Лабораторная работа № 1

Математическое моделирование

Королёв Иван Андреевич

Содержание

Список иллюстраций

Список таблиц

# 1 Цель работы

Вспомнить и закрепить навыки работы с git и markdown.

# 2 Задание

## 2.1 git

1. Параметры установки окончаний строк
2. Установка отображения unicode
3. Создание проекта
4. Внесение изменений
5. Индексация изменений
6. Отмена локальных изменений (до индексации)
7. Отмена проиндексированных изменений (перед коммитом)
8. Отмена коммитов
9. Удаление коммиттов из ветки
10. Удаление тега oops
11. Внесение изменений в коммиты
12. Перемещение файлов
13. Второй способ перемещения файлов
14. Подробнее о структуре
15. Git внутри: Каталог .git
16. Работа непосредственно с объектами git
17. Создание ветки
18. Навигация по веткам
19. Изменения в ветке master
20. Сделайте коммит изменений README.md в ветку master
21. Слияние
22. Создание конфликта
23. Разрешение конфликтов
24. Сброс ветки style
25. Сброс ветки master
26. Перебазирование
27. Слияние в ветку master
28. Клонирование репозиториев
29. Просмотр клонированного репозитория
30. Что такое origin?
31. Удаленные ветки
32. Изменение оригинального репозитория
33. Слияние извлеченных изменений
34. Чистые репозитории
35. Создайте чистый репозиторий
36. Добавление удаленного репозитория
37. Отправка изменений
38. Извлечение общих изменений

## 2.2 Markdown

1. Сделайте отчёт по лабораторной работе в формате Markdown

# 3 Теоретическое введение

Markdown — облегчённый язык разметки, созданный с целью обозначения форматирования в простом тексте, с максимальным сохранением его читаемости человеком, и пригодный для машинного преобразования в языки для продвинутых публикаций.

Git — это инструмент DevOps, используемый для управления исходным кодом . Это бесплатная система контроля версий с открытым исходным кодом, которая позволяет эффективно управлять как небольшими, так и очень большими проектами. Git используется для отслеживания изменений в исходном коде, позволяя нескольким разработчикам работать вместе над нелинейной разработкой.

# 4 Выполнение лабораторной работы

# 5 Параметры установки окончаний строк

Установка имени и электронной почты. Настройка core.autocrlf с параметрами true и input делает все переводы строк текстовых файлов в главном репозитории одинаковы. (рис. 1), (рис. 2)

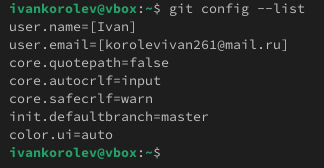


Рис. 1: Установка имени и электронной почты

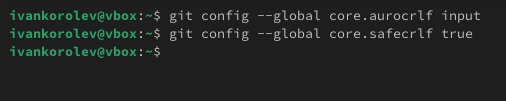


Рис. 2: переводы строк текстовых файлов

# 6 Установка отображения unicode

Установка отображения unicode (рис. 3)

Рис. 3: unicode

Рис. 3: unicode

# 7 Создание проекта

Начинаем работу в пустом рабочем каталоге с создания пустого каталога с именем hello, затем входим в него и создаём там файл с именем hello.html. Создаём репозиторий, добавляем файл в репозиторий и проверяем состояние репозитория. (рис. 4), (рис. 5)

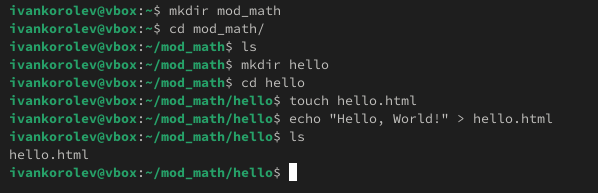


Рис. 4: Создаём страницу «Hello, World»

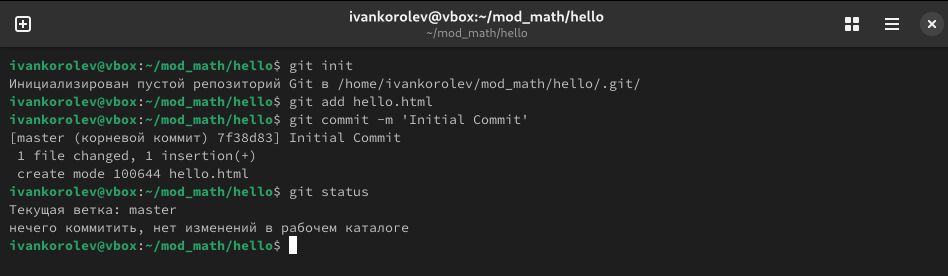


Рис. 5: Репозиторий

# 8 Внесение изменений

Добавим HTML-теги к нашему приветствию и проверим состояние рабочего каталога. (рис. 6), (рис. 7)

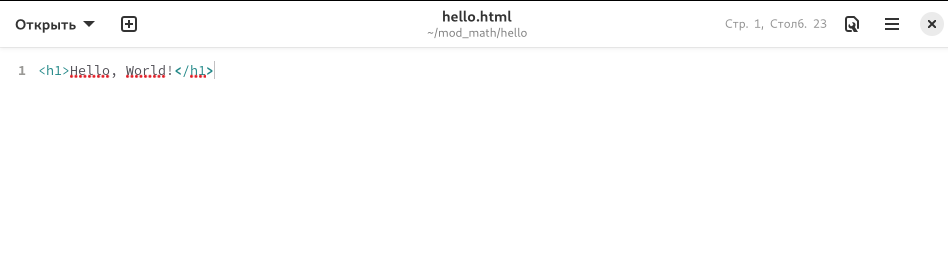


Рис. 6: HTML-теги

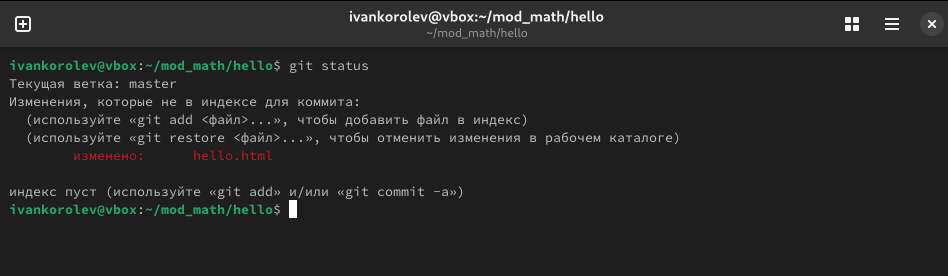


Рис. 7: состояние рабочего каталога

# 9 Индексация изменений

Проиндексируем изменения и проверим состояния. Коммиты изменений. Добавляем стандартные теги html и body, добавляем изменение в индекс git и после, добавляем секцию head. Первое изменение (добавление стандартных тегов) проиндексировано и готово к коммиту. Второе изменение (добавление заголовков HTML) является непроиндексированным. Производим коммит проиндексированного изменения (значение по умолчанию), а затем еще раз проверяем состояние. Теперь добавляем второе изменение в индекс, а затем проверяем состояние с помощью команды git status и делаем коммит второго изменения.(рис. 8), (рис. 9), (рис. 10), (рис. 11), (рис. 12), (рис. 13)

