**Лабораторна робота. Ігнорування файлів. Порівняння змін. Видалення і переміщення файлів.**

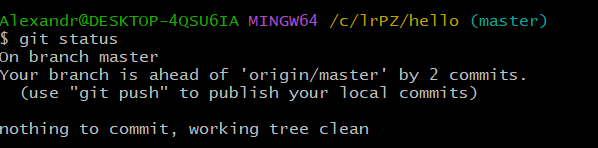
**Мета роботи:** навчитися виключати файли, які немає необхідності вести в системі контролю версій. Отримати практичні навички порівняння виконаних змін в файлах.

**Хід роботи**

**Основне завдання:**

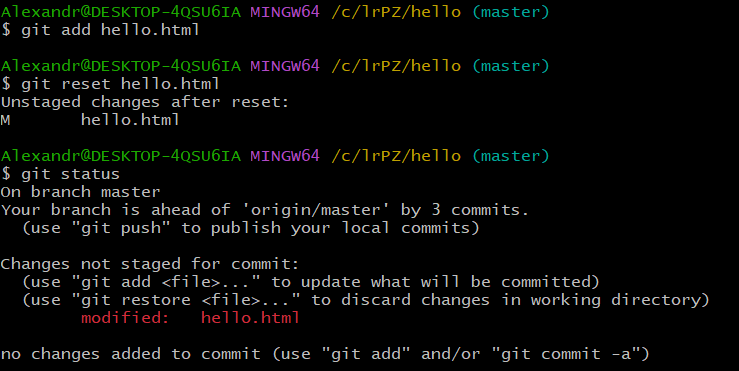
Для того, щоб зробити ігнорування файлів .txt у папці temp у файлі .gitignore вкажу: temp/\*.txt . Проіндексую зміни й виконаю commit:

У файл some.txt я записав текст, після чого виконав перевірку статусу репозиторію, у результаті чого отримав:

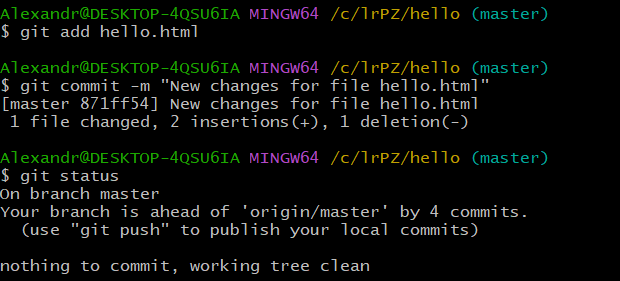


Отже, Git проігнорував зміни, які були виконані в текстовому файлі.

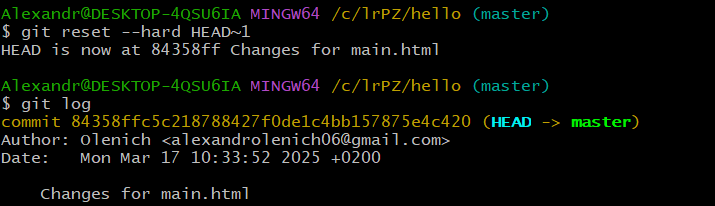
Внесу зміни в файл hello.html і проіндексую його. Після цього відміню зміни:



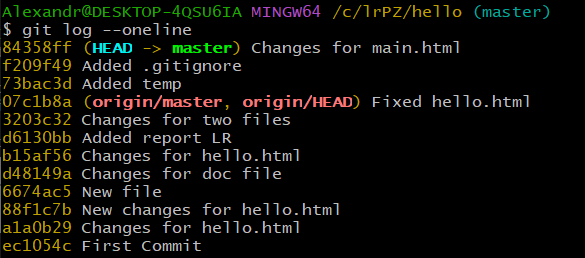
Знову внесу зміни в hello.html, проіндексую його й виконаю commit:



Тепер видалю останній commit за допомогою команди: git reset --hard HEAD~1.

