Лабораторная работа №1

Работа с git

Магомед Асхабович Мажитов

Содержание

# 1 Цель работы

* Изучить идеологию и применение средств контроля версий.
* Освоить умения по работе с git.

# 2 Ход работы

**1.** Я создал папку *hello*, а в нем *hello.html* файл. Далее инициализировался в *git*, изменил *hello.html* и добавил файл в репозиторий.(рис. 1)

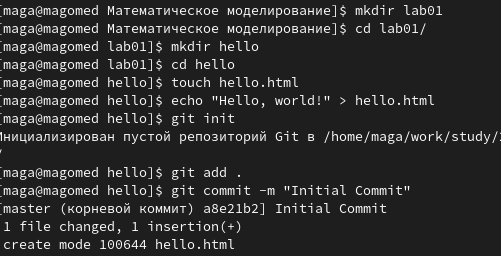


Рис. 1: Создание репозитория и первого файла

**2.** Далее я добавил теги в *hello.html* и отправил изменения в индекс git.(рис. 2, 3)

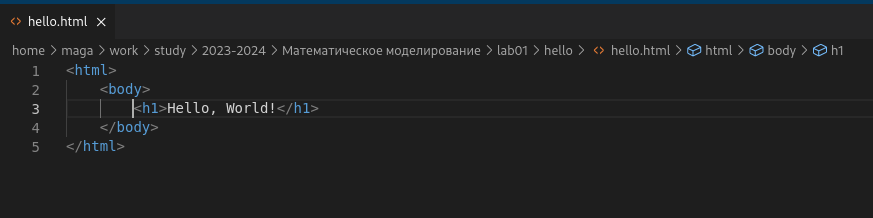


Рис. 2: Измененный файл

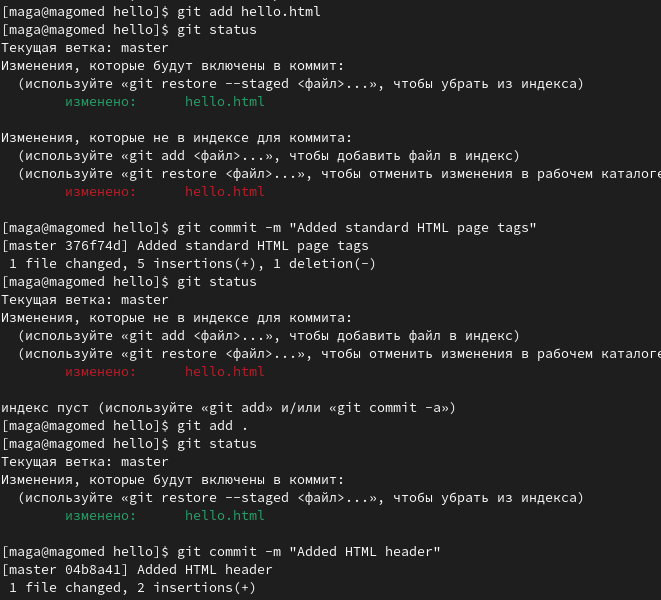


Рис. 3: Индексация изменений

**3.** Получил список проведенных изменений.(рис. 4)

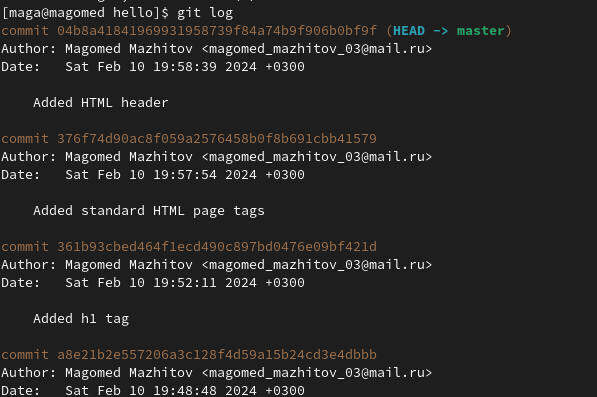


Рис. 4: Список проведенных изменений

**4.**Затем создал новый tag *v1*. (рис. 5)

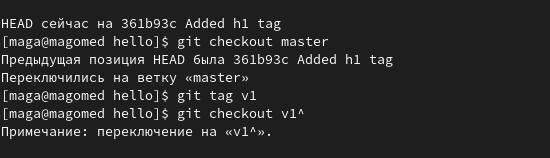


Рис. 5: Создание тегов версий

Попробовал переключаться между тегами. (рис. 6)

