

UNIwersytet MIKOŁAJA KOPERNIKA

WYDZIAŁ MATEMATYKI I INFORMATYKI  
WYDZIAŁ FIZYKI, ASTRONOMII I INFORMATYKI  
STOSOWANEJ  
INSTYTUT FIZYKI

Tomasz Radecki

Nr albumu: 176045

Praca magisterska  
na kierunku Informatyka

Zarządzanie kopiami zapasowymi danych użytkownika

*Praca wykonana pod kierunkiem  
dr. hab. Jacka Kobusa  
Zakład Mechaniki Kwantowej*

TORUŃ 2006

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>3</b>
<b>2</b>	<b>Tworzenie kopii zapasowych</b>	<b>6</b>
2.1	Strategie tworzenia kopii zapasowych . . . . .	7
2.1.1	Dlaczego i co archiwizować? . . . . .	7
2.1.2	Jak i kiedy archiwizować? . . . . .	9
2.1.3	Rodzaje kopii zapasowych . . . . .	9
2.1.4	Strategia tworzenia kopii zapasowych . . . . .	10
2.2	Linuksowe narzędzia do archiwizacji . . . . .	12
2.2.1	tar . . . . .	13
2.2.2	cpio . . . . .	13
2.2.3	dump/restore . . . . .	14
2.2.4	dd . . . . .	15
2.2.5	pax . . . . .	15
2.2.6	rsync . . . . .	15
2.2.7	Podsumowanie . . . . .	16
2.3	Systemy archiwizujące . . . . .	18
<b>3</b>	<b>Architektura modułu BMM</b>	<b>22</b>
3.1	Architektura klient-serwer . . . . .	23
3.2	Struktura katalogów . . . . .	25
3.3	Komunikacja . . . . .	26
3.4	Konfiguracja . . . . .	26
3.5	Narzędzia . . . . .	26
3.6	Zalety architektury . . . . .	27
<b>4</b>	<b>Implementacja modułu BMM</b>	<b>29</b>
4.1	Wprowadzenie . . . . .	29
4.2	Rejestracja i instalacja . . . . .	30
4.2.1	Rejestracja w bazie hostów systemu FOLA . . . . .	30
4.2.2	Rejestracja w systemie BMM . . . . .	31

4.3	Schemat pracy programów . . . . .	33
4.3.1	Program nadzorcy . . . . .	35
4.3.2	Program klienta . . . . .	37
4.4	Dzienniki zdarzeń . . . . .	39
4.5	Interfejs użytkownika . . . . .	40
4.6	Struktura baz danych . . . . .	41
4.7	Realizacja zleceń . . . . .	45
4.7.1	Klienci . . . . .	45
4.7.2	Dyski . . . . .	46
4.7.3	Archiwa . . . . .	47
4.8	Wymagania . . . . .	48
<b>5</b>	<b>Podsumowanie</b>	<b>49</b>
<b>A</b>	<b>Struktura katalogów modułu BMM</b>	<b>53</b>
<b>B</b>	<b>Pliki konfiguracyjne modułu FOLA::BMM</b>	<b>58</b>
<b>C</b>	<b>Dokumentacja modułu BMM</b>	<b>61</b>
C.1	BMM::Server . . . . .	61
C.1.1	BMMs-drive.pl . . . . .	61
C.1.2	BMMs-host.pl . . . . .	66
C.1.3	BMMs-examiner.pl . . . . .	71
C.1.4	BMMs-executor.pl . . . . .	73
C.1.5	BMMs-processor.pl . . . . .	75
C.1.6	Server.pm . . . . .	77
C.1.7	Common.pm . . . . .	80
C.2	BMM::Client . . . . .	82
C.2.1	BMMc-backup.pl . . . . .	82
C.2.2	BMMc-drive.pl . . . . .	85
C.2.3	BMMc-examiner.pl . . . . .	88
C.2.4	BMMc-executor.pl . . . . .	90
C.2.5	BMMc-mount.pl . . . . .	92
C.2.6	Client.pm . . . . .	93

