

Devpractice

Разработка программного обеспечения, технологии и наука

Git для начинающих. Часть 1. Что такое системы контроля версий?

Автор: Marat Abdrakhmanov | 06.03.2017

7 комментариев

Система контроля версий (*Version Control System, VCS*) представляет собой программное обеспечение, которое позволяет отслеживать изменения в документах, при необходимости производить их откат, определять, кто и когда внес исправления и т.п. В статье рассмотрены виды *VCS*, принципы их работы, а также приведены примеры программных продуктов.

Что такое система контроля версий?

Наверное, всем знакома ситуация, когда при работе над проектом, возникает необходимость внести изменения, но при этом нужно сохранить работоспособный вариант, в таком случае, как правило, создается новая папка, название которой скорее всего будет “Новая папка” с дополнением в виде даты или небольшой пометки, в нее копируется рабочая версия проекта и уже с ним производится работа. Со временем количество таких папок может значительно возрасти, что создает трудности в вопросе отката на предыдущие версии, отслеживании изменений и т.п. Эта ситуация значительно ухудшается, когда над проектом работает несколько человек.

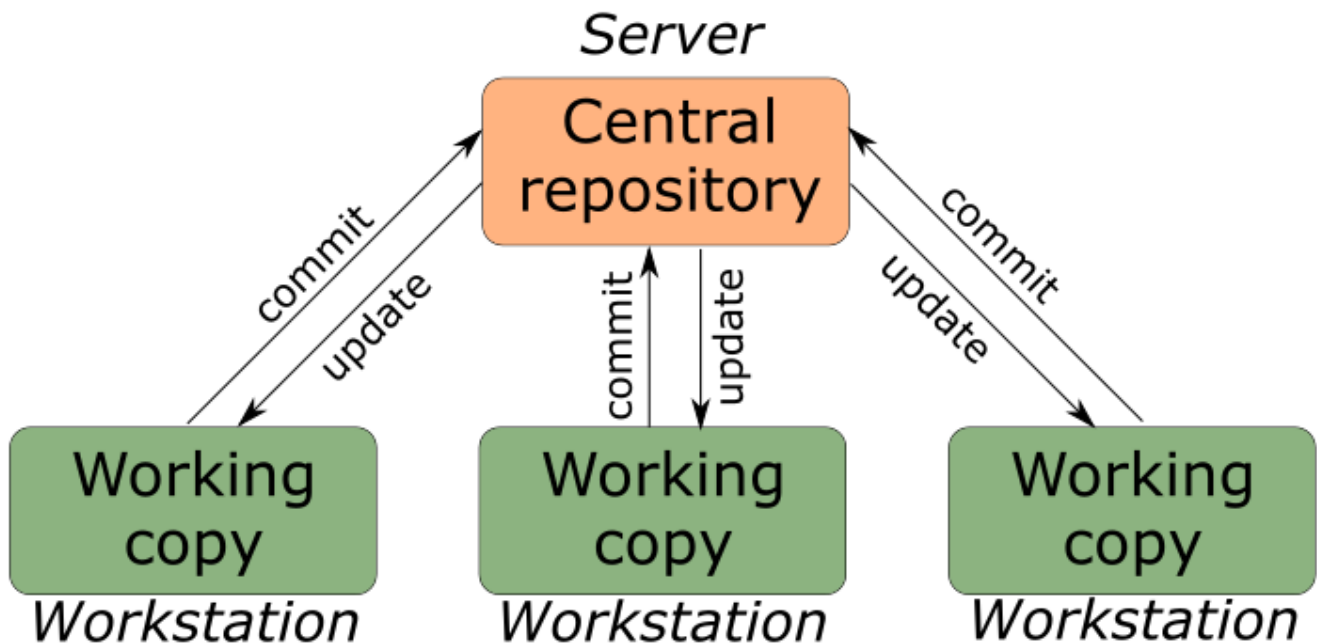
Для решения таких проблем как раз и используется система контроля версий, она позволяет комфортно работать над проектом как индивидуально, так в коллективе. *VCS* отслеживает изменения в файлах, предоставляет возможности для создания новых и слияние существующих ветвей проекта, производит контроль доступа пользователей к проекту, позволяет откатывать исправления и определять кто, когда и какие изменения вносил в проект. Основным понятием *VCS* является репозиторий (*repository*) – специальное хранилище файлов и папок проекта, изменения в которых отслеживаются. В распоряжении разработчика имеется так называемая “рабочая копия” (*working copy*) проекта, с которой он непосредственно работает. Рабочую копию необходимо периодически синхронизировать с репозиторием, эта операция предполагает отправку в него изменений, которые пользователь внес в свою рабочую копию (такая операция называется *commit*) и актуализацию рабочей копии, в процессе которой к пользователю загружается последняя версия из репозитория (этот процесс носит название *update*).

