Лабораторная работа №1

Дисциплина: Математическое моделирование

Пронякова Ольга Максимовна

Содержание

# 1 Цель работы

Вспомнить умения работы с git

# 2 Выполнение лабораторной работы

Настройка core.autocrlf с параметрами true и input делает все переводы строк текстовых файлов в главном репозитории одинаковы. core.autocrlf true - git автоматически конвертирует CRLF->LF при комми- те и обратно LF->CRLF при выгрузке кода из репозитория на файловую систему (используют в Windows). core.autocrlf input - конвертация CRLF в LF только при коммитах (используют в Mac/Linux). Если core.safecrlf установлен в true или warm, git проверяет, если пре- образование является обратимым для текущей настройки core.autocrlf. core.safecrlf true - отвержение необратимого преобразования lf<->crlf. Полезно, когда специфические бинарники похожие на текстовые файлы. core.safecrlf warn - печать только предупреждение, но принимает необрати- мый переход.

В пустом рабочем каталоге создаю пустой каталога с именем hello, затем вхожу в него и создаю там файл с именем hello.html(рис.1).

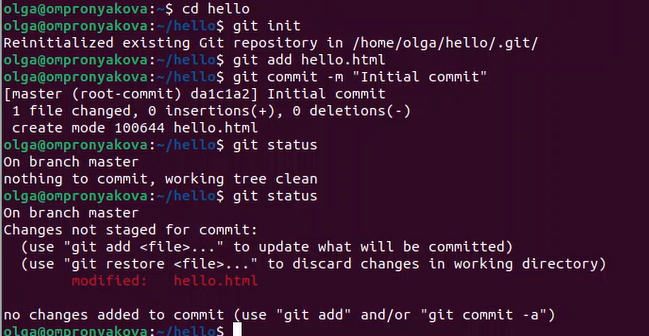


Рис. 1: Выполнение команд

Создаю git репозиторий из этого каталога, добавляю файл в реаозиторий и проверяю статус(рис.2).

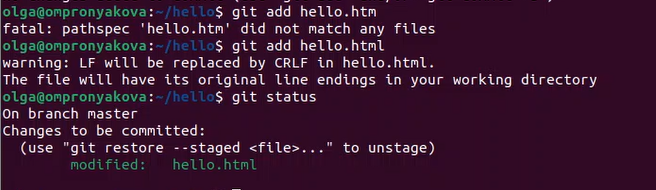


Рис. 2: выполнение команд

Изменяю содержимое файла hello.html и проверяю состояние рабочего каталога(рис.3).

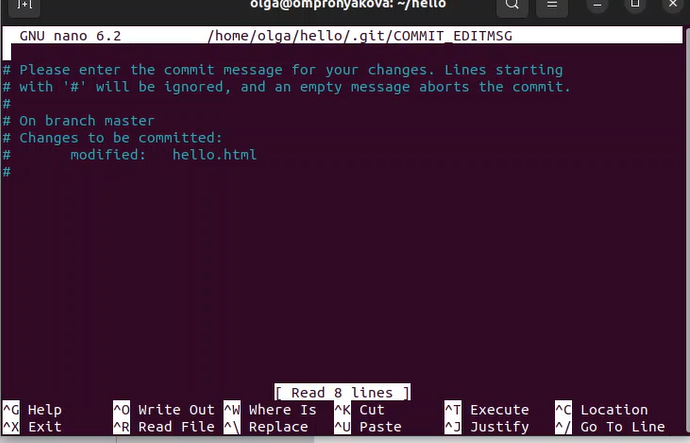


Рис. 3: Заполение файла

Выполняю команду git, чтобы проиндексировать изменения и роверяю состояние(рис.4).

