

# SUBVERSION

TOMASZ ŁUKASZUK

**STRESZCZENIE:** Dokument przedstawia system Subversion jako następcę systemu CVS. Zawarto w nim instrukcję utworzenia i pracy z systemem Subversion.

## 1. WPROWADZENIE

Subversion (znany również jako SVN) jest zaawansowanym systemem kontroli wersji, który powstał w celu zastąpienia CVS. Funkcjonalnie jest z nim zgodny w większości przypadków, z kompatybilności zrezygnowano tylko tam, gdzie było to niezbędne.

Od niepamiętnych czasów kod otwartych projektów był zarządzany przez system CVS. Jego cechy, w szczególności możliwość niezależnej pracy kilku deweloperów nad jednym plikiem, łączenia zmian dokonanych przez różnych deweloperów, pracy na gałęziach i łączenia gałęzi, były źródłem jego sukcesu. Niestety, CVS, jako system powstały z RCS - systemu kontroli rewizji plików - posiada pewne wady. Należą do nich brak możliwości sprawnego przenoszenia plików, zmian nazwy, brak atomowych zapisów, skomplikowany system zarządzania gałęziami, konieczność łączenia się z serwerem przy wielu operacjach. Projekt Subversion powstał w jednym specyficznym celu - stworzyć coś równie dobrego, co CVS, lecz pozbawionego jego wad. Można z całą pewnością powiedzieć, że cel został osiągnięty.

Subversion zapisuje repozytorium z plikami wewnątrz bazy danych Berkeley DB. Sposób zapisu danych w repozytorium obejmuje zarządzanie zmianami całego drzewa plików i katalogów, nie tylko plików, jak w przypadku CVS. Podczas wprowadzania zmian do repozytorium, wszystkie z nich następują jednocześnie i otrzymują wspólny numer rewizji. Zmiany są atomiczne - albo wszystkie znajdują się w repozytorium, albo wprowadzenie nie dojdzie do skutku. Powyższe metody korzystania z repozytorium pozwalają na uzyskanie wielu korzyści.

Pierwszą i najważniejszą z tych korzyści jest poprawna obsługa kopiowania i zmiany nazwy. Jeśli w systemie Subversion zostanie skopiowany bądź

przeniesiony jakiś plik lub katalog, repozytorium będzie pamiętać jego oryginalne położenie oraz historię zmian - żadna informacja nie jest tracona.

Niespodziewanym efektem śledzenia zmian drzewa i atomicznych modyfikacji jest natomiast bardzo prosta obsługa znaczników i gałęzi. W celu oznaczenia konkretnej rewizji drzewa kodu wystarczy skopiować te drzewo w inne miejsce w repozytorium. Kopia drzewa będzie pamiętać, z jakiej konkretnej rewizji pochodzi, stając się efektywnie znacznikiem wersji kodu. Gdyby teraz zacząć wprowadzać zmiany do skopiowanego drzewa, zachowywałoby ono zmiany osobno z oryginalnym drzewem, pamiętając jednak wspólną historię - kopia stałaby się gałęzią. Aby skorzystać z tych właściwości, przyjmuje się specjalny sposób budowania repozytorium. W głównym katalogu repozytorium tworzy się trzy podkatalogi: trunk, zawierający aktualną, rozwijaną, deweloperską wersję kodu, branches, zawierający gałęzie kodu, oraz tags, zawierający znaczniki.

Do każdego pliku w repozytorium Subversion można przyporządkować dowolne metadane - tzw. atrybuty. Można je wykorzystać do zapisywania informacji razem z plikiem, np. komentarzy. Istnieją też specjalne atrybuty, zapisujące np. atrybut wykonywalności plików lub listę plików nie obejmowanych kontrolą wersji. Atrybuty są wersjonowane.

Dostęp do repozytorium Subversion możliwy jest na trzy sposoby. Możliwe jest bezpośrednie korzystanie z repozytorium na lokalnym dysku. W ten sposób najłatwiej utrzymywać prywatne repozytoria. Drugą metodą jest otwarcie dedykowanego serwera dostępnego przez SSH. Można również użyć modułu mod\_dav\_svn, aby udostępniać repozytorium Subversion jako katalog WebDAV przez serwer Apache 2.

## 2. Utworzenie repozytorium

Pracę z repozytorium rozpoczynamy od jego utworzenia. Służy do tego komenda `svnadmin create`.

```
$ svnadmin create path/to/repository
```

Mamy już działające repozytorium dostępne pod URL `file://path/to/repository`. Można teraz dokonywać w nim operacji: dodawać pliki i katalogi, modyfikować je, kopiować, itd.

Struktura zawartości repozytorium może być dowolna, ale istnieją pewne zalecane standardy organizacji: katalog `trunk` przechowuje „główną linię” rozwojową, katalog `branches` zawiera gałęzie, a katalog `tags` tagi (kopie projektu, których

nie będziemy zmieniać). Jeżeli repozytorium przechowuje tylko jeden projekt wówczas zalecane jest stworzenie następującej struktury katalogów:

```
/trunk  
/branches  
/tags
```

Jeżeli repozytorium zawiera wiele projektów administratorzy często tworzą poniższą strukturę:

```
/paint/trunk  
/paint/branches  
/paint/tags  
/calc/trunk  
/calc/branches  
/calc/tags
```

Stwórzmy kopię roboczą repozytorium i utwórzmy zalecaną w nim strukturę katalogów.

```
$ svn co file://path/to/repository workcopy  
$ cd workcopy  
$ svn mkdir trunk branches tags  
$ svn commit -m "first commit"
```

### 3. PRACA Z REPOZYTORIUM

Polecenie `svn co` służy do utworzenia kopii roboczej wybranego katalogu z repozytorium. Polecenie to należy użyć w momencie, gdy chcemy pobrać zawartość repozytorium i rozpocząć pracę.