Централизованные и распределенные системы контроля версий

Системы контроля версий можно разделить на две группы: распределенные и централизованные.

Централизованные системы контроля версий

Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда репозиторий проекта существует в единственном экземпляре и хранится на сервере. Доступ к нему осуществлялся через специальное клиентское приложение. В качестве примеров таких программных продуктов можно привести *CVS*, *Subversion*.



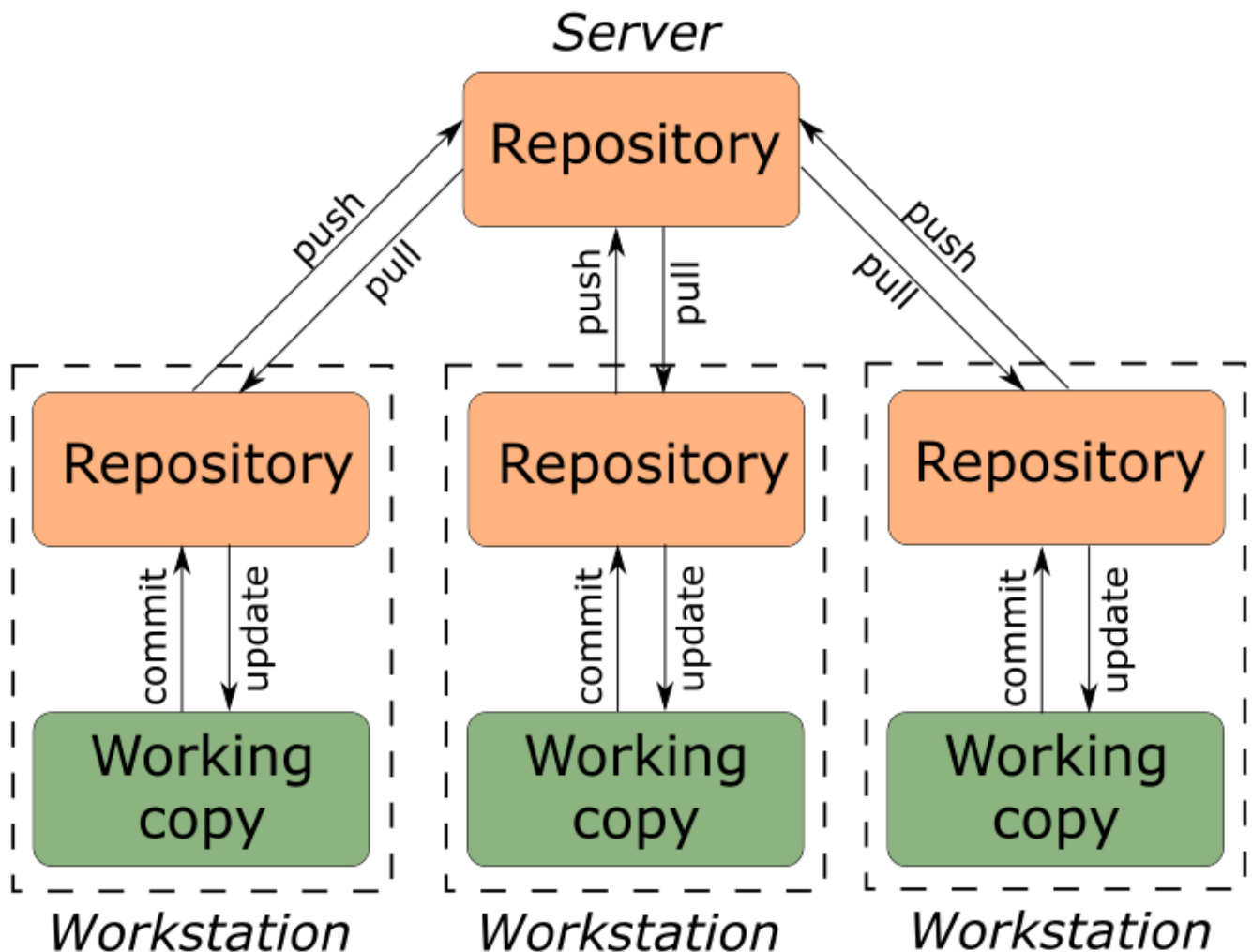
CVS (*Concurrent Versions System*, Система одновременных версий) одна из первых систем получивших широкое распространение среди разработчиков, она возникла в конце 80-х годов прошлого века. В настоящее время этот продукт не развивается, это в первую очередь связано с рядом ключевых недостатков, таких как невозможность переименования файлов, неэффективное их хранение, практически полное отсутствие контроля целостности.



