Лабораторная работа №1

Настройка рабочего пространства. Система контроля версий Git

Щербак Маргарита Романовна, НПИбд-02-21

2024

Содержание

# Цель работы

Настроить рабочее пространство для лабораторной работы. Изучить систему контроля версий Git.

# Задание

Создать репозиторий курса, настроить каталог курса и выполнить задания из файла по работе с Git.

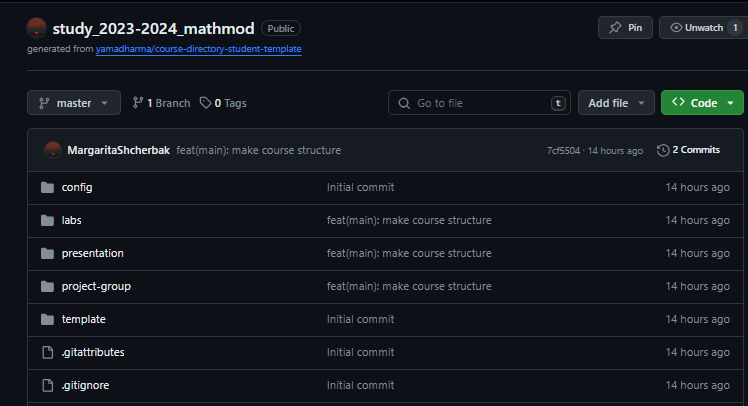
# Теоретическое введение

Git — это система управления версиями, которая используется для отслеживания изменений в файлах кода или любых других текстовых файлах в проекте. Он позволяет разработчикам работать вместе над проектами, фиксировать изменения, откатываться к предыдущим версиям кода и управлять конфликтами при слиянии изменений.  
Вот несколько ключевых концепций Git:  
- Репозиторий (Repository): Это место, где хранится весь исторический код и его версии. Репозиторий может быть локальным (на вашем компьютере) или удаленным (на сервере). - Коммит (Commit): Это снимок текущего состояния файлов в вашем проекте. Каждый коммит имеет уникальный идентификатор, сообщение о коммите и метку времени.  
- Ветка (Branch): Это отдельная линия разработки в вашем репозитории. Она позволяет вам работать над определенной функцией или исправлением без влияния на основную версию проекта. Ветки могут быть созданы, объединены и удалены.  
- Слияние (Merge): Это процесс объединения изменений из одной ветки с другой. Это позволяет объединить работу нескольких разработчиков в одном проекте.  
- Конфликт слияния (Merge Conflict): Это ситуация, когда Git не может автоматически слить изменения из разных веток из-за конфликта в одном и том же участке кода. Это требует ручного разрешения.

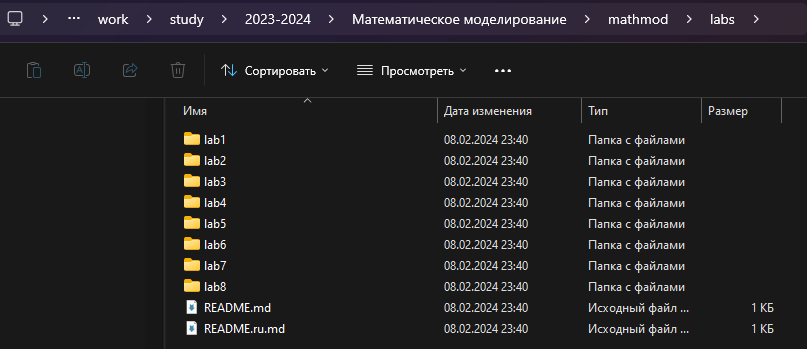
Git является широко используемым инструментом в современной разработке программного обеспечения благодаря своей гибкости, масштабируемости и эффективности.

# Выполнение лабораторной работы

1. Создала репозиторий курса и настроила каталог курса (рис.1 - рис. 2).

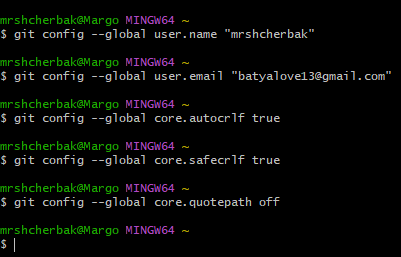


Репозиторий курса

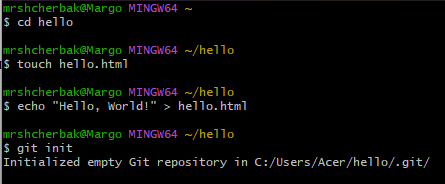


Рабочий каталог курса

1. Установила имя и эл. почту, а также параметры установки окончаний строк и установила отображения unicode, после чего создала страницу «Hello, World» и репозиторий (рис.3 - рис.4).

