КІТ-33 Стрєльцов Віталій Вікторович 30.03.2015 13.04.2015 \_\_\_\_\_\_ група ПІБ студента Дата виконання Дата прийому Оцінка

**ЗВІТ**

Лабораторна робота №2

Тема: Вивчення базових принципів роботи з Git

Харків 2015

**Мета:** Познайомитись з основами використання системи контролю версій Git.

**Завдання**

Частина 1. Ініціалізація сховища та перший комміт.

1. Створити робочий каталог. Ініціалізувати в ньому репозиторій. Налаштувати файл config.
2. Створити в ній робочий каталог c ім'ям student\_name (де name - Ваше прізвище).
3. Створити в робочому каталозі 3 файли проекту: (приклад hello\_name.rb) (де name - Ваше прізвище). Перевірити статус репозиторія.
4. Проіндексуйте файли.
5. Зробити перший commit з коментарем ''first commit name"(де name - Ваше прізвище).
6. Зробити зміни в файлах. Зробити commit змін.

Частина 2. Робота зі змінами.

1. Зробити зміни в одному з файлів. Проіндексувати файл.
2. Зробити нові зміни в файлі. Перевірити статус.
3. Зробити commit. Перевірити статус. Пояснити результат.
4. Вивести історію проекту. $ Git log.
5. Використовуйте команду $ git log --pretty = oneline, налаштовуючи відповідні параметри виведіть на екран останні commits за 5 хвилин.

Частина 3. Робота з попередніми станами репозиторія.

1. Виконайте повернення стану файлів робочого каталогу на другий commit з часу початку лабораторної роботи (ОБОВ'ЯЗКОВО використовуйте команду git checkout <hash>).
2. Проведіть повернення стану файлів до останнього commit.

Частина 4. Робота з тегами.

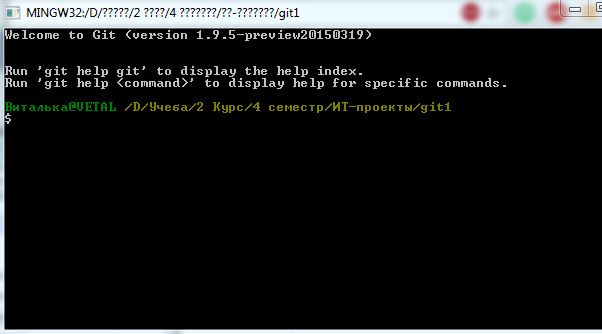
1. Створіть тег в поточному snapshot репозиторію. Назва тега - V1\_name (де name - Ваше прізвище).
2. Виконайте зміни і як мінімум два commits в поточному каталозі.
3. Продемонструйте повернення до стану репозиторія позначеного тегом.
4. Виконайте повернення до останнього стану master.

Частина 5. Робота з функцією revert.

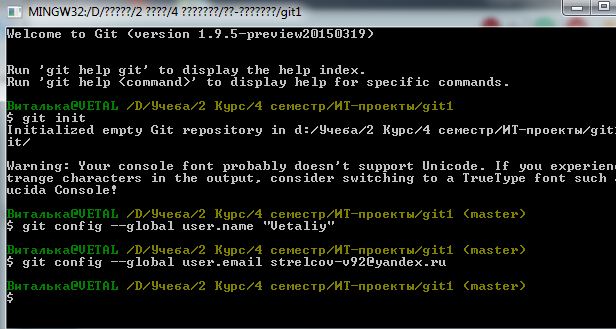
1. Виконайте зміни поточного стану робочого каталогу.
2. Проіндексуйте зміни.
3. Командою revert поверніть початковий стан файлів.
4. Зробіть зміну стану каталогу.
5. Проіндексуйте і виконайте commit.
6. Виконайте revert останнього commit.
7. Продемонструйте log.