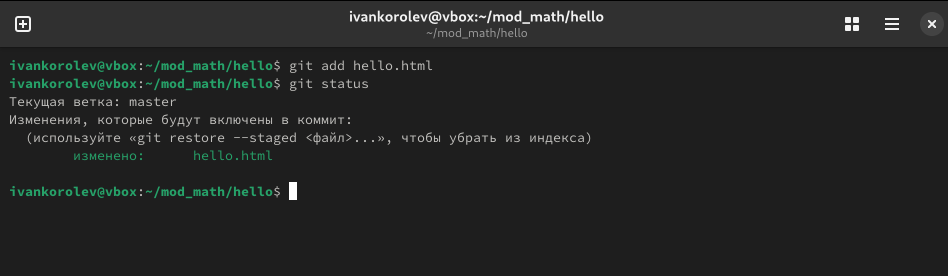


Рис. 8: Проиндексируем изменения и проверим состояния

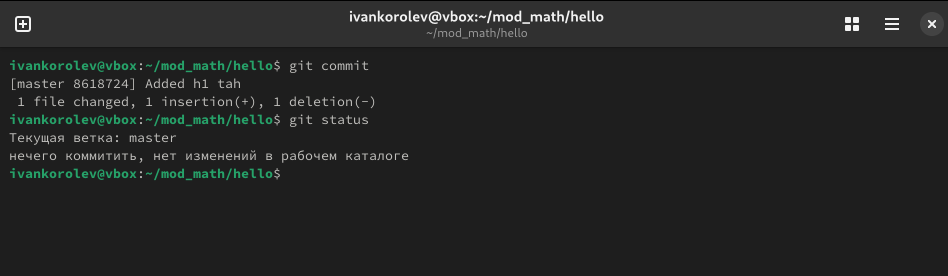


Рис. 9: Коммиты изменений

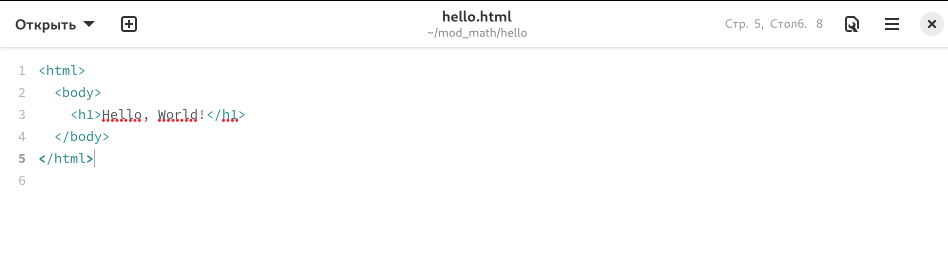


Рис. 10: HTML-теги

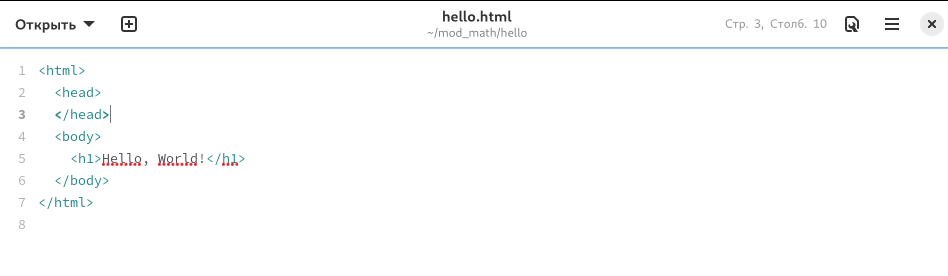


Рис. 11: head

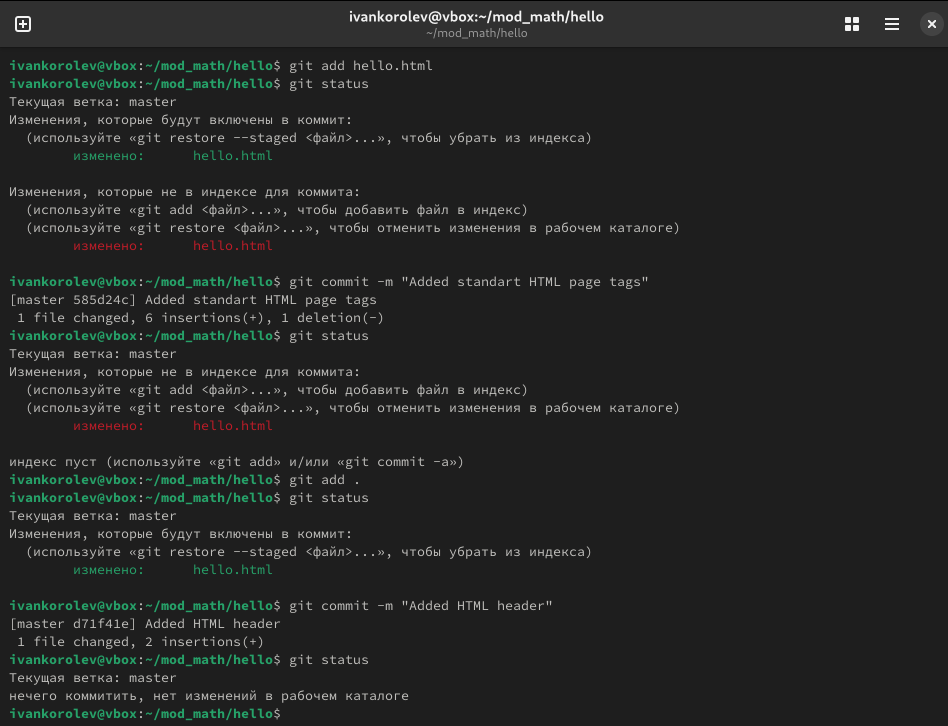


Рис. 12: HTML-теги

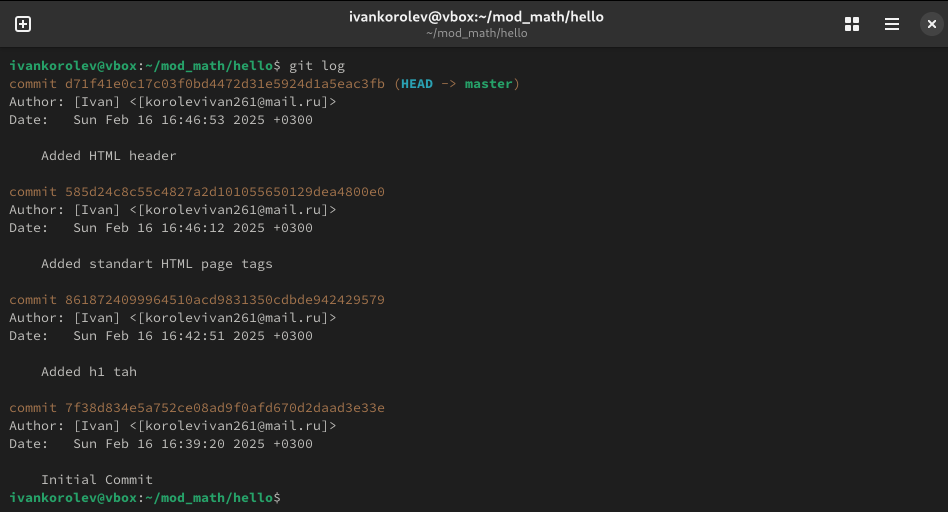


Рис. 13: Список произведенных изменений

# 10 Отмена локальных изменений (до индексации)

Вносим изменение в файл hello.html в виде нежелательного комментария. Проверяем состояние рабочего каталога и отменяем изменения в рабочем каталоге. (рис. 14), (рис. 15)



