МіністерствоосвітиінаукиУкраїни

Івано**-**Франківськийнаціональнийтехнічнийуніверситетнафтиігазу

Кафедра програмного забезпечення

автоматизованих систем

Лабораторнаробота№ 2

Тема :«Робота з системою керування версіями Git»

Виконав:

ст. групи ПІ-14-1

Демчук Любомир Перевірила:

Бандура В. В.

м. Івано**-**Франківськ

2017

**Мета:** здобути необхідні знання та навички для роботи з системою керування версіями Git.

**Завдання:**

1. Зареєструватися на GitHub;

2. Завантажити ПЗ для роботи з Git;

3. Встановити на робочий комп’ютер;

4. Створити репозиторій на GitHub;

5. Створити локальний репозиторій;

6. Додати в локальний репозиторій файл;

7. Закомітити даний файл в локальний репозиторій;

8. Закинути зміни з локального репозиторію який ви створили в пункті 4;

9. Додати два файли в локальний репозиторій. Закомітайте і запушайте на GitHub;

10. Відновіться до стану, який був в пункті 8;

11. Зробіть клон репозиторія з GitHub в іншу папку на вашому комп’ютері;

12. Створіть нову гілку і додайте в неї новий файл;

13. Закомітайте зміни в новій branch. Синхронізуйте з репозиторієм на GitHub;

14. Об’єднайте створену гілку з основною;

15. Змініть файл, який ви додали в пункті 12 в основній branch;

16. Переключіться на гілку, яку ви створили в пункті 12 і також зробіть зміни в файлі, що додали в пункті 12.

17. Об’єднайте гілку з основною. Конфлікт файлу вирішіть в ручну.

**Короткі теоретичні відомості:**

Git — розподілена система керування версіями файлів та спільної роботи. Проект створив Лінус Торвальдс для управління розробкою ядра Linux, а сьогодні підтримується Джуніо Хамано (англ. Junio C. Hamano). Git є однією з найефективніших, надійних і високопродуктивних систем керування версіями, що надає гнучкі засоби нелінійної розробки, що базуються на відгалуженні і злитті гілок. Для забезпечення цілісності історії та стійкості до змін заднім числом використовуються криптографічні методи, також можлива прив'язка цифрових підписів розробників до тегів і комітів.

Прикладами проектів, що використовують Git, є ядро Linux, Android, LibreOffice, Cairo, GNU Core Utilities[2], Mesa 3D[3], Wine, багато проектів з X.org[4], XMMS2[5], GStreamer, Debian DragonFly BSD, Perl, Eclipse, GNOME, KDE, Qt, Ruby on Rails,PostgreSQL, VideoLAN, PHP, Beryl[6], One Laptop Per Child (OLPC), АБІС Koha, GNU LilyPond[7] та ELinks[8] і деякі дистрибутивиGNU/Linux (див. нижче).

Програма є вільною і випущена під ліцензією GNU GPL версії 2.

Система спроектована як набір програм, спеціально розроблених з врахуванням їхнього використання у скриптах. Це дозволяє зручно створювати спеціалізовані системи управління версіями на базі Git або користувацькі інтерфейси. Наприклад, Cogito[9] є саме таким прикладом фронтенда до репозиторіїв Git. А StGit[10] використовує Git для управління колекцією латок.

Система має ряд користувацьких інтерфейсів: наприклад, gitk та git-gui розповсюджуються з самим Git.

