

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

Fedora Core 2. Biblia

Autor: Christopher Negus

Tłumaczenie: Piotr Pilch

ISBN: 83-7361-723-X

Tytuł oryginału: [Red Hat Fedora Linux 2 Bible](#)

Format: B5, stron: 1216



Poznaj wszystkie możliwości najnowszej wersji jednej z najpopularniejszych dystrybucji Linuksa

- Wykorzystaj zalety nowych technologii zaimplementowanych w Fedora Core 2 – obsługi multimediów, mechanizmów bezpieczeństwa i najnowszej wersji jądra
- Naucz się zasad pracy ze środowiskami graficznymi KDE i GNOME oraz w trybie konsoli
- Zainstaluj i skonfiguruj usługi sieciowe, serwer WWW i poczty elektronicznej oraz bazy danych
- Korzystaj w pracy i w domu z oprogramowania dołączonego do systemu

Fedora Core 2 to najnowsza wersja jednej z najpopularniejszych dystrybucji systemu Linux, znanej dotychczas pod nazwą Red Hat. Nazwa Red Hat Linux jest obecnie zarezerwowana dla dystrybucji rozprowadzanych na zasadach komercyjnych, a Fedora Core – dla dystrybucji dostępnych nieodpłatnie. Mimo polityki firmy jej najnowszy produkt pozostaje doskonałym systemem operacyjnym, wyposażonym we wszystkie nowe technologie, jakie pojawiły się w świecie Linuksa. Wykorzystano w nim najnowszą wersję jądra – 2.6, mechanizm zabezpieczeń Security Enhanced Linux i najnowsze wersje oprogramowania dołączanego do kolejnych wersji tej dystrybucji Linuksa.

„Fedora Core 2. Biblia” to przewodnik po najnowszej dystrybucji produktu firmy Red Hat Inc. Przeznaczony jest dla użytkowników, którzy rozpoczynają pracę z Linuksem i chcą poznać jego możliwości, wykonując rzeczywiste zadania. Dzięki takiemu przedstawieniu zawartych w książce wiadomości łatwiej jest nie tylko zapamiętać sposób realizacji określonych czynności, ale także zrozumieć powiązania i wzajemne zależności pomiędzy składnikami systemu.

- Geneza dystrybucji Fedora Core
- Instalacja systemu z płyt CD lub z wykorzystaniem innych mediów
- Praca ze środowiskami graficznymi GNOME i KDE
- Korzystanie z trybu tekstowego i poleceń konsoli
- Uruchamianie aplikacji
- Zastosowanie Linuksa do pracy i rozrywki
- Połączenie z internetem
- Administrowanie systemem plików i kontami użytkowników
- Tworzenie skryptów powłoki
- Zabezpieczanie systemu przed atakami hakerów
- Konfigurowanie usług sieciowych w systemie Fedora Core



Spis treści

O Autorze	21
Przedmowa	23
Część I Rozpoczynamy pracę z systemem Fedora Linux	31
Rozdział 1. Wprowadzenie do systemu Fedora Linux	33
Fedora Linux — wizytówka systemu	34
Czym jest Linux?	35
Korzenie systemu Linux sięgają Uniksa	36
Ogólne cechy systemu Linux	39
Podstawowe zalety systemu Linux	40
Czym jest Red Hat Linux i Fedora Linux?	41
Firma Red Hat Inc. rozpoczyna projekt Fedora Project	42
Firma Red Hat Inc. skupia się na rozwoju dystrybucji Red Hat Enterprise Linux	43
Fedora Linux czy Red Hat Enterprise Linux?	43
Dlaczego należy wybrać system Fedora Linux lub Red Hat Enterprise Linux?	44
Nowości w dystrybucji Fedora Core 2	46
Jądro systemu Linux w wersji 2.6	48
System dźwiękowy ALSA	48
Technologia Security Enhanced Linux	48
Systemowe narzędzia konfiguracyjne	48
Serwer X i inne środowiska graficzne	49
Dodatkowe pakiety oprogramowania	50
Idea otwartej licencji oprogramowania	50
Podsumowanie	51
Rozdział 2. Instalacja systemu Fedora Linux	53
Szybka instalacja systemu Fedora Linux	53
Instalacja systemu — krok po kroku	57
Instalacja systemu Fedora Core 2	57
Wybór metody instalacji	57
Wybór odpowiedniego sprzętu	59
Rozpoczynamy instalację	61
Fedora Setup Agent	73
Instalacja dodatkowych pakietów oprogramowania Fedora Linux	75
Instalacja systemu Fedora Linux — procedury specjalne	76
Inne metody rozpoczynania instalacji	76
Instalacja systemu Fedora Linux z wykorzystaniem innych mediów	78
Wykonywanie instalacji typu kickstart	82

Instalacja systemu Fedora Linux — zagadnienia specjalne.....	88
Partycjonowanie dysków twardych.....	88
Odzyskiwanie wolnego miejsca z istniejących partycji.....	98
Zastosowanie programów ładujących GRUB lub LILO.....	104
Diagnozowanie i usuwanie problemów z instalacją	114
Podsumowanie.....	116
Rozdział 3. Uruchamianie środowiska graficznego	117
Logowanie w systemie Fedora Linux.....	118
Uruchamianie pulpitu	120
Środowisko graficzne GNOME.....	130
Zastosowanie menedżera okien Metacity.....	132
Zastosowanie panelu GNOME.....	134
Zastosowanie menedżera plików Nautilus	140
Modyfikacja właściwości środowiska GNOME.....	143
Zakończenie pracy ze środowiskiem GNOME	146
Środowisko graficzne KDE	147
Uruchamianie środowiska KDE	147
Krótki opis pulpitu środowiska KDE	148
Zastosowanie menedżera plików Konqueror.....	152
Konfiguracja opcji menedżera plików Konqueror.....	158
Zarządzanie oknami	161
Dodawanie aktywatorów aplikacji oraz typów MIME	167
Przygotowanie pulpitu do pracy	168
Po załadowaniu systemu nie uruchamia się środowisko graficzne.....	169
Konfiguracja karty graficznej i monitora	170
Konfiguracja kart graficznych pod kątem gier	172
Gdzie szukać dodatkowych informacji?	173
Podsumowanie.....	173
Rozdział 4. Polecenia systemu Linux	175
Powłoka systemu Linux.....	175
Kontrola sesji logowania.....	176
Sprawdzanie katalogów i uprawnień.....	177
Kontrola procesów systemu Linux	178
Kończenie pracy z powłoką.....	180
Powłoka systemu Fedora Linux.....	180
Korzystanie z powłoki systemowej	181
Odszukiwanie poleceń powłoki.....	182
Powtórne wykonywanie poleceń.....	184
Łączenie i rozwijanie poleceń	191
Zastosowanie zmiennych środowiskowych powłoki.....	194
Zarządzanie procesami pierwszo- i drugoplanowymi	197
Konfiguracja powłoki systemu.....	200
Praca z systemem plików Fedora Linux	204
Tworzenie plików i katalogów	205
Przenoszenie, kopiowanie i usuwanie plików	212
Edytor vi — przyjaciel czy wróg?	213
Pracujemy z edytorem vi	213
Nawigacja w dokumencie.....	217
Wyszukiwanie tekstu.....	217
Zastosowanie liczb w poleceniach	218
Podsumowanie.....	219

Część II Fedora Linux w praktyce 221**Rozdział 5. Uruchamianie aplikacji..... 223**

Fedora Linux jako platforma do uruchamiania aplikacji	224
Odszukiwanie odpowiedników aplikacji systemu Windows w systemie Linux	226
Pobieranie aplikacji dla systemu Fedora Linux	226
A może poszukać na pulpicie?	228
Poszukiwanie aplikacji w sieci internetowej	229
Pobieranie i instalowanie aplikacji za pomocą programu yum.....	230
Pobieranie oprogramowania dla systemu Linux.....	233
Konwencje nazywania pakietów oraz ich formaty	235
Instalowanie aplikacji w systemie Fedora Linux	238
Instalacja pakietów RPM i zarządzanie nimi.....	238
Tworzenie oraz instalacja aplikacji z kodu źródłowego	250
Uruchamianie aplikacji dla X Window System	253
Uruchamianie aplikacji z menu systemowego.....	253
Uruchamianie aplikacji z okna Uruchom program.....	253
Uruchamianie aplikacji z okna terminala	255
Uruchamianie zdalnych aplikacji X	256
Zastosowanie emulatorów do uruchamiania aplikacji z innych systemów operacyjnych.....	261
Uruchamianie aplikacji przeznaczonych dla systemu DOS.....	262
Zastosowanie pakietu WINE do uruchamiania aplikacji przeznaczonych dla systemu Windows.....	266
Zastosowanie pakietu ARDI Executor do uruchamiania aplikacji przeznaczonych dla komputerów Macintosh	271
Podsumowanie.....	272

Rozdział 6. Tworzenie dokumentów w systemie Fedora Linux..... 273

Pakiet biurowy OpenOffice.org.....	274
Inne edytory tekstu	276
Pakiet StarOffice	277
Edytor tekstu AbiWord	278
Pakiet KOffice.....	279
Zastosowanie tradycyjnych narzędzi systemu Linux do publikacji dokumentów	280
Tworzenie dokumentów przy użyciu systemów formatowania Groff oraz LaTeX	281
Przetwarzanie tekstu przy użyciu systemu formatowania Groff.....	282
Przetwarzanie tekstu przy użyciu procesora TeX/LaTeX.....	293
Metody konwersji dokumentów	297
Tworzenie dokumentów typu DocBook.....	299
Kilka słów o SGML oraz XML.....	299
Drukowanie dokumentów w systemie Fedora Linux.....	304
Korzystanie z domyślnej drukarki systemowej	304
Drukowanie plików z poziomu powłoki systemu.....	305
Kontrola stanu kolejek wydruku.....	305
Usuwanie zadań drukowania z kolejki	306
Sprawdzanie stanu drukarki	307
Wyświetlanie dokumentów przy użyciu pakietów Ghostscript oraz Acrobat.....	307
Zastosowanie poleceń ghostscript oraz gv	307
Zastosowanie przeglądarki Adobe Acrobat Reader.....	308

Programy graficzne w systemie Fedora Linux	310
Przetwarzanie plików graficznych przy użyciu programu GIMP	310
Wykonywanie zrzutów ekranu	311
Przetwarzanie plików graficznych przy użyciu programu KPaint	312
Skanowanie dokumentów przy użyciu pakietu SANE	313
Podsumowanie	314
Rozdział 7. Gry i zabawy w systemie Fedora Linux	315
Gry w systemie Linux? Ależ oczywiście!	316
Gdzie można znaleźć informacje na temat gier w systemie Linux?	316
Gry przeznaczone dla systemu Linux	317
Dobór karty graficznej obsługującej gry	318
Gry w środowisku X	319
Gry dla środowiska GNOME	319
Gry dla środowiska KDE	319
Gry komercyjne w systemie Linux	330
Gry firmy id Software	330
Pakiet TransGaming WineX	332
Wersje demonstracyjne gier firmy Loki Software	335
Civilization: Call to Power	337
Myth II: Soulblighter	337
Heretic II	338
Neverwinter Nights	339
Podsumowanie	339
Rozdział 8. Fedora Linux a multimedia	341
Odtwarzanie audio w systemie Linux	342
Konfiguracja karty dźwiękowej	344
Wybór odtwarzacza płyt audio-CD	345
Zastosowanie odtwarzacza MIDI	356
Konwersja oraz kompresowanie plików audio	356
Karty TV i kamery internetowe w systemie Linux	360
Oglądanie TV przy użyciu programu Tvtime	361
Prowadzenie videokonferencji przy użyciu pakietu GnomeMeeting	363
Odtwarzanie wideo w systemie Linux	365
Odtwarzacz plików wideo xine	366
Zastosowanie odtwarzacza RealPlayer	370
Zastosowanie pakietów gtkam oraz gphoto2 do pracy z cyfrowymi aparatami fotograficznymi	371
Pobieranie cyfrowych obrazów przy użyciu pakietu gtkam	373
Użycie aparatu cyfrowego w roli urządzenia przechowującego dane	374
Nagrywanie dysków audio-CD	376
Tworzenie płyt audio-CD przy użyciu polecenia cdrecord	376
Zapisywanie zawartości płyt audio-CD na dysku	378
Tworzenie etykiet dysków CD przy użyciu polecenia cdlabelgen	380
Podsumowanie	380
Rozdział 9. Narzędzia umożliwiające korzystanie z sieci internetowej i sieci WWW	383
Korzystanie z narzędzi obsługi sieci WWW	384
Przeglądanie zasobów sieci WWW	384
Adresy URL	385
Strony WWW	387

Przeglądanie witryn WWW za pomocą aplikacji Mozilla.....	388
Korzystanie z przeglądarek WWW pracujących w trybie tekstowym.....	399
Komunikacja z wykorzystaniem poczty elektronicznej.....	400
Podstawy obsługi poczty elektronicznej.....	401
Klient pocztowy Evolution.....	402
Klient pocztowy Mozilla Mail.....	406
Programy pocztowe pracujące w trybie tekstowym	408
Narzędzia odczytywania i zarządzania wiadomościami poczty elektronicznej	409
Korzystanie z grup dyskusyjnych	410
Obsługa grup dyskusyjnych z wykorzystaniem pakietu Mozilla.....	411
Program odczytywania wiadomości grup dyskusyjnych Pan	414
Korzystanie z Instant Messaging za pomocą programu Gaim	414
Korzystanie z poleceń zdalnego logowania, kopiowania i uruchamiania programów	415
Korzystanie z polecenia telnet do zdalnego logowania	415
Kopiowanie plików za pomocą FTP	417
Pobieranie plików za pomocą wget	423
Korzystanie z ssh do zdalnego logowania i uruchamiania poleceń	425
Zastosowanie polecenia scp do zdalnego kopiowania plików	425
Korzystanie z poleceń „r”: rlogin, rcp i rsh	426
Podsumowanie.....	426

Część III Zarządzanie systemem Fedora Linux 427

Rozdział 10. Podstawy zarządzania systemem..... 429

Użytkownik root.....	430
Jak zostać superużytkownikiem (polecenie su)	430
Polecenia i graficzne narzędzia administracyjne, pliki konfiguracyjne oraz dzienniki zdarzeń	432
Narzędzia administracyjne dysponujące graficznym interfejsem użytkownika.....	432
Polecenia administracyjne	436
Administracyjne pliki konfiguracyjne.....	437
Pliki dzienników administracyjnych.....	442
Korzystanie z innych kont administracyjnych.....	442
Administrowanie systemem Fedora Linux	445
Konfiguracja urządzeń.....	446
Modyfikowanie konfiguracji sprzętowej z wykorzystaniem kudzu	446
Konfigurowanie modułów.....	447
Zarządzanie systemem plików i przestrzenią dyskową.....	450
Podłączanie systemów plików.....	453
Korzystanie z polecenia mkfs do tworzenia systemu plików	459
Dodawanie dysku twardego	460
Zastosowanie macierzy RAID.....	463
Sprawdzanie przestrzeni systemowej	465
Monitorowanie wydajności systemu.....	467
Analiza wykorzystania komputera za pomocą narzędzia Monitor systemu	467
Monitorowanie wykorzystania procesora za pomocą top.....	468
Monitorowanie wykorzystania zasilania laptopów.....	469
Wybór alternatywnego oprogramowania.....	472
Wybór alternatywnych rozwiązań poczty elektronicznej i drukowania	472
Korzystanie z alternatywnych usług przesyłania poczty	474

Uaktualnianie oprogramowania Linux	474
Pobieranie uaktualnień ze zbiorów oprogramowania dla systemu Fedora Linux	474
Narzędzie powiadamiające Red Hat Network	475
Rejestrowanie na stronie Red Hat Network	475
Pobieranie uaktualnień	477
Korzystanie z Red Hat Network	479
Podsumowanie	480
Rozdział 11. Konfigurowanie kont użytkowników i zarządzanie nimi.....	481
Tworzenie kont użytkowników	482
Dodawanie użytkowników za pomocą useradd	482
Dodawanie kont użytkowników za pomocą narzędzia Menedżer użytkowników	486
Konfigurowanie ustawień domyślnych	488
Określanie początkowych skryptów logowania	490
Początkowy plik .bashrc	491
Początkowy plik .tcshrc	491
Konfigurowanie globalnych opcji powłoki systemowej	492
Konfigurowanie profili systemowych	493
Tworzenie przenośnych pulpitu	494
Obsługa użytkowników	495
Tworzenie skrzynki pocztowej wsparcia technicznego	495
Zmiana hasła użytkownika	496
Modyfikowanie ustawień kont użytkowników	497
Modyfikowanie ustawień konta użytkownika za pomocą polecenia usermod	497
Modyfikowanie ustawień kont użytkowników za pomocą okna Menedżer użytkowników	499
Usuwanie kont użytkowników	500
Usuwanie kont użytkowników z wykorzystaniem polecenia userdel	500
Usuwanie kont użytkowników za pomocą okna Menedżer użytkowników	501
Sprawdzanie wykorzystania przestrzeni dyskowej	502
Limit zasobów dyskowych jako metoda kontroli przestrzeni dyskowej	502
Korzystanie z polecenia du do kontroli wykorzystania dysku	506
Automatyczne usuwanie plików tymczasowych	507
Wysyłanie wiadomości e-mail do użytkowników	507
Podsumowanie	509
Rozdział 12. Automatyzacja zadań systemowych	511
Skrypty powłoki systemowej	512
Uruchamianie i debugowanie skryptów powłoki	512
Zmienne powłoki systemowej	513
Wykonywanie operacji arytmetycznych w skryptach powłoki	515
Wykorzystywanie struktur programistycznych w skryptach powłoki	516
Kilka przydatnych zewnętrznych narzędzi	521
Tworzenie prostych skryptów powłoki	523
Inicjalizacja systemu	524
Uruchamianie procesu init	525
Plik inittab	525
Uruchamianie i zamykanie systemu	529
Uruchamianie skryptów poziomu uruchamiania	529
Zrozumienie skryptów poziomu uruchamiania	530
Zrozumienie działania skryptów uruchomieniowych	532
Zmiana zachowania skryptu poziomu uruchamiania	534

Reorganizacja i usuwanie skryptów poziomu uruchamiania.....	535
Dodawanie skryptów poziomu uruchamiania.....	536
Zarządzanie usługami xinetd.....	537
Manipulowanie poziomami uruchamiania.....	538
Planowanie zadań systemowych.....	539
Korzystanie z at.allow i at.deny.....	540
Określanie czasu uruchamiania zadań.....	540
Przekazywanie zaplanowanych zadań.....	540
Przeglądanie zaplanowanych zadań.....	541
Usuwanie zaplanowanych zadań.....	542
Korzystanie z polecenia batch.....	542
Korzystanie z narzędzia cron.....	543
Podsumowanie.....	546
Rozdział 13. Tworzenie kopii bezpieczeństwa i przywracanie plików	547
Wykonywanie podstawowej archiwizacji przy użyciu narzędzia rsync.....	548
Lokalna archiwizacja plików.....	549
Zdalna archiwizacja plików.....	550
Wybór strategii tworzenia kopii bezpieczeństwa.....	551
Pełna kopia bezpieczeństwa.....	551
Przyrostowa kopia bezpieczeństwa.....	552
Tworzenie kopii lustrzanej.....	552
Sieciowa kopia bezpieczeństwa.....	552
Wybór nośnika kopii bezpieczeństwa.....	553
Taśma magnetyczna.....	553
Nagrywalne dyski CD-ROM.....	556
Nagrywalne dyski DVD.....	559
Tworzenie kopii bezpieczeństwa na dysku twardym.....	560
Instalowanie mirrordir do klonowania katalogów.....	561
Klonowanie katalogu za pomocą mirrordir.....	561
Automatyczne tworzenie kopii lustrzanych.....	562
Tworzenie kopii bezpieczeństwa za pomocą polecenia dump.....	562
Tworzenie kopii bezpieczeństwa.....	563
Poziomy operacji dump.....	565
Automatyzacja tworzenia kopii bezpieczeństwa z wykorzystaniem narzędzia cron.....	566
Przywracanie plików z kopii bezpieczeństwa.....	568
Przywracanie całego systemu plików.....	568
Odzyskiwanie poszczególnych plików.....	570
Konfigurowanie narzędzia Amanda do wykonywania sieciowych kopii bezpieczeństwa.....	572
Wykonywanie kopii bezpieczeństwa narzędziem Amanda.....	577
Korzystanie z narzędzia archiwizacji pax.....	578
Podsumowanie.....	581
Rozdział 14. Bezpieczeństwo systemu	583
Haker a włamywacz.....	583
Metody ataku.....	584
Ochrona przed atakami „odmowa usługi”.....	585
Mailbombing.....	585
Blokowanie spamu.....	587
Smurfing — wzmożony atak.....	588

Metody obrony przed atakami DDoS	589
Obrona przed atakami intruzów.....	593
Analizowanie dostępu do usług sieciowych	593
Blokowanie usług sieciowych	594
Korzystanie z pośrednictwa TCP	596
Ochrona sieci z wykorzystaniem firewalli.....	598
Konfigurowanie filtrującego firewalla iptables	599
Wykrywanie włamań na podstawie plików dzienników	610
Rola syslogd	611
Przekierowywanie wiadomości dzienników za pomocą syslogd.....	613
Zrozumienie wiadomości logfile.....	614
Monitorowanie plików dzienników z wykorzystaniem LogSentry.....	615
Pobieranie i instalowanie LogSentry	616
Instalowanie LogSentry.....	616
Uruchamianie LogSentry.....	616
Korzystanie z LogSentry	617
Dopasowywanie LogSentry do potrzeb danego systemu	618
Ochrona hasłem.....	624
Wybór dobrego hasła	625
Korzystanie z pliku haseł ukrytych	626
Wykorzystanie metod szyfrowania.....	628
Kryptografia symetryczna	628
Kryptografia przy użyciu klucza publicznego	629
Protokół SSL (Secure Socket Layer).....	629
Eksportowanie technologii kodowania.....	638
Korzystanie z pakietu Secure Shell	638
Uruchamianie usługi SSH	638
Korzystanie z poleceń ssh, sftp i scp	639
Korzystanie z poleceń ssh, sftp i scp bez podawania hasła.....	640
Ochrona komputera z wykorzystaniem PortSentry.....	641
Pobieranie i instalowanie PortSentry.....	642
Korzystanie z PortSentry.....	642
Konfigurowanie PortSentry.....	643
Testowanie PortSentry	647
Śledzenie włamań PortSentry	649
Przywracanie dostępu.....	649
Podsumowanie.....	650

Część IV Konfiguracja usług sieciowych serwera Fedora Linux .. 651

Rozdział 15. Konfiguracja sieci lokalnej LAN 653

Sieci lokalne	653
Konfiguracja urządzeń sieci lokalnej	654
Konfigurowanie protokołu TCP/IP sieci lokalnej	659
Tworzenie i konfiguracja bezprzewodowej sieci lokalnej	665
Sieci bezprzewodowe	666
Dobór komponentów sieci bezprzewodowych.....	667
Instalacja programów systemu Linux współpracujących z sieciami bezprzewodowymi.....	672
Konfiguracja bezprzewodowej sieci lokalnej	673
Pomiar zasięgu sieci	682
Modyfikowanie ustawień związanych z obsługą sieci bezprzewodowych.....	683

