

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
УКРАЇНИ “КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ
ІНСТИТУТ”

Лабораторна робота №1
з дисципліни: «Проектуванню інформаційних систем»
“Системи контролю версій SVN, GIT”

Виконав:
студент групи ДА-71
Шнітковський Юрій

Київ 2020

Мета роботи

За допомогою системи контролю версій завантажити коди програми у репозиторій. Відтворити типовий цикл розробки програмного забезпечення з використанням системи контролю версій.

Завдання

1. Обрати безкоштовну систему репозиторія для системи контролю версіями, наприклад projectlocker, або інш.
2. Встановити клієнтське безкоштовне програмне забезпечення для роботи с системою контролю версій (GIT, SVN clients).
3. Протягом роботи над лабораторними роботами 2-6 використовувати систему контролю версіями.
4. Описати цикл розробки програмного забезпечення з використанням системи контролю версій.

Хід роботи

1. Для виконання лабораторної роботи було обрано онлайн репозиторій GitHub та систему контролю версій Git.
2. Було встановлено клієнт Git (Git GUI + Git Bash) для ОС Windows.
Його основні команди:
 - a. git clone - створює локальну копію віддаленого репозиторію.
 - b. git init – ініціалізація пустого локального репозиторію в директорії.
 - c. git status - відображення поточного стану репозиторію, які файли були змінені та які відслідковуються чи ні.
 - d. git checkout - перенесення вказівника HEAD, параметр -b вказує, що потрібно створити нову гілку.
 - e. git add - додає всі поточні зміни в індекс (увійдуть в наступний коміт).
 - f. git commit - зберігає всі індексовані зміни у новостворений коміт, переміщає поточний вказівник на цей коміт.
 - g. git push - публікує локальні зміни у віддалений репозиторій.
 - h. git pull – додає зміни з віддаленого репозиторію в гілку.
 - i. git fetch – завантажує об'єкти і гілки\теги з віддаленого репозиторію.
 - j. git merge – команда для злиття двох гілок.

3. Створення локальної копії та додавання файлу прикладу за допомогою консольних команд:

```
C:\Users\domin\Documents>git clone https://github.com/TarasVolovodenko/da71.git
Cloning into 'da71'...
remote: Enumerating objects: 121, done.
remote: Counting objects: 100% (121/121), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 326 (delta 29), reused 74 (delta 12), pack-reused 205
Receiving objects: 100% (326/326), 5.84 MiB | 1.16 MiB/s, done.
Resolving deltas: 100% (88/88), done.

C:\Users\domin\Documents\da71>git checkout -b shnitkovskiy_yurii
Switched to a new branch 'shnitkovskiy_yurii'

C:\Users\domin\Documents\da71>git branch
  master
* shnitkovskiy_yurii

C:\Users\domin\Documents\da71>git add Shnitkovskiy\README.md

C:\Users\domin\Documents\da71>git commit -m "Yurii S. Initial commit"
[shnitkovskiy_yurii b178f83] Yurii S. Initial commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Shnitkovskiy/README.md

C:\Users\domin\Documents\da71>git push origin shnitkovskiy_yurii
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 255 bytes | 255.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'shnitkovskiy_yurii' on GitHub by visiting:
remote:   https://github.com/TarasVolovodenko/da71/pull/new/shnitkovskiy_yurii
remote:
To https://github.com/TarasVolovodenko/da71.git
 * [new branch]      shnitkovskiy_yurii -> shnitkovskiy_yurii
```

4. Цикл розробки ПЗ з використанням системи контролю версій має наступний вигляд:

- a. Створення локальної копії репозиторію.
- b. Зміна локальної копії (файлів).
- c. Фіксування змін в локальних гілках.
- d. Об'єднання змін з інших гілок в локальному репозиторії.
- e. Публікація змін до віддаленого репозиторію.

Висновок

У ході виконання лабораторної роботи було виконано роботу з системою контролю версій Git. Базова частина, у виді консольних команд, має досить легке застосування, а також наявний GUI інтерфейс на ОС Windows, що дозволяє навіть новачку порівняно легко увійти в систему контролю версій. Система контролю версій дозволяє з легкістю контролювати зміни в репозиторії, надає можливість бачити всю історію змін та дозволяє, у разі потреби, повернутися до попередніх версій, а також ефективно працювати групам розробників, наприклад при розробці ядра ОС Linux, над якими працюють тисячі людей.