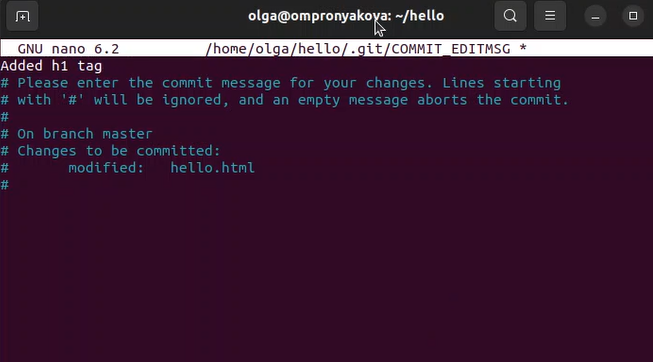


Рис. 4: Заполение файла

Делаю коммит. проверяю состояние. Изменяю страницу «Hello, World», чтобы она содержала стандартные теги

и

и добавляю это изменение в индекс git(рис.5).

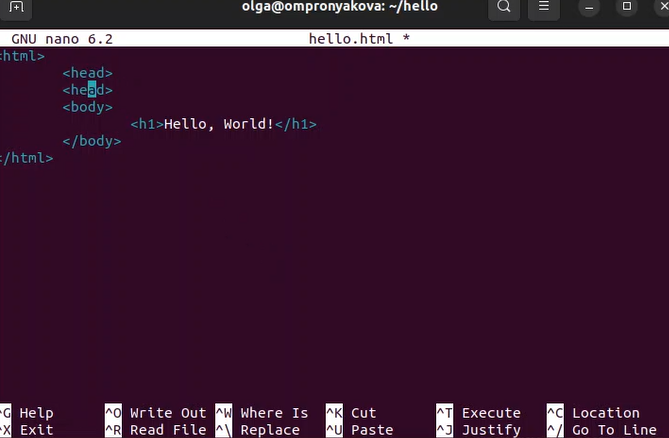


Рис. 5: Заполение файла

произвожу коммит проиндексированного изменения (значение по умолчанию), а затем еще раз проверяю состояние(рис.6).

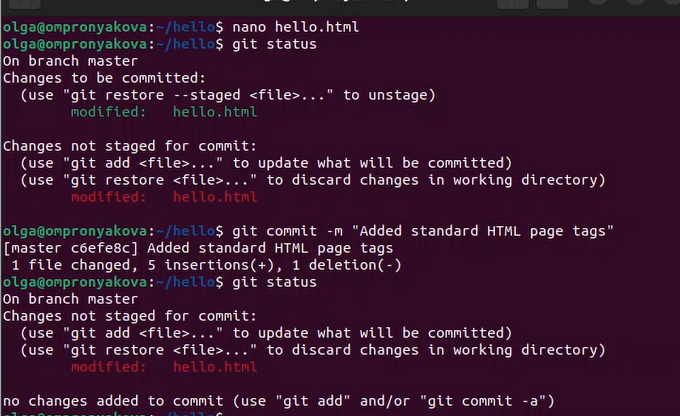


Рис. 6: Заполение файла

Получяю список произведенных изменений(рис.7).

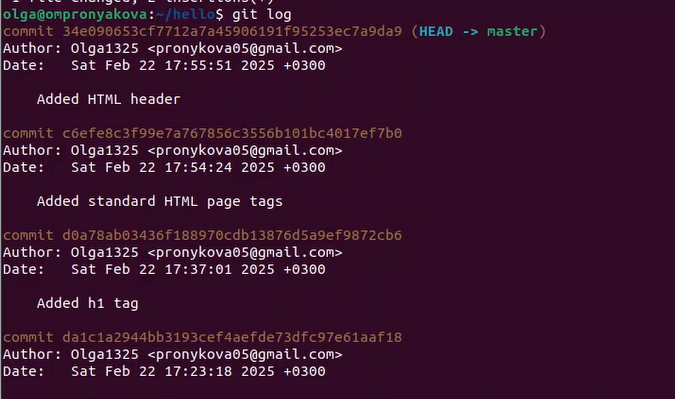


Рис. 7: Заполение файла

Применяю несколько вариантов отображения лога(рис.8).

