Slackware Linux Essentials

Alan Hicks

Chris Lumens

David Cantrell

Logan Johnson

1-57176-338-4

Spis treci

Spis tabel

Spis rysunków

Spis przykładów

Przedmowa

Do kogo adresowana jest ta książka

System operacyjny Slackware Linux to potężna platforma dla komputerów z procesorami o architekturze intelowskiej. Został zaprojektowany, by być stabilnym, bezpiecznym oraz funkcjonalnym zarówno w roli serwera, jak i jako system do użytku domowego.

Ta książka została napisana, by pomóc ci rozpocząć pracę z systemem Slackware Linux. Nie jest jej celem odkrycie każdego najdrobniejszego aspektu tej dystrybucji; skupia się raczej na pokazaniu tego, do czego jest zdolna oraz przekazaniu ci podstawowej wiedzy o systemie.

Mamy nadzieję, iż w trakcie zbierania doświadczeń w swojej pracy ze Slackware uznasz tę książkę jako przydatny podręcznik. Wierzymy także, że nie omieszkasz pożyczyć jej swoim przyjaciołom, gdy ci przyjdą dowiedzieć się od ciebie czegoś na temat tego świetnego Linuksa, którego używasz.

Oczywiście książka ta nie jest pierwszej klasy powieścią, niemniej jednak staraliśmy się uczynić ją jak najbardziej interesującą. Trochę szczęścia i dostaniemy propozycję nakręcenia filmu :) Oczywiście, przede wszystkim jednak mamy nadzieję, że będzie ci ona dobrym pomocnikiem w nauce i okaże się przydatna.

No więc, przedstawienie czas zacząć.

Zmiany z pierwszej edycji

Drugie wydanie jest owocem lat ciężkiej pracy ofiarnych członków Slackware Documentation Project. Oto najważniejsze zmiany w bieżącej edycji:

- Rozdział 3, Instalacja, została uzupełniona nowymi screenshot'ami instalatorów oraz odzwierciedla zmiany, jakie zaszły w instalacji z dyskietek oraz CD.
- Rozdział 4, Konfiguracja systemu, została uaktualniona o informacje na temat nowych jąder Linuksa z serii 2.6.x.
- Rozdział 5, Konfiguracja sieci, została rozszerzona o więcej wyjaśnień odnośnie Samby, NFS oraz DHCP. Dodano też sekcję o sieciach bezprzewodowych. Rozdział ten skupia się teraz przede wszystkim na głównych zmianach w obsłudze sieci przez Slackware.
- Rozdział 6, System X Window, został całkowicie przepisany na nowo, by opisać menedżery bazujące na Xorg. Zawarty jest tu także opis graficznego menedżera logowania xdm.
- Rozdział 13, Podstawowe Polecenia Sieciowe, został rozszerzony o informacje na temat dodatkowych narzędzi sieciowych.
- Rozdział 14, Bezpieczeństwo, to całkiem nowy rozdział w tej edycji. Opisuje, w jaki sposób dbać o bezpieczeństwo w Slackware Linux.
- Rozdział 17, Emacs, kolejny nowy rozdział w tej edycji. Opisuje sposób używania Emacsa, potężnego edytora dla Uniksów.
- Rozdział 18, Zarządzanie Pakietami, został powiększony o informacje na temat skryptów SlackBuild.
- Poza wymienionymi, jest jeszcze cała masa innych zmian, drobnych i tych większych, które musiały się pojawić w związku z rozwojem Slackware.

Organizacja książki

Rozdział 1, Wprowadzenie

> Dostarcza wstępnych informacji o Linuksie, o dystrybucji Slackware, oraz o ruchach Open Source i Free Software.

Rozdział 2, Pomoc

> Opisuje pomoc dostępną w Slackware oraz w sieci.

Rozdział 3, Instalacja

> Krok po kroku opisuje proces instalacji, dodatkowo prezentując screenshoty dla zilustrowania całego

procesu.

Rozdział 4, Konfiguracja systemu

Opisuje najważniejsze pliki konfiguracyjne oraz odkrywa, w jaki sposób przeprowadzić rekompilację jądra systemu.

Rozdział 5, Konfiguracja sieci

Dowiemy się z niego, jak podłączyć maszynę z zainstalowanym Slackware do sieci. Opisane tutaj są też protokół TCP/IP, PPP/dial-up, sieci bezprzewodowe i wiele więcej.

Rozdział 6, System X Window

Opisuje, w jaki sposób ustawić i jak używać systemu graficznego X Window w Slackware.

Rozdział 7, Początkowe uruchamianie systemu

Omawia proces, w których na komputerze zostaje uruchomiony Slackware Linux. Opisuje także dual-booting, gdy na dysku zainstalowany jest też system Microsoft Windows.

Rozdział 8, Powłoka

Opisuje potężny interfejs linii poleceń linuksa.

Rozdział 9, Struktura plików

Przedstawia strukturę plików, m.in. właścicieli plików, prawa dostępu oraz linkowanie.

Rozdział 10, Pliki i katalogi

Zawiera opis poleceń używanych do zarządzania plikami i katalogami z poziomu linii poleceń.

Rozdział 11, Zarządzanie procesami

Opisuje potężne narzędzia Linuksa do zarządzania procesami, użyteczne przy równoległym uruchamianiu wielu aplikacji.

Rozdział 12, Podstawy administracji systemem

Opisuje podstawowe zadania administracyjne, takie jak dodawanie i usuwanie użytkowników, prawidłowe zamykanie systemu oraz wiele innych.

Rozdział 13, Podstawowe polecenia sieciowe

Zawiera opis kolekcji programów sieciowych dostarczanych razem ze Slackware.

Rozdział 14, Bezpieczeństwo

Opisuje wiele różnorodnych narzędzi pomocnych przy utrzymywaniu bezpieczeństwa twojego systemu, takich jak `iptables` oraz `tcpwrappers`.

Rozdział 15, Archiwizacja plików

Opisuje różne sposoby kompresji oraz narzędzia do archiwizowania plików, dostępne na Linuksa.

Rozdział 16, vi

Opisuje potężny edytor tekstowy `vi`.

Rozdział 17, Emacs

Opisuje potężny edytor tekstowy `Emacs`.

Rozdział 18, Zarządzenie pakietami w Slackware

Opisuje narzędzia zarządzające paczkami w Slackware oraz proces tworzenia własnych paczek i plików tagfiles.

Rozdział 19, ZipSlack

Zawiera opis systemu ZipSlack, który może być uruchomiony z poziomu systemu Windows i nie wymaga uprzedniej instalacji.

Dodatek A, The GNU General Public License

Opisuje warunki licencji, na których system Slackware Linux oraz ta książka może być kopiowana i dystrybuowana.

Konwencje przyjęte w tej książce

By zapewnić spójny oraz łatwy w czytaniu tekst, pewne konwencje są przestrzegane w tej książce.

Konwencje typograficzne

Italic

Czcionka *pochyła* użyta jest dla określenia poleceń, ważnego tekstu oraz, gdy po raz pierwszy zostaje użyte techniczne wyrażenie.

`Monospace`

`Czcionka o stałej szerokości` użyta jest dla błędów, poleceń, zmiennych środowiskowych oraz nazw: portów, hostów, użytkowników, grup i urządzeń, a także dla określenia zmiennych i fragmentów kodu.

Bold

Czcionka **pogrubiona** jest używana do oznaczenia danych, wpisywanych przez użytkownika.

Dane wprowadzane przez użytkownika

Klawisze pokazane są za pomocą czcionki **pogrubionej** dla odróżnienia od zwykłego tekstu. Kombinacje klawiszy, które mają zostać wciśnięte w jednej chwili są pokazane ze znakiem `+' pomiędzy nimi, jak w przykładzie poniżej:

Ctrl+**Alt**+**Del**

Oznacza to, że użytkownik powinien wcisnąć klawisze **Ctrl**, **Alt**, oraz **Del** naraz.

Klawisze, które należy wciskać w ustalonej kolejności jeden po drugim, będą oddzielane przecinkami; na przykład:

Ctrl+**X**, **Ctrl**+**S**

Co oznacza, że użytkownik powinien wcisnąć naraz **Ctrl** oraz **X**, po czym znowu równocześnie klawisze **Ctrl** oraz **S**.

Przykłady

Przykłady rozpoczynające się od `E:\>` dotyczą poleceń systemu MS-DOS®. Dopóki nie zostanie powiedziane inaczej, polecenia takie mogą zostać wykonane z okna “Command Prompt” w każdym nowszym systemie z rodziny Microsoft® Windows®.

```
D:\> rawrite a: bare.i
```

Przykłady zaczynające się od # określają, że polecenie może zostać wywołane tylko przez superużytkownika w Slackware. Możesz zalogować się do systemu jako `root`, by wpisać komendę, albo też zalogować się na swoje zwykłe konto i poleceniem su(1) uzyskać przywileje superużytkownika.

```
# dd if=bare.i of=/dev/fd0
```

Przykłady zaczynające się znakiem `%` dotyczą poleceń, które mogą zostać wywołane z poziomu zwykłego użytkownika. Dopóki nie zostanie zaznaczone inaczej, przyjmujemy iż składnia powłoki c jest użyta do określania zmiennych środowiskowych oraz do innych poleceń powłoki.

```
% top
```

Podziękowania

Projekt ten jest wynikiem wielomiesięcznej pracy wielu oddanych osób. Niemożliwe byłoby dla mnie ukończenie tego projektu bez ich pomocy. Wiele osób zasługuje na nasze podziękowania za ich bezinteresowne działanie: Keith Keller za jego pracę nad sieciami bezprzewodowymi, Joost Kremers za swoją wielką pracę samodzielnego napisania rozdziału o Emacsie, Simon Williams za rozdział o bezpieczeństwie, Jurgen Phillippaerts za rozdział Podstawowych polecenia sieciowe, Cibao Cu Ali G Colibri za inspirację oraz celnie wymierzone kopniaki. Niezliczona ilość osób przysyłała sugestie i poprawki. Niepełna ich lista zawiera: Jacob Anhoej, John Yast, Sally Welch, Morgan Landry, and Charlie Law. Pragnę też złożyć podziękowania Keithowi Kellerowi za utrzymywanie listy mailingowej dla tego projektu, oraz Carl Inglis za jej początkowe zajęcie się web hostingiem. Na samym końcu składam jakże ważne podziękowania dla Patricka J Volkerdinga za system Slackware Linux, oraz Davidowi Cantrellowi, Loganowi Johnsonowi i Chrisowi Lumensowi za ich Slackware Linux - podstawy, wydanie pierwsze. Bez ich pracy nasza nawet by nie zaistniała. Jeszcze wiele osób przyczyniło się mniej lub bardziej przy tworzeniu tego projektu. Mam nadzieję, że wybaczą mi moją słabą pamięć.