**Хід роботи**

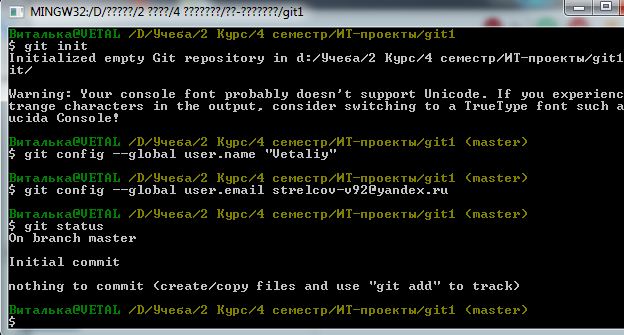
Створимо робочий каталог на диску С з назвою «Рабочий каталог». Зайдемо в цей каталог, клік правою кнопкою миші, натискаємо на «Git Init Here». Клікаємо павою кнопкою миші на каталозі і вибираємо «Git Bash». Відкриється консоль для роботи з Git.



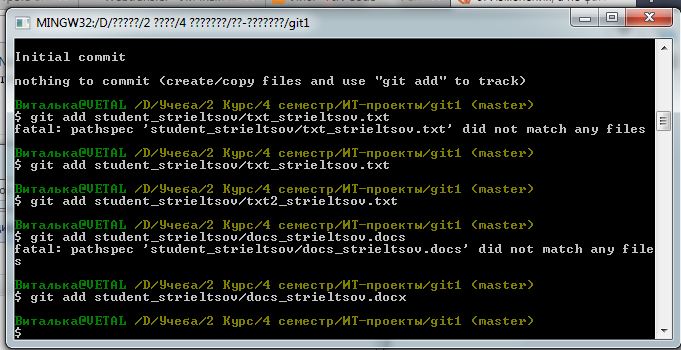
Налаштуємо файл config. Використаємо команди git config --global user.name і git config --global user.email.



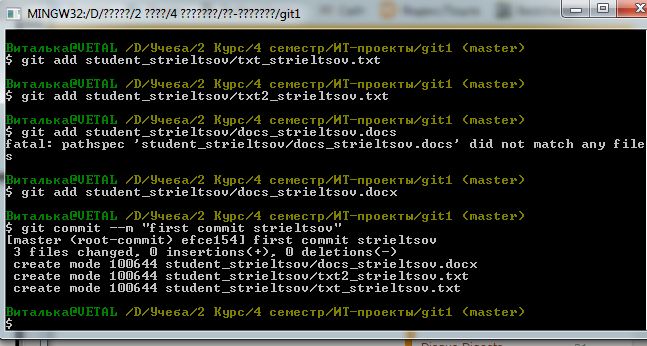
Створимо каталог student\_strieltsov всередині даного каталогу. Створимо всередині student\_strieltsov три файли: txt\_strieltsov.txt, txt2\_strieltsov.txt, docx\_strieltsov.docx. Перевіримо статус репозиторія.



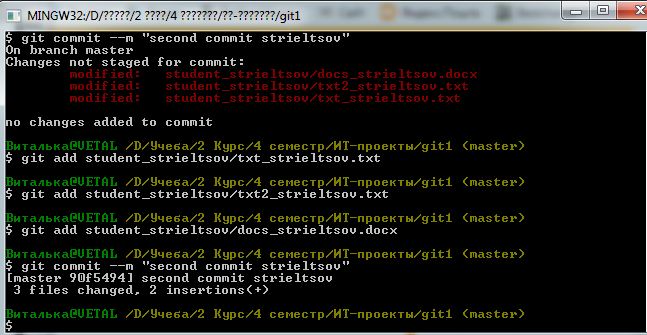
Проіндексуємо дані три файли командою $ git add.