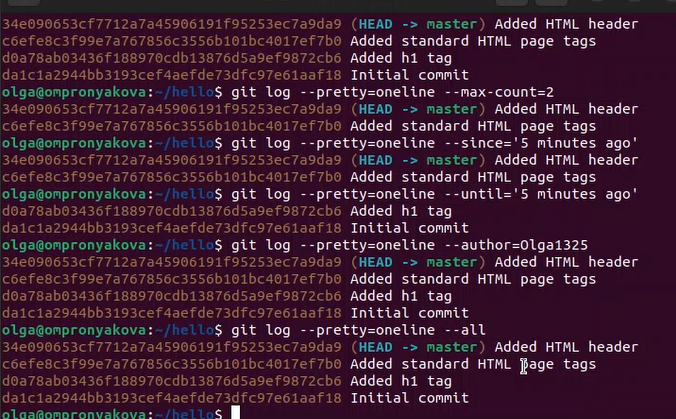


Рис. 8: результаты команд

Справочная страница (рис.9).

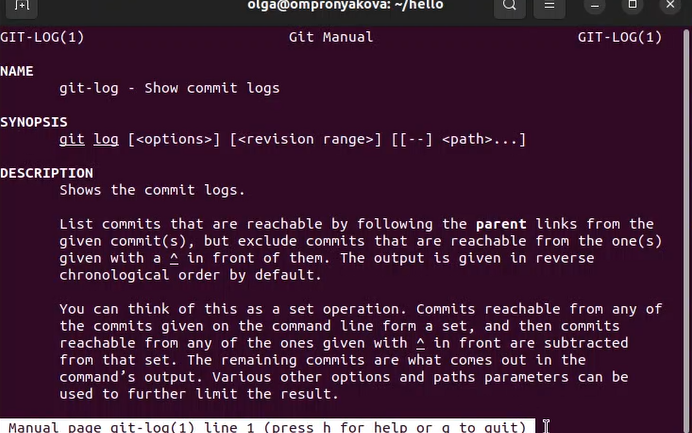


Рис. 9: Результат команды

Изучаю данные лога и найдите хэш для первого коммита. Он должен быть в последней строке данных. Использую этот хэш-код (достаточно первых 7 знаков) в команде ниже. Затем проверяю содержимое файла hello.html(рис.10).

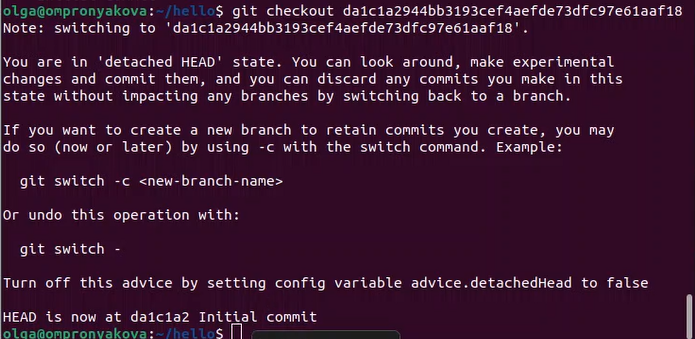


Рис. 10: Заполение файла

Теги для предыдущих версий Давайте создадим тег для версии, которая идет перед текущей версией и назовем его v1-beta. В первую очередь нам надо переключиться на предыдущую версию. Вместо поиска до хэш, мы будем использовать ^, обозначающее «родитель v1». Вместо обозначения v1^ можно использовать v1~1. Это обозначение можно определить как «первую версию предшествующую v1»(рис.11).

