МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГИЙ УПРАВЛІННЯ

Лабораторна робота № 4

з курсу «Проектування інформаційних систем»

«Знайомство з розподільною системою управління версіями GIT»

Виконав:

студент групи КН 36-б

Чуркін Радислав

Перевірив:

Копп А. М.

ХАРКІВ 2019

Мета роботи:

Знайомство з розподільною системою управління версіями GIT.

Завдання до лабораторної роботи:

Ознайомитись розподільною системою управління версіями GIT.

Хід виконання

1 Підготовка к виконанню роботи.

Створено на диску каталог та поміщено у нього підкаталог зі створеними даними у попередніх лабораторних роботах діаграмами структурного аналізу предметної області.

Установлено GIT.

2 Виконання роботи

До складу системи Git входить утиліта git config, яка дозволяє переглядати і встановлювати параметри, що контролюють всі аспекти роботи Git і його зовнішній вигляд. Запустимо Git Bash і введемо команди, як показано на рисунку 1.

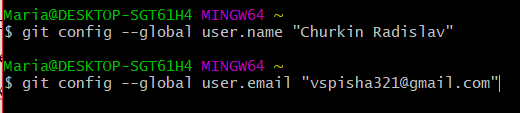


Рисунок 1 – Первісні налаштування

Створимо репозиторій у Git. Для цього перейдемо у проектний каталог та введемо у командній строчці команду, як показано на рисунку 2.

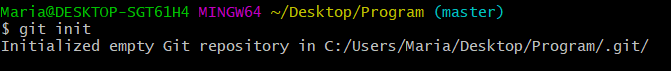


Рисунок 2 – Команда ініціалізації Git репозиторія

Визначимо у якому стані які файли знаходяться. Для цього впишемо команду, як показано на рисунку 3.

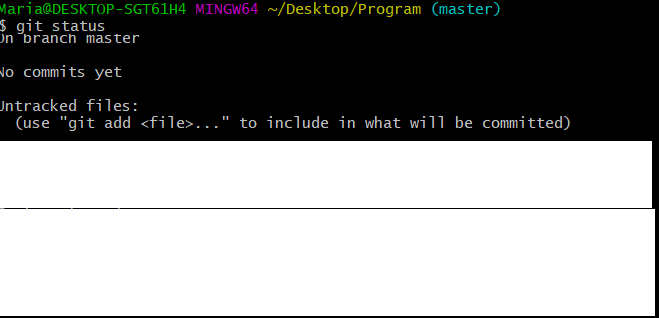


Рисунок 3 – Перевірка стану файлів

Для відстеження нового файлу використаємо команду git add. Щоб додати під версійність контроль один з файлів каталогу IS, виконаємо дії, як показано на рисунку 4.

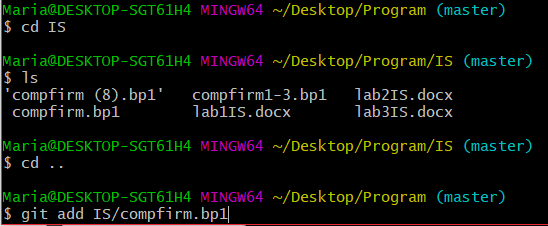


Рисунок 4 – Відстеження нового файла

Перевірка статусу файлу показано на рисунку 5.

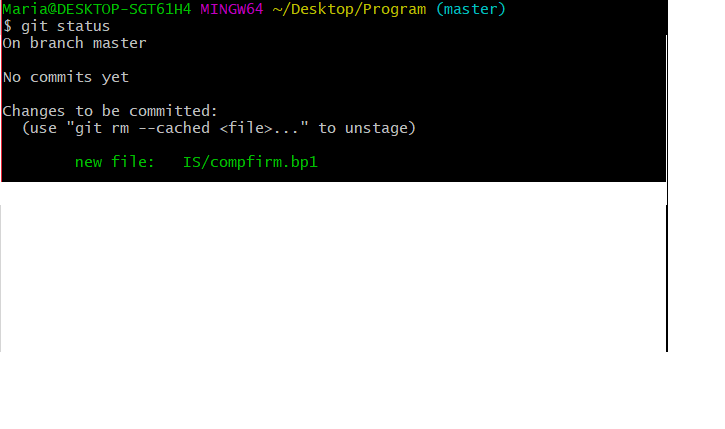


Рисунок 5 – Перевірка статусу файла

Додамо залишені файли до індексації та перевіримо їх статус (рисунок 6).

