



UNIwersytet MIKOŁAJA KOPERNIKA  
w TORUNIU



Wydział Matematyki i Informatyki  
Wydział Fizyki, Astronomii  
i Informatyki Stosowanej



---

Marcin Rzewucki

## Zarządzanie pakietami oprogramowania

*Praca magisterska napisana  
pod kierunkiem dra hab. Jacka Kobusa*

TORUŃ 2005

# Spis treści

<b>Wstęp</b>	<b>4</b>
<b>1 Zarządzanie oprogramowaniem</b>	<b>7</b>
1.1 Wprowadzenie . . . . .	7
1.2 Konwencje nazewnictwa pakietów RPM . . . . .	9
1.3 Zawartość pakietów RPM . . . . .	10
1.4 Numeracje wersji programów . . . . .	11
<b>2 Budowa modułu FOLAPackages</b>	<b>13</b>
2.1 Architektura . . . . .	13
2.2 Struktura katalogowa . . . . .	17
2.3 Konfiguracja . . . . .	17
<b>3 Implementacja modułu FOLAPackages</b>	<b>18</b>
3.1 Wprowadzenie . . . . .	18
3.2 Schemat pracy programów . . . . .	19
3.3 Operacje na plikach . . . . .	20
3.4 Interfejs użytkownika . . . . .	21
3.5 Dzienniki zdarzeń . . . . .	23
3.6 Porównywanie wersji pakietów . . . . .	24
3.7 Repozytorium . . . . .	25
3.7.1 Tworzenie repozytorium . . . . .	25
3.7.2 Aktualizowanie . . . . .	31
3.8 Ustalanie zależności dla pakietów . . . . .	34
3.9 Nadzorca pakietów . . . . .	36
3.9.1 Rejestracja stacji roboczych . . . . .	36
3.9.2 Przygotowywanie zadań . . . . .	37
3.9.3 Pomocnicy . . . . .	38
3.10 Klient pakietów . . . . .	39
3.10.1 Przygotowywanie zleceń . . . . .	39
3.10.2 Odtwarzanie uszkodzonej bazy pakietów . . . . .	39
3.10.3 Pomocnicy . . . . .	39
<b>Podsumowanie</b>	<b>41</b>
<b>A WYMAGANIA</b>	<b>42</b>

---

<b>B</b>	<b>INSTALACJA</b>	<b>44</b>
<b>C</b>	<b>STRUKTURA KATALOGOWA</b>	<b>48</b>
<b>D</b>	<b>PLIK KONFIGURACYJNY</b>	<b>51</b>
<b>E</b>	<b>TESTY</b>	<b>55</b>
<b>F</b>	<b>CD-ROM</b>	<b>58</b>
<b>G</b>	<b>DOKUMENTACJA FOLAPackages</b>	<b>60</b>
G.1	Client.pm . . . . .	60
G.2	Config.pm . . . . .	61
G.3	Database.pm . . . . .	63
G.4	Interface.pm . . . . .	66
G.5	Logs.pm . . . . .	70
G.6	Repository.pm . . . . .	72
G.7	Updater.pm . . . . .	77
G.8	Versions.pm . . . . .	80
G.9	Workers.pm . . . . .	82
	<b>Spis literatury</b>	<b>90</b>

# Wstęp

Celem projektu **FOLA**<sup>1</sup> jest stworzenie modułowego, w szerokim zakresie konfigurowalnego systemu do sprawnego i wygodnego nadzorowania i zarządzania grupą serwerów oraz stacji roboczych. Zakłada się, aby podległe maszyny pracowały pod kontrolą tego samego systemu GNU<sup>2</sup>/Linux<sup>3</sup> z oprogramowaniem dostarczanym w postaci plików RPM<sup>4</sup>. W ramach projektu przewiduje się zbudowanie oprogramowania m.in. do rejestrowania, monitorowania, konfigurowania i archiwizowania danych wszystkich komputerów w zarządzanej sieci lokalnej. Każda z części systemu, jako odrębny moduł, działa samodzielnie, ale w oparciu o wspólne (co do zawartości i formatu) dane konfiguracyjne. Poszczególne moduły posiadają własne interfejsy, ale architektura modułów powinna umożliwić stworzenie osobnego programu (modułu centralnego) stanowiącego jednolity interfejs.

Zarządzanie oprogramowaniem jest prawdopodobnie jedną z najczęstszych czynności jakie wykonuje administrator. Praca z setkami różnych pakietów, które mają być dodawane, usuwane oraz aktualizowane na komputerach w nadzorowanej sieci, jest bardzo mozolna i czasochłonna. Między programami istnieją liczne powiązania, które sprawiają, że instalacja wybranego programu powoduje konieczność instalacji dodatkowych programów lub bibliotek. Ustalenie tych powiązań wymaga przechowywania informacji o każdym z programów. Pominięcie w trakcie instalacji zależności występujących między programami/bibliotekami zwykle uniemożliwia poprawne działanie programu. Śledzenie zaś zależności jest niezmiernie skomplikowanym zagadnieniem, które utrudnia administratorowi sprawne zarządzanie oprogramowaniem. Dodatkowym problemem może być spełnienie zależności przy aktualizacji oprogramowania, jak również przy próbie instalowania dodatkowych programów spoza dystrybucji systemu GNU/Linux. Użytkownikom systemów komputerowych zależy na tym, aby oprogramowanie, z którego korzystają było aktualne. Jeśli są to programy dostarczane w ramach danej dystrybucji, to proces ich aktualizacji jest stosunkowo łatwy i daje się zautomatyzować. W przypadku programów spoza dystrybucji na administratora spada obowiązek śledzenia nowych wersji tych programów i ich uaktualnianie.