Subversion (SVN) – система контроля версий, созданная на замену *CVS*. *SVN* была разработана в 2004 году и до сих пор используется. Несмотря на многие преимущества по сравнению с *CVS* у *SVN* все-таки есть недостатки, такие как проблемы с переименованием, невозможность удаления данных из хранилища, проблемы в операции слияния ветвей и т.д. В целом *SVN* был (и остается) значительным шагом вперед по сравнению с *CVS*.

Распределенные системы контроля версий

Распределенные системы контроля версий (*Distributed Version Control System, DVCS*) позволяют хранить репозиторий (его копию) у каждого разработчика, работающего с данной системой. При этом можно выделить центральный репозиторий (условно), в который будут отправляться изменения из локальных и, с ним же эти локальные репозитории будут синхронизироваться. При работе с такой системой, пользователи периодически синхронизируют свои локальные репозитории с центральным и работают непосредственно со своей локальной копией. После внесения достаточного количества изменений в локальную копию они (изменения) отправляются на сервер. При этом сервер, чаще всего, выбирается условно, т.к. в большинстве *DVCS* нет такого понятия как “выделенный сервер с центральным репозиторием”.



Большое преимущество такого подхода заключается в автономии разработчика при работе над проектом, гибкости общей системы и повышение надежности, благодаря тому, что каждый разработчик имеет локальную копию центрального репозитория. Две наиболее известные DVCS – это *Git* и *Mercurial*.



Начнем с *Mercurial*, эта система представляет собой свободную DVCS, которая построена таким образом, что в ней отсутствует понятие центрального репозитория, для работы с этой VCS используется (как правило) консольная утилита *hg*. *Mercurial* обладает всеми возможностями системы контроля версий, такими как ветвление, слияние, синхронизация с другими репозиториями. Данный проект используют и поддерживают большое количество крупных разработчиков, среди них *Mozilla*,

OpenOffice, *OpenJDK* и многие другие. Сам продукт написан на языке *Python* и доступен на большинстве современных операционных систем (*Windows*, *Mac OS*, *Linux*), также существует значительное количество утилит с графическим интерфейсом для работы с *Mercurial*. Основным конкурентом *Mercurial* на рынке распределенных систем контроля версий является *Git*, который, на сегодняшний день, выиграл гонку за лидерство.



Git – распределенная система контроля версий, разработанная Линусом Торвальдсем для работы над ядром операционной системы *Linux*. Среди крупных проектов, в рамках которых используется *git*, можно выделить ядро *Linux*, *Qt*, *Android*. *Git* свободен и распространяется под лицензией *GNU GPL 2* и, также как *Mercurial*, доступен практически на всех операционных системах. По своим базовым возможностям *git* схож с *Mercurial* (и другими *DVCS*), но благодаря ряду достоинств (высокая скорость работы, возможность интеграции с другими *VCS*, удобный интерфейс) и очень активному сообществу, сформировавшемуся вокруг этой системы, *git* вышел в лидеры рынка распределенных систем контроля версий. Необходимо отметить, что несмотря на большую популярность таких систем как *git*, крупные корпорации, подобные *Google*, используют свои *VCS*.

Это была вводная лекция по системам контроля версий. В дальнейшем, все изложение будет касаться только *git*.

Если вам больше нравится **учиться по видео-лекциям**, то рекомендуем классный курс по *git* от **GeekBrains**, перейдите по [ссылке](#) и найдите в разделе "Курсы" курс "**Git. Быстрый старт**". Он **бесплатный**, нужно только зарегистрироваться на сайте. Рекомендуем повнимательнее посмотреть на этот ресурс, на нем ещё очень много чего интересного!

Часть 2. Установка *Git* >>>

Раздел: *Git* *Git* для начинающих Метки: *Git*, *Git* для начинающих, *VCS*, Системы контроля версий

Git для начинающих. Часть 1. Что такое системы контроля версий?: 7 комментариев



Петр

08.12.2017

Отличная статья! Все понятно и без воды.



writer

18.12.2017

Спасибо! Стараемся!



Александр

04.09.2018

Отличная статья. Хотел прочитать про систему контроля версий и нашел эту статью. Коротко, доступно, понятно! Супер!



writer

04.09.2018

Спасибо! Стараемся)



Евгений

22.11.2020

Спасибо!



Дмитрий

25.10.2021

Спасибо!

Исправьте "В целом SVN был (и остается) значительнОм шагом вперед по сравнению с CVS. "



writer

01.11.2021

Спасибо! Исправили