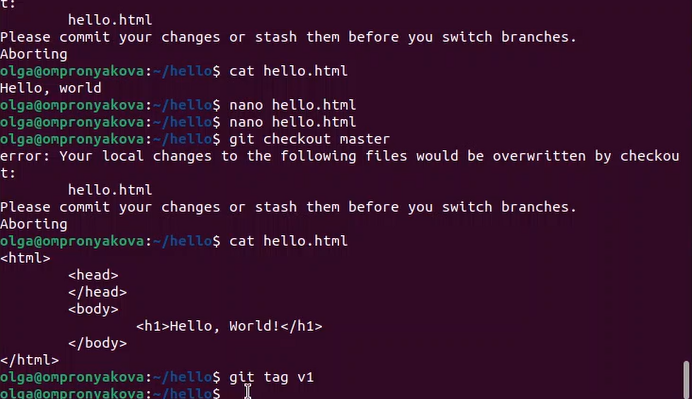


Рис. 11: результат команд

делаю ее версией v1-beta и пробую попереключаться между двумя отмеченными версиями(рис.12).

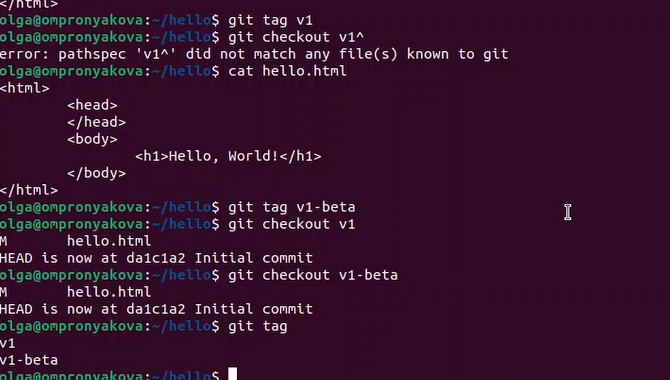


Рис. 12: Результат команд

Смотрю теги(рис.13).

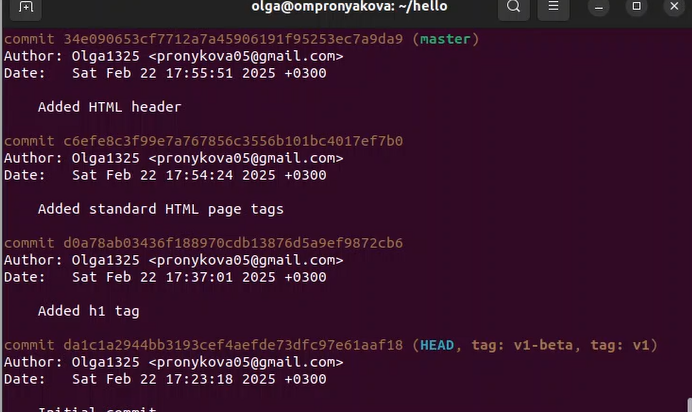


Рис. 13: Результат комманды

Вношу изменение в файл hello.html в виде нежелательного комментария(рис.14).

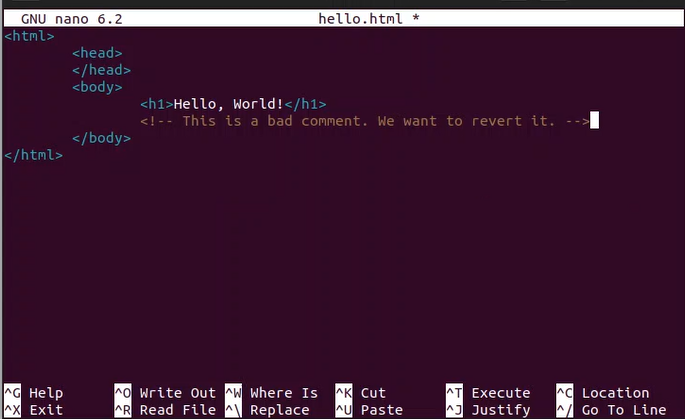


Рис. 14: Изменение файла

Использую команду git checkout для переключения версии файла hello.html в репозитории(рис.15).