# Rozdział 1

## Wstęp

Pewnie jedną z ostatnich rzeczy, o których myślimy wyłączając codziennie komputer, to zabezpieczanie danych. Tak jednak jest do chwili, aż stracimy cenne pliki. . . Tworzenie kopii zapasowych to – wbrew pozorom – jedno z trudniejszych, ważniejszych i bardziej odpowiedzialnych zadań związanych z administrowaniem systemem komputerowym. Kopia zapasowa to ostatnia deska ratunku w przypadku awarii sprzętu, naruszenia bezpieczeństwa systemu, uszkodzenia danych wskutek błędów oprogramowania, systemu czy też użytkownika. Choć coraz częściej dane są gromadzone nie na pojedynczych dyskach, lecz na macierzach dyskowych, a nadmiarowość danych pozwala na uniknięcie kłopotów w przypadku awarii jednego z dysków, to ten sposób zabezpieczeń nie eliminuje całkowicie ryzyka utraty danych.

Celem projektu FOLA (*the Friend Of a Lazy Administrator*) jest zbudowanie modułowego i w dużej mierze konfigurowalnego systemu wspomagającego zarządzanie grupą serwerów i stacji roboczych. Z założenia FOLA ma być na tyle elastycznym systemem, aby umożliwiać nadzorowanie nie tylko niewielkiej sieci firmowej (biurowej) lub szkolnej, ale także np. sieci akademickiej obejmującej kilka wydziałów. Wspomniana budowa modułowa to efekt rozdzielenia i pogrupowania pewnych funkcji jakie system powinien oferować. M.in. w systemie FOLA wydzielono moduły: zarządzanie kontami użytkowników, monitorowanie stanu serwerów i stacji roboczych, zarządzanie pakietami oprogramowania, nadzorowanie hostów i lokalnej sieci komputerowej pod kątem bezpieczeństwa sieciowego, a także moduł do tworzenie i zarządzania kopiami zapasowymi. Każdy z wymienionych modułów z założenia może pracować samodzielnie, ale docelowo wszystkie moduły powinny działać pod nadzorem specjalnego modułu centralnego umożliwiającego, za pomocą ujednoliczonego interfejsu użytkownika, konfigurację i zarządzanie każdym z podsistemów z osobna. Domyślną platformą, na której ma praco-

wać system FOLA jest GNU/Linux, jednak system ma umożliwiać również w pewnym zakresie administrację stacjami pracującymi pod nadzorem systemu Microsoft Windows.

Głównym celem niniejszej pracy jest stworzenie rozproszonego systemu zarządzania archiwami użytkownika. System ten ma ułatwić administratorowi czynności związane z archiwizacją danych pochodzących od wielu maszyn. Poprzez ułatwienie rozumiemy automatyzację tej części zadań, która nie wymaga bezpośredniej interwencji administratora systemu. Maszyna, na której zainstalowany jest moduł nadzorcy, pełni rolę serwera dla zarejestrowanych w systemie klientów. Serwer przyjmuje i przetwarza zlecenia nadsyłane przez klientów systemu. Moduł klienta oraz moduł nadzorcy mogą być zainstalowane na tej samej maszynie, dzięki czemu maszyna-serwer może być jednocześnie klientem systemu, którym zarządza. Możliwe jest także jednoczesne występowanie kilku serwerów w danej sieci.