Adresy IP	683
Klasy adresów IP	685
Maski sieciowe	686
Routing bezklasowy CIDR (Classless Inter-Domain Routing)	686
Uzyskiwanie adresów IP	687
Rozwiązywanie problemów z siecią lokalną	688
Czy system Linux w trakcie uruchamiania odnalazł sterownik karty sieciowej?	689
Czy można się połączyć z innym komputerem znajdującym się w sieci lokalnej?	689
Czy karta sieciowa Ethernet jest aktywna?	690
Rozwiązywanie problemów z bezprzewodowymi sieciami lokalnymi	691
Monitorowanie natężenia ruchu w sieci lokalnej LAN przy użyciu programu Ethereal	695
Podsumowanie	701
Rozdział 16. Podłączenie do internetu	703
Budowa sieci internetowej	704
Domeny internetowe	705
Nazwy i adresy IP komputerów	707
Routing	709
Usługa proxy	710
Łączenie się z internetem za pomocą modemu	710
Uzyskiwanie wymaganych parametrów	710
Konfiguracja połączenia PPP	712
Uruchamianie połączenia PPP	715
Uruchamianie połączenia PPP na żądanie	715
Testowanie połączenia PPP	716
Połączenie sieci lokalnej z internetem	722
Konfiguracja komputera z systemem Fedora Linux jako routera	723
Konfiguracja funkcji routingu systemu Fedora Linux	724
Konfiguracja klientów	727
Konfiguracja klientów wyposażonych w system Windows	729
Konfiguracja wirtualnej sieci prywatnej VPN (Virtual Private Network)	729
Protokół IPSEC	731
Protokół szyfrowania CIPE	733
Konfiguracja połączenia VPN opartego na protokole CIPE	734
Testowanie połączenia VPN CIPE	738
Konfiguracja komputera z systemem Linux jako serwera proxy	739
Uruchamianie modułu squid	740
Prosty plik konfiguracyjny squid.conf	742
Modyfikacja pliku konfiguracyjnego serwera Squid	743
Diagnozowanie serwera Squid	748
Konfiguracja klientów proxy	749
Konfiguracja przeglądarki Mozilla pod kątem wykorzystania usługi proxy	750
Konfiguracja przeglądarki Internet Explorer pod kątem wykorzystania usługi proxy	751
Konfiguracja innych przeglądarek pod kątem wykorzystania usługi proxy	752
Podsumowanie	753
Rozdział 17. Konfiguracja serwera druku	755
Usługa wydruku CUPS czy LPRng?	756
Usługa CUPS (Common UNIX Printing Service)	756
Usługa LPRng (Line Printer New Generation)	757
Przełączanie usługi wydruku	757

Konfiguracja drukarek	758
Konfiguracja lokalnych drukarek	758
Konfigurowanie drukarek sieciowych	763
Usługa wydruku CUPS	767
Administrowanie usługą CUPS przy użyciu przeglądarki internetowej	767
Konfigurowanie serwera CUPS (plik cupsd.conf)	769
Konfigurowanie opcji drukarki CUPS	771
Zarządzanie drukarkami	772
Polecenia obsługujące drukowanie	773
Drukowanie przy użyciu programu lpr	773
Wyświetlanie statusu przy użyciu programu lpc	774
Usuwanie zadań wydruku przy użyciu polecenia lprm	774
Konfiguracja serwerów wydruku	775
Konfigurowanie współdzielonej drukarki CUPS	775
Konfiguracja drukarki udostępnionej przez system NetWare	777
Konfiguracja drukarki udostępnionej przez serwer Samba	778
Podsumowanie	780
Rozdział 18. Konfiguracja serwera plików	781
Dlaczego warto skonfigurować serwer plików?	782
Konfiguracja serwera plików NFS	782
Udostępnianie systemu plików przez serwer NFS	784
Systemy plików NFS	792
Odmontowanie systemu plików NFS	797
Inne interesujące możliwości usługi NFS	798
Konfiguracja serwera plików Samba	799
Instalacja pakietu Samba	800
Konfigurowanie prostego serwera plików Samba	801
Konfigurowanie serwera Samba przy użyciu narzędzia SWAT	804
Pliki i polecenia serwera Samba	814
Konfigurowanie klientów serwera Samba	818
Rozwiązywanie problemów z serwerem Samba	821
Konfiguracja serwera plików NetWare	824
Konfiguracja serwera plików NetWare	824
Polecenia używane przez klienta NetWare	827
Podsumowanie	828
Rozdział 19. Konfiguracja serwera poczty	829
Omówienie protokołu SMTP i serwera sendmail	830
Instalacja i uruchomienie serwera sendmail	831
Uruchamianie serwera sendmail	831
Inne programy	832
Rejestracja zdarzeń związanych z działaniem serwera sendmail	833
Konfiguracja serwera Sendmail	834
Uzyskanie nazwy domeny	835
Określanie podstawowych ustawień serwera sendmail (plik sendmail.mc)	836
Definiowanie dostępu do wiadomości wychodzących	840
Konfigurowanie wirtualnych serwerów	842
Konfigurowanie kont wirtualnych użytkowników	842
Dodanie kont użytkowników	843
Uruchomienie serwera sendmail i wygenerowanie plików bazy danych	844
Przekierowywanie wiadomości	845

Serwer Postfix	847
Rozpoznawanie spamu przy użyciu programu SpamAssassin.....	848
Zastosowanie programu SpamAssassin na własnym serwerze pocztowym	849
Konfigurowanie klienta pocztowego pod kątem filtracji spamu	851
Pobieranie wiadomości z serwera poczty (protokoły IMAP i POP3)	852
Dostęp do skrzynek pocztowych serwera poczty działającego w systemie Linux.....	853
Konfiguracja usług POP3 i IMAP przy użyciu usługi dovecot	854
Pobieranie wiadomości z poziomu przeglądarki internetowej przy użyciu narzędzia Squirrelmail..	855
Administrowanie listą dystrybucyjną przy użyciu narzędzia mailman	856
Podsumowanie.....	860
Rozdział 20. Konfiguracja serwera FTP	861
Serwery FTP.....	862
Funkcje serwera FTP.....	862
Typy użytkowników serwera FTP.....	863
Zastosowanie serwera vsFTPD	864
„Szybkie” uruchomienie serwera vsFTPD	864
Konfiguracja serwera vsFTPD	865
Dodatkowe informacje na temat serwerów FTP.....	871
Podsumowanie.....	871
Rozdział 21. Konfiguracja serwera WWW	873
Podstawowe informacje na temat serwerów WWW	874
Serwer Apache	874
Serwer TUX	875
Inne serwery WWW przeznaczone dla systemu Fedora Linux	876
„Szybkie” uruchomienie serwera Apache.....	877
Konfiguracja serwera Apache.....	879
Konfiguracja serwera WWW (plik httpd.conf)	880
Konfigurowanie modułów i powiązanych z nimi usług (pliki /etc/httpd/conf.d/*.conf)	916
Zatrzymywanie i uruchamianie serwera Apache	918
Monitorowanie pracy serwera	918
Wyświetlanie informacji na temat serwera.....	920
Wyświetlanie statusu serwera.....	920
Dodatkowa kontrola dostępu do strony Server Info i Server Status	922
Rejestrowanie błędów	922
Rejestrowanie żądań.....	923
Analiza danych przetwarzanych przez serwer WWW	923
Podsumowanie.....	925
Rozdział 22. Konfiguracja serwera adresowego LDAP	927
Protokół LDAP	928
Definiowanie informacji w schematach	929
Tworzenie struktury katalogów LDAP.....	929
Zastosowanie serwera OpenLDAP	930
Instalowanie pakietów serwera OpenLDAP.....	931
Konfigurowanie serwera OpenLDAP (plik slapd.conf)	931
Uruchamianie usługi serwera OpenLDAP	933
Tworzenie książki adresowej LDAP	933
Inne zadania związane z konfigurowaniem katalogu LDAP	938
Korzystanie z książki adresowej przy użyciu programu Mozilla Mail	939
Podsumowanie.....	941

Rozdział 23. Konfiguracja serwera DHCP i NIS.....	943
Zastosowanie protokołu DHCP (Dynamic Host Configuration Protocol)	944
Konfigurowanie serwera DHCP	944
Konfiguracja firewalla pod kątem serwera DHCP	944
Konfiguracja pliku dhcpd.conf.....	945
Uruchamianie serwera DHCP	951
Konfiguracja klienta DHCP.....	952
Usługa NIS (Network Information Service)	953
Konfiguracja komputera z systemem Fedora Linux jako klienta NIS	955
Określenie nazwy domeny NIS	956
Konfiguracja pliku /etc/yp.conf.....	957
Konfiguracja modułów klienta NIS.....	957
Kontrola poprawności działania usługi NIS	958
Zastosowanie map usługi NIS	958
Konfiguracja komputera z systemem Fedora Linux jako serwera nadrzędnego NIS.....	959
Tworzenie map usługi NIS.....	960
Konfiguracja komputera z systemem Fedora Linux jako serwera podrzędnego NIS	964
Podsumowanie.....	964
Rozdział 24. Konfiguracja serwera bazodanowego MySQL	965
Pakiety serwera MySQL.....	966
Konfiguracja serwera MySQL.....	966
Zastosowanie konta użytkownika i grupy o nazwie mysql.....	967
Tworzenie kont administracyjnych	967
Definiowanie opcji serwera MySQL	968
Zastosowanie przykładowych plików my.cnf	973
Uruchomienie serwera MySQL	975
Kontrola poprawności działania serwera MySQL	975
Praca z bazami danych serwera MySQL	976
Uruchomienie polecenia mysql	976
Tworzenie bazy danych przy użyciu polecenia mysql.....	977
Wprowadzanie danych do tabeli bazy danych serwera MySQL.....	979
Tabele bazy danych serwera MySQL	982
Wyświetlanie danych zawartych w bazie danych MySQL	984
Wyświetlanie wszystkich lub wybranych rekordów.....	987
Wyświetlanie wybranych kolumn	988
Sortowanie danych	989
Modyfikowanie tabel oraz ich rekordów	990
Modyfikacja struktury tabel bazy danych MySQL.....	990
Aktualizowanie i usuwanie rekordów bazy danych serwera MySQL	991
Dodawanie i usuwanie użytkowników	991
Dodawanie użytkowników i nadawanie im uprawnień	992
Odbieranie uprawnień	993
Kontrola i naprawianie baz danych.....	993
Podsumowanie.....	995
Rozdział 25. Publiczne udostępnianie usług sieciowych przy użyciu serwera DNS.....	997
Określenie przeznaczenia serwera	998
Korzystanie z usług zewnętrznych firm internetowych.....	998

Podłączenie serwera publicznego	999
Wybór dostawcy usług internetowych.....	1000
Rejestracja nazwy domeny	1002
Konfiguracja serwera publicznego	1005
Konfiguracja sieci	1005
Konfiguracja serwerów	1006
Zarządzanie zabezpieczeniami	1007
Instalacja i konfiguracja serwera DNS	1009
Serwer DNS	1010
Serwer DNS — przykład.....	1012
„Szybkie” uruchomienie serwera DNS	1014
Kontrola poprawności pracy serwera DNS	1022
Dodatkowe źródła informacji na temat serwera BIND	1023
Podsumowanie.....	1024
Rozdział 26. Współpraca systemów Macintosh z serwerem Linux.....	1025
System Mac OS X od wnętrza	1026
Usługi sieciowe systemu Mac OS X.....	1027
Zastosowanie protokołu AppleTalk (pakiet netatalk) w systemie Mac OS X	1028
Zastosowanie protokołu AppleTalk w systemach Mac OS 8 i OS 9	1029
Serwer Samba (komputery z systemami Windows i Linux).....	1030
Współużytkowanie aplikacji środowiska graficznego X	1031
Konfigurowanie serwera AppleTalk w systemie Linux.....	1032
Co należy wiedzieć przed uruchomieniem serwera netatalk	1032
Konfigurowanie serwera netatalk	1033
Zabezpieczanie udziałów serwera netatalk.....	1038
Rozwiązywanie problemów z serwerem netatalk.....	1045
Dostęp do serwerów NFS z komputerów z systemem Mac.....	1046
Łączenie się z serwerem NFS przy użyciu okna Connect to Server	1047
Łączenie się z serwerem NFS z poziomu wiersza poleceń.....	1047
Podsumowanie.....	1049
Część V Nowe technologie	1051
Rozdział 27. Zapoznanie się z jądrem systemu Linux w wersji 2.6.....	1053
Podstawowe informacje na temat jądra	1054
Sprawdzenie aktualnie używanego jądra.....	1054
Zastosowanie katalogu /sys	1055
Innowacje związane z wydajnością stacji roboczych	1056
Innowacje związane z obsługą laptopów	1058
Innowacje związane z wydajnością serwerów	1058
Innowacje związane z obsługą sprzętu	1059
Obsługa urządzeń USB	1060
Korzystanie ze starszych urządzeń	1060
Zastosowanie jądra w wersji 2.6 poza systemem Fedora Core 2.....	1060
Uruchamianie jądra w wersji 2.6 na urządzeniach wbudowanych	1061
Zastosowanie jądra systemu Linux w wersji 2.6 w innych architekturach	1061
Podsumowanie.....	1062

Rozdział 28. Zastosowanie technologii Security Enhanced Linux 1063

Technologia Security Enhanced Linux	1064
Użytkownicy	1065
Zasady	1065
Zastosowanie technologii SELinux w systemie Fedora Core 2	1065
Uaktywnienie technologii SELinux	1066
Dodatkowe informacje na temat technologii SELinux	1069
Podsumowanie	1070

Dodatki 1071**Dodatek A Co znajduje się na płytach CD-ROM? 1073**

Dyski CD-ROM z kodem źródłowym pakietów systemu Fedora Linux	1073
Ratunkowy dysk CD-ROM systemu Fedora Core 2	1074

Dodatek B Pakiety RPM systemu Fedora Core 2 1077

Różnice między pakietami systemów Fedora Linux i Red Hat Enterprise Linux	1077
Wycofane pakiety RPM	1077
Pakiety RPM systemu Fedora Linux	1079

Dodatek C Uruchamianie usług sieciowych..... 1133

Lista kontrolna związana z uruchamianiem usług sieciowych	1133
Moduły usług sieciowych	1135
Superserwer xinetd	1135
Skrypty startowe init.d	1136
Rozwiązania alternatywne	1137
Przegląd usług sieciowych	1138
Serwer WWW	1138
Serwery plików	1139
Serwery logowania	1140
Serwery poczty	1141
Serwer grup dyskusyjnych	1142
Serwery wydruku	1142
Serwery zarządzające siecią	1143
Serwer NTP	1143
Serwer Portmap	1143
SWAT	1143
Serwer Arpwatch	1144
Serwer SNMP	1144
Serwery informacyjne	1144
Serwery bazodanowe	1146
Usługi użytkowników	1146
Usługi zabezpieczeń	1148

Dodatek D Publiczna licencja GNU..... 1155

Preambuła	1155
Warunki licencji GNU: kopiowanie, dystrybucja i modyfikowanie plików	1156

Skorowidz 1161

Rozdział 5.

Uruchamianie aplikacji

W tym rozdziale:

- ◆ Fedora Linux jako platforma do uruchamiania aplikacji
- ◆ Pobieranie aplikacji dla systemu Fedora Linux
- ◆ Instalowanie aplikacji w systemie Fedora Linux
- ◆ Uruchamianie aplikacji dla X Window System
- ◆ Zastosowanie emulatorów do uruchamiania aplikacji z innych systemów operacyjnych
- ◆ Uruchamianie aplikacji przeznaczonych dla systemu DOS
- ◆ Zastosowanie pakietu *WINE* do uruchamiania aplikacji przeznaczonych dla systemu Windows

Aby wykonać określone zadania, korzystasz z różnego rodzaju aplikacji. Aplikacje pozwalają m.in. na tworzenie dokumentów, przetwarzanie danych czy komunikację z innymi użytkownikami. Jako środowisko do uruchamiania aplikacji Fedora Linux każdego dnia coraz bardziej umacnia swoją pozycję. Jeszcze nie tak dawno mieliśmy dostęp do zaledwie garści aplikacji, o których można było z czystym sumieniem powiedzieć, że są przyjazne dla użytkownika — dziś użytkownicy mają do dyspozycji dziesiątki, jeżeli nie setki rozbudowanych, profesjonalnych i łatwych w użyciu aplikacji, a z dnia na dzień ich liczba gwałtownie rośnie.

W tym rozdziale omówimy zagadnienia związane z zaopatrywaniem się w aplikacje dla Fedora Linux, ich instalacją oraz uruchamianiem. Ponieważ Fedora Linux jest mimo wszystko nadal, w głębi serca, systemem operacyjnym dla programistów oraz administratorów sieci, to w tym rozdziale dowiesz się, w jaki sposób można pobrać aplikacje dla systemu Linux z internetu, jak je rozpakować, a następnie jak zainstalować. W niniejszym rozdziale skoncentrujemy się głównie na aplikacjach pracujących w środowisku graficznym (uruchamianych z wykorzystaniem X, włączając w to aplikacje pracujące zdalnie przez sieć).

Poza programami, które zostały specjalnie opracowane (lub dostosowane) do pracy w systemie Linux, istnieje możliwość uruchamiania aplikacji, które pierwotnie przeznaczone były dla innych systemów operacyjnych. W niniejszym rozdziale omówimy emulatory i oprogramowanie poprawiające zgodność umożliwiające uruchamianie aplikacji utworzonych dla systemów operacyjnych Windows, DOS oraz Macintosh.

Fedora Linux jako platforma do uruchamiania aplikacji

Mimo że system operacyjny sam w sobie jest całkiem sympatyczny i niezbędny, to jednak zdecydowana większość użytkowników wykorzystuje go głównie do uruchamiania aplikacji. Jeżeli chcesz postawić serwer, to Fedora Linux będzie na pewno jednym z pierwszych systemów operacyjnych, które weźmiesz pod uwagę — jednak w sektorze komputerów domowych system Fedora Linux jeszcze nieprędko będzie mógł poważnie zagrozić dominacji wszechobecnych systemów operacyjnych firmy Microsoft. Dzieje się tak z kilku powodów.

- ♦ Mimo że dla systemu Linux istnieje cały szereg edytorów tekstu, arkuszy kalkulacyjnych, programów graficznych oraz innych aplikacji praktycznie z dowolnej kategorii, to jednak na rynku komputerów PC funkcjonuje bardzo wiele szeroko rozpowszechnionych i lubianych aplikacji, które jeszcze nie mają swojego odpowiednika w systemie Linux. Przykładowo najnowsza wersja pakietu Microsoft Office nie będzie działała pod kontrolą systemu Linux (choć programistom z firmy *Codeweavers*¹ i innych udało się uruchomić pod systemem Linux starsze wersje pakietu). Jeżeli Twoja firma używa do edycji tekstu programu MS Word, do wykonywania kalkulacji programu MS Excel itd., to w systemie Fedora Core możesz spróbować wykorzystywać pliki danych z tych programów w pakietach *StarOffice* lub *OpenOffice* — jednak powinieneś się liczyć z tym, że nie zawsze będziesz mógł dokonać konwersji dokumentów bez przeszkód. Ponadto pakiet *OpenOffice* nie w pełni obsługuje wiele wbudowanych w pakiet Microsoft Office niestandardowych funkcji (np. makra, skrypty i formularze/pola).
- ♦ Dla środowiska Microsoft Windows istnieje wiele więcej komercyjnych, popularnych i „wypróbowanych w bojach” aplikacji, niż ma to miejsce w przypadku systemu Linux. Ze względu na fakt, że rynek opanowany przez system Windows jest tak ogromny, wiele firm programistycznych ogranicza się do pisania oprogramowania przeznaczonego wyłącznie dla systemu Windows.
- ♦ Aplikacje dla systemu Linux są z definicji (co wynika także z zaszłości historycznych) trudniejsze do prawidłowego zainstalowania i skonfigurowania niż wiele komercyjnych aplikacji dla środowiska Windows.

To były złe wiadomości. Dobra wiadomość jest taka, że Linux stopniowo zyskuje coraz więcej gorących zwolenników. Obecnie, korzystając z komputera pracującego pod kontrolą systemu Linux i wyposażonego w środowisko graficzne, możesz zrobić praktycznie wszystko to, co jest możliwe do uzyskania na takim samym komputerze, ale pracującym w systemie Windows. W najbliższej przyszłości perspektywa wymiany systemów operacyjnych większości komputerów na system Linux wydaje się być coraz bardziej realna. Jestem przekonany, że już niebawem (a na pewno wcześniej niż można by się tego spodziewać) system Linux stanie się znakomitą i powszechnie wykorzystywaną platformą do uruchamiania aplikacji. Niżej podajemy przyczyny, dla których tak może się stać.

¹ Więcej informacji na ten temat znajdziesz na witrynie <http://codeweavers.com> — przyp. tłum.

- ♦ Wśród coraz liczniejszych rzesz użytkowników wyraźnie rysuje się przekonanie, że przyszłość komputerów leży w aplikacjach sieciowych. W przeciwieństwie do pierwszych wersji systemu Windows, których korzenie sięgały jednozadaniowego i jednostanowiskowego systemu DOS, Fedora Linux jest oparty na systemie Unix, który od podstaw został zaprojektowany jako pełne środowisko wielozadaniowe dla całej grupy użytkowników działających w środowisku sieciowym — dzięki temu Fedora Linux udostępnia bardzo użyteczne mechanizmy wspomagające pracę w środowisku sieciowym.
- ♦ Nad projektowaniem i rozwojem aplikacji dla systemu Linux pracuje (dzięki idei powszechnej dostępności kodu źródłowego — licencja GNU) cała rzesza programistów tworzących silną społeczność szybko reagującą na potrzeby innych użytkowników systemu. Co więcej, ostatnimi czasy pojawia się coraz więcej firm programistycznych oferujących profesjonalne i w pełni komercyjne pakiety oprogramowania dla systemu Linux.
- ♦ Idea otwartości i powszechnej dostępności oprogramowania jest swego rodzaju motywem przewodnim systemu Linux oraz licencji GNU (*GNU is Not Unix*), co powoduje, że większość aplikacji tego systemu jest udostępniana bezpłatnie bądź za niewielką tylko opłatą. W praktyce oznacza to dla użytkownika możliwość bezpłatnego lub przeprowadzonego bardzo tanim kosztem testowania większości dostępnych aplikacji. Uzyskanie pełnych wersji pakietów oprogramowania dla systemu Linux jest naprawdę bardzo tanie. Przykładowo pakiet OpenOffice dołączony do dystrybucji Fedora Core (wersja *open source* pakietu StarOffice) jest dostępny bezpłatnie, natomiast pudełkowa wersja pakietu StarOffice (zawierająca instalacyjne dyski CD wraz kompletem dokumentacji w postaci książkowej) kosztuje tylko 79,95 dolarów (<http://www.sun.com/staroffice>) — z drugiej strony, jeżeli chcesz używać np. tylko edytora MS Word, to i tak musisz zapłacić co najmniej kilkaset dolarów za cały pakiet MS Office².
- ♦ Dla dotychczasowych użytkowników systemu Windows migracja do środowiska Linux staje się coraz łatwiejsza, zwłaszcza dzięki projektom, takim jak WINE. Pakiet WINE (opisany szczegółowo w dalszej części tego rozdziału) pozwala na uruchamianie bezpośrednio w systemie Linux wielu aplikacji przeznaczonych dla systemu Windows. WINE udostępnia również odpowiednie mechanizmy dla programistów, którzy chcieliby w prosty sposób przenieść swoje oprogramowanie na platformę Linux. Dotąd największe postępy poczyniono w zakresie uruchamiania w systemie Linux gier i aplikacji biurowych przeniesionych z systemu Windows.

Z całą pewnością można powiedzieć, że dla większości użytkowników decyzja o kompletnym zarzuceniu systemu Windows i migracji na platformę Linux będzie wymagała mniejszego czy większego wysiłku. Z drugiej jednak strony, jeżeli jesteś zdecydowany rozpocząć swoją przygodę z systemem Fedora Linux, to istnieje już kilka sposobów ułatwiających ten proces. W systemie Linux pojawiło się już sporo emulatorów, które umożliwiają uruchamianie programów przeznaczonych dla innych systemów operacyjnych, dostępne są również różnego rodzaju konwertery umożliwiające przetworzenie plików danych z formatów używanych np. przez pogramy graficzne i edytory tekstu systemu Windows, na formaty akceptowane w aplikacjach dostępnych na platformie Linux.

² W chwili obecnej (wrzesień 2004 r.) cena pełnego pakietu MS Office 2003 Pro PL kształtuje się na poziomie 2300 – 2400 zł — *przyp. tłum.*



Więcej informacji na temat eksportowania oraz importowania danych z programów graficznych i edytorów tekstu znajdziesz w rozdziale 6.

