

Wprowadzenie do systemu kontroli wersji Git

AM

29 września 2024



1 Wstęp

W 2005 roku fiński programista Linus Torvalds, twórca jądra Linux, wydał pierwszą wersję rozproszonego systemu kontroli wersji **Git**, mającego wspomóc prace nad rozwojem Linuksa i stanowić open-source'owy zamiennik używanego przez niego dotychczas programu BitKeeper.

Najważniejsze zadania narzędzia Git to:

- Śledzenie zmian w projekcie i wgląd w jego rozwój na przestrzeni czasu,
- Zarządzanie wersjami,
- Rozwiązywanie konfliktów zaistniałych podczas pracy wielu twórców nad tym samym plikiem,
- Synchronizacja zmian oraz praca w trybie offline.

Krótko mówiąc, Git umożliwia komfortową i efektywną pracę nad projektem współtworzonym przez wielu programistów i jest on w dzisiejszych czasach najpopularniejszym systemem kontroli wersji używanym przez większość programistów i przedsiębiorstw w branży informatycznej.

2 Główne cechy Git'a

Dzięki doświadczeniu z prowadzeniem i zarządzaniem dużego projektu, oraz wiedzy nabytej podczas pracy nad Linuksem, autor Git'a był w stanie zaprojektować system kontroli wersji, który zaspokajał by jego potrzeby rozwiązując problemy innych dostępnych dotychczas systemów.

Git posiada kilka kluczowych cech, które wyróżniają go na tle innych systemów kontroli wersji. Pierwszą z nich jest **zapisywanie "migawek" całego drzewa plików** zamiast rejestrowania samych zmian w poszczególnych rewizjach. Podczas tworzenia commita zapisywany jest aktualny stan wszystkich plików repozytorium. Jeżeli dany plik nie został zmieniony w danej rewizji, to nie zostanie on nadpisany, a zapisana zostanie **referencja** do wersji pliku z poprzedniej rewizji.

Git stawia na **nieliniowy rozwój projektów**, implementując funkcjonalność **branch'y** (gałęzi/odnóg) i ich **merge'owania** (łączenia/scalania), która pozwala twórcom dodawać do projektów nowe elementy bez ryzyka uszkodzenia już działającego kodu. System umożliwia decydowanie o tym, która ze zmian w pliku dokonanych przez wielu różnych programistów zostanie wybrana i zaimplementowana. Pomaga to uniknąć wzajemnego nadpisywania pracy. Twórcy mają również możliwość cofania nowo dodanych zmian i powrotu do poprzednich wersji plików. Gdy nowa gałąź jest gotowa i zostanie zatwierdzona scalić ją można z **gałęzią główną**.

Git **nie wymaga nieustannego dostępu do internetu**. Każdy lokalny zapis repozytorium jest jego pełnym backupem. Pozwala to na pracę offline i synchronizację wszelkich zmian w momencie połączenia z internetem. Można dzięki temu np. przeglądać stare wersje plików i wracać do poprzednich rewizji lokalnie, bez potrzeby ciągłej wymiany informacji z serwerem.

Git jest również znany ze swojej **szybkości**. Autorowi zależało na jak największej wydajności i efektywnej obsłudze dużych projektów. Potwierdzają to liczne testy wykonane między innymi przez fundację Mozilla.

3 Sposoby korzystania z Git'a

Istnieją dwa główne sposoby na korzystanie z systemu Git - poprzez tekstowy terminal i graficzny interfejs. Tekstowy terminal umożliwia korzystanie ze wszystkich możliwych poleceń Git'a, a interfejsy graficzne bardzo często implementują tylko pewien zbiór wszystkich jego funkcji. Istnieje wiele serwisów i aplikacji, które umożliwiają nam korzystanie z Git'a poprzez wygodny interfejs graficzny. Do aplikacji desktopowych należą GitHub Desktop, GitKraken czy SourceTree. W tych czasach szczególną popularnością cieszą się również interfejsy webowe, oferowane przez serwisy takie jak GitHub, GitLab, SourceForge czy Bitbucket. Warto wspomnieć o tym, że niektóre IDE posiadają integrację z systemem Git, co przyspiesza pracę nad projektami i nie wymaga korzystania z wielu oddzielnych aplikacji. Do tych IDE należą: Visual Studio Code, IntelliJ IDEA czy Eclipse.

4 Instalacja Git'a

4.1 Instalacja w systemie Windows

System Git dostępny jest do pobrania na [oficjalnej stronie Git'a](#). Jeżeli jednak zdecydujemy się korzystać z interfejsu graficznego możemy zainstalować jedną z aplikacji desktopowych pokroju [GitHub Desktop](#), lub korzystać z serwisu dostępnego przez przeglądarkę.

4.2 Instalacja w systemie Linux

System Git najłatwiej zainstalować przy pomocy odpowiedniej komendy w terminalu. Istnieją dwie różne komendy zależnie od dystrybucji.

W przypadku korzystania z dystrybucji opartych na Debianie:

```
sudo apt-get install git
```

W przypadku korzystania z Fedory:

```
sudo yum install git
```

4.3 Instalacja w systemie OS X

Git'a na systemy firmy Apple możemy zainstalować na kilka sposobów.

Jednym sposobem jest zainstalowanie Git'a za pomocą instalatora dostępnego na [SourceForge](#).

Innym sposobem jest wykorzystanie [Homebrew](#). Po zainstalowaniu Homebrew wpisujemy następującą komendę:

```
brew install git
```

Jeżeli chcemy zainstalować graficzny interfejs Git-gui wpisujemy:

```
brew install git-gui
```

Git jest również zawarty w paczce [Xcode](#).