|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА - Московский технологический университет»**  **РТУ МИРЭА** |

Институт Информационных Технологий

Кафедра Математического Обеспечения и Стандартизации Информационных Технологий (МОСИТ)

**ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1**

по дисциплине

«Технология разработки программных приложений»

|  |  |
| --- | --- |
| Выполнил студент группы  ИНБО-13-20 | Черкашин А.С. (\_\_\_\_\_\_\_\_\_\_)  *подпись* |
| Приняла | Овчинникова М.А. (\_\_\_\_\_\_\_\_\_\_)  *подпись* |
| Практическая работа выполнена | «\_\_»\_\_\_\_\_\_\_20\_\_ г. |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_20\_\_ г. |

Москва 2022

**Оглавление**

[Часть 1. Основные команды git 2](#_Toc95596293)

# **Часть 1. Основные команды git**

Установим и настроим клиент git на своей рабочей станции.

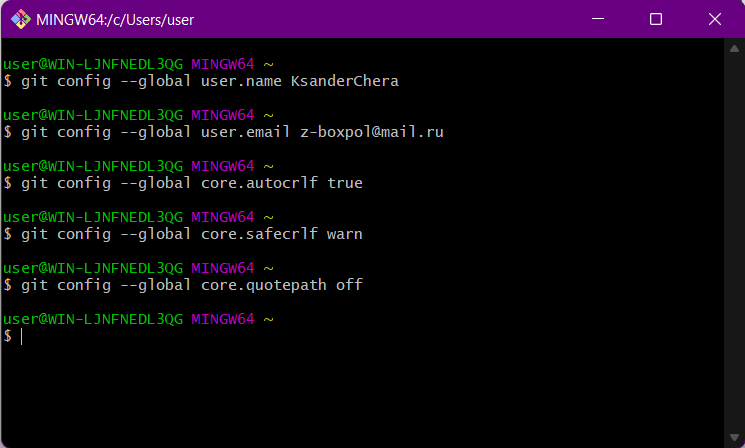


Рисунок 1 – Настройка клиента git

Начнём работу в пустом рабочем каталоге с создания пустого каталога с именем «hello», затем войдём в него и создадим там файл с именем hello.html. Чтобы создать git репозиторий из этого каталога, выполним команду git init. Теперь добавим в репозиторий страницу «Hello, World».

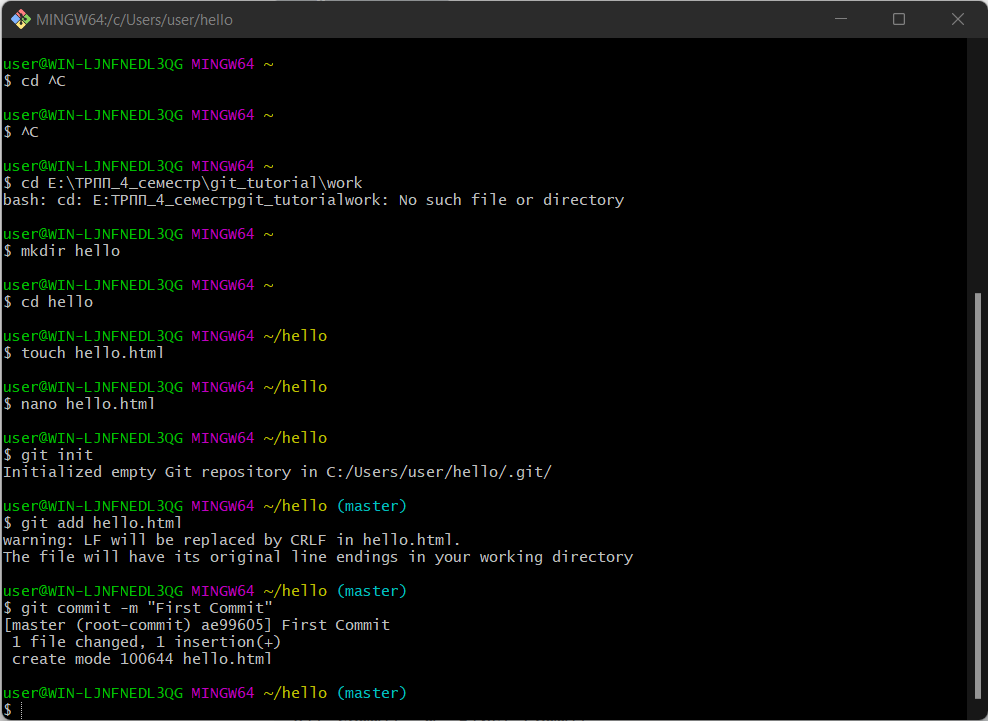


Рисунок 2 – Создание репозитория и коммит

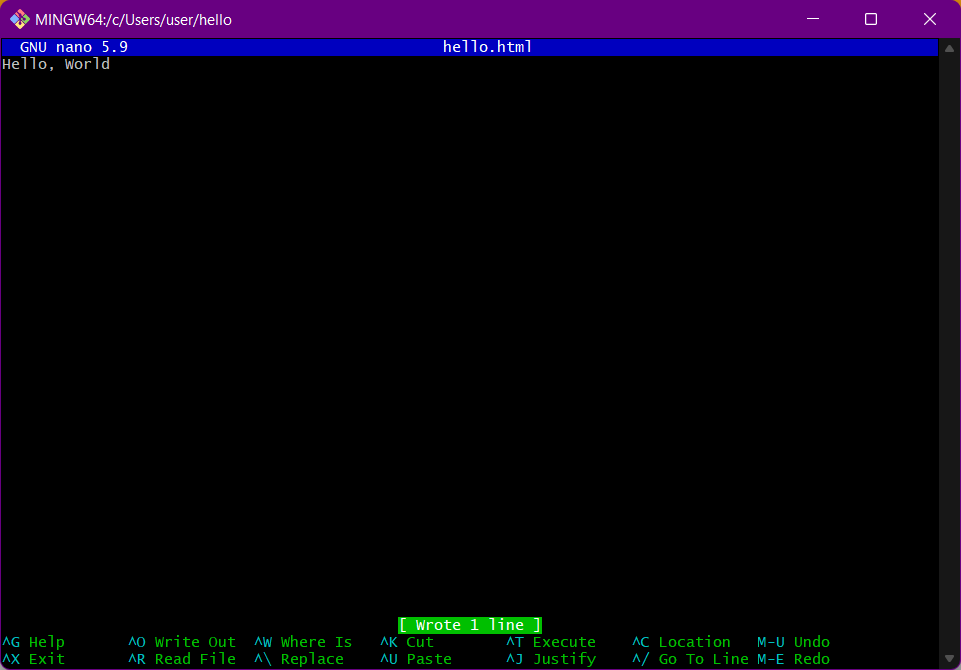


Рисунок 3 – Содержимое файла

Используем команду git status, чтобы проверить текущее состояние репозитория.

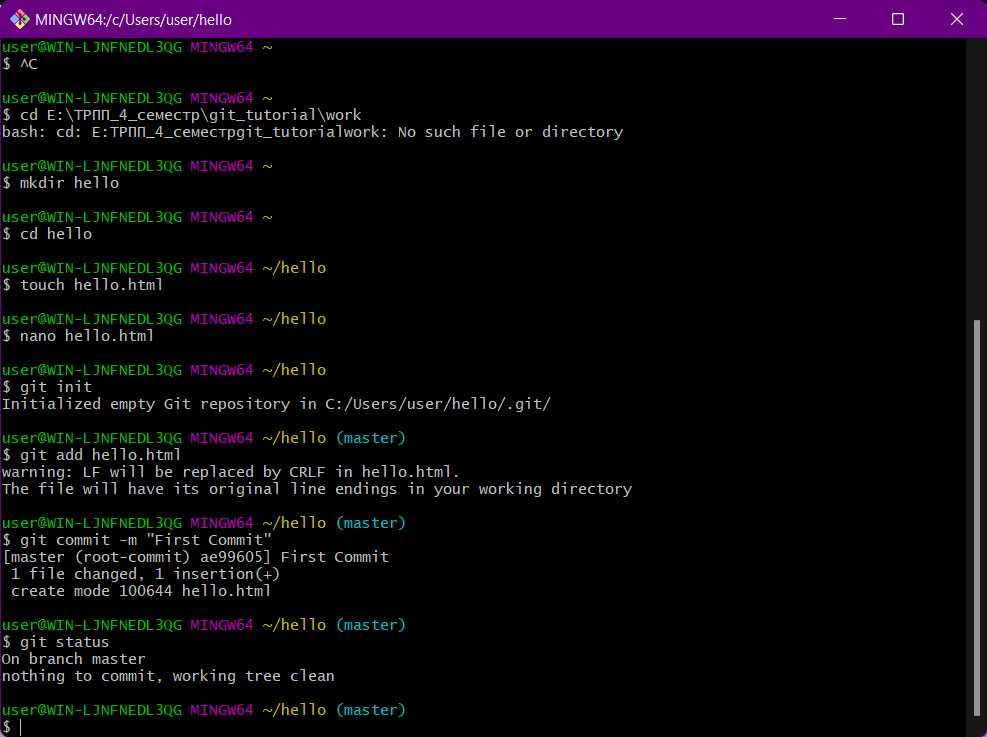


Рисунок 4 – Проверка состояния репозитория

Добавим кое-какие HTML-теги к нашему приветствию. И проверим состояние рабочего каталога.

