



Введение в системы контроля версий. Введение в GIT.

Кирилл Юхин

MIPT iLab

19 сентября, 2017

Система контроля версий Version Control System

VCS – система для хранения и управления различными версиями проекта.

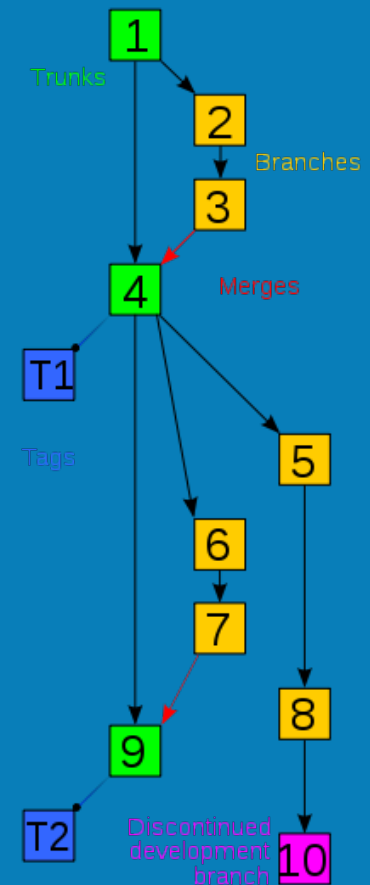
Распространённые системы контроля версий:

- Централизованные – SVN (2000), CVS (1986)
- Распределённые – GIT (2005), Mercurial (2005), Bazaar (2005), BitKeeper (1998).

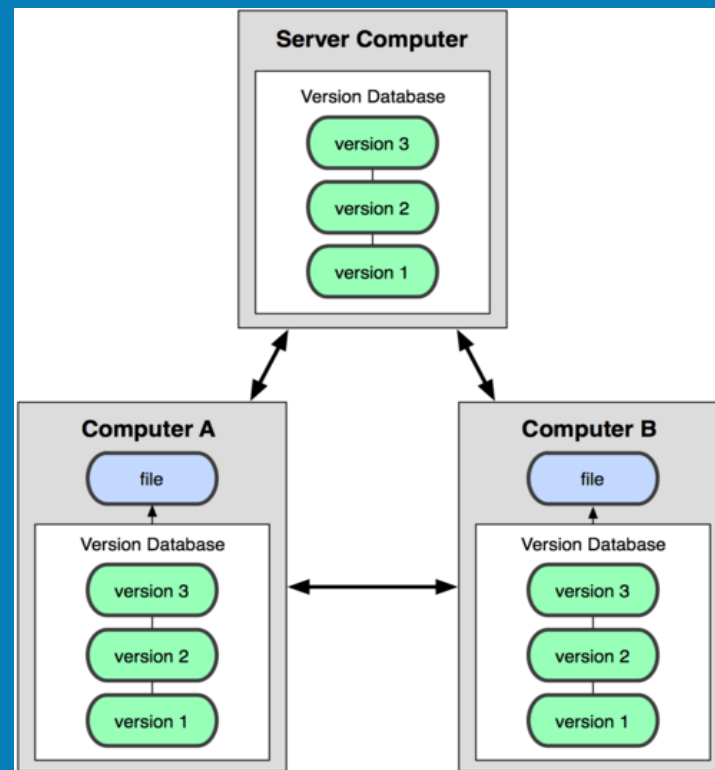
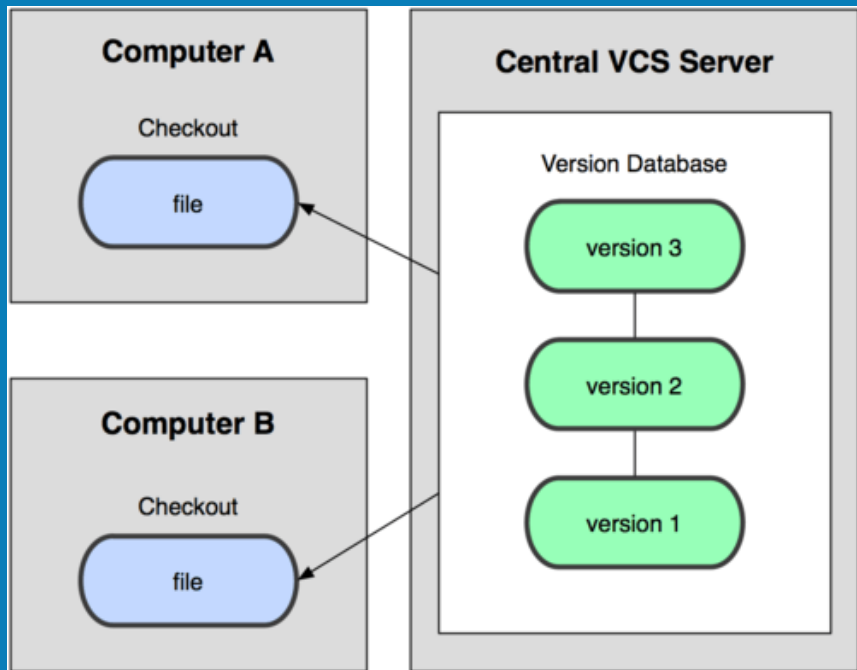
VCS – **неотъемлемая** часть любого процесса разработки ПО