Jeżeli swój system Linux masz zainstalowany na komputerze PC, to są duże szanse na to, że już kiedyś zapłaciłeś za zakup systemu Windows 95, 98, Me, NT, 2000 bądź XP. W takiej sytuacji możesz korzystać z tych dwóch systemów na dwóch różnych komputerach bądź zainstalować je na dwóch różnych partycjach dysków tego samego komputera. To drugie rozwiązanie będzie wymagało restartowania komputera, za każdym razem kiedy będziesz chciał zmienić system operacyjny.

W następnym podrozdziale opiszemy szereg aplikacji systemu Linux, których możesz użyć jako odpowiedników dobrze znanych aplikacji systemu Windows.

Odszukiwanie odpowiedników aplikacji systemu Windows w systemie Linux

Jeżeli chcesz korzystać z systemu Fedora Linux na swoim komputerze PC, to będziesz miał np. pełne możliwości edytowania tekstów, przetwarzania grafiki oraz wykonywania obliczeń. Będziesz oczywiście mógł uruchamiać również inne programy, takie jak odtwarzacze audio i wideo, przeglądarki sieciowe czy też programy do obsługi poczty elektronicznej.



Przy wykorzystaniu technologii WINE programiści firmy *Codeweavers Inc.* utworzyli produkt o nazwie *CrossOver Office*, który pozwala na zainstalowanie i uruchomienie pod systemem Linux aplikacji wchodzących w skład pakietu *Microsoft Office*. Zapoznaj się z zawartością podrozdziału „Zastosowanie pakietu WINE do uruchamiania aplikacji przeznaczonych dla systemu Windows” znajdującego się w dalszej części rozdziału.

Aby mieć pewne rozeznanie w tym, jakie aplikacje są dostępne w systemie Linux, w tabeli 5.1 przedstawiono zestawienie popularnych aplikacji systemu Windows, ich odpowiedników w Linuksie oraz informacje, skąd można pobrać takie oprogramowanie. Mimo że wiele tych linuksowych odpowiedników nie osiągnęło jeszcze poziomu wyrafinowania ich konkurentów z Windows, to jednak z punktu widzenia kosztów zakupu mogą one być rozsądną alternatywą.

W kolejnych podrozdziałach omówimy zagadnienia związane z wyszukiwaniem i pracą aplikacji zaprojektowanych specjalnie dla systemu Linux.

Pobieranie aplikacji dla systemu Fedora Linux

Jako użytkownik systemu Linux możesz być pewny, że w żadnym sklepie z oprogramowaniem nie znajdziesz zbyt wielu pozycji dla Twojego systemu (jeśli w ogóle znajdziesz tam jakiegokolwiek aplikacje dla tego systemu). Najlepszym sposobem na zaopatrzenie się w nowe aplikacje (niedołączone do dystrybucji Fedora Core) jest po prostu

Tabela 5.1. Linuksowe odpowiedniki aplikacji systemu Windows

Aplikacja Windows	Odpowiednik w systemie Linux	Skąd pobrać?	Koszt
Microsoft Office (pakiet biurowy)	OpenOffice (http://openoffice.org)	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
	Koffice	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
	StarOffice	http://www.sun.com/staroffice	\$79,95
Microsoft Word (edytor tekstu)	OpenOffice Writer	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
	AbiWord	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
	Kword	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
Microsoft Excel (arkusz kalkulacyjny)	OpenOffice Calc	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
	Gnumeric	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
	Kspread	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
Microsoft PowerPoint (program do tworzenia prezentacji)	OpenOffice Impress	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
	KPresenter	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
Microsoft Internet Explorer (przeglądarka sieciowa)	Mozilla	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
	Epiphany	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
	Konqueror	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
	Opera	http://www.opera.com	do pobrania bezpłatnie
Microsoft Outlook (program do obsługi poczty elektronicznej)	Evolution	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
	Kmail	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
	Mozilla Mail	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
Adobe Photoshop (edytor plików graficznych)	The Gimp (gimp)	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie

Tabela 5.1. *Linuksowe odpowiedniki aplikacji systemu Windows — ciąg dalszy*

Aplikacja Windows	Odpowiednik w systemie Linux	Skąd pobrać?	Koszt
Microsoft Front Page (edytor plików HTML)	Quanta	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
Quicken lub Microsoft Money (zarządzanie osobistymi finansami)	Gnucash	Załączony na dyskach instalacyjnych CD Fedora Linux	bezpłatnie
Quicken lub Microsoft Money	gnucash	Zawarty na płycie instalacyjnej systemu Fedora Linux	bezpłatnie
AutoCAD (program do wspomagania projektowania)	LinuxCAD	http://www.linuxcad.com	\$159,–

pobieranie ich z internetu, choć niektóre witryny WWW zajmujące się oprogramowaniem dla systemu Linux oferują możliwość zamówienia takiego oprogramowania drogą wysyłkową na dyskach CD.

Pakiety oprogramowania kompilowane z myślą o systemie Fedora Linux i innych dystrybucjach Red Hat Linux prawie zawsze są dostępne w formacie RPM. A zatem, gdy zaczniesz szukać w internecie aplikacji dla systemów Linux (więcej na ich temat w dalszej części rozdziału), rozglądaj się za zbiorami pakietów RPM utworzonych specjalnie dla posiadanej wersji systemu Fedora Core.

A może poszukać na pulpicie?...

Z każdą kolejną dystrybucją systemu Fedora Core sukcesywnie powiększa się ilość wysokiej jakości aplikacji, które zostają po prostu zintegrowane jako część środowiska graficznego GNOME lub KDE. Innymi słowy, może się okazać, że, poszukując doskonałych aplikacji biurowych, gier, odtwarzaczy multimedialnych czy też narzędzi komunikacyjnych, nie będziesz musiał szukać dalej niż... w menu głównym swojego systemu operacyjnego.

Pamiętaj zatem, że zanim zaczniesz surfować po internecie w poszukiwaniu oprogramowania, którego akurat potrzebujesz, powinieneś sprawdzić, czy taka lub podobna aplikacja nie została już zainstalowana razem z systemem Fedora Linux. W rozdziale 6. omówimy szczegółowo szereg aplikacji przeznaczonych dla biura, do gier, aplikacji multimedialnych i komunikacyjnych, które są standardowo dołączane do dystrybucji Fedora Core lub są dostępne w prosty sposób z internetu. W dodatku B umieszczono listę pakietów oprogramowania, które są dostępne w dystrybucji Fedora Core załączonej do tej książki.



Aby być na bieżąco z uaktualnieniami pakietów oprogramowania wchodzących w skład dystrybucji Fedora Core, firma Red Hat oferuje usługę automatycznego pobierania i instalowania pakietów. W rozdziale 10. wyjaśniono, w jaki sposób za pomocą programów *up2date* i *yum* uaktualniać oprogramowanie dla systemu Fedora Core.

Poszukiwanie aplikacji w sieci internetowej

Jeśli nie znasz jeszcze nazw aplikacji, z których chcesz skorzystać w systemie Linux, w internecie istnieje mnóstwo miejsc, gdzie można ich poszukać. Gdy jednak szukasz konkretnego programu, najlepszym rozwiązaniem może być skorzystanie ze zbioru oprogramowania zawierającego pakiety utworzone specjalnie z myślą o systemie Fedora Linux.

Aby zdobyć wysokiej jakości popularne aplikacje niedołączone do używanej dystrybucji Linuksa, najlepiej skorzystać ze zbioru oprogramowania gromadzącego pakiety RPM przeznaczone dla posiadanej wersji systemu. W celu uzyskania instrukcji pozwalających na odszukanie zbiorów pakietów dla systemu Fedora Linux obsługiwanych przez program *yum* i skorzystanie z nich należy zapoznać się z zawartością następnego podrozdziału „Pobieranie i instalowanie aplikacji za pomocą programu yum”.

Poniżej przedstawiono listę kilku witryn w internecie, które są dobrym punktem startu do poszukiwania aplikacji dla systemu Linux.

- ♦ **Freshmeat** (<http://www.freshmeat.net>) — witryna *Freshmeat* udostępnia ogromny indeks oprogramowania Linux, które można pobrać z internetu. Strona umożliwia szukanie aplikacji przez wprowadzanie słów kluczowych i przeglądanie programów według kategorii.
- ♦ **Sourceforge** (<http://www.sourceforge.net>) — na tej witrynie znajdziemy tysiące projektów oprogramowania typu *open-source*. Użytkownik ma możliwość pobierania zarówno oprogramowania, jak i dokumentacji bezpośrednio z tej witryny.
- ♦ **Tucows Linux** (<http://linux.tucows.com>) — witryna udostępnia zarówno bezpłatne, jak i komercyjne pakiety oprogramowania w systemie Linux. Znajdziesz tutaj również bieżące artykuły i ciekawostki ze świata Linuksa oraz ranking pobieranych z tej witryny pakietów oprogramowania podzielonych według kategorii.

Kiedy zamawiasz oprogramowanie komercyjne, to zazwyczaj otrzymujesz je w postaci instalacyjnego dysku CD. Sam proces instalacji jest często bardzo uproszczony, a dodatkowo zazwyczaj dostarczana jest pełna dokumentacja programu w postaci drukowanej (książkowej). Oczywiście, jeżeli pobierzesz oprogramowanie z internetu, nie ruszając się nawet sprzed komputera, będziesz mógł od razu z aplikacji skorzystać.



Czasami pakiety oprogramowania są dostępne zarówno w formacie *libc5*, jak i *libc6*. To oznaczenie odnosi się do wersji bibliotek języka C, które były używane do tworzenia aplikacji. Jeżeli masz możliwość wyboru, to wybierz pakiety w wersji *libc6* — są kompatybilne z systemem Red Hat Linux 7 i późniejszymi (lub dowolnym jądrem systemu Linux w wersji 2.2 lub wyższej). W obecnej chwili wszystkie ważniejsze dystrybucje systemu Linux używają formatu *libc6*. W miarę możliwości warto poszukać pakietów przeznaczonych dla posiadanej wersji dystrybucji, takiej jak Fedora Core 2 lub Red Hat Linux 9.

Jeśli mniej więcej wiesz, czego szukasz, możesz zaglądać na serwery FTP zawierające pakiety RPM. Najpierw warto zapoznać się z plikami *README* i *INDEX*, aby zorientować się, czy na serwerze znajduje się poszukiwane oprogramowanie. Poniżej przedstawiono listę kilku miejsc, od których możesz rozpocząć poszukiwania pakietów RPM.

- ♦ **Fedora Extras** (<http://www.fedora.us>) — osoby realizujące projekt Fedora Project oczekują, że strona będzie rekomendowana jako miejsce, skąd można pobrać dodatkowe pakiety przeznaczone dla systemu Fedora Linux. Z czasem może się okazać, że strona Fedora Extras stanie się oficjalnym miejscem gromadzącym dobrej jakości aplikacje, które nie znajdują się w podstawowej dystrybucji systemu Fedora Linux.
- ♦ **Rpmfind** (<http://www.rpmfind.net>) — witryna oferuje pakiety oprogramowania w formacie RPM (*Red Hat Package Management*). Witrynę można przeszukiwać przy użyciu słów kluczowych, a następnie pobierać znalezione pakiety *.rpm*.
- ♦ **FreshRPMs** (<http://www.freshrpms.net>) — kolejna witryna zawierająca wiele znakomitych pakietów RPM.
- ♦ **Lista mirrorów serwera FTP firmy Red Hat zawierających dystrybucję Fedora Core** (<http://fedora.redhat.com/download/mirrors.html>) — na stronie prezentowana jest lista serwerów FTP, które udostępniają oprogramowanie dla systemu Fedora Core. Większość z nich udostępnia również cały szereg innych aplikacji zarówno bezpłatnych, jak i na licencji shareware.

Często jest tak, że nie wystarczy pobrać pakiet, aby skorzystać z aplikacji, które się tam znajdują. Wiele pakietów powiązано z innymi. Przykładowo pakiety oprogramowania służącego do odtwarzania plików dźwiękowych i wideo zwykle są uzależnione od innych pakietów zawierających dekodery różnego typu danych. W celu rozwiązania tego problemu do dystrybucji Fedora Core dołączono pakiet *yum*.

Pobieranie i instalowanie aplikacji za pomocą programu yum

Pakiet *yum* (*Yellow Dog Updater Modified*) pozwala zainstalować i uaktualnić wybrane pakiety oprogramowania formatu RPM znajdujące się w sieciowych zbiorach aplikacji. Gdy jest znana nazwa pakietu oprogramowania, program *yum* prawdopodobnie jest najlepszym narzędziem pozwalającym na pobranie i zainstalowanie pakietu.

Pakiet *yum* znajduje się na dysku CD dystrybucji Fedora Core dołączonej do książki. W celu zainstalowania pakietów RPM za pomocą programu *yum* należy wykonać następujące, podstawowe kroki.

1. **Określ żądany pakiet oprogramowania.** Wiele popularnych dodatkowych pakietów dla systemu Linux powstało dla konkretnych wersji dystrybucji Fedora Core (1 i 2) i Red Hat Linux (8 i 9); umieszczono je w sieciowych zbiorach oprogramowania. Na stronie internetowej poświęconej programowi *yum* (<http://linux.duke.edu/projects/yum/repos>) znajduje się niewielka lista zbiorów oprogramowania. W celu wyświetlenia szczegółowych informacji, które będzie można skopiować i wkleić do własnego pliku *yum.conf*, należy kliknąć dowolną z nazw zbiorów.
2. **Skonfiguruj program yum.** W pliku */etc/yum.conf* konieczne jest określenie zbioru zawierającego żądane oprogramowanie. Po wykonaniu tej operacji możliwe będzie zainstalowanie dowolnego pakietu znajdującego się w zbiorze.

- 3. Uruchom program *yum*.** W celu pobrania i zainstalowania dowolnego pakietu zawartego w zbiorze włącznie z pakietami z nim powiązanymi należy wykonać polecenie *yum*.



W przypadku systemu Red Hat Linux i jak dotąd również systemu Fedora Core firma Red Hat Inc. dołożyła wszelkich starań, aby zagwarantować dobrą jakość oferowanego oprogramowania nie wzbudzającego żadnych zastrzeżeń w kwestii patentów. Podczas pobierania pakietów spoza dystrybucji systemu Linux oferowanych przez firmę Red Hat we własnym zakresie należy sprawdzić ich jakość i legalność.

Program *yum*, poza umożliwianiem pobrania i instalacji nowych pakietów oprogramowania, może też posłużyć do sprawdzania dostępności uaktualnień i wyświetlania różnych informacji na temat dostępnych pakietów.

Konfigurowanie programu *yum* (plik */etc/yum.conf*)

Plik */etc/yum.conf* zawiera predefiniowaną konfigurację umożliwiającą zainstalowanie podstawowych pakietów systemu Fedora Core, a także ich uaktualnień. A zatem, w celu uaktualnienia pakietów będących częścią aktualnej wersji systemu Fedora Core prawdopodobnie w ogóle nie będzie konieczne modyfikowanie pliku *yum.conf*.

Aby było możliwe pobranie ze zbioru oprogramowania niedołączonego do dystrybucji Fedora Core, konieczne będzie dodanie do pliku */etc/yum.conf* informacji na temat tego zbioru. Oto przykład wpisu pliku *yum.conf* odwołującego się do takiego zbioru:

```
[freshrpms-fc-1]
name=Freshrpms - pakiety dla systemu Fedora Core 2
baseurl=http://ayo.freshrpms.net/fedora/linux/2/i386/freshrpms
```

W przykładzie w sekcji *freshrpms-fc-1* nazwa zbioru jest identyfikowana przez łańcuch *Freshrpms - pakiety dla systemu Fedora Core 2*. Gdy przy użyciu programu *yum* spróbujesz pobrać pakiet oprogramowania, poszukiwania odbędą się w podkatalogach katalogu */fedora/linux/2/i386/freshrpms*, który znajduje się na serwerze *http://ayo.freshrpms.net*.

Zastosowanie programu *yum* do pobrania i zainstalowania pakietów RPM

Po zidentyfikowaniu zbioru pakietów RPM w pliku *yum.conf* w celu pobrania i zainstalowaniażądanego pakietu wystarczy jedynie uruchomić program *yum* z opcją *install*. Po nawiązaniu połączenia z internetem w oknie terminala należy zalogować się jako użytkownik *root*.

Na początku program *yum* pobiera nagłówki wszystkich pakietów, które możesz pobrać ze zbioru. Po zaprezentowaniu listy powiązań żądanych pakietów program *yum* pyta użytkownika, czy chce zainstalować wymagane pakiety. Oto przykład zastosowania programu *yum* do pobrania odtwarzacza multimedialnego *mplayer*.

```
# yum install mplayer
Gathering header information file(s) from server(s)
Server: Fedora Core 2
Server: Freshrpms - pakiety dla systemu Fedora Core 2
Finding updated packages
```

```

Downloading needed headers
Resolving dependencies
..Dependencies resolved
I will do the following:
[install: player 1.0-0.9.20040415.1.fc2.fr.i386]
I will install/upgrade these to satisfy the dependencies:
.
.
.
Is this ok [y/N]: y
.
.
.
Getting mplayer-fonts-1.1-1.fr.noarch.rpm
Getting mplayer-1.0-0.1.20031002.fr.i386.rpm
Calculating available disk space - this could take a bit
mplayer-fonts 100 % done 2/14
mplayer 100 % done 14/14
Installed: mplayer-1.0-0.1.20031002.fr.i386
Dep Installed: mplayer-fonts-1.1-1.fr.noarch
Transaction(s) Complete

```

W powyższym przykładzie widać, że program *yum* sprawdził trzy różne zbiory oprogramowania: Fedora Core 2 (pliki podstawowe i uaktualnienia) i Freshrpms - pakiety dla systemu Fedora Core 2. Po wyświetleniu zależnych pakietów narzędzie *yum* pyta użytkownika, czy je zainstalować. Po wprowadzeniu znaku *y* wybrany pakiet i wszystkie powiązane z nim pakiety zostaną zainstalowane.

Zastosowanie programu yum do wyświetlania listy pakietów i ich uaktualniania

Program *yum*, poza pobieraniem i instalowaniem nowych pakietów RPM, może też posłużyć do wyświetlania listy dostępnych pakietów i uaktualniania już zainstalowanych. Oto przykład ilustrujący jedno z zastosowań programu *yum*:

```
# yum check-update
```

Opcja *check-update* powoduje, że program *yum* sprawdza, czy w zbiorach oprogramowania są dostępne uaktualnienia zainstalowanych wersji pakietów RPM. Jeśli pakiet wymaga aktualizacji, należy użyć opcji *update*. Przykładowo w celu uaktualnienia pakietu *nmap-frontend* należy wykonać polecenie:

```
# yum update nmap-frontend
```

Jeśli zależy Ci na zapoznaniu się z listą wszystkich pakietów, które można pobrać z używanych zbiorów, wykonaj następujące polecenie:

```
# yum list | less
```

Przez umieszczenie na końcu polecenia *less* możliwe będzie stopniowe przeglądanie listy pakietów (w zależności od wybranych zbiorów lista może być długa). Jeśli podejmie się próbę instalacji pakietu i nieudana operacja zakończy się wyświetleniem komunikatu typu: pakiet *xyz* wymaga brakującego pliku *xyz*, za pomocą opcji *provides* można sprawdzić, które pakiety zawierają ten plik. Oto przykład:

```
# yum provides brakujący_plik
```

Po zastosowaniu opcji `provides` program *yum* poszuka w zbiorach pliku o podanej nazwie (wstawionej w miejsce łańcucha *brakujący_plik*) i wyświetli nazwę każdego pakietu, w którym plik zostanie znaleziony.

Pobieranie oprogramowania dla systemu Linux

Pakiety oprogramowania dla systemu Linux można pobierać i instalować za pomocą programu *yum*, można je także pobrać z internetu zarówno za pomocą przeglądarki sieciowej (np. *Mozilla*), jak i klienta FTP (np. uruchamianego poleceniem *ncftp*). Przeglądarka sieciowa zwykle pozwala na przeglądanie zawartości serwera FTP udostępnianej przez interfejs HTTP (poszukaj pliku *index.html* zlokalizowanego zwykle w głównym katalogu serwera). Korzystanie z polecenia *ftp* (np. w wariancie programu *ncftp*) oferuje więcej możliwości, ale jest mniej intuicyjne. Istnieją również aplikacje klienta FTP działające w środowisku graficznym, takie jak np. *gFTP*, które zdecydowanie upraszczają korzystanie z usług serwerów FTP (polecenie *gFTP* zostało szczegółowo opisane w rozdziale 9.).



Opisane poniżej procedury zakładają, że dysponujesz aktywnym połączeniem z internetem.

Pobieranie oprogramowania przy użyciu przeglądarki Mozilla

Aby pobrać pakiet oprogramowania z internetu przy użyciu przeglądarki *Mozilla*, wykonaj poniższe polecenia.

1. Uruchom przeglądarkę sieciową *Mozilla* (ikona tej przeglądarki jest dostępna na panelu zarówno środowiska KDE jak i GNOME).
2. W polu adresu przeglądarki wpisz nazwę serwera FTP, z którego chcesz pobrać oprogramowanie i naciśnij klawisz *Enter*.
3. Aby przejść do katalogu, niżej wystarczy kliknąć jego ikonę; aby powrócić do katalogu nadrzędnego, kliknij łącze *Up to higher level directory*.
4. Po odszukaniu potrzebnego pakietu oprogramowania ustaw wskaźnik myszy nad reprezentującą go ikoną, kliknij prawym klawiszem myszy i z menu podręcznego wybierz polecenie *Save Link Target As*.
5. Na ekranie pojawi się okno dialogowe *Save As*. W celu określenia miejsca, w którym zostanie zapisany pakiet, kliknij przycisk *Go Up a Level* (spowoduje przejście w hierarchii katalogów o jeden poziom wyżej) lub dwukrotnie kliknij docelowy katalog.
6. Naciśnij przycisk *OK*.

W czasie pobierania pakietu na ekranie wyświetlane jest okno *Saving*, zawierające informacje na temat postępu pobierania pliku. Obecnie *Mozilla* jest wyposażona w ładnie wyglądający, całkiem nowy menedżer pobierania ułatwiający monitorowanie operacji pobierania wielu plików. Po zakończeniu pobierania aplikacja jest gotowa do rozpakowania i zainstalowania (lub od razu do zainstalowania, jeżeli pobierałeś pakiet w formacie *RPM*).

Pobieranie oprogramowania przy użyciu programu ncFTP

Jeżeli do pobierania plików z internetu wolisz wykorzystywać programy pracujące w trybie tekstowym (zamiast programu Mozilla), możesz wykorzystać jeden z kilku tekstowych klientów FTP, dostępnych w systemie Fedora Linux. Jednym z najbardziej popularnych i lubianych programów jest *ncftp* (inne to *ftp* i *lftp*). Poniżej podano przykład zastosowania klienta *ncftp*.



Polecenie *sftp* oferuje bezpieczniejszą metodę łączenia się z serwerami FTP. Aby jednak było to możliwe, serwer FTP musi obsługiwać żądania programu *ssh* (nie wszystkie serwery spełniają ten warunek). W celu połączenia się z serwerem FTP za pomocą narzędzia *sftp* należy wykonać polecenie `sftp uzytkownik@ftp.abc.com`. Po nawiązaniu połączenia należy użyć poleceń zawartych w poniższej procedurze w celu wykonania dalszych operacji.

1. W oknie terminala bądź bezpośrednio z poziomu powłoki systemu wpisz polecenie `ncftp lokalizacja`, gdzie parametr *lokalizacja* powinieneś zastąpić adresem serwera FTP, z którym chcesz się połączyć, przykładowo:

```
$ ncftp metalab.unc.edu
```

lub

```
$ ncftp -u kuba ftp://ftp.mojserwerftp.com.pl
```

Jeżeli nie podasz nazwy użytkownika (tak jak w pierwszym przykładzie), to *ncftp* będzie próbował zalogować się jako użytkownik *anonymous*. Do adresu można też dodać nazwę użytkownika (np. `kuba@ftp://ftp.mojserwerftp.com.pl`).

Jeżeli podasz nazwę użytkownika (patrz drugi przykład, opcja `-u kuba`), to zostaniesz poproszony o wpisanie hasła dostępu do serwera FTP dla tego użytkownika.