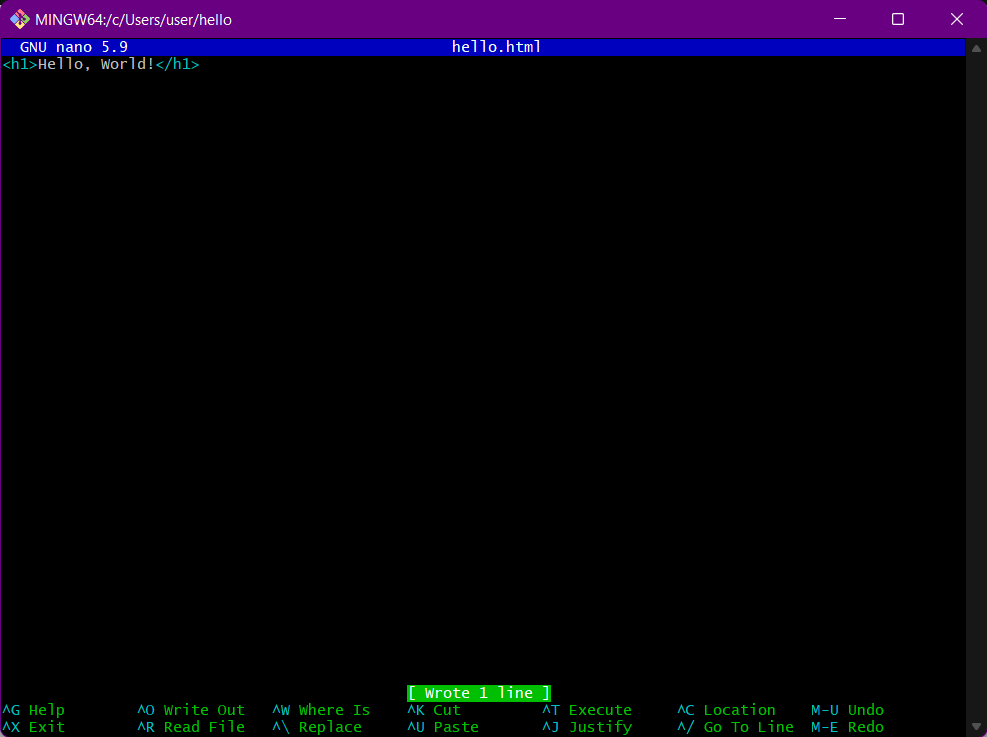


Рисунок 5 – Содержимое файла

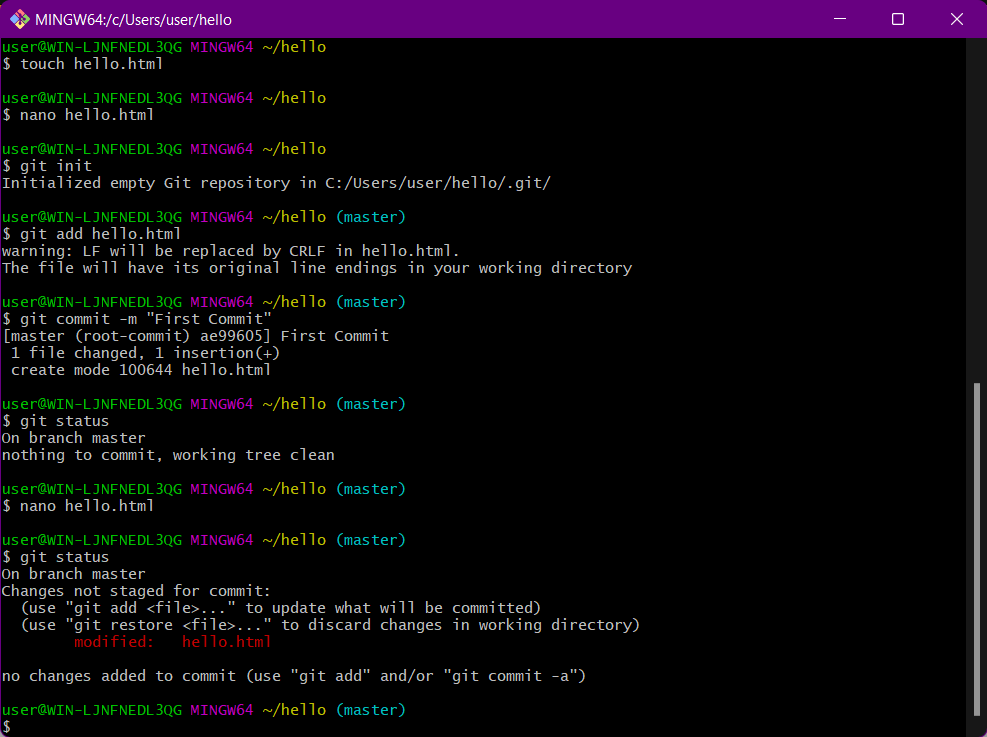


Рисунок 6 - Проверка состояния репозитория

Теперь проиндексируем изменения и проверим состояние.

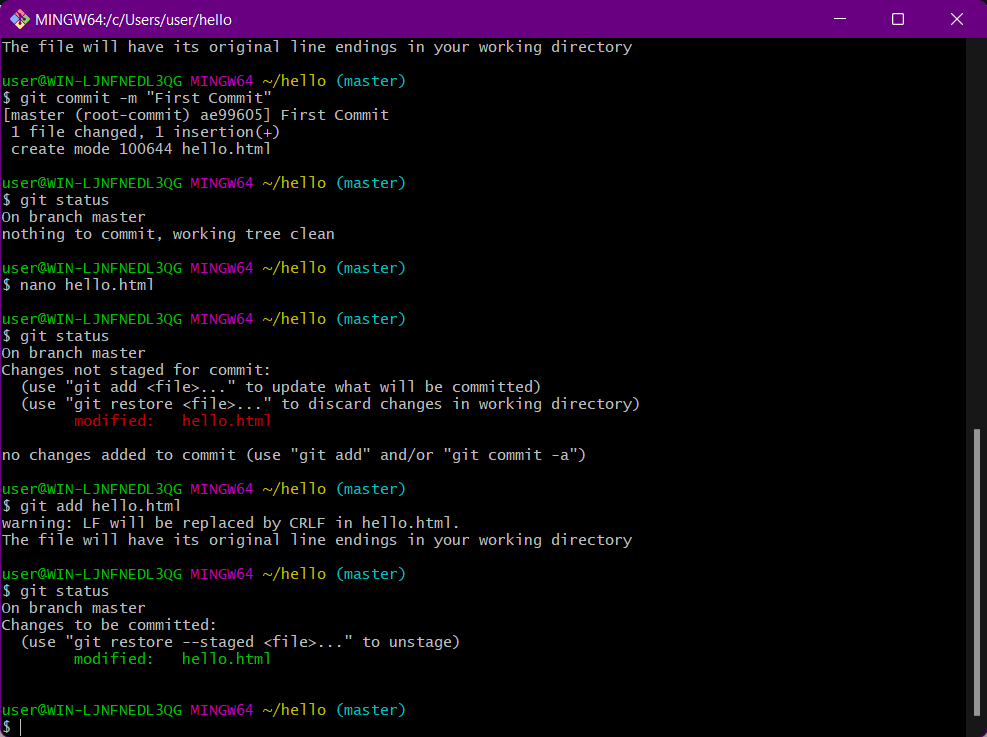


Рисунок 7 - Проверка состояния репозитория после индексации

Сделаем коммит того, что мы проиндексировали, в репозиторий. В конце еще раз проверим состояние.

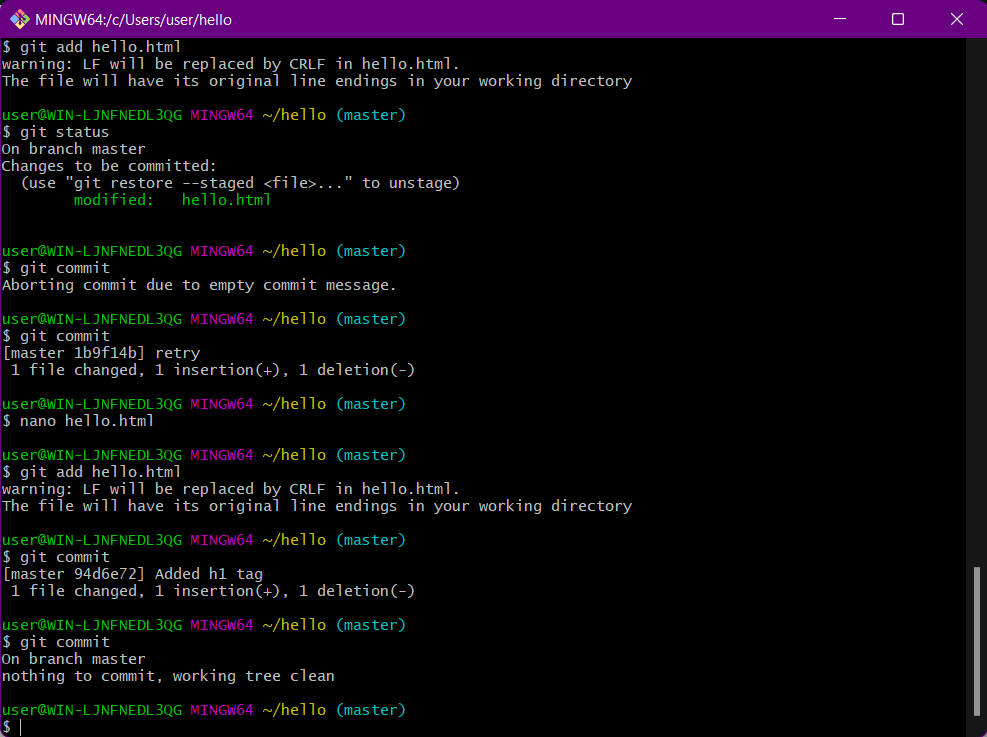


Рисунок 8 – Коммит и проверка

Изменим страницу «Hello, World». Добавим это изменение в индекс git. Теперь добавим заголовки HTML. И проверим текущий статус.