Рис. 14: Вносим изменение в файл hello.html

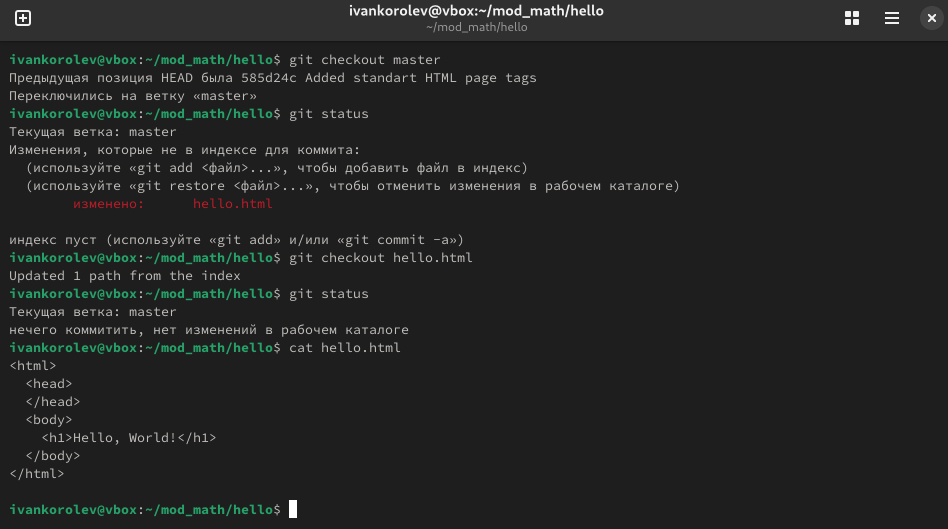


Рис. 15: отменяем изменения

# 11 Отмена проиндексированных изменений (перед коммитом)

Вносим изменение в файл hello.html в виде нежелательного комментария и проиндексируем изменения. Проверяем состояние рабочего каталога. Выполняем сброс буферной зоны. (рис. 16), (рис. 17)

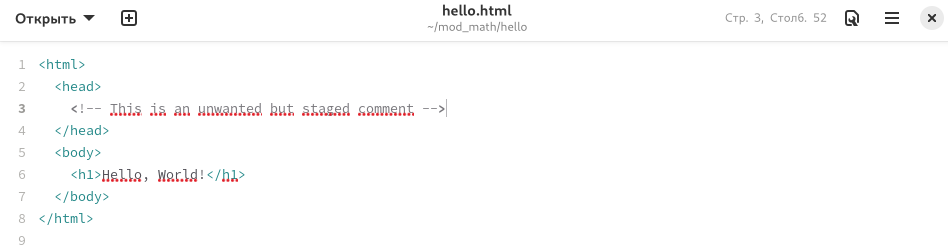


Рис. 16: Вносим изменение в файл hello.html

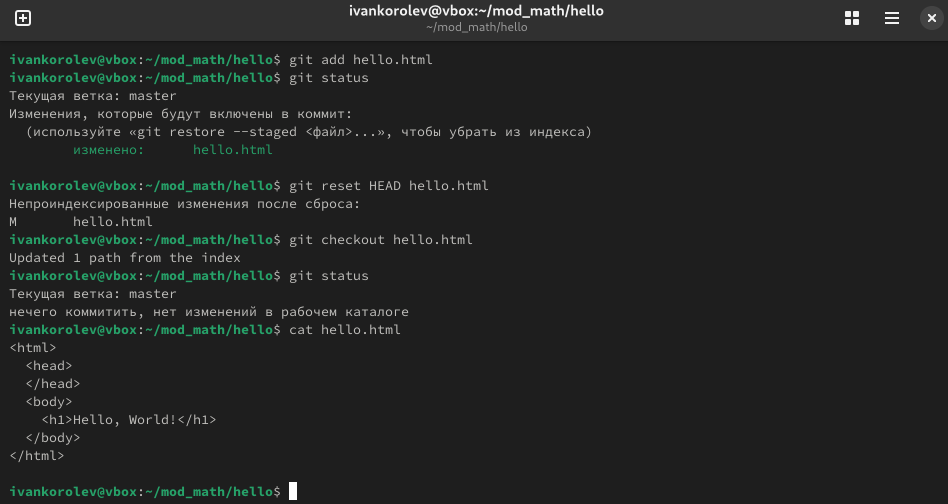


Рис. 17: отменяем изменения

# 12 Отмена коммитов

Вносим изменение в файл hello.html в виде нежелательного комментария. Проверяем состояние рабочего каталога. Выполняем сброс буферной зоны. (рис. 18), (рис. 19)



Рис. 18: Вносим изменение в файл hello.html

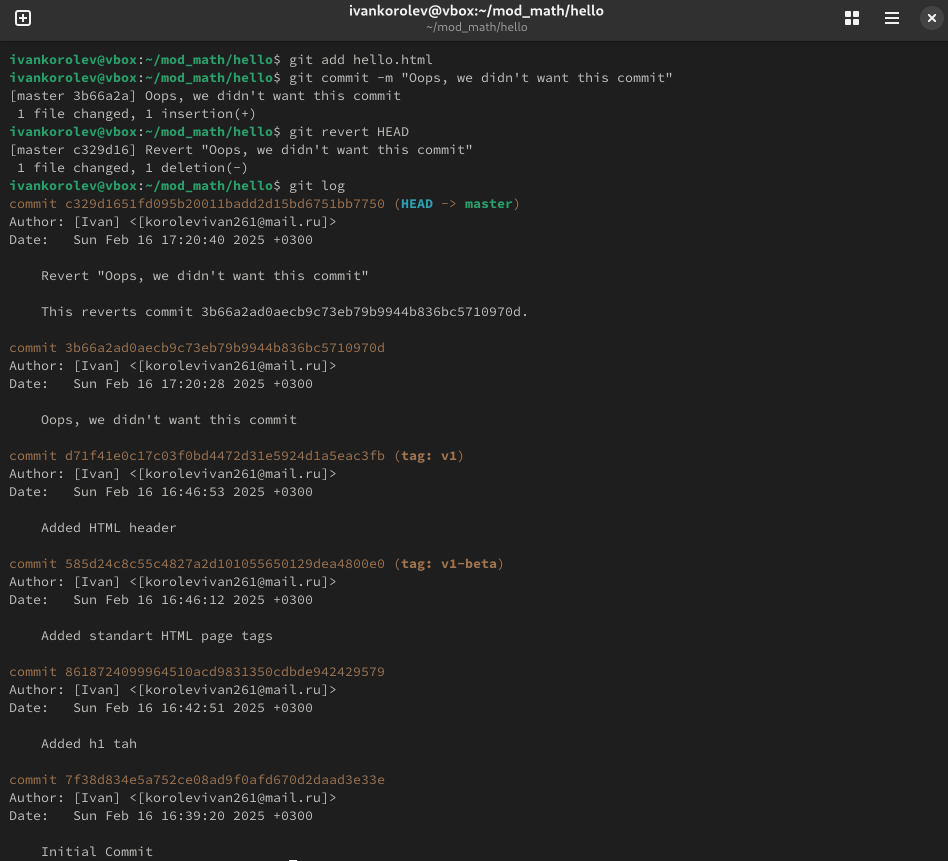


Рис. 19: отменяем изменения

# 13 Удаление коммиттов из ветки

Команда git reset.Для начала отмечаем эту ветку и удаляем коммит из ветки. (рис. 20)