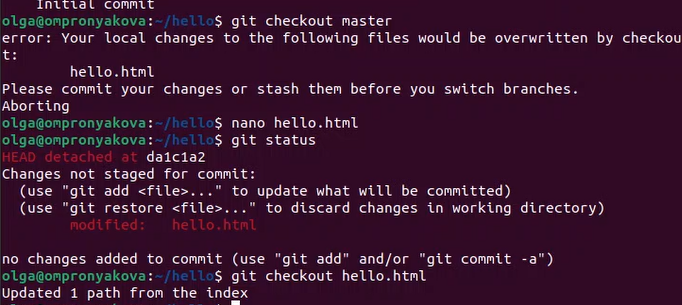


Рис. 15: Результат комманды

Вношу изменение в файл hello.html в виде нежелательного комментария(рис.16).

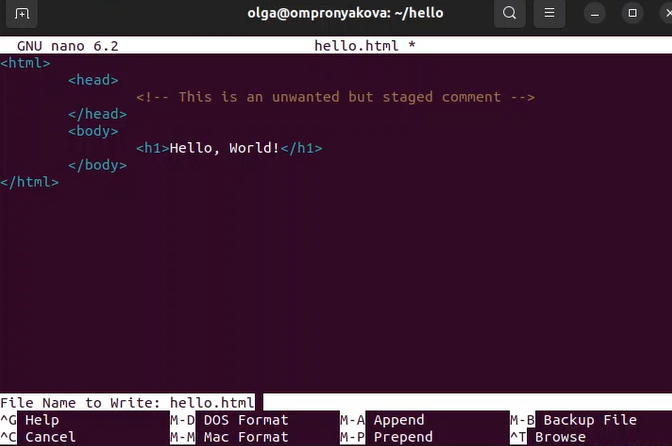


Рис. 16: Изменение файла

Проиндексирую это изменение и проверяю состояние нежелательного изменения(рис.17).



Рис. 17: результат команды

делаю отмену индексации изменений и проверяю статус(рис.18).

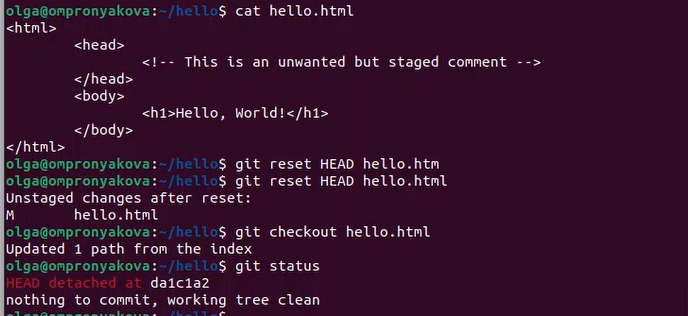


Рис. 18: Результат команды

Изменяю файл hello.html на следующий(рис.19).

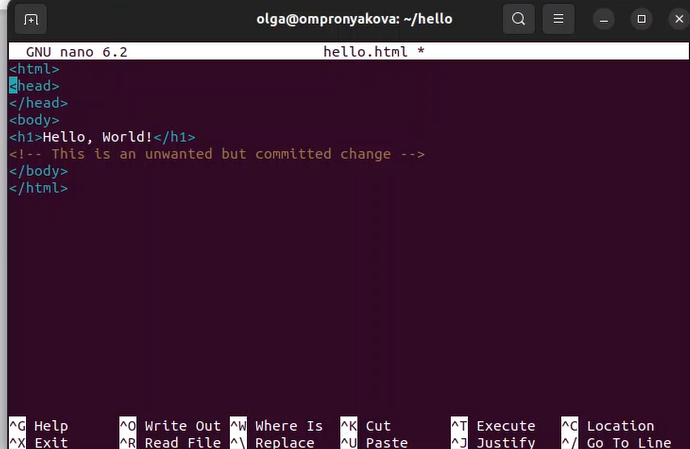


Рис. 19: Изменение файла

Проверяю лог на нежелательные и отмененные коммиты в мой репозиторий(рис.20).

