PRI - Projekt

System Zarządzania Dystrybucją

Leszek Krupiński

13 czerwca 2003

Spis treści

Rozdział 1

Opis dziedziny problemowej

PLD (PLD Linux Distribution), polska dystrybucja systemu Linux, oparta jest o system pakietów RPM (RedHat Package Manager). Pakiet jest to program wraz z informacjami o nim: nazwą, opisem, numerem wersji, pakietami od których zależy (to znaczy które muszą być zainstalowane w systemie aby dany pakiet mógł działać). Wszystkie te informacje mogą służyć do wygenerowania pliku „spec", który jest niezbędny do wygenerowania pakietu. Pakiety mogą być kompilowane dla jednej z wielu architektur procesorów, na których może być zainstalowana dystrybucja PLD (i386, i586, i686, Sparc, Alpha, PowerPC). Budowaniem pakietów zarządza specjalny program, tzw. builder. Za pomocą zleceń od developerów dystrybucji kompiluje on żądane pakiety wraz ze wszystkimi pakietami, które od danego zależą. W razie niepowodzenia kompilacji do developerów dystrybucji przekazywane są informacje o tym. Jako że pakiety nie mogą być kompilowane jednocześnie, builder zarządza kolejką zleceń.

Dystrybucja musi zbierać informacje od użytkowników o ewentualnych błędach w funkcjonowaniu pakietów. Dlatego też powinna udostępniać system BTS (Bug Tracking System). Zbiera on informacje od użytkowników przez interfejs WWW.

Rozdział 2

Cel

Efektywny program buildera wyeliminuje konieczność ręcznego zarządzania zleceniami budowania pakietów, a także pozwoli na efektywne wykorzystywanie czasu procesorów maszyn, na których odbywa sie proces kompilacji.

Natomiast BTS pozwoli na szybie i efektywne usuwanie wszelkich błędów w systemie, jak tylko zostaną zauważone.

Rozdział 3

Zakres

Builder w szczególności będzie odpowiedzialny za:

- przechowywanie informacji o pakietach
- przyjmowanie zleceń przebudowania pakietów od developerów
- informowanie o ewentualnym niepowodzeniu kompilacji
- przechowywanie informacji o kolejnych wersjach programów
- zachowywanie zmian w plikach spec w postaci różnicowej
- kontrolę dostępu do systemu
- modyfikację kolejki zleceń przez upoważnionych developerów
- wywoływanie procesu budowania pakietu na maszynach zajmujących się określoną architekturą

Zadania systemu BTS to:

- rejestracja użytkowników dystrybucji
- przyjmowanie zgłoszeń o błędach w funkcjonowaniu aplikacji wraz z opisami
- przydzielanie zgłoszeń do developerów
- przechowywanie informacji o stanie zgłoszenia (oczekujące, przydzielone, rozwiązane, błędne)

Rozdział 4

Kontekst

W systemie występuje 3 aktorów zewnętrznych oraz aktor systemowy.

administrator – osoba zajmująca się administrowaniem systemu, posiada możliwość ingerencji w kolejkę budowania pakietów, rejestrowania developerów w systemie. Zajmuje się też ręcznym przeprowadzaniem operacji nieprzewidzianych w systemie.

developer – aktor będący osobą zajmującą się rozwojem dystrybucji. Developer tworzy pakiety, dodaje informacje o nich, zajmuje się także błędami przydzielonymi mu poprzez system BTS. Developer może mieć status opiekuna pakietu. W takim przypadku, tylko od niego zależy uznanie danej wersji pakietu za stabilną, ma też prawo cofać zmiany w specu zrobione przez innych developerów.

użytkownik – użytkownik systemu nie będący twórcą dystrybucji. Po zarejestrowaniu w systemie posiada on możliwość zgłaszania błędów w funkcjonowaniu poszczególnych pakietów dystrybucji. Użytkownik wypełnia formularz zawierający wszystkie informacje o pakiecie: wersję, opis problemu, jego typ.

Rozdział 5

Opis wymagań

5.1 Wymagania funkcjonalne

1. System ma przechowywać informacje o developerach dystrybucji. Na podstawie tych danych możliwe jest rejestrowanie zmian w specach, przydzielanie opiekunów pakietów, a także wysyłanie zleceń wstawienia pakietu do kolejki do zbudowania.

2. System ma także służyć jako baza danych informacji o pakietach. Wszystkie pakiety muszą posiadać opisy, które mogą być wyświetlone na przykład poprzez interfejs WWW. Dodatkowo, każdy pakiet musi posiadać informacje o najnowszej wersji stabilnej i niestabilnej.

