

CVS dla programisty lub amatora

CVS dla zwykłych użytkowników

Czym jest CVS i co robi?

CVS to system klient / serwer pozwalający programistom przechowywać swoje projekty w centralnej lokalizacji, zwanej repozytorium. Korzystając z narzędzi klienta cvs, programiści mogą wprowadzać zmiany w zawartości repozytorium. Z kolei repozytorium cvs śledzi każdą zmianę wprowadzoną do każdego pliku, tworząc pełną historię ewolucji projektu deweloperskiego. Programiści mogą żądać starszych wersji konkretnego pliku źródłowego, wyświetlać dziennik zmian i wykonywać inne przydatne zadania w razie potrzeby.

Rola CVS

Wiele projektów otwartego oprogramowania ma własne serwery CVS, które są używane przez programistów jako centralne repozytorium dla wszystkich ich prac. Deweloperzy często ulepszają źródła w repozytorium CVS codziennie; często deweloperzy są rozproszeni po całym świecie, ale CVS zapewnia niezbędny mechanizm do zjednoczenia ich projektu w scentralizowaną, spójną całość. CVS tworzy "klej organizacyjny", dzięki któremu programiści mogą wprowadzać ulepszenia do kodu, nie wchodząc sobie w palce, tracąc ważne dane lub tracąc nawzajem krytyczne aktualizacje do konkretnych plików źródłowych.

CVS - najnowsze źródła programistyczne

Kiedy programiści będą gotowi, przekażą swoją obecną pracę nad CVS w pliku .tar.gz i wydadzą ją jako nową oficjalną wersję swojego pakietu oprogramowania. Jednak ostatnie oficjalne wydanie czasami nie jest wystarczająco aktualne z wielu możliwych powodów. W pierwszej części tego samouczka pokażę, jak używać CVS w tym celu - nabywając najnowszą i najlepszą wersję deweloperską źródeł do własnego użytku osobistego.

CVS - czy je masz?

Zanim będziesz mógł używać CVS, musisz go zainstalować w swoim systemie. Najprostszym sposobem na sprawdzenie, czy jest zainstalowany, jest wpisanie:

```
# cvs
```

Jeśli zostanie znalezione polecenie cvs, masz je! W przeciwnym razie musisz albo wyśledzić pakiet binarny dla danej dystrybucji, albo zainstalować go ze źródeł. Instalowanie CVS ze źródeł jest w rzeczywistości dość proste, a ja pokażę ci, jak w następnym panelu.

Instalowanie CVS ze źródeł

Instalowanie CVS ze źródeł jest łatwe. Najpierw pobierz archiwum cvs-1.11.tar.gz z <ftp://ftp.cvshome.org/pub/cvs-1.11/cvs-1.11.tar.gz> (jeśli jest tutaj wymieniona nowsza wersja, również dobrze możesz zgarnąć nowy zamiast tego.) Następnie wykonaj następujące kroki (wyjście polecenia zostało pominięte dla zwięzłości):

```
# tar xzvf cvs-1.11.tar.gz
```

```
# cd cvs-1.11
```

```
# ./configure
```

robić

make install

Teraz powinieneś być gotowy do pracy.

CVSROOT

Zanim zaczniemy, istnieje kilka podstawowych zasad CVS, które musisz znać. Po pierwsze, aby połączyć się z repozytorium CVS, musisz najpierw znać ścieżkę o nazwie "CVSROOT". CVSROOT to ciąg znaków, podobnie jak adres URL, który informuje polecenie cvs, gdzie znajduje się zdalne repozytorium i jak chcemy się z nim połączyć. Aby zrobić coś interesującego, CVS ma wiele formatów CVSROOT, w zależności od tego, czy repozytorium CVS jest lokalne czy zdalne i jakiej metody użyjesz, aby się z nim połączyć. Oto kilka przykładowych CVSROOTów wraz z objaśnieniami ...

Lokalny CVSROOT

```
CVSROOT = / home / cvsroot
```

To jest przykład lokalnej ścieżki CVSROOT; użyjesz CVSROOT w ten sposób, jeśli chcesz połączyć się z lokalnym repozytorium, które istnieje w / home / cvsroot; a może masz repozytorium zamontowane przez NFS w / home / cvsroot.