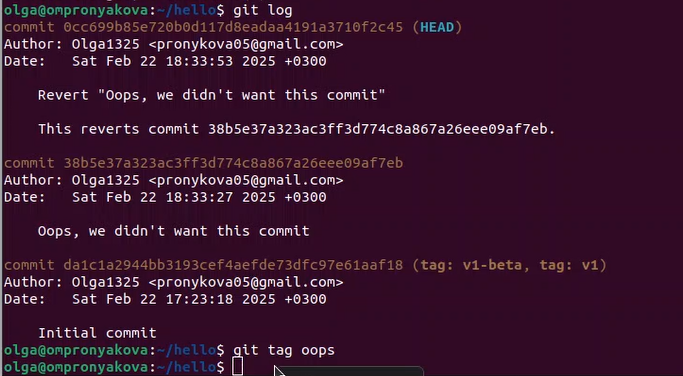


Рис. 20: Результат команд

Но прежде чем удалить коммиты, давайте отметим последний коммит тегом, чтобы потом можно было его найти. сбросим ветку до этой точки. Поскольку ветка имеет тег, мы можем использовать имя тега в команде сброса. Тег oops свою функцию выполнил. Удаляю его и коммиты, на которые он ссылался, сборщиком мусора(рис.21).

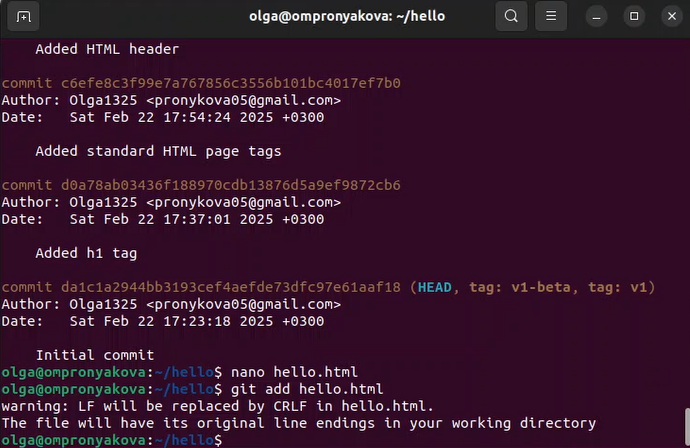


Рис. 21: Результат команд

Обновляю страницу hello, включив в нее email(рис.22).

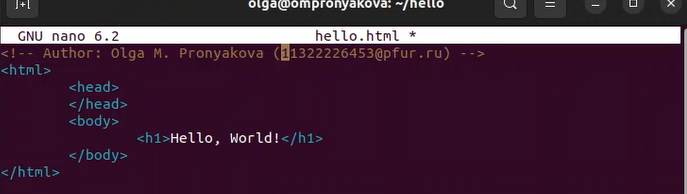


Рис. 22: Изменение файла

Действительно не хотим создавать отдельный коммит только ради электронной почты. Давайте изменим предыдущий коммит, включив в него адрес электронной почты(рис.23).

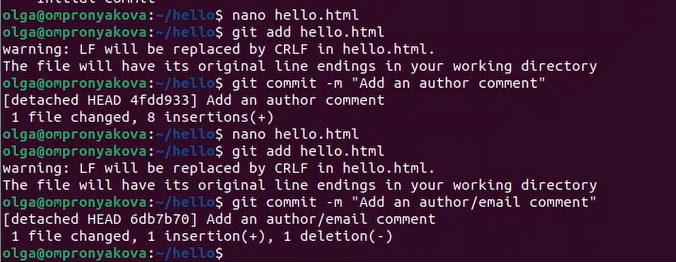


Рис. 23: Результат команд

Создаю структуру нашего репозитория. Давайте перенесем страницу в каталог lib(рис.24).

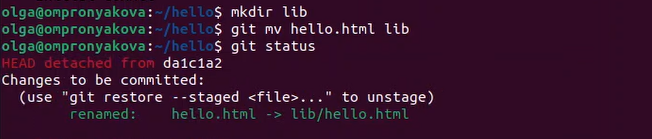


Рис. 24: Результат команд

Добавляю файл index.html в наш репозиторий и делаю коммит(рис.25).