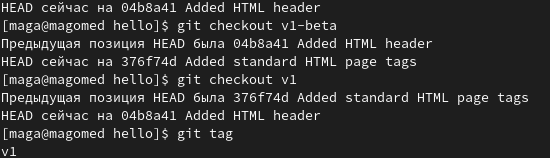


Рис. 6: Переключение между тегами

Далее просмотрел все теги. (рис. 7)

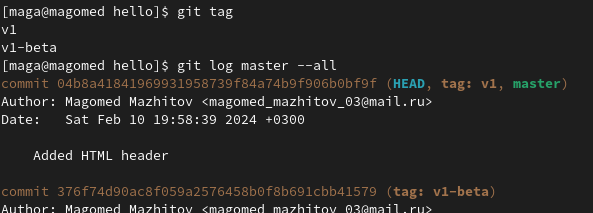


Рис. 7: Все теги

**5.** Внес изменение в файл hello.html в виде нежелательного комментария, проиндексировал это изменение и с помощью команды git reset сбросил буферную зону к HEAD.(рис. 8)

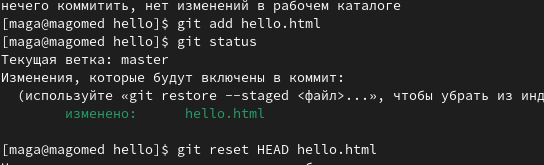


Рис. 8: Отмена проиндексированных изменений

Далее я снова изменил файл hello.html, но в этот раз внес также *commit*. Чтобы отменить коммит, сделал коммит, который удаляет изменения, сохраненные нежелательным коммитом.(рис. 9)

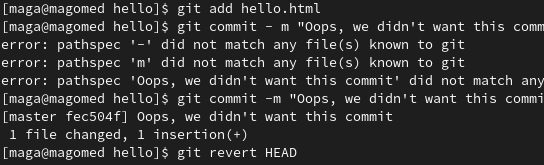


Рис. 9: Создание нового коммита, для дальнейшего удаления изменений

Сбросил коммиты к предшествующим коммиту Oops.(рис. 10)

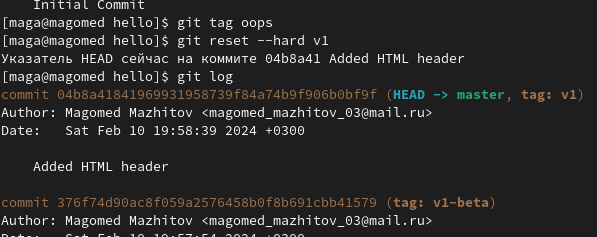


Рис. 10: Сброс коммита

Далее просто удаляем тег *Oops*.(рис. 11)

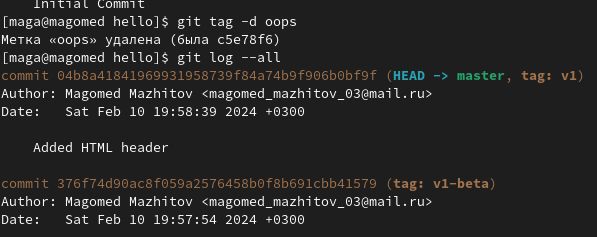


Рис. 11: Удаление тега Оops

**6.** Дабавил в страницу комментарий, проиндексировал его и закоммитил. Далее я снова внес изменения в *hello.html* и изменил последний коммит. (рис. 12)

