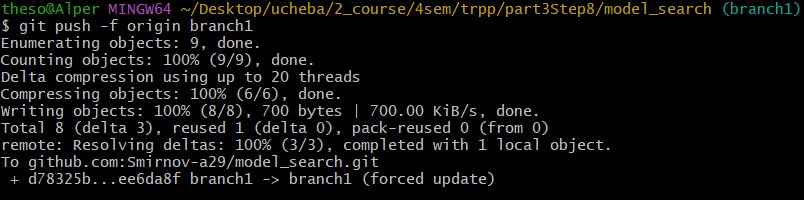


Рисунок 40 – выгрузка всех изменений

## Выгрузка всех изменений в удаленный репозиторий

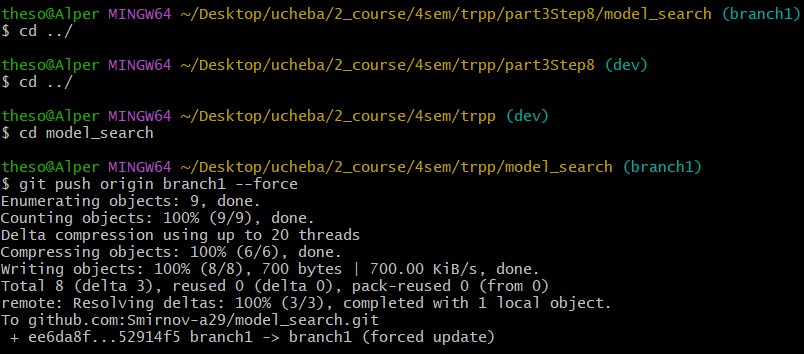


Рисунок 41 – выгрузка изменений с опцией –force

## Получение всех изменений в новом репозитории

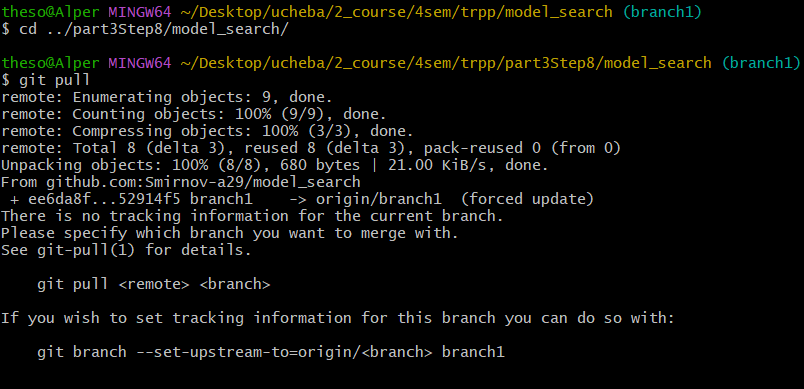


Рисунок 42 – в новом репозитории получили все изменения

# ОТВЕТЫ НА 8 ВОПРОСОВ

## Как отменить слияние веток, если произошел конфликт? Отменить слияние веток при появлении конфликта можно командой git merge --abort.

## Для чего нужен .gitignore? .gitignore используется для указания файлов и директорий, которые не должны учитываться системой контроля версий Git. Это позволяет игнорировать временные файлы, конфигурационные файлы, и прочее, чтобы они не попадали в репозиторий.

## Что делает команда git status? Команда git status показывает текущий статус репозитория, включая измененные, добавленные и неотслеживаемые файлы.

## Что делает команда git add? Команда git add добавляет изменения файлов в индекс, подготавливая их к следующему коммиту.

## Что делает команда git log? Команда git log показывает историю коммитов. Она перечисляет коммиты, которые можно достичь, следуя связям родительских коммитов от указанных коммитов.

## Что делает команда git diff? Команда git diff показывает различия между рабочим деревом и индексом, а также между индексом и последним коммитом.

## Что делает команда git show? Команда git show отображает информацию о конкретном коммите, включая изменения, внесенные этим коммитом.

## Что делает команда git stash? Команда git stash временно сохраняет изменения в рабочей директории, чтобы можно было переключиться на другую ветку или выполнить другие операции.

# ВЫВОД

Освоены основные команды Git для управление локальным репозиторием (создание, изменение индексирование, коммиты и отмена изменений). Изучена работа с тегами для меток коммитов и использование систем управления

репозиториями (GitHub), в том числе, создание, клонирование, связывание,

форк и выгрузка репозиториев по SSH-ключу. Изучено создание, переключение и удаление веток, а также разрешения конфликтов при слиянии. Познакомились с работой с ветками в удаленном репозитории и использованием опции –force для принудительной выгрузки изменений. Кроме того, мы проверили форк

репозитория и склонировали его для работы с ним. Полученный опыт работы с Git и GitHub очень важен в дальнейшей работе и в будущих проектах.