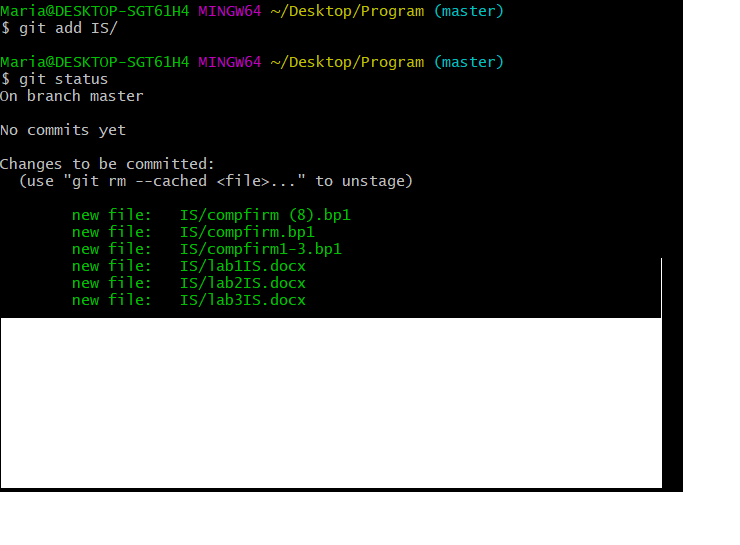


Рисунок 6 – Додавання залишених файлів та перевірка їх статусу

Якщо необхідно файл зробити таким, щоб його не відслідковувати, необхідно створити файл .gitignore з шаблоном цього файлу всередині.

Створимо в каталозі файл, що не потрібно відслідковувати (рисунок 7).

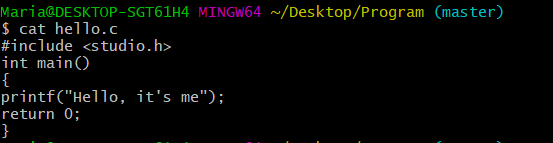


Рисунок 7 – Додання нового файлу

Створимо файл .gitignore (рисунок 8).



Рисунок 8 – Створення файлу .gitignore

Перевіримо коректність створення файлу за допомогою команди git status (рисунок 9).

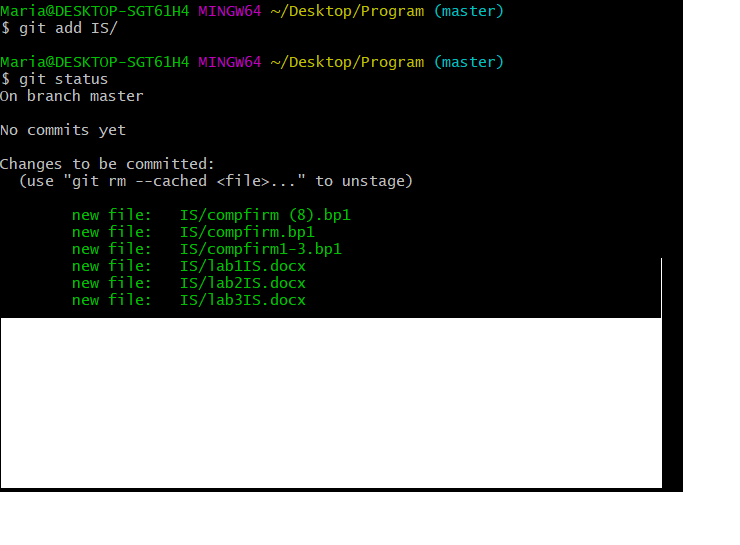


Рисунок 9 – Перевірка статусу файлу .gitignore

Підготуємо файл .gitignore до фіксації змін (рисунок 10).