2. Po zalogowaniu się w celu odszukania interesującego Cię pakietu oprogramowania możesz skorzystać z następujących poleceń.

- ♦ `ls` — wyświetla zawartość bieżącego katalogu.
- ♦ `cd nazwa_katalogu` — zmienia bieżący katalog na katalog o podanej nazwie (przykładowo, `cd /pub/Linux/apps/doctools`). Jeżeli chcesz powrócić do katalogu nadrzędnego, to powinieneś skorzystać z polecenia `cd ...`.

3. Wpisz polecenie `binary` (aby upewnić się, że plik zostanie przesłany w postaci binarnej).
4. Aby pobrać plik zlokalizowany w katalogu bieżącym, wpisz polecenie `get nazwa_pliku`, gdzie parametr *nazwa_pliku* odpowiada pełnej nazwie pobieranego pliku. Aby np. pobrać aplikację o nazwie *whichman* (zakładając, że `/pub/Linux/apps/doctools` jest obecnie katalogiem bieżącym), powinieneś wpisać polecenie:


```
> get whichman-2.1.tar.gz
```
5. Po zakończeniu pobierania wykonaj polecenie `exit`.



Zanim uruchomisz polecenie *ncftp*, upewnij się, że bieżącym katalogiem jest katalog, w którym docelowo chcesz umieścić ściaganą z serwera FTP aplikację. Alternatywną metodą jest (już po uruchomieniu *ncftp*) posłużenie się poleceniem `lcd`. Przykładowo aby zmienić lokalny katalog bieżący na `/tmp/abcapp`, powinieneś wykonać polecenie `lcd /tmp/abcapp`.

Konwencje nazywania pakietów oraz ich formaty

Jeśli tylko jest to możliwe, aplikacje używane w systemie Fedora Core warto instalować z pakietów RPM (pliki z rozszerzeniem *.rpm*). Jeśli jednak pakiet RPM nie jest dostępny, żądane oprogramowanie może istnieć w postaci pakietów innego formatu.

Załóżmy, że właśnie pobrałeś z internetu kilka plików, których nazwy pełne są różnych cyferek, kropek, oznaczeń, *gzów* i innych *tarów* i z pewnością zadajesz sobie pytanie, co to wszystko znaczy? Nie musisz się jednak przejmować — nie jest to tak skomplikowane, jakby się na pierwszy rzut oka wydawało.

Zdecydowana większość powszechnie spotykanych skompresowanych plików zawierających aplikacje dla systemu Linux jest nazywana zgodnie z konwencją nazw pakietów oprogramowania określoną przez zalecenia organizacji GNU. Poniżej podano kilka przykładów nazw pakietów zgodnych z tą konwencją:

```
mojaaplikacja-4.2.3-1.i386.rpm
mojaaplikacja-4.2.3.tar.gz
mojaaplikacja-4.2.3.src.tar.gz
mojaaplikacja-4.2.3.bin.SPARC.tar.gz
mojaaplikacja-4.2.3.bin.ELF.static.tar.gz
```

Powyższe przykłady reprezentują kilka różnych pakietów tej samej aplikacji. Nazwą tego pakietu jest *mojaaplikacja*. Po nazwie następuje zestaw liczb określających wersję pakietu, w naszym przypadku jest to 4.2.3 (numer głównej wersji to 4, po którym następuje numer podwersji oraz uaktualnień 2.3). Po numerze wersji występuje kropka, po niej umieszcza się zazwyczaj kilka opcjonalnych informacji o pakiecie, a na końcu informacje o rodzaju programu użytego do skompresowania i archiwizacji całego pakietu.

Pierwszy przykład reprezentuje plik zapisany w formacie Red Hat Package Manager (*.rpm*). Ciąg *.i386* widoczny przed rozszerzeniem *.rpm* wskazuje, że pakiet zawiera pliki binarne przeznaczone do uruchamiania na komputerach opartych o architekturę Intel i386 (potocznie mówiąc, przeznaczone dla pecetów). Łańcuch *-1* identyfikuje numer kompilacji (w celu wprowadzenia drobniejszych zmian ten sam pakiet mógł być wielokrotnie ponownie kompilowany). W ramce zatytułowanej „Pakiety RPM kontra kompilacja kodu źródłowego” znajdziesz zestawienie wad i zalet pakietów RPM porównanych do „klasycznej” metody własnoręcznej kompilacji kodu źródłowego aplikacji.

Kolejne dwa przykłady nazw reprezentują pliki zawierające kody źródłowe aplikacji. Poszczególne pliki składowe aplikacji zostały zarchiwizowane za pomocą programu *tar* i następnie skompresowane przy użyciu programu *gzip*. Aby przygotować pakiet do instalacji, będziesz musiał użyć tych dwóch poleceń (lub polecenia *tar* z opcją *-z*) do przywrócenia plików składowych do pierwotnej postaci.

Między numerem wersji aplikacji i przyrostkami *.tar.gz* mogą się znajdować, oddzielone kropkami, dodatkowe znaczniki, reprezentujące dodatkowe informacje o zawartości pakietu. Jeżeli dany pakiet zawiera pliki binarne, umieszczana jest tam informacja, dla jakiej architektury są one przeznaczone. W przykładzie trzecim opcjonalny znacznik *.src* został umieszczony po to, aby odróżnić pakiet zawierający kody źródłowe od pakietu zawierającego pliki binarne. W przykładzie czwartym znacznik *.bin.SPARC* informuje, że zawartość pakietu stanowią pliki binarne przeznaczone dla stacji roboczej SPARC. Ostatni przykład

reprezentuje pakiet binarny, składający się ze statycznie skonsolidowanych plików wykonywalnych zapisanych w formacie *ELF*.

Poniżej przedstawiono szczegółowy opis kolejnych części, z jakich składa się nazwa pakietu zgodna z GNU.

- ♦ **nazwa aplikacji** — zazwyczaj jest to pisany małymi literami ciąg znaków identyfikujący daną aplikację.
- ♦ **myślnik (-)**.
- ♦ **numer wersji** — kolejne liczby, oddzielone kropkami, reprezentujące numer wersji aplikacji i jej kolejną aktualizację.
- ♦ **kropka (.)**.
- ♦ **src lub bin** — znacznik opcjonalny, zazwyczaj jeżeli nie jest podany, to przyjmuje się, że pakiet zawiera kod źródłowy (*src*).
- ♦ **kropka (.)**.

Pakiety RPM kontra kompilacja kodu źródłowego

Pakiety binarne zapisane w formacie RPM można przy użyciu odpowiednich narzędzi systemu Fedora Linux w prosty sposób zainstalować, a co więcej, dają się one łatwo zarządzać oraz w razie potrzeby odinstalować. Jest to metoda postępowania polecana dla użytkowników, którzy nie mają jeszcze zbyt dużego doświadczenia w pracy z systemem Fedora Core. Zdarzają się jednak sytuacje, w których preferowanym rozwiązaniem jest budowanie aplikacji od podstaw korzystając z kodu źródłowego. Poniżej przedstawiono kilka argumentów i dla obydwu rozwiązań:

- ♦ **RPM** — instalacja aplikacji rozprowadzanych w pakietach RPM jest stosunkowo prosta. Po zainstalowaniu aplikacji użytkownik dysponuje narzędziami do zarządzania, weryfikacji, aktualizacji oraz usuwania takiej aplikacji, dostępnymi zarówno z poziomu powłoki systemu jak i graficznego interfejsu użytkownika. Korzystając z pakietów RPM użytkownik nie musi nic wiedzieć na temat programów konsolidujących i kompilatorów. Narzędzia do instalacji pakietów RPM sprawdzą nawet, czy wszystkie inne pakiety wymagane do poprawnego działania danej aplikacji są już zainstalowane. Ponieważ specyfikacja formatu RPM została przez firmę Red Hat udostępniona na zasadach licencji GNU, to coraz częściej również inne dystrybucje systemu Linux zaczynają wykorzystywać ten format do rozpowszechniania swoich aplikacji. Ze względu na wiele zalet RPM należy więc oczekiwać, że większość powszechnie dostępnych aplikacji dla systemu Linux albo już jest albo niebawem będzie dostępna w formacie RPM.
- ♦ **Kompilacja kodu źródłowego** — nie wszystkie pakiety RPM zawierające kody źródłowe dysponują odpowiednikami binarnymi. Z tego powodu teoretycznie może się zdarzyć sytuacja, w której najnowsza dostępna wersja RPM danej aplikacji będzie już o kilka wersji starsza od najnowszej wersji dostępnego kodu źródłowego tej aplikacji. W takiej sytuacji użytkownik może po prostu pobrać kod źródłowy aplikacji i po wykonaniu kilku poleceń `tar` i `make` mieć do dyspozycji jej najnowszą wersję. Co więcej, jeżeli jesteś doświadczonym użytkownikiem, możesz przed kompilacją zmodyfikować kod źródłowy danej aplikacji tak aby jej funkcjonalność lepiej odpowiadała Twoim indywidualnym potrzebom.

W celu uzyskania dodatkowych informacji na temat pakietów RPM należy zapoznać się z książką *Red Hat RPM Guide* autorstwa Erica Fostera-Johnsona (wydaną przez Red Hat Press/Wiley, 2003 r.).

- ♦ **rodzaj pliku binarnego** — część opcjonalna, mogąca składać się z kilku kolejnych znaczników określających zawartość archiwum binarnego. Przykładowo *i386* oznacza pliki binarne przeznaczone dla architektury procesorów Intel (386, 486, Pentium itd.), *SPARC* oznacza pliki binarne przeznaczone dla architektury procesorów SPARC.
- ♦ **kropka (.)**.
- ♦ **typ archiwum** — zazwyczaj do utworzenia archiwum używany jest program *tar* (znacznik *.tar*).
- ♦ **typ programu pakującego** — zazwyczaj do pakowania używany jest program *gzip* (znacznik *.gz* lub *.gzip*).

Korzystanie z innych formatów pakietów i dokumentów

Wiele pakietów oprogramowania niepowiązanych z konkretnymi dystrybucjami systemu Linux (np. Fedora Core lub Debian) zazwyczaj wykorzystuje format *tar/gzip* do archiwizacji i pakowania plików. Przeglądając zawartość stron internetowych poświęconych aplikacjom, możesz się jednak czasami spotkać z plikami o innych rozszerzeniach³.



Poszukując aplikacji dla systemu Fedora Linux, zazwyczaj będziesz się spotykał z pakietami w formacie Red Hat Package Management (*.rpm*). Dopóty, dopóki nie zechcesz samodzielnie kompilować aplikacji (wtedy niezbędne będzie posiadanie kodu źródłowego), właśnie RPM jest formatem aplikacji, których powinieneś używać. Korzystanie z aplikacji przygotowanych w formacie RPM (powszechnie dostępnych na serwerach FTP z oprogramowaniem dla Fedora Linux) daje Ci ogromną szansę na to, że dany pakiet oprogramowania da się poprawnie zainstalować i będzie działał w systemie Fedora Core bez konieczności dokonywania dodatkowych modyfikacji. Kolejna polecana witryna znajduje się pod adresem *fedora.us*.

W tabeli 5.2 przedstawiono zestawienie różnych formatów plików, z którymi możesz spotkać się podczas przeszukiwania serwerów FTP udostępniających oprogramowanie dla systemu Linux. W tabeli 5.3 umieszczono zestawienie najpopularniejszych formatów dokumentów używanych do wymiany informacji w środowisku Linux.

Jeżeli nie jesteś pewny, w jakim formacie jest zapisany dany plik, możesz sprawdzić to za pomocą polecenia *file*, np.:

```
$ file nazwa_pliku_archiwum
```

Rezultatem działania powyższego polecenia będzie informacja o formacie danego pliku (czy jest to plik GNU *tar*, RPM, *gzip*, czy też jeszcze inny format). Polecenie *file* jest szczególnie przydatne, w sytuacji kiedy oryginalna nazwa pliku została zmieniona i ewentualne przyrostki pozwalające zidentyfikować typ pliku zostały utracone.

³ W zasadzie bardziej trafne byłoby tutaj określenie „o innych przyrostkach”, ponieważ w systemie Linux określenie „rozszerzenie” niemal całkowicie przestało mieć znaczenie znane np. z systemów DOS czy Windows — *przyp. tłum.*

Tabela 5.2. *Formaty plików archiwum używanych w systemie Linux*

Format	Przyrostek	Opis
<i>Pliki programu gzip</i>	<code>.gz</code> lub <code>.z</code>	Oznacza, że plik został spakowany za pomocą narzędzia GNU <i>gzip</i> . Aby go rozpakować, powinieneś użyć polecenia <i>gzip</i> lub <i>gunzip</i> (w zasadzie to te same programy)
<i>Pliki programu tar</i>	<code>.tar</code>	Oznacza, że plik został zarchiwizowany za pomocą narzędzia <i>tar</i> . Program <i>tar</i> jest wykorzystywany do tworzenia pojedynczego pliku archiwum złożonego z wielu plików składowych. Aby rozpakować takie archiwum, powinieneś użyć polecenia <i>tar</i> z odpowiednimi opcjami
<i>Pliki programów tar i gzip</i>	<code>.tgz</code>	Częstą praktyką stosowaną przy nazywaniu plików, będących archiwami programu <i>tar</i> skompresowanymi przy użyciu narzędzia <i>gzip</i> , jest nadawanie im rozszerzenia <code>.tgz</code>
<i>Bzip2</i>	<code>.bz2</code>	Oznacza, że plik został skompresowany przy użyciu programu <i>bzip2</i>
<i>Tar/compressed (spakowany tar)</i>	<code>.tarz</code> lub <code>.tz</code>	Oznacza, że plik został zarchiwizowany poleceniem <i>tar</i> , a następnie skompresowany poleceniem <i>compress</i>
<i>Linux Software Map</i>	<code>.lsm</code>	Oznacza, że plik zawiera informację tekstową opisującą zawartość archiwum
<i>Debian Binary Package</i>	<code>.deb</code>	Oznacza pakiet binarny używany w dystrybucji Debian Linux (w dalszej części niniejszego rozdziału znajdziesz informacje na temat sposobów konwersji pakietów w formacie <code>.deb</code> na <code>.rpm</code>)
<i>Red Hat Package Management</i>	<code>.rpm</code>	Oznacza pakiet binarny używany w dystrybucji Fedora Core. Format <code>.rpm</code> jest używany również w innych dystrybucjach

Instalowanie aplikacji w systemie Fedora Linux

W niniejszym podrozdziale omówimy zagadnienia dotyczące instalowania i zarządzania aplikacjami przy użyciu narzędzi przeznaczonych dla pakietów RPM. W dalszej części tego rozdziału dowiesz się również, w jaki sposób należy budować oraz instalować aplikacje, które są udostępniane w postaci kodu źródłowego.



Polecenie *rpm* może zostać użyte przy instalacji dowolnego pakietu zapisanego w formacie RPM. Jeśli zależy Ci na zainstalowaniu pakietów RPM należących do dystrybucji systemu Fedora Core, wtedy zamiast narzędzia *rpm* należy skorzystać z okna programu *Package Management*, służącego do dodawania i usuwania pakietów. Okno programu omówiono w rozdziale 2. W celu uaktualnienia istniejących pakietów należy użyć narzędzia *update* (opisanego w rozdziale 10.) lub *yum* (które zostało omówione wcześniej w tym rozdziale).

Instalacja pakietów RPM i zarządzanie nimi

Kiedy pobierasz np. z serwera FTP pakiet oprogramowania zapisany w formacie RPM, otrzymujesz tylko jeden plik. Narzędziem przeznaczonym do pracy z pakietami w formacie

Tabela 5.3. Formaty plików dokumentów używane w systemie Linux

Format	Przyrostek	Opis
<i>Hypertext Markup Language</i> (hipertekst)	<i>.html</i> lub <i>.htm</i>	Oznacza plik zapisany w języku HTML, który może być odczytywany za pomocą przeglądarek sieciowych (np. <i>Mozilla</i>)
<i>PostScript</i>	<i>.ps</i>	Oznacza plik postscriptowy, przeznaczony do wydruku na drukarkach obsługujących język PostScript
<i>SGML</i>	<i>.sgml</i>	Oznacza plik zapisany w formacie SGML, tzw. standardowym formacie zapisu plików. Format SGML jest często używany do tworzenia dokumentów, które następnie mogą być konwertowane na wiele różnych formatów
<i>DVI</i>	<i>.dvi</i>	Oznacza plik zapisany w formacie DVI, standardowym formacie procesora tekstu LaTeX. Za pomocą poleceń <i>dvips</i> lub <i>dvilj</i> można skonwertować takie dokumenty odpowiednio na format PostScript bądź na format PCL (używany w drukarkach firmy Hewlett-Packard)
<i>plik tekstowy</i>	brak	Pliki bez żadnych rozszerzeń są zazwyczaj w systemie Fedora Linux zwykłymi plikami tekstowymi (w formacie ASCII). Trzeba w tym miejscu wspomnieć, że mnóstwo plików poleceń linuxowych, takich jak zawarte w katalogach <i>/usr/bin</i> i <i>/usr/sbin</i> , też nie posiada rozszerzenia. Jeśli napotkałeś na plik pozbawiony rozszerzenia, przed wykonaniem jakiegokolwiek operacji najlepiej użyj polecenia <i>file</i> . Wykonanie tego polecenia dla pliku, który wcześniej nie został sprawdzony, może faktycznie zapobiec problemom. Plik z rozszerzeniem <i>.txt</i> zawierający sporo kodu binarnego może posłużyć do wywołania edytora tekstu i wykonania w systemie niewskazanych operacji

Red Hat Package Management jest polecenie *rpm*⁴. Do zarządzania tymi pakietami polecenie *rpm* posiada cały szereg opcji pozwalających na wyświetlanie listy zainstalowanych pakietów, aktualizację zainstalowanych pakietów do nowych wersji oraz wyświetlanie różnego rodzaju informacji dotyczących pakietów⁵ (np. listy plików czy dokumentacji załączonej do pakietu). Dodatkowo polecenie *rpm* posiada opcje pozwalające na weryfikację integralności pakietu (czyli sprawdzenie, czy wszystkie pliki składowe pakietu są aktualne i czy nie zostały w jakiś sposób zmodyfikowane).

Program *rpm* może pracować w następujących trybach:

- ♦ *instalowania* (-i),
- ♦ *uaktualniania* (-U),
- ♦ *odświeżania*⁶ (-F),
- ♦ *zapytywania* (-q),

⁴ Polecenie to nazywane jest również *menedżerem pakietów* — przyp. tłum.

⁵ W polskiej dokumentacji systemu ten tryb pracy nazywany jest *zapytywaniem pakietów* — przyp. tłum.

⁶ Jest to niezbyt fortunne sformułowanie, używane w polskiej dokumentacji polecenia *rpm*. W oryginalnej dokumentacji ten tryb pracy jest określany jako *upgrade*, więc bardziej na miejscu byłoby raczej określenie *tryb aktualizacji* — przyp. tłum.

- ♦ *weryfikowania* (-V),
- ♦ *sprawdzania sygnatury* (--checksig),
- ♦ *odinstalowywania* (-e),
- ♦ *przebudowywania bazy danych* (--rebuilddb),
- ♦ *nadawania uprawnień* (--setperms),
- ♦ *ustawiania właścicieli i grup* (--setugids),
- ♦ *pokazywania konfiguracji* (--showrc).

Posługując się poleceniem `rpm` z powyższymi opcjami, masz możliwość zarówno instalacji pakietów RPM, jak i weryfikacji ich integralności, co pozwala na szybkie usunięcie potencjalnych problemów, które mogą wystąpić podczas pracy z takimi pakietami. Możliwe jest też wykonanie specjalnych operacji, takich jak przebudowa bazy danych RPM i zmian praw właściciela. W celu dodania lub usunięcia pakietów konieczne jest zalogowanie się jako użytkownik *root*. Bez uprawnień użytkownika *root* można jednak uzyskać dostęp do listy zainstalowanych pakietów, wyświetlać informacje ich dotyczące i sprawdzać ich zawartość.

W kolejnych sekcjach omówimy zagadnienia związane z wykorzystaniem polecenia `rpm` do instalacji i pracy z pakietami RPM.

Instalacja pakietów z wykorzystaniem polecenia `rpm`

Aby zainstalować pakiet oprogramowania (który nie został jeszcze zainstalowany w systemie) za pomocą polecenia `rpm`, większość osób zazwyczaj korzysta z takich samych opcji jak w przypadku operacji aktualizacji (opcja `-U`). Poniżej zawarto przykład polecenia powodującego instalację nowego pakietu RPM.

```
# rpm -U [opcje] nazwa_pakietu
```

Parametr *nazwa_pakietu* reprezentuje nazwę pliku archiwum zapisanego w formacie RPM. Pakiet może znajdować się na dysku lokalnym (np. w bieżącym katalogu), na zamontowanym napędzie CD (np. `/mnt/cdrom/Fedora/RPMS/mojaaplikacja.rpm`) bądź nawet na serwerze FTP, do którego masz dostęp (np. `ftp://sunsite.unc.edu/pub/Linux/games/strategy/gaxis-1.7-1.i386.rpm`).



Przerwanie działania programu `rpm` w trakcie instalacji pakietu może spowodować zablokowanie plików i uszkodzenie bazy danych. W efekcie kolejne wykonanie polecenia `rpm` może zakończyć się niepowodzeniem. Jeśli do tego dojdzie, wtedy prawdopodobnie w celu wyeliminowania problemu konieczne będzie usunięcie istniejących blokad plików bazy danych. Jeśli to nie pomoże, wtedy należy sprawdzić, czy baza danych pakietów RPM uległa uszkodzeniu. Jeśli tak jest, należy ją ponownie utworzyć. Poniżej zawarto polecenia, które kolejno usuwają blokady plików, sprawdzają bazę danych i ponownie ją odbudowują (należy zalogować się jako użytkownik *root*):

```
# rm -f /var/lib/rpm/__.db*
# db_verify /var/lib/rpm/Packages
```

Łącznie z opcją `-U` możesz korzystać z opcji umożliwiających wyświetlanie informacji o postępie instalacji.

- ♦ -v — drukuje szczegółowe informacje o przebiegu kolejnych etapów instalacji⁷. Dołączenie tej opcji jest świetnym sposobem na zrozumienie, co tak naprawdę dzieje się podczas procesu instalacji (ponieważ ilość wyświetlanych informacji może być bardzo duża, to może należałoby rozważyć zastosowanie polecenia `| less`). Wielokrotnie używając opcji -v (np. -vv), możesz uzyskać większą ilość informacji.
- ♦ -h — odwzorowuje postępowanie procesu rozpakowywania archiwum za pomocą drukowania na ekranie kolejnych znaków #. Pojedynczy znak # odpowiada postępowi o 2%, stąd po zakończeniu procesu rozpakowywania na ekranie widocznych jest 50 znaków #⁸. Dzięki zastosowaniu tej opcji masz możliwość sprawdzenia, czy instalacja posuwa się do przodu, czy też cały proces zawiesił się.
- ♦ -percent — wyświetla na ekranie informacje o postępie całego procesu instalacji, wyrażone w procentach⁹.

Przed rozpoczęciem procesu instalacji *rpm* sprawdza, czy podczas instalacji nie zostaną nadpisane istniejące pliki o nowszej niż instalowana wersja oraz czy wszystkie inne pakiety niezbędne do poprawnej pracy pakietu instalowanego są już zainstalowane. Poniżej przedstawiono zestawienie opcji, które mogą być użyte do wymuszenia pracy instalatora mimo zaistnienia warunków, które w normalnej sytuacji powodują przerwanie procesu instalacji.