Zdalny serwer haseł CVSROOT

```
CVSROOT =: pserver: cvs@foo.bar.com: / home / cvsroot
```

Oto przykład CVSROOT dla zdalnego repozytorium, które istnieje na hoście foo.bar.com i znajduje się w katalogu / home / cvsroot na tym komputerze. Wiodąca część ": pserver:" mówi naszemu klientowi, aby połączyć się z tym zdalnym komputerem za pomocą protokołu serwera haseł CVS, protokołu wbudowanego w CVS. Zazwyczaj publiczne repozytoria CVS korzystają z protokołu serwera haseł, aby umożliwić dostęp do anonimowych użytkowników.

Zdalny CVSROOT rsh / ssh

```
CVSROOT=drobbins@foo.bar.com: / data / cvs
```

Oto przykład CVSROOT używającego protokołu RSH lub SSH; w tym przykładzie serwer CVS spróbuje uzyskać dostęp do repozytorium na foo.bar.com przy użyciu konta drobbins. Jeśli zmienna środowiskowa CVS_RSH jest ustawiona na "ssh", wówczas nasz klient cvs spróbuje użyć ssh do połączenia; w przeciwnym razie zostanie użyty rsh. Metoda dostępu ssh jest popularna wśród osób, które niepokoją się bezpieczeństwem; jednak ani metoda RSH, ani SSH nie zapewnia anonimowym użytkownikom dostępu do źródeł. Aby skorzystać z tej metody, musisz mieć konto logowania na stronie foo.bar.com.

Jeszcze kilka rzeczy ...

Oprócz CVSROOT musisz również znać nazwę modułu (zbiór źródeł), który chcesz sprawdzić, a także anonimowe hasło, które musisz zalogować się do hasła CVS serwer. W przeciwieństwie do anonimowego ftp, nie ma "standardowego" formatu anonimowego hasła, więc musisz uzyskać konkretne hasło od strony internetowej programisty lub od samych programistów. Po uzyskaniu wszystkich tych informacji możesz rozpocząć.

Interakcja z CVS, część 1

Chwywanie źródeł jest procesem dwustopniowym. Najpierw logujemy się do serwera haseł. Następnie pobieramy źródła za pomocą polecenia "kasy". Oto przykładowy zestaw poleceń, które można wykorzystać do sprawdzenia najnowszych źródeł Samby, popularnego projektu integracji UNIX / Windows:

```
# export CVSROOT =: pserver: cvs@pserver.samba.org: / cvsroot
```

To pierwsze polecenie ustawia zmienną środowiskową CVSROOT. Jeśli nie ustawisz tej zmiennej, następujące dwie komendy będą wymagały dodatkowego "-d: pserver: cvs@pserver.samba.org: / cvsroot" po poleceniu "cvs". Eksportowanie CVSROOTA oszczędza nam trochę pisania.

Interakcja z CVS, część 2

Oto polecenia potrzebne do uzyskania aktualnej kopii źródeł programistycznych. Możesz przeskoczyć do następnego panelu, aby przeczytać wyjaśnienie tych poleceń, a następnie wskoczyć tutaj:

```
# cvs zaloguj się
```

(Logowanie na cvs@pserver.samba.org)

Hasło CVS: (podaj tutaj hasło)

```
# cvs -z5 co samba
```

```
U samba / COPYING
```

```
U samba / Manifest
```

```
U samba / README
```

```
U samba / Read-Manifest-Now
```

```
U samba / Roadmap
```

```
U samba / WHATSNEW.txt
```

(to tylko fragment kompletnego wyjścia cvs co)

Interakcja z CVS - wyjaśnienie

Pierwsze polecenie cvs powyżej loguje nas do pserver, a drugie poleca naszemu klientowi CVS, aby wyewidencjonował ("co") moduł samby przy użyciu poziomego kompresji gzip 5 ("-z5"), aby przyspieszyć transfer na wolny link. Dla każdego nowego pliku, który jest tworzony lokalnie, cvs wypisuje "U [ścieżka]" wskazującą, że ten konkretny plik został zaktualizowany na dysku.