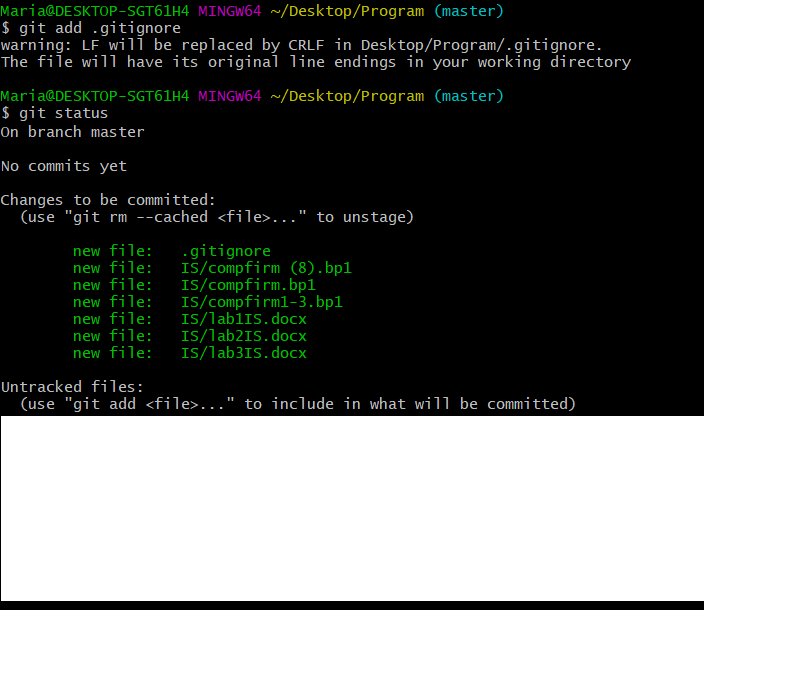


Рисунок 10 – Додавання файлу .gitignore до фіксації змін

Для того, щоб додати файли до фіксації використаємо команду git commit, вказуючи при цьому коментар до нього (рисунок 11).

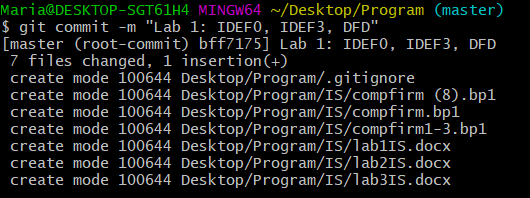


Рисунок 11 – Фіксація файлів

Видалимо файл з DFD-діаграмою з індексу, але залишимо його доступним у робочому каталозі для подальшого редагування чи виправлення виявлених помилок (рисунок 12).

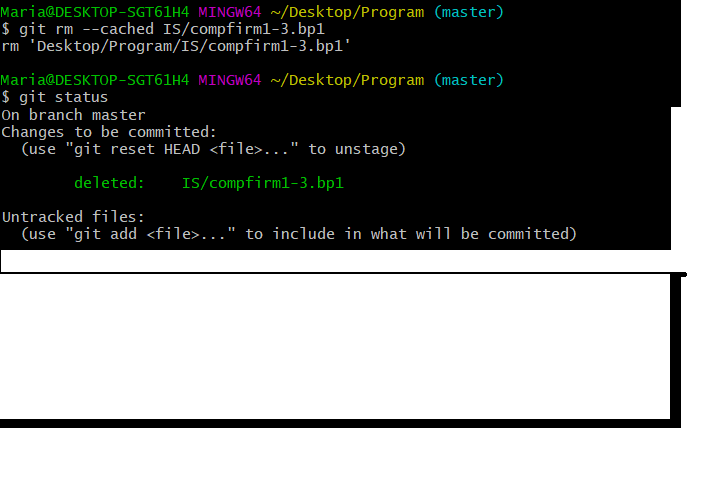


Рисунок 12 – Видалення з індексу файлу DFD-діаграми

Зафіксуємо видалення файлу коментарем на рисунку 13.



