

Narzędzia programistyczne - GIT

Kamil Maras

kamil.maras@gmail.com
@KamilMaras

Agenda

- Zintegrowane środowisko programistyczne
- **Systemy kontroli wersji**
- Narzędzia wspomagające wytwarzanie aplikacji
- Narzędzia ciągłej integracji

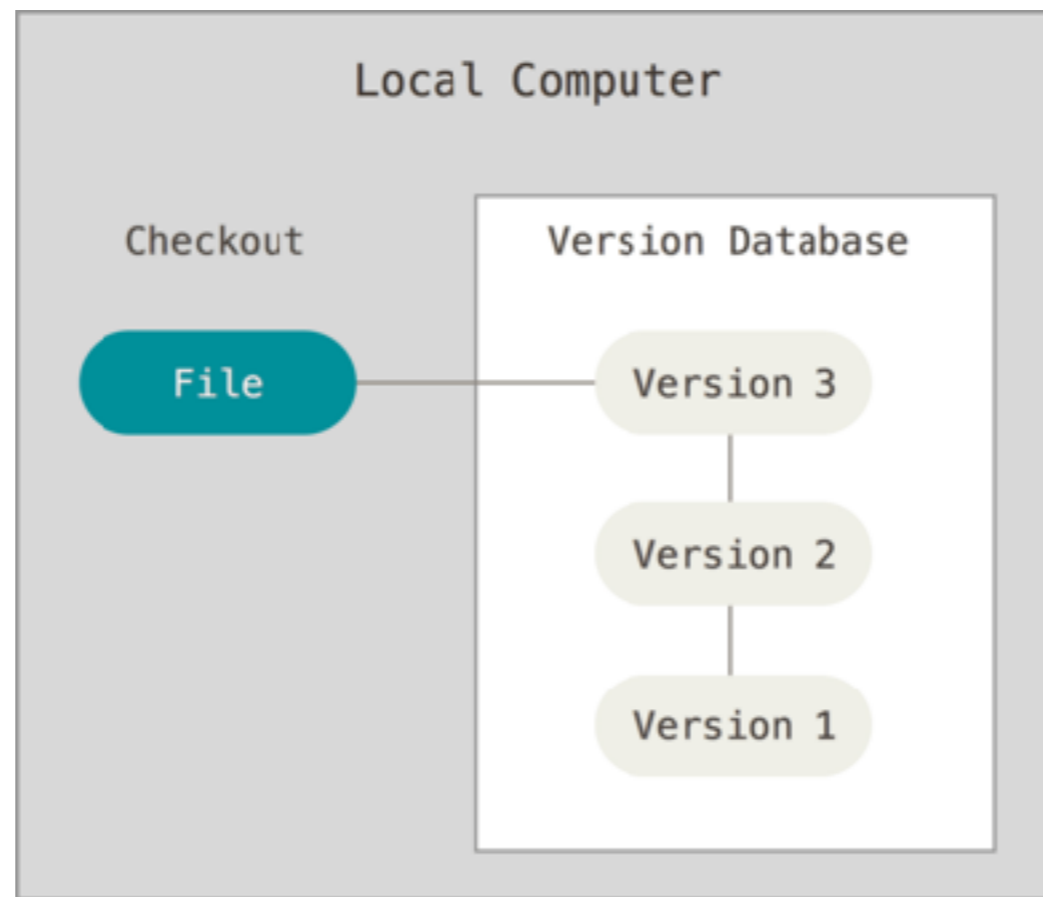
Systemy kontroli wersji

System kontroli wersji (ang. version/revision control system) – oprogramowanie służące do śledzenia zmian głównie w kodzie źródłowym oraz pomocy programistom w łączeniu zmian dokonanych w plikach przez wiele osób w różnym czasie.