Zamówienie zostało zrealizowane

Po zakończeniu pobierania pojawi się katalog "samba" w bieżącym katalogu roboczym, który zawiera najnowsze źródła. Zauważysz również, że wszystkie katalogi mają w sobie katalog "CVS" - CVS przechowuje informacje rozliczeniowe wewnątrz tych katalogów i można je bezpiecznie zignorować. Od tego momentu nie musimy się martwić o ustawienie zmiennej środowiskowej CVSROOT, ani nie musimy jej określać w wierszu poleceń, ponieważ jest ona teraz przechowywana w pamięci podręcznej wewnątrz wszystkich tych dodatkowych katalogów "CVS". Pamiętaj - musisz tylko ustawić CVSROOT na początkowe logowanie i kasę.

Aktualizacja źródeł

Cóż, jesteś - świeże źródła! Teraz, gdy masz już źródła, możesz je kompilować i instalować, sprawdzać lub robić, co chcesz. Od czasu do czasu możesz zsynchronizować swój wyeksploatowany katalog źródłowy z bieżącą wersją na CVS. Aby to zrobić, nie musisz logować się ponownie do pserver; informacje o uwierzytelnieniu są również buforowane przez cvs w tych katalogach księgowych "CVS". Najpierw przejdź do głównego katalogu wymeldowanego (w tym przypadku "samba") i wpisz:

```
# cvs update -dP
```

Patrząc na "aktualizację cvs", część 1

Jeśli są jakieś nowe pliki, cvs wypisze linie "U [ścieżka]" dla każdego z nich podczas ich aktualizacji. Ponadto, jeśli skompilowałeś źródła, prawdopodobnie zobaczysz wiele linii "? [ścieżka]"; są to pliki obiektowe, których cvs nie są ze zdalnego repozytorium.

Patrząc na "aktualizację cvs", część 2

Zwróć także uwagę na dwie opcje wiersza polecenia, które wykorzystaliśmy do "aktualizacji cvs". "-d" mówi cvs, aby utworzył nowe katalogi, które mogły zostać dodane do repozytorium (domyślnie nie jest to możliwe), a "-P" mówi cvs, aby usunął puste katalogi z lokalnie wyrejstrowanej kopii źródła. "-P" jest dobrym pomysłem, ponieważ cvs ma tendencję do zbierania wielu pustych (raz używanych, ale teraz opuszczonych) drzew katalogów w czasie. Jeśli chodzi o pobieranie najnowszych źródeł, to wszystko, co musisz wiedzieć. Teraz przyjrzymy się, jak współpracować z CVS jako programista.

CVS dla programistów

Modyfikowanie plików

Jako programista musisz zmodyfikować pliki w CVS. Aby to zrobić, po prostu wprowadź odpowiednie zmiany w lokalnej kopii repozytorium. Zmiany wprowadzone w źródłach nie są stosowane do zdalnego repozytorium, dopóki wyraźnie nie powiesz cvs, aby "zatwierdził" zmiany. Po przetestowaniu wszystkich modyfikacji, aby upewnić się, że działają one poprawnie i jesteś gotowy do zastosowania zmian w repozytorium, wykonaj ten dwuetapowy proces. Najpierw zaktualizuj źródła, wpisując następujące polecenie w głównym katalogu źródłowym:

```
# cvs update -dP
```

CVS łączy zmiany innych osób

Jak widzieliśmy wcześniej, "aktualizacja cvs" zapewni aktualność źródeł w aktualnej wersji w repozytorium - ale co się stanie ze zmianami, które wprowadziłeś? Nie martw się, nie są wyrzucane. Jeśli inny programista wprowadził zmiany w pliku, którego nie dotknąłeś, plik lokalny zostanie zaktualizowany, aby był zsynchronizowany z wersją w repozytorium. A jeśli zmodyfikowałeś linie 1-10 pliku lokalnego, a inny programista skasował linie 40-50, dodał 12 nowych linii na końcu pliku, zmodyfikował linie 30-40, a następnie zatwierdził swoje zmiany w repozytorium przed tobą, cvs inteligentnie scali te zmiany z lokalnie zmodyfikowaną kopią, aby żadna z twoich zmian nie została utracona. Pozwala to dwóm lub więcej programistom pracować w różnych częściach tego samego pliku w tym samym czasie.

Scalanie nie jest doskonałe

Jeśli jednak dwaj lub więcej deweloperów dokonało zmian w tym samym regionie tego samego pliku, to sprawy nieco się komplikują. Jeśli tak się stanie, wtedy cvs powie ci, że nastąpił konflikt. Żadna praca nie zostanie utracona, ale wymagana będzie niewielka ręczna interwencja, ponieważ cvs wymaga teraz wprowadzenia informacji o sposobie łączenia sprzecznych zmian.