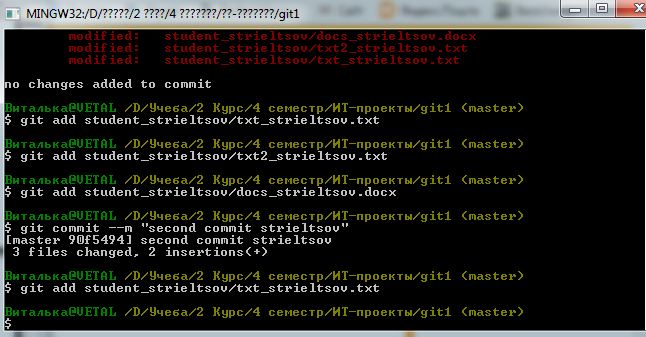
Робимо перший commit з коментарем «first commit strieltsov». Команда git commit –m «first commit strieltsov».



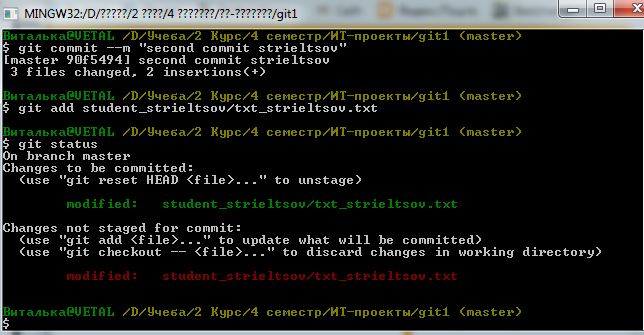
Зробимо зміни в файлах і зробимо комміт. Але спочатку проіндексуємо файли.



Зробимо зміни в файлі txt\_strieltsov.txt і проіндексуємо файл (git add).

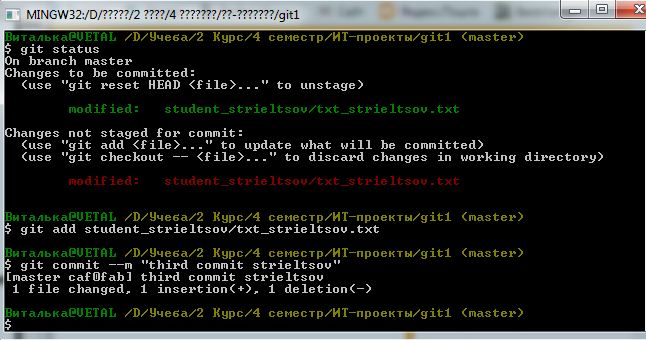


Зробимо нові зміни в файлі txt\_strieltsov.txt. Перевіримо статус (git status).



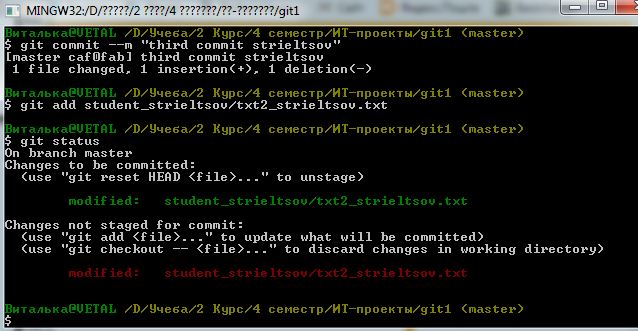
Ми бачимо, що файл txt\_strieltsov.txt непроіндексований(виділено червоним).

Проіндексуємо його і зробимо комміт.



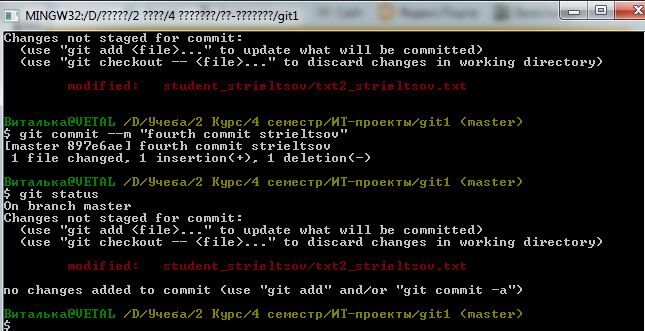
Все отрималось, отже тільки після того як файл проіндексовано, можна робити його комміт.

