Міністерство освіти та науки України

Національний технічний університет України

«Київський політехнічний інститут ім. І.Сікорського»

Навчально-науковий комплекс

«Інститут прикладного системного аналізу»

**Лабораторна робота №1**

З курсу «Проектування інформаційних систем»

Тема: « Системи контролю версій SVN, GIT»

*Варіант №17*

Виконала:

Студентка 3-го курсу

Групи ДА-62

Петренко М.М.

Київ 2019

**Мета роботи:** за допомогою системи контролю версій завантажити коди програми у репозиторій. Відтворити типовий цикл розробки програмного забезпечення з використанням системи контролю версій.

1. **Обрати безкоштовну систему репозиторія для системи контролю версіями, наприклад projectlocker, або інш.**

Для виконання лабораторних робіт було вибрано систему контролю версій Github.

Було створено репозиторій для половини студентів групи, у якому є дві гілки: dev and master. В гільці dev буде проводитись вся робота.

1. **Встановити клієнтське безкоштовне програмне забезпечення для роботи с системою контролю версій (GIT, SVN clients).**

Було встановлено системою контролю версій – GIT.

1. **Опис команд, які використовувалися протягом виконання роботи з системою контролю версіями**

### git clone

Клонирует репозиторий во вновь созданный каталог, создает ветки удаленного отслеживания для каждой ветви в клонированном репозитории

### git add

Команда git add добавляет содержимое рабочей директории в индекс (staging area) для последующего коммита. По умолчанию git commit использует лишь этот индекс, так что вы можете использовать git add для сборки слепка вашего следующего коммита.

### git status

Команда git status показывает состояния файлов в рабочей директории и индексе: какие файлы изменены, но не добавлены в индекс; какие ожидают коммита в индексе. Вдобавок к этому выводятся подсказки о том, как изменить состояние файлов.

### git commit

Команда git commit берёт все данные, добавленные в индекс с помощью git add, и сохраняет их слепок во внутренней базе данных, а затем сдвигает указатель текущей ветки на этот слепок.

### git reset

Команда git reset, как можно догадаться из названия, используется в основном для отмены изменений. Она изменяет указатель HEAD и, опционально, состояние индекса. Также эта команда может изменить файлы в рабочей директории при использовании параметра --hard, что может привести к потере наработок при неправильном использовании, так что убедитесь в серьёзности своих намерений прежде чем использовать его.

### git branch

Команда git branch — это своего рода “менеджер веток”. Она умеет перечислять ваши ветки, создавать новые, удалять и переименовывать их.

### git checkout

Команда git checkout используется для переключения веток и выгрузки их содержимого в рабочую директорию.

### git merge

Команда git merge используется для слияния одной или нескольких веток в текущую. Затем она устанавливает указатель текущей ветки на результирующий коммит.

### git pull

Команда git pull работает как комбинация команд git fetch и git merge, т.е. Git вначале забирает изменения из указанного удалённого репозитория, а затем пытается слить их с текущей веткой.

### git push

Команда git push используется для установления связи с удалённым репозиторием, вычисления локальных изменений отсутствующих в нём, и собственно их передачи в вышеупомянутый репозиторий. Этой команде нужно право на запись в репозиторий, поэтому она использует аутентификацию.



Преимущество использования svn:

* хранения истории изменений в централизованном хранилищи;
* копирование объектов с разветвлением истории;
* обеспечение переноса изменений между копиями объектов, даже полного слияния копий;
* обеспечение ветвления – создание и слияние ветвей;
* обеспечение меток;
* история изменений и копии объектов  хранятся в виде связанных простых копий;
* с помощью атомарных транзакций осуществляется фиксация изменений в хранилище;
* эффективная работа с текстовыми и двоичными файлами;
* много вариантов доступа к хранилищу;
* допустимость зеркалирования хранилища;
* возможные внутренние форматы хранилища: база данных или набор обычных файлов;
* наличие библиотек для популярных языков и возможность их подключения и использования;
* многоуровневая архитектура библиотек.

Преимущество использования git:

* Git имеет все  плюсы  использования VCS, что и в Subversion;
* имеет легкий процес шифрования “из коробки”;
* в случае приостановки сервера с главным репозиторием, можно делать коммиты в локальный сервер и ждать восстановления работы сервера;
* служебная информация хранится в корне репозитория;
* разработано много полезных утилит: Qgit, gitk, gitweb и другие. «Из коробки» есть импорт и экспорт в/из Subversion/CVS;
* система пользуется огромной популярностью и с ней работают много веб-разработчиков, поэтому помощь можно будет легко найти в любом случае, например, на специализированном форуме;
* с помощью Git можно размещать свой код на GitHub, BitBucket и Google