



*МІНІСТЕРСТВО ОСВІТИ І 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»

Виконав
студент IV курсу
групи ДА-61
Воловик Іван

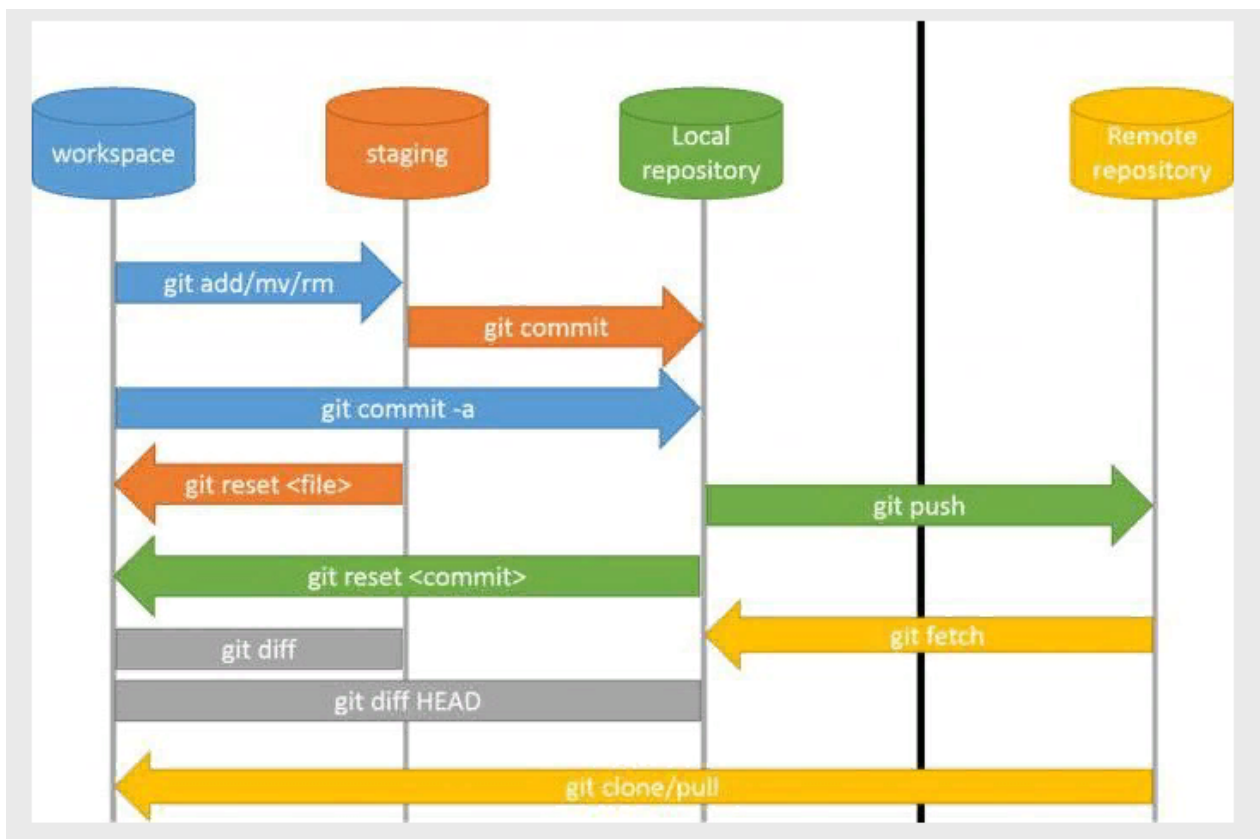
Київ – 2019

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

Хід роботи:

Для виконання лабораторних робіт було вибрано систему контролю версій Github. Було створено репозиторій для половини студентів групи, у якому є дві гілки: dev та master. В гілці dev буде проводитись вся робота. Було встановлено системою контролю версій – GIT.

Опис команд, які використовувалися протягом виконання роботи:



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 слепок 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 отменяет изменения, перенося ссылку на ветку назад, на более старый коммит. Это своего рода "переписывание истории"; `git reset` перенесёт ветку назад, как будто некоторых коммитов вовсе и не было.

git revert

Создает отмену коммитов которыми можно делится.

git rebase

При ребейзе Git, по сути, копирует набор коммитов и переносит их в другое место.

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 cherry-pick

Это очень простой и прямолинейный способ сказать, что ты хочешь копировать несколько коммитов на место, где сейчас находишься (HEAD).

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, и их передачи в репозиторий

SVN проти Git: у чому різниця між Git та SVN?