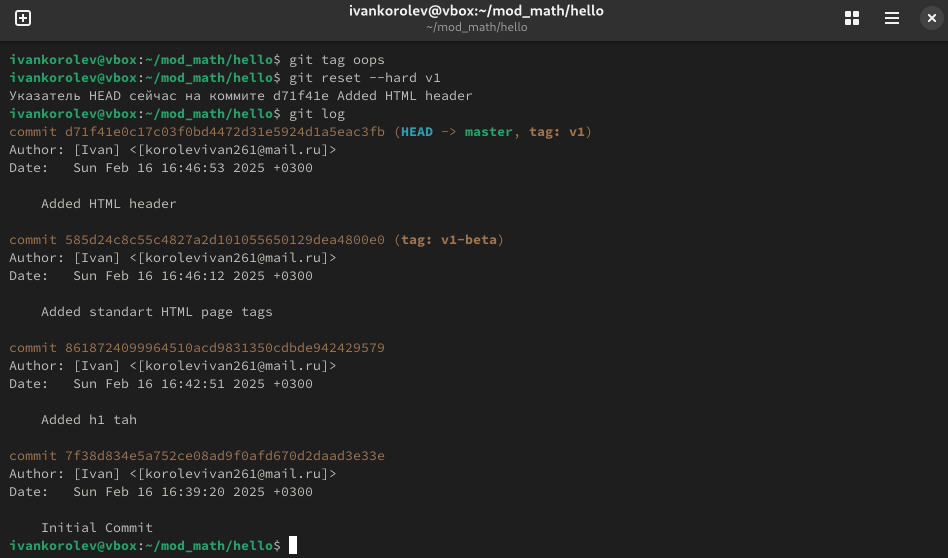


Рис. 20: git reset

# 14 Удаление тега oops

Тег oops свою функцию выполнил. Давайте удалим его и коммиты, на которые он ссылался, сборщиком мусора.(рис. 21)

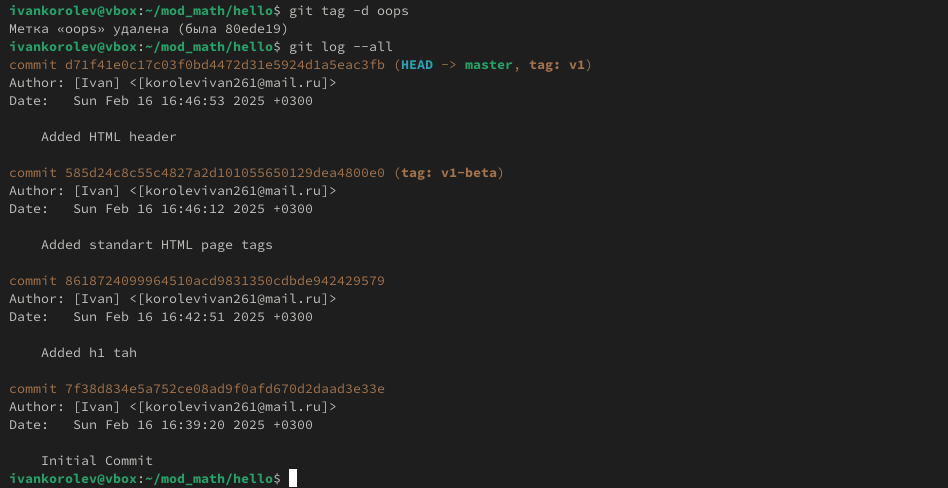


Рис. 21: git tag -d oops

# 15 Перемещение файлов

Переместите файл hello.html в каталог lib и Коммит в новый каталог.(рис. 22)

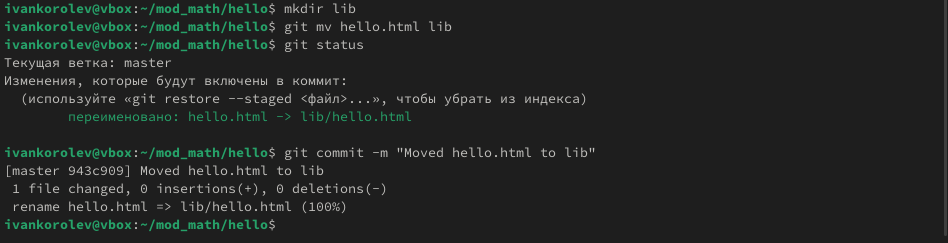


Рис. 22: git mv

# 16 Подробнее о структуре

Добавление index.html (рис. 23)



Рис. 23: index.html

# 17 Git внутри: Каталог .git

Git внутри: Каталог .git (рис. 24)

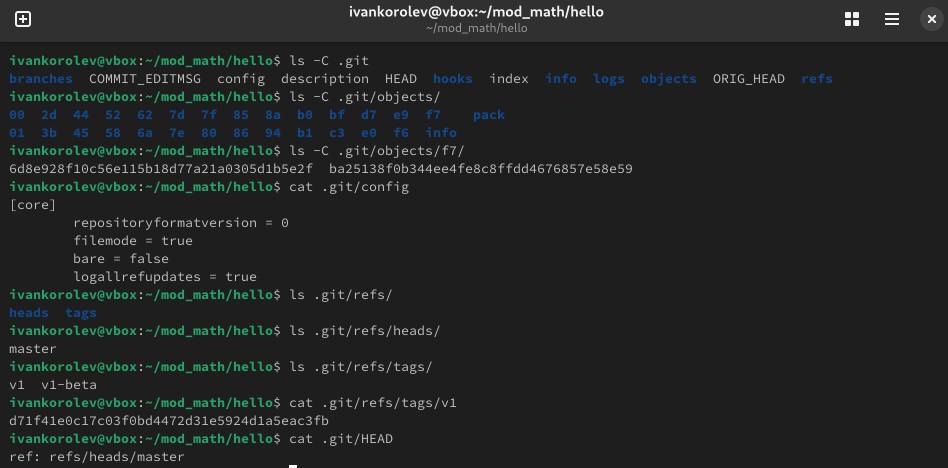


Рис. 24: .git

# 18 Работа непосредственно с объектами git

Поиск последнего коммита. Исследуйте git репозиторий вручную самостоятельно. Смотрите, удастся ли вам найти оригинальный файл hello.html с самого первого коммита вручную по ссылкам SHA1 хэша в последнем коммите.(рис. 25), (рис. 26)