Пример использования. Терминология.

- Trunk / master – выделенная или «главная» ветка разработки.
- Branch – ветвь разработки, которая позволяет сохранять изменения, не влияя на другие ветви.
- Commit – единица сохранения изменений в проекте.
- Merge – слияние независимых изменений в единую версию.
- Tag – метка, которой можно пометить определённую ревизию проекта.



Централизованные и распределённые системы.



Git глазами пользователя SVN



*<http://developerslife.ru/latest/189>

Идеология Git

- Git – распределённая система контроля версия
- Ваш воркспейс = ваш репозиторий
- Мощная поддержка работы с ветками (branches)
- «Безопасность» и надёжность за счёт использования SHA-1

Создание репозитория

- `git init` – создание нового (пустого) репозитория; больше информации доступно в help: `git help init`
 - `cd project_folder`
 - `git init`
 - `git add .`
 - `git commit`
- `git clone` – клонирование существующего репозитория
 - `git clone https://github.com/kyukhin/tarantool local_folder`

Информация о репозитории

- `git log` – список коммитов, начиная с последнего. `git help log` – полный список опций. Полезная опция `--graph`:
 - `git log --graph`
- `git branch` – список локальных бранчей. “`git branch -v`” для более полной информации.
- `git status` – информация об изменённых файлах.

Настройка удалённых репозиториев

- `git remote -v` – список удалённых репозиториев
- `git remote add ispc https://github.com/ispc/ispc.git` - добавление удалённого репозитория.
- `git help remote` – более полная информация

Бранчи

- `git branch -v` – информация о бранчах
- `git checkout <branch_name>` – выбрать существующий бранч
- `git branch <new_branch_name>` – создать бранч
- `git checkout -b <new_branch_name>` – создать бранч и выбрать его

Pull / push

- pull = fetch + merge
- git pull <remote> <branch_name> - выкачать изменения и попробовать смёрджить с текущей веткой
- git push <remote> <branch_name> - пуш в удалённый репозиторий в заданную ветку
- git push <remote> HEAD – пуш в удалённый репозиторий в ветку с таким же именем, как и локальная ветка.

rebase

- `git rebase master` – взять за основу мастер и приложить к нему все изменения из текущей ветки.

КОММИТЫ

- `git add file1.cpp file2.cpp` – добавить файлы в список для коммита.
- `git reset HEAD file1.cpp` – удалить файлы из списка для коммита.
- `git commit -m "Message"` – закоммитить файлы.

Типичный workflow

- `git checkout -b new_feature`
- <изменить файлы>
- `git add file1.cpp file2.cpp`
- `git commit -m "New feature"`
- `git push origin HEAD`
- <послать pull request на github>

Ресурсы

- Документация, дистрибутивы для большинства платформ
<http://git-scm.com>
- Документация на русском <http://git-scm.com/book/ru>
- Линус Торвальдс об истории создания Git:
 - <http://www.youtube.com/watch?v=4XpnKHJAok8> (английская версия)
 - <http://www.youtube.com/watch?v=BtAlN4MaBr8&list=PLF85270518253F2B3> (русская версия)
- <https://github.com> – популярный хостинг git репозиториев.