Do budowy systemu archiwizacji FOLA BMM (*Backup Management Module*) wykorzystano architekturę klient-serwer. Rejestracja klientów, rejestracja zasobów do archiwizowania, instalacja oprogramowania klienta, archiwizacja jak i przydział zasobów do archiwizowania, nadzorowane są przez moduł serwera. Pozwala to na łatwiejsze zarządzanie i śledzenie poprawności działania systemu. Dołączenie do systemu wymaga w pierwszej kolejności wysłania ze stacji roboczej zgłoszenia rejestracyjnego, które później musi zostać zaakceptowane przez administratora systemu. Dodanie nowego klienta do systemu powoduje utworzenie dla niego konta i katalogu domowego na serwerze. Połączenie z własnym kontem na serwerze, bez konieczności podawania hasła, możliwe jest dzięki przesłanemu podczas rejestracji kluczowi SSH. Taki kanał komunikacji jest wystarczający do wymiany plików zleceń od klienta oraz paczek z poleceniami od serwera – wymiana odbywa się w całości w katalogu domowym klienta na serwerze. Zadania są przetwarzane przez moduł nadzorcy w kolejności nadsyłania – specjalny program w ustalonych odstępach czasu dokonuje inspekcji katalogów domowych klientów na serwerze w poszukiwaniu plików zleceń. Jeżeli wykonanie zadania jest możliwe (dostępne są zasoby), to jest ono realizowane, w przeciwnym wypadku program przechodzi do następnego zlecenia. W sytuacjach wyjątkowych, np. gdy zlecenie straciło na aktualności, utknęło w trakcie realizacji lub kiedy brakuje przestrzeni dyskowej, moduł nadzorcy powiadamia administratora systemu za pomocą listu elektronicznego. W zamyśle system ma wykorzystywać do przechowywania archiwów dyski twarde, modułarna architektura powinna jednak umożliwiać w przyszłości wykorzystanie również innych nośników da-

nych. Według początkowych planów system miał pokrywać o wiele większy zakres funkcjonalności, m.in. miał być wyposażony graficzny interfejs użytkownika, kilka algorytmów sortujących zlecenia w zależności od trybu pracy lub pory dnia, implementacja schematów i szablonów archiwizacji. W wyniku realizacji projektu udało się jednak stworzyć system o otwartej, modułowej budowie, który pozwala na przeprowadzanie archiwizacji oraz odtwarzania wskazanych zasobów.

Do implementacji systemu został wykorzystany język Perl. Perl jest językiem programowania wysokiego poziomu wywodzącym się po części z języka C, a po części z sed, awk, języka powłoki Uniksa, i przynajmniej z tuzina innych narzędzi i języków. Jego zdolność do wydajnego przetwarzania tekstu czynią z niego doskonałe narzędzie do szybkiego tworzenia prototypów, narzędzi systemowych, narzędzi programowych, interfejsu bazodanowego, programowania grafiki i programowania sieciowego. Nie sposób nie wspomnieć bogatym archiwum CPAN (*Comprehensive Perl Archive Network*), pełnego gotowych rozwiązań i użytecznych narzędzi. Perl istnieje od wczesnych lat 90., jest wyjątkowo dobrze znany, a duża ilość darmowych samouczków i stron pomocy w internecie tworzy pokaźne wsparcie w razie napotkanych problemów. Wspomniane zalety czynią ten język szczególnie popularnym wśród administratorów systemowych i programistów skryptów CGI oraz, co ciekawe, wśród matematyków, genetyków, dziennikarzy i menedżerów.

Praca rozpoczyna się od przedstawienia pewnych podstawowych zagadnień związanych z tworzeniem, planowaniem i zarządzaniem kopiami zapasowymi (rozdział 2). Omówione zostaną sposoby tworzenia kopii zapasowych, algorytmy zarządzania, na podstawie których planowana jest strategia zabezpieczania danych, a także powszechnie dostępne w środowisku uniksowym narzędzia służące zabezpieczeniu i archiwizowaniu danych. Następnie, w rozdziale 3 przyjrzymy się ogólnie architekturze modułu, który służy do zarządzania kopiami zapasowymi, a w rozdziale 4, zapoznamy się ze szczegółami implementacji tego modułu. Całość modułu klienta i serwera została udokumentowana w dodatkach na końcu pracy. Kod został udokumentowany w formacie POD (*Plain Old Documentation*) w dodatku C, oprócz tego w dodatku A można znaleźć strukturę katalogów z lokalizacją poszczególnych plików, a pliki konfiguracyjne w dodatku B.

Kod modułu, który jest przedmiotem niniejszej pracy rozpowszechniany jest na zasadach licencji GNU GPL [1].