Wprowadzenie do CVS

Jarek Miszczak

24 stycznia 2007

Streszczenie

Dokument ten omawia podstawowe zagadnienia związane z systement kontroli wersji CVS. Przedstawione są tylko polecenia potrzebne w codziennej pracy z tym systemem.

1 Wprowadzenie

Kiedy warto zainteresować się kontrolową wersji? Napewo jeżeli musimy współpracować z innymi w trakcie tworzenia oprogramowania. Wówczas CVS (lub jemu podobny system) jest konieczny do utrzymania projeku w całości. Ale CVS przydaje się także gdzy pracujemy sami i chcemy przechowywać swoją pracę w hednym miejscu. Nasz prijekt nie musi być związany z pisaniem oprogramowania – kontrolę wersji można zastosować z powodzeniam do tworzenia dokumentacji czy stron WWW.

2 Podstawowe komendy

Z pośród kilkudziesięciu komed programu **cvs** do codziennej pracy z tym systemem wystarczy znajomość zaledwie kilku.

2.1 Tworzenie repozytorium

Do rozpoczęcia pracy z systemem CVS potrzebujemy dostępu do repozytorium – może się ono znajdować w naszym katalogu domowym lub na jakimś serwerze.

Jeżeli chcemy utworzyć repozytorium we swoim katalogu domowym możemy to zrobić pisząc

```
cvs -d /moj/katalog/repozytorium init
```

Wszystkie operacje związane z CVS mają postać

```
cvs opcje_globalne polecenie opcje_polecenia
```

W tym wypadku wywołane zostało polecenie **init** z opcją −d ustawioną na `/moj/katalog/repozytorium`. Opcja −d wzkazuje programowi **cvs** z jakiego repozytorium ma korzystać. Jeżeli nie jest ona wyszczególniona w lini poleceń, to program **cvs** korzysta ze zmiennej środowskowej CVSROOT. Zatem powyższe polecenie odpowiada wykonaniu

```
export CVSROOT=/moj/katalog/repozytorium
```

w przypadku powłoki bash lub

```
setenv CVSROOT /moj/katalog/repozytorium
```

w przypadku powłoki csh, a następnie

```
cvs init
```

Inne możliwe postacie zmiennej CVSROOT to

```
export CVSROOT=:ext:user@server.pl:/usr/cvs
```

dla repozytoriów zdalnych z którymi komunikować się będziemy prze SSH lub

```
export CVSROOT=:pserver:user@server.pl:/usr/cvs
```

dla repozytorów z którymi będziemy się komunikować wykorzystując mechniza uwieżytelniania wbudowany w CVS. Wszędzie poniżej zakładamy, że zmienna CVSROOT jest ustawiona.

2.2 Importowanie projektu

Aby umieścić pliki z katalogu w repozytorium przejść do tego katalogu i wydać polecenie

```
cvs import nazwa_modulu oznaczenie_tworcy oznaczenie_wydania
```

gdzie nazwa_modulu okresla pod jaka nazwa nasz projekt/katalog zostanie umieszczony w repozytorium – ta nazwa bedzie słuzyła do poźniejszego pobeierania projektu z repozytorium. Parametry oznaczenie_tworcy oraz oznaczenie_wydania sa przeznaczone do użytku wewnętrznego dla systemu i obia musza zaczynac się od litery.

2.3 Praca z plikami

2.4 Pobieranie plików z repozytorium

Pobranie plików z repozytorium odbywa się po wydaniu polecenia

```
cvs checkout nazwa_modulu
```

lub jest skróconej wersji

```
cvs co nazwa_modulu
```

2.4.1 Wprowadzanie zmian

Modyfikując pliki w katalogu powstałym po pobraniu modułu zmieniamy jesynie nasze lokalne wersje tych plików. Jeżeli uznamy, że praca została zakonczona musimy przesłać zmiany do repozytorium wykonując

```
cvs commit
```

lub w skróconej wersji

```
cvs com
```

Możliwe jest także wyspecyfikowanie które pliki chcemy wysłać na server

```
cvs commit plik_nr1 plik_nr1
```

Przed wysłaniem zmian do repozytorium **cvs** poprosi nas o wprowadzenie opisu dokonanych zmian. W tym celu zostnie otwarty domyślny edytor (vi w systemie Linux bądź notepad.exe w systmie Windows). Jeżeli chcemy korzystać z innego edytora musimy zmodyfikować zmienną CVSEDITOR:

```
export CVSEDITOR=/usr/bin/kate
```

2.4.2 Uaktualnienie

Wydane polecenia

cvs update

lub

cvs up

spowoduje uaktualnienie plików w bieżączym katalogu oraz pobranie nowych plików dodanych przez naszych współpracowników. Jeżeli nastąpiły jakieś zmiany w centralnym repozytorium to zostaną pobrane nowe wersje i **cvs** poinformuje nas o tym dodając `U` przed nazwą uaktualnionego pliku:

```
cvs update: Updating cvs-tut
U cvs-tut/cvs-tut.tex
```

2.4.3 Dodawanie plikiów i katalogów

Dodawanie plików i katalogów odbywa się poprzez wykonanie komendy

cvs add nazwa_pliku

Należy pamiętać o zatwierdzeniu tych zmian poleceniem

cvs commit

2.4.4 Usuwanie plików

cvs remove −f nazwa_pliku

Opcja −f polecenia **remove** spowoduje usunięcie pliku z lokalnej kopi repozytorium.

Usuwanie pliku a następnie dodanie do repozytorium z nową nazwą jest jedyną metodą zmieniania nazw plików.

Nie jest możliwe także usuwanie katalogów. Jeżeli nie chcemy pobierać pustych katalogów podczas pobierania modułów to należ dodać opcję −P do poleceń **cvs**:

cvs co −P nazwa_modulu

2.5 Export plików

Polecenie **export** pozwala na pobranie wszystkich plików naszego projektu bez tworzenia lokalnej kopi roboczej. Ma ono opowiązkową opcję −D określającą datę plików które mają zostać pobrane. Przykładowo

cvs export −D 2004−10−14 02:40 moj_projekt

spowoduje pobranie plików w wersjach dodanych do modulu moj_projekt nie później niż 14 Października 2004 o godzinie 02:40. Jeżeli chcemy pobrać najnowszą wersję to możemy wyspecyfikować czas modyfikacji jako today:

cvs export −D today moj_projekt

Opcję −D można zastosować także do komendy **commit**.

Literatura

[1] Strona główna projektu CVS,
https://www.cvshome.org/.

[2] Wikipedia, *Concurrent Versions System*,
http://en.wikipedia.org/wiki/Concurrent_Versions_System.

[3] Wikipedia, *Revision control*,
http://en.wikipedia.org/wiki/Revision_control

[4] Rozdział *CVS*, w: *PDDP – Jak pomóc?*,
http://debian.linux.org.pl/zrobione/pomoc/pomoc.html/ch-cvs.html