- ♦ --force — wymusza instalację bieżącego pakietu, niezależnie od tego, czy bieżąca wersja pakietu jest starsza niż wersja zainstalowana; czy bieżąca wersja ponownie instaluje pliki, które zostały już wcześniej zainstalowane w systemie przez inne pakiety lub bieżąca wersja pakietu jest identyczna jak wersja już zainstalowana (działanie tej opcji jest identyczne jak w przypadku łącznego zastosowania trzech innych opcji: `oldpackage`, `replacefiles` oraz `replacepkgs`). Nie jest to, co prawda, bezpieczne, ale użytkownicy często korzystają z tej opcji w celu zignorowania wszystkich problemów (np. starszy pakiet RPM), które mogą być przyczyną niepowodzeń podczas instalacji pakietu.
- ♦ --oldpackage — wymusza instalację bieżącego pakietu, nawet jeżeli wersja instalowanego pakietu jest starsza niż wersja już zainstalowana w systemie.
- ♦ --replacefiles — wymusza instalację bieżącego pakietu, nawet jeżeli instalacja tego pakietu spowoduje nadpisanie plików zainstalowanych przez inne pakiety.
- ♦ --replacepkgs — wymusza instalację bieżącego pakietu, nawet jeżeli w systemie jest już zainstalowana ta sama wersja pakietu.
- ♦ --nodeps — pomija sprawdzanie zależności pakietów i wymusza instalację pakietu, nawet jeżeli nie wszystkie inne pakiety niezbędne do poprawnej pracy pakietu

⁷ W polskiej dokumentacji (`man rpm`) przeznaczenie tej opcji zostało ujęte bardzo enigmatycznie, (czy też raczej kuriozalnie): „-vv — *Drukuję dużo danych debuggowych*” (sic!) — *przyjp. tłum.*

⁸ Przy opisie tej opcji autor polskiej dokumentacji również nie popisał się precyzją: „-h — *Drukuję 50 znaków krzyżyka, pokazując proces rozpakowywania archiwum. Używając z opcją -v, uzyskasz ładny obraz.*” (sic!). Można się tylko domyślać, że autorowi chodziło prawdopodobnie o obraz przebiegu rozpakowywania — *przyjp. tłum.*

⁹ Tutaj autor dokumentacji przeszedł samego siebie ☺: „-percent — *Drukuj procenty podczas rozpakowywania plików z archiwum. Jest to zrobione w celu ułatwienia wywoływania RPM z innych narzędzi.*” (sic!). Komentarz pozostawiam Czytelnikowi ☺ — *przyjp. tłum.*

instalowanego zostały już zainstalowane. Opcja powinna być stosowana z wyjątkową ostrożnością! Na skutek niepoprawnego rozwiązania problemu z zależnymi pakietami może dojść do uszkodzenia oprogramowania.

- ♦ `--ignorearch` — wymusza instalację pakietu, nawet jeżeli pliki binarne pakietu nie są przeznaczone dla architektury Twojego komputera.
- ♦ `--ignoreos` — wymusza instalację pakietu, nawet jeżeli pliki binarne pakietu nie są przeznaczone dla architektury Twojego systemu operacyjnego.

Poniżej przedstawiono przykładowe polecenie `rpm` użyte do zainstalowania wybranego pakietu:

```
# rpm -U audiofile-devel-0.1.6-1.i386.rpm
```

Zazwyczaj użytkownik bardzo się cieszy, jeżeli podczas instalacji na ekranie wyświetlane są informacje o jej przebiegu (domyślnie polecenie `rpm` jest zadziwiająco „małomówne” nawet po pomyślnym zakończeniu instalacji). Poniżej przedstawiono przykładowe fragmenty rezultatów działania polecenia `rpm` wywołanego z dodatkową opcją `-vv`:

```
# rpm -Uvv audiofile-devel-0.2.3-3.i386.rpm
D: counting packages to install
D: found 1 packages
D: looking for packages to download
D: retrieved 0 packages
D: finding source and binary packages
D: New Header signature
.
D: opening database in //var/lib/RPM/
D: installing binary packages
Installing audiofile-devel-0.2.3-3.i386.rpm
.
D: package: audiofile-devel-0.2.3-3 files test = 0
D: running preinstall script (if any)
D: running postinstall script (if any)
```

Przedstawione powyżej wyniki działania sugerują, że polecenie `rpm` odnalazło jeden pakiet oprogramowania, sprawdziło sygnaturę pakietu, utworzyło bazę RPM, zainstalowało pakiet i po zakończeniu operacji zamknęło bazę. Inną metodą na kontrolę postępu procesu instalacji jest dołożenie opcji `-h`, tak jak to zostało przedstawione na poniższym przykładzie:

```
# RPM -Uvh audiofile-devel-0.2.3-3.i386.rpm
audiofile-devel #####
```

Dołączenie opcji `-h` powoduje odwzorowanie na ekranie postępu procesu rozpakowywania archiwum za pomocą drukowania na ekranie kolejnych znaków `#` (jak już wspomniano wcześniej, pojedynczy znak `#` odpowiada postępowi o 2%, stąd po zakończeniu procesu rozpakowywania na ekranie widzimy 50 znaków `#`). Jak łatwo zauważyć, jeżeli wszystko idzie bez żadnych przeszkód, to proces instalowania pakietów oprogramowania przy użyciu polecenia `RPM` wydaje się całkiem nieskomplikowany. Zawsze jednak należy się liczyć z tym, że podczas instalacji mogą wystąpić jakieś problemy. Poniżej omówiono dwa wybrane scenariusze błędów.

- ♦ **Błędy zależności pakietów** — jeżeli pakiet, który właśnie jest instalowany, wymaga do poprawnej pracy obecności innych, zainstalowanych pakietów, to na ekranie pojawi się błąd informujący o brakujących pakietach. Przed ponowną próbą zainstalowania danego pakietu należy zainstalować wszystkie inne brakujące pakiety (jak zapewne pamiętasz, istnieje możliwość wymuszenia instalacji bez sprawdzania zależności pakietów, ale nie jest to polecane rozwiązanie — zazwyczaj bez obecności wszystkich niezbędnych pakietów pomocniczych instalowany pakiet nie będzie działał poprawnie).
- ♦ **Błąd braku uprawnień użytkownika root** — jeżeli polecenie `rpm -i` zostanie wydane przez użytkownika innego niż `root`, to jego działanie zakończy się niepowodzeniem — na ekranie pojawi się informacja, że baza danych `/var/lib/rpm` nie może zostać otwarta. W takiej sytuacji powinienes zalogować się jako użytkownik `root` i ponownie spróbować uruchomić instalację pakietu.

Aktualizacja pakietów oprogramowania z wykorzystaniem polecenia rpm

Tryb aktualizacji pakietów polecenia `rpm` (opcja `-U`) działa bardzo podobnie do trybu instalacji. Składnia polecenia jest taka sama jak powyżej:

```
# rpm -U [opcje] pakiet
```



Chociaż jest dostępna oddzielna opcja instalacyjna (`-i`), to jednak niezależnie od tego, czy jest przeprowadzana nowa instalacja, czy aktualizacja, osobiście namawiam do użycia opcji `-U`. W przypadku tej opcji pakiet jest zawsze instalowany. Z tego też powodu polecenie `rpm -U` zawsze zadziała (poza jednym wyjątkiem), natomiast polecenie `rpm -i` będzie działać tylko wtedy, gdy pakietu jeszcze nie zainstalowano.

Wyjątek ma miejsce, gdy są instalowane pakiety jądra. Podczas instalowania nowego jądra należy użyć opcji `-i`, ponieważ w przeciwnym razie poprzednia wersja jądra (raczej działająca) zostanie usunięta i w efekcie nie będzie możliwe uruchomienie systemu!

Osobnym problemem może być instalowanie starszej wersji danego pakietu. Jeżeli np. zainstalowałeś nową wersję pakietu `abc` (np. `abc-2.1.4-3.i386.rpm`) i okazuje się, że nie działa ona zbyt dobrze, to być może będziesz chciał powrócić do poprzedniej wersji (np. `abc-2.0.9-3.i386.rpm`). Aby tego dokonać, powinienes wykonać polecenie `rpm` z dołączoną opcją `--oldpackage`:

```
# rpm -U --oldpackage abc-2.0.9-3.i386.rpm
```

Jeżeli pakiet o takiej nazwie jest już zainstalowany, to istniejąca wersja zostanie najpierw usunięta, a na jej miejsce zostanie zainstalowana starsza wersja pakietu.

Odświeżanie pakietów z wykorzystaniem polecenia rpm

Opcja odświeżania `-F` jest podobna do opcji uaktualniania `-U`. Podstawowa różnica występująca między obiema opcjami uwidacznia się, gdy odświeżany lub uaktualniany pakiet RPM nie został wcześniej zainstalowany w systemie Fedora Linux. Opcja `-U` umożliwia przeprowadzenie nowej instalacji lub aktualizacji, natomiast opcja `-F` pozwala tylko na uaktualnienie. A zatem, jeśli pakiet nie został wcześniej zainstalowany, po wykonaniu polecenia `rpm -F` nie zostanie wykonana żadna operacja.

Odszukiwanie bardzo się przydaje, gdy dysponujesz katalogiem pełnym aktualnych wersji pakietów RPM, które chcesz zainstalować w systemie. Jednak w tym przypadku uaktualnione zostaną tylko te pakiety, które już wcześniej zainstalowano w systemie. Inaczej mówiąc, w katalogu może znajdować się wiele niepotrzebnych pakietów RPM. Odszukiwanie umożliwia uaktualnienie jedynie już posiadanych pakietów.

Załóżmy, że pobrałeś cały katalog pakietów RPM i zależy Ci na odświeżeniu tylko już zainstalowanych. Po uaktywnieniu katalogu z pakietami RPM należy wykonać następujące polecenie:

```
# rpm -Fhv *.rpm
```

Już zainstalowane pakiety RPM zostaną uaktualnione przez nowsze wersje. Wszystkie pozostałe pakiety będą pominięte.



Jeszcze raz należy zauważyć, że nie powinno się odświeżać lub uaktualniać pakietów jądra, ponieważ może dojść do tego, że podczas instalowania nowej wersji jądra zostanie usunięta jedyna działająca starsza wersja.

Usuwanie pakietów z wykorzystaniem polecenia rpm

Jeżeli już dłużej nie zamierzasz korzystać z określonego pakietu oprogramowania (lub po prostu chcesz odzyskać trochę miejsca na dysku), to powinieneś wykorzystać polecenie `rpm` z opcją `-e`. Składnia polecenia jest następująca:

```
# rpm -e pakiet
```

Jeżeli dany pakiet nie jest powiązany z innymi pakietami, to jest po prostu „po cichutku” usuwany. Zanim jednak usuniesz dany pakiet, warto się przekonać, czy rzeczywiście nie posiada on żadnych powiązań z innymi pakietami. Do odpytania pakietu służy polecenie `rpm` z opcją `-q` i całą masą dodatkowych opcji (użytkownik nie musi dokonywać odrębnej kontroli zależności z innymi pakietami, ponieważ zanim `rpm` rozpocznie proces usuwania pakietu aplikacji, automatycznie sprawdza zależności — od czasu do czasu warto jednak wykonać takie polecenie dla zaspokojenia własnej ciekawości). Aby sprawdzić zależności od innych pakietów, wykonaj następujące polecenie:

```
# rpm -q --whatrequires pakiet
```

Jeżeli zdecydowałeś się na usunięcie wybranego pakietu, to zalecaną metodą postępowania jest dołączenie opcji `-vv` do polecenia `rpm -e`, co pozwoli na bieżąco kontrolować proces usuwania pakietu. Ponieważ ilość informacji wyświetlanych po zastosowaniu opcji `-vv` może być bardzo duża, to należy rozważyć możliwość przekierowania wyników działania do polecenia `less`, np.:

```
# rpm -evv jpilot | less
```

Wykonanie powyższego polecenia spowoduje usunięcie pakietu `jpilot`; w trakcie usuwania na ekranie wyświetlane będą strona po stronie (`less`) kolejne informacje o jego przebiegu (aby wyświetlać kolejne strony, użyj klawisza spacji).

Istnieją również inne opcje, które użyte łącznie z poleceniem `rpm -e` pozwalają na m.in. zignorowanie okoliczności uniemożliwiających usunięcie pakietu lub zapobiegają

wykonywaniu pewnych określonych czynności (np. uruchamianiu skryptów przeznaczonych do wykonania przed i po procesie anulowania instalacji). Poniżej przedstawiono kilka z tych opcji.

- ♦ `--nodeps` — powoduje usunięcie pakietu bez sprawdzania jego zależności z innymi pakietami.
- ♦ `--noscripts` — powoduje usunięcie pakietu bez uruchamiania skryptów przeznaczonych do wykonania przed i po procesie anulowania instalacji).
- ♦ `--notriggers` — powoduje usunięcie pakietu bez uruchamiania skryptów inicjowanych zdarzeniem usunięcia pakietu.

Jeżeli masz pewne obiekty przed usunięciem pakietu, to przed rzeczywistym usunięciem takiego pakietu możesz uruchomić proces anulowania instalacji w trybie testowym (`--test`). W tym trybie na ekranie pojawi się pełny przebieg „symulowanego” procesu anulowania instalacji danej aplikacji, ale w rzeczywistości nie zostanie ona naruszona (nie zapomnij o dodaniu opcji `-vv`), np.:

```
# rpm -evv --test jpilot | less
D: opening db environment /var/lib/rpm/Packages joinenv
D: opening db index      /var/lib/rpm/Packages rdonly mode=0x0
D: locked db index      /var/lib/rpm/Packages
D: opening db index      /var/lib/rpm/Name rdonly mode=0x0
D: opening db index      /var/lib/rpm/Pubkeys rdonly mode=0x0
.
.
D: closed db index      /var/lib/rpm/Name
D: closed db index      /var/lib/rpm/Packages
D: closed db environment /var/lib/rpm/Packages
```

Jeżeli w wynikach symulowanego procesu anulowania instalacji nie znajdziesz niczego podejrzanego, to możesz z czystym sumieniem ponownie wykonać to samo polecenie, tym razem bez opcji `--test` — wybrany pakiet zostanie usunięty.

Zapytywanie pakietów z wykorzystaniem polecenia rpm

Używając opcji `-q` polecenia rpm, możesz otrzymywać różne informacje dotyczące pakietów oprogramowania (np. może to być zwykła lista wszystkich zainstalowanych pakietów, ale równie dobrze możesz zażądać szczegółowych informacji o wybranym pakiecie). Podstawowa składnia polecenia jest następująca (wymagane jest dołączenie co najmniej jednej opcji):

```
# rpm -q[lista opcji]
```

Poniżej przedstawiono zestawienie kilku przydatnych opcji.

- ♦ `-qa` — wyświetla listę wszystkich zainstalowanych pakietów.
- ♦ `-qf plik` — wyświetla informacje o pakiecie, w którego skład wchodzi dany plik (jeżeli nie podasz pełnej ścieżki do pliku, to rpm zakłada, że znajduje się on w katalogu bieżącym).
- ♦ `-qi pakiet` — wyświetla szczegółowe informacje o wybranym pakiecie.

- ♦ `-qR pakiet` — wyświetla listę składników (takich jak biblioteki i polecenia), z których korzysta dany pakiet.
- ♦ `-qI pakiet` — wyświetla listę wszystkich plików wchodzących w skład danego pakietu.
- ♦ `-qd pakiet` — wyświetla listę wszystkich plików dokumentacji dołączonych do danego pakietu.
- ♦ `-qc pakiet` — wyświetla listę wszystkich plików konfiguracyjnych dołączonych do danego pakietu.
- ♦ `-qp [opcja] pakiet` — pyta pakiety, które jeszcze nie zostały zainstalowane. Przy użyciu tej opcji i innych opcji zapytywania możliwe jest zapytywanie pakietów jeszcze niezainstalowanych.

Aby wyświetlić listę wszystkich pakietów oprogramowania zainstalowanych na Twoim komputerze, powinieneś skorzystać z opcji `-qa`. Ponieważ z pewnością będzie to długa lista, to powinieneś przekierować wynik zapytania do polecenia `less` lub skorzystać z polecenia `grep` w celu wyświetlenia tylko konkretnych pakietów.

Poniższe polecenie przygotowuje pełną listę nazw wszystkich zainstalowanych pakietów RPM, a następnie wyświetla na ekranie tylko te, których nazwy zawierają ciąg znaków *xorg* (opcja `-i` polecenia `grep` powoduje, że ignorowana jest wielkość liter w nazwach).

```
# rpm -qa | grep -i xorg
```

Jeżeli jesteś zainteresowany szczegółowymi informacjami dotyczącymi wybranego pakietu, to powinieneś skorzystać z opcji `-qi`. Poniżej podano przykład polecenia wyświetlającego szczegółowe informacje na temat pakietu *dosfstools* (który umożliwia odczytywanie systemu plików DOS w systemie Linux) oraz wyniki jego działania:

```
# rpm -qi dosfstools
Name       : dosfstools                Relocations: (not relocateable)
Version    : 2.8                      Vendor: Red Hat, Inc.
Release    : 5                       Build Date: nie 13 cze 2004 02:59:57 CET
Install date: nie 08 lut 2004 04:09:31 CET
Build Host: perf90.perf.redhat.com
Group      : Applications/System Source RPM: dosfstools-2.8-12.src.rpm
Size       : 70346                    License: GPL
Signature  : DSA/SHA1, wto 07 wrz 2004 11:48:55 CET, Key ID 219180cddb42a60e
Packager   : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
Summary    : Utilities for making and checking MS-DOS FAT filesystems on Linux.
Description:
The dosfstools package includes the mkdosfs and dosfsck utilities,
which respectively make and check MS-DOS FAT filesystems on hard
drives or on floppies.
```

Jeżeli chcesz się dowiedzieć czegoś więcej na temat zawartość pakietu, to powinieneś skorzystać z opcji `-qI`. Poniżej przedstawiono fragment listy plików (wraz z ich lokalizacją) należących do pakietu *kmail*¹⁰:

¹⁰*Kmail* to domyślny program do obsługi poczty elektronicznej środowiska KDE — *przyp. tłum.*

```
# rpm -ql kmail | less
/usr/bin/kmail
/usr/bin/kmailcvt
/usr/share/applications/kde-KMail.desktop
/usr/share/applications/kde-kmailcvt.desktop
/usr/share/apps/kconf_update/kmail-pgpidentity.pl
/usr/share/apps/kconf_update/kmail.upd
/usr/share/apps/kconf_update/kpgp.upd
/usr/share/apps/kconf_update/upgrade-transport.pl
/usr/share/apps/kmail
.
.
.
```

Jeżeli chciałbyś uzyskać dodatkowe informacje na temat wybranego pakietu oprogramowania, to powinieneś skorzystać z opcji `-qd` powodującej wyświetlenie listy dokumentacji (pliki pomocy systemowej *man*, pliki *README*, *HOWTO* itd.), która jest dołączona do danego pakietu. Jeżeli np. masz jakieś problemy z prawidłowym funkcjonowaniem X Window System, to, wykonując poniższe polecenie, możesz zorientować się, gdzie szukać pomocy:

```
# rpm -qd xorg-x11 | less
/usr/X11R6/man/man1/Xorg.1x.gz
/usr/X11R6/man/man1/Xmark.1x.gz
/usr/X11R6/man/man1/Xserver.1x.gz
/usr/X11R6/man/man1/appres.1x.gz
/usr/X11R6/man/man1/atobm.1x.gz
/usr/X11R6/man/man1/bdfpcf.1x.gz
/usr/X11R6/man/man1/bitmap.1x.gz
/usr/X11R6/man/man1/bmta.1x.gz
/usr/X11R6/man/man1/dga.1x.gz
/usr/X11R6/man/man1/dpsexec.1x.gz
/usr/X11R6/man/man1/dpsinfo.1x.gz
/usr/X11R6/man/man1/editres.1x.gz
/usr/X11R6/man/man1/iceauth.1x.gz
/usr/X11R6/man/man1/lbproxy.1x.gz
/usr/X11R6/man/man1/libxrx.1x.gz
/usr/X11R6/man/man1/lndir.1x.gz
/usr/X11R6/man/man1/luit.1x.gz
/usr/X11R6/man/man1/makedepend.1x.gz
/usr/X11R6/man/man1/makepsres.1x.gz
/usr/X11R6/man/man1/makestrs.1x.gz
/usr/X11R6/man/man1/mkcfm.1x.gz
/usr/X11R6/man/man1/mkdirhier.1x.gz
/usr/X11R6/man/man1/oclock.1x.gz
/usr/X11R6/man/man1/pctweak.1x.gz
/usr/X11R6/man/man1/proxymngr.1x.gz
/usr/X11R6/man/man1/resize.1x.gz
/usr/X11R6/man/man1/revpath.1x.gz
/usr/X11R6/man/man1/rstart.1x.gz
.
.
.
```

... itd¹¹.

¹¹Pełna lista obejmuje kilkadziesiąt pozycji — *przyp. tłum.*

Wiele pakietów posiada również swoje pliki konfiguracyjne. Aby sprawdzić, jakie pliki konfiguracyjne powiązane są z określonym pakietem oprogramowania, powinieneś skorzystać z opcji `-qc`. Poniżej przedstawiono wyniki działania takiego polecenia dla pakietu `ppp`:

```
# rpm -qc ppp
/etc/pam.d/ppp
/etc/ppp/chap-secrets
/etc/ppp/options
/etc/ppp/pap-secrets
```

Jeżeli kiedykolwiek zastanawiałeś się nad tym, z którego pakietu pochodzi dane polecenie lub plik konfiguracyjny, to powinieneś skorzystać z opcji `-qf`. Jak widać na poniższym przykładzie, zastosowanie opcji `-qf` pozwoliło ustalić, że polecenie `chgrp` pochodzi z pakietu `fileutils`:

```
# rpm -qf /bin/chgrp
fileutils-4.1.4
```

Przed zainstalowaniem pakietu można go poddać zapytywaniu w takim samym stopniu jak pakiet już zainstalowany. Może to być znakomita metoda wyszukiwania informacji zawartych w pakiecie, który znajduje się w bieżącym katalogu lub nawet w zbiorze oprogramowania. Oto przykład zastosowania opcji `-qp` w połączeniu z opcją `-i` w celu uzyskania opisu pakietu znajdującego się w zbiorze oprogramowania:

```
# rpm -qp -i \
http://ayo.freshrpms.net/fedora/linux/1/i386/freshrpms/RPMS/blackbox-0.65.0\
-8.1.fc1.fr.i386.rpm
Name       : blackbox           Relocation: (not relocatable)
Version    : 0.65.0            Vendor: Freshrpms.net
Release    : 8.1.fc1.fr        Build Date: Thu 25 Mar 2004 12:45:45 PM PST
Install Date: (not installed)  Build Host: python2.freshrpms.net
.
.
.
Blackbox is a window manager for the X Window environment, which is...
```

W powyższym przykładzie długie polecenie zostało podzielone na trzy wiersze, które faktycznie powinny być zapisane w jednym. Jeśli zastanawiasz się nad zawartością i legalnością pobranego pakietu, poniższy przykład umożliwi Ci wyświetlenie opisu pakietu jeszcze przed rozpoczęciem jego pobierania.

W przykładzie polecenie generuje listę plików zawartych w pakiecie znajdującym się w bieżącym katalogu.

```
# rpm -qp -l RealPlayer-8.0-1.i386.rpm
```

Jeszcze raz powtórzę, że jest to znakomity sposób na stwierdzenie, co znajduje się w pakiecie jeszcze przed rozpoczęciem jego pobierania.

Weryfikacja integralności pakietów z wykorzystaniem polecenia rpm

Jeżeli jakiś element składowy pakietu oprogramowania nie działa zgodnie z oczekiwaniami lub podejrzewasz, że Twój system został w jakiś sposób zmieniony, to posłużenie się opcją `-V` pozwoli Ci porównać zainstalowane oprogramowanie z oryginalnym pakietem,

z którego zostało zainstalowane. Informacje dotyczące wszystkich zainstalowanych na Twoim komputerze pakietów oprogramowania są przechowywane w lokalnej bazie danych RPM. Korzystając z trybu weryfikacji, możesz sprawdzić, czy poszczególne składniki pakietów były w jakiś sposób modyfikowane.



Opcji weryfikacji jest przypisana duża litera V (-V), natomiast opcji wyświetlającej szczegółowe informacje — mała litera v (-v).