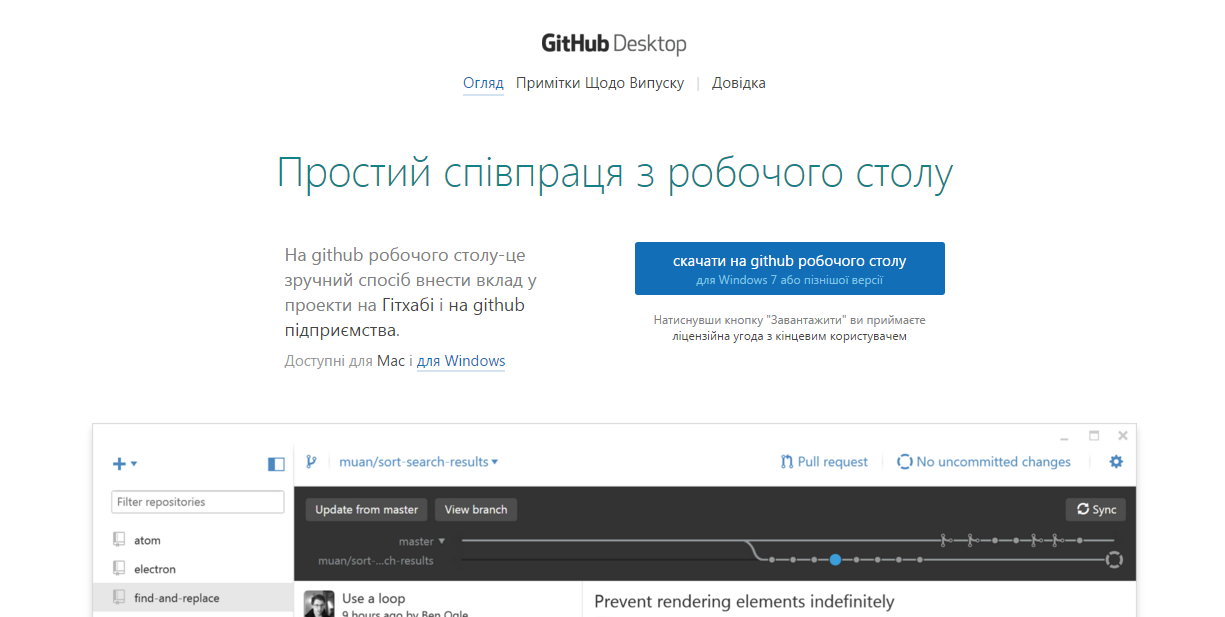
Віддалений доступ до репозиторіїв Git забезпечується git-демоном, SSH або HTTP сервером. TCP-сервіс git-daemon входить у дистрибутив Git і є разом з SSH найпоширенішим і надійним методом доступу. Метод доступу HTTP, незважаючи на ряд обмежень, дуже популярний в контрольованих мережах, тому що дозволяє використання існуючих конфігурацій мережевих фільтрів.

**Виконання:**

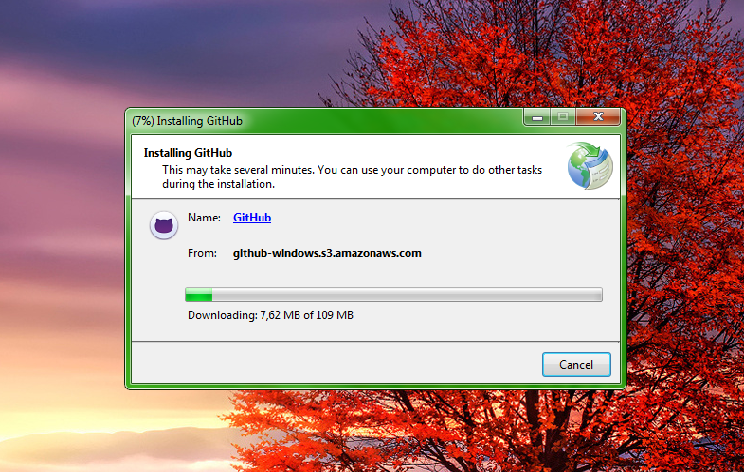
1. На GitHub зареєструвався раніше:



