

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет "Львівська політехніка"
Інститут комп'ютерних наук та інформаційних технологій



Кафедра САП

Звіт

до лабораторної роботи №1

на тему: «Керування версіями за допомогою GIT»

З курсу: «Методи нечіткої логіки та еволюційні алгоритми при
автоматизованому проектуванні»

Виконав:
ст.гр. СПКс-11
Гуменний Л.О.

Прийняв:
Кривий Р.З.

ЛЬВІВ 2016

Мета роботи – закріпити на практиці можливості системи керування версіями.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Система керування версіями (англ. source code management, SCM) — програмний інструмент для керування версіями одиниці інформації: вихідного коду програми, скрипту, веб-сторінки, веб-сайту, 3D моделі, текстового документу тощо. Система керування версіями — це потужний інструмент, який дозволяє одночасно, без завад один одному, проводити роботу над груповими проектами.

Системи керування версіями зазвичай використовуються при розробці програмного забезпечення для відстеження, документування та контролю над поступовими змінами в електронних документах: у сирцевого коду застосунків, кресленнях, електронних моделях та інших документах, над змінами яких одночасно працюють декілька людей.

Кожна версія позначається унікальною цифрою чи літерою, зміни документу занотовуються. Зазвичай також зберігається автор зробленої зміни та її час. Інструменти для контролю версій входять до складу багатьох інтегрованих середовищ розробки.

Система керування версіями існують двох основних типів: з централізованим сховищем та розподіленням.

Централізовані системи контролю версії

Централізована система контролю версії (клієнт-серверна) — система, дані в якій зберігаються в єдиному «серверному» сховищі. Весь обмін файлами відбувається з використанням центрального сервера. Є можливість створення та роботи з локальними репозиторіями (робочими копіями).

Переваги:

1. Загальна нумерація версій;
2. Дані знаходяться на одному сервері;
3. Можлива реалізація функції блокування файлів;

4. Можливість керування доступом до файлів;

Недоліки: Потреба в мережевому з'єднанні для оновлення робочої копії чи збереження змін;

До таких систем відносять Subversion, Team Foundation Server.

Розподілені системи контролю версії

Розподілена система контролю версії (англ. Distributed Version Control System, DVCS) — система, яка використовує замість моделі клієнт-сервер, розподілену модель зберігання файлів. Така система не потребує сервера, адже всі файли знаходяться на кожному з комп'ютерів.

Переваги:

1. Кожний з розробників працює зі своїм власним репозиторієм;
2. Рішення щодо злиття гілок приймається керівникам проекту;
3. Немає потреби в мережевому з'єднанні;

Недоліки:

1. Немає можливості контролю доступу до файлів;
2. Відсутня загальна нумерація версії файла;
3. Значно більша кількість необхідного дискового простору;
4. Немає можливості блокування файлів;

До таких систем відносять Git, Mercurial, SVK, Monotone, Codeville, BitKeeper.

Система контролю дозволяє зберігати попередні версії файлів та завантажувати їх за потребою. Вона зберігає повну інформацію про версію кожного з файлів, а також повну структуру проекту на всіх стадіях розробки.

Місце зберігання даних файлів називають репозиторієм. В середині кожного з репозиторіїв можуть бути створені паралельні лінії розробки — гілки.

Гілки зазвичай використовують для зберігання експериментальних, незавершених(alpha, beta) та повністю робочих версій проекту(final).

Більшість систем контролю версії дозволяють кожному з об'єктів присвоювати теги, за допомогою яких можна формувати нові гілки та репозиторії.

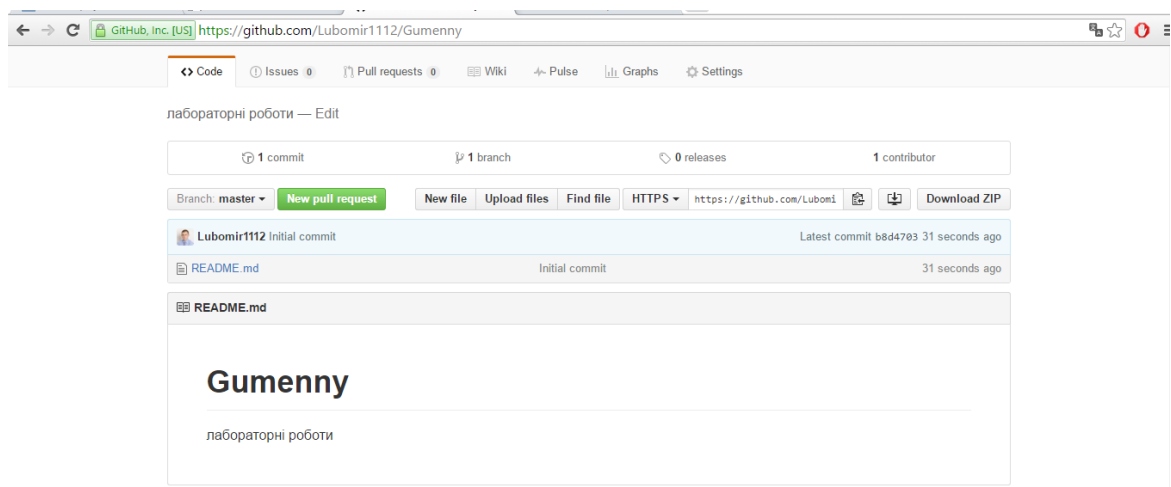
Використання системи контролю версії є необхідним для роботи над великими проектами, над якими одночасно працює велика кількість розробників. Системи контролю версії надають ряд додаткових можливостей: який рядок змінив); можливість його обмеження; записом змін в залежності від версії;