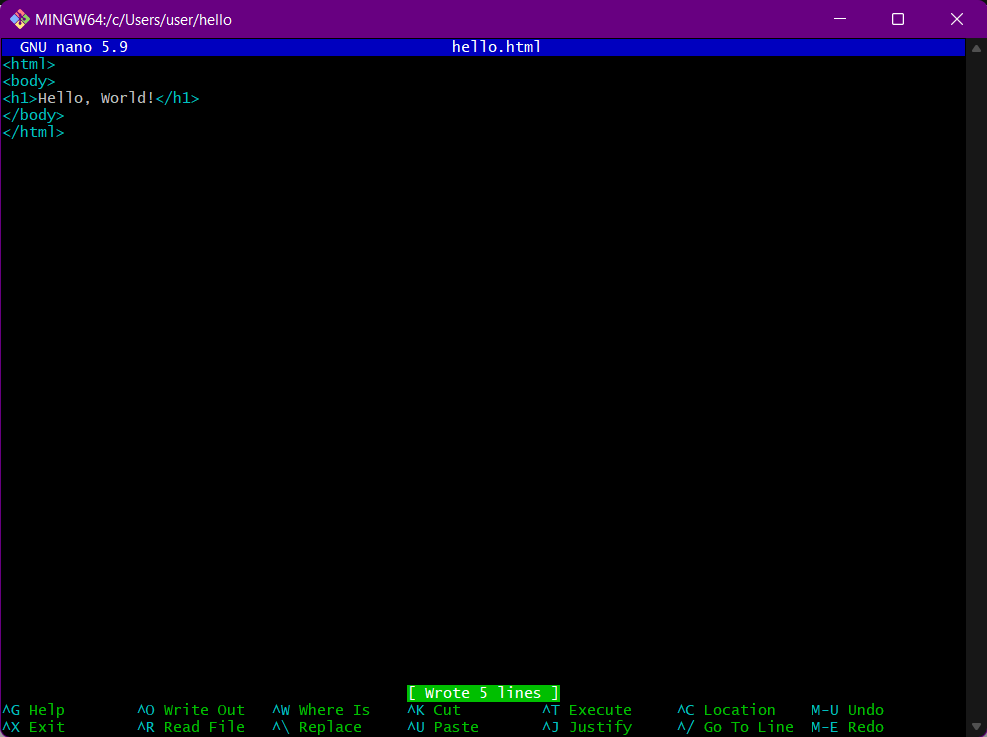


Рисунок 9 – Содержимое файла

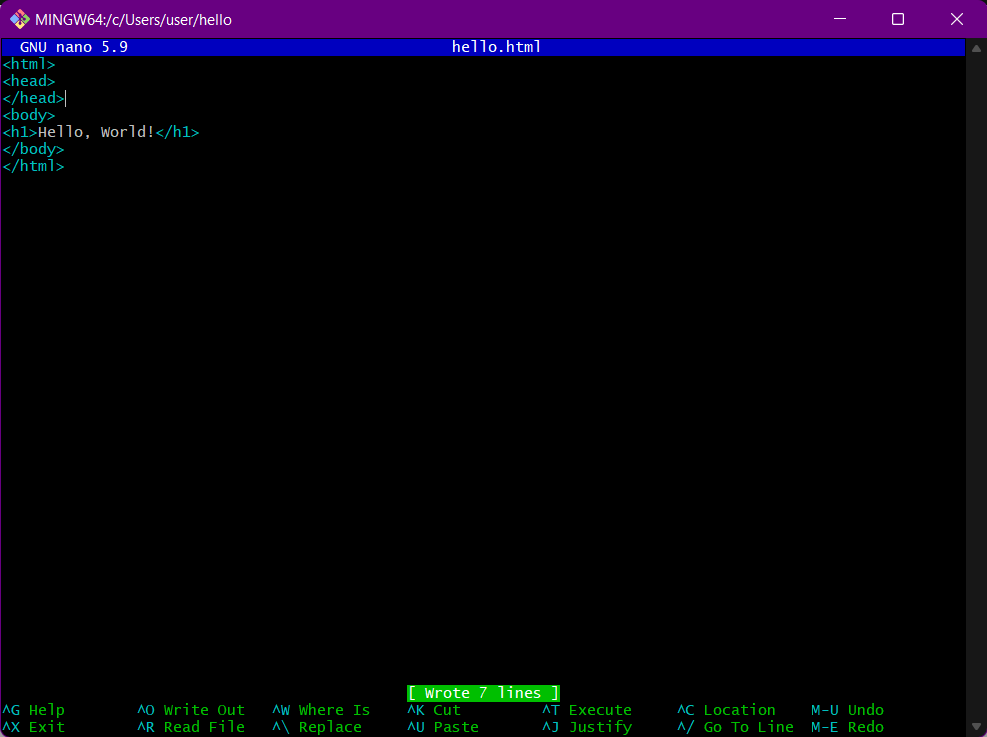


Рисунок 10 – Изменение содержимого файла после индексации

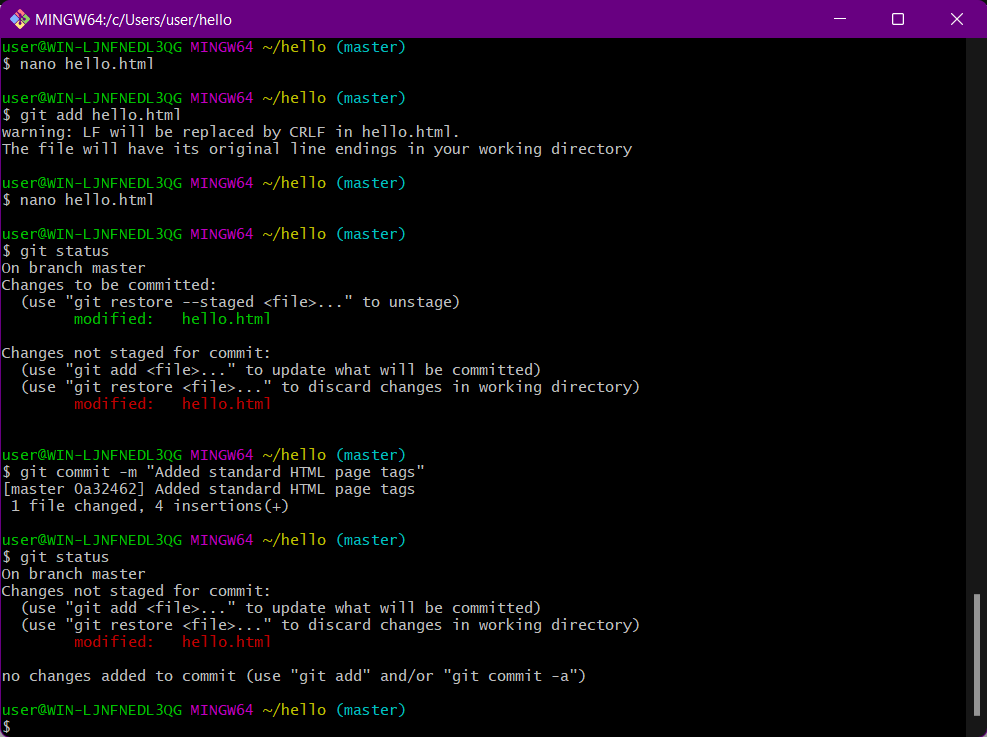


Рисунок 11 – Проверка статуса

Произведём коммит проиндексированного изменения, а затем еще раз проверим состояние.

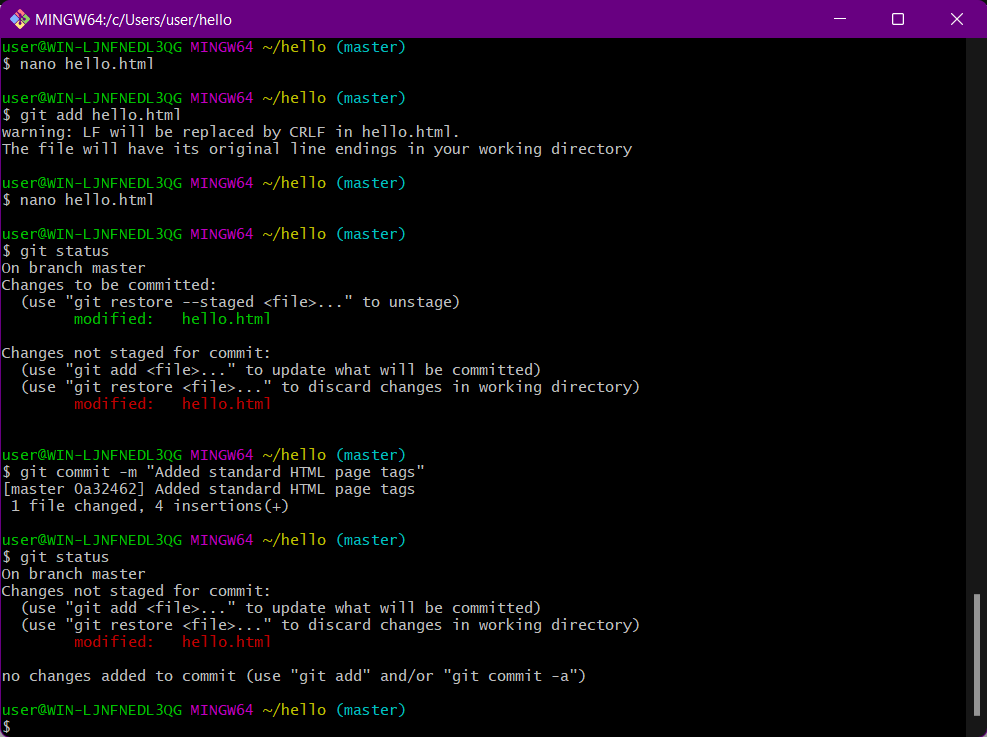


Рисунок 12 – Коммит и проверка состояния

Теперь добавим второе изменение в индекс, а затем проверим состояние.