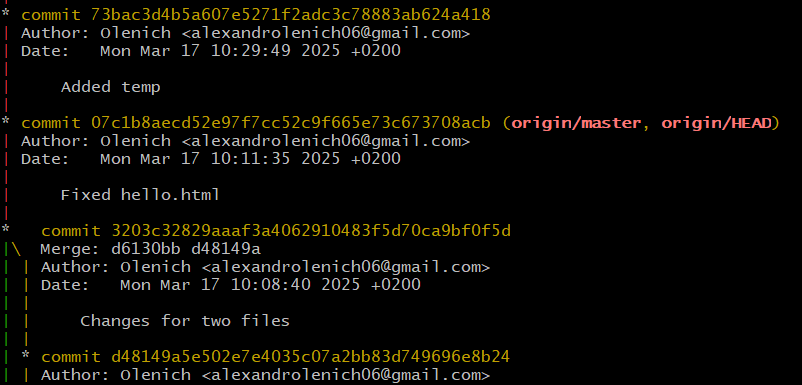


Після цього ні змін у файлі, ні коміту не буде. За потреби до команди git log можна використати додаткові прапори, наприклад:



Цей прапор виводить кожен коміт в одну лінію.

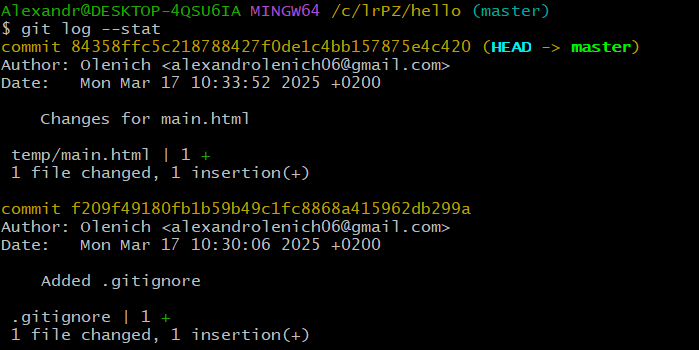
git log --graph виводить граф, що показує структуру комітів:



Можна також скористатися командою git log --date=relative для того, щоб переглянути наскільки давно був виконаний commit:

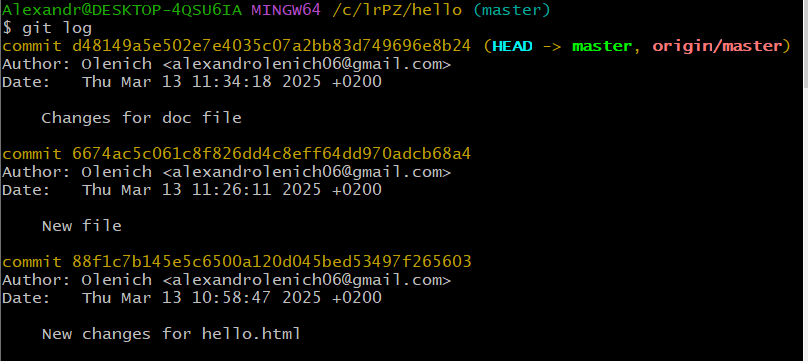


Можна також переглядати файли, які були змінені під час коміту, фільтрувати пошук за певними критеріями і т.д. за допомогою різних додаткових прапорів.