Індивідуальне завдання

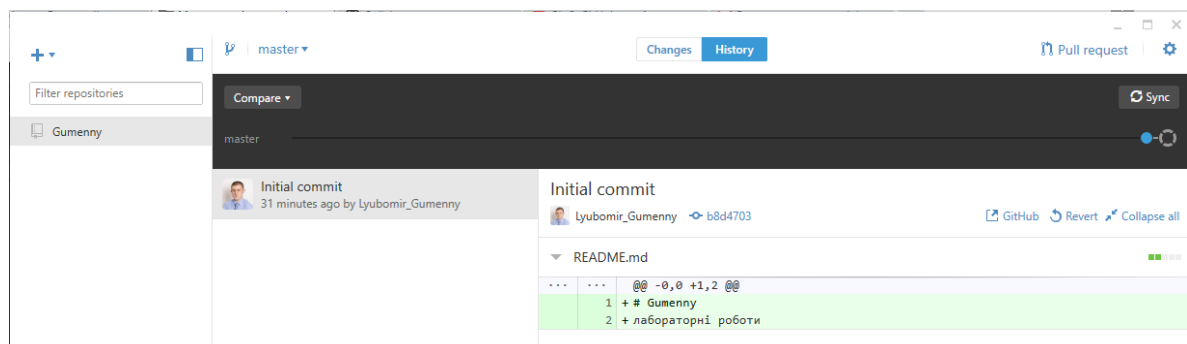
Закріпити на практиці можливості системи керування версіями.

Хід виконання

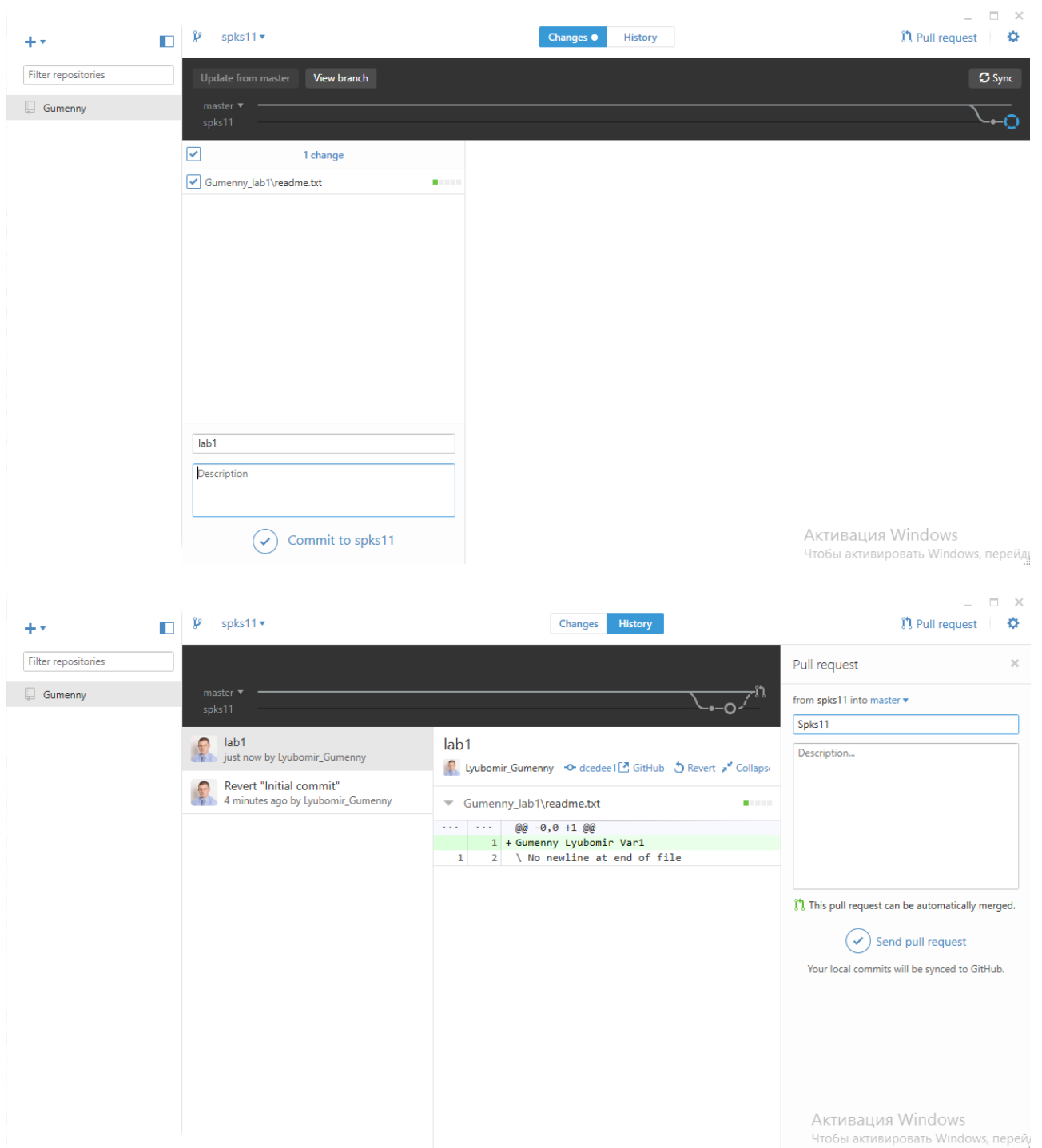
1. Зареєструвався у системі github, та увійшов під своїм логіном і створив репозиторій.