Zatwierdzenie

Przyjrzymy się dokładnie, w jaki sposób można rozwiązać konflikty, ale na razie założmy, że nie ma konfliktów po wpisaniu "cvs update -dP" - zazwyczaj nie ma takich konfliktów. Bez konfliktów lokalne źródła są aktualne i możesz zatwierdzić zmiany w repozytorium, wpisując następujące polecenie w głównym katalogu źródłowym:

```
# commit cvs
```

Co robi commit

"cvs commit" nie tylko stosuje twoje zmiany do repozytorium. Przed wprowadzeniem zmian do zdalnego repozytorium program cvs uruchomi domyślny edytor, aby można było wpisać opis modyfikacji. Po wprowadzeniu komentarzy, zapisaniu pliku i wyjściu z edytora, twoje zmiany (i komentarze) zostaną zastosowane do zdalnego repozytorium i będą dostępne dla innych programistów w twoim zespole.

Wyświetlanie dziennika

Bardzo łatwo jest przejrzeć pełną historię danego pliku wraz z komentarzami, które programiści (w tym ty) mogli wprowadzić podczas zatwierdzania. Aby wyświetlić te informacje, wpisz:

```
# cvs log myfile.c
```

Polecenie "cvs log" jest rekursywne, więc jeśli chcesz zobaczyć pełny log dla całego drzewa katalogów, po prostu wpisz katalog i wpisz:

```
# cvs log | mniej
```

Opcje zatwierdzania

Możesz użyć innego edytora, który domyślnie zaczyna się od cvs po wpisaniu "cvs commit". Jeśli tak, ustaw zmienną środowiskową EDITOR na nazwę edytora, którego chcesz użyć. Ustawienie takiego ustawienia w twoim ~ / .bashrc byłoby dobrym pomysłem:

```
export EDITOR = jpico
```

Alternatywnie można również określić komunikat dziennika jako opcję wiersza poleceń, tak aby cvs nie musiał ładować edytora w pierwszej kolejności:

```
# cvs commit -m 'Naprawiłem kilka głupich błędów w portage.py'
```

Plik .cvsrc

Zanim przejdziemy do dalszych komend cvs, zalecamy ustawienie pliku ~ / .cvsrc. Tworząc plik .cvsrc w swoim katalogu domowym, możesz powiedzieć cvs, aby używał domyślnie preferowanych opcji wiersza poleceń, tak abyś nie musiał pamiętać o wpisywaniu ich za każdym razem. Oto zalecany domyślny plik .cvsrc:

```
cvs -q
```

```
diff -u -b -B
```

```
checkout -P
```

```
update -d -P
```

Plik .cvsrc, kontynuacja

Oprócz ustawienia użytecznych opcji dla kilku poleceń cvs, pierwsza linia .cvsrc umieszcza cvs w trybie cichym, co ma podstawową zaletę, czyniąc wyjście "cvs update" bardziej zwięzłym i czytelnym. Ponadto, jeśli masz już zainstalowany .cvsrc, możesz wpisać "aktualizację cvs" zamiast wpisywać "cvs update -dP".

Dodawanie pliku do repozytorium

Bardzo łatwo jest dodać plik źródłowy do CVS. Najpierw utwórz plik z ulubionym edytorem tekstu. Następnie wpisz następujące polecenie:

```
# cvs dodaj plik myfile.c
```

cvs server: użyj "cvs commit", aby dodać ten plik na stałe

To spowoduje, że cvs doda ten plik do repozytorium następnym razem, gdy wykonasz "commit cvs". Do tego czasu inni deweloperzy nie będą w stanie go zobaczyć.