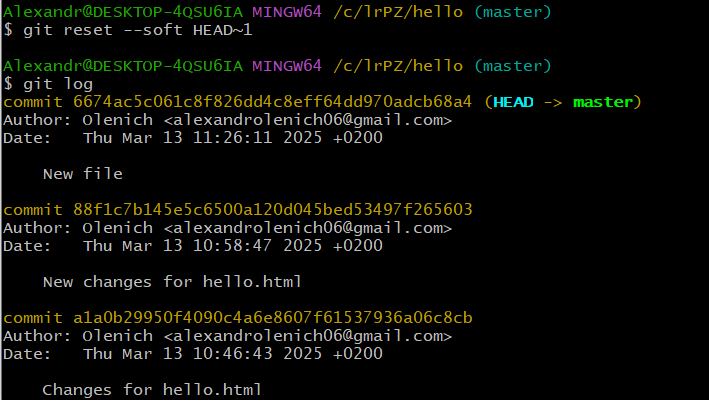


**Додаткове завдання:**

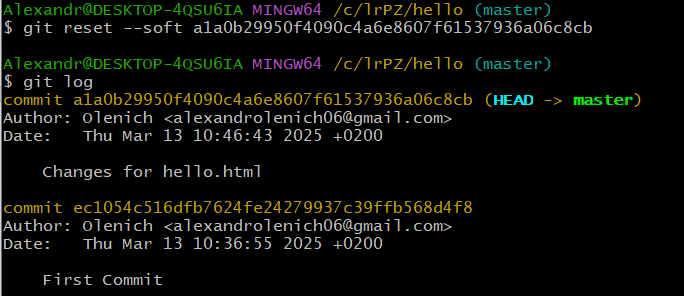
Спочатку виведу статус локального репозиторію, для подальшого порівняння:



Після цього зроблю git reset --soft HEAD~1, за допомогою цієї команди виконається переміщення на один коміт назад, на скріншоті це добре видно:



Якщо потрібно відмінити зміни до якогось визначеного коміту, можна просто замість «HEAD~1» ввести хеш цього коміту, наприклад виконаю переміщення до «Changes for hello.html».



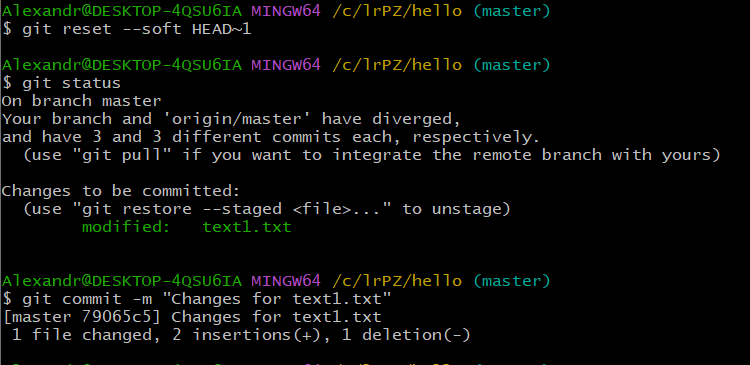
Виконавши ці дії, у git status можна помітити не зафіксовані дії, бо коміти, які ці дії затверджували були відмінені. Зроблю ще декілька commit для того, щоб перейти до наступного способу скидання.

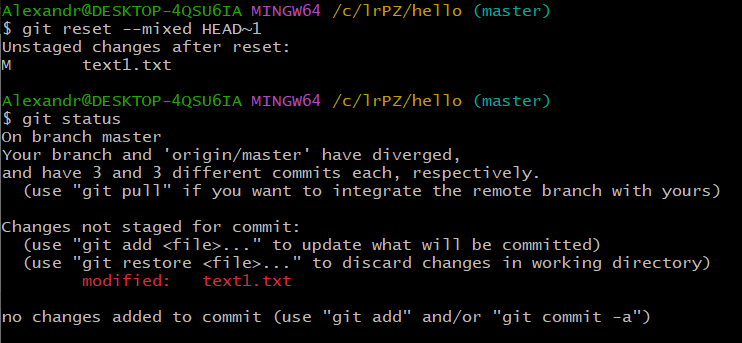


Відміню один коміт:

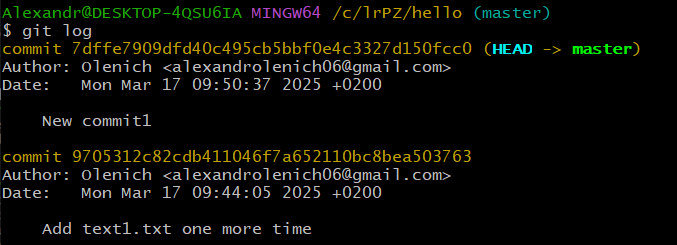


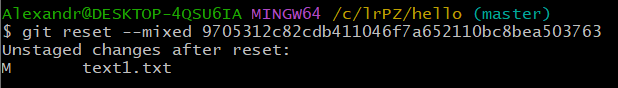
На наступному прикладі можна помітити відмінність перших двох способів виконання reset. При м’якому відкаті коміту файл залишається проіндексованим, залишається просто його комітувати, але ж якщо зміни комітувати і зробити git reset --mixed, то потрібно буде знову індексувати файл за допомогою git add і робити коміт.