Alan Hicks, May 2005

Rozdział 1 Wprowadznie do Slackware Linux

1.1 Co to jest Linux?

Linus Torvalds rozpoczął pracę z Linuksem - jądrem systemu operacyjnego, jako prywatnym projektem w 1991 roku. Uczynił to ponieważ chciał używać systemu opartego opartego na Uniksie bez wydawania mnóstwa pieniędzy. Dodatkowo interesowały go wejścia i wyjścia procesora 386. Linux został wydany bezpłatnie tak żeby każdy mógł go poznawać i wprowadzać poprawki w myśl linencji GPL. (Zobacz Sekcja 1.3 i Dodatek A.) Dziś Linux urósł do rangi ważnego gracza na rynku systemów operacyjnych. Został zaimplementowany do pracy na różnych platformach systemowych, włączając HP/Compaq's Alpha, Sun's SPARC i UltraSPARCoraz Motorola's PowerPC chips (przez Apple Macintosh i IBM RS/6000 computers.) Setki, jeśli nie tysiące, programistów na całym świecie rozwijają Linuksa. Uruchamiają programy takie jak Sendmail, Apache i BIND, które są niezwykle popularne wśród oprogramownia serwerów internetowych. Ważne jest by pamiętać, iż termin “Linux” odnosi się do kernela - jądra systemu operacyjnego. Jądro odpowiedzialne jest za kontrolę procesów w komputerze, pamięć, dyski i urządzenia peryferyjne. Tak naprawdę to wszytsko robi Linux: kontroluje operacje wykonywane przez komputer i dba o zachowanie wszystkich programów. Różne firmy i osoby prywatne łączą jądro i rozmaite oprogramowanie w całość tworząc system operacyjny. Połączenie to nazywamy dystrybucją Linuksa.

1.1.1 Świat GNU

Projekt Linux kernel rozpoczęty został jako prywatne dążenie Linusa Torvaldsa w 1991, ale jak powiedział Isaac Newton “Jeśli byłem w stanie spojrzeć dalej to tylko dlatego, że stałem na ramionach olbrzymów.” Fundacja Wolnego Oprogramowania miała już gotową ideę grupowego oprogramowania (collaborative software) kiedy to Linus rozpoczął pracę nad krenelem. Swoje próby tytułowali GNU to akronim oznaczający poprostu “GNU to nie Unix”. Oprogramowanie GNU działa na jądrze Linux od samego początku. Ich kompilator `gcc` został użyty do skompilowania jądra. Dziś wiele narzędzi GNU, od `gcc` do `gnutar` jest wciąż podstawą wielu dystrybucji Linuksa. Z tego też powodu wielu zwolenników Free Software Foundation utrzymuje, że ich praca powinna być obdarzona tym samym zaufaniem (credit), co jądro Linuksa. Sugerują, by dystrybucje Linuksa nazywane były “dystrybucjami GNU/Linux”.

Powyższa kwestia jest tematem wielu sporów, które przebija tylko odwieczna święta wojna pomiędzy `vi` a `emacs`em. Celem tej książki nie jest rozniecanie płomiennych dyskusji, ale raczej wyjaśnienie nowicjuszom terminologii. Poprzez GNU/Linux rozumiemy dystrybucję Linuksa. Gdy piszemy Linux może to tak samo odnosić się do samego jądra, jak i do całej dystrybucji. Może to być mylące. Powszechnie jednak określenie GNU/Linux nie jest używane, ponieważ jest po prostu zbyt długie.

1.2 Co to jest Slackware?

Slackware zapoczątkowany przez Patricka Volkerdinga pod koniec 1992 roku i oficjalnie wypuszczony 17 czerwca 1993r., był pierwszą dystrybucją Linuksa, która dotarła do szerokiego grona odbiorców. Volkerding pierwszy raz poznał Linuksa gdy potrzebował taniego interpretera LISP dla projektu. Jedną z kilku dostępnych dystrybucji w tym czasie był SLS Linux z Soft Landind Systems, której Volkerding użył. Ostatecznie zdecydował się dołączyć wszystkie swoje poprawki do własnej prywatnej dystrybucji, tak żeby jego przyjaciele mogli jej używać. Szybko zyskała ona popularność, więc Volkerding zadecydował się nadać jej nazwę Slackware i udostępnić ją ogółowi. Z czasem Volkerding dodał nowe rzeczy do Slacka: łatwiejszy instalator bazujący na systemie wyboru z menu, jak również koncepcję zarządzania pakietami, która pozwala użytkownikom na łatwe dodawanie i usuwanie oraz aktualizowanie oprogramowania w ich systemie.

Jest wiele powodów, dla których Slackware jest najstarszą żyjącą dystrybucją Linuksa. Nie stara się on emulować Windowsa, stara się być Unikso-podobny jak to tylko możliwe. Nie stara się ukrywać procesów za fantazyjnymi interfejsami graficznymi typu "najedź i kliknij". Zamiast tego daje użytkownikowi pełną kontrolę nad systemem, pozwalając mu widzieć dokładnie to co się w nim dzieje. Jego twórcy nie gonią terminy i limity czasowe, każda wersja wychodzi wtedy kiedy jest gotowa.

Slackware jest dla ludzi, którzy czerpią przyjemność z nauki i dopasowywania systemu do własnych potrzeb. Stabilność i prostota Slacka jest powodem, dla którego ludzie przez lata go używają. Slackware cieszy się obecnie reputacją porządnego systemu serwerowego i całkiem niezłej stacji roboczej. Widuje się stacje z niemal

wszystkimi menedżerami okien, lub bez żadnego w ogóle. Biznesowe serwery na Slackware występują w każdym charakterze, w jakim serwer może zostać wykorzystany. Użytkownicy Slacka należą do najbardziej zadowolonych użytkowników Linuksa. Oczywiście musieliśmy to powiedzieć :^).

1.3 Otwarte źródła i wolne oprogramowanie

W obrębie społeczności linuksowej działają dwa główne ruchy ideologiczne. Ruch wolnego oprogramowania (free software) ma na celu uczynienie wszelkiego oprogramowania wolnym od ograniczeń praw własności intelektualnej. Członkowie tego ruchu wierzą, iż restrykcje te hamują postęp techniczny i prace nad dobrem ogółu. Ruch Open Source pracuje generalnie na rzecz tych samych celów, ale ma bardziej pragmatyczne podejście do tematu. Jego członkowie wolą opierać swoje argumenty na ekonomicznych i technicznych aspektach tworzenia kodu powszechnie dostępnego, niż na moralnych i etycznych przesłankach kierujących ruchem Wolnego Oprogramowania.

W opozycji do powyższych znajdują się grupy chcące utrzymać jak najściślejszą kontrolę nad ich oprogramowaniem.

Ruchowi wolnego oprogramowania przewodniczy Fundacja Wolnego Oprogramowania (FSF - Free Software Foundation), stworzona dla projektu GNU. Wolne Oprogramownie jest właściwie ideologią. Jego istotą jest zagwarantowanie określonych praw zarówno dla developerów, jak i dla użytkowników. Ta wolność odnosi się do wolności uruchamiania programów bez względu na powód : studiowanie, poznawanie i modyfikowanie kodu źródłowego, redystrybucję kodu i dzielenie się dokonanymi modyfikacjami. Dla zagwrantowania tych wolności utworzono licencję GNU General Public License (GPL). w skrócie nakazuje każdemu rozprowadzającemu programy skompilowane na licencji GPL dostarczenie kodu źródłowego. Dozwolone jest także modyfikowanie programu tak długo jak te modyfikacje dostępne są również w postaci kodu. Gwarantuje to “otwartość” oprogramowania dla społeczności, nie moze on zostać “zamknięty” dopóki zgody nie wyrazi każdy z autorów kodu (nawet modyfikacji). Większość porgramów linuksowych jest na licencji GPL.

Istotnym jest fakt, że GPL nie mówi nic o cenie. Dziwnie to może zabrzmieć, ale możesz pobierać opłaty za wolne oprogramowanie. "Wolne" oznacza prawo do posiadania kodu źródłowego, nie do ceny jaką płacisz za program. Jednakże kiedy ktoś sprzedaje Ci, lub daje skompilowany program na licencji GPL, jest również zobligowany do dostarczenia kodu źródłowego.

Kolejną popularną licencją jest licencja BSD. W odróżnieniu do GPL licencja BSD nie wymaga dostarczenia kodu źródłowego programu. Oprogramowanie wypuszczone na podstawie licencji BSD można rozprowadzać w postaci źródłowej lub formie binarnej z zachowaniem tylko paru warunków. Kwalifikacje autorów nie mogą być użyte jako element reklamy programu. Zabezpiecza ona także autora przed odpowiedzialnością za szkody wywołane na skutek użytkowania oprogramowania. Znaczna część oprogramowania dołącznonego do Slackware Linux jest na licencji BSD.

Inicjatywa Open Source, młodszy ruch Open Source, jest organizacją która istnieje jedynie aby dać wsparcie otwartemu oprogramowaniu, to jest takiemu, które ma tak samo dostępne źródło jak i gotowy do uruchomienia program. Nie oferują specjalnej licencji, ale w zamian wspierają różne typy dostępnych licencji wolnego oprogramowania.

