

UNIwersytet MIKOŁAJA KOPERNIKA

WYDZIAŁ MATEMATYKI I INFORMATYKI
WYDZIAŁ FIZYKI, ASTRONOMII I INFORMATYKI
STOSOWANEJ

Marcin HENRYKOWSKI

Nr albumu: 158069

Praca magisterska
na kierunku Informatyka

Archiwizacja systemów rodziny Windows

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

TORUŃ 2008

Spis treści

1	Wstęp	5
2	Archiwizacja urządzeń blokowych	8
2.1	Czym jest archiwizacja urządzeń blokowych	8
2.2	Porównanie ze standardowymi metodami archiwizacji	10
2.3	Dodatkowe możliwości	12
2.4	Wymagania funkcjonalne dla implementacji	16
3	Architektura modułu FolaPA	22
3.1	Elementy architektury	23
3.1.1	Operacje na urządzeniach blokowych	23
3.1.2	Przetwarzanie strumienia danych	24
3.1.3	Przetwarzanie zbiorów archiwum	25
3.1.4	Operacje na zbiorach identyfikowanych przez URI	26
3.1.5	Interfejs linii poleceń	27
3.2	Format zbioru archiwum	29
3.2.1	Ogólna struktura zbioru	29
3.2.2	Znaczniki opisujące dane	31
4	Implementacja modułu FolaPA	38
4.1	Oprogramowanie tworzące bazę dla implementacji	38
4.2	Moduły Partition Archivist	40
4.2.1	PA::InOut	40
4.2.2	PA::Filters	41
4.2.3	PA::XMLpd	43
4.2.4	PA::Image	44
4.2.5	PA::Tools	49
4.3	Skrypty Partition Archivist	52
4.3.1	pacreate	52
4.3.2	parestore	53
4.3.3	pafind	54
4.4	Instalator Partition Archivist	55

5	Podsumowanie	56
II	Załączniki	61
A	Lista użytych modułów	62
A.1	Moduły standardowe Perl	62
A.1.1	Cwd	62
A.1.2	Digest::MD5	62
A.1.3	Fcntl	62
A.1.4	File::Spec	63
A.1.5	FindBin	63
A.1.6	Getopt::Long	63
A.1.7	Pod::Usage	63
A.1.8	Sys::Hostname	63
A.2	Moduły CPAN	64
A.2.1	Compress::Bzip2	64
A.2.2	Compress::Zlib	64
A.2.3	Net::SFTP::Foreign::Compat	64
A.2.4	Net::SFTP::Foreign::Attributes	64
A.2.5	Net::SFTP::Foreign::Constants	64
A.2.6	Net::SFTP::Foreign::Util	64
A.2.7	URI	65
A.2.8	URI::file;	65
A.2.9	URI::sftp;	65
B	Wymagane pakiety oprogramowania	66
B.1	Dodatkowe pakiety Perl	66
B.1.1	Compress-Bzip2	66
B.1.2	Compress-Zlib	66
B.1.3	Compress-Raw-Zlib	67
B.1.4	Digest-MD5	67
B.1.5	IO-Compress-Base	67
B.1.6	IO-Compress-Zlib	67
B.1.7	MIME-Base64	68
B.1.8	Net-SFTP-Foreign	68
B.1.9	URI	68
B.1.10	URI-sftp	68
B.2	Inne pakiety oprogramowania	68
C	Przykłady wywołania skryptów FolaPA	70

D Dokumentacja POD	74
D.1 pcreate	74
D.2 pafind	77
D.3 parestore	80
D.4 PA::InOut	82
D.5 PA::InOut::SFTP	90
D.6 PA::Filters	97
D.7 PA::XMLpd	109
D.8 PA::Image	111
D.9 PA::Tools	143

Rozdział 1

Wstęp

Każdy, kto choć raz utracił dane zapisane w obrębie systemu komputerowego, doskonale zdaje sobie sprawę jak istotną sprawą jest regularne tworzenie kopii zapasowych. Archiwizacja danych zapisanych w systemie komputerowym jest jednym z podstawowych sposobów na zabezpieczenie przed utratą, a nierzadko staje się ostatnią deską ratunku przy próbie ich odtworzenia. Niestety użytkownicy większych sieci komputerowych, takich jak sieć w środowisku akademickim lub korporacyjnym obarczają administratorów odpowiedzialnością za zabezpieczenie przechowywanych informacji. Dlatego istnieje potrzeba rozwijania zaawansowanych narzędzi wspierających tworzenie i odtwarzanie kopii bezpieczeństwa w dużych heterogenicznych sieciach komputerowych.