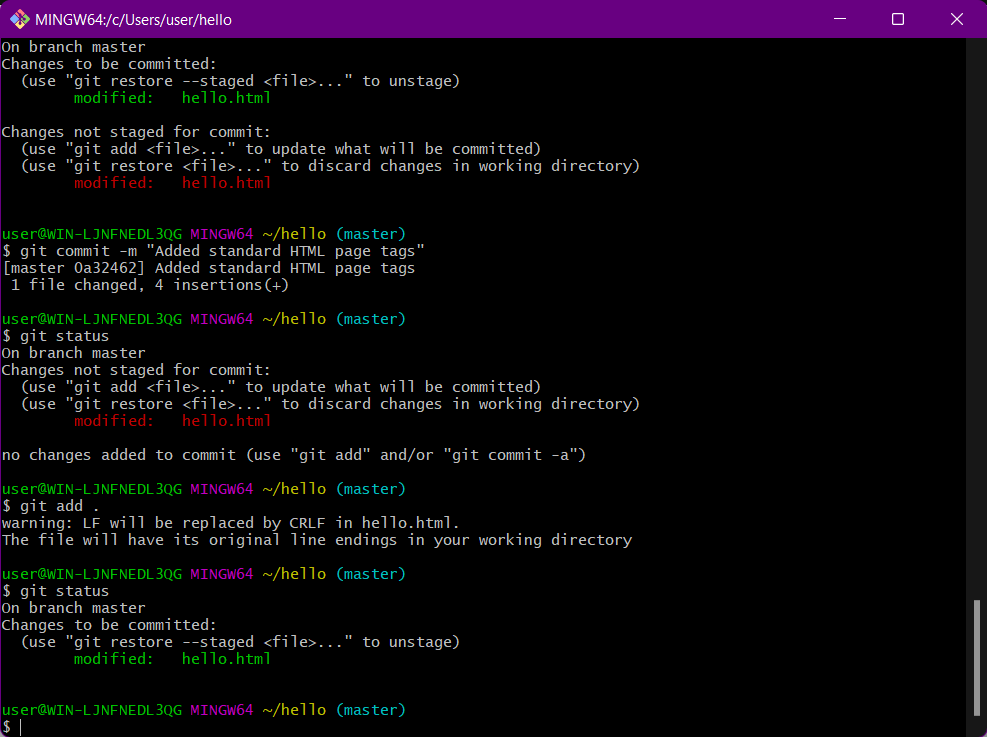


Рисунок 13 – Индексация и проверка

Сделаем коммит второго изменения.

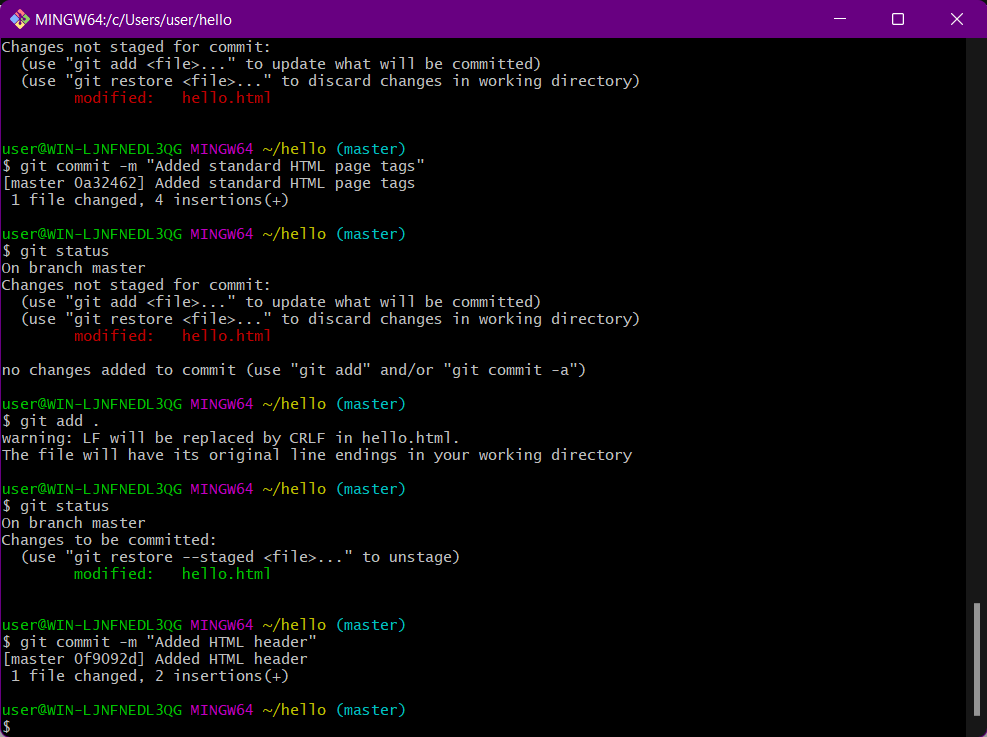


Рисунок 14 – Коммит

Просмотрите историю коммитов с помощью команды git log. Ознакомьтесь с параметрами команды и используйте некоторые из них для различного формата отображения истории коммитов.

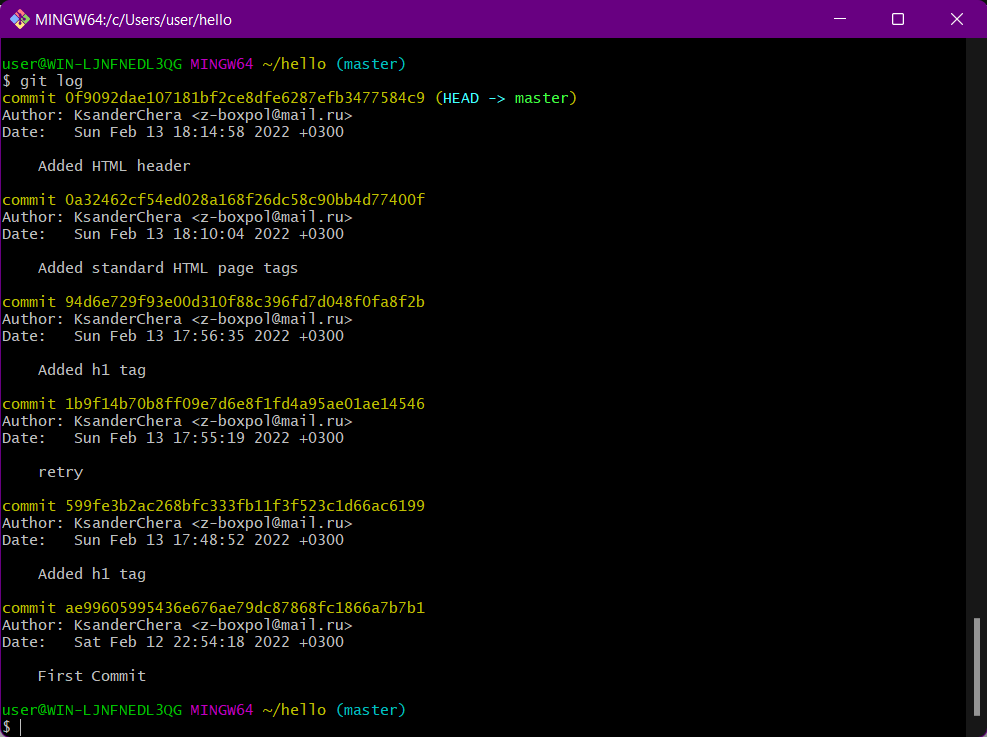


Рисунок 15 – История коммитов

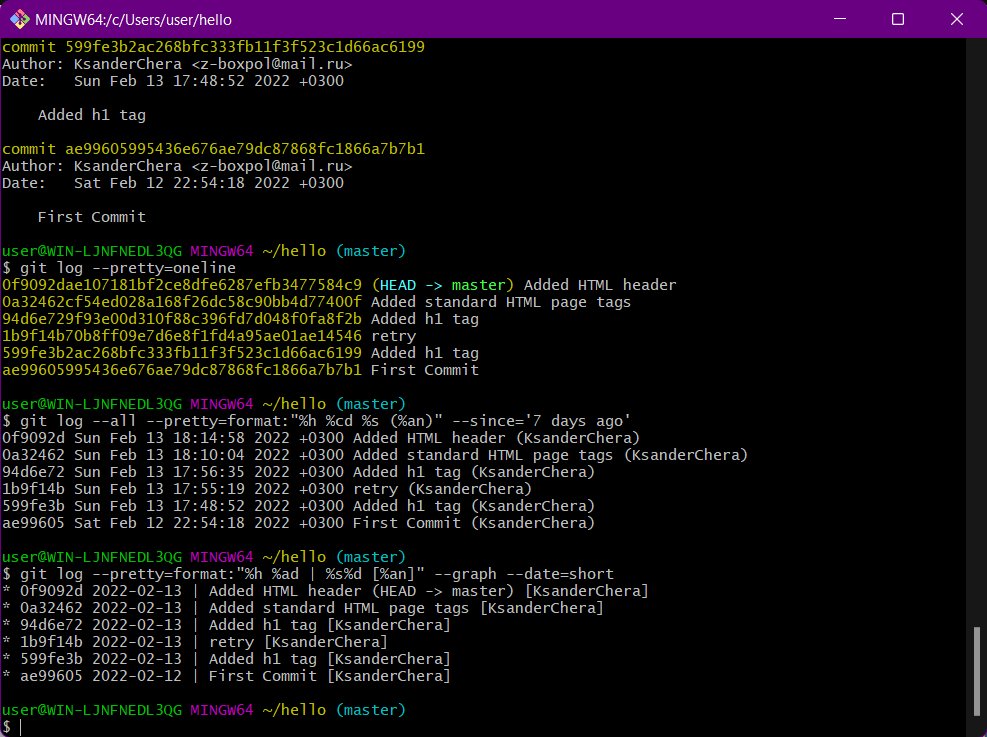


Рисунок 16 – Форматированная история коммитов

Вернём рабочий каталог к одному из предыдущих состояний.

