
Git. Просто git.
undev.coursify.ru

Лекция 1.

1. О контроле версий
2. Какие СУВ бывают
3. Краткая история Git
4. Основы git
5. Установка git
6. Первоначальная настройка
7. Как получить помощь
8. Итоги

О контроле версий.

> *Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.*

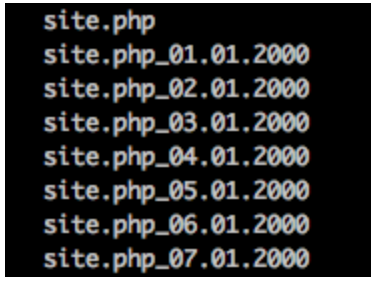
Локальные системы контроля версий

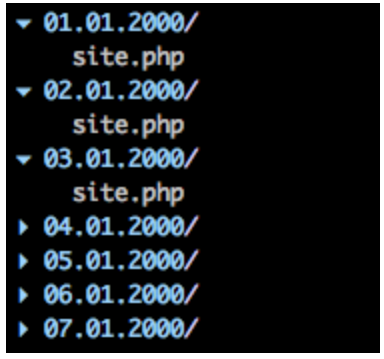
1. Создали файл
2. Написали код
3. Сохранили файл, выпустили релиз
4. Сделали резервную копию
5. Начали работать над новыми фичами
6. Поступил фидбек. Баги :(
7. Делаем резервную копию текущего файла, достаем предыдущий
8. Вносим правки, выкатываем изменения. Делаем резервную копию

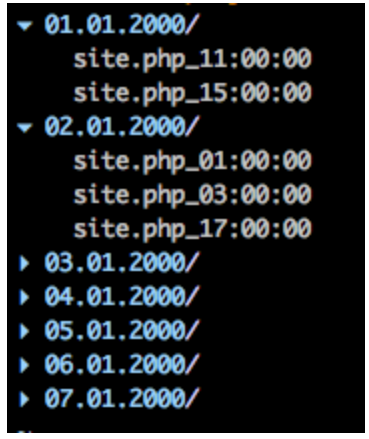
Локальные системы контроля версий

1. Создали файл
2. Написали код
3. Сохранили файл, выпустили релиз
4. Сделали резервную копию
5. Начали работать над новыми фичами
6. Поступил фидбек. Баги :(
7. Делаем резервную копию текущего файла, достаем предыдущий
8. Вносим правки, выкатываем изменения. Делаем резервную копию

Какие варианты?



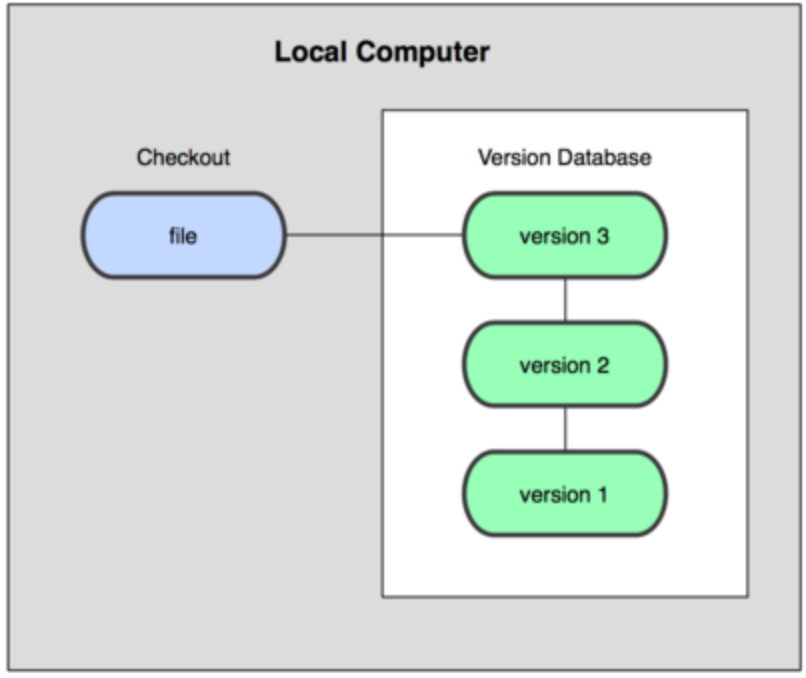




Стоп!


Всё время
спрашивайте себя:
«Не хуйню ли
я делаю?»

Они все же существуют.