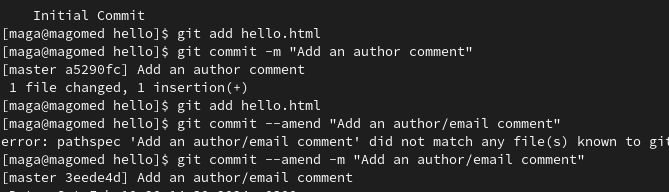


Рис. 12: Изменение коммита

**7.** Создал папку *lib* и перенес туда *hello.html* .(рис. 13)

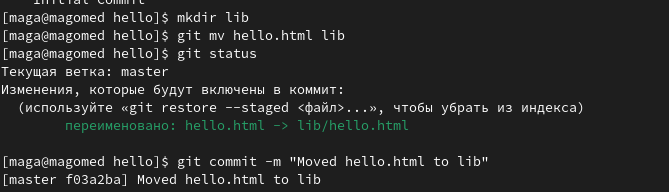


Рис. 13: Перемещение файла

**8.** Далее я создал *index.html*, проиндексировал его и добавил коммит.(рис. 14)

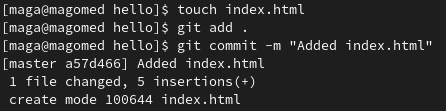


Рис. 14: Создание нового файла

**9.** Создал новую ветку *style*.(рис. 15)

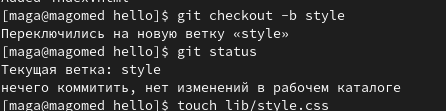


Рис. 15: Создание новой ветки

**10.** Добавил файл стилей.(рис. 16)

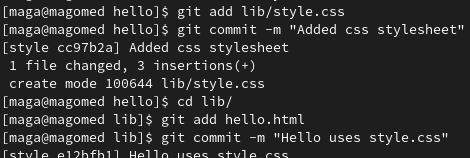


Рис. 16: Добавление стилей

**11.** ДДля переноса изменений из двух веток в одну, сделал слияние *master* c *style*.(рис. 17)

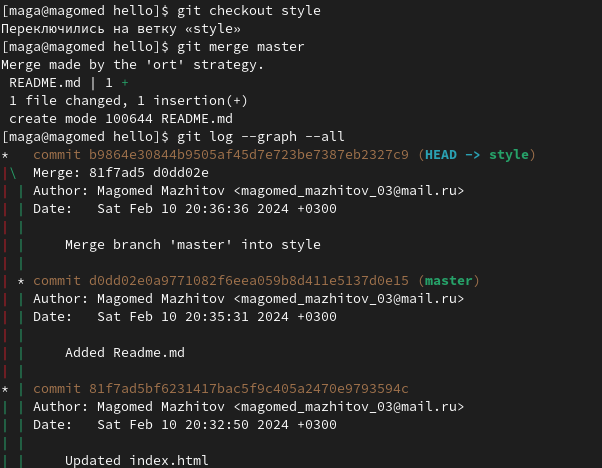


Рис. 17: Слияние веток

**12.** Вернемся на ветке style к точке перед тем, как мы слили ее с веткой master. Мы можем сбросить ветку к любому коммиту.(рис. 18)