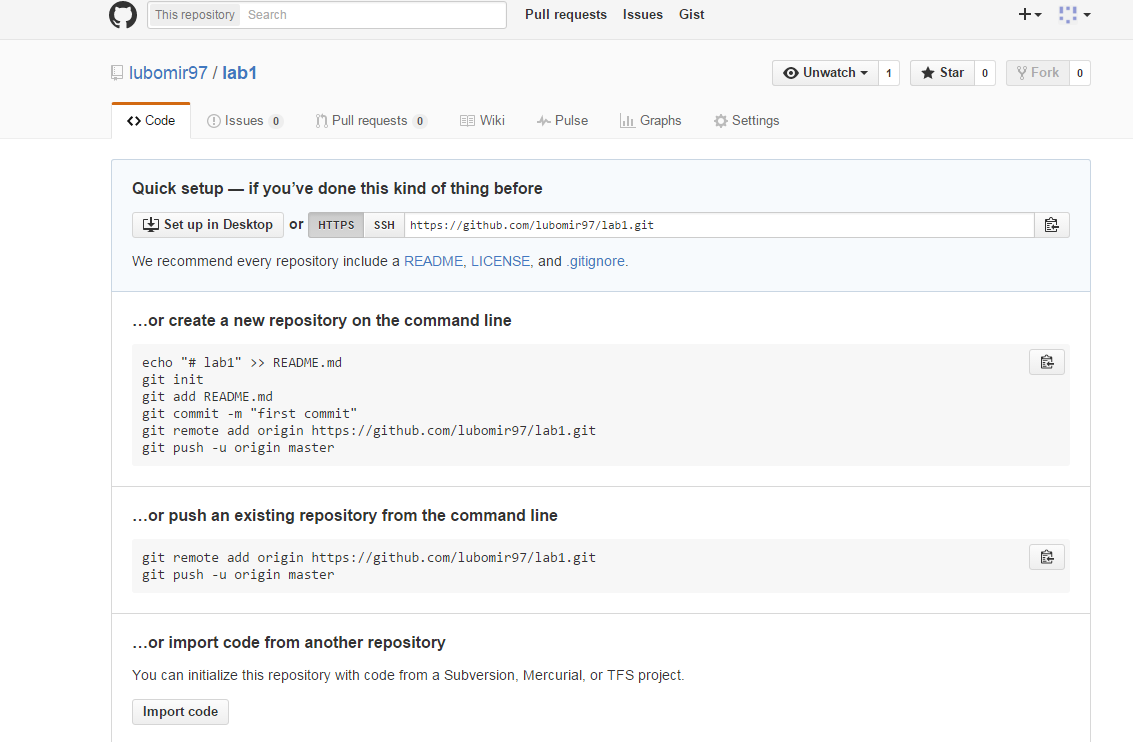
1. Завантаження ПЗ для роботи з Git:



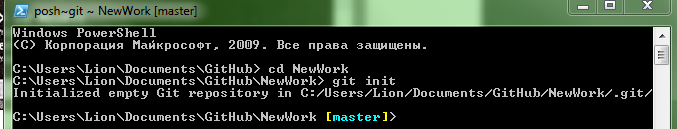
1. Встановлення на робочий комп’ютер:



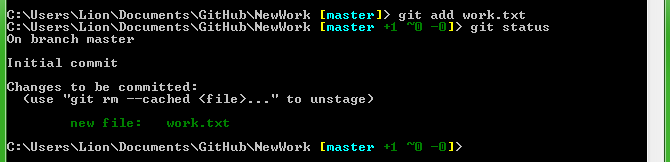
1. Створення репозиторія на GitHub:



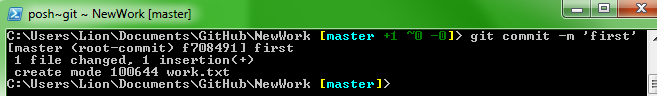
1. Створення локального репозиторію:



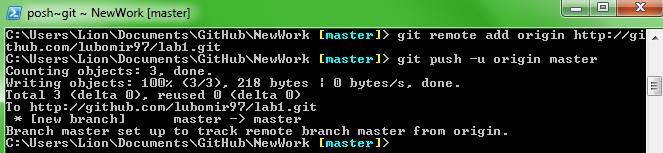
1. Додавання файла в локальний репозиторій:



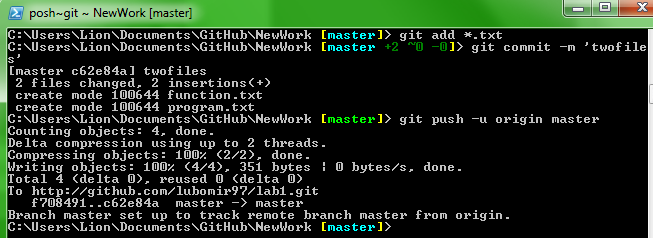
1. Закомічення файла в локальному репозиторії:



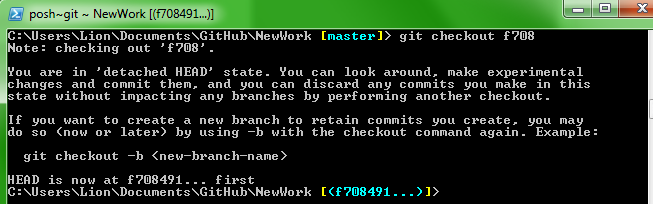
1. Закидання змін з лок. репозиторію в репозиторій, створений в пункті 4:



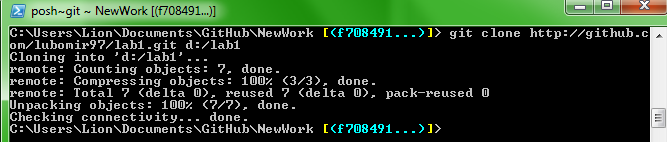
1. Додавання двох файлів в лок.репозиторій із закоміченням і пушанням на GitHub:



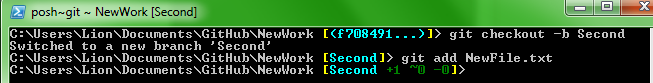
1. Відновлення до стану, який був в пункті 8:



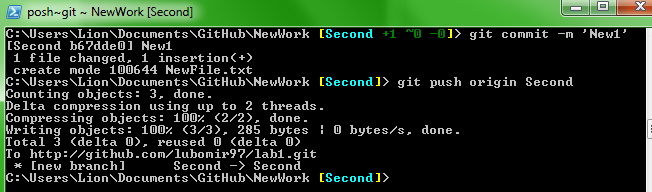
1. Створення клона репозиторія з GitHub в іншій папці на комп’ютері:



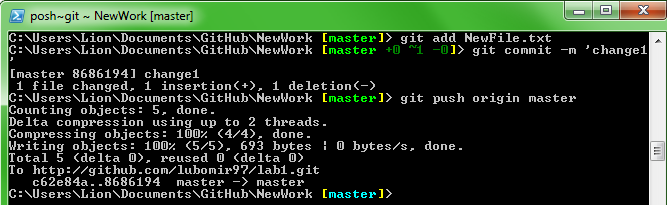
1. Створення нової гілки і додавання в неї нового файлу:



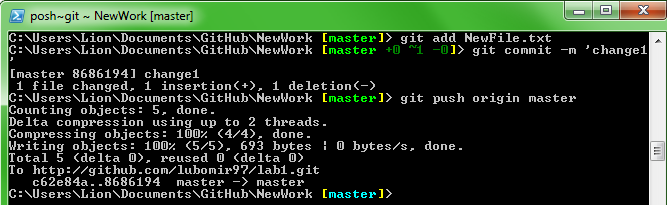
1. Закомічення змін в новій гілці і синхронізація з репозиторієм на GitHub:



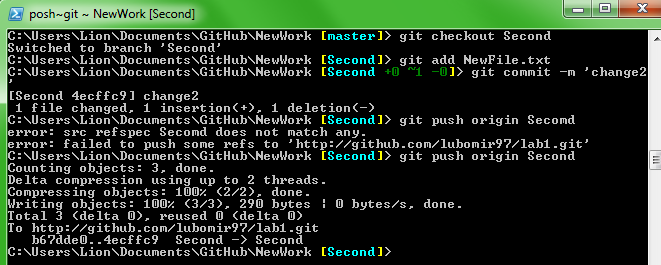
1. Об’єднання створеної гілки з основною:



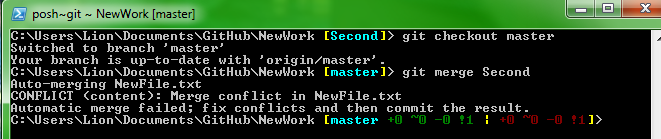
1. Змінення файл, який був доданий в пункті 12 в основній гілці:

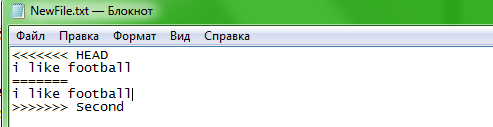


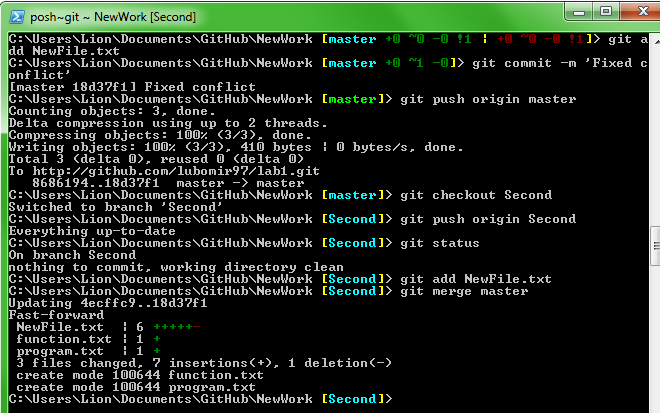
1. Переключення на гілку, яка була створена в пункті 12 а також змінення файлу, що був доданий в пункті 12:



1. Об’єднання створеної гілки з основною. Вирішення конфлікту в ручну:







**Висновок.** На цій лабораторній роботі я здобув необхідні знання та навички для роботи з системою керування версіями Git.