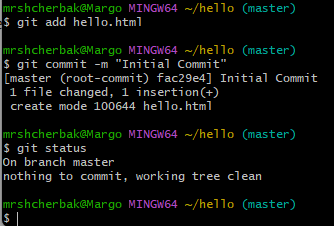


Выполнение команд



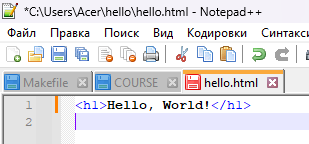
Выполнение команд

1. Добавила файл в репозиторий и проверила состояние репозитория (рис.5).



Выполнение команд

1. Изменила страницу «Hello, World» (рис.6).



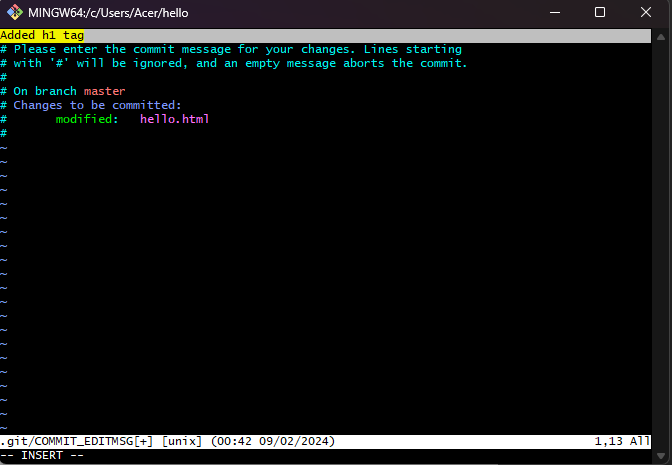
Редактирование файла hello.html

1. Перешла к индексации изменений (рис.7). Изменения файла hello.html были проиндексированы. Это означает, что git теперь знает об изменении, но изменение пока не записано в репозиторий. Следующий коммит будет включать в себя проиндексированные изменения.

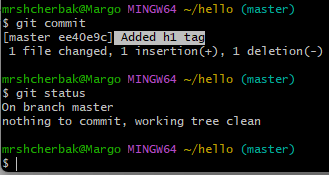


Индексация изменений

1. Сделала коммит изменений (рис.8 - рис.9). В первой строке ввела комментарий: «Added h1 tag». Сохранила файл и вышла из редактора. Рабочий каталог чистый, можно продолжить работу.

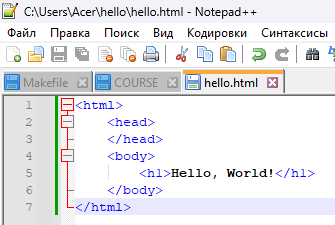


Редактирование файла hello.html

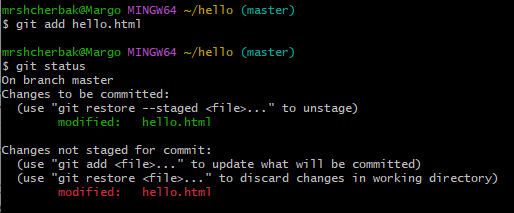


Выполнение команд

1. Добавила стандартные теги страницы (Рис.10 - рис.11). hello.html указан дважды в состоянии. Первое изменение (добавление стандартных тегов) проиндексировано и готово к коммиту. Второе изменение (добавление заголовков HTML) является непроиндексированным. Если бы вы делали коммит сейчас, заголовки не были бы сохранены в репозиторий.

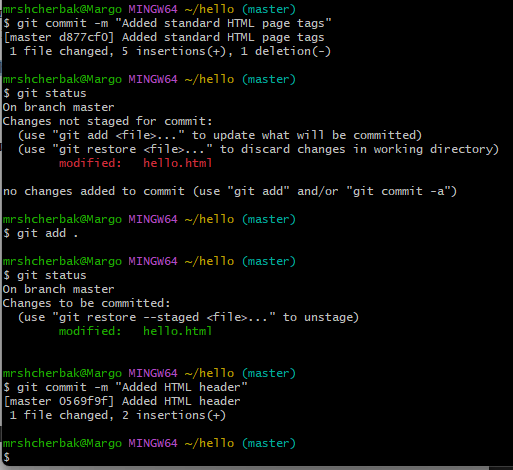


Редактирование файла hello.html



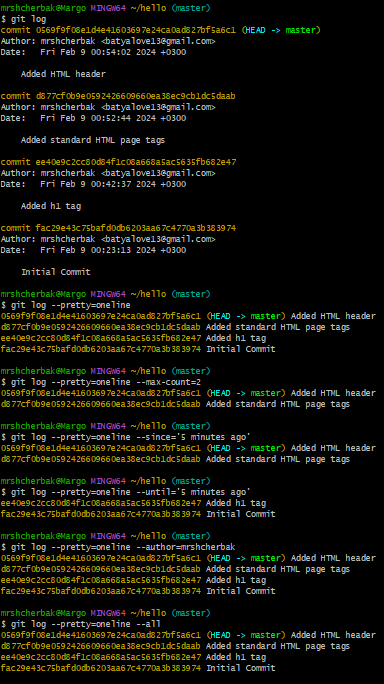
Выполнение команд

Произведла коммит проиндексированного изменения (значение по умолчанию), а затем еще раз проверила состояние. Состояние команды говорит о том, что hello.html имеет незафиксированные изменения, но уже не в буферной зоне. Теперь добавила второе изменение в индекс, а затем проверила состояние с помощью команды git status. Сделала коммит второго изменения (рис.12).