Szukaj w Webstorm - VCS

Lokalny

Local Version Control Systems

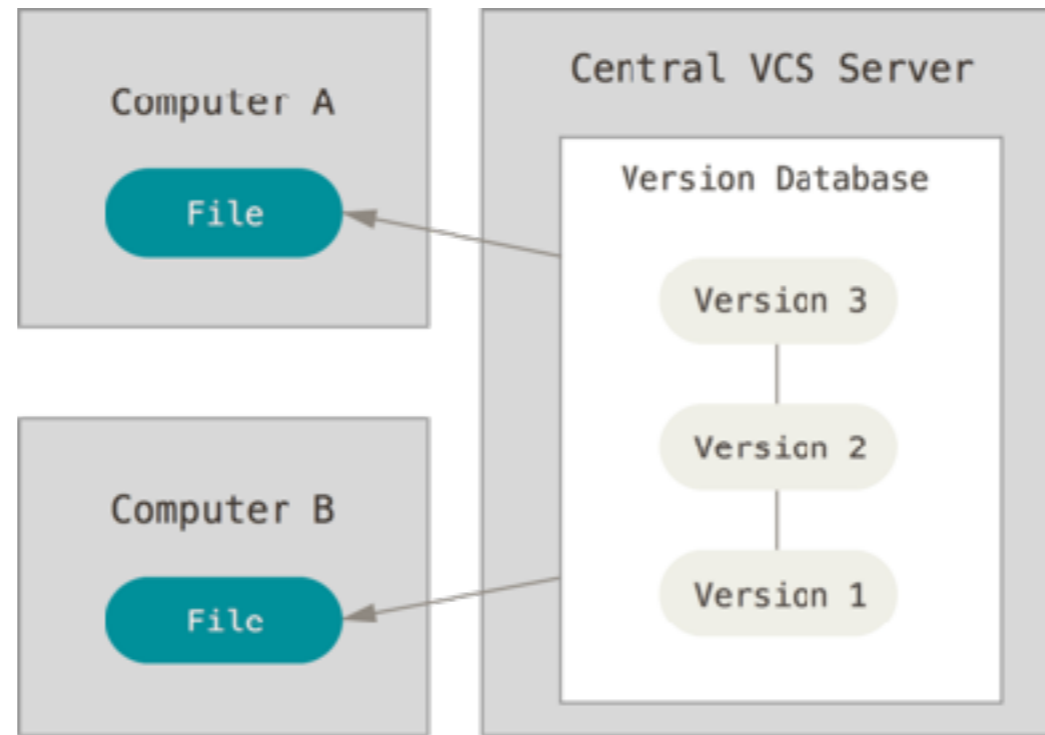


Grafika na licencji Creative Commons Attribution 3.0 Unported License. git-scm.com

Np. RCS w Mac OS X

scentralizowany

Centralized Version Control Systems - CVCS

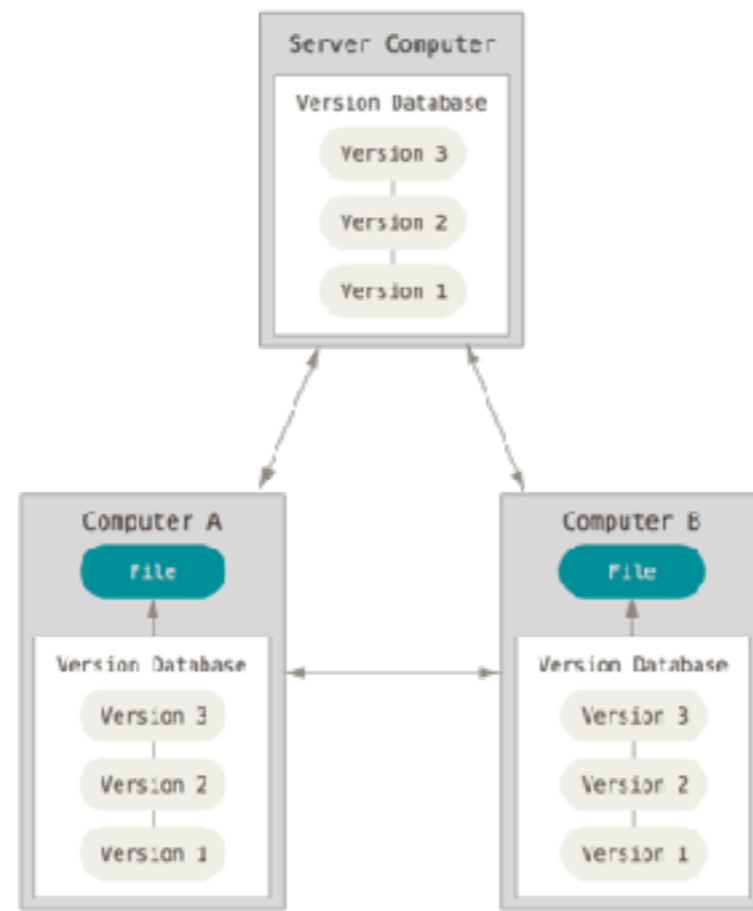


Grafika na licencji Creative Commons Attribution 3.0 Unported License. git-scm.com

np. CVS, Subversion

rozproszony

Distributed Version Control Systems



Grafika na licenciji Creative Commons Attribution 3.0 Unported License. git-scm.com

np. GIT, Mercurial

Dlaczego GIT ?