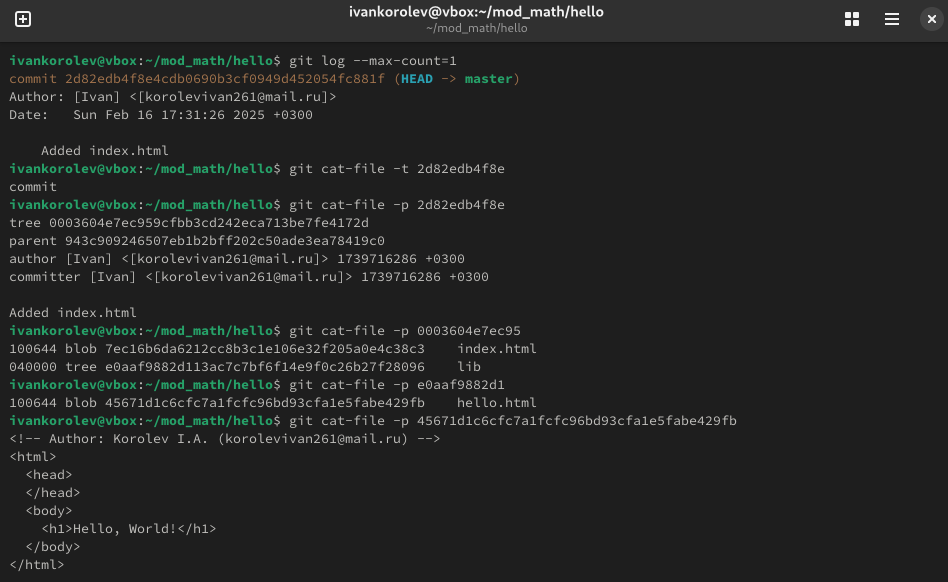


Рис. 25: Поиск последнего коммита

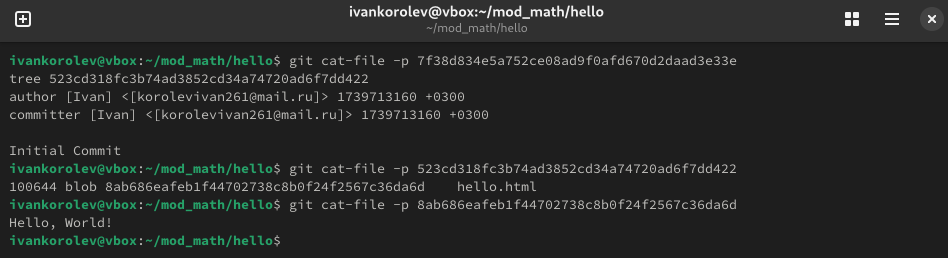


Рис. 26: Задание

# 19 Создание ветки

Пора сделать наш hello world более выразительным. Так как это может занять некоторое время, лучше переместить эти изменения в отдельную ветку, чтобы изолировать их от изменений в ветке master.(рис. 27), (рис. 28), (рис. 29)



Рис. 27: Измените основную страницу



Рис. 28: Измените index.html

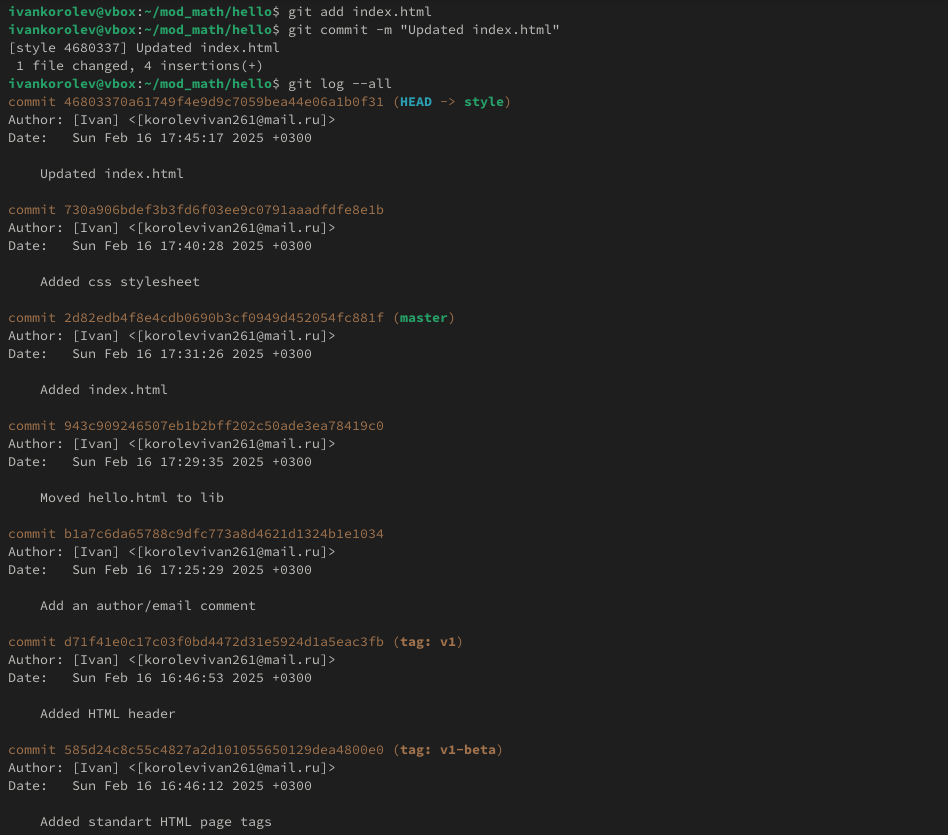


Рис. 29: Переключение на ветку master

# 20 Изменения в ветке master

Создайте файл README в ветке master. Сделайте коммит изменений README.md в ветку master.(рис. 30)

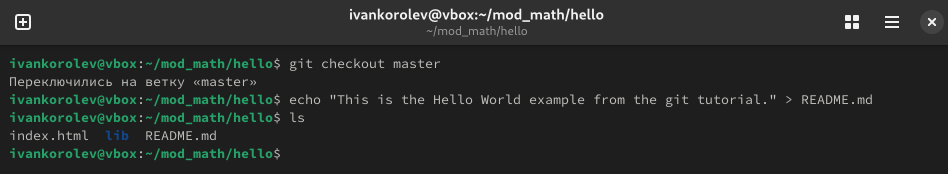


Рис. 30: Изменения в ветке master

# 21 Создание конфликта. Разрешение конфликтов.

Создание конфликта. Разрешение конфликтов (рис. 31), (рис. 32), (рис. 33), (рис. 34)