RCS (Revision Control System)

Пример сессии

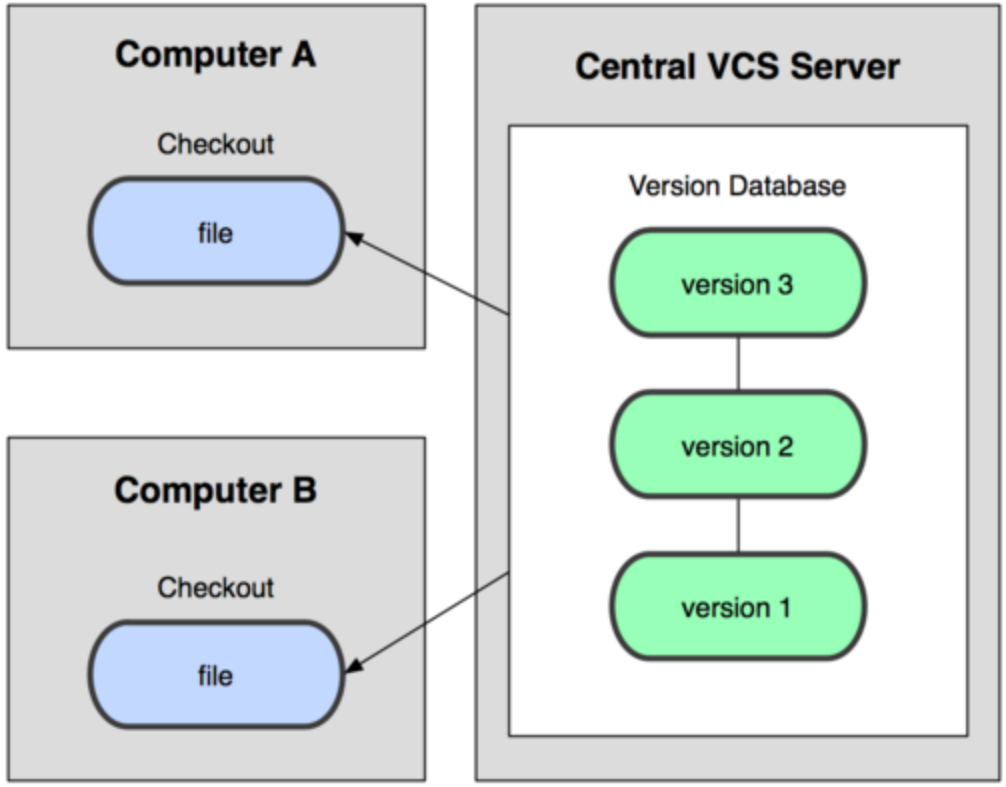
1. ci file.txt
2. co file.txt
3. ci file.txt

RCS была разработана в начале 1980-х годов Вальтером Тичи (Walter F. Tichy).

Недостатки

1. Работа только с одним файлом, каждый файл должен контролироваться отдельно;
2. Неудобный механизм одновременной работы нескольких пользователей с системой, хранилище просто блокируется пока заблокировавший его пользователь не разблокирует его;
3. От бекапов вас никто не освобождает, вы рискуете потерять всё.

Централизованные СКВ



CVS (Concurrent Versions System)

Пример сессии

1. cd some-project
2. cvs import -m "New project" path-in-repository none start
3. cd some-working-dir
4. cvs checkout path-in-repository
5. cvs commit -m "Some changes"
6. cvs update

Недостатки

1. Нет возможности сохранять версии директорий. Стандартный способ обойти это препятствие - это сохранить какой-либо файл (например, README.txt) в директории;
2. Перемещение, или переименование файлов не подвержено контролю версий. Стандартный способ сделать это: сначала скопировать файл, удалить старый с помощью команды cvs remove и затем добавить с его новым именем с помощью команды cvs add;

Subversion (SVN)

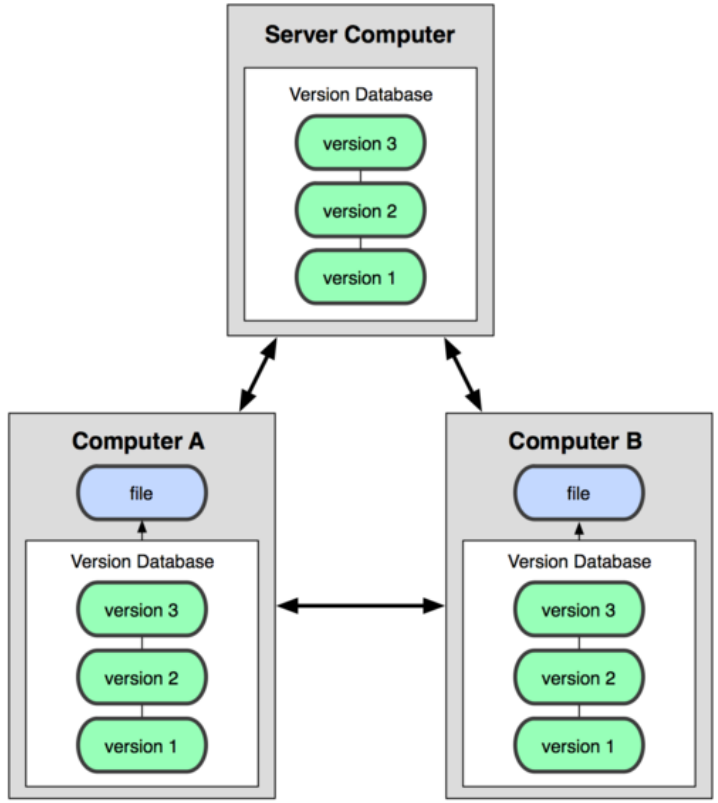
1. Атомарное внесение изменений (commit). В случае если обработка коммита была прервана не будет внесено никаких изменений.
2. Переименование, копирование и перемещение файлов сохраняет всю историю изменений.
3. Директории, символические ссылки и мета-данные подвержены контролю версий.
4. Эффективное хранение изменений для бинарных файлов.