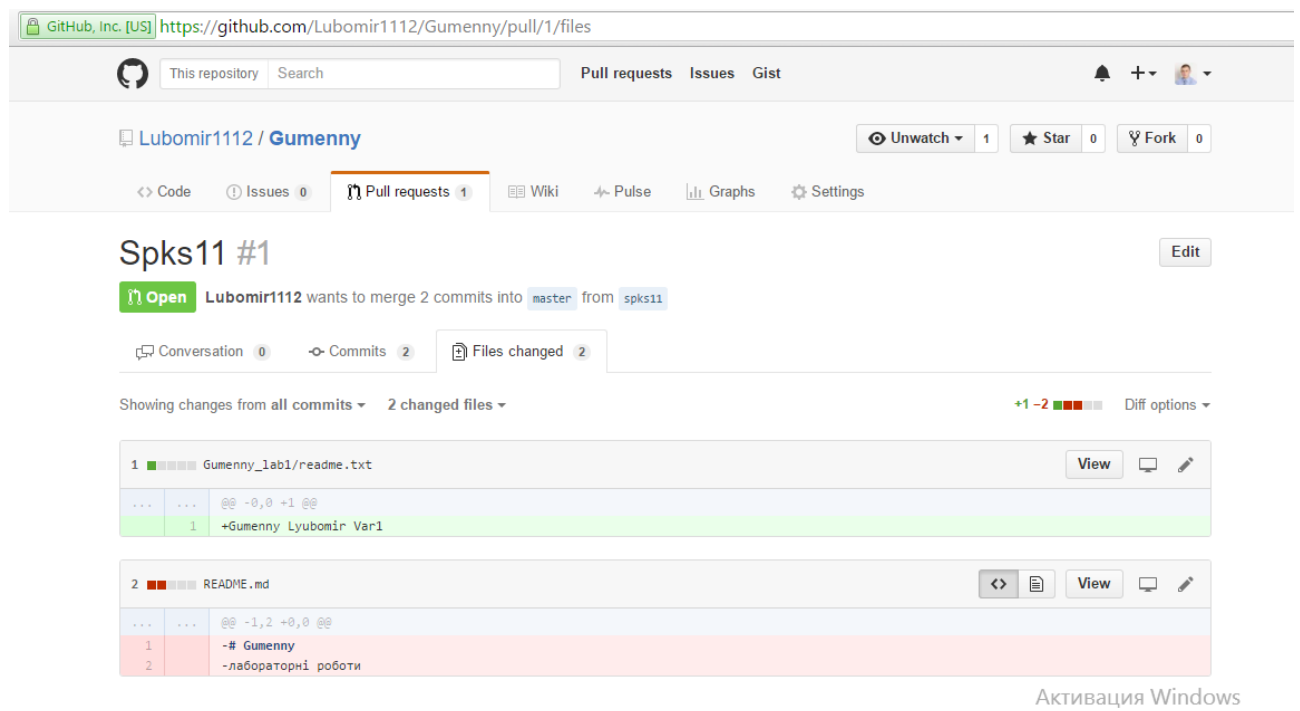


2. Завантаживши клієнт GitHub для Windows та встановив і увійшов в клієнт під своїм логіном.



3. Зробив відгалуження від основної гілки репозиторія
<https://github.com/Lubomir1112/Gumenny/>
4. В утвореній гілці, створити папку Gumenny_lab1
5. В утвореній папці, створити файл readme.txt в якому вказав прізвище, ім'я і номер варіанту.
6. Зробив commit і pull request на основну гілку.





Аналіз отриманих результатів лабораторної роботи та загальні висновки: Закріпив на практиці можливості системи керування версіями.