<https://rhodecode.com/insights/version-control-systems-2016>

Krótką historia



As with many great things in life, Git began with a bit of creative destruction and fiery controversy.

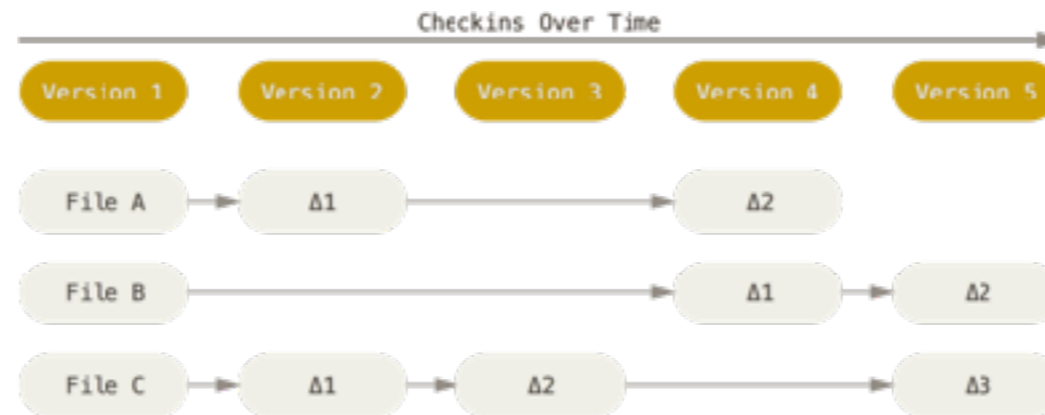
The Linux kernel is an open source software project of fairly large scope. For most of the lifetime of the Linux kernel maintenance (1991–2002), changes to the software were passed around as patches and archived files. In 2002, the Linux kernel project began using a proprietary DVCS called BitKeeper.

In 2005, the relationship between the community that developed the Linux kernel and the commercial company that developed BitKeeper broke down, and the tool's free-of-charge status was revoked. This prompted the Linux development community (and in particular Linus Torvalds, the creator of Linux) to develop their own tool based on some of the lessons they learned while using BitKeeper. Some of the goals of the new system were as follows:

1. Speed
2. Simple design
3. Strong support for non-linear development (thousands of parallel branches)
4. Fully distributed
5. Able to handle large projects like the Linux kernel efficiently (speed and data size)

Charakterystyka

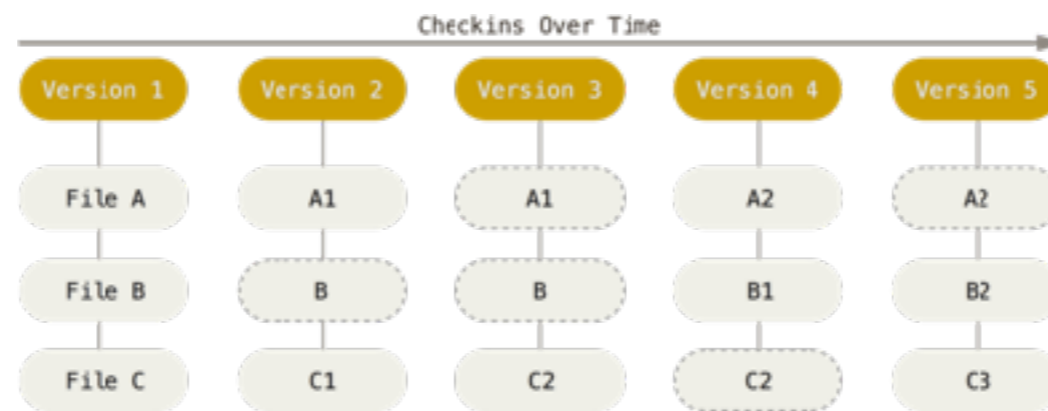
Przechowywanie zmian



Grafika na licencji Creative Commons Attribution 3.0 Unported License. git-scm.com

Większość systemów trzyma zmiany jako drzewo małych zmian w poszczególnych rewizjach.