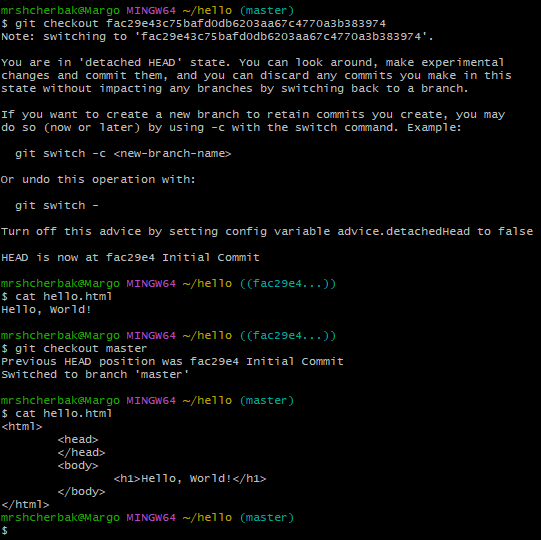
Выполнение команд

1. Получила список произведенных изменений. Есть много вариантов отображения лога (рис.13).



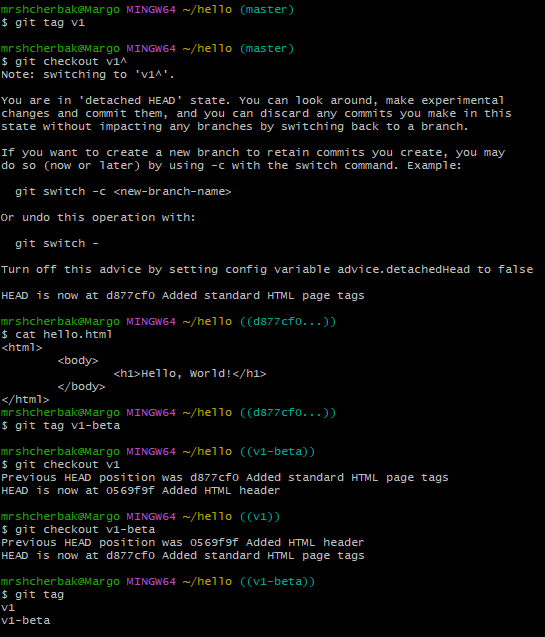
Выполнение команд

1. Перешла к этапу получения старых версий (рис.14).



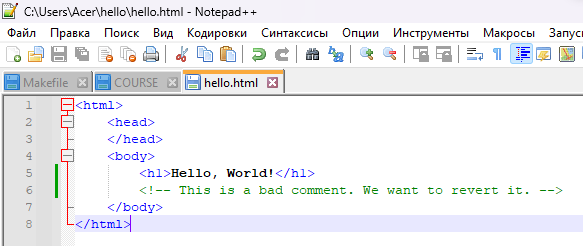
Выполнение команд

1. Перешла к созданию тегов версий. Переключение по имени тега. Просмотр тегов с помощью команды tag (рис.15).

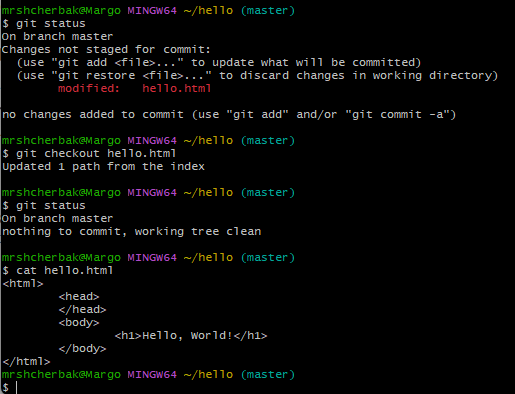


Выполнение команд

1. Перешла к этапу отмены локальных изменений (до индексации). Внесла изменение в файл hello.html в виде нежелательного комментария. hello.html был изменен, но еще не проиндексирован. Использовала команду git checkout для переключения версии файла hello.html в репозитории. Команда git status показывает нам, что не было произведено никаких измнений, не зафиксированных в рабочем каталоге (рис.16 - рис.17).