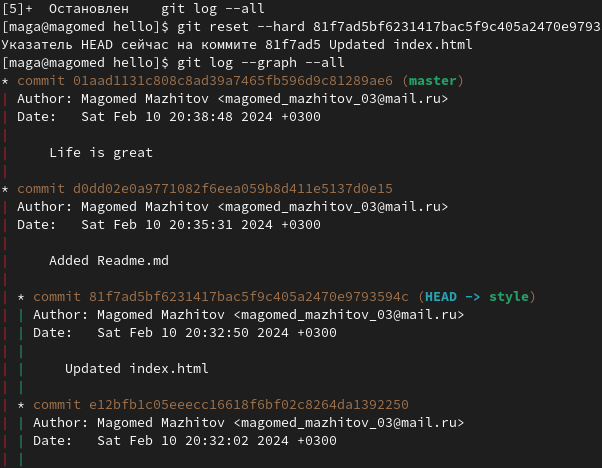


Рис. 18: Сброс ветки style

**13.**Создал клон репозитория.(рис. 19)

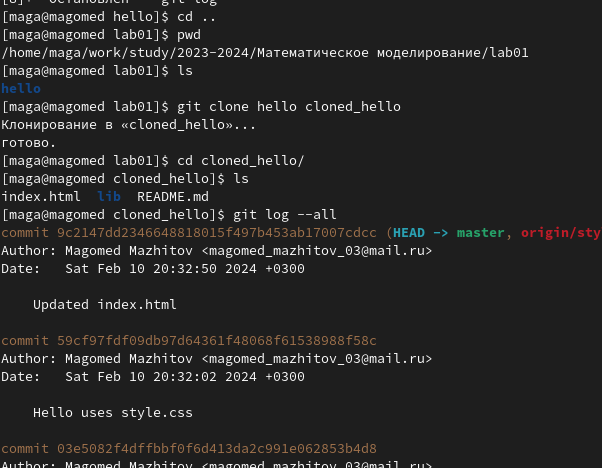


Рис. 19: Создание клона репозитория

**14.** Прописав команду *git remote* можно увидеть, что клонированный репозиторий знает об имени по умолчанию удаленного репозитория. Также с помощью *git branch -а* можно посмотреть все ветки, в том числе удаленные.(рис. 20)

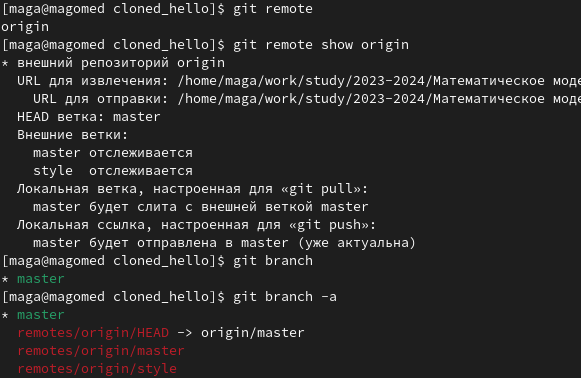


Рис. 20: Применение команд git remote и git branch

**15.** Далее я создал чистый репозиторий и добавил удаленный репозиторий *hello.git* к оригинальному репозиторию.(рис. 21)

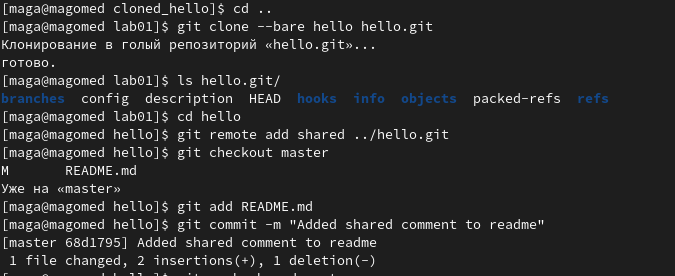


Рис. 21: Создание чистого репозитория и добавленние удаленного репозитория

**16.** Запушил все изменения.(рис. 22)

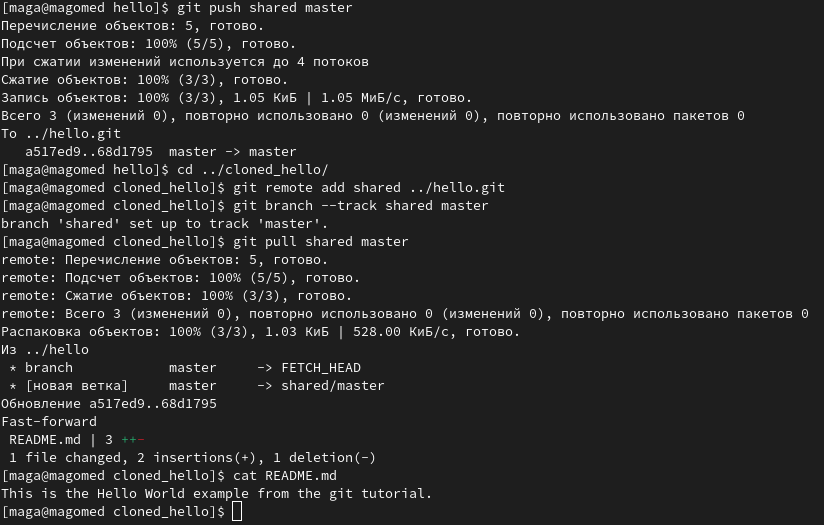


Рис. 22: Отправка всех изменений

# 3 Вывод

Мы изучили идеологию и применение средств контроля версий, а также освоили умения по работе с git.