

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
ІНСТИТУТ КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ, АВТОМАТИКИ ТА
МЕТРОЛОГІЇ

Кафедра СКС



Звіт

з лабораторної роботи №1

дисципліни: «Дослідження і проектування програмних систем»

Варіант 1

Виконав:
Ст.гр КІСК-11з
Алексєєнко Д.С.
Перевірів:
Шпіцер А.С.

Львів-2022

Завдання 1. Створити Git репозиторій із попередньою ініціалізацією README файлом. Склонувати проект. Навчитися працювати з системою контролю версії на прикладі Git. Створити файл README з інформацією про проект.

Теоретичні матеріали

Git — це розподілена система контролю версій. Система контролю версій дає змогу:

- Контролювати зміни, які внесені до програми
- Бачити історію всіх змін у програмі (хто, коли і що змінював), виконувати пошук у історії змін
- Значно зменшити ризик втратити змін через необережність чи помилку, а отже і зберегти виконану роботу
- Тримати кілька різних версій програми
- Ефективно працювати у команді з іншими розробниками над одним вихідним кодом програми

Переваги Git над іншими системами

Головна перевага Git – в тому, що він дуже швидкий і прозорий. Він зручний для нелінійної розробки і ефективний як для невеликих проектів, так і для великих систем з тисячами учасників. На відміну від Perforce, CVS та інших, Git зберігає знімки репозиторіїв, а не списки змін в файлах, і внаслідок цього працює набагато швидше.

Git – розподілена система. Якщо сервер з віддаленим репозиторієм вийде з ладу, можна відновити код з локальної копії. Якщо

локальна копія постраждає, можна завантажити код із сервера за кілька хвилин.

Як працює Git – найпопулярніша CVS

У Git знімки станів системи зберігаються в репозиторії коду у вигляді коммітів. Кожен комміт може належати одній або кільком гілкам. Основна гілка зазвичай носить назву master, в ній заведено зберігати повністю робочий код. Нову гілку можна створити з будь-якого комміту. Зміни разом з раніше розробленим кодом стануть ще однією версією програми.

Гілки можна об'єднувати (наприклад, щоб додати зміни в основну версію продукту). Є можливість перенесення змін з однієї гілки в іншу, обговорення, створення запитів на зміни в певній гілці тощо. Кожен розробник може створити для себе локальну копію всієї історії розробки і взаємодіяти з нею. Свої зміни він може відправити у віддалений репозиторій при наявності відповідних прав. Інші розробники отримають доступ до цих змін, завантаживши їх з сервера по протоколах SSH або HTTP.

Робота з системою контролю версій - це будні будь-якого розробника. Кожного дня розробники так чи інакше працюють з кодом під контролем версій, роблять зміни та об'єднують їх зі змінами інших розробників.

Хід роботи

- 1) Ініціалізація репозиторія (git init)
- 2) Створення гілки (git checkout -b <branch-name>)
- 3) Додавання змін (git add .)
- 4) Створення коміту (git commit -m "Коментар")
- 5) Створення зв'язку між локальним і віддаленим репозиторіями

(git remote add origin <remote-url>)

6) Надсилання змін до віддаленого репозиторію

(git push -u origin branch-name)

7) Створення тегу git tag <tagname> і внесення його у віддалений репозиторій git push origin <tagname>

Створюю репозиторій в GitHub

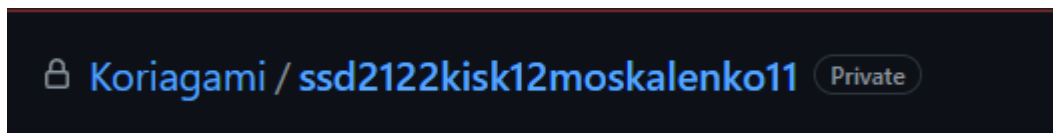
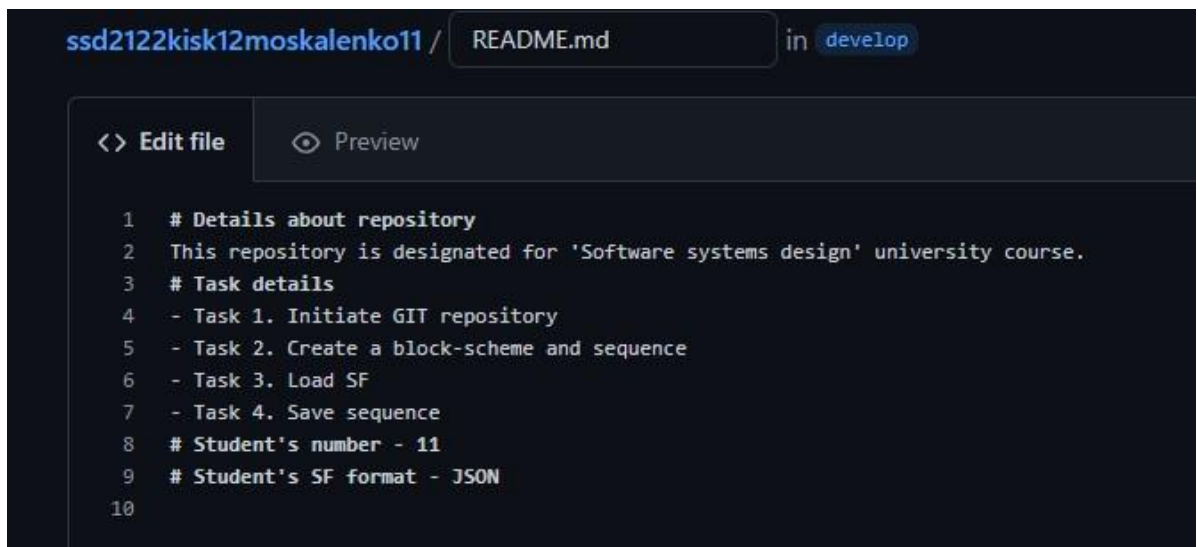


Рис. 1. Створення GitHub репозиторію Клонуємо проект

Name	Date modified	Type	Size
.git	21.06.2022 15:59	File folder	
README.md	21.06.2022 15:58	Markdown Source...	2 KB

Рис. 2. Клон проекту

Створюємо нову гілку від основної, та перемикаємося на неї. Редагуємо README файл згідно завданню.



```
ssd2122kisk12moskalenko11 / README.md in develop

<> Edit file Preview

1 # Details about repository
2 This repository is designated for 'Software systems design' university course.
3 # Task details
4 - Task 1. Initiate GIT repository
5 - Task 2. Create a block-scheme and sequence
6 - Task 3. Load SF
7 - Task 4. Save sequence
8 # Student's number - 11
9 # Student's SF format - JSON
10
```

Рис. 3. Редагований файл

Створюємо тег і додаємо його в origin

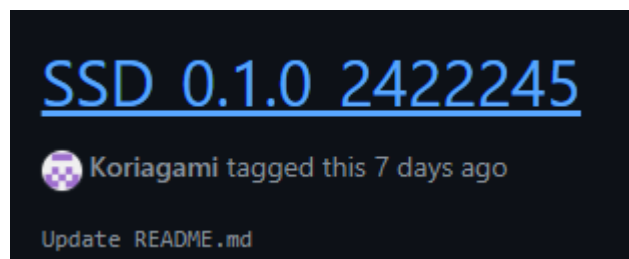


Рис. 4. Тег у гітхабі

Висновок: в ході виконання лабораторної роботи було створено Git репозиторій, завантажено початкову структуру проекту, а також вивчено алгоритм роботи з гілками в системі Git.