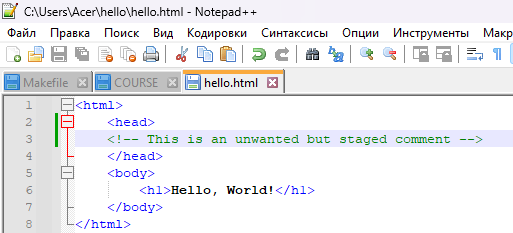


Редактирование файла hello.html

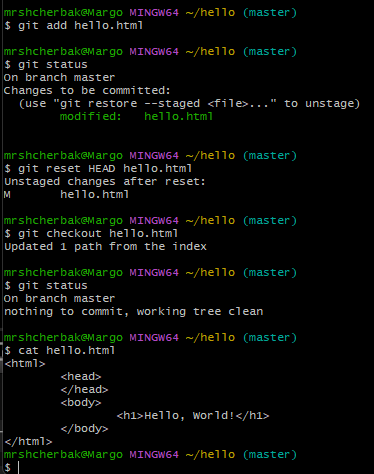


Выполнение команд

1. Отмена проиндексированных изменений (перед коммитом). Изменила файл и проиндексировала изменения. Проверила состояние. Выполнила сброс буферной зоны. Переключилась на версию коммита (Рис.18 - рис.19).

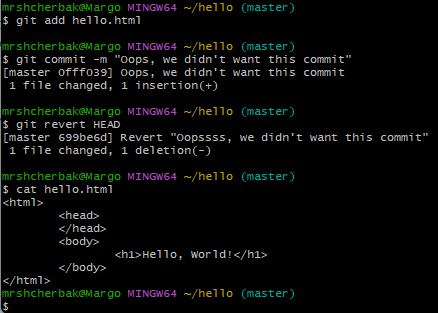


Редактирование файла hello.html

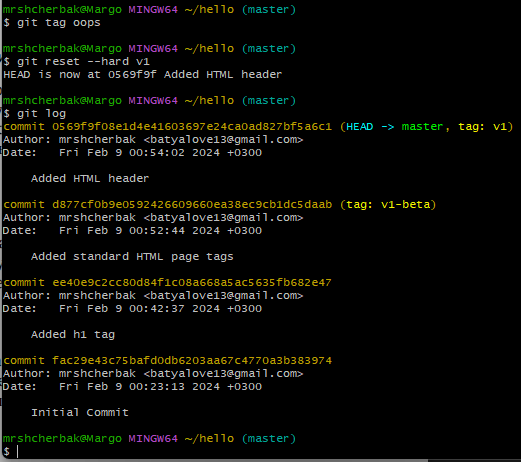


Выполнение команд

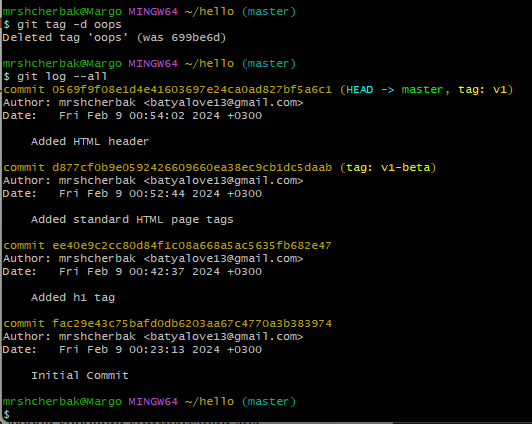
1. Перешла к этапу отмены коммитов. Отменила коммит путем создания нового коммита, отменяющего нежелательные изменения. Чтобы отменить коммит, необходимо сделать коммит, который удаляет изменения, сохраненные нежелательным коммитом. Сброс коммитов к предшествующим коммиту Oops. Удаление тега oops (рис.20 - рис.22).



Выполнение команд

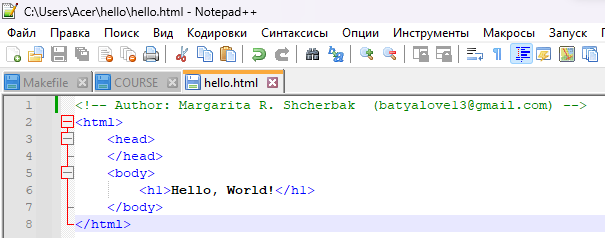


Выполнение команд

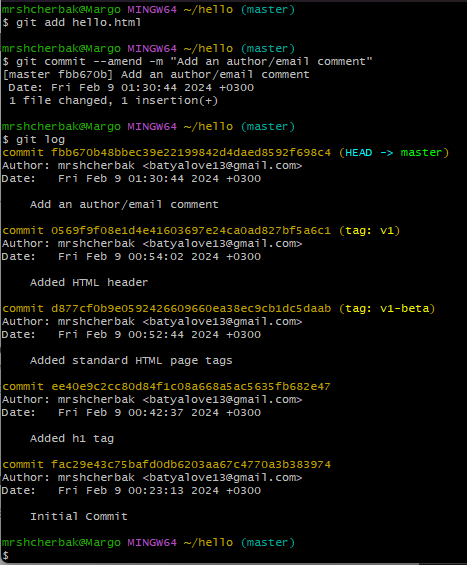


Выполнение команд

1. Внесение изменений в коммиты. Мы можем увидеть, что оригинальный коммит «автор» заменен коммитом «автор/email». Этого же эффекта можно достичь путем сброса последнего коммита в ветке, и повторного коммита новых изменений (рис.23 - рис.24).