Kopia robocza repozytorium Subversion jest lokalnym obrazem fragmentu repozytorium. Wszelka praca z danymi zapisanymi w repozytorium odbywa się właśnie w kopii roboczej. Polecenie `svn` ma zestaw komend przywodzących na myśl komendy basha, które służą do manipulowania repozytorium. Są to komendy `svn cat`, `cp`, `rm`, `ls`, `mkdir`, `mv`.

Komenda `svn commit` służy do utrwalania zmian w repozytorium. Zapis do repozytorium powoduje naniesienie lokalnych zmian w kopii roboczej do repozytorium i nadanie im unikalnego numeru rewizji.

```
$ svn co file://path/to/repository/rep_directory  
$ cd rep_directory  
$ svn [cat, cp, rm, ls, mkdir, mv] --operacje w kopii lokalnej
```

```
$ svn ci -i "comment"
```

#### 4. PRZYDATNE KOMENDY

svn add – dodaje nowe pliki i katalogi  
svn copy – kopiuje pliki lub katalogi  
svn delete – usuwa plik lub katalog  
svn delete <URL> - usuwa bezpośrednio z repozytorium  
svn diff – pokazuje szczegółowe zmiany pomiędzy lokalną wersją a ostatnią w repozytorium  
svn export – działa tak jak checkout ale pomija katalogi .cvs używane przez system  
svn import – umieszcza strukturę z plikami w repozytorium  
svn list – wyświetlenie zawartości repozytorium  
svn log – pokazuje historię zmian dla wybranego pliku lub katalogu  
svn merge – scala zmiany z dwóch źródeł/wersji  
svn mkdir – tworzy katalog  
svn mkdir <URL> - tworzy katalog bezpośrednio w repozytorium  
svn move – przenosi pliki lub katalogi  
svn resolve – usuwa pliki używane przy rozwiązywaniu konfliktu  
svn revert – cofa zmiany w kopi roboczej  
svn status – pokazuje zmiany wprowadzone lokalnie w kopi roboczej  
svn switch – przełącza kopię roboczą pomiędzy gałęziami  
svn update – pobiera zmiany wprowadzone przez innych użytkowników

#### 5. METADANE (PROPERTIES)

W repozytorium Subversion dowolny plik lub katalog może mieć ustawione specjalne właściwości. Właściwości to para typu klucz-wartość. Istnieje pewna ilość predefiniowanych właściwości, np. svn:ignore, svn:mime-type, svn:eol-style, ale można również definiować własne klucze. Wartości właściwości mogą być tekstowe lub binarne (np. pliki). Metadane także są wersjonowane.

Manipulowanie metadanymi:

svn propset – ustawienie właściwości  
svn propget – odczytanie właściwości  
svn proplist – wyświetlenie kluczy przypisanych określonemu plikowi/katalogowi  
svn propdel – usunięcie właściwości

svn propedit – edycja właściwości

## 6. GAŁĘZIE

Gałęzie są fundamentalną częścią wszystkich systemów kontroli wersji. Pozwalają one na rozwój oprogramowania w kilku w pewnym stopniu niezależnych kierunkach.

Tworzenie gałęzi w Subversion jest bardzo proste, należy jedynie utworzyć kopię projektu w repozytorium przy użyciu komendy `svn copy`. Kopia może być umieszczona w dowolnym miejscu repozytorium, przyjęte jest jednak umieszczanie kopii w katalogu `branches`.

Istnieją dwa sposoby tworzenia kopii: zrobienie kopii lokalnie i zapisanie zmian do repozytorium lub zrobienie kopii bezpośrednio w repozytorium.

Pierwszy sposób:

```
$ svn co file://path/to/repository/calc workcopy
```

```
A workcopy/trunk/  
A workcopy/trunk/Makefile  
A workcopy/trunk/integer.c  
A workcopy/trunk/button.c  
A workcopy/branches/  
Checked out revision 340.
```

```
$ cd bigwc
```

```
$ svn copy trunk branches/my-calc-branch
```

```
$ svn status
```

```
A + branches/my-calc-branch
```

```
$ svn commit -m "Creating a private branch"
```

```
Adding          branches/my-calc-branch  
Committed revision 341.
```

Drugi sposób:

```
$ svn copy file://path/to/repository/calc/trunk \  
file://path/to/repository/branches/my-calc-branch \  
-m "Creating a private branch"
```

```
Committed revision 341.
```

## 7. BIBLIOGRAFIA

1. <http://linux.gery.pl/dzialy/programowanie/narzedzia/subversion>
2. <http://jakilinux.org/aplikacje/subversion-dla-kazdego>
3. <http://pl.wikipedia.org/wiki/SVN>
4. <http://www.subversion.org>

---

Politechnika Białostocka  
Wydział Informatyki  
Katedra Oprogramowania  
ul. Wiejska 45A  
15-351 Białystok

21 lutego 2007