Ideą stojącą za OSI (Open Source Initiative) jest zdobywanie coraz to większej liczby środowisk stojących za wolnym oprogramowaniem umożliwiając im pisanie własnych opensourceowych licencji i sygnowanie ich przez OSI. Wiele kompanii chce udostępniać kod źródłowy, ale nie chcą używać GPL. Nie mogąc radykalnie zmienić GPL, mają zawsze możliwość dostarczenia własnej licencji i zatwierdzenia jej przez organizację.

Podczas gdy FSF i OSI pomagają sobie nawzajem, to jednak nie są tym samym. FSF używa określonej licencji i dostarcza oprogramowanie na tej licencji. OSI szuka wsparcia dla wszystkich licencji otwartego oprogramowania, włącznie z FSF. Podstawowe argumenty za tworzeniem wolnodostępnego kodu źródłowego czasem różni oba ruchy, ale już sam fakt, że dwie odmienne ideologicznie grupy pracują na rzecz jednego celu daje wiarę w osiągnięcie tych celów.

Rozdział 2 Pomoc

Często przychodzi chwila, kiedy potrzebujesz pomocy z jakąś komendą, konfiguracją programu czy sprawieniem, żeby urządzenie zaczęło działać. Może chcesz poprostu lepiej zrozumieć daną komendę, albo zobaczyć jakie są inne opcje. Na szczęście jest kilka sposobów na uzyskanie pomocy. Kiedy instalujesz Slackware masz opcję instalacji paczek z serii "F" która zawiera FAQ i HOWTO. Programy również dostarczają pomocy o ich opcjach, plikach konfiguracyjnych i użytkowaniu.

2.1 System Pomocy

2.1.1 `man`

Polecenie `man` (skrót od “manual”) jest tradycyjną formą dokumentacji online dla systemów Unix i Linux. Zawartość specjalnie sformatowanych plików “man pages” jest pisana dla wiekszości poleceń i rozprowadzana wraz z programem. Wykonując `man jakaś-komenda` wyświetli się strona manuala dla wybranego polecenia, w naszym przykładzie będzie to wymyślony program `jakaś-komenda`.

Jak można się domyślać, liczba stron manuala szybko wzrasta. W efekcie korzystanie z tej metody pomocy może stać się kłopotliwe i bardzo zawiłe nawet dla zaawansowanego użytkownika. Dlatego tez strony man pogrupowane są ponumerowane sekcje. System jest używany od dawna, wystarczająco długo by komendy i polecenia a nawet biblioteki programów odsyłały do własnych sekcji stron manuala.

Na przykład:

Można spotkać odwołanie do `man`(1). Numer informuje, że opis polecenia “`man`” jest w sekcji pierwszej (komendy użytkownika); można dokładnie określić “rejon poszukiwań” wybierając pożdaną sekcję “man”: `mam 1 man`. Określanie sekcji gdzie man powinien szukać jest przydatne gdy kilka rzeczy ma tę samą nazwę.

Tabela 2-1. Sekcje stron Man

Sekcja	Zawartość
Sekcja 1	komendy użytkownika (tylko wprowadzenie)
Sekcja 2	wywołania systemowe
Sekcja 3	wywołania bibliotek C
Sekcja 4	urządzenia (np., `hd`, `sd`)
Sekcja 5	formaty plików i protokoły (np., wtmp, `/etc/passwd`, nfs)
Sekcja6	gry (tylko wprowadznie)
Sekcja 7	konwencje, makro pakiety(macro packages, etc.) (np. nroff, ascii)
Sekcja 8	administracja systemem (tylko wprowadzenie)

Dodatkowe narzędzia wspomagające używanie polecenia `man`(1) to i . Ich zadaniem jest ułatwienie odszukania informacji w systemie man.

Polecenie `whatis` daje bardzo zwięzły opis poleceń systemowych; coś w stylu kieszonkowego słownika komend.

Przykład:

```
% whatis whatis
whatis (1)  - search the whatis database for complete words
```

Komenda `apropos` używana jest do szukania stron man zawierających podane słowo. Jeśli szukasz dalszych informacji na temat powyższych poleceń, przeczytaj ich strony manuala.

Przykład:

```
% apropos wav
cdda2wav    (1)  - a sampling utility that dumps CD audio data into wav sound files
netwave_cs  (4)  - Xircom Creditcard Netwave device driver
oggdec      (1)  - simple decoder, Ogg Vorbis file to PCM audio file (WAV or RAW)
wavelan     (4)  - AT&T GIS WaveLAN ISA device driver
wavelan_cs  (4)  - AT&T GIS WaveLAN PCMCIA device driver
wvlan_cs    (4)  - Lucent WaveLAN/IEEE 802.11 device driver
```

Jeśli szukasz dalszych informacji na temat powyższych komend, przeczytaj ich strony manuala.

2.1.2 Katalog `/usr/doc`

Źródła większości paczek, które zostaną zbudowane dostarczane są ze swoistą dokumentacją: README, intsrukcjami użytkowania, plikami licencji itp. Każda dokumentacja dostarczona ze źródłami jest dołączana i instalowana w systemie w katalogu `/usr/doc`. Każdy program (zazwyczaj) instaluje własną dokumentacje według wzoru:

`/usr/doc/$program-$version`

Gdzie *`$program`* jest nazwą programu, o którym chcesz poczytać, a *`$version`* jest odpowiednią wersją paczki progamu zainstalowanego w systemie.

Na przykład, aby odczytac dokumentację dla komendy `man`(1) należy wejść do katalogu:

```
% cd /usr/doc/man-$version
```

Jeżeli manual nie da odpowiedzi na pożądane pytania, `/usr/doc` to następne miejsce do którego powinno się zajrzeć.

2.1.3 HOWTOs i mini-HOWTOs

Pliki te są dokładnie tym o czym mówi ich nazwa - dokumentami i przewodnikami opisującymi jak coś zrobić. Gdy została zainstalowana kolekcja HOWTO, będzie ona znajdować się w `/usr/doc/Linux-HOWTOs` Dokumenty mini-HOWTO zostaną umieszczone natomiast w `/usr/doc/Linux-mini-HOWTOs`.

W tej serii pakietów załączana jest również kolekcja FAQs, co jest akronimem od:

*F*requently

*A*sked

*Q*uestions

Doktumenty te są napisane w formie “pytań i odpowiedzi”. FAQs mogą być często bardzo przydatnym miejscem poszukiwań “szybkiego rozwiązania” problemu. Jeśli zdecydujesz się na instalację FAQ (podczas procesu intalacyjnego systemu), zostaną umieszczone w kataliogu `/usr/doc/Linux-FAQs`.

Warto się z nimi zapoznać szczególnie w przypadku, gdy nie do końca wiadomo jak coś zrobić. Zakres tematyki poruszany w tych dokumentach jest zaskakujący i godny uwagi. Naprawdę dobra rzecz :)

2.2 Pomoc Online

Dodatkiem do dokumentacji dostarczanej i instalowanej wraz z systemem, jest rozległa baza zasobów sieciowych dostępnych dla Ciebie do nauki.

2.2.1 Oficjalna witryna web i forum pomocy

Oficjalna strona Slackware

Oficjalna witryna slackware jest czasami zdezakutalizowana, ale ciągle zawiera najświeższe wiadomości o ostatniej wersji systemu. Utrzymywanie forum zaczynało być zbyt pracochłonne, więc Pat zamknął je. Starą jego wersję wraz z możliwością przeszukiwania archiwum można znaleźć pod adresem http: http://www.userlocal.com/phorum/.

Po tym jak zamknięto forum na http://slackware.com, kilka innych witryn zaproponowało wsparcie dla Slackware. Po dokładnym przemyslaniu Pat wybrał www.linuxquestions.org jako oficjalne forum Slackware Linux.

2.2.2 Wsparcie poprzez E-mail

Każdy kto zakupi oficjalną wersje CD jest upoważniony do darmowego wsparcia poprzez email w dziedzinie instalacji systemu. Mówi się, miej to na uwadze, że my, developerzy(i większość uztykowników) Slackware należymy do “starej szkoły”. Oznacza to, że preferujemy pomagać tym którzy są szczerze zainteresowani i chętni pomagać sobie nawzajem w przyszłości. Zawsze robimy co w naszej mocy by pomóc każdemu, kto napisze do nas z pytaniem o wsparcie. Jednakże proszę sparwdz dokumentację i zasoby internetu (szczególnie FAQ i fora wymienione poniżej) przed napisaniem do nas. Możesz w ten sposób uzyskać szybszą odpowiedź. Dzięki temu będziemy mogli szybciej udzielić pomocy tym, którzy jej naprawdę potrzebują.

Adres email wsparcia technicznego to: `support@slackware.com`. Pozostałe adresy email i kontakty są wyszczególnione na stronie.

2.2.2.1 Lista mailingowa projektu Slackware Linux

Posiadamy kilka list mailingowych dostępnych w formie kompendium i forów. Oto jak z nich korzystać.

Aby zapisać się na listę, napisz:

`majordomo@slackware.com`

z wyrażeniem “`subscirbe [nazwa listy]`” w treści wiadomośći. Nazwy list podane są niżej.

Archiwum listy mailingowej można znaleźć na stronie Slackware:

`http://slackware.com/lists/archive/`

`slackware-announce`

> Lista `slackware-announce` służy do ogłaszania informacji o nowych wersjach, głównych aktualizacjach .

`slackware-security`

> Lista `slackware-security` używana jest do ogłaszeń dotyczących zagadnień bezpieczeństwa. Informacje o exploitach lub innych wrażliwych kwestiach dotyczących bezpośrednio systemu będą natychmiast publikowane na liście.

Listy te dostepne są także w formie kompendiów(wydań). Oznacza to że dostaniesz jedną skondensowaną

wiadomość dziennie zamiast kilku. Od czasu gdy lista nie umożliwia użytkownikom pisania nowych postów i cechuje się małym ruchem, większość użytkowników wybiera właśnie formę kompendium. Nadal są one dostępne. By je otrzymywać zapisz się na `slackware-announce-digest` lub `slackware-security-digest`.

2.2.3 Nieoficjlane witryny i fora internetowe

2.2.3.1 Witryny

Google

> Mistrz wśród wyszukiwarek. Pozwala dotrzeć szybko do samego jądra interesującego zagadnienia. Nie ma sobie równych.