Standardowo pojęta archiwizacja polega na utworzeniu kopii poszczególnych zbiorów w obrębie systemu plików. Oznacza to, że narzędzia wykorzystywane do standardowej archiwizacji wymagają dostępu do struktury katalogów i plików zarówno w trybie do odczytu jak i zapisu. Oczywiście nie stanowi to problemu dla aplikacji działających pod kontrolą tego samego systemu operacyjnego, który był użyty podczas tworzenia archiwizowanych danych. Należy jednak zauważyć, że takie podejście ma bardzo poważną wadę. Mianowicie, użycie programu archiwizującego do odtworzenia kopii zapasowej wymaga sprawnego działania wspomnianego systemu operacyjnego. Oznacza to, że w przypadku poważnej awarii, w wyniku której uszkodzony został również system operacyjny, odtworzenie kopii bezpieczeństwa może okazać się bardzo uciążliwe. Dlatego jest wskazane, aby narzędzia do tworzenia i odtwarzania kopii bezpieczeństwa były niezależne od archiwizowanego systemu operacyjnego. Niestety, takie podejście rodzi kolejne problemy. Przykładem na to jest archiwizacja systemów z rodziny *Windows* wykorzystujących zaawansowany system plików *NTFS*. Restrykcyjne umowy licencyjne ograniczające dostęp do pełnej specyfikacji *NTFS* przez długi czas utrudniały rozwój otwartego i darmowego oprogramowania w pełni wspierającego ten system plików. Dlatego standardowa archiwizacja systemów *Windows* z poziomu innego

systemu operacyjnego była niemożliwa bez wykorzystania rozwiązań komercyjnych.

Celem niniejszej pracy jest zaprezentowanie odmiennego podejścia do problemu tworzenia kopii bezpieczeństwa, mianowicie poprzez archiwizację partycji w trybie surowym. Archiwizacja ta polega na tworzeniu niskopoziomowych obrazów całej partycji bądź dysku twardego. Pozwala to całkowicie uniezależnić proces tworzenia i odtwarzania kopii bezpieczeństwa od struktury archiwizowanych zbiorów, zastosowanego systemu plików czy też systemu operacyjnego. Takie podejście umożliwia zabezpieczenie danych znajdujących się na partycjach *NTFS* systemu *Windows* przy użyciu przyjaznego dla administratora systemu operacyjnego *Linux*. System *Linux* podobnie jak inne systemy z rodziny *UNIX* zapewnia użytkownikowi łatwy dostęp do dowolnego obszaru pamięci masowej poprzez interfejs urządzeń blokowych. Operacje niskopoziomowego odczytu i zapisu w wybranym obszarze dysku twardego są równie proste jak odczyt bądź zapis w zwykłym pliku. Dzięki temu problem archiwizacji systemów rodziny *Windows* można sprowadzić do tworzenia obrazu urządzenia blokowego w systemie *Linux*.

Integralną częścią pracy jest implementacja systemu archiwizacji partycji w trybie surowym o nazwie „*Fola Partition Archivist*“ w skrócie „*FolaPA*”. Implementacja wchodzi w skład projektu *FOLA* (*the Friend Of a Lazy Administrator*).

Projekt *FOLA* oferuje bogaty zestaw narzędzi wspomagających administrowanie grupą stacji roboczych oraz serwerów zarówno w niewielkich sieciach biurowych jak i sieciach akademickich czy korporacyjnych. Narzędzia wchodzące w skład projektu *FOLA* wspierają administratora w zarządzaniu kontami użytkowników, monitorowaniu stanu serwerów i stacji roboczych ([21], [23]), zarządzaniu pakietami oprogramowania ([19]), nadzorowaniu hostów i lokalnej sieci komputerowej pod kątem bezpieczeństwa sieciowego ([20], [24]) oraz tworzeniu i zarządzaniu kopiami zapasowymi ([22], [25]). *Fola Partition Archivist* uzupełnia ten zestaw o możliwość tworzenia i odtwarzania obrazów urządzeń blokowych, czyli między innymi o zdalną archiwizację systemów z rodziny *Windows*. Wszystkie elementy projektu *FOLA* działają pod kontrolą systemu operacyjnego *Linux* i rozwijane są przy wykorzystaniu interpretowanego języka programowania *Perl*.

Praca rozpoczyna się od omówienia koncepcji archiwizacji urządzeń blokowych (rozdział 2). W pierwszej kolejności jest omawiany interfejs urządzeń blokowych oraz pojęcie ich archiwizacji. Następnie proponowana metoda archiwizacji jest porównywana ze standardowym sposobem tworzeniem kopii bezpieczeństwa. Kolejnym poruszonym zagadnieniem jest przegląd dodatkowych możliwości jakie oferuje archiwizacja urządzeń blokowych. W tym rozdziale przedstawione zostały także wymagania jakim powinna sprostać proponowana implementacja. W rozdziale 3 została nakreślona architektura projektu *FolaPA* oraz opisana struktura zbiorów archiwum, która w istotny sposób wpływa na funkcjonalność projek-

towanego systemu. Rozdział 4 przedstawia poszczególne elementy implementacji (moduły, skrypty, najważniejsze procedury i struktury danych) oraz omawia platformę programistyczną, na której opiera się system. Załączniki stanowią uzupełnienie rozdziału 4: A zawiera listę użytych modułów bogatej biblioteki języka Perl, B – listę wymaganych pakietów oprogramowania, a D – dokumentację kodu źródłowego utworzoną w formacie *POD* (*Plain Old Documentation*).