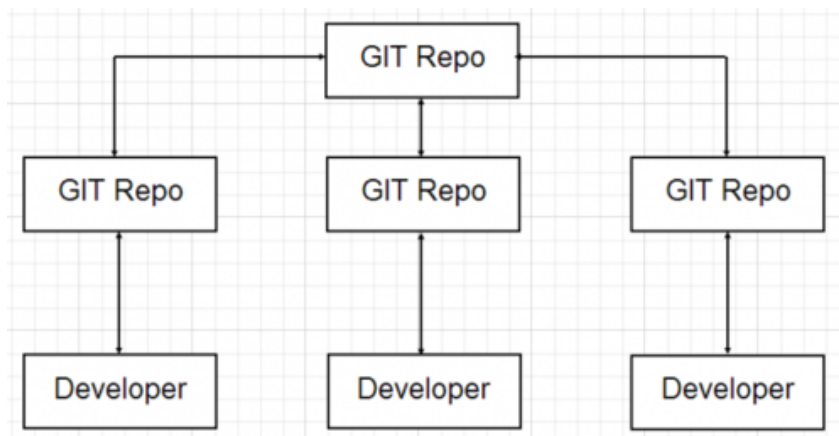
Хоча Git і SVN - це системи управління версіями корпоративних версій (VCS), які допомагають у процесі роботи та управління проектами при кодуванні, вони мають свої відмінності. Різниця між системами керування версіями Git і SVN полягає в тому, що Git - це система управління розподіленою версією, тоді як SVN - це централізована система управління версіями. Git використовує кілька сховищ, включаючи централізоване сховище та сервер, а також деякі локальні сховища. SVN не має централізованого сховища чи сервера.

Що таке GIT?

Git - це система управління розподіленою версією - це означає, що коли ви робите клон git (+ URL вашого сховища), то, що ви насправді отримуєте, є повною копією всієї вашої історії цього проекту.

Які переваги Git?

Git має місце постановки. Це просто означає, що якщо ви внесли 100 нових змін у свій код, ви можете розбити ці 100 змін на 10 чи 20 чи більше комісій, кожен із власних коментарів та власного детального пояснення того, що щойно сталося! Ви не тільки можете поетапно виконати свої зобов'язання, щоб логічно відобразити зміни, які були внесені, але ви також можете виконати постановку патча, яка запитає вас, чи хочете ви. Ви б використовували постановку виправлення, якщо ви та колега працюєте над одним файлом, і ви хочете виконати лише певну функцію, над якою працювали. Ви робите патч Git, використовуючи "git add -p"

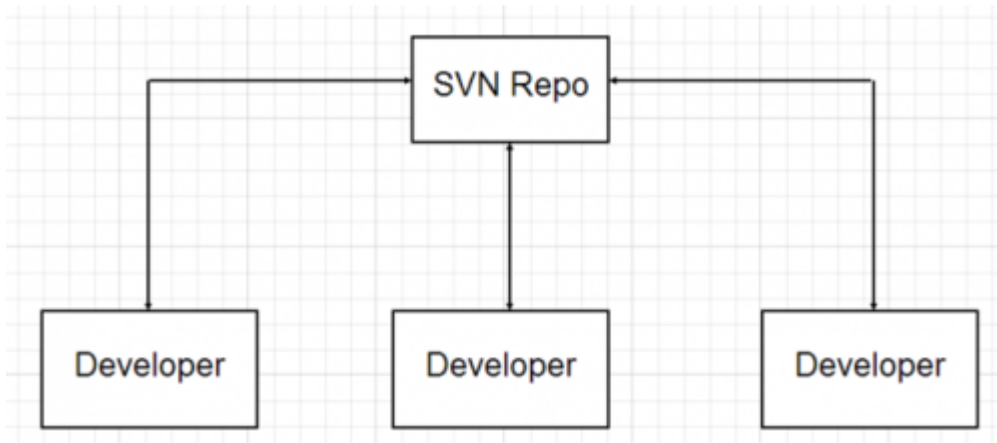


Що таке SVN?

Subversion (SVN) може бути однією з найбільш відомих централізованих систем управління версіями. У Subversion або SVN ви перевіряєте одну версію сховища. За допомогою SVN ваші дані зберігаються на центральному сервері. Наявність усієї історії у вашому локальному сховищі означає, що навіть коли ви не підключені до Інтернету, ви все одно можете робити коміти, розрізники, журнали, гілки, об'єднання, анотації до файлів тощо.

Які переваги SVN?

SVN має одне центральне сховище - це полегшує менеджерам більше підходу зверху вниз до контролю, безпеки, дозволів, дзеркал та скидів. Крім того, багато хто каже, що SVN легше використовувати, ніж Git. Наприклад, легше створити нову функцію. З Git потрібно зробити додатковий крок для створення нової функції. Інші кажуть, що спосіб налаштування SVN призводить до більшої стабільності магістралі, а наявність на центральному сервері для деяких відчуває себе більш контрольованою та захищеною.



SVN vs Git

SVN дозволяє перевіряти лише субдерева (або гілки), тоді як Git вимагає від вас перевірити весь сховище як одиницю. Це відбувається тому, що у кожній з ваших папок є `.svn`, а у git є лише одна `.git` у батьківському каталозі верхнього рівня

Чи Git кращий за SVN?

Git і SVN - це кожна життєздатна система управління робочим процесом і версіями, але з різних причин. У Git можуть виникнути більші труднощі при стисненні та зберіганні двійкових файлів, у той час як SVN не так багато. Однак, багато хто стверджує, що Git кращий за SVN, оскільки він працює добре навіть для розробників, які не завжди підключені до головного сховища, оскільки це доступно в режимі офлайн. Підтримка розгалуження та об'єднання також вважається вищою для Git. Що стосується зберігання дискового простору, він майже близький до рівнів між SVN та Git сховищами. Git також трохи новіший ніж SVN.

Мій проект: Система керування комп'ютером за допомогою розпізнавання жестів

Висновки: в даній лабораторній роботі ми познайомились с git та помістили свої роботи у папку dev.