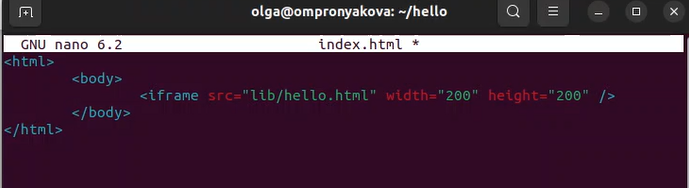


Рис. 25: Изменение файла

Каталог, в котором хранится вся информация git. здесь набор каталогов, имена которых состоят из 2 символов. Имена каталогов являются первыми двумя буквами хэша sha1 объекта, хранящегося в git. Смотрим в один из каталогов с именем из 2 букв. Вы увидите файлы с именами из 38 символов. Это файлы, содержащие объекты, хранящиеся в git(рис.26).

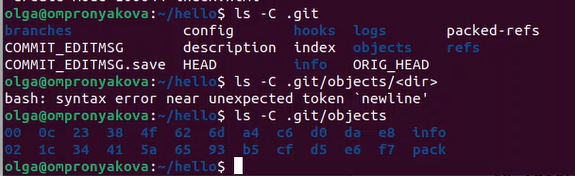


Рис. 26: Результат команд

Узнаю файлы в подкаталоге тегов. Каждый файл соответствует тегу, ранее созданному с помощью команды git tag. Его содержание — это всего лишь хэш коммита, привязанный к тегу. Каталог heads практически аналогичен, но используется для веток, а не тегов. На данный момент у нас есть только одна ветка, так что все, что вы увидите в этом каталоге – это ветка master(рис.27).

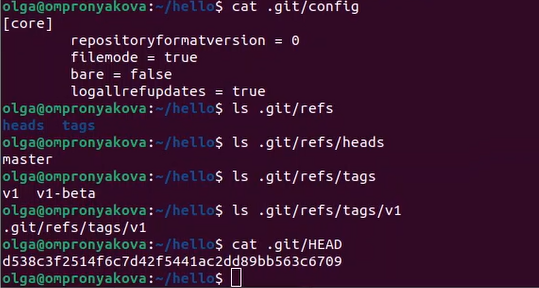


Рис. 27: Результат команд

Мы можем вывести дерево каталогов, ссылка на который идет в коммите. Это долж- но быть описание файлов (верхнего уровня) в нашем проекте (для конкретного коммита). Используйте SHA1 хэш из строки «дерева»(рис.28), (рис.29).

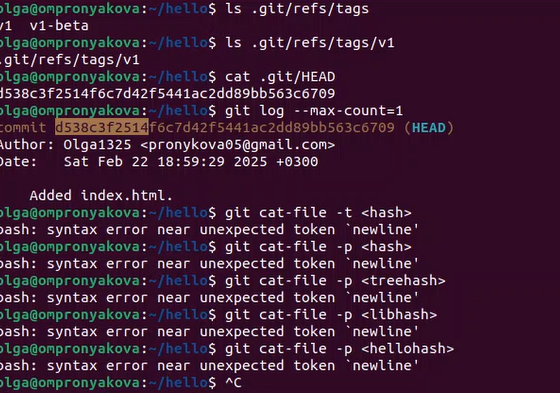


Рис. 28: Результат команд

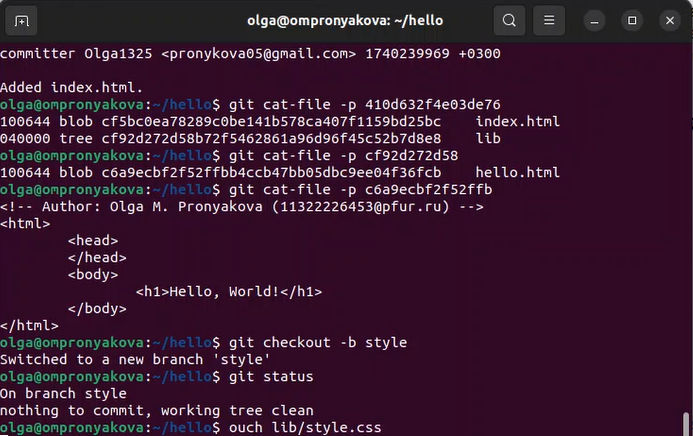


Рис. 29: Результат команд

создаем новую ветку «style». Добавляю файл стилей style.css(рис.30).