Google:Linux

> szukanie ukierunkowane na Linuksa

Google:BSD

> Szukanie ukierunkowane na BSD. Slackware jest typowym systemem uniksopodobnym. Prawie 100% informacji na temat Uniksa odnosi się również do Slacka. Niejednokrotnie przeszukiwanie ukierunkowane na BSD owocuje w znajdowaniu informacji bardzo “technicznej”.

Google:Groups

> Przeszukiwanie niezliczonych ilościach wiadomości pochodzących z grup dyskusyjnych.

http://userlocal.com

> Wirtualna skarbnica wiedzy, dobre porady doświadczonych użytkowników i ciekawe artykuły. Często pierwsze miejsce gdzie usłyszysz o nowych rozwiązaniach w świecie Slackware.

2.2.3.2 Zasoby www

linuxquestions.org

> Oficjanlnie forum internetowe dla użytkowników Slackware.

LinuxISO.org Slackware Forum

> “Miejsce gdzie można pobranć i uzyskać pomoc nt. Linuksa.”

alt.os.linux.slackware FAQ

> Kolejne FAQ

2.2.3.3 Grupy dyskusyjne (NNTP)

Usnet(grupy dyskusyjne) był przez długi czas miejscem spotkań geeków gdzie nawzajem sobie pomagali. Jest kilka grup poświęconych Slackware. Spotyka się tam osoby z pokaźną wiedzą.

`alt.os.linux.slackware`

`alt.os.linux.slackware`, lepiej znane jako aols (nie związne z AOL®!) jest jednym z bardziej aktywnych miejsc gdzie można znaleźć pomoc techniczną nt. Slacka. Jak na każdej grupie usnetowej, znajdują się tam również niemili goście (trole). Mogą oni uprzykrzać życie ciągłymi kłotniami. Nauka ignorowania troli i rozpoznawnia naprawde pomocnych osób jest kluczem do większości tego typu zasobów.

Rozdział 3 Instalacja

Przed rozpoczęciem używania Slackware Linux, należy pozyskać go i zainstalować. Pozyskanie Slackware jest proste - wystarczy zakupić go lub ściągnąć z internetu (za darmo). Instalacja również jest łatwa, o ile posiadasz podstawową wiedzę o swoim komputerze oraz chcesz nauczyć się kilku nowych rzeczy. Sama instalacja jest procesem przeprowadzanym krok po kroku, dzięki czemu będziesz mógł szybko przejść do użytkowania systemu. Tak naprawdę instalacja Slackware jest najkrótszą spośród wszystkich instalacji “pełnych” dystrybucji Linuksa.

3.1 Pozyskiwanie Slackware

3.1.1 Oficjalny Zestaw płyt CD Box

Oficjalny zestaw Slackware Linux CD dostępny jest od Slackware, Inc. Zestaw CD składa się z 4 dysków CD. Pierwszy zawiera całe oprogramowanie potrzebne do instalacji podstawowego serwera oraz systemu X Window. Druga płyta jest typu “live cd” - jest to bootowalna płytka, która tymczasowo instaluje się do RAM'u i pozwala na odzyskanie danych bądź systemu. Zawiera ona również kilka pakietów takich jak środowiska KDE czy GNOME. Do tego krążka jest też załączonych kilka innych ciekawych rzeczy: patrz katalog “extra”). Trzecia i czwarta płyta zawiera kod źródłowy wszystkiego związanego ze Slackware, między innymi kod źródłowy tej książki.

Można również kupić pudełko z zestawem zawierającym 4 płyty i kopię tej książki jak również wiele innych rzeczy związanych ze Slackware, by móc dumnie pokazać je światu. Realizowane są również zamówienia po obniżonych kosztach.

Polecana metoda zakupu Slackware to sklep internetowy.

http://store.slackware.com

Możesz również zadzwonić lub napisać e-mail by złożyć zamówienie.

Tabela 3-1. Slackware Linux, Inc. Informacja kontaktowa

Metoda	Szczegóły kontaktu
Telefon	1-(925) 674-0783
Strona web	http://store.slackware.com
Email	orders@slackware.com
Poczta	1164 Claremont Drive, Brentwood, CA 94513

3.1.2 Poprzez internet

Slackware Linux jest również darmowo dostępny poprzez Internet. Możesz przesłać (e-mailem) swoje pytania, lecz pierwszeństwo w uzyskaniu odpowiedzi mają osoby, które kupiły oficjalny Zestaw Slackware CD. Dostajemy dużo e-maili, a nasz czas jest raczej ograniczony. Przed wysłanieniem pytania rozważ przeczytanie

Rozdział 2pomocy.

Oficjalna strona Projektu Slackware Linux znajduje się pod adresem:

http://www.slackware.com/

Lokalizacja głównego serwera FTP Slackware Linux:

ftp://ftp.slackware.com/pub/slackware/

Miej na uwadze, że pomimo tego iż nasz serwer dostępny jest dla każdego, nie dysponujemy nieograniczoną przepustowością. Proszę, rozważ użycie serwera (mirrora) znajdującego się blisko ciebie. Niepełna lista serwerów (mirrorów) znajduje się na stronie: http://www.slackware.com/getslack.

3.2 Wymagania Systemowe

Dla łatwej instalacji Slackware wymagane są:

Tabela 3-2. Wymagania systemowe

Sprzęt	Wymagany
Procesor	586
RAM	32 MB
Przestrzeń dyskowa	1GB
Napęd	4x CD-ROM

Jeżeli posiadasz bootowalną płytkę CD, to prawdopodobnie stacja dyskietek nie będzie wymagana. Oczywiście w wypadku braku napędu CD-ROM niezbędny będzie napęd dyskietek, aby móc wykonać instalację przez sieć. Przy instalacji NFS niezbędna będzie też karta sieciowa. Zobacz sekcję nazwaną NFS aby uzyskać więcej informacji.

Wymagana ilość miejsca na dysku to mały podstęp ;-). Rekomendacja 1GB jest zwykle bezpieczna, ale jeżeli wykonasz pełną instalację to bedziesz potrzebowć około dwóch gigabajtów wolnego miejsca na dysku, plus dodatkowe miejsce na pliki osobiste. Większość użytkowników nie wykonuje pełnej instalacji. Faktem też jest, że dla wielu wystarcza ok. 100MB przestrzeni dyskowej.

Slackware może zostać zainstalowany na systemach z mniejszą ilością pamięci RAM i mniej pojemnych dyskach twardych, ale dokonanie tego wymaga trochę pracy. Jeżeli jesteś na to gotowy, pomocne rady znadziesz w pliku `LOWMEM.TXT` dostarczonym razem z dystrybucją.

3.2.1 Grupy programów

Dla uproszczenia sprawy Slackware został historycznie podzielony na grupy/zbiory programów. Niekiedy nazywane “zbiorami dyskowymi” - ponieważ były zaprojektowane do instalacji z dyskietek - Grupy programów są aktualnie używane głównie w celu podzielenia pakietów Slackware na kategorie. W dzisiejszych czasach instalacja z dyskietek nie jest już możliwa.

Poniżej przedstawiony został krótki opis każdej grupy programów.

Tabela 3-3. Grupy programów

Grupy	Zawartość
A	System podstawowy. Zawiera wystarczającą ilość oprogramowania potrzebną do

Grupy	Zawartość
	uruchomienia i używania systemu operacyjnego, wraz z edytorem tekstowym i podstawowymi programami do komunikacji.
AP	Różne aplikacje, które nie wymagają X Window.
D	Narzędzia do rozwijania oprogramowania. Kompilatory, debuggery, interpretery oraz strony manuala wszystkie one znajdują się w tej grupie.
E	GNU Emacs.
F	Pliki FAQ, HOWTO i pozostała dokumentacja.
GNOME	Środowisko GNOME.
K	Źródła jądra Linuxa.
KDE	The K Desktop Environment - środowisko dla X'ów, podobne wizualnie i funkcjonalnie do MacOS i Windows. Biblioteka Qt, której wymaga KDE, znajduje się również w tej grupie.
KDEI	Pakiety dla KDE umożliwiające internacjonalizację środowiska.
L	Biblioteki. Dynamicznie linkowane biblioteki potrzebne wielu programom.
N	Programy sieciowe. Demony, programy pocztowe, telnet, programy do przeglądania grup dyskusyjnych itd.
T	system formatowania dokumentów teTeX.
TCL	Narzędzia języków skryptowych. Tk, TclX oraz TkDesk.
X	podstawowy system X Window.
XAP	aplikacje dla X'ów, które nie są integralną częścią środowisk okienkowych (np. Ghostscript i Netscape).
Y	Gry konsolowe z BSD.

3.2.2 Metody instalacji

3.2.2.1 Z dyskietki

Kiedyś możliwe było zainstalowanie całego Slackware z dyskietek, jednakże rosnące rozmiary pakietów oprogramowania (zwłaszcza niektórych programów) doprowadziły do porzucenia tej metody. Do wersji 7.1 Slackware częściowa instalacja z dyskietek była jeszcze możliwa. Grupy A i N mogły być niemal całkowicie zainstalowane, dostarczając podstawowego systemu z którego można było zainstalować pozostałą część dystrybucji. Użytkownikom rozważającym instalację z dyskietek (np. na starsze platformy) typowo rekomenduje się by znaleźć inny sposób lub użyć starszego wydania. Slackware 4.0 jest wciąż bardzo chętnie używany do tego celu, podobnie jak wersja 7.0.

Zauważ, że dyskietki są wciąż wymagane do instalacji z CD-ROMu, jeżeli nie posiadasz bootowalnej płytki CD, jak również w wypadku instalacji z użyciem NFS.

3.2.2.2 CD-ROM

Jeżeli posiadasz bootowalną płytę CD, dostępną w oficjalnym zestawie publikowanym przez Slackware Inc., (patrz sekcja: Pozyskiwanie Slackware), instalacja oparta o CD będzie trochę łatwiejszym wyjściem. Jeżeli nie, będziesz musiał bootować system z dyskietek. Również, gdy posiadasz niestandardowy sprzęt, dla którego bootowanie jądra z CD jest problematyczne, możesz potrzebować specjalistycznych dyskietek.