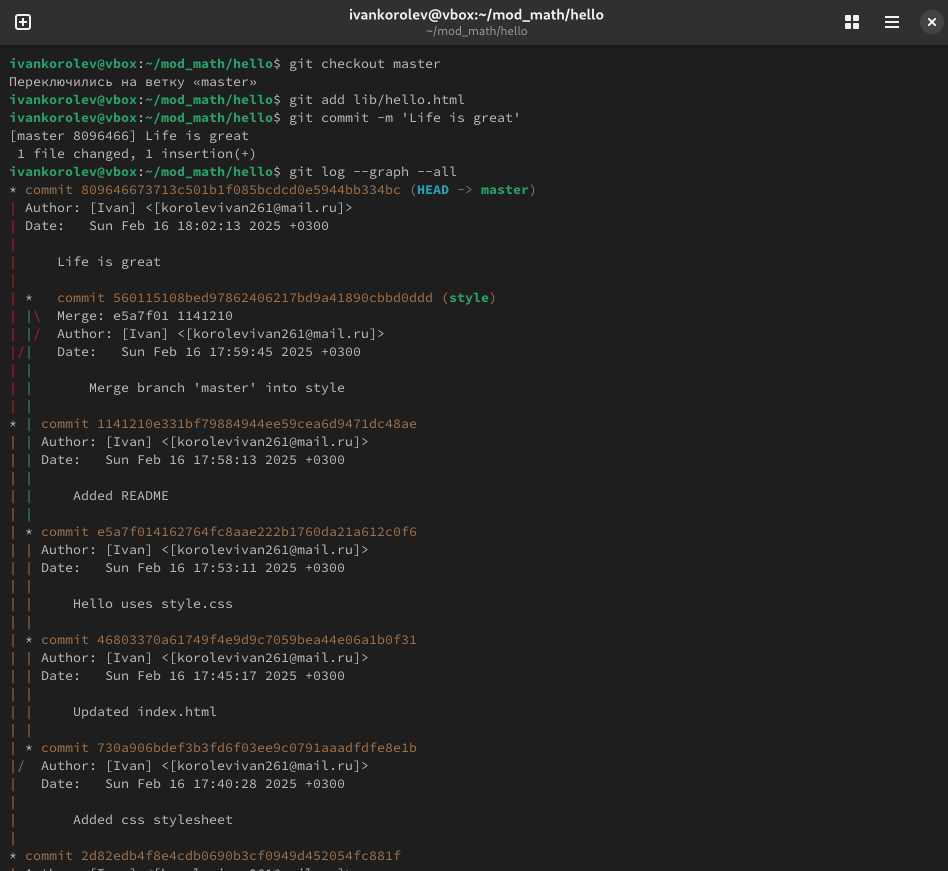


Рис. 31: Создание конфликта

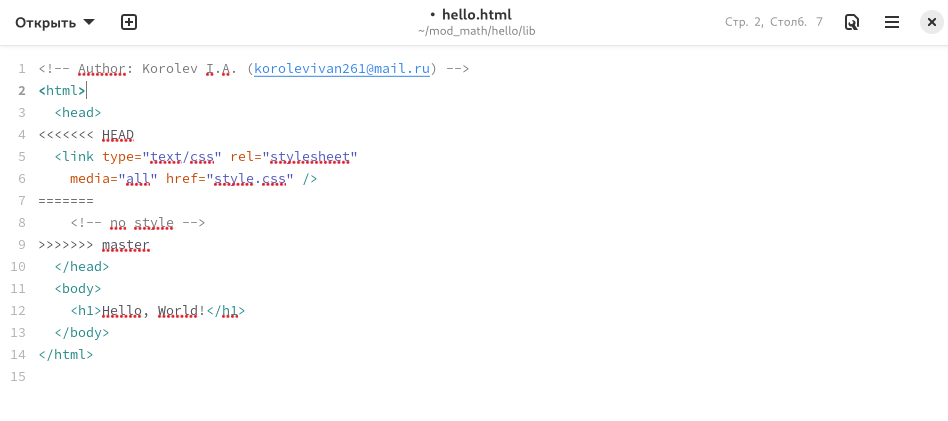


Рис. 32: Создание конфликта

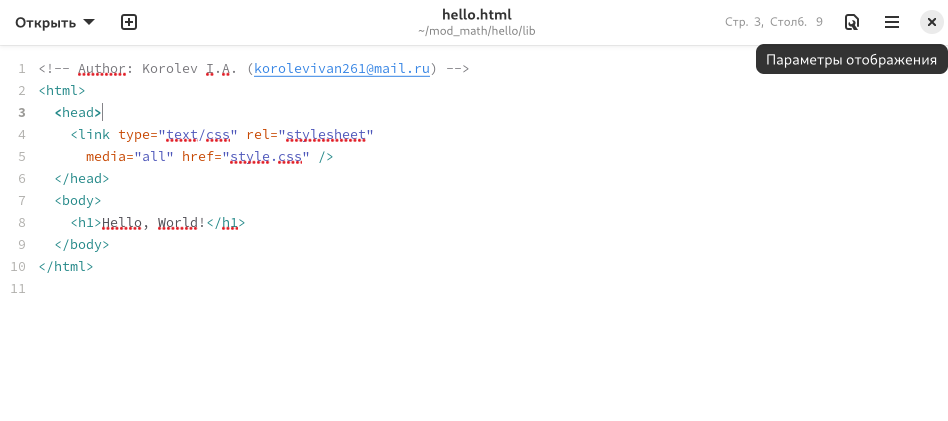


Рис. 33: Разрешение конфликтов

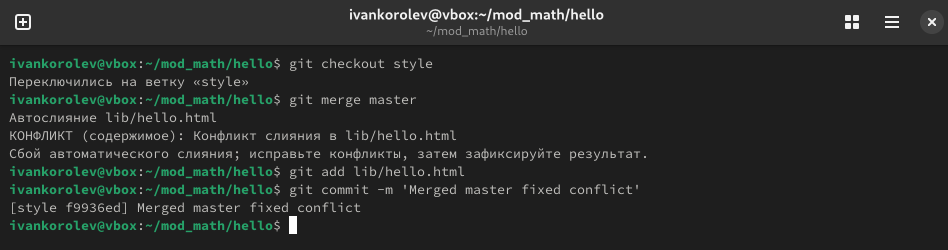


Рис. 34: Разрешение конфликтов

# 22 Сброс ветки style

Сброс ветки style. (рис. 35)

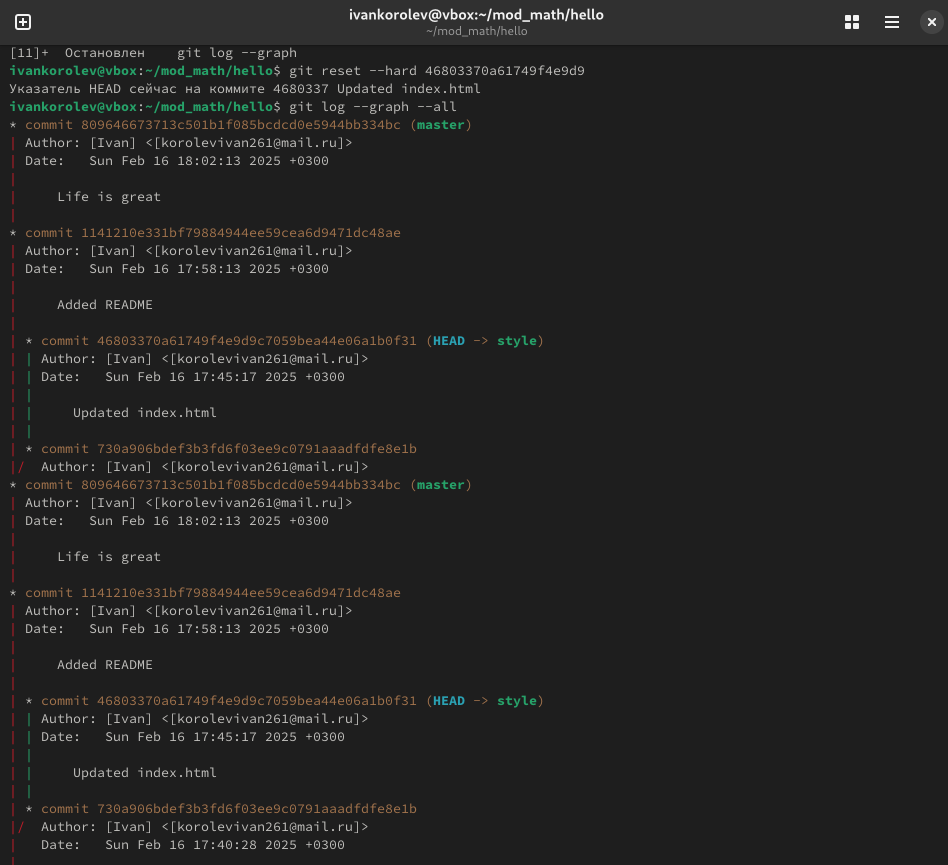


Рис. 35: Сброс ветки style

# 23 Перебазирование

Перебазирование. (рис. 36)