Подивимось на інший результат. Зробимо зміни в файлі txt2\_strieltsov.txt та проіндексуємо його.



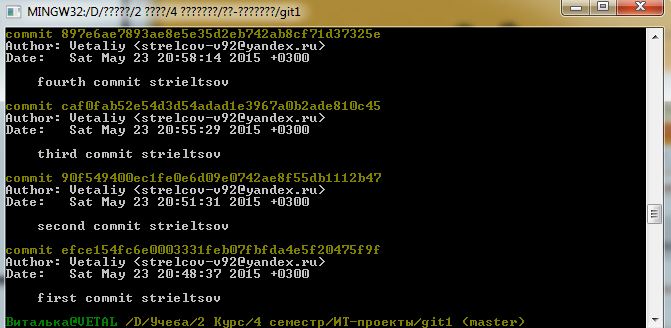
Зробимо нові зміни в файлі txt2\_strieltsov.txt. Перевіримо статус.

Робимо комміт. Перевіряємо статус.

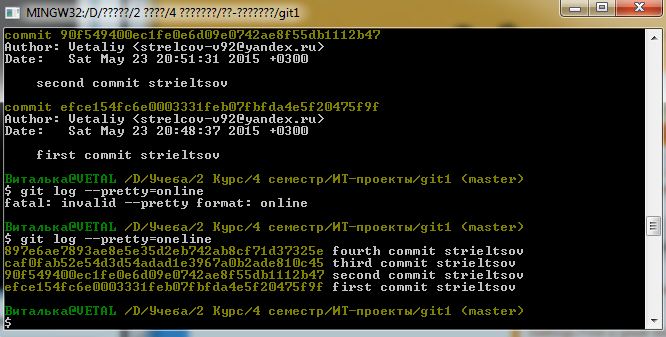


Можна зробити висновок, що дані в проіндексованому файлі закоммічені, а в непроіндексованому ні.

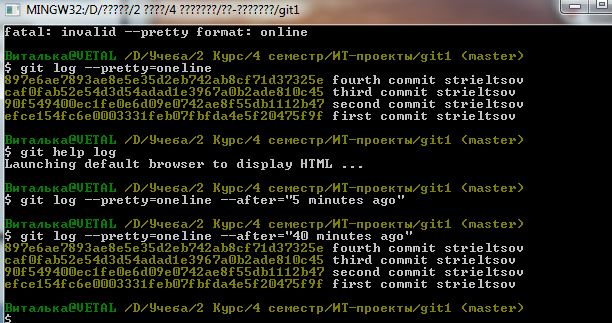
Виведемо історію проекту командою git log.



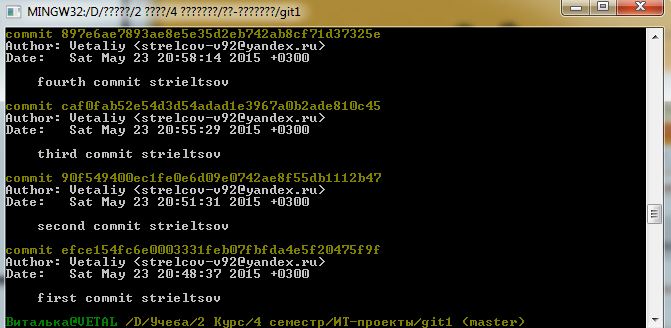
Введемо команду git log –pretty=oneline для перегляду історії коммітів.



