**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**„КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**

**НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС**

**„ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ”**

Лабораторна робота №1

з курсу «Проектування інформаційних систем» на тему:

“Системи контролю версій SVN, GIT.”

Виконав:

студент 4 курсу

групи ДА-62

Чернюк Андрій

Київ 2019

**Мета роботи:** за допомогою системи контролю версій завантажити коди програми у репозіторій. 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.

**Короткі теоретичні відомості:**

**Система керування версіями** — програмний інструмент для керування версіями одиниці інформації: вихідного коду програми, скрипту, веб-сторінки, веб-сайту, 3D-моделі, текстового документу тощо.

Система керування версіями — інструмент, який дозволяє одночасно, не заважаючи один одному, проводити роботу над груповими проектами.

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:

* Можливість створення різних варіантів одного документу;
* Документування всіх змін (коли ким було змінено/додано, хто який рядок змінив);
* Реалізує функцію контролю доступу користувачів до файлів. Є можливість його обмеження;
* Дозволяє створювати документацію проекту з поетапним записом змін в залежності від версії;
* Дозволяє давати пояснення до змін та документувати їх;

**Git** — розподілена система керування версіями файлів та спільної роботи. Проект створив Лінус Торвальдс для управління розробкою ядра Linux, а сьогодні підтримується Джуніо Хамано ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *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.

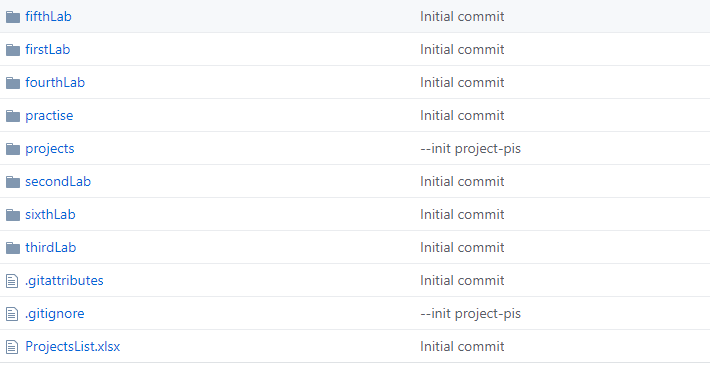
### 

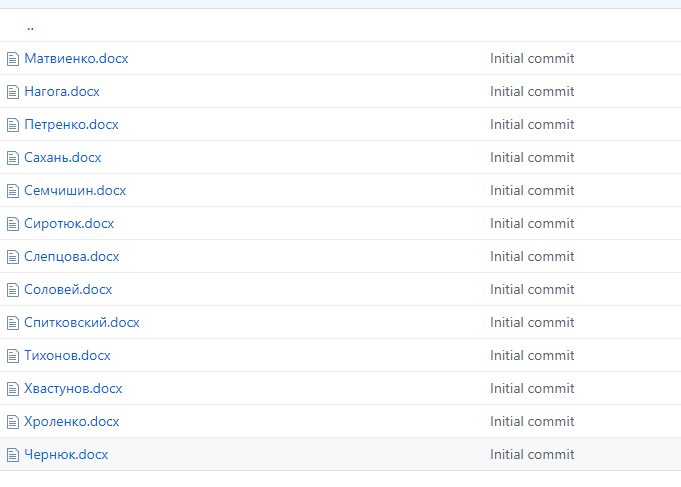
### Різниця SVN і GIT

1. GIT распределяется, а SVN - нет. Другими словами, если есть несколько разработчиков работающих с репозиторием у каждого на локальной машине будет ПОЛНАЯ копия этого репозитория. Разумеется есть и где-то и центральная машина, с которой можно клонировать репозиторий. Это напоминает SVN. Основной плюс в том, что если вдруг у вас нет доступа к интернету, сохраняется возможность работать с репозиторием. Потом только один раз сделать синхронизацию и все остальные разработчики получат поолную историю.
2. GIT сохраняет метаданные изменений, а SVN целые файлы. Это экономит место и время.
3. Система создания branches, versions и прочее в GIT и SVN отличаются значительно. В GIT проще переключатся с ветки на ветку, делать merge между ними.

**Основні команди Git:**

1. git commit
2. git cherry-pick
3. git pull (сливает клонируемые данные с локальными)
4. git clone
5. git remote (-v)
6. git fetch (не сливает клонируемые данные с локальными)
7. git branch (-f)
8. git add
9. git reset
10. git merge
11. git log
12. git show
13. git rebase
14. git status
15. git checkout
16. git push





**Висновки:** у ході роботи було набуто навичок роботи з системою контролю версій - Git, опановано основні команди clone, branch, checkout, push, commit.