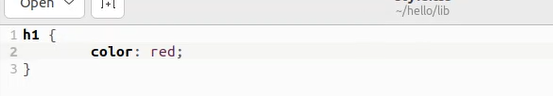


Рис. 30: Изменение файла

Обновляю файл hello.html, чтобы использовать стили style.css(рис.31).

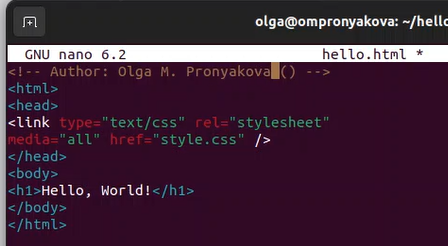


Рис. 31: Изменение файла

Обновляю файл index.html, чтобы он тоже использовал style.css(рис.32).

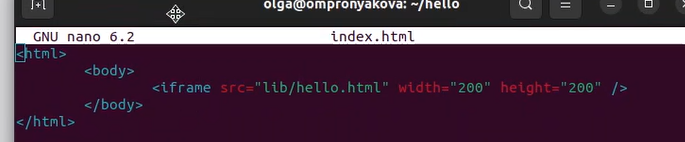


Рис. 32: Изменение файла

Теперь у нас 2 веткирис(рис.33), (рис.34).

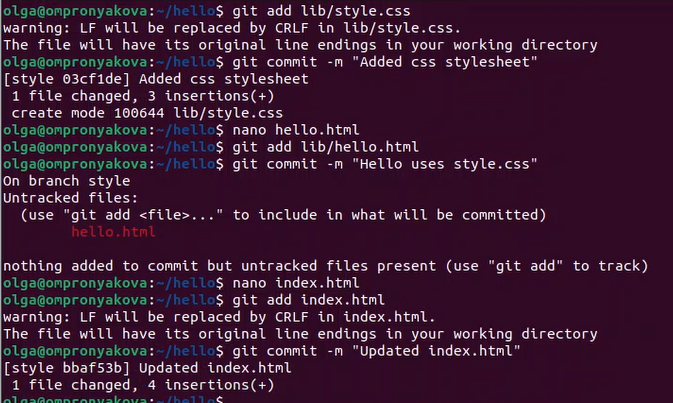


Рис. 33: Результат команд

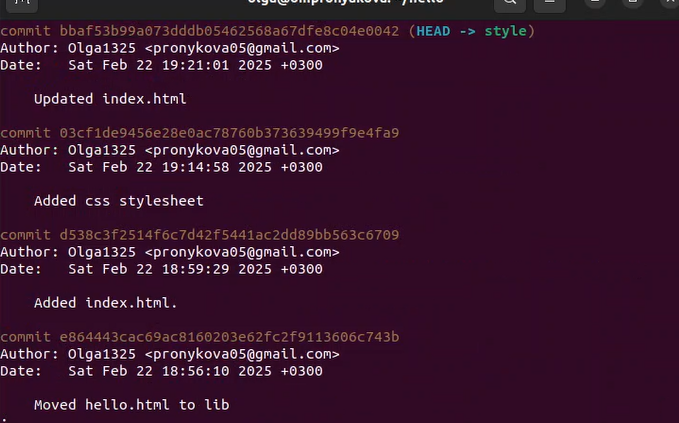


Рис. 34: Результат команд

# 3 Выводы

Вспомнила умения работы с git

# Список литературы