Рисунок 13 – Коментар до видалення

Переглянемо вивід команди git commit (рисунок 14).

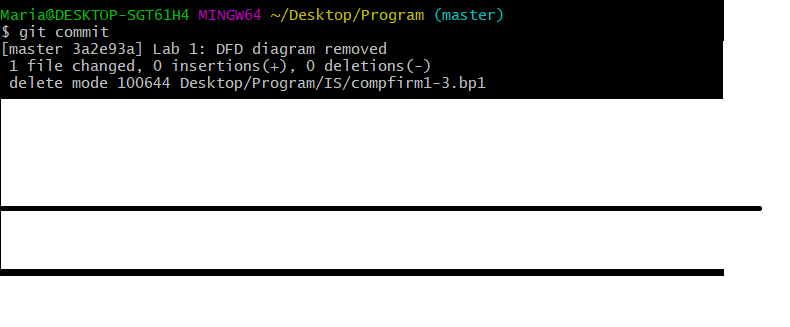


Рисунок 14 – Вивід команди git commit

Переконаємося, що «видалений» файл все ще доступний до редагування у робочому каталозі (рисунок 15).

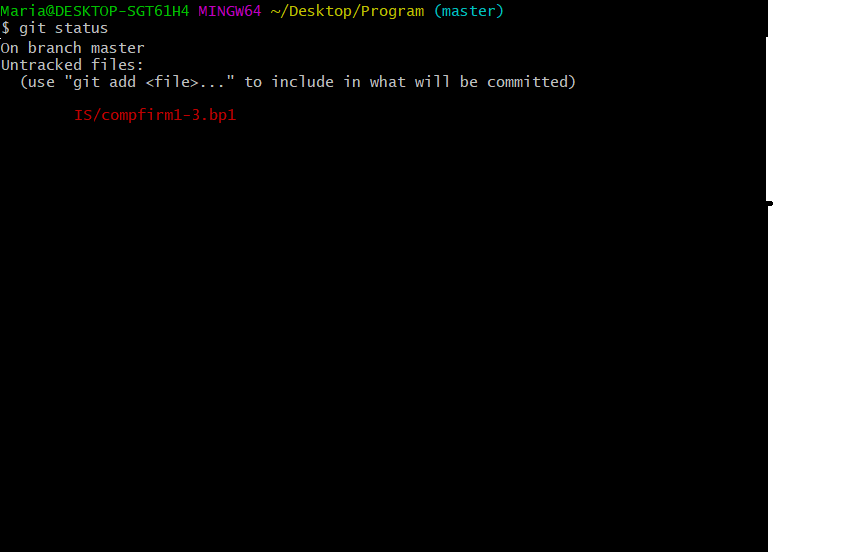


Рисунок 15 – Перевірка статусу «видаленого» файлу

Зафіксуємо зміни коментарем (рисунок 16).

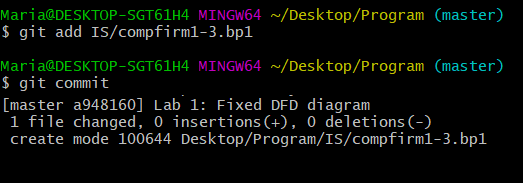


Рисунок 16 – Зафіксовані зміни

Для того, щоб перейменувати файл, зробимо дії, як на рисунку 17.