3. W systemie powinny być zapisane informacje o plikach, na które składają się pakiety (czyli kody źródłowe programów, skrypty itp). Przy każdym pliku powinna być zapisana jego suma kontrolna MD5 (dzięki czemu można sprawdzić, czy plik został przesłany poprawnie), URL (Uniform Resource Locator) miejsca w internecie, skąd został ten plik oryginalnie pobrany a także jego wersja.

4. Wszystkie modyfikacje w specach powinny być przechowywane w formacje różnicowym, aby można było łatwo te zmiany cofnąć. Każda zmiana powinna posiadać swój numer rewizji, datę, oraz zapisany identyfikator developera wykonującego zmianę.

5. Developer ma mieć możliwość wstawienia pakietu do kolejki do zbudowania dla konkretnej architektury bądź dla wszystkich możliwych. System powinien automatycznie zbudować wszystkie pakiety, które zależą od danego pakietu.

6. Administrator ma mieć możliwość dowolnego przestawiania pakietów oczekujących w kolejce na zbudowanie, a zwłaszcza wstawienie pakietu bezwarunkowo na początek kolejki. Powinien mieć także możliwość usuwania pakietów z kolejki jeśli zajdzie taka potrzeba (na przykład pakiet został wstawiony do kolejki przez przypadek).

7. W systemie mają być przechowywane wszelkie informacje na temat każdej próby zbudowania pakietu, zwłaszcza jeśli zakończyła się ona niepowodzeniem. Informacja o niepowodzeniu procesu powinna zostać przesłana do osoby zlecającej budowanie.

8. Zarejestrowani użytkownicy dystrybucji powinni mieć możliwość zgłaszania informacji o błędach. Zgłoszenia te (tickety) powinny być opatrzone odpowiednim komentarzem, powinny być przypisane do konkretnej wersji programu. System powinien umożliwiać przypisanie ticketu do konkretnego developera, który zajmie się problemem. Każdy z ticketów powinien mieć określony stan - czy problem oczekuje na rozwiązanie, jest w trakcie rozwiązania, czy może nie da się zaobserwować podobnego błędu.

9. Co miesiąc powinien być przygotowywany raport, zawierający informacje o wszystkich pakietach, których najnowszych wersji nie udało się zbudować.

5.2 Wymagania niefunkcjonalne

1. opiekunem pakietu może być tylko developer o stażu większym niż rok

2. stabilność pakietu może ustalić tylko jego opiekun

3. użytkownik musi przy rejestracji podać prawdziwy, istniejący adres e-mail

Rozdział 6

Ewolucja systemu

W przyszłości system ten może zintegrować wszystkie aspekty tworzenia dystrybucji. Dzięki temu administracja pakietami i kontami developerów zostanie scentralizowana.

System można poszerzyć o system CVS (Concurrent Versioning System). System ten służy do usprawnienia pracy grupowej nad wspólnym projektem. Umożliwia on pracę wielu developerów nad jednym plikiem, zapobiegając wzajemnemu „przeszkadzaniu”.

System ten może w przyszłości służyć także do wymiany informacji między developerami. Integracja systemu umożliwi stworzenie wielu metod dostępu do dyskusji: przez interfejs WWW, pocztowy czy NNTP.

System Zarządzania Dystrybucją może także służyć pomocą dla CDG (Core Developers Group) - grupy developerów podejmujących najważniejsze dla dystrybucji decyzje. System może być rozbudowany o system do głosowań.

Rozdział 7

Słownik

dystrybucja – zbiór pakietów oraz jądro systemu Linux

developer – osoba rozwijająca dystrybucję PLD

administrator – osoba zarządzająca Systemem Zarządzania Dystrybucją

RPM – RedHat Package Manager - system zarządzania pakietami

pakiet – skompilowany program lub kod źródłowy, wraz z informacjami niezbędnymi do jego instalacji – informacje o pakietach których dany pakiet wymaga, opis, skrypty które mają być wykonane po instalacji, przed nią lub po odinstalowaniu pakietu.

spec – zbiór informacji niezbędnych do zbudowania pakietu

BTS – Bug Tracking System – system śledzenia błędów

ticket – notatka o błędzie występującym w pewnym pakiecie; zawiera informacje o wersji tego pakietu, okolicznościach zajścia, osobie zgłaszającej i developerze, który jest przydzielony do tego błędu