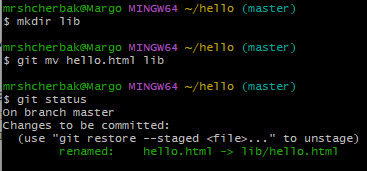


Редактирование файла hello.html



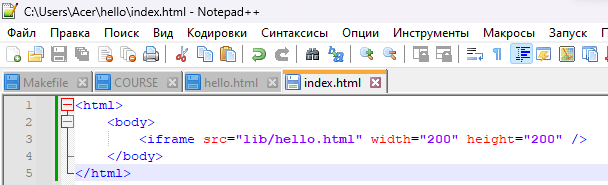
Выполнение команд

1. Перемещение файлов. Сейчас мы собираемся создать структуру нашего репозитория. Давайте перенесем страницу в каталог lib (рис.25).



Выполнение команд

1. Подробнее о структуре. Добавление index.html. Теперь при открытии index.html, вы должны увидеть кусок страницы hello в маленьком окошке (рис.26 - рис.27).

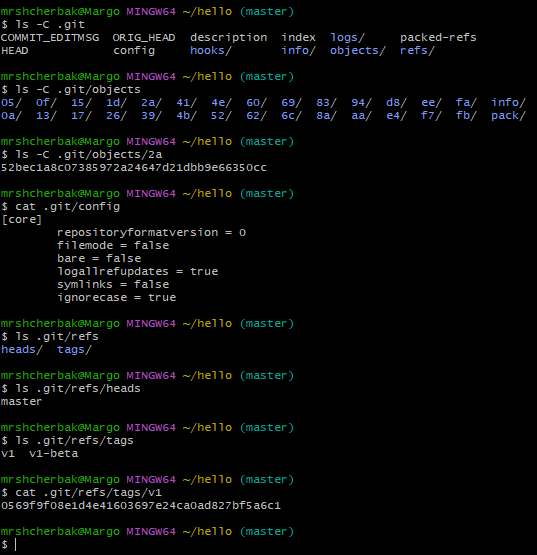


Редактирование файла index.html



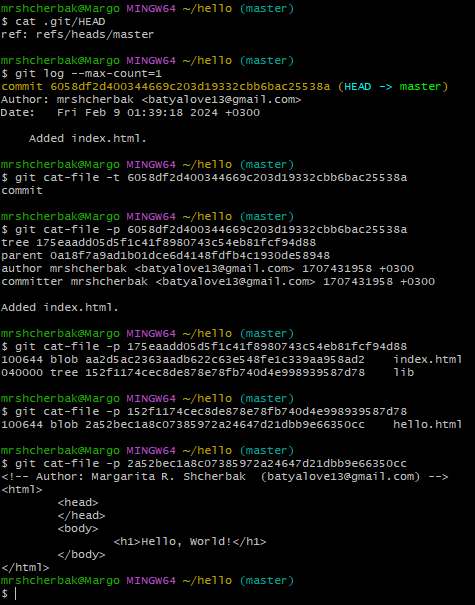
Кусок страницы hello в маленьком окошке

1. Git внутри: каталог .git (рис.28).



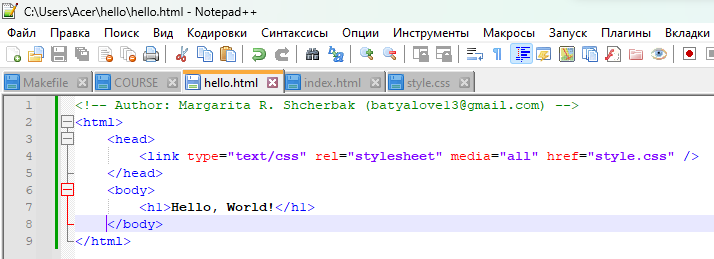
Выполнение команд

1. Работа непосредственно с объектами git. Поиск последнего коммита. Вывод последнего коммита с помощью SHA1 хэша. Поиск дерева.Вывод каталога lib. Вывод файла hello.html (рис.29).