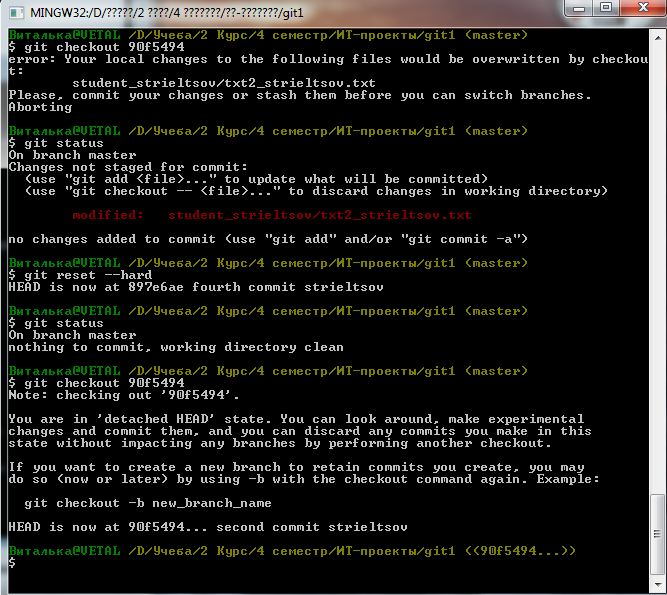
Використовуючи команду git log –pretty=oneline, налаштуємо відповідні параметри для виведення на екран коммітів за останні п’ять хвилин. Так як за останні п’ять хвилин коммітів не було, виведемо за останні 40 хвилин.



Виконаємо повернення стану файлів робочого каталогу на другий commit з часу початку лабораторної роботи. Використаємо команду git checkout. Спочатку виведемо інформацію про всі комміти командою git log.

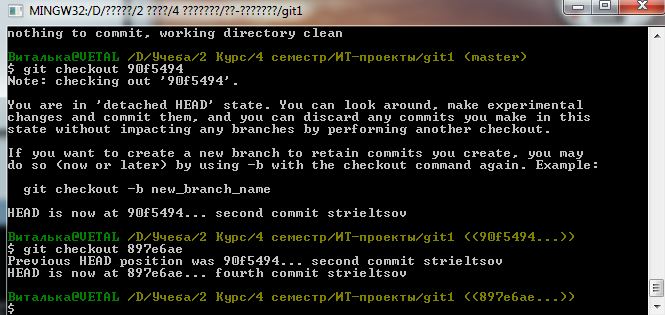


Використаємо команду git checkout.

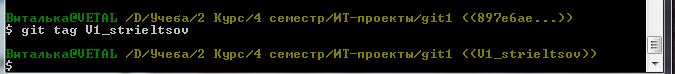


Видано повідомлення про помилку. Після перевірки статусу можна побачити, що файл txt2\_strieltsov.txt не закоммічений. Можна зробити його commit або зробити скидання до останнього комміта. Зробимо скидання командою git reset --hard. Перевіримо статус, помилки більше немає. Командою git checkout <hash> повернемо стан файлів на другий commit з часу початку роботи.

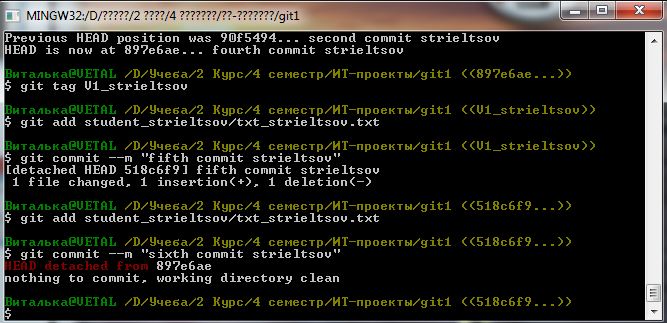
Зробимо повернення стану файлів до останнього комміту командою git checkout <hash>. Таким чином повернемося до останнього четвертого комміта.