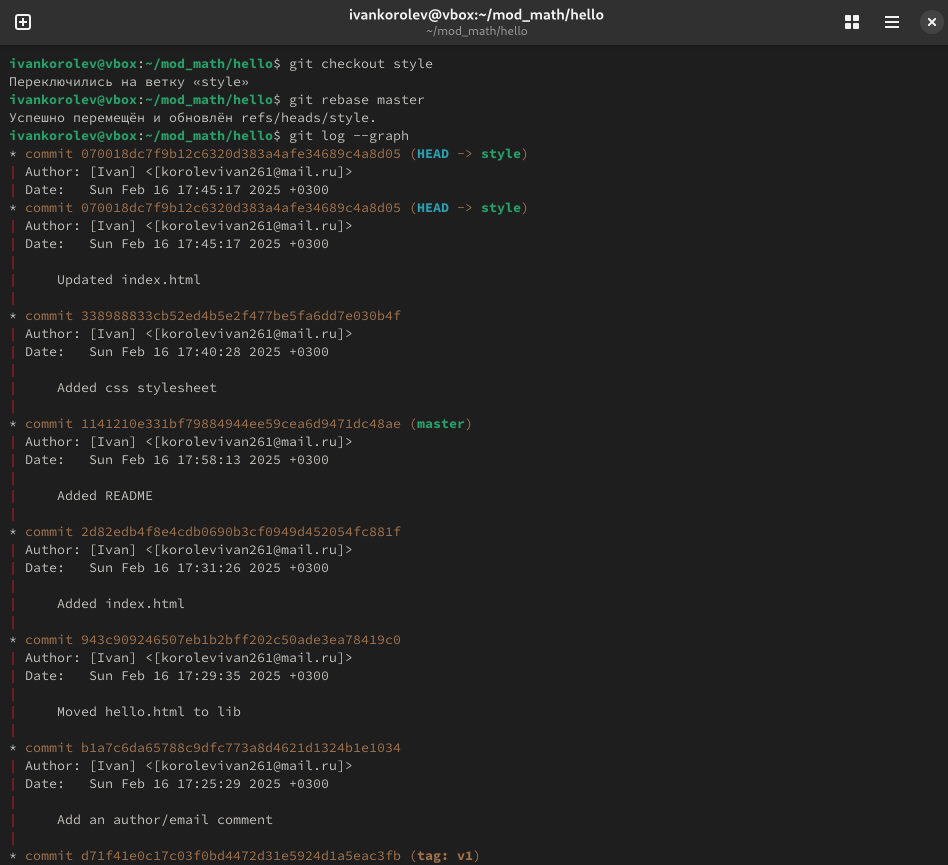


Рис. 36: Перебазирование

# 24 Клонирование репозиториев/usr/local/texlive/2024/bin/x86\_64-linux/

Клонирование репозиториев (рис. 37)

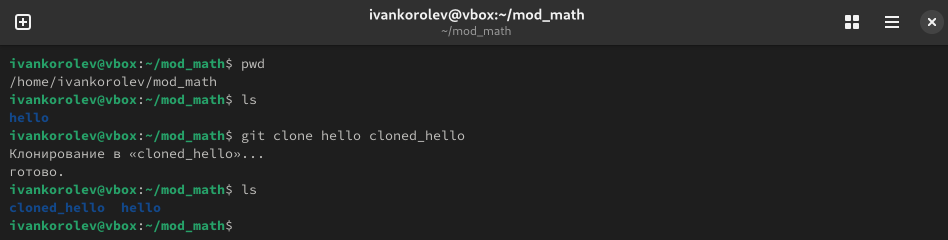


Рис. 37: Клонирование репозиториев

# 25 Что такое origin?

Удаленные репозитории обычно размещаются на отдельной машине, возможно, централизованном сервере. Однако, как мы видим здесь, они могут с тем же успехом указывать на репозиторий на той же машине. Нет ничего особенного в имени «origin», однако существует традиция использовать «origin» в качестве имени первичного централизованного репозитория (если таковой имеется). (рис. 38)

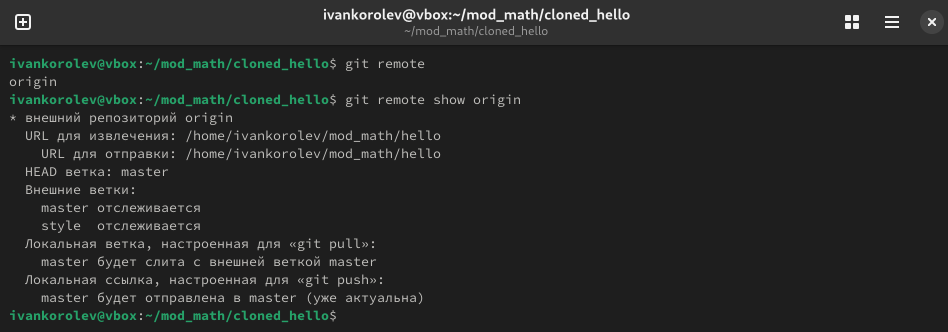


Рис. 38: origin

# 26 Изменение оригинального репозитория

Внесите некоторые изменения в оригинальный репозиторий, чтобы затем попытаться извлечь и слить изменения из удаленной ветки в текущую. (рис. 39), (рис. 40), (рис. 41), (рис. 42)