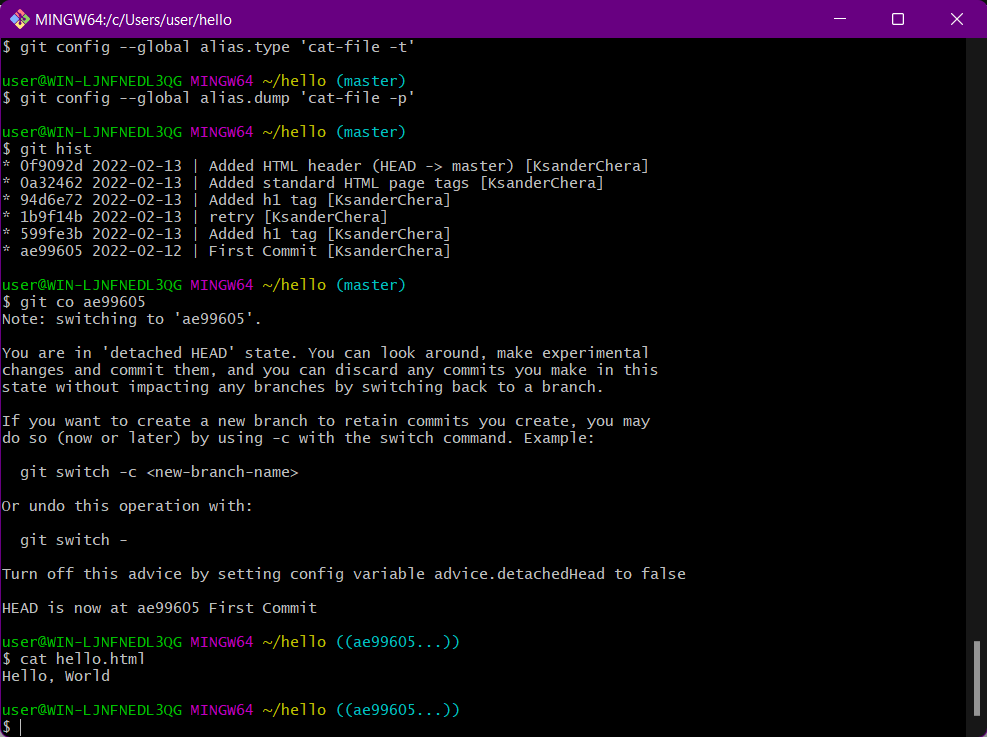


Рисунок 17 – Возвращение рабочего каталога к предыдущему состоянию

Назовем текущую версию страницы hello первой (v1). Создадим тег для версии, которая идет перед текущей версией и назовем его v1-beta. Попробуем переключаться между двумя отмеченными версиями. Можем увидеть, какие теги доступны, используя команду git tag. Также можем посмотреть теги в логе.

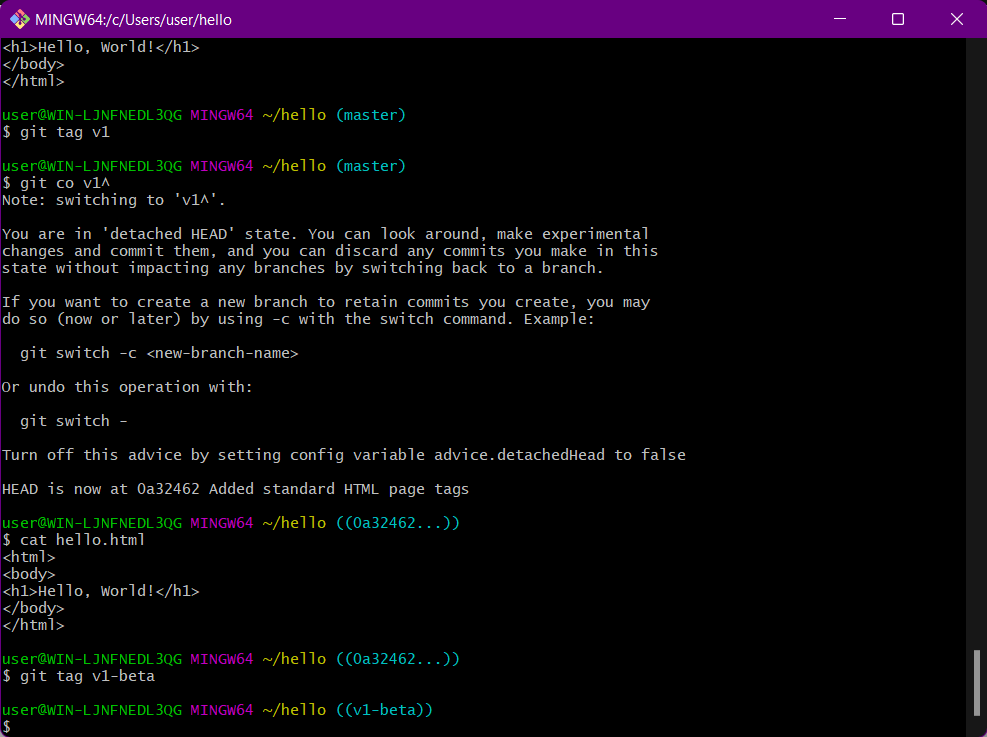


Рисунок 18 – Создание тега

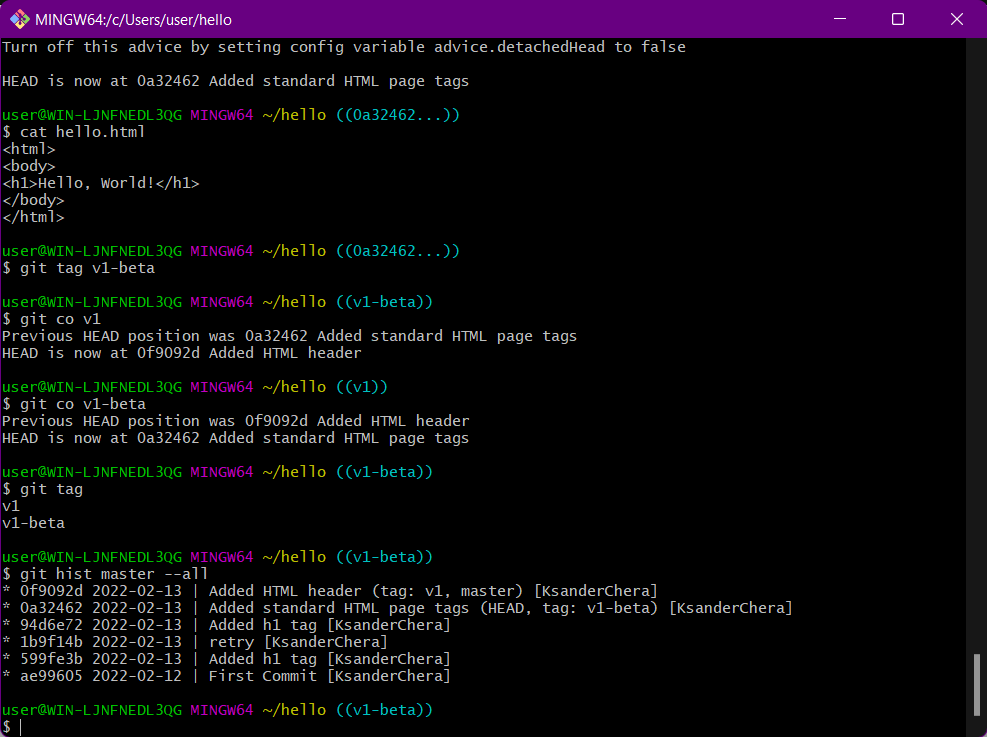


Рисунок 19 – Переключение между тегами, просмотр тегов в логах

Отменим некоторые изменения в рабочем каталоге (до и после индексирования).