Od wersji 8.1 Slackware używana jest nowa metoda tworzenia bootowalnych płytek CD, które nie współpracują dobrze z pewnymi rodzajami chipów BIOSu (w dzisiejszych czasach większość dystrybucji

Linuksa cierpi z tego powodu). W tym wypadku rekomendujemy bootowanie z dyskietki.

Sekcja 3.2.3 oraz Sekcja 3.2.5 dostarczą informacji pomocnych przy wyborze oraz tworzeniu dyskietek startowych, jeżeli zajdzie konieczność utworzenia takowych.

3.2.2.3 NFS

NFS (the Network File System - Sieciowy system plików) to sposób na utworzenie sytemu plików dostępnego dla maszyn łączących się zdalnie. Instalacja NFS pozwala na zainstalowanie Slackware z innego komputera w sieci. Maszyna, z której odbyć ma się instalacja, musi być skonfigurowana do wyeksportowania drzewa dystrybucji do komputera docelowego. Wymaga to oczywiście pewnej wiedzy o NFS. Kwestia została omówiona w sekcji NFS rozdziału Sekcja 5.6.

Możliwe jest wykonanie instalacji NFS przy użyciu metod takich jak: PILP (przez port równoległy), SLIP, i PPP (jednak nie przez połączenie modemowe). Jakkolwiek, rekomendujemy użycie karty sieciowej jeżeli jest to możliwe. W końcu instalowanie systemu operacyjnego przez port drukarki to bardzo, bardzo powolny proces.

3.2.3 Dyski Bootowalny - Boot disks

Boot disk to dyskietka, z której następuje uruchamianie systemu, gdy rozpoczynamy instalację. Zawiera skompresowany obraz jądra używanego do kontroli sprzętu podczas instalacji. Dlatego jest tak istotny (chyba, że bootujesz z CD, tak jak to zostało omówione w podrozdziale CD-ROM). Dyski bootowalne są umieszczone w katalogu `bootdisks/` w drzewie dystrybucji.

Rodzajów boot disków Slackware jest więcej niż przypuszczasz (około 16). Kompletna ich lista, z opisem każdego, dostępna jest w pliku `bootdisks/README.TXT`. Jakkolwiek, większość ludzi może użyć obrazu `bare.i` (dla urządzeń IDE) lub `scsi.s` (dla urządzeń SCSI).

Zobacz Sekcja 3.2.6, by uzyskać informacje o tworzeniu dysku bootowalnego z obrazu.

Po uruchomieniu komputera z dyskietki zostaniesz poproszony o włożenie dysku nazwanego "root disk". Sugerujemy wykonanie tego polecenia, by możliwe stało się przejście do dalszej części procesu instalacji.

3.2.4 Root Disk

Root Disk zawiera program instalacyjny i system plików używany podczas instalacji. One także są wymagane przy instalacji. Obrazy root disk'ów są umieszczone w katalogu rootdisks w drzewie katalogów dystrybucji. Będziesz musiał utworzyć dwie dyskietki z obrazów: `install.1` oraz `install.2`. We wspomnianym katalogu znajdziesz także pliki `network.dsk`, `pcmcia.dsk`, `rescue.dsk`, oraz plik `sbootmgr.dsk`.

3.2.5 Dodatkowe Dyski (supplemental disk)

Dysk dodatkowy jest potrzebny gdy wykonujesz instalację NFS lub instalujesz dystrybucję na platformie z urządzeniami PCMCIA. Dyski dodatkowe znajdują się w katalogu rootdsks w drzewie katalogów dystrybucji. Ich nazwy to: `network.dsk` oraz `pcmcia.dsk`. Ostatnimi czasy dodane zostały inne obrazy tj.: `rescue.dsk` a także `sbootmgr.dsk`. Dysk 'ratunkowy' (rescue disk) to mały obraz dyskietki (root disk), który działa na 4MB RAM-dysku. Zawiera podstawowe narzędzia sieciowe oraz edytor vi do dokonywania szybkich poprawek na uszkodzonych w jakiś sposób systemach. Dysk `sbootmgr.dsk` jest używany do bootowania z innych urządzeń. Użyj go jeżeli twój napęd CD-ROM nie chce uruchomić Slackware z płyty. Zostaniesz zapytany o wiele rzeczy odnośnie bootowania. Może to być wygodny sposób na obejście felernych BIOS'ów.

Root disk, gdy zostanie załadowany, poinformuje cię na temat użycia dysków dodatkowych.

3.2.6 Tworzenie dysków

Gdy wybierzesz już obraz dysku bootowalnego, musisz umieścić go na dyskietce. Proces ten jest zależny od używanego systemu. W wypadku Linuksa (lub jakiegoś innego systemu Uniksowego) potrzebne będzie polecenie `dd`(1). Zakładając, że `bare.i` to twój plik z obrazem, zaś napęd dyskietek to `/dev/fd0`, polecenie utworzenia dyskietki z obrazu będzie wyglądało tak:

```
% dd if=bare.i of=/dev/fd0
```

W przypadku systemu Microsoftu potrzebny będzie program `RAWRITE.EXE`, załączony do drzewa dystrybucji w tym samym katalogu co obrazy dyskietek. Ponownie zakładając, że `bare.i` to Twój plik z obrazem a stacja dyskietek to A:, w trybie MS-DOS wpisz:

```
C:\ rawrite a: bare.i
```

3.3 Partycjonowanie

Po uruchomieniu komputera z preferowanego nośnika, należy podzielić dysk twardy na partycje. Partycja dyskowa jest to miejsce gdzie system plików Linuksa będzie utworzony i gdzie Slackware będzie zainstalowany. Rekomendujemy utworzenie co najmniej dwóch partycji: jedna dla głównego (root) systemu plików(`/`) i druga dla wymiany (swap space).

Po załadowaniu root disku zostaniesz zapytany o login. Zaloguj się jako root (nie ma hasła). W powłoce uruchom `cfdisk`(8) lub `fdisk`(8). Program `cfdisk`(8) dostarcza bardziej przyjaznego dla użytkownika interfejsu niż `fdisk`, ale brakuje mu pewnej funkcjonalności. Poniżej wyjaśniamy pokrótce użycie programu `fdisk`.

Uruchom `fdisk` dla twojego dysku twardego. W Linuksie nie ma liter dla określenia urządzeń, każde za to jest reprezentowane przez plik. Pierwszy dysk IDE (primary master) to `/dev/hda`, drugi (primary slave) to `/dev/hdb` itd. Dyski SCSI oznaczane są analogicznie ale w formie `/dev/sdX`. Należy uruchomić zatem `fdisk` i przekazać mu jako parametr żądany dysk.

```
# fdisk /dev/hda
```

Jak każdy dobry program Uniksowy, `fdisk` oferuje linię poleceń (chciałoby się dostać to menu, he?). Pierwszą rzeczą, którą należy zrobić jest sprawdzenie aktualnych partycji. Wykonujemy to poprzez wpisanie **p** do linii poleceń programu `fdisk`:

```
Command (m for help): p
```

To wyświetli całą masę informacji o twoich aktualnych partycjach. Większość ludzi wybiera urządzenie, na którym będzie instalować system, zaś następnie usuwa wszystkie istniejące na nim partycje by utworzyć miejsce dla partycji Linuksowych.

WAŻNE: ZACHOWAJ DANE KTÓRYCH NIE CHCESZ STRACIĆ, ZANIM USUNIESZ PARTYCJĘ NA KTÓREJ SIĘ ONE ZNAJDUJĄ.

Nie ma prostej metody by odzyskać dane po usunięciu partycji, więc przed manipulacjami związanymi z partycjami twórz kopie bezpieczeństwa.

Przyglądając się tablicy partycji zobaczyć można następujące informacje: rozmiar partycji oraz jej typ. Jest tam też więcej rzeczy, ale nie martw się tym teraz. Zamierzamy usunąć wszystkie partycje na tym dysku, by utworzyć Linuksowe. Wywołujemy polecenie **d** by tego dokonać:

```
Command (m for help): d
Partition number (1-4): 1
```

Proces ten powinien być powtarzany dla każdej partycji. Po dokonaniu tego jesteśmy gotowi do utworzenia partycji Linuksowych. Zdecydowaliśmy się utworzyć jedną partycję dla głównego (root) systemu plików i jedną przeznaczoną do wymiany (swap). Nie jest istotne, że partycjonowanie jest tematem wojen oraz, że większość użytkowników wyjawi ci 'jedyny i najlepszy' sposób podzielenia dysku na partycje. Nasza rada to utworzenie dwóch partycji na początek. Jedną dla głównego systemu plików `/` i jedną dla wymiany. Później nauczysz się partycjonować dysk tak, aby dopasować go do swojego systemu.

Ja używam dwóch podstawowych schematów partycjonowania. Pierwszy jest dla komputera domowego. Tworzę 4 partycje, `/`, `/home`, `/usr/local`, oraz swap. To pozwala mi na przeinstalowanie lub upgrade'owanie całego systemu znajdującego sie pod `/` bez wyrzucania wszystkich plików z danymi (`/home`) i skopiowanymi przeze mnie aplikacjami (`/usr/local`). Dla serwerów często zamiast partycji dla `/usr/local` tworzę ją dla `/var`. Wiele różnych serwerów przechowuje informacje na tej partycji, a trzymanie ich oddzielnie od `/` przynosi pewne korzyści związane z wydajnością. Jak na razie przyjmujemy, że mamy dwie partycje: `/` oraz swap (wymiany).

Teraz utwórzmy partycję poleceniem **n** :

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4):1
First cylinder (0-1060, default 0):0
 Last cylinder or +size or +sizeM or +sizeK (0-1060, default 1060):+64M
```

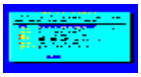
Musisz się upewnić, że tworzysz partycje podstawowe (primary partition). Pierwsza z nich będzie partycją wymiany. Mówimy fdisk'owi by utworzył partycję numer 1 jako partycję podstawową (primary partition). Początek partycji ustawiany na cylinder 0, a dla ostatniego cylindra wpisujemy +64M. Spowoduje to utworzenie 64 megabajtowej partycji wymiany. (Rozmiar tej partycji zależy głównie od ilości pamięci RAM jaką dysponujesz. Ogólnie rozsądnie jest przyjąć podwojoną ilość pamięci RAM jako rozmiar partycji wymiany). Następnym krokiem jest utworzenie partycji podstawowej numer 2, rozpoczynającej się na pierwszym dostępnym cylindrze i zajmującą całą dostępną przestrzeń dyskową.