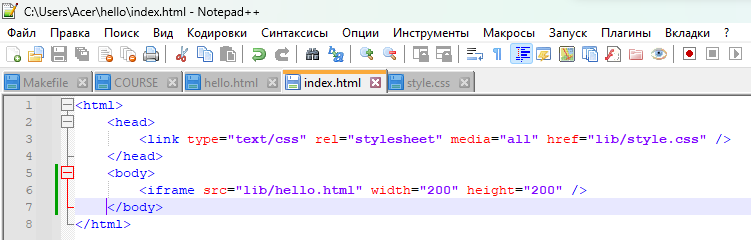


Выполнение команд

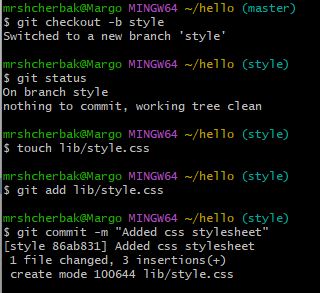
1. Создание ветки. Добавила файл стилей style.css. Изменила основную страницу. Изменила index.html (рис.30 - рис.32).



Редактирование файла hello.html

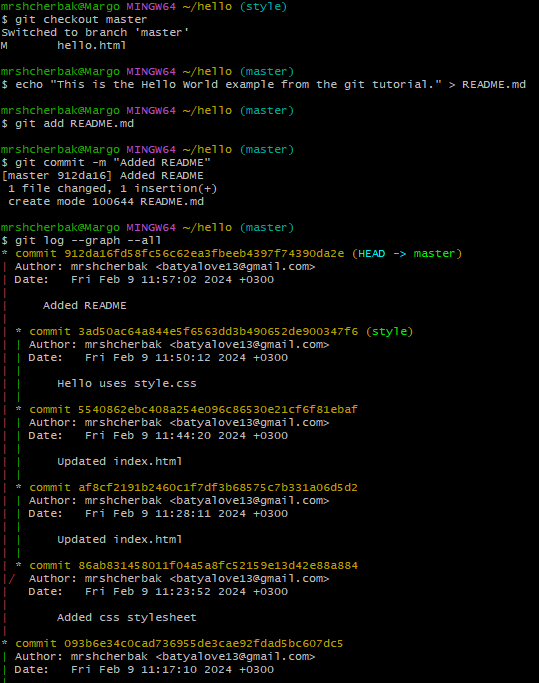


Редактирование файла index.html

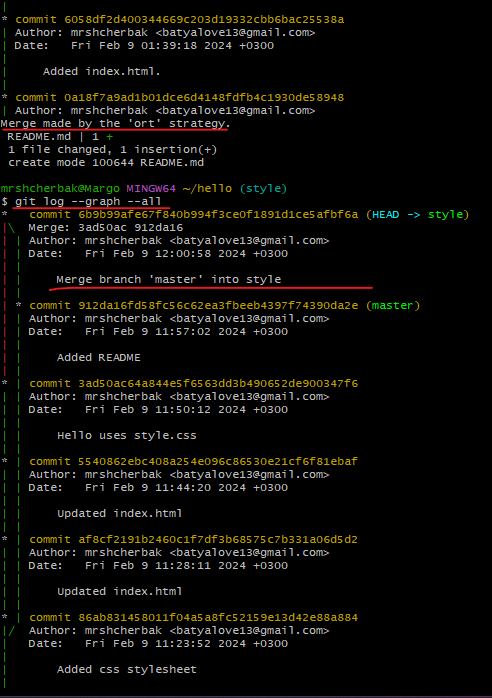


Выполнение команд

1. Изменения в ветке master. Сделала коммит изменений README.md в ветку master. Слияние веток (рис.33 - рис.34).



Выполнение команд

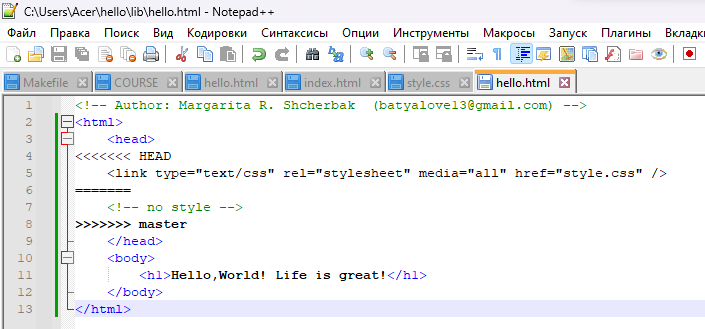


Выполнение команд

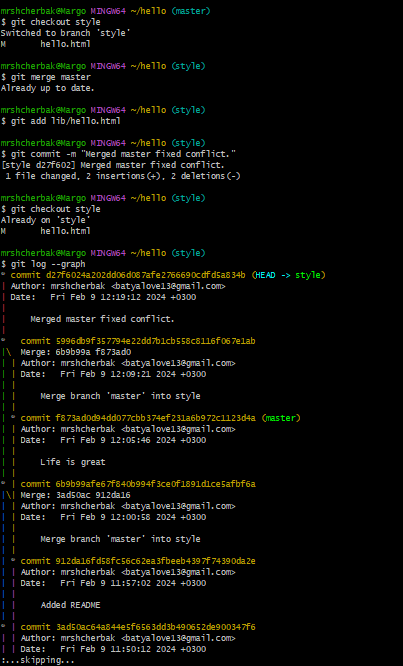
1. Создание конфликта и разрешение конфликтов. Слияние master с веткой style. Необходимо вручную разрешить конфликт. Сброс ветки style. Сброс ветки master. Слияние в ветку master. Теперь ветки style и master идентичны (рис.35 - рис.40).