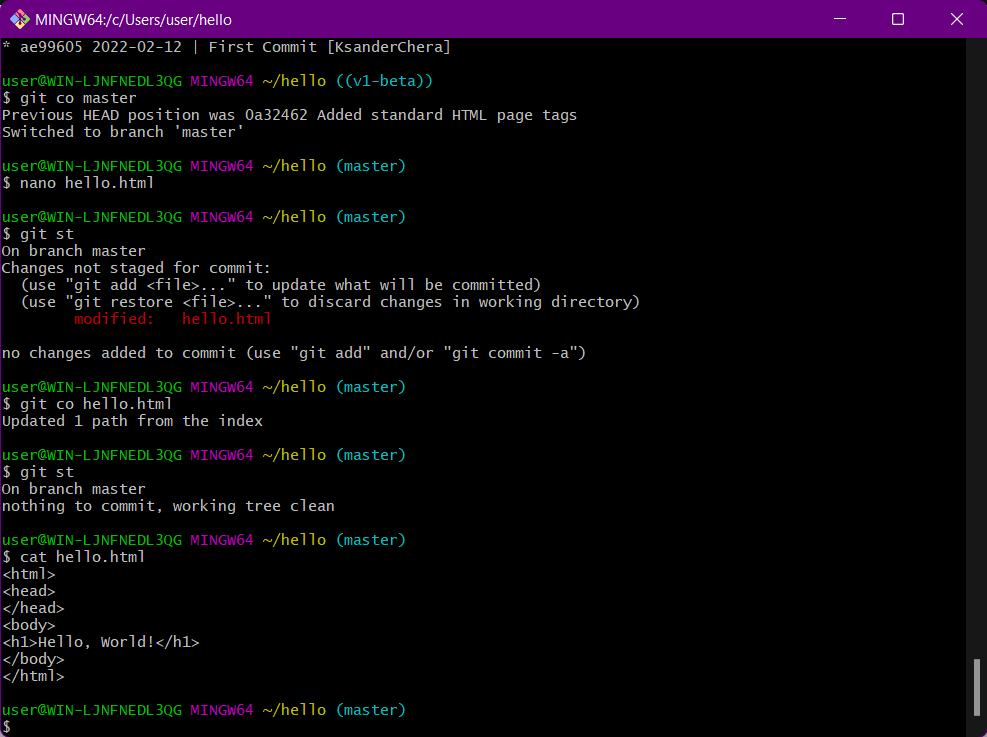


Рисунок 20 – Отмена изменений до индексирования

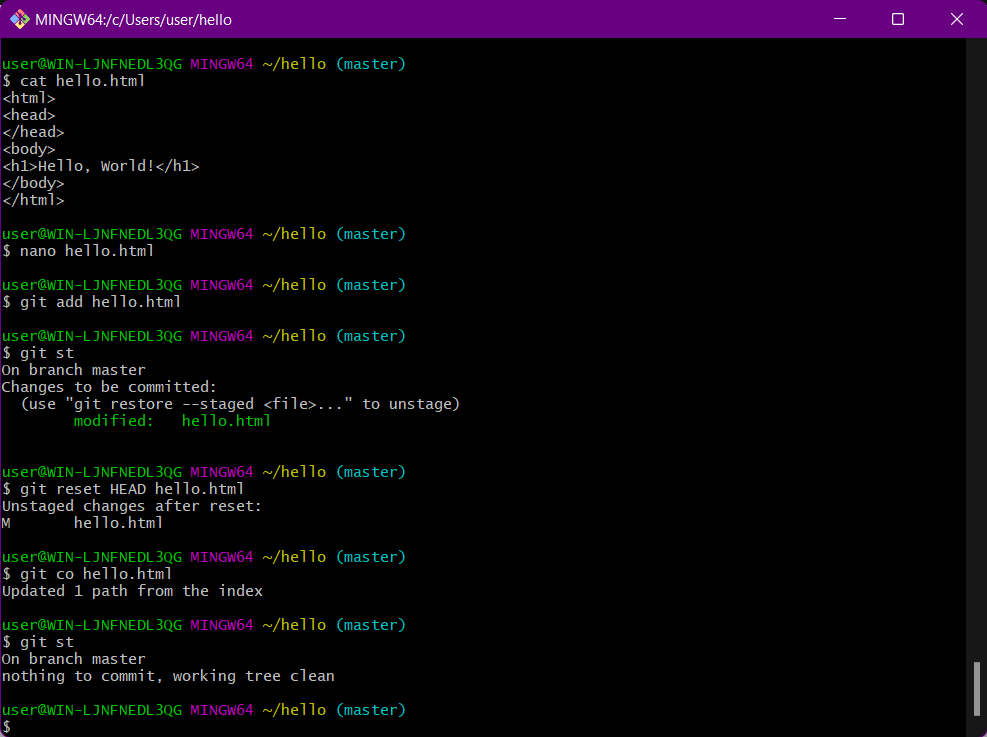


Рисунок 21 – Отмена изменений после индексирования

Отменим один из коммитов в локальном репозитории.

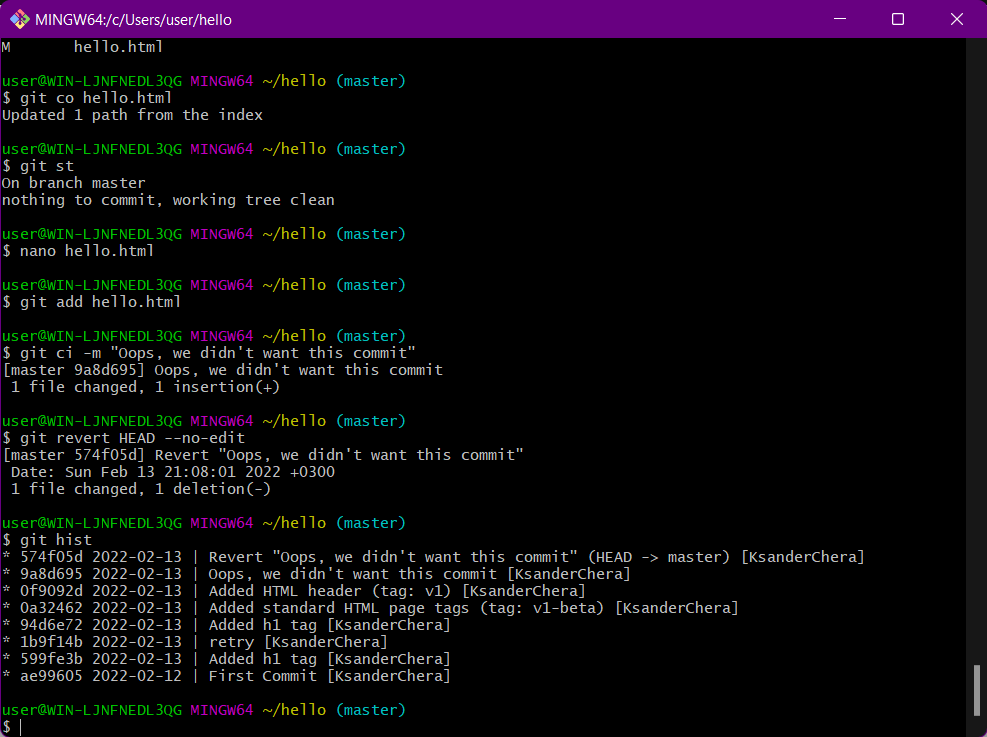


Рисунок 22 - Коммит с новыми изменениями, отменяющими предыдущие

Удалим самые последние коммиты из ветки с помощью команды reset.

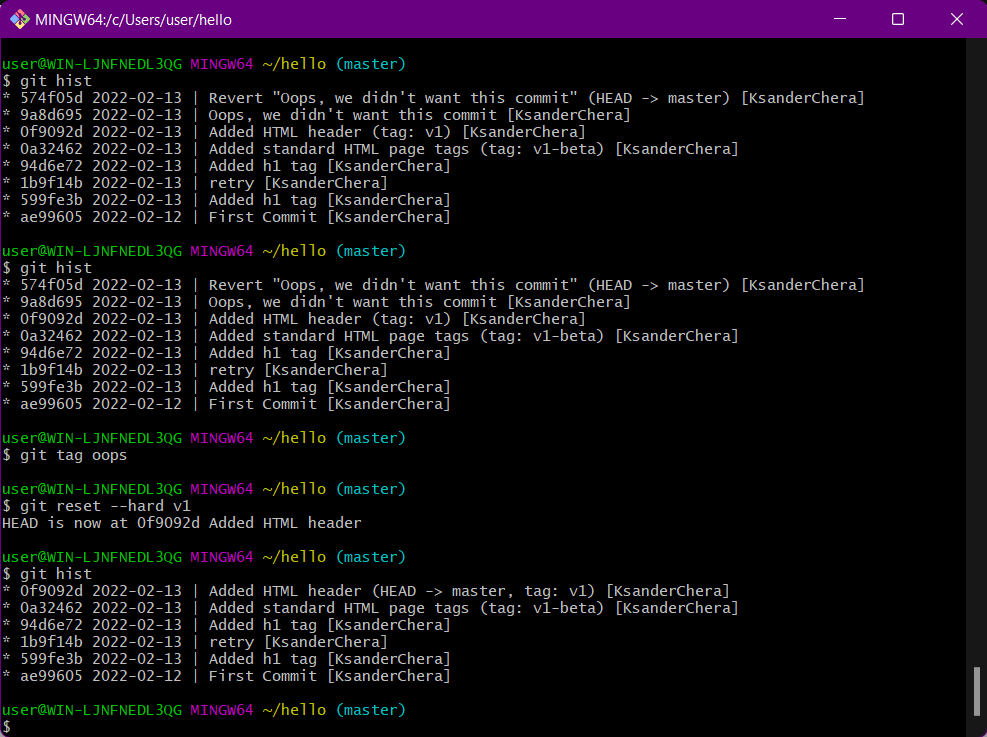


Рисунок 23 - Сброс коммитов к предшествующим коммиту Oops

Удаление тега oops.

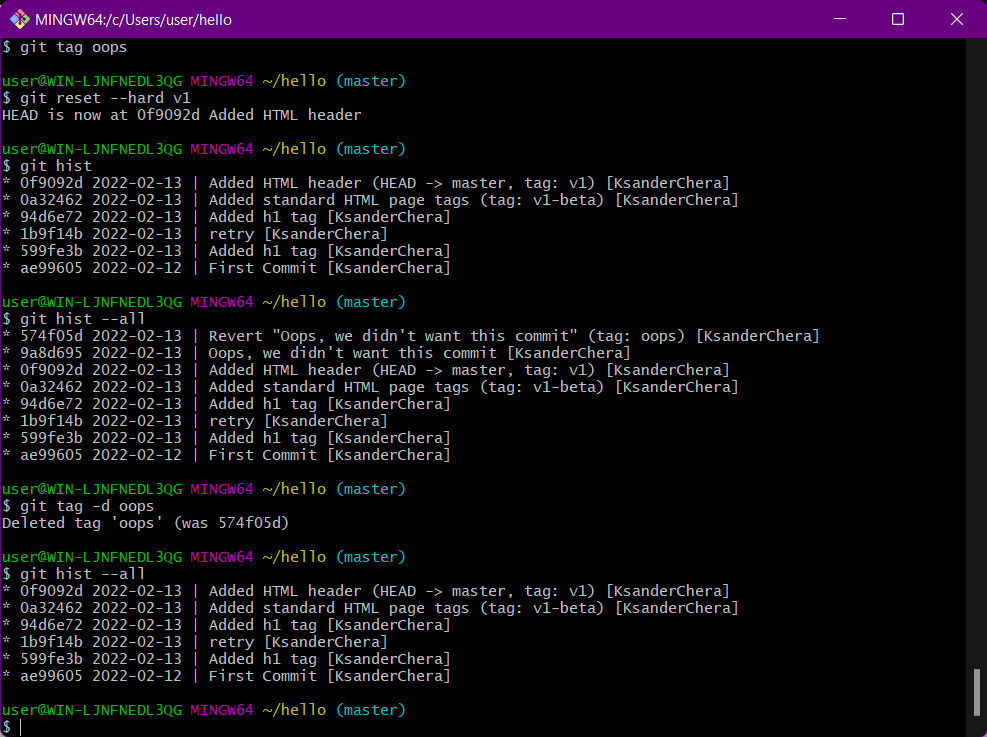


Рисунок 24 – Удаление тега

Изменим существующие коммиты.