```
Command (m for help):n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4):2
First cylinder (124-1060, default 124):124
Last cylinder or +size or +sizeM or +sizeK (124-1060, default 1060):1060
```

Już prawie skończyliśmy. Musimy zmienić typ pierwszej partycji na 82 (Linux swap). Wpisz **t** by zmienić typ, wybierz pierwszą partycję, oraz wpisz `82`. Przed zapisaniem zmian na dysku powinieneś spojrzeć na nową tablicę partycji po raz ostatni. Użyj polecenia **p** w `fdisk`, by wyświetlić tablicę partycji. Jeżeli wszystko wygląda dobrze, wpisz **w**, by zapisac zmiany na dysk oraz wyjść z programu.

3.4 Program `setup`

Gdy już utworzyłeś partycje, jesteś gotowy do zainstalowania Slackware. Następnym krokiem w procesie instalacji jest uruchomienie programu `setup`(8). Aby to wykonać, wpisz po prostu `setup` w linii poleceń shell'a. `setup` to program oparty o system menu, stworzony do instalacji pakietów i konfiguracji systemu.



Proces setup wygląda mniej więcej tak: przechodzisz przez każdą opcję w programie `setup` w kolejności, w jakiej są wyświetlone. (Oczywiście możesz to robić w niemalże dowolnym porządku, jednak szanse, że to zadziała, są nikłe). Elementy menu wybiera się za pomocą kursorów góra-dół, przyciski “Okay” oraz “Cancel” wybiera się używając strzałek lewo-prawo. Można też inaczej: każda pozycja menu posiada odpowiadający jej klawisz, który jest podświetlony w nazwie opcji. Opcje, które można zaznaczać i odznaczać (te wskazywane przez `[X]`) wybiera się spacją.

Oczywiście wszystko to jest opisane w sekcji “help” programu `setup`, ale uważamy, że skoro użytkownicy zapłacili już za tę książkę to powinni w niej otrzymać kompletne informacje.

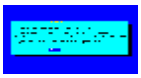
3.4.1 HELP

Jeżeli instalujesz Slackware po raz pierwszy, możesz chcieć rzucić okiem na ekran pomocy (help). Ujrzysz tam opis każdej części `setup` (coś podobnego do tego co teraz czytasz, jednak bardziej zwięźle) oraz instrukcje, jak przejść do końca instalacji.

3.4.2 KEYMAP

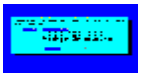
Jeżeli potrzebujesz mapy klawiszy innej niż amerykańska “qwerty”, powienieneś przejrzeć tę sekcję. Będziesz mógł tam dokonać wyboru spośród wielu dostępnych układów klawiszy.

3.4.3 ADDSWAP



Jeżeli utworzyłeś partycję wymiany (patrz Sekcja 3.3), ta sekcja pozwoli ci na uaktywnienie jej. Dokonana zostanie autodetekcja oraz wyświetlone zostaną partycje wymiany dostępne na twoim dysku, byś mógł wybrać jedną do sformatowania i uaktywnienia.

3.4.4 TARGET



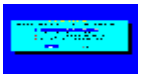
Sekcja “target” służy do formatowania i mapowania innych partycji (nie wymiany) na punkty montażu w systemie plików. Wyświetlona zostanie lista partycji na twoim dysku twardym. Dla każdej partycji będziesz mógł zdecydować czy formatować ją, czy też nie. W zależności od użytego jądra będziesz mógł wybrać

pomiędzy systemami plików: reiserfs (domyślnie), ext3, ext2, jfs oraz xfs. Większość użytkowników wybiera reiserfs lub ext3. W niedalekiej przyszłości być może pojawi się możliwość wyboru reiserfs4.

Pierwsza opcja w sekcji “target” to wybór na jakiej partycji ma być zainstalowany główny (root) system plików (`/`). Po tym będziesz mógł mapować inne partycje na systemy plików jakie wybierzesz. (Na przykład możesz chcieć użyć trzeciej partycji, powiedzmy `/dev/hda3`, jako domowy (/home) system plików. To oczywiście tylko przykład, mapuj partycje jak uważasz).

3.4.5 SOURCE

Sekcja “source” to miejsce, w którym wybierasz nośnik, z którego instalujesz Slackware. Aktualnie można wybrać spośród czterech: CD-ROM, dysk twardy, NFS oraz katalog uprzednio zamontowany.



Wybór CD-ROM uaktywnia instalację z płytki. Oferuje skanowanie napędu lub wyświetlenie listy, z której wybierzesz jego typ ręcznie. Upewnij się, że w napędzie znajduje się płyta Slackware przed rozpoczęciem skanowania.

W tej sekcji zapytany zostaniesz o informacje odnośnie sieci oraz serwera NFS. Serwer ten musi być przygotowany wcześniej. Zauważ również, że nie możesz użyć nazwy hosta tylko adresu IP dla Twojej maszyny i serwera NFS (na dysku nie ma bowiem programu, który rozpoznałby IP po nazwie (name resolver). Oczywiście musisz użyć dyskietki z obrazu `network.dsk`, aby dodać obsługę twojej karty sieciowej (network controller).

Uprzednio zamontowany katalog dostarcza najwięcej możliwości. Możesz użyć tej metody do instalacji z takich rzeczy jak dyski Jaz, NFS poprzez PLIP, oraz systemy plików FAT. Zamontuj system plików w odpowiednim miejscu przed uruchomieniem setup'u, potem wybierz tę lokację w tej części instalacji.

3.4.6 SELECT

Opcja select pozwala ci na wybranie grup programów, które chcesz zainstalować. Te grupy są opisane w Sekcja 3.2.1. Zauważ proszę, że musisz zainstalować grupę A by mieć działający system podstawowy. Wszystkie inne serie są opcjonalne.

3.4.7 INSTALL

Zakładając, że masz już za sobą “target”, “soruce” oraz “select”, opcja `install` pozwoli ci na wybranie pakietów z wcześniej zaznaczonych grup programów; jeżeli tak nie jest, to zostaniesz poproszony o cofnięcie się i dokończenie wcześniej wspomnianych sekcji setup. Opcja pozwala na wybranie spośród sześciu metod instalacji: `full`, `newbie`, `menu`, `expert`, `custom` oraz `tag path`.

Wybór opcji `full` spowoduje zainstalowanie każdego pakietu z wszystkich grup programów, które wybrałeś w sekcji “select”. W dalszym toku instalacji nie będziesz już o nic pytany. Jest to najłatwiejsza metoda instalacji, gdyż nie musisz podejmować żadnych decyzji co do pakietów które mają zostać zainstalowane. Zabiera ona jednak najwięcej wolnego miejsca na dysku twardym.

Następną opcją jest `newbie`. W tym wypadku zainstalowane zostaną wszystkie wymagane pakiety z wybranych

grup. Dla wszystkich innych zostanie wyświetlone pytanie, na które można udzielić odpowiedzi “Yes” - tak, “No” - nie lub “Skip”. Dwie pierwsze odpowiedzi są oczywiste, natomiast Skip spowoduje przejście do następnej grupy programów. Dodatkowo dla każdego pakietu zostanie wyświetlony jego opis oraz wymagana ilość wolnej przestrzeni dyskowej, aby pomóc ci zdecydować czy go potrzebujesz. Polecamy ten typ instalacji nowym użytkownikom, ponieważ daje on pewność, że wszystkie wymagne pakiety zostaną zainstalowane. Jakkolwiek jest ona dosyć powolna z powodu wielu pytań.

Opcja `Menu` jest szybszą i bardziej zaawansowaną wersją wcześniej omawianej opcji newbie. Dla każdej grupy programów wyświetlane jest menu, z którego należy wybrać wszystkie opcjonalne pakiety, które chcesz zainstalować. Pakiety wymagane nie są wyświetlane.

Dla najbardziej zaawansowanych użytkowników istnieje opcja `expert`. Pozwala ona na kompletną kontrolę nad tym jakie pakiety zostaną zainstalowane. Można odznaczyć nawet pakiety absolutnie wymagane, co w rezultacie spowoduje, że system będzie niesprawny. Z drugiej strony można kontrolować dokładnie wszystko, co będzie zainstalowane na komputerze. Poprostu zaznacz pakiety z każdej grupy, które chcesz zainstalować. Opcja ta nie jest zalecana dla początkujących, ponieważ bardzo łatwo tu o popełnienie błędu.

Opcje `custom` oraz `tag path` są przeznaczone również dla zaawansowanych użytkowników. Pozwalają one na instalację opartą o specjalne pliki, tzw. tag files, które tworzy się w drzewie dystrybucji. Jest to metoda użyteczna gdy należy zainstalować system równolegle na znacznej liczbie maszyn. Aby uzyskać więcej informacji na ten temat zobacz Sekcja 18.4.

Po wybraniu odpowiedniej metody instalacji zdarzy się jedna z kilku rzeczy. Jeżeli wybrana została opcja expert lub menu, wyświetlony zostanie ekran z menu pozwalający na wybranie pakietów do zainstalowania. Jeżeli wybrana została opcja full, pakiety automatycznie zostaną zainstalowane w docelonym miejscu. Opcja newbie spowoduje instalację do momentu napotkania pakietu opcjonalnego.

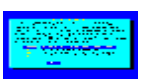
Zauważ, że może dojść do sytuacji wyczerpania miejsca na dysku podczas instalacji. Jeżeli wybranych zostało zbyt dużo pakietów w stosunku do wolnego miejsca na dysku, pojawią się problemy. Najbezpieczniej jest wybrać kilka pakietów i doinstalować resztę później, gdy będą potrzebne. Można tego łatwo dokonać przy pomocy narzędzi do administracji pakietami Slackware (ang. Slackware's package management tools). Aby uzyskać więcej informacji na ten temat zobacz Rozdział 18.

3.4.8 CONRysunek

Sekcja conRysunek pozwala na dokonanie podstawowej konfiguracji systemu, gdy pakiety zostały już zainstalowane. To co się pojawi w tej sekcji zależy w dużej mierze od tego co zostało zainstalowane. W każdym razie ujrzysz przynajmniej co następuje:

3.4.8.1 Kernel selection

W tym miejscu pojawi się pytanie o wybranie jądra do zainstalowania. Można wybrać je z boot disk'u który użyty został do instalacji, z Slackware CD-ROM lub z innej dyskietki, która (zawsze myślimy wprzód) została przygotowana wcześniej. Można również wybrać opcję Skip. W tym przypadku zostanie wybrane i zainstalowane domyślne jądro systemu.



3.4.8.2 Tworzenie boot disk'u

Tworzenie boot disk'u dla późniejszego użytku to dobry pomysł. W takim wypadku pojawi się opcja formatowania dyskietki i wybór spośród dwóch typów boot disk'u. Pierwszy rodzaj `simple` to poprostu zapisanie jądra na dyskietkę. Bardziej elastyczna (i polecana) jest opcja `lilo`, która spowoduje utworzenie boot