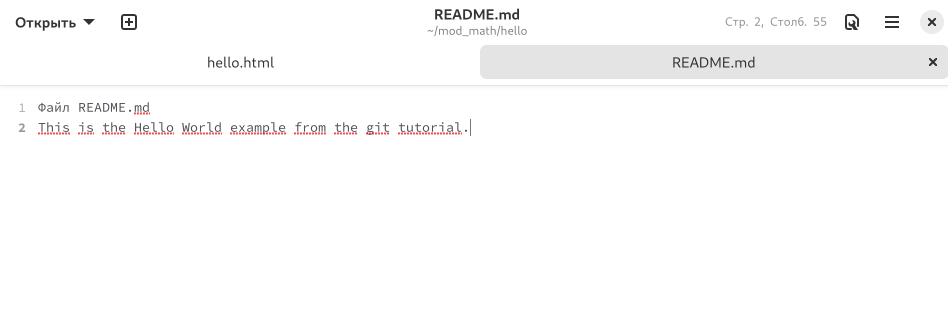


Рис. 39: Изменение оригинального репозитория

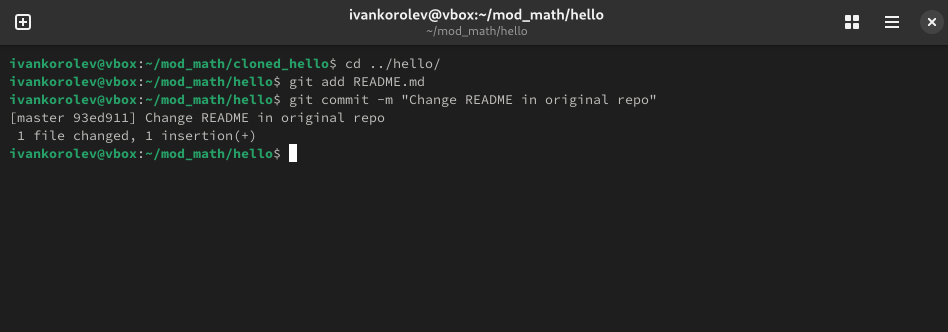


Рис. 40: Изменение оригинального репозитория

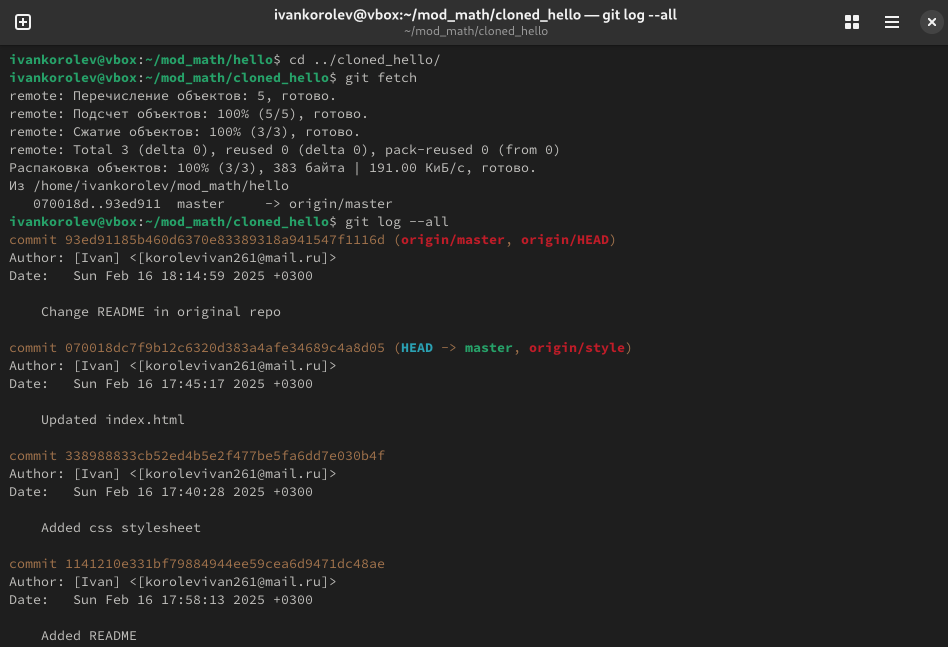


Рис. 41: Изменение оригинального репозитория

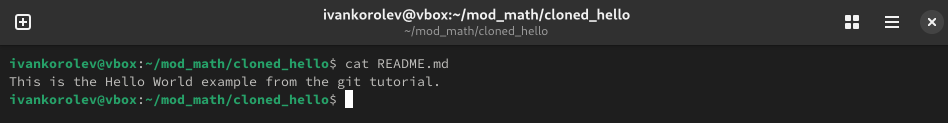


Рис. 42: Изменение оригинального репозитория

# 27 Слияние извлеченных изменений

Слияние извлеченных изменений (рис. 43)

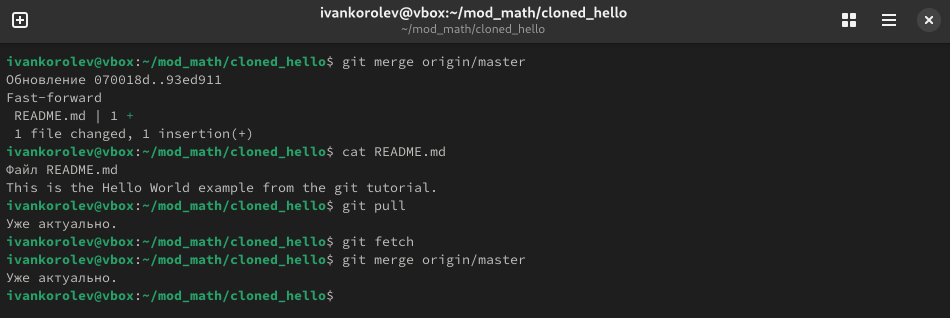


Рис. 43: Слияние извлеченных изменений

# 28 Отправка изменений

Отправка изменений (рис. 44)

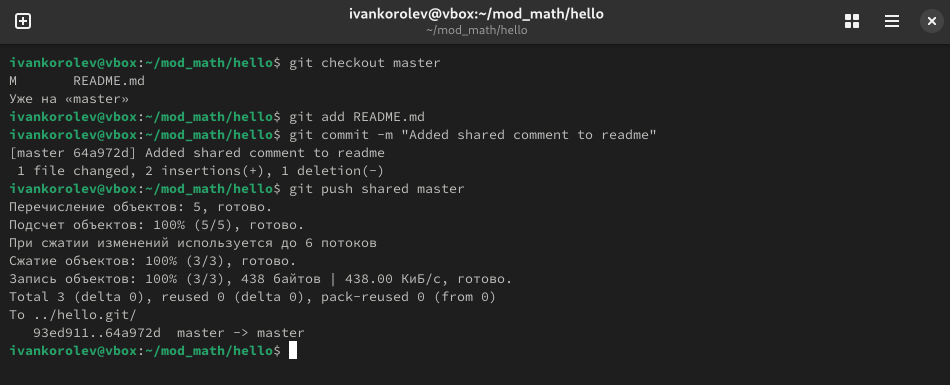


Рис. 44: Отправка изменений

# 29 Markdown

Markdown (рис. 45)

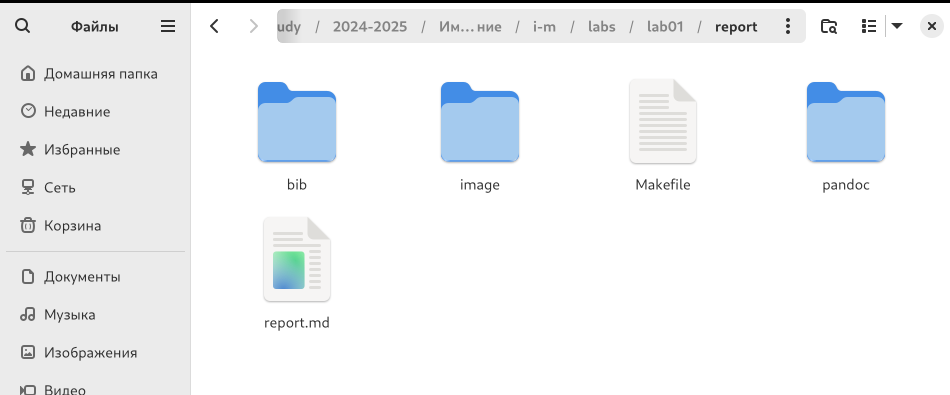


Рис. 45: Markdown

# 30 Выводы

Вспомнил и закрепил навыки работы с git и markdown.

# Список литературы