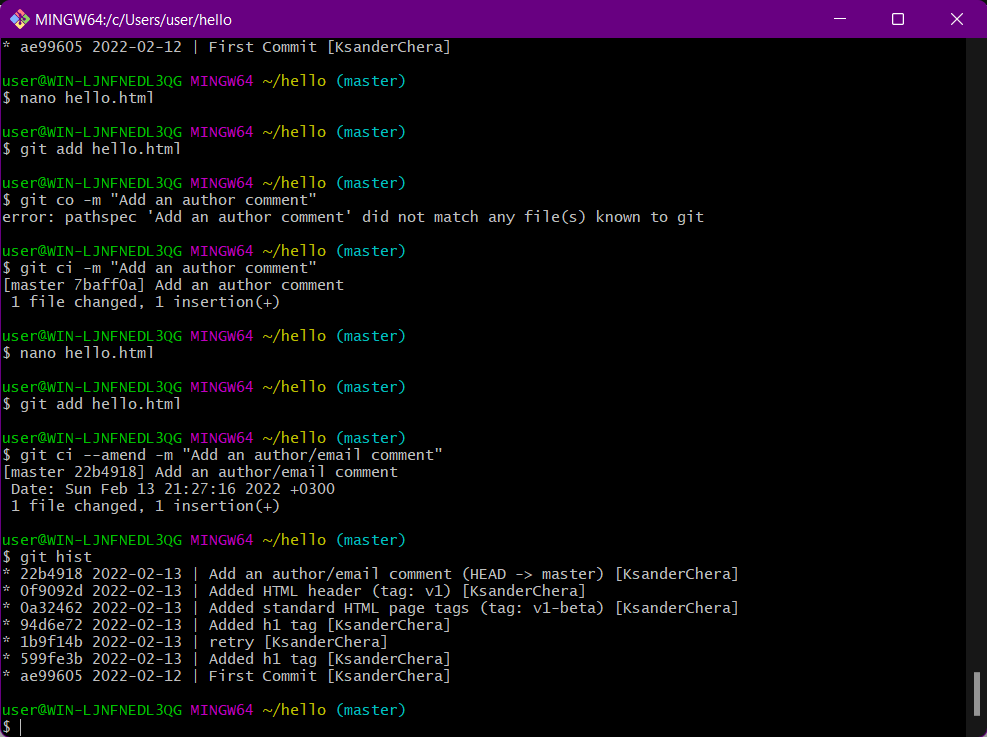


Рисунок 25 – Изменение предыдущего коммита с помощью --amend

Переместим файл hello.html в каталог lib.

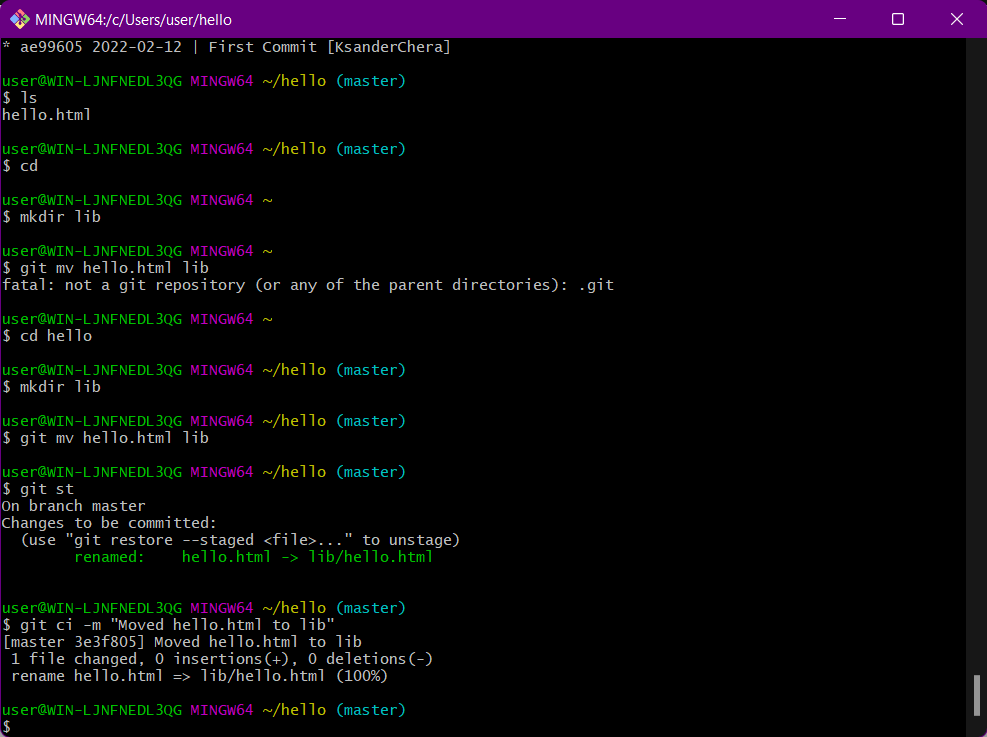


Рисунок 26 – Перемещение файла

# **Часть 2. Ветвление и разрешение конфликтов**

Поиск последнего коммита. Вывод последнего коммита. Поиск дерева. Вывод каталога lib. Вывод файла hello.html.

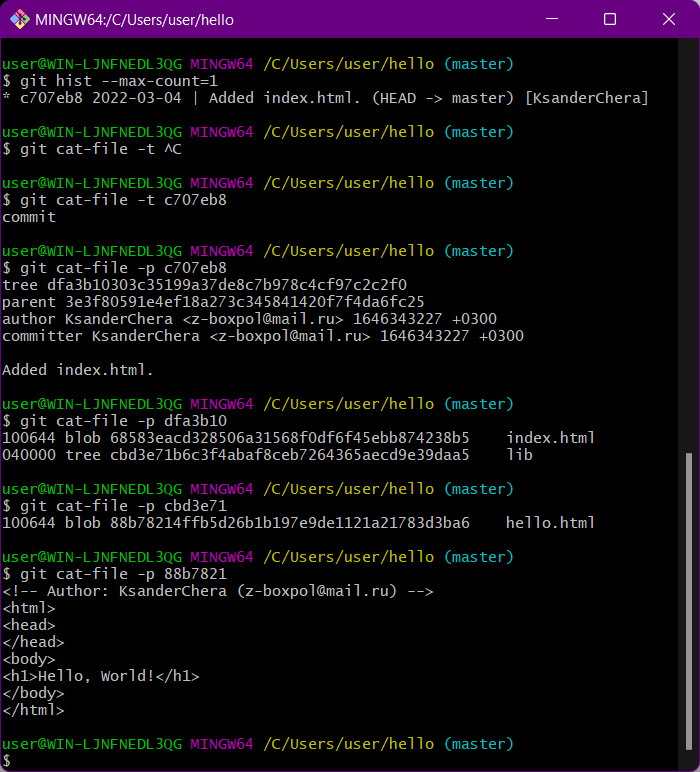


Рисунок 27 - Работа с объектами git

Создать локальную ветку в репозитории. Добавить файл стилей style.css. Измените основную страницу.

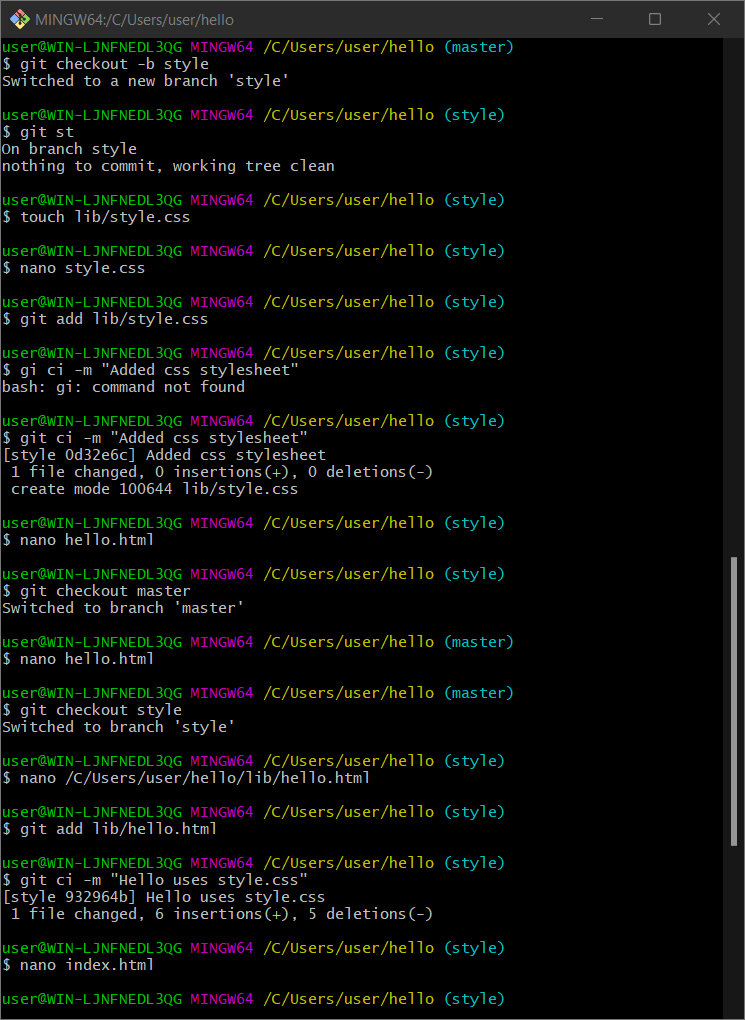


Рисунок 28 – Создание ветки, добавление файла стиля

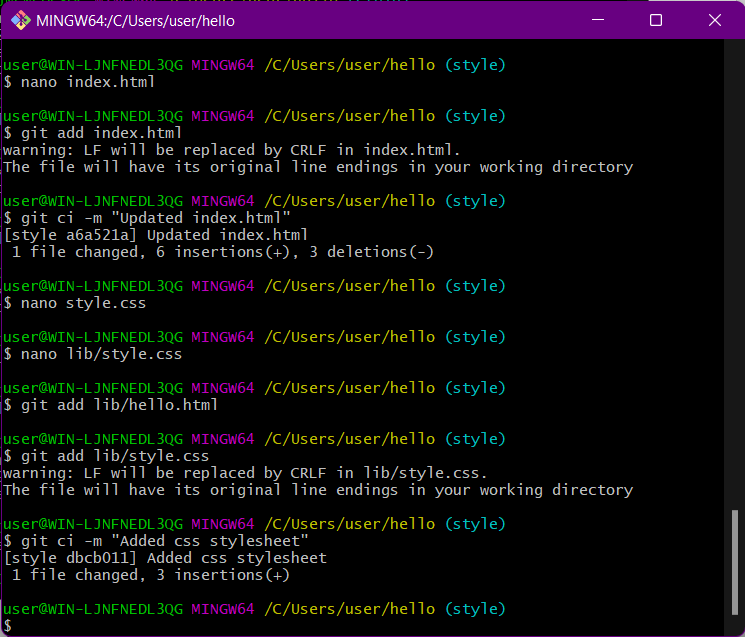


Рисунок 29 - Создание ветки, добавление файла стиля

Создать файл README в ветке master и сделать коммит изменений README.