Podczas operacji weryfikacji pakietu wykonywany jest cały szereg testów sprawdzających rozmiary poszczególnych plików, uprawnienia i wiele innych. Jeżeli wszystko jest w porządku, to wykonanie polecenia weryfikacji nie powoduje wyświetlenia żadnych komunikatów. Jeśli jednak jakieś elementy zostały w dowolny sposób zmienione, to zostaną one wypisane na ekranie łącznie z informacjami wskazującymi na naturę modyfikacji np.:

```
# rpm -V ppp
S.5...T c /etc/ppp/chap-secrets
S.5...T c /etc/ppp/options
S.5...T c /etc/ppp/pap-secrets
```

Przedstawione powyżej wyniki działania polecenia `rpm -V` informują, że w pakiecie `ppp` (wykorzystywany do realizacji połączeń typu *dial-up* z sieciami TCP/IP, np. z siecią internetową) od czasu ostatniej instalacji zostały zmodyfikowane trzy pliki. Tajemnicze symbole na początku każdego wiersza oznaczają, że dla każdego z tych plików zmienił się rozmiar (S), suma kontrolna MD5 (5) oraz czas ostatniej modyfikacji (T). Litera `c` oznacza, że są to pliki konfiguracyjne. Po dokładnym sprawdzeniu zawartości tych plików i przekonaniu się, że ich modyfikacje nastąpiły np. na skutek rekonfiguracji połączenia PPP, użytkownik może się upewnić, iż integralność została pomyślnie zweryfikowana.

Poszczególne znaczniki, które mogą się pojawiać na wydrukach podczas weryfikacji pakietów, oznaczają kolejno:

- ♦ 5 — (**MD5 checksum**), suma kontrolna MD5 — zmiany sumy kontrolnej sugerują, że zawartość danego pliku uległa zmianie,
- ♦ S — (**File size**) — oznacza, że rozmiar pliku uległ zmianie,
- ♦ L — (**SymLink**) — oznacza, że plik stał się odnośnikiem do innego pliku,
- ♦ T — (**mTime**) — oznacza, że zmienił się czas ostatniej modyfikacji pliku,
- ♦ D — (**Device**) — oznacza, że plik stał się plikiem reprezentującym określone urządzenie,
- ♦ U — (**User**) — oznacza, że właścicielem pliku stał się inny użytkownik,
- ♦ G — (**Group**) — oznacza, że zmieniła się grupa użytkowników przypisana do tego pliku,
- ♦ M — (**Mode**) — oznacza, że zmienione zostały prawa dostępu do tego pliku.



Istnieje narzędzie umożliwiające przeglądanie zawartości pakietów RPM z poziomu systemu Windows. `rpmbrowser.exe`, bo o nim tu właśnie mowa, pozwala nie tylko na przeglądanie zawartości, ale również na rozpakowywanie plików z pakietów RPM. Program `rpmbrowser.exe` możesz pobrać z witryny <http://www.winsite.com> (przejdź na stronę <http://www.winsite.com/search> i wpisz do wyszukiwarki słowo `rpmbrowser`).

Tworzenie oraz instalacja aplikacji z kodu źródłowego

Jeżeli potrzebny pakiet oprogramowania nie jest dostępny w postaci binarnej lub po prostu chcesz dokładnie dopasować daną aplikację do swoich potrzeb, to zawsze masz możliwość zainstalowania pakietu przy użyciu jego kodu źródłowego. Kody źródłowe programów, zamieszczone na instalacyjnych dyskach CD dystrybucji Fedora Core, są dokładnym ekwiwalentem (w formacie *SRPMS*) pakietów binarnych, z których korzystałeś podczas instalacji systemu Fedora Linux. Użytkownik może w dowolny sposób modyfikować i rekompilować kody źródłowe, tak aby poszczególne aplikacje jak najlepiej spełniały jego wymagania.



Kod źródłowy jądra systemu Linux Fedora znajduje się na dysku CD nr 3 dołączonym do książki.

Pakiety oprogramowania, niedostępne w formacie RPM, są zazwyczaj dostarczane w postaci „paczki” narzędzia *tar* (grupa plików umieszczonych w jednym pliku przetworzonym przez program *tar*) dodatkowo poddanej kompresji (zwykle przy użyciu programu *gzip*). Mimo że dokładne procedury postępowania przy tworzeniu i instalacji aplikacji z kodu źródłowego z różnych archiwów mogą się od siebie różnić, to jednak większość pakietów zapisanych w formatach *.tgz*, *.gz* oraz *.tar* będzie podlegała takim samym metodom postępowania.



Zanim będziesz mógł tworzyć aplikacje z kodu źródłowego, musisz zainstalować cały szereg pakietów niezbędnych do prawidłowej kompilacji i konsolidacji źródeł programów. Jeżeli dysponujesz odpowiednią ilością miejsca na dysku, to zalecanym sposobem postępowania jest zainstalowanie wszystkich pakietów rekomendowanych do instalacji w kategorii *Development tools* (podczas instalacji systemu Fedora Linux).

Poniżej przedstawiono minimalną listę narzędzi programistycznych języka C niezbędnych do samodzielnego tworzenia aplikacji z kodu źródłowego.

- ♦ *gcc* — kompilator języka C (*GNU C compiler*).
- ♦ *make* — narzędzie do tworzenia pakietów oprogramowania na podstawie informacji zawartych w plikach *Makefile*.
- ♦ *glibc* — pakiet podstawowych bibliotek systemowych, bibliotek języka C oraz standardowych bibliotek matematycznych.
- ♦ *glibc-devel* — pakiet standardowych plików nagłówek niezbędnych do tworzenia plików wykonywalnych.
- ♦ *binutils* — pakiet narzędzi niezbędnych do kompilacji programów, takich jak assembler i konsolidator (linker).
- ♦ *kernel-source* — pakiet zawierający kod źródłowy jądra systemu Linux, niezbędny do rekompilacji jądra.
- ♦ *libc* — pakiet bibliotek niezbędnych dla programów opartych na *libc5*, umożliwia pracę starszych aplikacji na systemach wykorzystujących *glibc* (*libc6*).

Instalacja oprogramowania dostępnego w formacie SRPMS

Aby zainstalować wybrany pakiet oprogramowania w formacie SRPMS z instalacyjnego CD-ROM-u nr 3 dystrybucji Fedora Core (zawierającego kody źródłowe programów), powinieneś postępować zgodnie z poniższą procedurą.

1. Włóż CD-ROM nr 3 dystrybucji systemu Fedora Core załączonej do niniejszej książki (bądź dowolny inny dysk CD zawierający źródła SRPMS) do napędu CD-ROM. Dysk powinien zostać automatycznie zamontowany (jeżeli tak się nie stanie, po zalogowaniu się jako użytkownik *root* wykonaj z poziomu konsoli polecenie `mount /mnt/cdrom`).

2. Zmień katalog bieżący na dysk CD, przykładowo:

```
# cd /mnt/cdrom/SRPMS
```

3. Wybierz pakiet, który chcesz zainstalować (aby przejrzeć listę pakietów, wykorzystaj polecenie `ls`), a następnie zainstaluj go, wykonując poniższe polecenie:

```
# rpm -iv pakiet*.src.rpm
```

(zamień ciąg znaków *pakiet** na nazwę pakietu, który chcesz zainstalować). Źródła pakietu zostaną zainstalowane w drzewie źródeł systemu Fedora Core (*/usr/src/redhat*). Pliki *SPEC* są kopiowane do katalogu */usr/src/redhat/SPECS*.

4. Zmień katalog bieżący na katalog *SPEC*:

```
# cd /usr/src/redhat/SPECS
```

5. Rozpakuj źródła za pomocą polecenia:

```
# rpm -bp pakiet*.spec
```

6. Kod źródłowy pakietu zostanie umieszczony w katalogu */usr/src/redhat/BUILD/pakiet*, gdzie *pakiet* jest nazwą instalowanego pakietu oprogramowania.

7. Możesz teraz przystąpić do modyfikowania (w miarę potrzeb) plików w katalogu *BUILD*. Więcej informacji na temat tworzenia poszczególnych pakietów znajdziesz w plikach *README*, *Makefile* i innych plikach dokumentacji danego pakietu.

Instalacja oprogramowania dostępnego w formacie tar.gz lub tar.bz2

Poniżej przedstawiono ogólne metody postępowania podczas instalacji kodów pakietów oprogramowania dostępnych w formacie *gzip* lub *tar*.

1. Pobierz pakiet kodu źródłowego danej aplikacji z internetu bądź dystrybucyjnego dysku CD i skopiuj go do pustego katalogu (zalecany sposób postępowania jest nadanie takiemu katalogowi nazwy jednoznacznie identyfikującej dany pakiet).
2. Zakładając, że plik został spakowany programem *gzip*, rozpakuj go, wykonując następujące polecenie:

```
# gunzip -d pakiet.tar.gz
```

W wyniku działania tego polecenia pakiet zostanie rozpakowany i przyrostek *.gz* zostanie usunięty z jego nazwy (np. zostanie tylko *pakiet.tar*).

3. Dokonaj rozpakowania otrzymanego archiwum *tar*, wykonując polecenie:

```
# tar xvf pakiet.tar
```

W wyniku działania tego polecenia poszczególne pliki zostaną wypakowane z archiwum *tar* i umieszczone w nowym podkatalogu katalogu bieżącego (przy użyciu polecenia *tar xvfz pakiet.tar.gz* możliwe jest jednoczesne wykonanie kroków 2. i 3.).

4. Zmień bieżący katalog na podkatalog (zwykle o nazwie pakietu) utworzony w punkcie 3.:

```
# cd nowykatalog
```

5. Poszukaj pliku o nazwie *INSTALL* lub *README*. W jednym z tych plików powinienś znaleźć instrukcje na temat sposobu przeprowadzenia instalacji. Ogólnie rzecz biorąc, do instalacji pakietu używane będzie polecenie *make*. Sprawdź zawartość katalogu aplikacji w poszukiwaniu następujących plików:

Makefile.in — jeżeli znajdziesz ten plik, to spróbuj wykonać następujące polecenia:

```
# ./configure -prefix=/usr/local
# make all
```

Imake — jeżeli znajdziesz ten plik, to spróbuj wykonać następujące polecenia:

```
# xmkmf -a
# make all
```

Makefile — jeżeli znajdziesz ten plik, to spróbuj wykonać następujące polecenia:

```
# make all
```

Po zakończeniu tworzenia programu oraz jego zainstalowaniu niezbędne może się okazać przeprowadzenie dodatkowej konfiguracji. Więcej informacji na ten temat powinienś znaleźć w plikach *HOWTO* lub plikach pomocy systemowej (*man*), które są dostarczane razem z danym pakietem.



Nawet jeżeli nie jesteś programistą, to analiza kodu źródłowego często może dać Ci niejaki pojęcie na temat tego, jak działa dany program.

Aby wypróbować podaną powyżej procedurę postępowania, pobrałem z internetu pakiet *whichman*, który zawiera narzędzia pozwalające na odszukiwanie odpowiednich stron pomocy systemowej (*man*) na podstawie podanych słów kluczowych. Pobrany plik, o nazwie *whichman-2.2.tar.gz*, umieszczony został w katalogu, który nazwałem */usr/sw/which*. Następnie skorzystałem z poleceń *gzip* oraz *tar*, podając jako ich argumenty odpowiednio pliki *whichman-2.2.tar.gz* oraz *whichman-2.2.tar*.

Zmieniłem bieżący katalog poleceniem *cd /usr/sw/which/whichman-1.5*, a następnie wyświetliłem na ekranie jego zawartość. Znalazłem i zapoznałem się z treścią pliku *README*, a następnie, zgodnie z sugestiami tam zawartymi, wpisałem kolejno polecenia *make* oraz *make install*. W rezultacie tych operacji nowe polecenia, *whichman*, *ftwhich* oraz *ftff*,

zostały zainstalowane w katalogu `/usr/bin` (w tym momencie mam już możliwość sprawdzenia stron pomocy systemowej i przekonania się, do czego służą wymienione polecenia).

Ostatnią sprawą, o której wspominał plik *README*, była konieczność dokonania niewielkiej konfiguracji. Do pliku *\$HOME/.bashrc* dodałem nową zmienną, *MANPATH*, która przechowuje lokalizację stron pomocy systemowej na moim komputerze — dzięki temu polecenie *whichman* będzie „wiedziało”, gdzie należy ich poszukiwać. Do pliku *.bashrc* dodałem następujący wiersz:

```
export MANPATH=/usr/share/man:/usr/X11R6/man:/usr/share/doc/samba-2.2.3a/docs
```

Jeżeli interesuje Cię przeznaczenie poleceń *whichman*, *ftwhich* oraz *ftff*, to mogę powiedzieć, że ich zadaniem jest przeszukiwanie stron pomocy systemowej pod kątem występowania określonych słów kluczowych. Za ich pomocą można odpowiednio odszukiwać miejsca występowania określonych słów kluczowych, miejsca występowania słów zbliżonych do podanych słów kluczowych lub też (polecenie *ftff*) strony pomocy systemowej (*man*) zlokalizowane poniżej określonego miejsca w strukturze plików.

Uruchamianie aplikacji dla X Window System

Konfiguracja i dopasowywanie X Window System do Twoich potrzeb jest dosyć uciążliwym zajęciem. Dla porównania uruchamianie aplikacji opartych na X jest stosunkowo proste. Jeżeli kiedykolwiek używałeś systemu Microsoft Windows, to w zasadzie powinienś już doskonale znać większość sposobów uruchamiania aplikacji w środowisku graficznym. Powinieneś jednak zdawać sobie sprawę z tego, że X Window System jest znacznie bardziej elastycznym środowiskiem do uruchamiania aplikacji systemu Linux.



Więcej informacji na temat konfiguracji środowiska X Window znajdziesz w rozdziale 3.

Uruchamianie aplikacji z menu systemowego

W celu ułatwienia uruchamiania wybranych aplikacji większość menedżerów okien korzystających z X udostępnia użytkownikowi system menu podobny do menu *Start* znanego z systemu Microsoft Windows. Poszczególne aplikacje są zazwyczaj pogrupowane w szereg kategorii. Z menu głównego środowiska graficznego GNOME lub KDE systemu Fedora Core należy wybrać kategorię, a następnie aplikację, która ma zostać uruchomiona. Na rysunku 5.1 pokazano przykładowe menu główne systemu Fedora Core i podmenu *Multimedia*.

Uruchamianie aplikacji z okna Uruchom program

Nie wszystkie aplikacje są dostępne przez menu menedżera okien. Aby zapewnić możliwość uruchamiania również innych aplikacji, większość menedżerów okien udostępnia specjalne okno, podobne do okna *Uruchom program* znanego ze środowiska GNOME, gdzie możesz podać nazwę programu, który chcesz uruchomić.

Rysunek 5.1.

*Uruchamianie aplikacji
z menu głównego
systemu Fedora Linux*



Aby otworzyć okno *Uruchom program*, powinieneś wykonać poniższe polecenia.

1. Otwórz główne menu środowiska graficznego systemu Fedora Linux.
2. Wybierz pozycję *Uruchom program*. Zostanie wyświetlone okno *Uruchom program*.
3. Kliknij opcję *Lista znanych aplikacji* odszukaj i zaznacz wybrany program, a następnie naciśnij przycisk *Uruchom*.

Jeżeli poszukiwanej aplikacji nie ma na liście, możesz bezpośrednio wpisać polecenie (łącznie ze wszystkimi niezbędnymi opcjami i argumentami), które uruchomi program, bądź nacisnąć przycisk *Dołącz plik*. Po wybraniu tej opcji na ekranie pojawi się okno dialogowe *Wybór pliku*, za pomocą którego będziesz mógł odszukać żądane polecenie. Jeżeli chcesz użyć programu, który musi być uruchomiony w oknie terminala (taki jak np. edytor *vi*), to przed naciśnięciem przycisku *Uruchom* powinieneś zaznaczyć opcję *Uruchomienie w terminalu*. Na rysunku 5.2 przedstawiono przykładowe okno *Uruchom program*.

Rysunek 5.2.

Z listy Lista znanych aplikacji możesz szybko wybrać program, który chcesz uruchomić



Uruchamianie aplikacji z okna terminala

Bardzo często zdarza mi się uruchamiać różne aplikacje X z poziomu okna terminala (a przynajmniej robię tak za pierwszym razem). Istnieje kilka istotnych powodów, dla których wolę uruchamiać aplikacje z okna terminala bądź okna *Uruchom program*.

- ♦ Jeżeli podczas uruchamiania wystąpi jakiś problem, to zwykle w oknie terminala otrzymuję odpowiedni komunikat o błędzie. Jeżeli w analogicznej sytuacji aplikacja była uruchamiana z menu systemowego, to zwykle żadne błędy nie są wyświetlane.
- ♦ Aplikacje wybierane z menu są uruchamiane z predefiniowanym zestawem opcji. Jeżeli chciałbyś zmienić taki zestaw opcji, to musisz modyfikować odpowiedni plik konfiguracyjny menu (a przynajmniej właściwości wybranej pozycji menu).
- ♦ Jeżeli chcesz wypróbować działanie aplikacji z różnym zestawem opcji, to właśnie okno terminala umożliwia Ci łatwe uruchamianie, zatrzymywanie i modyfikację opcji danej aplikacji.

Gdy już przetestowałeś daną aplikację i dysponujesz odpowiednio dobranym, optymalnym zestawem opcji, to możesz pokusić się o umieszczenie aktywatora takiej aplikacji w menu lub na panelu (jeśli menadżer okien, którego używasz, oferuje takie możliwości). Dzięki takiemu rozwiązaniu będziesz mógł otworzyć aplikację dokładnie w taki sposób, jaki Ci odpowiada, zamiast zdawać się na domyślny sposób uruchamiania jej predefiniowany w menu.

Poniżej przedstawiono procedurę uruchamiania wybranej aplikacji z okna terminala.

1. Otwórz okno terminala na pulpicie (możesz tego dokonać, korzystając z ikony terminala na panelu bądź wybierając z menu głównego polecenie *Terminal*¹²).
2. Po otwarciu okna terminala na ekranie wpisz polecenie:

```
$ echo $DISPLAY
```

Rezultat działania tego polecenia powinien być mniej więcej taki:

```
:0.0
```

Oznacza to, że domyślnie okno terminala będzie kierowało wszystkie uruchamiane z niego aplikacje X na ekran (konsolę) 0 Twojego systemu (jeżeli polecenie `echo $DISPLAY` nie zwróci żadnej wartości, to aby ustawić numer konsoli, wpisz polecenie `export DISPLAY=:0.0`).

3. Następnie wpisz polecenie:

```
$ xmms &
```

Na ekranie powinno się pojawić okno programu *xmms*¹³. Powinieneś zwrócić uwagę na następujące fakty.

- ♦ Polecenie `xmms` działa w tle okna terminala (zostało wywołane z parametrem `&`). Oznacza to, że w czasie jego działania nadal możesz korzystać z okna terminala.

¹²Polecenie *Terminal* zazwyczaj jest umiejscowione w kategorii *System Tools* — przyp. tłum.

¹³Jest to popularny odtwarzacz audio przeznaczony dla środowiska X Window — przyp. tłum.

- ♦ Uruchamianie *xmms* w opisany powyżej sposób nie powinno sprawiać żadnych problemów. Z drugiej jednak strony, z pewnością znajdą się aplikacje, po uruchomieniu których w oknie terminala będą wyświetlane różne komunikaty. Mogą to być np. informacje, że aplikacja nie może odnaleźć określonych danych lub konkretna czcionka nie może zostać wyświetlona. Jeżeli uruchamiasz aplikację z menu systemowego, to takie informacje mogłyby zostać utracone.
- 4. Jeżeli chcesz się dowiedzieć, jakie opcje dodatkowe akceptuje program *xmms*, wykonaj następujące polecenie:

```
$ xmms --help
```
- 5. Wypróbuj kilka opcji. Jeżeli np. chcesz uruchomić odtwarzanie pliku *abc.wav*, to powinieneś wykonać następujące polecenie:

```
$ xmms abc.wav
```
- 6. Kiedy zechcesz zakończyć pracę aplikacji *xmms* i zamknąć jej okno, to możesz tego dokonać, klikając prawym klawiszem myszy okno *xmms* i wybierając z menu podręcznego polecenie *Zakończ* bądź *zabijając*¹⁴ odpowiedni proces w oknie terminala. Aby tego dokonać, wpisz polecenie *jobs* i sprawdź, jaki jest numer procesu *xmms*. Jeżeli numerem tego procesu jest np. 2, to aby go zabić, wpisz polecenie *kill %2*.

Powinieneś wypróbować działanie innych poleceń X. Wśród kilku starszych poleceń X, które można wypróbować, są polecenia *xeyes* i *xcalc*.

Uruchamianie zdalnych aplikacji X

Środowisko X pozwala na uruchamianie wybranej aplikacji na danym komputerze z dowolnego innego komputera podłączonego do sieci i wyświetlanie rezultatów jej działania (i ekranu) na Twojej konsoli X. Zamiast podlegać ograniczeniom rozmiaru swojego dysku twardego, mocy obliczeniowej procesora oraz pamięci RAM, możesz skorzystać z zasobów dowolnego komputera, do którego masz odpowiednie uprawnienia.

Pomyśl, jakie to daje możliwości. W razie potrzeby możesz pracować, wykorzystując aplikacje uruchamiane na dowolnym innym komputerze pracującym ze środowiskiem X — począwszy od małego komputera PC, a na ogromnym superkomputerze skończywszy. Mając odpowiednie uprawnienia, możesz pracować z plikami, wykorzystywać drukarki, urządzenia do tworzenia kopii bezpieczeństwa danych, dyski wymienne oraz dowolne inne zasoby zdalnego komputera, tak jakby były one dostępne bezpośrednio na Twoim komputerze.

Należy jednak pamiętać o tym, że za takimi możliwościami idzie również odpowiedzialność. Korzystając z takiego rozwiązania, powinieneś starać się maksymalnie zabezpieczyć dostęp do swojej konsoli, zwłaszcza w sieciach, gdzie znajdują się komputery użytkowników, których nie znasz lub którym nie możesz zaufać. Z pewnością nie chciałbyś, aby któryś z użytkowników wyświetlił na Twojej konsoli ekran logowania, zachęcając Cię do niezamierzonego ujawnienia niepowołanym osobom nazwy Twojego użytkownika i hasła dostępu.

¹⁴To określenie pochodzi od nazwy polecenia *kill* służącego m.in. do zakańczania działania uruchomionych procesów — *przyp. tłum.*

W najprostszym przypadku, aby uruchomić zdalną aplikację X, wystarczy wiedzieć, jak zidentyfikować odległą konsolę X oraz w jaki sposób używać środków bezpieczeństwa zaimplementowanych do ochrony Twoich zasobów sieciowych. Zastosowanie ssh do uruchamiania aplikacji X jest jeszcze łatwiejsze i bardziej bezpieczne niż metoda tradycyjna. W kolejnych podrozdziałach omówiono te zagadnienia bardziej szczegółowo.

Tradycyjna metoda uruchamiania zdalnych aplikacji X

Jeżeli na innym komputerze zlokalizowanym w Twojej sieci znajduje się aplikacja, której chcesz użyć na swoim pulpicie, to powinieneś wykonać następujące polecenia.

- ♦ Utwórz odpowiednie uprawnienia dla swojego serwera X, tak aby zdalna aplikacja mogła wykorzystać Twoją konsolę.
- ♦ Udostępnij Twoją konsolę X dla uruchamianej aplikacji.

Kiedy uruchamiasz klienta X na swoim komputerze, to zazwyczaj lokalna konsola jest identyfikowana jako :0. Ciąg :0 reprezentuje pierwszą konsolę Twojego systemu lokalnego. Aby jednak identyfikacja konsoli była dostępna dla zdalnego komputera, to musisz do niej dodać nazwę swojego komputera (hosta). Jeżeli np. Twój komputer nazywa się *antares*, to identyfikacja konsoli będzie wyglądała następująco:

```
antares:0
```



W większości przypadków nazwą hosta jest po prostu jego adres IP. Nazwy komputerów podłączonych do Twojej sieci lokalnej mogą być umieszczone w lokalnym pliku */etc/hosts*, albo określane za pomocą usługi DNS (*Domain Name System*). Istnieje również możliwość zastosowania pełnej nazwy domenowej, np. *hatbox.handsonhistory.com*. X obsługuje również inne protokoły, nie tylko TCP/IP, choć w praktyce inne protokoły nie są już wykorzystywane.