Przechowywanie migawek



Grafika na licencji Creative Commons Attribution 3.0 Unported License. git-scm.com

Większość systemów trzyma zmiany jako drzewo małych zmian w poszczególnych rewizjach.

Charakterystyka

- Lokalne zmiany
- Kontrola spójności
- Standardowo git tylko dodaje nowe dane

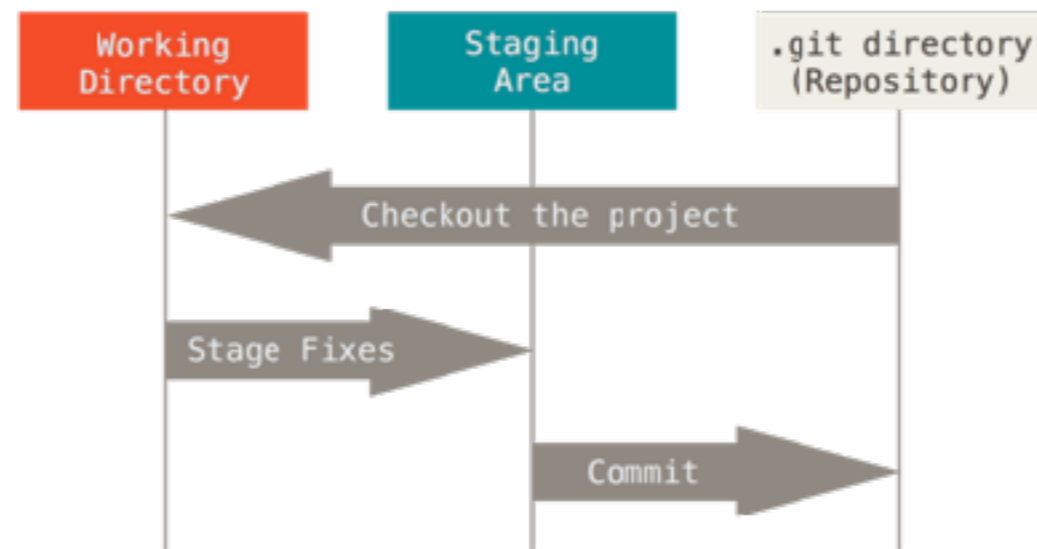
Większość zmian w repozytorium jest lokalna, co powoduje:

1. Zmiany są szybsze - np. przeglądanie historii, porównania plików
2. Możesz umieszczać zmiany w lokalnym repozytorium.

Każdy obiekt w repozytorium ma wyliczony skrót SHA-1 po którym następuje kontrola spójności.

git reset —hard

Trzy stany



Grafika na licencji Creative Commons Attribution 3.0 Unported License. git-scm.com

1. **Zatwierdzony** oznacza, że dane zostały bezpiecznie zachowane w Twojej lokalnej bazie danych.
2. **Zmodyfikowany** oznacza, że plik został zmieniony, ale zmiany nie zostały wprowadzone do bazy danych.
3. **Śledzony** - oznacza, że zmodyfikowany plik został przeznaczony do zatwierdzenia w bieżącej postaci w następnej operacji commit.

.git - to katalog gdzie jest całe lokalne repozytorium - zawiera obiektową bazę danych i metadane.

working directory - to katalog z jedną z wersji kodu - tą nad którą pracujemy

przechowalnia (staging area) - miejsce gdzie przechowywana jest informacja o rzeczach które mają być następne umieszczone w repozytorium.

Jeśli jakaś wersja pliku znajduje się w katalogu git, uznaje się ją jako zatwierdzoną. Jeśli plik jest zmodyfikowany, ale został dodany do przechowalni, plik jest śledzony. Jeśli zaś plik jest zmodyfikowany od czasu ostatniego pobrania, ale nie został dodany do przechowalni, plik jest w stanie zmodyfikowanym.

Podstawowy flow

1. Dokonujesz modyfikacji plików w katalogu roboczym.
2. Oznaczasz zmodyfikowane pliki jako śledzone, dodając ich bieżący stan (migawkę) do przechowalni.
3. Dokonujesz zatwierdzenia (commit), podczas którego zawartość plików z przechowalni zapisywana jest jako migawka projektu w katalogu Git.

Bibliografia

- <https://try.github.io/levels/1/challenges/1>
- <http://justinhileman.info/article/git-pretty/git-pretty.png>
- <https://git-scm.com/doc>
- <https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud>
- <http://ohshitgit.com/>

Dziękuję