---

<sup>1</sup>the Friend Of a Lazy Administrator – przyjaciel leniwego administratora

<sup>2</sup>GNU's Not Unix – GNU nie jest Uniksem [2]

<sup>3</sup>Linux Is Not UNIX – Linux nie jest Uniksem [2]

<sup>4</sup>the RPM Package Manager – menadżer pakietów RPM [2]

Z powyższego opisu wynika, że zarządzanie oprogramowaniem na pojedynczej stacji roboczej jest dość złożonym zagadnieniem. Sprawa komplikuje się jeszcze bardziej przy próbie centralnego administrowania oprogramowaniem nie identycznie skonfigurowanych serwerów i stacji roboczych. Trudność dodatkową stwarza pogodzenie potrzeb administratorów pojedynczych serwerów lub grup stacji roboczych z ideą centralnego nadzoru nad całością zasobów komputerowych. Z taką sytuacją możemy mieć często do czynienia w sieciach większych placówek naukowo-badawczych.

Celem niniejszej pracy jest stworzenie odrębnego systemu umożliwiającego wygodne administrowanie pakietami oprogramowania przez administratorów serwerów i stacji roboczych w dużych sieciach komputerowych. Zaproponowano, aby instalacja programów na nadzorowanych maszynach odbywała się przy wykorzystaniu centralnego serwera, który nie tylko pełni rolę repozytorium pakietów (zarówno ze wspieranej dystrybucji jak i spoza niej), ale przede wszystkim dostarcza bazy danych o zależnościach między tymi pakietami. To rozwiązanie umożliwia również głównemu administratorowi na przeprowadzanie niezbędnych zmian oprogramowania, takich jak instalacja, aktualizacja oraz usuwanie pakietów w odniesieniu do wybranego komputera lub wybranej grupy komputerów w sieci.

Architektura proponowanego systemu wykorzystuje mechanizm typu klient-serwer. Dowolna stacja robocza staje się klientem z chwilą zarejestrowania na głównym serwerze. Proces rejestracji obejmuje utworzenie na serwerze katalogu domowego użytkownika związanego z daną stacją i umożliwienie mu łączenia się z serwerem w bezpieczny sposób bez konieczności podawania hasła (protokół SSH). Dzięki temu klient może pozostawiać na serwerze zlecenia i w ten sam sposób pobierać gotowe do wykonania zadania. Serwer z zadaną częstotliwością (lub na wyraźne żądanie administratora) przegląda katalogi klientów w poszukiwaniu nowych zleceń i w miarę możliwości je realizuje. Rozważmy przypadek, w którym administrator stacji roboczej chce zainstalować dodatkowy pakiet. Serwer po przyjęciu zlecenia, w oparciu o posiadaną bazę danych o zainstalowanych na kliencie pakietach, sprawdza zależności i przygotowuje listę pakietów do zainstalowania w postaci gotowych do wykonania przez klienta komend. Klient instaluje wszelkie potrzebne pakiety pobierając je z udostępnianego przez serwer repozytorium. W celu ustalenia zależności między instalowanymi pakietami wykorzystywane są dwie bazy danych: systemowa baza pakietów RPM oraz związana z nią baza informacyjna. Wszelkie operacje wykonywane po stronie klienta i serwera są rejestrowane w odpowiednich dziennikach zdarzeń, które mogą być przeglądane zarówno przez administratora stacji klienckiej jak i administratora serwera.

Do implementacji przedstawionej wyżej architektury systemu użyty został język programowania Perl, gdyż doskonale nadaje się do manipulowania tekstami i programowania sieciowego. Ponadto, jego możliwości są nieustannie rozszerzane o starannie konstruowane przez programistów moduły dostępne w archiwum CPAN (*Comprehensive Perl Archive Network*) [2]. Na przykład

do budowy interfejsu użytkownika wykorzystano moduł do obsługi biblioteki systemowej ncurses. To sprawia, że Perl jest niezwykle użytecznym narzędziem w pracy administratora, gdyż pozwala mu się skupić głównie na rozwiązywanym problemie. W trakcie tworzenia oprogramowania okazało się, że do przechowywania informacji o pakietach i ich zależnościach wygodnie jest wykorzystać technologię XML (*Extensible Markup Language*) [2]. Do manipulowania zawartością plików w formacie XML wykorzystano odpowiednie moduły z dystrybucji Perla oraz zaawansowane możliwości jego wyrażeń regularnych.

Układ pracy jest następujący. Po wstępie, w rozdziale pierwszym wyjaśniam czym są pakiety RPM oraz jak się oznacza wersje programów. Rozdział drugi jest w całości poświęcony dokładnemu przedstawieniu proponowanej architektury systemu do zarządzania pakietami RPM. Zasadniczą część pracy stanowi kolejny rozdział, w którym została szczegółowo omówiona implementacja modułu. Pracę kończy krótkie podsumowanie, po którym następują uzupełnienia zawierające m.in. opis instalacji systemu, jego pliku konfiguracyjnego oraz dokumentacja POD (*Plain Old Documentation*) stworzonego oprogramowania.

Do niniejszej pracy została dołączona płyta CD-ROM zawierająca jej elektroniczną wersję, stworzone przeze mnie oprogramowanie, dokumentację, a także dodatkowe oprogramowanie potrzebne do instalacji/odinstalowania systemu oraz jego uruchomienia. Oprogramowanie jest udostępniane w ramach licencji GNU GPL<sup>5</sup>.

---

<sup>5</sup>General Public License – powszechna licencja publiczna [1]