МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ КОМП’ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра економічної кібернетики

Комплексне практичне індивідуальне завдання

з дисципліни «Конструювання ПЗ»

на тему:

«Переваги і недоліки Git »

Виконав :

студент групи ПЗС-32

Процюк О.М.

Перевірив:

Сирник О. Й.

Тернопіль

2015

Зміст

1.Вступ…………………………………………………………………………….….2

2 .Git…………………………………………………………………………………..3

3. Переваги і недоліки Git…………………………………………………………...4

4.Висновок ………………………………………………………………………..….7

# 5. Список використаних джерел……………………………………………………8

Вступ

Переваги та недоліки git в порівнянні з централізованими системами управління версіями (такими як, наприклад, Subversion) типові для будь розподіленої системи і описані в статті «Система управління версіями». Якщо ж порівнювати git з «спорідненими» їй розподіленими системами, можна відзначити, що git спочатку ідеологічно орієнтований на роботу із змінами, а не з файлами, «одиницею обробки» для нього є набір змін, або патч. Ця особливість простежується як у структурі самої системи (зокрема - в структурі репозиторію), так і в принципах побудови команд; вона відображається на продуктивності системи в різних варіантах її використання і на достоїнства і недоліки git в порівнянні з іншими DVCS..

Git

Git – розподілена система керування версіями файлів та спільної роботи. Проект створив ЛінусТорвальдс для управління розробкою ядра Linux, а сьогодні підтримується ДжуніоХамано. 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 CoreUtilities, Mesa 3D, Wine, багато проектів з X.org, XMMS2, GStreamer, DebianDragonFly BSD, Perl, Eclipse, GNOME, KDE, Qt, RubyonRails, PostgreSQL, VideoLAN, PHP, Beryl, OneLaptopPerChild (OLPC), АБІС Koha, GNU LilyPond та ELinks і деякі дистрибутиви GNU/Linux.

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 є саме таким прикладом фронтендадо репозиторіївGit. А StGit використовує Git для управління колекцією латок.

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

Віддалений доступ до репозиторіївGit 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.

Переваги і недоліки Git

Часто звані переваги git перед іншими DVCS:

Висока продуктивність.

Розвинені засоби інтеграції з іншими VCS, зокрема, до CVS, SVN і Mercurial. Крім різноспрямованих конвертерів репозиторіїв, наявні в комплекті програмні засоби дозволяють розробникам використовувати git при розміщенні центрального репозиторію в SVN або CVS, крім того, git може імітувати cvs-сервер, забезпечуючи роботу через клієнтські програми і підтримку в середовищах розробки, спеціально не підтримують git.

Продумана система команд, що дозволяє зручно вбудовувати git в скрипти.

Якісний веб-інтерфейс «з коробки».

Репозиторії git можуть поширюватися і оновлюватися загальносистемними файловими утилітами архівації та оновлення, такими як rsync, завдяки тому, що фіксації змін і синхронізації не міняють існуючі файли з даними, а тільки додають нові (за винятком деяких службових файлів, які можуть бути автоматично оновлені за допомогою наявних у складі системи утиліт). Для роздачі репозиторію по мережі достатньо будь-якого веб-сервера.

У числі недоліків git зазвичай називають:

Відсутність наскрізної нумерації коммітов монотонно безупинно зростаючими цілими числами. У багатьох проектах використовується автоматичні отримання номера цієї версії (наприклад, командою svnversion), побудова .H файлу на основі цього числа, і далі його використання при створенні штампа версії виконуваного файлу, деяких вшитих в нього рядків і так далі.

Відсутність переносимої на інші операційні системи підтримки шляхів в кодуванні Unicode в Microsoft Windows (для версій msysgit до 1.8.1). Якщо шлях містить символи, відмінні від ANSI, то їх підтримка з командного рядка вимагає специфічних налаштувань, які не гарантують правильного відображення файлових імен при користуванні тим же репозиторієм з інших ОС. Одним із способів вирішення проблеми для git 1.7 є використання спеціально пропатченний консольного клієнта. Інший варіант - використання графічних утиліт, що працюють безпосередньо через API, таких як TortoiseGit.

Деяка незручність для користувачів, перехідних з інших VCS. Команди git, орієнтовані на набори змін, а не на файли, можуть викликати подив у користувачів, які звикли до файл-орієнтованим VCS, таким як SVN. Наприклад, команда «add», яка в більшості систем управління версіями виробляє додавання файлу до проекту, в git підготовляє до фіксації зроблені у файлах зміни. При цьому зберігається трохи патч, що описує зміни, а нова версія цільового файлу.

Використання для ідентифікації ревізій хешів SHA1, що призводить до необхідності оперувати довгими рядками замість коротких номерів версій, як у багатьох інших системах (хоча в командах допускається використання неповних хеш-рядків).

Великі накладні витрати при роботі з проектами, в яких робляться численні незв'язані між собою зміни файлів. При роботі в такому режимі розміри наборів змін стають досить великі й відбувається швидке зростання обсягу репозиторіїв.

Великі витрати часу, у порівнянні з файл-орієнтованими системами, на формування історії конкретного файлу, історії правок конкретного користувача, пошуку змін, що відносяться до заданого місця певного файлу.

Відсутність окремої команди перейменування / переміщення файлу, яка відображалася б в історії як відповідне єдине дію. Існуючий скрипт git mv фактично виконує перейменування, копіювання файлу і видалення його на старому місці, що вимагає спеціального аналізу для визначення, що насправді файл був просто перенесений (цей аналіз виконується автоматично командами перегляду історії). Однак, враховуючи той факт, що наявність спеціальної команди для перейменування / переміщення файлів технічно не змушує користувача використати саме її (і, як наслідок, в цьому випадку можливі розриви в історії), поведінка git може вважатися перевагою.

Система працює тільки з файлами і їх вмістом, і не відслідковує порожні каталоги.

# 

# Висновки

У ряді публікацій, відноситься переважно до 2005-2008 років можна зустріти також нарікання щодо документації git, відсутності зручної windows-версії і зручних графічних клієнтів. В даний час ця критика неактуальна: існує версія git на основі MinGW («рідна» збірка під Windows), і кілька високоякісних графічних клієнтів для різних операційних систем, зокрема, під Windows є клієнт TortoiseGit, ідеологічно дуже близький до широко поширеній TortoiseSVN - клієнтові SVN, вбудовується в оболонку Windows.

По даній тематиці можна зробити висновок, що кожен клієнт має право вибору, користуватися даною системою чи ні. Як і кожний проект має свої плюси і мінуси. На мою думку git- є одна з зручніших [розподілених систем керування версіями](https://uk.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B7%D0%BF%D0%BE%D0%B4%D1%96%D0%BB%D0%B5%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D1%96%D1%8F%D0%BC%D0%B8) [файлів](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB), гнучка і мобільна в роботі.

# Список використаних джерел

1. Git– [Електронний ресурс] – <https://ru.wikipedia.org/wiki/Git>
2. Система керуванняверсіями– [Електронний ресурс] – <https://uk.wikipedia.org/wiki/Система_керування_версіями>
3. Система контролю версій – [Електронний ресурс] – <http://gamersoul.ru/sistema-kontrolya-versij-v-igrax.html>
4. Unity(Рушій гри) – [Електронний ресурс] – https://uk.wikipedia.org/wiki/Unity\_(рушій\_гри)