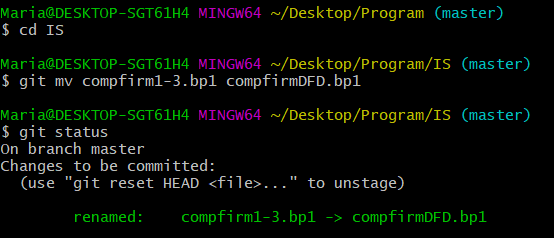


Рисунок 17 – Перейменування файлу

Історію коммітов показано на рисунку 18. Це необхідно, якщо репозиторій було клоновано вже з існуючого зі своєю історією коммітів.

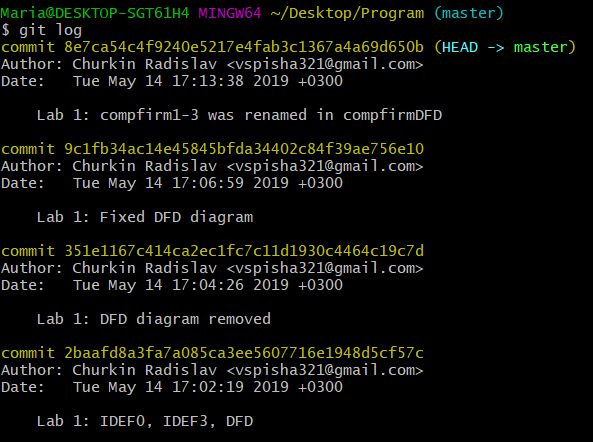


Рисунок 18 – Історія коммітів

Створимо файл як на рисунку 19.

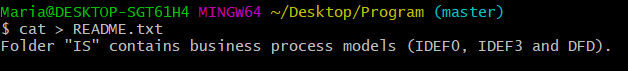


Рисунок 19 – Створення файлу README.txt

На рисунку 20 показано як додати до попереднього комміту файл, що було створено вище на рисунку 19.

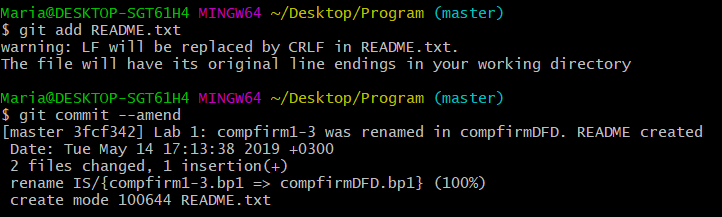


Рисунок 20 - Додання файлу README.txt до попереднього комміту

Для демонстрації того, як скасувати індексацію файлу, створимо два файли index.html і script.js в каталозі web (рисунок 21).

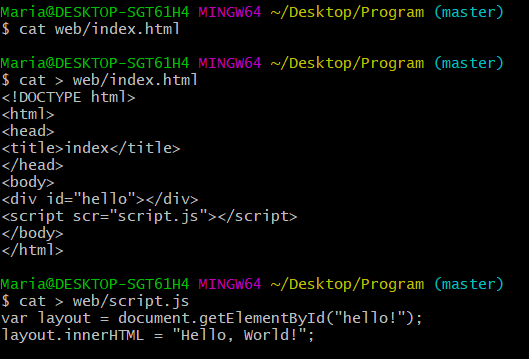


Рисунок 21 – Створення файлів у каталозі web

Проіндексуємо створені файли (рисунок 22).

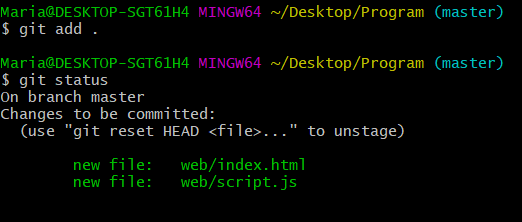


Рисунок 22 – Індексація створених файлів у каталозі web

Відмінимо індексацію одного з цих файлів (рисунок 23).