Prawdopodobnie w większości przypadków podczas uruchamiania zdalnych aplikacji X będziesz korzystał ze sposobu identyfikacji konsoli opisanego powyżej, to jednak w niektórych sytuacjach może to wyglądać nieco inaczej. Jeżeli Twój komputer posiada kilka osobnych konsoli (zestawów klawiatura, mysz, monitor), to numery konsoli będą się również zmieniać (oprócz konsoli :0 dostępne będą również :1, :2 itd.). Istnieje również możliwość, że za pomocą jednego zestawu klawiatura-mysz sterujesz więcej niż jednym monitorem — w takiej sytuacji do identyfikacji konsoli powinieneś dodać numer monitora, np.:

```
antares:0.1
```

Powyższy adres identyfikuje drugi monitor (.1) pierwszej konsoli (:0) komputera *antares*. Pierwszy monitor konsoli jest identyfikowany jako .0 (co jest wartością domyślną, ponieważ większość konsoli dysponuje tylko jednym monitorem). Jeżeli nie korzystasz z systemu wielomonitorowego, możesz swobodnie pominąć identyfikację monitora.

Istnieją dwa sposoby identyfikacji nazwy konsoli dla zdalnej aplikacji X.

- ♦ **Zmienna środowiskowa DISPLAY** — możesz ustawić jej wartość na ciąg znaków identyfikujący nazwę Twojego systemu oraz numer konsoli. Po zakończeniu rezultaty działania dowolnej aplikacji X uruchomionej z danej powłoki będą się pojawiały na konsoli określonej wartością tej zmiennej. Aby np. ustawić wartość zmiennej

DISPLAY, tak aby wskazywała na pierwszą konsolę komputera *antares*, użyj następującego polecenia:

```
export DISPLAY=antares:0
```

lub

```
setenv DISPLAY antares:0
```

Pierwszy przykład odnosi się do ustawiania wartości zmiennej dla powłoki *bash* lub *ksh*, przykład drugi ma zastosowanie dla powłoki *csh*.

- ♦ **Opcja** `-display` — kolejnym sposobem identyfikacji konsoli dla zdalnej aplikacji X jest dołączenie opcji `-display` do polecenia uruchamiającego taką aplikację. Wartość opcji `-display` nadpisuje ustawienia zmiennej `DISPLAY`. Jeżeli chcesz, aby okno terminala uruchamiane poleceniem `xterm` na zdalnym komputerze wyświetlało się na pierwszej konsoli komputera *antares*, to powinieneś wykonać następującą komendę:

```
xterm -display antares:0
```

Biorąc pod uwagę podane powyżej zalecenia, powinieneś już uruchomić dowolną aplikację X z wybranego, zdalnego komputera, do którego masz dostęp. W kolejnych podrozdziałach dowiesz się, jak wykorzystać te informacje w praktyce.

Zdalne uruchamianie aplikacji X

Załóżmy, że chcesz uruchomić wybraną aplikację X z komputera o nazwie *remote1*, zlokalizowanego w Twojej sieci lokalnej (i w tej samej domenie) oraz załóżmy, że Twój komputer nazywa się *antares*. Aby uruchomić taką aplikację z komputera *remote1*, tak aby jej okno pojawiło się na konsoli Twojego komputera (*antares*), powinieneś wykonać następujące polecenia.



Opisana poniżej procedura zakłada, że nie zostały zainstalowane żadne dodatkowe mechanizmy zabezpieczające zasoby tych komputerów. Jest to rozwiązanie domyślne i przeznaczone do współużytkowania aplikacji między zaufanymi komputerami (zwykle jednorazowymi stacjami roboczymi) w sieci lokalnej. Opisana metoda z definicji nie zapewnia żadnego bezpieczeństwa i zakłada, że masz pełne zaufanie do użytkowników komputerów, do których masz dostęp. Jeżeli wymagana jest bardziej bezpieczna metoda, to powinieneś zapoznać się z treścią podrozdziału „Zastosowanie *ssh* do uruchamiania zdalnych aplikacji X” umieszczonego w dalszej części tego rozdziału.

1. Na komputerze lokalnym otwórz okno terminala.
2. Nadaj komputerowi zdalnemu (*remote1*) uprawnienia do korzystania z lokalnej konsoli X, wykonując następujące polecenie:

```
$ xhost +remote1
remote1 being added to access control list
```

3. Zaloguj się do zdalnego komputera, korzystając z dowolnego polecenia umożliwiającego zdalne logowanie, np.:

```
$ telnet -l użytkownik remote1
Password:
```

Zamiast słowa *użytkownik* użyj nazwy Twojego użytkownika, właściwej dla tego odległego komputera. Po uzyskaniu połączenia zostaniesz poproszony o wpisanie hasła dostępu.



Domyślnie w systemie Fedora Core usługa *telnet* nie jest aktywna. Użytkownik, który decyduje się na uruchomienie tej usługi (w naszym przypadku użytkownik komputera *remote1*), musi rozważyć ewentualne konsekwencje, jakie to posunięcie może implikować w dziedzinie bezpieczeństwa.

4. Wpisz hasło dostępu do zdalnego komputera (jeżeli wszystko pójdzie dobrze, to w oknie terminala będziesz zalogowany na zdalnym komputerze).
5. Ustaw wartość zmiennej `DISPLAY` komputera zdalnego, tak aby wskazywała na Twój lokalny komputer. Jeżeli Twój komputer nazywa się *antares* (w domenie lokalnej), to takie polecenie może wyglądać następująco:

```
$ export DISPLAY=antares:0
```

(Jeżeli na systemie zdalnym korzystasz z powłoki *csh*, to powinieneś wykonać polecenie `setenv DISPLAY antares:0`).

6. Po wykonaniu powyższych poleceń każda aplikacja *X* uruchamiana na zdalnym komputerze z poziomu tej powłoki będzie wyświetlana na konsoli :0 Twojego komputera lokalnego. Aby np. na zdalnym komputerze uruchomić okno terminala, tak by pojawiło się na konsoli lokalnej, powinieneś wykonać polecenie:

```
$ xterm -title "Okno terminala z komputera remote1"
```

Na ekranie pojawi się okno terminala zatytułowane *Okno terminala z komputera remote1*.



Uruchamiając zdalne aplikacje, zawsze z definicji używam opcji `-title` i w tytule okna staram się umieścić nazwę zdalnego komputera — w ten sposób na bieżąco mam na ekranie informacje, że pracuję na zdalnym komputerze.

Pracując ze zdalnymi aplikacjami, powinieneś pamiętać o kilku sprawach.

- ♦ Jeżeli często korzystasz z aplikacji uruchamianych zdalnie na innych komputerach, to możesz do pliku konfiguracyjnego Twojej powłoki na zdalnym komputerze (np. do pliku *.bashrc* dla powłoki *bash*) na stałe dodać definicję zmiennej `DISPLAY`. Dzięki temu każda aplikacja, którą uruchomisz zdalnie, będzie kierowana na Twoją konsolę.
- ♦ Nawet jeżeli wygląda to tak, jakby aplikacja pracowała lokalnie, to nie możesz zapominać o tym, że tak naprawdę cała praca jest wykonywana na systemie zdalnym. Jeżeli np. uruchomisz zdalnie edytor tekstu, to będzie on wykorzystywał zasoby procesora tego zdalnego komputera, a kiedy zapiszesz edytowany plik, to zostanie on zapisany na zdalnym komputerze.



Nigdy nie zapominaj, że, uruchamiając zdalne aplikacje, widzisz na swojej konsoli tylko obraz — aplikacja pracuje na innym komputerze! Czasami zdarza się, że użytkownicy zapominają, iż pracują na zdalnym komputerze i omyłkowo modyfikują zawartość istotnych konfiguracyjnych plików systemowych (takich jak np. */etc/fstab*) — taka pomyłka może się zakończyć poważnym uszkodzeniem systemu zdalnego.

Zabezpieczanie zdalnych aplikacji X

Istnieje kilka różnych, tradycyjnych metod zabezpieczających przed pojawieniem się na Twojej konsoli zdalnych aplikacji X. Każda z tych metod oferuje inny poziom bezpieczeństwa, a także wymaga zastosowania różnego nakładu sił i środków. Poniżej omówiono dwie najczęściej stosowane metody zabezpieczeń.

- ♦ **Definiowanie listy uprawnionych komputerów** — metoda ta (opisana już wcześniej jako metoda domyślna) pozwala użytkownikowi na decydowanie, które komputery będą miały dostęp do konsoli X Twojego systemu. Do określania listy uprawnionych komputerów powinieneś użyć poleceń `xhost +nazwa_hosta` (aby dodać dany komputer do listy uprawnionych systemów) oraz `xhost -nazwa_hosta` (aby usunąć dany komputer z list uprawnionych systemów). Jeżeli dodasz dany komputer do listy uprawnionych systemów, to automatycznie dostęp do Twojej konsoli będzie miał każdy użytkownik zalogowany na takim komputerze. Takie rozwiązanie jest zalecane do stosowania w małych sieciach łączących komputery użytkowników, do których masz pełne zaufanie.
- ♦ **Magiczne ciasteczko (*Magic Cookie*)** — metoda ta, określana często jako metoda *MIT-MAGIC-COOKIE-1*, pozwala na przydzielenie uprawnień dostępu do Twojej konsoli X indywidualnym użytkownikom. Zanim zdalny klient będzie mógł się połączyć z Twoją konsolą, będzie musiał przedstawić aktualne ciasteczko (*cookie*) uprawniające go do takiego połączenia.
- ♦ Procedura połączenia wygląda następująco: najpierw musisz wygenerować „magiczne ciasteczko” (korzystając z polecenia `mcookie`); do tego ciasteczka będziesz się odwoływał podczas uruchamiania swojego serwera X (zazwyczaj korzystając z kopii ciasteczka zlokalizowanej w pliku `$HOME/.Xauthority`). Następnie rozsyłasz kopie ciasteczka do wszystkich zdalnych użytkowników, którzy powinni mieć uprawnienia do wyświetlania swoich klientów X na Twojej konsoli — a zdalni klienci X odwołują się do tego ciasteczka podczas uruchamiania.

Istnieją również inne metody zabezpieczania sesji X, w zdecydowanej większości oparte na przechowywaniu jednego lub więcej losowo wygenerowanych kluczy w pliku `.Xauthority`. Metoda *XDM-AUTHORIZATION-1* wykorzystuje do uzyskania dostępu klucz dwuczęściowy — pierwsza część, 56-bitowa, oparta jest na kluczu wygenerowanym za pomocą algorytmu DES, druga część, 64-bitowa, zawiera pewne dane wygenerowane losowo. Jeżeli poszczególne systemy obsługują bezpieczny algorytm RPC, to istnieje możliwość zastosowania metody *SUN-DES-1* polegającej na uzyskiwaniu dostępu za pomocą pary kluczy prywatnych i publicznych, które mogą być weryfikowane osobno dla poszczególnych użytkowników. Istnieje wreszcie metoda *MIT-KERBEROS-5* umożliwiająca wzajemną weryfikację klienta X oraz serwera X w oparciu o zaufany serwer usług Kerberos.



Więcej informacji na temat usługi Kerberos znajdziesz na witrynie internetowej Massachusetts Institute of Technology, <http://web.mit.edu/kerberos>.

Uruchamianie zdalnych aplikacji X przy użyciu ssh

Polecenie `ssh` stanowi nie tylko bezpieczny mechanizm umożliwiający logowanie się na innych komputerach, ale również zapewnia bezpieczny sposób zdalnego uruchamiania aplikacji X. Po zalogowaniu się do zdalnego komputera przy użyciu `ssh` możesz używać

bezpiecznego kanału transmisji do przesyłania obrazu zdalnych aplikacji na Twoją lokalną konsolę X. Poniżej przedstawiono przykład takiego połączenia.

1. Połącz się ze zdalnym komputerem przy użyciu polecenia `ssh`:

```
$ ssh kuba@remotel
kuba@remotel's password: *****
```

2. Po zalogowaniu się wpisz polecenie uruchamiające wybraną aplikację X — okno powiązane z tą aplikacją pojawi się na Twojej lokalnej konsoli. Przykładowo, uruchom edytor tekstu *gedit*, wydając polecenie:

```
$ gedit &
```

Aby powyższa operacja się powiodła, nie musisz już udostępniać lokalnej konsoli za pomocą polecenia `xhost`. Dzieje się tak dlatego, że demon SSH (`sshd`) na zdalnym systemie kreuje bezpieczny kanał połączenia do Twojego komputera dla aplikacji X. Aby uniknąć konfliktów z innymi, rzeczywistymi numerami konsoli, demon SSH domyślnie wykorzystuje konsolę *localhost:10.0*.

W systemie Fedora Linux funkcja przekierowywania `ForwardX11` jest domyślnie włączona. W celu jej uaktywnienia należy w pliku `/etc/ssh/sshd_config` (w sposób zdalny) parametrowi `ForwardX11` przypisać wartość *yes*.

Zastosowanie emulatorów do uruchamiania aplikacji z innych systemów operacyjnych

Linux pozwala na uruchamianie aplikacji przeznaczonych oryginalnie dla systemu Linux, X Window System oraz wielu systemów uniksowych. Co więcej, duża ilość aplikacji opracowanych pierwotnie dla innych systemów operacyjnych została dostosowana do pracy na platformie Linux. Z drugiej jednak strony, istnieją tysiące aplikacji działających na innych systemach operacyjnych, które do tej pory nie mają swoich wersji dla systemu Linux.

W systemie Linux istnieje możliwość uruchamiania (przynajmniej niektórych) aplikacji przeznaczonych dla innych systemów operacyjnych przy użyciu tzw. *emulatorów*. Emulator, jak sama nazwa wskazuje, jest programem, który usiłuje działać jak coś, czym tak naprawdę nie jest. W systemach operacyjnych emulator usiłuje udostępnić aplikacji środowisko, które z jej punktu widzenia wygląda jak system operacyjny, dla którego jest ona przeznaczona.



Najbardziej znany w środowisku Linux emulator *WINE*¹⁵, tak naprawdę nie jest emulatorem. Technicznie rzecz biorąc, *WINE* jest mechanizmem, który zamiast emulować pełny system Windows, po prostu udostępnia w systemie Linux interfejs Windows API — czyli interfejs, na który aplikacja Windows czeka po uruchomieniu. Jako ciekawostkę podamy, że niektórzy użytkownicy nazwę *WINE* tłumaczą jako „**WINE Is Not an Emulator**”¹⁶.

¹⁵ Popularny emulator systemu Windows — *przyp. tłum.*

¹⁶ z ang. „*WINE* nie jest emulatorem” — *przyp. tłum.*

W kolejnych podrozdziałach omówimy pokrótce dostępne dla systemu Linux emulatory, umożliwiające uruchamianie aplikacji przeznaczonych dla następujących systemów operacyjnych:

- ♦ DOS,
- ♦ Windows 3.1,
- ♦ Windows 95,
- ♦ Windows 98,
- ♦ Windows 2000,
- ♦ Windows NT,
- ♦ Windows XP,
- ♦ Macintosh (Mac OS).



Teoretycznie każda aplikacja kompatybilna z platformą Win32 powinna być uruchomiona po zastosowaniu takiego oprogramowania jak WINE (omówiono je w dalszej części książki). Niezależnie od tego, każda aplikacja, którą zamierzasz uruchamiać przy użyciu emulatora, powinna zostać indywidualnie, szczegółowo przetestowana.

Oto lista dostępnych programów emulujących.

- ♦ *DOSEMU* — pozwala na uruchamianie programów przeznaczonych dla systemu DOS (dostępny do pobrania na stronie internetowej znajdującej się pod adresem <http://dosemu.sourceforge.net>).
- ♦ *WINE* — pozwala na uruchamianie programów przeznaczonych dla systemów Windows 3.1, Windows 95, Windows 98, Windows 2000, Windows NT oraz Windows XP (choć programy przeznaczone dla ostatnich dwóch nie są jeszcze zbyt poprawnie obsługiwane).
- ♦ *ARDI Executor* — pozwala na uruchamianie programów przeznaczonych dla systemu operacyjnego Mac OS komputerów Macintosh.

Ogólnie rzecz biorąc, im starszy i mniej skomplikowany jest program, tym większa szansa na jego poprawne uruchomienie przy użyciu emulatora. Zazwyczaj aplikacje pracujące w trybie tekstowym działają w emulatorach lepiej niż aplikacje wykorzystujące środowisko graficzne. Powinieneś również pamiętać o tym, że programy uruchamianie przy użyciu emulatora zwykle działają nieco (a nawet bardziej niż nieco...) wolniej — dzieje się tak ze względu na to, że często musimy umieszczać w emulatorach dodatkowe fragmenty kodu umożliwiające usuwanie błędów (tzw. „odpluskwanie” — *debugging*).

Uruchamianie aplikacji przeznaczonych dla systemu DOS

Ponieważ system Linux został pierwotnie zaprojektowany dla komputerów PC, to w międzyczasie powstało bardzo wiele narzędzi ułatwiających programistom i użytkownikom w miarę „bezbolesne” przejście między systemami Linux oraz DOS. W systemie Linux istnieje zestaw narzędzi o nazwie *mtools*, który umożliwia użytkownikowi pracę z plikami i katalogami systemu DOS. Emulator systemu DOS, *dosemu*, umożliwia uruchamianie

aplikacji przeznaczonych dla tego systemu w środowisku DOS sztucznie utworzonym pod kontrolą systemu Linux (w bardzo podobny sposób działają programy uruchomione DOS-ie wywołanym z systemu Windows).

Zastosowanie zestawu narzędzi mtools

Pakiet *mtools* składa się w większości z odpowiedników poleceń systemu DOS wyróżnionych obecnością litery *m* na początku nazwy każdego polecenia (aczkolwiek istnieje kilka wyjątków od tej reguły). Korzystając z poleceń tego pakietu, możesz bez problemów pracować z plikami i katalogami systemu DOS. W tabeli 5.4 przedstawiono zestawienie poleceń pakietu *mtools*, dostępnych w systemie Linux (aby z nich skorzystać, powinieneś wcześniej zainstalować ten pakiet).

Pakietu *mtools* zazwyczaj używam do kopiowania plików (za pomocą np. dyskietek) między moim systemem Linux a systemami Windows, do których nie mam dostępu przez sieć. Najczęściej korzystam z polecenia *mcopy*, które pozwala na używanie literowych oznaczeń napędów zamiast nazw urządzeń. Jeżeli np. chcę skopiować plik *vi.exe* z dyskietki do bieżącego katalogu systemu Linux, to wykonuję polecenie:

```
# mcopy a:\vi.exe
```



Domyślnie zawartość dyskietek może być odczytywana (zapisywana) wyłącznie przez użytkownika *root* oraz innych, którzy mają odpowiednie uprawnienia. Aby napęd dyskietek był dostępny dla wszystkich użytkowników (zakładając, że napęd dyskietek jest oznaczony jako *A:*), powinieneś (jako użytkownik *root*) wykonać polecenie `chmod 666 /dev/fd0`.

Zastosowanie pakietów dosemu

Pakiet emulatora systemu DOS *dosemu* nie jest już od pewnego czasu dołączany do dystrybucji systemu Fedora Core. W celu użycia tego emulatora ze strony internetowej znajdującej się pod adresem <http://dosemu.sourceforge.net> należy pobrać pakiet RPM o nazwie *dosemu*.

Za pomocą narzędzi udostępnianych przez pakiet *dosemu* możesz uruchamiać aplikacje przeznaczone dla systemu DOS oraz korzystać z komputera, tak jakbyś miał zainstalowany „na pokładzie” system MS-DOS, czyli m.in. wykorzystywać urządzenia zewnętrzne, pracować z plikami konfiguracyjnymi systemu DOS oraz w pełni stosować system plików DOS.

Aby uruchomić emulator *dosemu*, możesz użyć jednego z poniższych poleceń.

- ♦ *dosemu* — uruchamia emulator systemu DOS z poziomu dowolnej powłoki.
- ♦ *xdosemu* — uruchamia emulator systemu DOS w osobnym oknie środowiska graficznego X.
- ♦ *xtermdos* — uruchamia emulator systemu DOS w oknie x-terminala (*xterm*).
- ♦ *dosdebug* — uruchamia debugger, który umożliwia przeglądanie informacji oraz komunikatów o błędach podczas uruchamiania programów DOS.

Tabela 5.4. Zestawienie poleceń pakietu *mtools*

Polecenie	Opis i przeznaczenie
<code>mattrib</code>	Odpowiednik DOS-owego polecenia <code>attrib</code> ; pozwala na zmianę atrybutów plików systemu MS-DOS
<code>mbadblocks</code>	Odpowiednik DOS-owego polecenia <code>badblocks</code> ; pozwala na sprawdzenie powierzchni dyskiety i zaznaczenie w jej tabeli alokacji plików (FAT) wszystkich znalezionych uszkodzonych sektorów
<code>mcd</code>	Odpowiednik DOS-owego polecenia <code>cd</code> ; pozwala na zmianę bieżącego katalogu w systemie plików MS-DOS (do którego odwołujesz się za pomocą pakietu <i>Mtools</i> — katalogiem domyślnym jest tutaj <code>A:\</code>)
<code>mcheck</code>	Odpowiednik DOS-owego polecenia <code>check</code> ; pozwala na włączenie weryfikacji zapisu plików
<code>mcopy</code>	Odpowiednik DOS-owego polecenia <code>copy</code> ; pozwala na kopiowanie plików między różnymi katalogami
<code>mdel</code>	Odpowiednik DOS-owego polecenia <code>del</code> ; pozwala na usuwanie plików
<code>mdeltree</code>	Odpowiednik DOS-owego polecenia <code>deltree</code> ; pozwala na usuwanie katalogu łącznie z jego zawartością oraz podkatalogami
<code>mdir</code>	Odpowiednik DOS-owego polecenia <code>dir</code> ; pozwala na wyświetlenie zawartości katalogu
<code>mdu</code>	Odpowiednik linuksowego polecenia <code>du</code> ; pozwala na wyświetlenie informacji o ilości miejsca zajmowanego na dysku przez katalog(i) systemu DOS
<code>mformat</code>	Odpowiednik DOS-owego polecenia <code>format</code> ; pozwala na formatowanie dyskietek DOS
<code>minfo</code>	Umożliwia wyświetlanie informacji na temat urządzeń DOS-owych, takich jak np. stacja dyskietek
<code>mkmanifest</code>	Umożliwia utworzenie skryptu powłoki odtwarzającego linuksowe nazwy plików, które zostały obcięte na skutek działania poleceń DOS-owych
<code>mlabel</code>	Odpowiednik DOS-owego polecenia <code>label</code> ; pozwala na nadanie etykiety dyskowi
<code>mmd</code>	Odpowiednik DOS-owego polecenia <code>md</code> ; pozwala na tworzenie nowych katalogów
<code>mmount</code>	Umożliwia zamontowanie dysku DOS-owego w systemie Linux
<code>mmove</code>	Odpowiednik DOS-owego polecenia <code>move</code> ; pozwala na przenoszenie plików między katalogami
<code>mrdd</code>	Odpowiednik DOS-owego polecenia <code>rd</code> ; umożliwia usuwanie katalogów
<code>mren</code>	Odpowiednik DOS-owego polecenia <code>ren</code> ; umożliwia zmianę nazwy plików
<code>mshowfat</code>	Wyświetla z tabeli alokacji plików wpis dotyczący danego pliku
<code>mtoolstest</code>	Sprawdza poprawność plików konfiguracyjnych pakietu <i>mtools</i>
<code>mtype</code>	Odpowiednik DOS-owego polecenia <code>type</code> ; wyświetla na ekranie zawartość pliku tekstowego
<code>mzip</code>	Umożliwia wykonywanie różnych operacji z dyskami wymiennymi typu <i>Zip</i> , min. wysuwanie z napędu, zabezpieczanie przed zapisem, wyświetlanie zawartości itp.

Podstawowe informacje na temat środowiska DOS są przechowywane w pliku */etc/dosemu/dosemu.conf*. Poniżej omówiono sposoby konfiguracji poszczególnych elementów środowiska DOS.