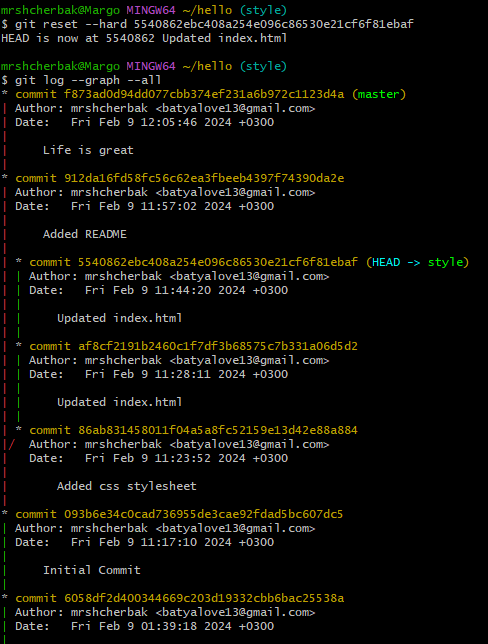
Выполнение команд



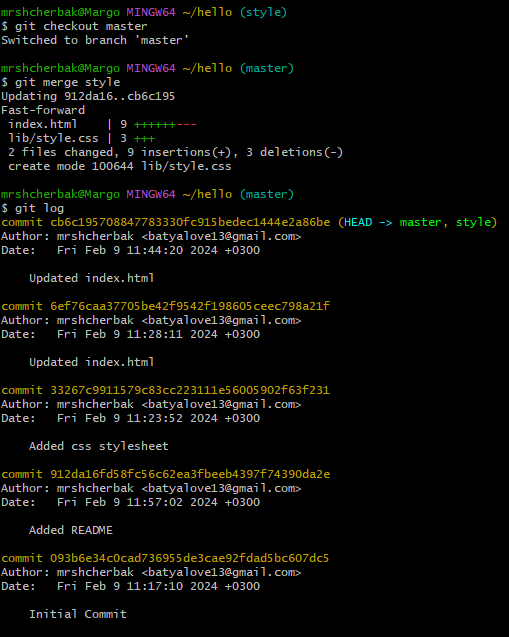
Содержимое файла lib/hello.html



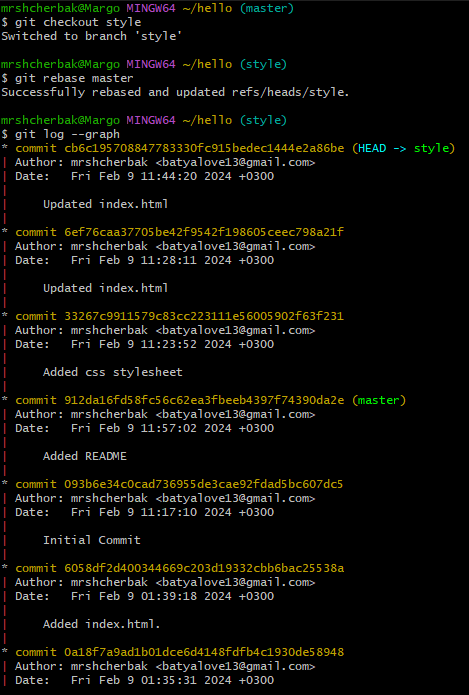
Выполнение команд



Выполнение команд

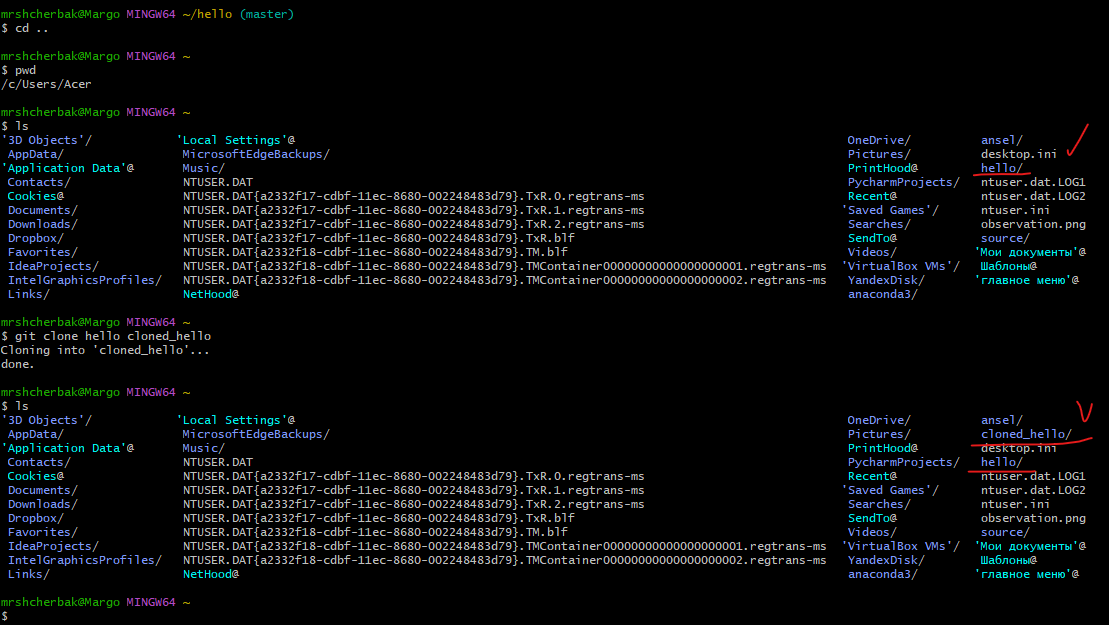


Выполнение команд



Выполнение команд

1. Клонирование репозиториев. Просмотр клонированного репозитория (рис.41 - рис.42).

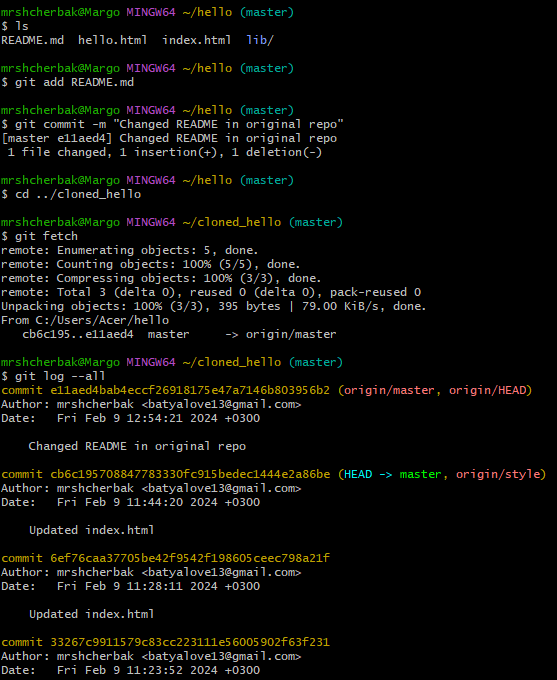


Выполнение команд

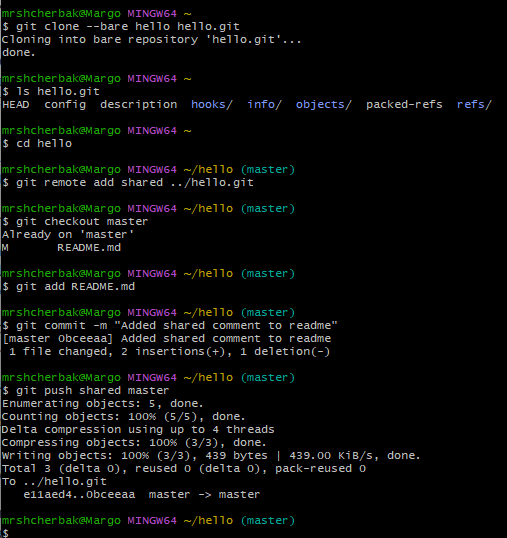


Выполнение команд

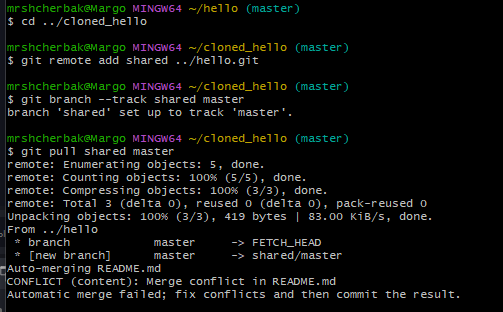
1. Изменение оригинального репозитория. Извлечение изменений. Слияние извлеченных изменений. Чистые репозитории. Добавление удаленного репозитория. Отправка изменений. Извлечение общих изменений (рис.43 - рис.46).



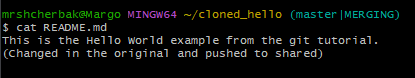
Выполнение команд



Выполнение команд



Выполнение команд



Выполнение команд

# Выводы

Таким образом, в ходе ЛР№1 я настроила рабочее пространство для лабораторной работы. Изучила систему контроля версий Git.