Subversion (SVN)

Пример сессии

1. cd some-project
2. svn import -m "New project" path-in-repository
3. cd some-working-dir
4. svn checkout path-in-repository
5. svn commit -m "Some changes"
6. svn update (up)

Распределённые СКВ



Распределённые СКВ

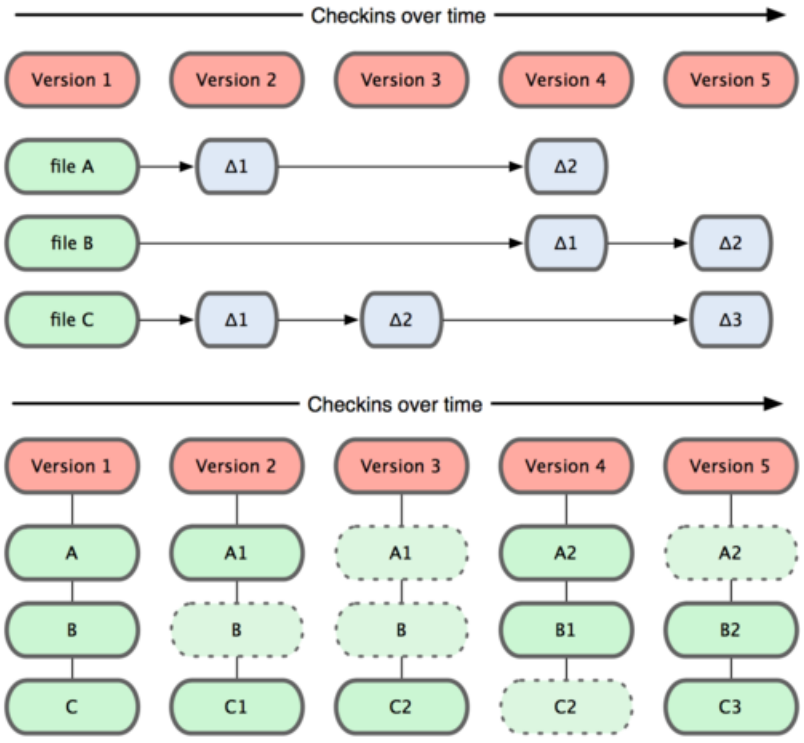
1. Git (http://git-scm.com/)
2. Mercurial (http://www.selenic.com/mercurial/)
3. Bazaar (http://bazaar-vcs.org/)
4. Codeville (http://codeville.org/)
5. Darcs (http://darcs.net/)
6. Monotone (http://monotone.ca/)

Краткая история Git

Основные требования

1. Скорость
2. Простота дизайна
3. Поддержка нелинейной разработки (тысячи параллельных веток)
4. Полная распределённость
5. Возможность эффективной работы с такими большими проектами, как ядро Linux (как по скорости, так и по размеру данных)

Распределённые СКВ



Отличия/требования к Git

1. Почти все операции — локальные
2. Git следит за целостностью данных
3. Чаще всего данные в Git только добавляются
4. Три состояния
   - каталог Git'а (Git directory),
   - рабочий каталог (working directory),
   - область подготовленных файлов (staging area).

Установка Git

1. Установка собранного пакета для вашей платформы
2. Установка из исходников
   - curl
   - zlib
   - openssl
   - expat
   - libiconv

Первоначальная настройка Git

1. git config
2. Виды конфигурации
   - Общие для всех пользователей системы (git config --system)
   - Настройки для конкретного пользователя (git config --global)
   - Конфигурационный файл в каталоге Git'а (git config)

Первоначальная настройка Git

Данные пользователя

1. git config --global user.name "John Doe"
2. git config --global user.email johndoe@example.com

Первоначальная настройка Git

Выбор редактора

1. git config --global core.editor emacs
2. git config --global core.editor vim

Первоначальная настройка Git

1. git config --global merge.tool vimdiff
2. git config --global merge.tool kdiff3
3. git config --global merge.tool tkdiff
4. git config --global merge.tool meld
5. git config --global merge.tool xxdiff
6. git config --global merge.tool emerge
7. git config --global merge.tool gvimdiff
8. git config --global merge.tool ecmerge

Проверка настроек

- git config --list

```
01. user.name=Scott Chacon
02. user.email=schacon@gmail.com
03. ...
04. color.status=auto
05. color.diff=auto
```

- git config user.name

Как получить помощь?

1. git help <команда>
2. git <команда> --help
3. man git-<команда>

На сегодня хватит

:)

И напоследок...

Оставляем фидбек - http://undev.coursify.ru/courses/18

Лекция оффлайн - http://zzet.org/learn-git/lection-1.html

Все лекции оффлайн - http://zzet.org/learn-git.html

Twitter - @zzetorg