- ♦ **Procesor** — domyślnie emulowany jest procesor 80386. Możesz zmodyfikować te ustawienia, zmieniając wartość parametru `$_cpu` na 80486 lub 80586 (w przypadku procesora Pentium).
- ♦ **Klawiatura** — domyślnym ustawieniem jest `auto`, co powoduje, że ustawienia klawiatury są oparte na bieżących ustawieniach konsoli systemu Linux. W razie potrzeby możesz je zmieniać, modyfikując wartość parametru `$_layout` na jedną z innych, dozwolonych kombinacji kraj/język, których pełną listę znajdziesz w pliku *dosemu.conf* (np. dla klawiatury US/English powinieneś wybrać wartość `us`, dla niemieckiego układu klawiatury — `de` itd.).
- ♦ **Ustawienia środowiska X** — do dyspozycji masz kilka parametrów (ich nazwy rozpoczynają się od `$_X`), które pozwalają na modyfikację takich ustawień, jak nazwa pojawiająca się na pasku tytułowym okna DOS, rodzaj używanej czcionki, częstotliwość migotania kursora itp.
- ♦ **Ustawienia stacji dyskietek** — napęd *A:* (*/dev/fd0*) jest standardowo ustawiony jako stacja dyskietek 3,5" (`threeinch`), natomiast napęd *B:* (*/dev/fd1*) nie jest przypisany. W razie potrzeby drugi napęd może być ustawiony jako stacja dyskietek 3,5" (`threeinch`), 5,25" (`fiveinch`), `atapi` lub `empty`.
- ♦ **Obraz dysku twardego** — system plików, który jest dostępny po uruchomieniu emulatora DOS, w rzeczywistości jest obrazem dysku zapisanym w pliku */var/lib/dosemu*. Domyślnie *dosemu* wykorzystuje obraz *hdimage.first* (dowiązany do *hdimage.freedos*). Możesz zmienić to dowiązanie na inny obraz dysku bądź na inny system plików DOS (np. */dev/hda1*, zakładając oczywiście, że jest to partycja DOS).
- ♦ **Ustawienia portu szeregowego** — domyślnie nie są przypisywane żadne porty szeregowy. Dowolny z dostępnych portów szeregowych (`$_com1` do `$_com4`) może zostać przypisany do urządzeń, takich jak modem (*/dev/modem*), mysz (*/dev/mouse*) lub konsola terminala (*/dev/tty0*).

Emulator *dosemu* jest skonfigurowany pod kątem użytkownika `root`. Udostępnianie środowiska DOS dla wielu użytkowników nie jest zbyt bezpieczne, ponieważ system plików DOS nie posiada żadnych mechanizmów bezpieczeństwa dla plików i katalogów, jak to ma miejsce w systemie plików Linux. Jeżeli mimo to chcesz, aby inni użytkownicy mieli dostęp do emulatora systemu DOS, to powinieneś zmodyfikować plik konfiguracyjny */etc/dosemu.users*, dodając do niego następujące wiersze:

```
root c_all
all c_all
```

Więcej informacji na temat emulatora *dosemu* znajdziesz na stronie domowej tego projektu, <http://www.dosemu.org>.

Zastosowanie pakietu WINE do uruchamiania aplikacji przeznaczonych dla systemu Windows

Zespół programistów projektu *WINE* (<http://www.winehq.com>) dokłada wszelkich starań, aby umożliwić uruchamianie aplikacji przeznaczonych dla systemu Microsoft Windows na komputerach pracujących pod kontrolą systemu Linux i pokrewnych. Pakietu *WINE* nie można do końca nazwać emulatorem, ponieważ tak naprawdę nie emuluje całego systemu operacyjnego Windows. *WINE* udostępnia interfejs Win32 API oraz odpowiednie interfejsy systemu Windows 3.x — z tego powodu powinno się go raczej określać jako *warstwę zgodności z systemem Windows (Windows compatibility layer)*.

Poszukując pakietu WINE przeznaczonego dla systemu Fedora Linux, można sprawdzić kilka poniższych witryn.

- ♦ **Różne strony internetowe** — istnieje możliwość pobrania za darmo pakietu WINE z kilku witryn. W celu uzyskania listy witryn należy zaglądnąć na stronę internetową znajdującą się pod adresem *winehq.com*. Najnowszą wersję pakietów RPM przeznaczonych dla systemu Fedora Core 1 pobrałem z poniższej strony: <http://sourceforge.net/projects/wine/>. Jeśli pojawi się wersja dla systemu Fedora Core 2, powinienś z niej skorzystać.
- ♦ **WineX** — komercyjna wersja pakietu WINE o nazwie WineX udostępniona przez firmę *TransGaming Inc.* (www.transgaming.com). Firma specjalizuje się w uruchamianiu pod systemem Linux w oparciu o pakiet WINE gier przeznaczonych dla systemów Windows. W celu zapoznania się z omówieniem pakietu WineX zajrzyj do rozdziału 7.
- ♦ **CodeWeavers** — jeśli zależy Ci na używaniu pakietu Microsoft Office lub „wtyczek” przeglądarki internetowej, wtedy warto skontaktować się z firmą *CodeWeavers* (www.codeweavers.com), która oferuje odpowiednie rozwiązania o nazwach *CrossOver Office* i *CrossOver Plug-in*. Chociaż oba produkty trochę kosztują, to jednak dysponują wygodnymi interfejsami służącymi do instalacji i zarządzania obsługiwanych przez nie aplikacji Windows i „wtyczek” przeglądarek internetowych.

Jeśli faktem jest stwierdzenie, że za pomocą pakietu *WINE* można uruchomić wiele aplikacji przeznaczonych dla systemu Windows, to trzeba również pamiętać o tym, że uruchamianie poszczególnych aplikacji będzie wymagało pewnego wysiłku. Jeżeli rozważasz możliwość zastosowania systemu Linux zamiast używanego do tej pory systemu Windows, to pakiet *WINE* daje Ci możliwość dalszego korzystania przynajmniej z niektórych używanych do tej pory aplikacji Windows.



W przypadku komputera, który jest niezbędny do wykonywania Twojej pracy, przed użyciem aplikacji wykorzystującej pakiet *WINE* należy ją dokładnie przetestować. Mimo że do tej pory jeszcze nie zdarzyło mi się, aby pakiet *WINE* wyrządził poważne szkody w moim systemie Linux, to jednak w pewnych okolicznościach od czasu do czasu zdarza się, że potrafi zawiesić całe środowisko graficzne. Na stronach internetowych, których adresy podano powyżej, musisz poszukać informacji na temat konkretnych kwestii związanych z uruchamianiem aplikacji systemu Windows w systemie Fedora Linux.

Poza coraz nowszymi wersjami pakietu *WINE* na witrynie internetowej projektu znajdziesz listę aplikacji, które możesz uruchomić, korzystając z tego pakietu. W obecnej chwili na tej liście znajduje się ponad 1000 aplikacji, aczkolwiek obsługa wielu z nich ciągle jeszcze

pozostawia wiele do życzenia. Warto jednak mieć świadomość, że lista tych aplikacji powiększa się z dnia na dzień oraz główny nurt wysiłków programistów projektu kierowany jest w stronę możliwości uruchamiania najważniejszych aplikacji przeznaczonych dla systemów Windows 98 i 2000.

Warto w tym miejscu zwrócić uwagę na inny komercyjny produkt, umożliwiający pełne korzystanie z aplikacji systemu Windows na komputerze pracującym pod kontrolą systemu Linux — *VMware*. Korzystając z tego pakietu, możesz równolegle uruchamiać na tym samym komputerze systemy, takie jak Windows 98/NT/2000/XP oraz Linux¹⁷. Więcej informacji na ten temat znajdziesz na witrynie <http://www.vmware.com>. Program Win4Lin (<http://www.netraverse.com>) pozwala na uruchomienie aplikacji systemu Windows w systemie Linux.

W kolejnym podrozdziale dowiesz się, w jaki sposób skonfigurować system Linux do uruchamiania aplikacji Windows przy użyciu pakietu *WINE*.

Zastosowanie pakietu WINE

Aby za pomocą pakietu *WINE* można było uruchamiać aplikacje przeznaczone dla systemu Windows, musi najpierw zostać utworzone środowisko, które będzie „udawało” system Windows. W dalszej części rozdziału krok po kroku zostanie omówiona instalacja i konfiguracja najnowszej wersji pakietu *wine* przeznaczonego dla systemu Fedora Core (pakiet RPM można pobrać ze strony internetowej znajdującej się pod adresem <http://sourceforge.net/projects/wine> po kliknięciu odnośnika *RedHat Packages*).

Po zainstalowaniu pakietu *WINE* w systemie Fedora Core zostaną automatycznie utworzone pliki */etc/wine/wine.conf* oraz */etc/wine/system.reg*, które, podobnie jak rejestry systemu Windows, identyfikują lokalizacje składników wymaganych podczas uruchamiania aplikacji Windows.

Podstawowe katalogi systemu operacyjnego Windows wykorzystywane przez program *WINE* znajdują się w katalogu *\$HOME/.wine/c* (dotyczy każdego użytkownika). Przyjmuje to mapowanie dysku *C:* na katalog programu *WINE*. Katalog */usr/share/wine-c* zawiera wersję powyższego katalogu obowiązującego w skali całego systemu, do którego może się odwołać każdy użytkownik. W tabeli 5.5 przedstawiono sposób mapowania dysku *C:* oraz innych dysków i urządzeń wykorzystywanych przez system Windows na poszczególne katalogi i urządzenia systemu Linux.

¹⁷ Ten bardzo pobieżny opis może być nieco mylący. *VMware* nie jest emulatorem — to program, który udostępnia użytkownikowi swego rodzaju wirtualne środowisko (wirtualny komputer), na którym możesz zainstalować niemal dowolny system operacyjny, pracujący całkowicie niezależnie od systemu operacyjnego hosta. Dzięki temu, pracując np. w systemie Linux, możesz przy wykorzystaniu *VMware* zainstalować i uruchomić pełny system Windows (wraz z dowolnymi aplikacjami) pracujący w osobnym okienku niezależnie od innych aplikacji Linux. Istnieje również wersja pakietu *VMware* przeznaczona dla systemu Windows, co pozwala na odwrócenie całej sytuacji i zainstalowanie na takim wirtualnym komputerze pełnego systemu Linux. W czasie tłumaczenia tej książki (wrzesień 2004 r.) pełną wersję pakietu *VMware* dla systemu Linux (przeznaczoną dla dystrybucji Red Hat, SuSE, Mandrake oraz Caldera OpenLinux) można było nabyć za \$329 — *przyp. tłum.*

Tabela 5.5. Oznaczenia dysków i urządzeń systemu Windows w pakiecie WINE

Opis	Litera dysku lub nazwa portu	Katalog systemu Linux	Nazwa urządzenia w systemie Linux
Stacja dyskietek	A:	<i>/mnt/floppy</i>	<i>/dev/fd0</i>
Dysk twardy nr 1	C:	<i>\$HOME/.wine/c</i> <i>/usr/share/wine-c</i>	
Napęd CD-ROM	D:	<i>/mnt/cdrom</i>	<i>/dev/cdrom</i>
Katalog tymczasowy	E:	<i>/tmp</i>	
Katalog domowy użytkownika	F:	<i>\$HOME</i> (katalog domowy użytkownika)	
Katalog główny	Z:	<i>/</i>	
Porty szeregowo	<i>COM1</i>		<i>/dev/ttyS0</i>
	<i>COM2</i>		<i>/dev/ttyS1</i>
	<i>COM3</i>		<i>/dev/ttyS2</i>
	<i>COM4</i>		<i>/dev/modem</i>
Port równoległy	<i>Lpt1</i>		<i>/dev/lp0</i>
Spooler drukarki	<i>LPT1</i>	Kieruje zadania dla drukarki do polecenia <i>lpr</i> (<i> lpr</i>)	
WineLook (system operacyjny, jaki „widzi” aplikacja po uruchomieniu)	<i>Win95</i> (domyślnie), <i>Win31</i> , <i>Win95</i> lub <i>Win98</i>		

W katalogu */usr/share/wine-c* (to jest, na Twoim dysku C:...) powinieneś znaleźć pliki i katalogi, które dla użytkowników systemu Windows powinny wyglądać znajomo: pliki *autoexec.bat* i *config.sys* (początkowo są one puste) oraz katalogi *My Documents*, *Program Files* oraz *windows*.

Nawet zakładając, że domyślna struktura katalogów Ci odpowiada, i tak możesz rozważyć konieczność wprowadzenia kilku zmian. Przykładowo możesz zmodyfikować polecenie, które będzie używane do drukowania (definicja znajduje się w pliku */etc/wine/wine.conf*, domyślnie jest to *| lpr*), tak aby wykorzystywana była inna drukarka lub odmienne opcje drukowania.



W celu uzyskania szczegółowych informacji na temat konfiguracji pliku *wine.conf* należy zapoznać się z dokumentem Wine User Guide. Podczas instalowania pakietu *wine* w wersji dla systemu Fedora Core dokument ten (*wine-user.pdf*) jest umieszczany w katalogu */usr/share/doc/wine**.

Konfiguracja aplikacji Windows

Najlepszym sposobem na zrozumienie zasad, jakie rządzą konfiguracją i uruchamianiem aplikacji Windows, będzie praktyczne przebrnięcie przez proces uruchamiania jakiejś prostej aplikacji przy użyciu pakietu *WINE*. Aby opisana poniżej procedura miała szansę zadziałania, muszą być spełnione następujące warunki.

- ♦ Pakiet *WINE* musi być poprawnie zainstalowany w systemie Fedora Core.
- ♦ Musisz mieć zapewniony dostęp do poszczególnych składników systemu Windows, włączając w to zarówno pliki aplikacji, którą chcesz uruchomić, jak i bibliotek DLL, z których dana aplikacja może korzystać.

Jeżeli używasz komputera wielosystemowego (posiadasz zainstalowane systemy Fedora Core oraz Windows), to najlepszym sposobem postępowania będzie uruchomienie systemu Fedora Linux, a następnie zamontowanie w wybranym miejscu partycji Windows, tak abyś miał możliwość swobodnego skopiowania niezbędnych plików. W przeciwnym wypadku możesz pominąć pierwszy krok poniższej procedury i po prostu skopiować niezbędne pliki z dysku CD lub z sieci.

W celu zilustrowania problemu spróbujemy uruchomić przy użyciu pakietu *WINE* dobrze znany użytkownikom systemu Windows program *Microsoft Paint* (*mspaint.exe*). Aby tego dokonać, wykonaj poniższe polecenia.

1. Jeżeli oprócz systemu Linux masz również zainstalowany (na innej partycji) system Windows, to przede wszystkim powinieneś zamontować partycję systemu Windows w wybranym miejscu. Przykładowo możesz tego dokonać w następujący sposób.

- ♦ Jako użytkownik *root* wpisz w oknie terminala polecenie:

```
# fdisk -l
```

W rezultacie na ekranie pojawi się lista partycji dostępnych na Twoim dysku(ach) twardym. Odszukaj partycję, która jest oznaczona *WIN95 FAT32* (lub podobnie).

- ♦ Utwórz katalog, do którego zamontujesz partycję Windows. Aby np. skorzystać z katalogu */mnt/win*, wykonaj poniższe polecenie:

```
# mkdir /mnt/win
```

- ♦ Zakładając, że partycja Windows jest pierwszą partycją zlokalizowaną na pierwszym dysku IDE (*/dev/hda1*), zamontuj partycję Windows za pomocą polecenia:

```
# mount -t vfat /dev/hda1 /mnt/win
```

Po wykonaniu tego polecenia zawartość partycji Windows powinna być dostępna przez katalog */mnt/win*.

- ♦ Aby partycja Windows była na stałe dostępna z poziomu systemu Linux, powinieneś dodać do pliku */etc/fstab* poniższy wiersz:

```
/dev/hda1      /mnt/win      vfat          defaults      0            0
```

Dodanie takiego wpisu powoduje, że partycja Windows jest automatycznie montowana do katalogu */mnt/win*, za każdym razem kiedy będziesz uruchamiał system Fedora Core.



Dla celów dalszej części niniejszej procedury zakładamy, że partycja Windows jest dostępna przez katalog */mnt/win*. Jeżeli w Twoim przypadku tak nie jest, to powinieneś w jakiś inny sposób skopiować niezbędne pliki systemu Windows do katalogów przedstawionych w opisanej procedurze.

2. Skopiuj plik *mspaint.exe* do katalogu, do którego masz dostęp z poziomu systemu Linux. Jeżeli np. partycja Windows została zamontowana w opisany powyżej sposób, możesz wykonać polecenie:

```
# cp /mnt/win/Program\ Files/Accessories/mspaint.exe /bin
```

3. Uruchom polecenie *wine*, podając jako argument *mspaint.exe* — w ten sposób przekonasz się, czy do uruchomienia tego programu niezbędne będą jakieś dodatkowe biblioteki DLL:

```
# wine /bin/mspaint.exe
err: module:PE_fixup_imports Module (file) MFC42.DLL (which is
needed by Z:\a\mspaint.exe) not found
```

Taki komunikat oznacza, że do uruchomienia programu *mspaint.exe* niezbędna jest biblioteka *mfc42.dll*.

4. Skopiuj niezbędny plik z partycji Windows. Przykładowo możesz tego dokonać w następujący sposób:

```
# cp /mnt/win/windows/system/mfc42.dll /usr/share/wine-c/windows/system/
```

5. Po skopiowaniu odpowiedniej biblioteki ponownie wykonaj polecenie *wine /bin/mspaint.exe* — na ekranie powinno pojawić się okno programu *Microsoft Paint*. Przykład takiego okna otwartego w systemie Fedora Core przedstawiono na rysunku 5.3.

Rysunek 5.3.

Program
Microsoft Paint
uruchomiony
w systemie Fedora
Linux przy użyciu
pakietu WINE



Na rysunku 5.3 przedstawiono program *Paint*, w którym został otwarty plik graficzny zapisany w systemie plików Linux. Aby otworzyć wybrany plik, powinieneś z menu głównego wybrać polecenie *File/Open* (jeżeli dysponujesz polską wersją systemu Windows, będzie to polecenie *Plik/Otwórz*). Na ekranie pojawi się okno dialogowe *Open (Otwórz)*, w którym będziesz mógł zorientować się, w jaki sposób ta aplikacja „widzi” dostępne

dyski. Kolejne litery dysków, *C:*, *D:*, *E:*, *F:* oraz *Z:* reprezentują poszczególne lokalizacje przedstawione w tabeli 5.5 (jeżeli Twój komputer dysponuje również stacją dyskietek, to powinieneś ją zobaczyć jako dysk *A:*).

Jakie aplikacje systemu Windows będą pracowały z pakietem WINE?

Lista aplikacji pomyślnie przetestowanych pod kątem współpracy z pakietem *WINE* jest przechowywana w bazie danych *WINE Application Database*, dostępnej w internecie pod adresem <http://appdb.codeweavers.com>. Baza danych jest podzielona na następujące kategorie.

- ♦ *Educational Software, CBT* — narzędzia edukacyjne, oprogramowanie do prowadzenia wspomaganych komputerowo szkoleń.
- ♦ *Games* — gry.
- ♦ *Multimedia* — programy multimedialne, do przetwarzania grafiki, dźwięku oraz wideo.
- ♦ *Networking & Communication* — oprogramowanie komunikacyjne, narzędzia sieciowe, programy do pracy z internetem.
- ♦ *Productivity* — bazy danych, narzędzia biurowe (edytory tekstu, arkusze kalkulacyjne), programy do tworzenia prezentacji, programy do tworzenia stron WWW itd.
- ♦ *Programming / Software Engineering* — narzędzia programistyczne, kompilatory itd.
- ♦ *Reference/Documentation/Info* — encyklopedie, słowniki, programy tłumaczące itd.
- ♦ *Scientific/Technical/Math* — aplikacje naukowe, inżynierskie i matematyczne.
- ♦ *Special Purpose* — programy specjalnego przeznaczenia (wszystko, co nie mieści się w pozostałych kategoriach).
- ♦ *Utilities* — różnego rodzaju narzędzia.

Jak widać, w bazie *WINE Application Database* w chwili obecnej znajduje się ponad 1200 aplikacji, aczkolwiek zdecydowana większość przypada na kategorie *Games* oraz *Networking & Communication*. Najczęściej osoba, która przetestowała daną aplikację, umieszcza w bazie informacje o platformie, na jakiej program był testowany oraz uwagi na temat problemów (i sposobów ich rozwiązania), jakie ewentualnie zdarzały się podczas uruchamiania. Aby w pełni skorzystać z zasobów bazy danych, musisz się zarejestrować i założyć swoje konto użytkownika (jest to bardzo prosta i szybka operacja).

Zastosowanie pakietu ARDI Executor do uruchamiania aplikacji przeznaczonych dla komputerów Macintosh

Poza umożliwieniem uruchamiania wielu popularnych aplikacji przeznaczonych dla komputerów Macintosh, pakiet *ARDI Executor*, produkt firmy ARDI, Inc. (<http://www.ardi.com>), pozwala na pracę z dyskietkami zapisanymi w formacie Macintosh oraz na współpracę z napędami stosowanymi w tych komputerach. Więcej informacji na temat pakietu *ARDI Executor* znajdziesz na stronie <http://www.ardi.com/executor.php>.

Firma ARDI udostępnia również bazę danych oprogramowania komputerów Macintosh, z którego można korzystać przy użyciu pakietu *Executor* — baza obejmuje już kilkaset aplikacji. Każda aplikacja jest oznaczona odpowiednim kolorem (zielonym, żółtym, pomarańczowym, czerwonym lub czarnym), w zależności od tego, jak dobrze dana aplikacja radzi sobie z pakietem *Executor*. Kolor zielony oraz żółty oznacza odpowiednio aplikacje pracujące bezbłędnie lub z niewielkimi, mało znaczącymi problemami. Kolor pomarańczowy oznacza, że aplikacja daje się uruchomić, ale jej użyteczność pozostawia wiele do życzenia. Kolor czerwony — aplikacja nie daje się uruchomić, natomiast kolor czarny informuje, że aplikacja nie będzie działać ze względu na fakt, iż wykorzystuje mechanizmy, które nie zostały jeszcze zaimplementowane w bieżącej wersji *Executor-a*. Pobieźnie przeglądając listę, można stwierdzić, że co najmniej 2/3 aplikacji plasuje się na pozycjach zielonych lub żółtych.



System Mac OS bazuje na systemach operacyjnych, takich jak Linux, możemy więc sądzić, że powstanie coraz więcej aplikacji, które będzie można uruchomić w obu wymienionych systemach. Współpraca systemów Mac OS i Linux bardziej szczegółowo została omówiona w rozdziale 26.

Podsumowanie

Biorąc pod uwagę aplikacje napisane bezpośrednio dla systemu Linux i innych systemów uniksowych, aplikacje zaadaptowane do pracy w systemie Linux i wreszcie te, które można uruchomić przy użyciu emulatorów innych systemów operacyjnych, możemy stwierdzić, że użytkownicy systemu Fedora Core mają do dyspozycji setki różnego rodzaju aplikacji. W sieci internetowej można znaleźć dziesiątki serwerów udostępniających oprogramowanie dla systemu Linux i ogromną ilość miejsc, w których można zamówić takie oprogramowanie na dyskach CD.

Aby uprościć proces instalacji oprogramowania i zarządzania aplikacjami systemu Linux, firma Red Hat opracowała specjalny format dystrybucji pakietów oprogramowania, Red Hat Package Manager. Korzystając z narzędzi RPM, takich jak np. polecenie `rpm`, użytkownik może w prosty sposób instalować, usuwać oraz aktualizować pakiety oprogramowania.

Pośród całego szeregu aplikacji dostępnych dla systemu Linux, najwyższy poziom stabilności i elastyczności oferują aplikacje napisane dla X Window System. Nic nie stoi jednak na przeszkodzie, by, korzystając z emulatorów innych systemów operacyjnych, uruchamiać w systemie Fedora Linux aplikacje pierwotnie przeznaczone dla systemów DOS, Windows95/98/2000/NT/XP oraz Macintosh.