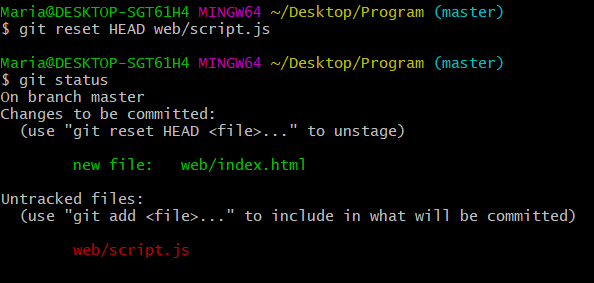


Рисунок 23 – Відміна індексації файлу

Тепер, коли другий створений файл (script.js) вважається не індексованим, виконаємо фіксацію першого файлу (index.html), додамо в область підготовлених файлів другий файл, і виконаємо його фіксацію в окремому комміті (рисунок 24).

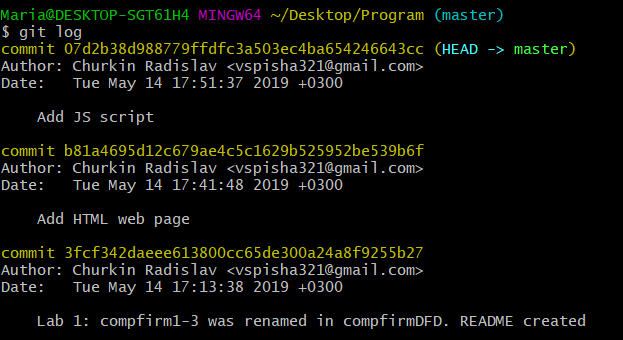


Рисунок 24 – Перевірка коммітів

Внесемо деякі зміни до файлу index.html (рисунок 25).

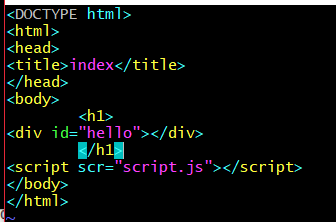


Рисунок 25 – Змінений index.html

Зробимо швидку відміну змін у файлі index.html, як показано на рисунку 26.

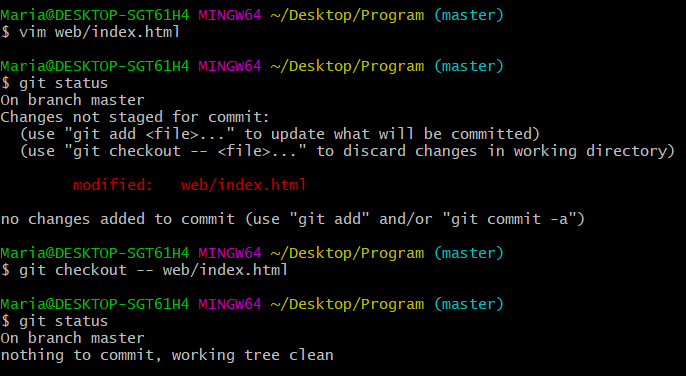


Рисунок 26 – Швидка відміна змін у файлі index.html

Перевіримо чи коректно пройшла відміна змін (рисунок 27).

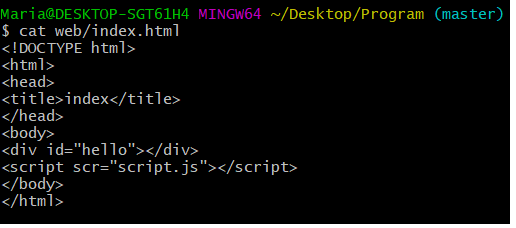


Рисунок 27 – Перевірка вмісту файлу index.html

Висновки:

Виконуючи лабораторну роботу було отримано наступні навички у використанні Git:

1. створення репозиторію;
2. створення файлів у репозиторії;
3. додавання файлів до індексації та створення коммітів;
4. зміна файлів;
5. редагування коммітів;
6. відміна коммітів за потреби.