Dodawanie katalogu do repozytorium

Proces dodawania katalogu do CVS jest podobny:

```
# mkdir foo
```

```
# cvs dodaj foo
```

Katalog / home / cvsroot / mycode / foo dodane do repozytorium

W przeciwieństwie do dodawania pliku, po dodaniu katalogu pojawia się on natychmiast w repozytorium; zatwierdzenie cvs nie jest wymagane. Po dodaniu katalogu lokalnego do cvs, zauważysz, że wewnątrz niego zostanie utworzony katalog "CVS", który będzie służyć jako kontener danych księgowych CVS. W ten sposób można łatwo stwierdzić, czy dany katalog został dodany do CVS, przeglądając go w katalogu "CVS".

cvs dodaje "notatki"

Aha, i jak można się domyślić, zanim dodasz plik lub katalog do repozytorium, upewnij się, że jego katalog nadrzędny został już dodany do CVS. W przeciwnym razie pojawi się błąd, który wygląda następująco:

```
# cvs dodaj plik myfile.c
```

cvs add: nie można otworzyć CVS / wpisów do czytania: Brak takiego pliku lub katalogu

```
cvs [add aborted]: brak repozytorium
```

Zapoznanie się z "aktualizacją cvs", część 1

Zanim przyjrzymy się sposobom rozwiązywania konfliktów, zapoznajmy się z wynikami polecenia "aktualizacja cvs". Jeśli utworzyłeś plik ~ / .cvsrc, który zawiera wiersz "cvs -q", dane wyjściowe "aktualizacji cvs" są znacznie łatwiejsze do odczytania. "update cvs" informuje o tym, co robi i widzi, drukując pojedynczy znak, spację i nazwę pliku; jako przykład:

```
# cvs update -dP
```

```
? distfiles
```

? pakiety

? profile

Zapoznanie się z "aktualizacją cvs", część 2

"aktualizacja cvs" używa "?" znak informujący, że nie wie nic o tych konkretnych plikach, które znajdzie w lokalnej kopii twojego repozytorium. Nie są one oficjalnie częścią repozytorium ani nie zostały zaplanowane do dodania. Oto lista wszystkich innych jednoznakowych wiadomości informacyjnych używanych przez CVS:

U [ścieżka]

Używane, gdy nowy plik zostanie utworzony w lokalnym repozytorium lub zaktualizowany zostanie nietknięty (przez ciebie) plik.

A [ścieżka]

Ten plik jest zaplanowany do dodania i zostanie oficjalnie dodany do repozytorium, gdy wykonasz "zatwierdzenie cvs".

Zapoznanie się z "aktualizacją cvs", część 3

R [ścieżka]

Podobnie jak "A", "R" informuje, że ten plik jest zaplanowany do usunięcia. Plik zostanie usunięty z repozytorium zaraz po "zatwierdzeniu cvs".

M [ścieżka]

Oznacza to, że ten plik został zmodyfikowany przez ciebie; dodatkowo możliwe jest, że nowe zmiany z repozytorium zostały pomyślnie połączone z tym plikiem.

C [ścieżka]

Znak "C" wskazuje, że ten plik zawiera konflikt i wymaga ręcznego poprawienia, zanim będzie można "zatwierdzić zmiany".

Rozwiązywanie konfliktu intro

Teraz przyjrzyjmy się, jak rozwiązać konflikt. Jestem bardzo zaangażowany w projekt Gentoo Linux i mamy własny serwer cvs skonfigurowany na cvs.gentoo.org. Deweloperzy spędzają większość czasu na próbach włączania się do źródeł w module "gentoo-x86". Wewnątrz modułu gentoo-x86 mamy plik o nazwie "ChangeLog", który zawiera (domyślcie się) opis najważniejszych zmian, które wprowadzamy do plików w repozytorium.

Przykładowy konflikt

Ponieważ plik ten jest modyfikowany prawie za każdym razem, gdy programista dokonuje istotnej zmiany w CVS, jest on głównym źródłem konfliktów - oto przykładowy konflikt. Powiedzmy, że dodaję następujące linie na górze listy zmian:

data 25 lutego 2018 r

To jest to, co dodałem sam

Powiedzmy jednak, że zanim będę mógł zatwierdzić te trzy nowe linie, inny programista doda te linie do góry listy zmian i zatwierdza wprowadzone zmiany:

data 25 lutego 2018 r

Jest to część dodana przez innego programistę

Przykładowy konflikt, kontynuacja

Teraz, gdy uruchomię "cvs update -dP" (tak jak powinno przed każdym zatwierdzeniem), cvs nie jest w stanie scalić swoich zmian z moją lokalną kopią ChangeLog, ponieważ oboje dodaliśmy linie do dokładnie tej samej części pliku - jak cvs wie, której wersji użyć? Dostaję następujący błąd z CVS:

Plik RCS: / home / cvsroot / gentoo-x86 / ChangeLog, v

pobieranie wersji 1.362

pobieranie wersji 1.363

łączenie różnic między 1,362 a 1,363 w dzienniku zmian

rcsmerge: warning: konflikty podczas łączenia

serwer cvs: konflikty wykryte w dzienniku zmian

C ChangeLog

Rozwiązywanie konfliktów, część 1

Argh - konflikt! Na szczęście rozwiązywanie konfliktów jest łatwe. Jeśli uruchomię swój ulubiony edytor tekstu, widzę następujący tekst na górze pliku ChangeLog:

<<<<<<< Dziennik zmian

data 25 lutego 2001 r

To jest to, co dodałem sam

=====

data 25 lutego 2001 r

Jest to część dodana przez innego programistę

>>>>>>> 1.363

Rozwiązywanie konfliktów, część 2

Zamiast wybierać jedną wersję z drugiej, cvs dodał obie wersje do pliku ChangeLog i otoczył je specjalnymi separatorami, aby wyraźnie zaznaczyć konflikt, o którym mowa. Teraz muszę zastąpić ten region tekstem, który powinien pojawić się w dzienniku zmian; w tym przypadku tekst zastępczy nie jest ani jedną, ani drugą wersją, ale kombinacją obu:

data 25 lutego 2001 r

To jest to, co dodałem sam

Jest to część dodana przez innego programistę

Teraz, gdy wymieniłem konfliktowy region pliku z odpowiednim tekstem (i usunąłem znaczniki "=====", itp.), Mogę teraz zatwierdzać moje zmiany do CVS bez żadnych problemów.

Wskazówki dotyczące rozwiązywania konfliktów

Ilekcję musisz edytować plik w poszukiwaniu konfliktów, upewnij się, że przeskanujesz cały plik, aby złapać wszystkie; jeśli zapomnisz zająć się konkretnym konfliktem, cvs nie pozwoli ci na zatwierdzenie dopóki nie zostanie rozwiązany! Oczywiście bardzo ważne jest usunięcie specjalnych znaczników dodanych przez cvs do pliku będącego w konflikcie. Kolejna wskazówka - jeśli popełnisz błąd podczas rozwiązywania konfliktu, a następnie ("D'oh!") Przypadkowo zapisz zmiany, możesz znaleźć oryginalną kopię swojej wersji w pliku ". # Filename.version".

Usuwanie pliku

Nadszedł czas, aby nauczyć się naszej ostatniej umiejętności CVS - usuwania plików z repozytorium. Usunięcie pliku jest procesem dwustopniowym. Najpierw usuń plik z lokalnej kopii źródeł, a następnie wykonaj odpowiednie polecenie "cvs remove":

```
# rm myoldfile.c
```

```
# cvs usuwa plik myoldfile.c
```

Usunięcie pliku, ciąg dalszy

Plik zostanie następnie zaplanowany do usunięcia z repozytorium przy następnym wykonaniu zatwierdzenia. Po zatwierdzeniu plik zostanie oficjalnie usunięty z bieżącej wersji repozytorium. Jednak cvs nie wyrzuci tego pliku i zachowa pełny zapis jego zawartości i historii, na wypadek, gdyby był potrzebny w przyszłości. Jest to tylko jeden z wielu sposobów, dzięki którym cvs chroni Twój cenny kod źródłowy. "cvs remove" jest rekurencyjne, co oznacza, że możesz usunąć kilka plików, a następnie uruchomić polecenie "cvs remove" bez żadnych innych argumentów z katalogu nadrzędnego. Wykonanie tej czynności spowoduje oznaczenie wszystkich usuniętych plików do usunięcia podczas następnego zatwierdzenia.

Usuwanie katalogu

Jeśli chcesz usunąć cały katalog, polecam następujący proces. Najpierw usuń fizycznie i "cvs usuń" wszystkie pliki w katalogu:

```
# rm * .c
```

```
# cvs usunąć
```

Usuwanie katalogu, ciąg dalszy

Następnie wykonaj commit:

```
# commit cvs
```

Oto sztuczka. Wykonaj następujące kroki, aby usunąć katalog:

```
# Płyta CD ..
```

```
# cvs remove mydir
```

```
# rm -rf mydir
```

Zauważ, że usunięcie katalogu nie wymagało kolejnego zatwierdzenia - katalogi są dodawane i usuwane z repozytorium w czasie rzeczywistym.