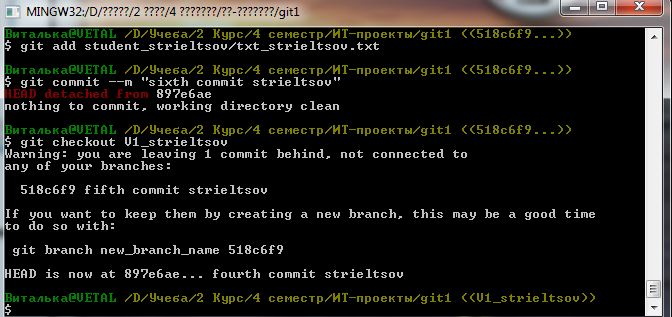
Створимо тег в поточному snapshot репозиторію. Назвемо його V1\_strieltsov.



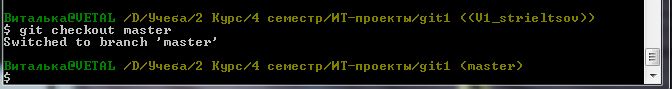
Виконаємо зміни файлу txt\_strieltsov.txt і два комміти цього файлу. Для цього спочатку змінимо його, потім проіндексуємо (команда git add), а потім зробимо commit (команда git commit -m).



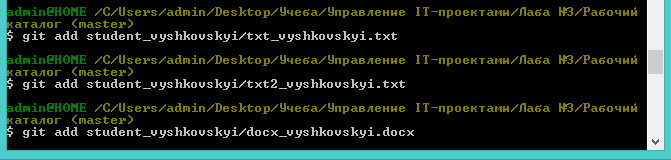
Продемонструємо повернення до стану репозиторію, поміченого тегом V1\_strieltsov командою git checkout.



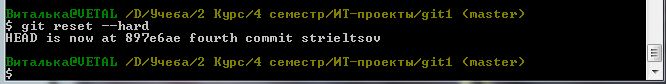
Виконаємо повернення до останнього стану master за допомогою команди git checkout.



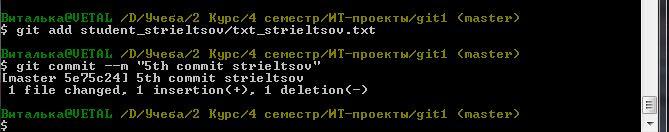
Виконаємо зміни поточного стану робочого каталогу, тобто фалів txt\_strieltsov.txt, txt2\_strieltsov.txt, docx\_strieltsov.docx. Проіндексуємо зміни цих файлів.



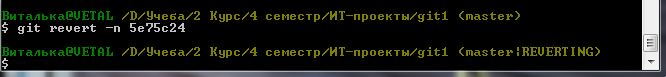
Командою git reset -hard повернемо файли в початковий стан.



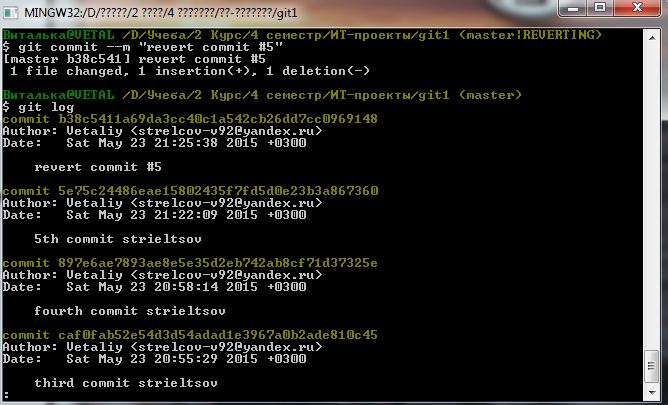
Зробимо зміни стану каталогу, а саме змінимо файл txt\_strieltsov.txt. Проіндексуємо його і зробимо його commit.



Виконаємо revert останнього комміта командою git revert –n.



Продемонструємо log командою git log.



**Висновки:** На даній лабораторній роботі я ознайомився з основами використання системи контролю версій Git.