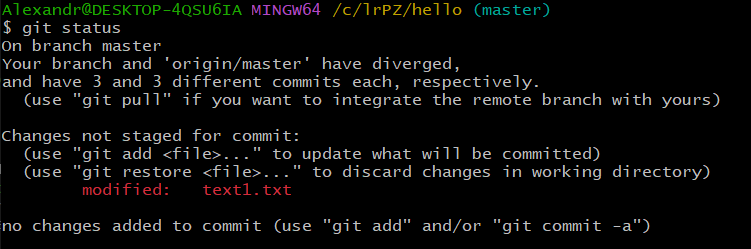




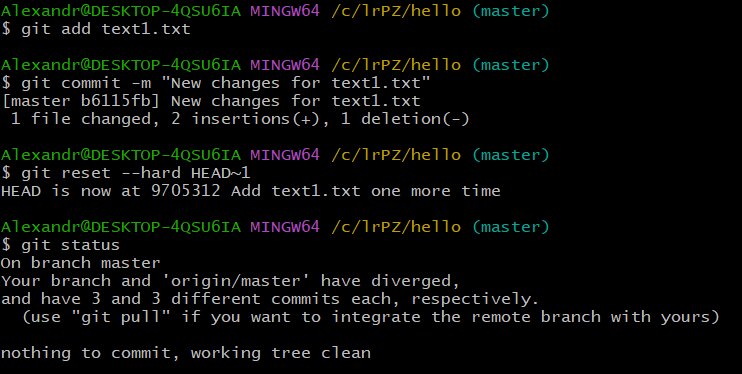
При відкаті за індексом коміту результат буде схожим на результат використання git reset --soft , але він, як згадано раніше відрізняється тим, що при відкаті до певного коміту, індексація змін також відміняється.







Знову проіндексую зміни й виконаю commit, тепер зроблю відкат за допомогою git reset --hard HEAD~1, у результаті зміни взагалі повністю очистилися, тепер файл text1.txt повернувся до початкового стану.

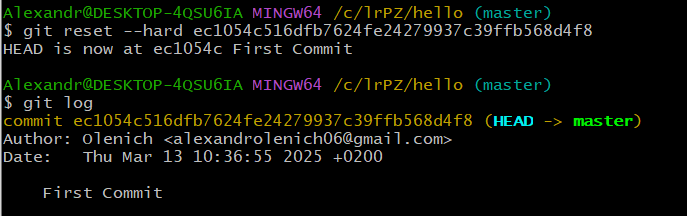


Ніяких змін підтверджувати немає потреби, бо тепер їх не існує, вони видалені повністю.

Зроблю відкат до коміту «New text in hello.html»:



Файл text1.txt і разом з ним всі зміни, що були після цього коміту видалилися повністю, тепер файлу немає в папці локального репозиторію. Для того, щоб зробити відкат до першого коміту, потрібно ввести його хеш.



Таким чином я повернувся до першого стану репозиторію, де є тільки hello.html.

Під час оновлення вмісту віддаленого репозиторію GitHub довелося зіткнутися з командою git pull, яка виконала злиття всіх комітів з локального репозиторію і комітів з віддаленого, у результаті чого файл hello.html отримав зміни, але їх можна було легко виправити. Також звітів з минулої роботи стало два, через це злиття.

**Висновок.** Під час виконання лабораторної роботи я навчився виключати файли, які немає необхідності вести в системі контролю версій. Отримав практичні навички порівняння виконаних змін в файлах. А також ознайомився з можливостями відхилення змін за допомогою git reset.