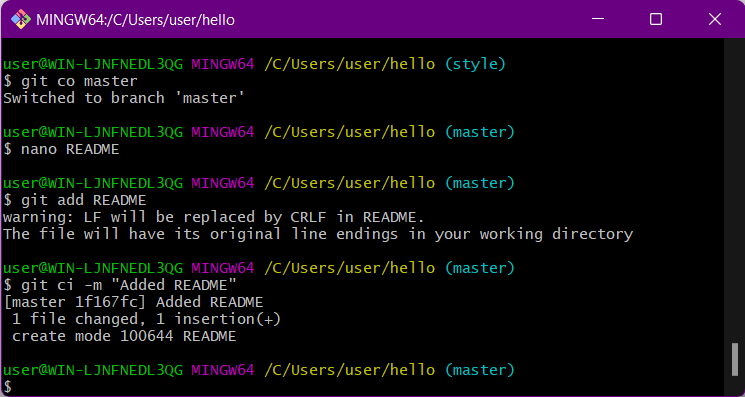


Рисунок 30 – Создание файла README

Слияние переносит изменения из двух веток в одну. Вернуться к ветке style и слить master с style.

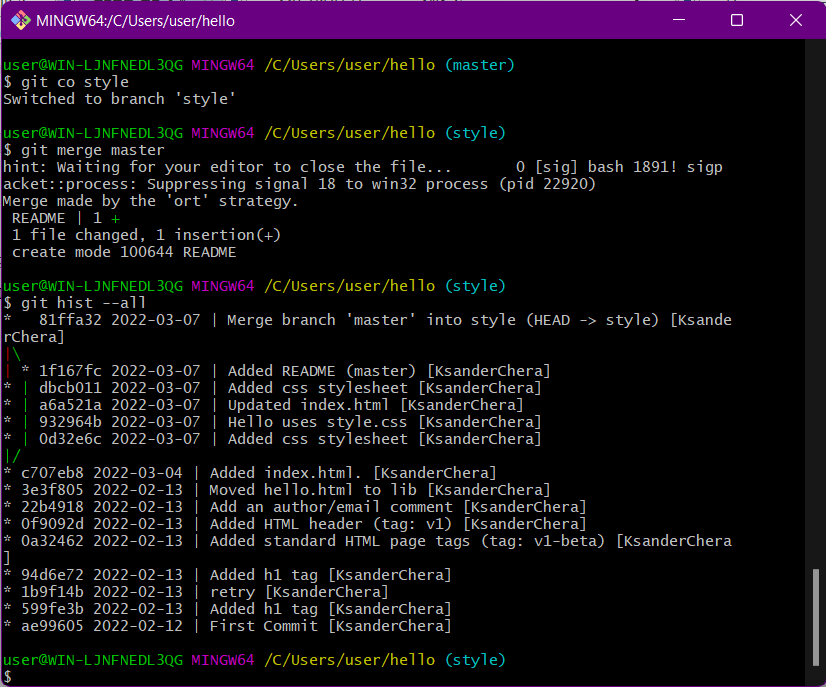


Рисунок 31 – Слияние веток

Вернуться в master и создать конфликт.

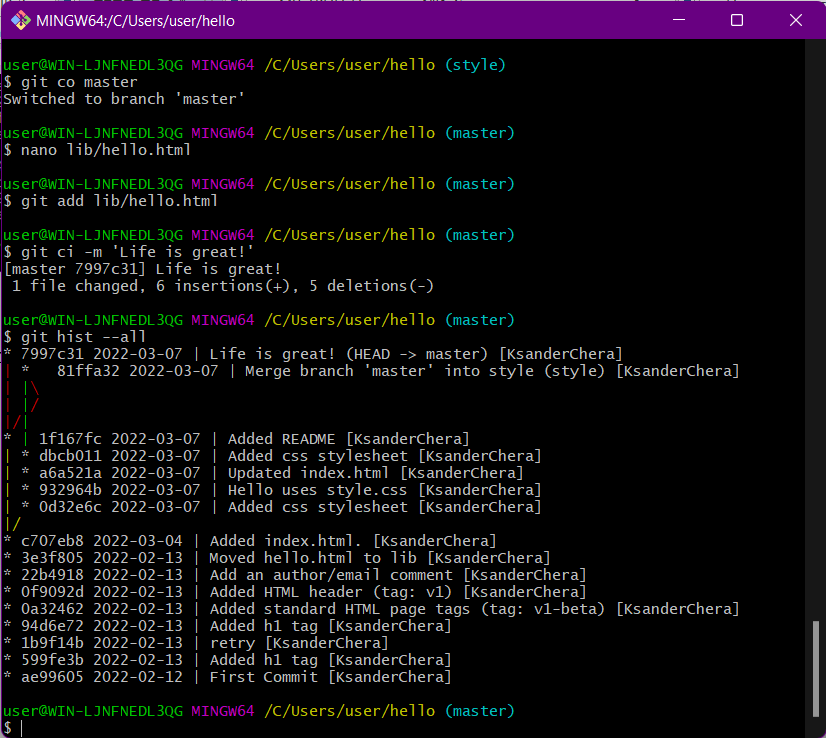


Рисунок 32 – Создание конфликта

Разрешение конфликтов.

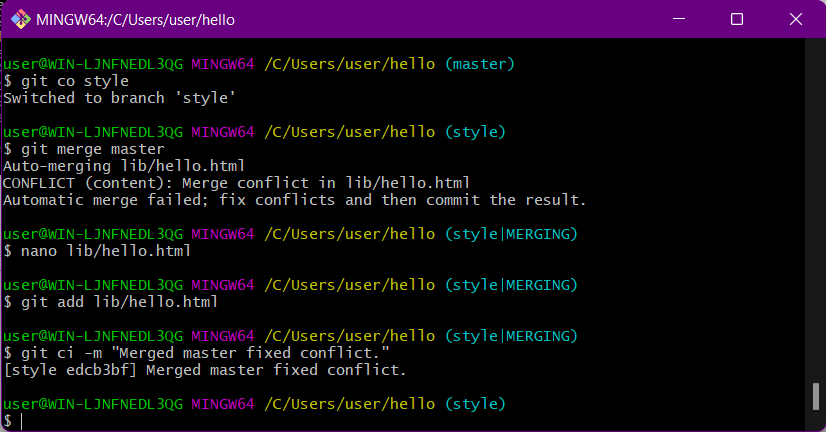


Рисунок 33 – Разрешение конфликта вручную

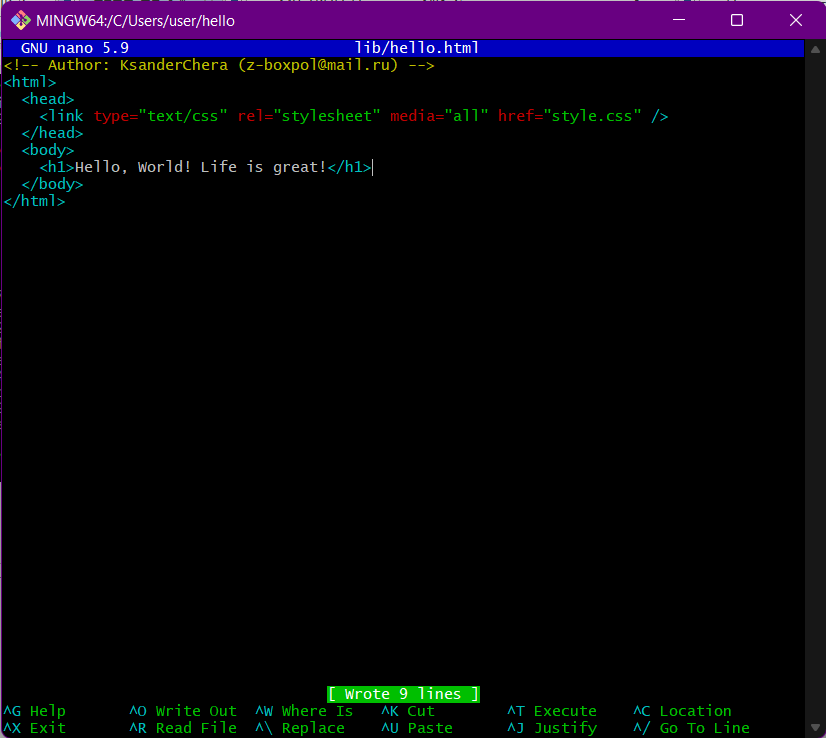


Рисунок 34 – Разрешение конфликта вручную

# **Часть 3. Системы управления репозиториями**

Создать репозиторий на GitHub.