disk'u lilo. Zobacz LILO w Sekcja 7.1, aby uzyskać więcej informacji. Możesz też wybrać `continue`. W tym wypadku dysk nie zostanie utworzony.

3.4.8.3 Modem

Zostaniesz zapytany o informacje o modemie. Konkretnie o to, czy masz modem i jeżeli tak, to na którym porcie szeregowym jest on podłączony.

Kolejne podsekcje mogą lecz nie muszą się pojawić, w zależności od tego jakie pakiety zostały wybrane do instalacji.

3.4.8.4 Timezone

To całkiem proste. Pojawi się pytanie o strefę czasową, w której się znajdujesz. Jeżeli jest to czas Zulu, bardzo nam przykro - zostanie wyświetlona bardzo długa, posortowana alfabetycznie lista, a twoja strefa jest na samym końcu :)

3.4.8.5 Mysz

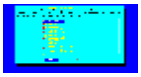
Ta podsekcja służy wyborowi typu myszy której używasz oraz zdecydowaniu, czy chcesz uaktywnić (podczas bootowania) obsługę myszy `gpm`(8) w konsoli.

3.4.8.6 Hardware clock

W tej podsekcji zapytany zostaniesz o to, czy zegar sprzętowy twojej maszyny jest ustawiony na Coordinated Universal Time (UTC lub GMT). Większość PC'tów nie jest, więc prawdopodobnie wybierzesz nie - No.

3.4.8.7 Font

Podsekcja font (czcionka) pozwala na wybór (z listy) czcionki używanej w konsoli.



3.4.8.8 LILO

W tym miejscu zostaniesz zapytany o instalację LILO (the LInux LOader; zobacz Sekcja 7.1 by uzyskać więcej informacji).

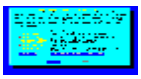
Jeżeli Slackware ma być jedynym systemem na twojej maszynie opcja `simple` jest odpowiednia dla tego celu. Jeżeli nie, to lepszym wyborem jest opcja `expert`. Zobacz rozdział Sekcja 7.3, by uzyskać więcej informacji na temat bootowania wielu systemów. Trzecia opcja `do not install`, nie jest zalecana o ile nie wiesz co dokładnie robisz i nie masz bardzo poważnego powodu by nie instalować LILO. Jeżeli wykonujesz instalację expert, pojawi się pytanie o miejsce umieszczenia LILO. Można tego dokonać w MBR (Master Boot Record) dysku twardego, w superbloku głównej (root) partycji Linuksa, lub na dyskietce.

3.4.8.9 Network

Podsekcja konfiguracji sieci aktualnie sprowadza się do wykonania polecenia `netconfig`. Zobacz Sekcja 5.1, by uzyskać więcej informacji.

3.4.8.10 Menedżer X Window

Ta podsekcja pozwoli ci na wybranie domyślnego menedżera okien dla X'ów. Zobacz Rozdział 6, by dowiedzieć się więcej na ich temat.



Bez względu na to, jakie pakiety zostały zainstalowane, ostatnią rzeczą którą wykona conRysunek będzie zapytanie o to czy chcesz ustawić hasło `root`'a. Z punktu widzenia bezpieczeństwa jest to ważna rzecz, jednak jak niemal wszystko w Slackware, tak i to zależy całkowicie od ciebie.

Rozdział 4 Konfiguracja Systemu

Zanim zaczniesz konfigurować bardziej zaawansowane części systemu, warto dowiedzieć się jak jest on zorganizowany i jakie polecenia będą używane do poszukiwania plików i programów. Dobrze również wiedzieć czy należy skompilować na własne potrzeby jądro i jakie kroki w tym celu należy podjąć. Ten rozdział zaznajomi cię z organizacją i konfiguracją (co odbywa się zwykle poprzez edycję plików) systemu. Zaznajomienie się z jego treścią da ci podstawy do konfiguracji bardziej zaawansowanych części systemu.

4.1 Ogólnie o Systemie

Ważne jest by zrozumieć jak Linux jest zbudowany przed zagłębieniem się różnorodne aspekty konfiguracji. System Linux jest w znaczącym stopniu różny od DOS'a, Windowsa i systemu Macintosh'a (za wyjątkiem bazowanego na Unix'ie Mac OS X), lecz te rozdziały pomogą ci oswoić się z jego budową i umożliwić łatwą jego konfigurację tak by sprostał twoim wymaganiom.

4.1.1 Wygląd systemu plików

Pierwszą zauważalną różnicą pomiędzy DOS'em lub Windows'em a Slackware Linux'em jest system plików. Informacja dla tych, którzy dopiero zaczynają: nie używamy liter by rozróżniać partycje. Pod Linux'em jest jeden główny katalog. Możesz go uważać za odpowiednik DOS'owego `C:`. Każda partycja w twoim systemie

jest montowana do (pod)katalogu w katalogu głównym.

Katalog główny nazywany jest z ang. root directory; jest on oznaczany jako pojedynczy slash (`/`). Ta koncepcja na pierwszy rzut oka wygląda dziwnie, jednak bardzo ułatwia życie gdy chcesz zwiększyć rozmiar dysku. Na przykład, powiedzmy, że kończy Ci się miejsce na partycji na której jest `/home`. Większość ludzi instaluje Slackware i tworzy jedną dużą partycję główną. Skoro więc partycja może być zamontowana w każdym katalogu, możesz poprostu pójść do sklepu, kupić nowy dysk i zamontować go do `/home`. Dodana zostanie przestrzeń dyskowa i to bez konieczności przenoszenia niczego.

Poniżej znajdziesz opisy głównych katalogów w Slackware.

`bin`

Najważniejsze programy użytkownika są przechowywane w tym miejscu. Jest to absolutne minimum programów wymaganych by system był używalny. Są tam rzeczy takie jak shell oraz polecenia do operowania na systemie plików (`ls`, `cp`, itp.). Katalog `/bin` właściwie nie zmienia się od momentu zakończenia instalacji. Jeżeli tak się dzieje zwykle są to uaktualnienia pakietów, które my dostarczamy.

`boot`

Pliki, które są używane przez LILO (Linux Loader). Ten katalog również jest modyfikowany w nieznacznym stopniu od momentu instalacji. W tym miejscu przechowywane jest jądro systemu poczynając od wersji 8.1 Slackware. Wcześniej było ono umieszczone w katalogu `/`, lecz powszechna praktyka bootowania kilku systemów (przy pomocy LILO) z jednej maszyny spowodowała przeniesienie jądra i związanych z nim plików do katalogu boot.

`dev`

Wszystko w Linux'ie jest traktowane jako plik, nawet urządzenia takie jak porty szeregowe, dyski twarde i skanery. Aby umożliwić do nich dostęp, specjalny plik nazwany węzłem urządzenia (device node) musi być obecny. Wszystkie takie pliki przechowywane są w katalogu `/dev`. Jest to powszechna praktyka w wielu systemach Unix'owych.

`etc`

W tym katalogu znajdują się pliki konfiguracyjne systemu. Poczynając od plików konfiguracji X Windows poprzez bazę danych użytkowników i kończąc na skryptach startowych. Administrator systemu powinien dokładnie poznać zawartość tego katalogu.

`home`

Linux jest wieloużytkownikowym systemem operacyjnym. Każdy użytkownik otrzymuje konto i unikalny katalog do przechowywania plików osobistych. Taki katalog nazywany jest domowym katalogiem użytkownika (home directory). Katalog `/home` tworzony jest jako domyślna lokacja dla katalogów użytkowników.

`lib`

Biblioteki systemowe wymagane do podstawowych operacji są przechowywane w tym miejscu. Biblioteka C, the dynamic loader, biblioteka ncurses i moduły jądra są wśród zawartości tego katalogu.

`mnt`

Ten katalog zawiera tymczasowe punkty montowania dla dysków twardych bądź innych napędów (dyskietek, CD-ROM, etc.).

`opt`

Opcjonalne pakiety oprogramowania. Idea jaka kryje się za `/opt` polega na tym, że każdy pakiet instaluje się do `/opt/`*`software-package`*, co ułatwia jego usunięcie później. Slackware umieszcza kilka pakietów w tym miejscu (np. KDE w `/opt/kde`), ale możesz umieścić tam co uważasz.

`proc`

To unikalny katalog. Nie jest właściwie częścią systemu plików, ale wiertualnym systemem plików, który pozwala na dostęp do informacji nt. jądra. Różnorodne informacje, o których powinieneś wiedzieć są przekazywane przez kernel (jądro) poprzez pliki w `/proc`. Możesz również wysyłac informacje do jądra poprzez niektóre z tych plików. Spróbuj wykonać `cat /proc/cpuinfo`.

`root`

Administrator jest w systemie nazwany `root`'em. Katalog domowy `root`'a przechowywany jest w `/root` zamiast w `/home/root`. Powód jest oczywisty. Co by się stało gdyby `/home` była oddzielną partycją i nie mogłaby zostać zamontowana? `root` naturalnie chciałby się zalogować i naprawić ten problem. Jeżeli jego katalog domowy byłby na uszkodzonym systemie plików, stałoby się to trudne.

`sbin`

Najważniejsze programy uruchamiane przez `root`'a i podczas startu systemu są przechowywane w tym katalogu. Zwykli użytkownicy nie uruchomią programów z tego katalogu.

`tmp`

Miejsce na tymczasowe przechowywanie plików. Każdy użytkownik ma prawa do odczytu i zapisu w tym katalogu.

`usr`

To duży katalog w systemie Linux. Prawie wszystko nie wymienione powyżej umieszczane jest w tym miejscu. Programy, dokumentacja, źródło jądra i system X Window. To katalog, w którym w większości przypadków będziesz instalował oprogramowanie.

`var`

Logi systemowe, podręczne dane oraz program lock files są przechowywane w tym miejscu. To katalog dla często zniemiających się danych.

W tym momencie masz już całkiem niezłe pojęcie co gdzie w systemie się znajduje. Bardziej szczegółowe informacje na temat wyglądu systemu plików dostępne są na stronach manuala do hier(7). Następny rozdział pomoże ci odnajdywać pliki, by nie było konieczne ręczne przeglądanie każdego katalogu.

4.1.2 Znajdowanie Plików

Wiesz już co każdy z głównych katalogów zawiera, lecz to wcale nie pomaga ci w odnalezieniu konkretnego pliku. Możnaby oczywiście przeglądać każdy katalog po kolei ale są dużo szybsze sposoby. W Slackware są 4 główne polecenia do wyszukiwania plików.

4.1.2.1 `which`

Pierwsze z nich to polecenie `which`(1). W rzeczywistości używa się go do przeszukiwania katalogów wpisanych do zmiennej `PATH`. Zwracane jest pierwsze wystąpienie pliku i ścieżka do niego. Rozważ przykład:

```
%  which bash
/bin/bash
```

Widzsz więc, że `bash` znajduje się w katalogu `/bin`. Jest to dość ograniczone polecenie do szukania plików, gdyż przegląda tylko `PATH`.

4.1.2.2 `whereis`

Komenda `whereis`(1) działa podbnie do `which`, ale szuka również w stronach manuala i plikach źródłowych. Poszukiwanie pliku `bash` przy pomocy `whereis` powinno zwrócić:

```
%  whereis bash
bash: /bin/bash /usr/bin/bash /usr/man/man1/bash.1.gz
```

Ta komenda mówni nie tylko o tym gdzie program jest zlokalizowany, lecz również gdzie znajduje się jego dokumentacja. Wciąż jednak jest to polecenie dość ograniczone. Co zrobić gdy należy znaleźć jakiś plik konfiguracyjny? Nie można użyć `which` lub `whereis` do tego celu.

4.1.2.3 `find`

Polecenie `find`(1) pozwala użytkownikowi na przeszukanie systemu plików przy pomocy bogatej kolekcji orzeczeń (predykatów). Przykładowo by odnaleźć plik `xinitrc` w systemie poniższa komenda zostanie użyta:

```
%  find / -name xinitrc
/var/X11R6/lib/xinit/xinitrc
```

Działanie polecenia `find` zajmie trochę czasu gdyż musi ono przeszukać całe główne (root) drzewo katalogów. Dodatkowo gdy jest uruchomione przez zwykłego użytkownika zostaną wyświetlone komunikaty o odmowie dostępu (permission denied) dla katalogów do których dostęp posiada tylko root. Lecz `find` znalazł poszukiwany plik, więc jest dobrze. Gdyby tylko działał odrobinę szybciej...

4.1.2.4 `slocate`

Komenda `slocate`(1) przeszukuje cały system plików, tak jak to robi `find`, lecz używa do tego bazy danych zamiast rzeczywistych plików. Baza jest aktualizowana automatycznie każdego ranka, więc jest ona w miarę aktualna. Można ręcznie uruchomić `updatedb`(1), aby rozpocząć proces aktualizacji bazy slocate (przed uruchomieniem `updatedb` należy najpierw zalogować się jako root (używając `su` na przykład). Poniżej jest przykład działania `slocate`:

```
%  slocate xinitrc   # we don't have to go to the root
/var/X11R6/lib/xinit/xinitrc
/var/X11R6/lib/xinit/xinitrc.fvwm2
/var/X11R6/lib/xinit/xinitrc.openwin
/var/X11R6/lib/xinit/xinitrc.twm
```

Dostaliśmy więcej niż oczekiwalismy i to bardzo szybko. Przy pomocy powyższych poleceń powinieneś być w stanie znaleźć wszystko czego szukasz w swoim systemie.

4.1.3 Katalog `/etc/rc.d`

Pliki inicjalizacji systemu są przechowywane w katalogu `/etc/rc.d`. W Slackware używane są pliki startowe oparte na BSD. Są one przeciwieństwem skryptów inicjalizacyjnych (startowych) System V, które czynią konfigurację znacznie trudniejszą jeżeli nie używać programów przeznaczonych do ich edycji. W skryptach startowych BSD, każdy runlevel ma przypisany pojedyńczy plik rc. W System V, każdy runlevel ma swój katalog zawierający pewną liczbę skryptów startowych. Tak zorganizowana struktura pozwala na stosunkowo łatwą administrację.

Jest kilka kategorii plików inicjalizacyjnych. Są to: start systemu (system startup), runlevels, inicjalizacja sieci, kompatybilność z System V. Aby tradycji stało sie zadość wrzucimy resztę rzeczy do oddzielnej kategorii.

4.1.3.1 Start systemu - System Startup

Pierwszy program uruchamiany w Slackware poza jądrem Linuxa to `init`(8). Czyta on plik `/etc/inittab`(5) by sprawdzić jak uruchomić system. Następnie wykonuje skrypt `/etc/rc.d/rc.S` by przygotować system do przejścia w pożądany runlevel. Plik `rc.S` uaktywnia pamięć wirtualną, montuje systemy plików, czyści niektóre katalogi z logami, inicjuje urządzenia Plug and Play, ładuje moduły kernela, konfiguruje urządzenia PCMCIA, ustawia porty szeregowe i uruchamia skrypty startowe System V (o ile takie istnieją). Oczywiście `rc.S` ma dużo do roboty ale w `/etc/rc.d` jest kilka skryptów, które `rc.S` uruchomi by wykonać swoje zadanie:

`rc.S`

> Jest to właściwy skrypt inicjalizacyjny systemu.

`rc.modules`

> Ładuje moduły jądra. Obsługa kart sieciowych, PPP itd. ładowane są w tym miejscu. Gdy skrypt znajdzie plik `rc.netdevice` zostanie on również uruchomiony.

`rc.pcmcia`

> Sprawdza (Probes) i konfiguruje urządzenia PCMCIA, które możesz mieć w systemie. Użyteczny dla posiadaczy laptopów, którzy najczęściej mają modem lub kartę sieciową korzystającą PCMCIA.

`rc.serial`

> Konfiguruje porty szeregowe uruchamiając odpowiednie polecenia `setserial`.

`rc.sysvinit`

> Szuka skryptów startowych System V dla pożądanego runlevelu i uruchamia je. Ta kwestia szczegółowo omawiana jest poniżej.

4.1.3.2 Skrypty inicjalizacyjne dla runlevel'i

Po zakończonej inicjalizacji `init` przechodzi do inicjalizacji runlevel'u. Runlevel opisuje stan w jakim twoja maszyna będzie pracować. Brzmi dziwnie? Cóż, runlevel mówi `init`'owi czy chcesz akceptować logowanie dla wielu użytkowników, czy chcesz używać usług sieciowych i czy będziesz używał systemu X Window czy `agetty`(8) by obsługiwać logowania. Poniższe pliki opisują dostępne runlevel'e w Slackware Linux.

`rc.0`

Zatrzymaj system (runlevel 0). Domyślnie plik jest symbolicznie dowiązany do `rc.6`.

`rc.4`

Start wieloużytkownikowy, lecz używając X11 oraz KDM, GDM lub XDM jako menedżera logowania.

`rc.6`

Restart systemu (runlevel 6).

`rc.K`

Start w trybie pojedynczego użytkownika (single user) - runlevel 1.

`rc.M`

Tryb wieloużytkownikowy (runlevel'e 2 i 3) lecz ze standardowym tekstowym logowaniem. Jest to domyślny runlevel w Slackware.

4.1.3.3 Inicjalizacja sieci

Runlevel'e 2, 3, i 4 uruchomią usługi sieciowe. Poniższe pliki są odpowiedzialne za inicjalizację sieci:

`rc.inet1`

Plik utworzony przez `netconfig`. Odpowiedzialny jest za konfigurację interfejsu sieciowego.

`rc.inet2`

Uruchamiany po `rc.inet1`. Uruchamia podstawowe usługi sieciowe.

`rc.atalk`

Odpala usługi AppleTalk.

`rc.httpd`

Plik uruchamia serwer webowy Apache. Jak kilka jeszcze innych skryptów rc, i ten może służyć również (oprócz uruchomienia) do zatrzymania i zrestartowania usługi. `rc.httpd` pobiera argumenty: stop, start lub restart.

`rc.news`

Uruchamia serwer newsów (grup dyskusyjnych).

4.1.3.4 Kompatybilność z System V

Kompatybilność z init System V została wprowadzona w Slackware w wersji 7.0. Wiele innych dystrybucji Linux'a używa tego stylu zamiast tego opartego o BSD. Generalnie wygląda to tak, że każdy runlevel ma swój katalog dla skryptów startowych. W stylu BSD jeden runlevel to jeden plik.

Plik `rc.sysvinit` przeszuka `/etc/rc.d` i sprawdzi czy masz tam jakieś pliki startowe System V a następnie je

uruchomi (o ile odpowiadają one aktualnemu runlevel'owi). Jest to użyteczne dla pewnych komercyjnych pakietów oprogramowania, które instalują skrypty startowe System V.

4.1.3.5 Inne pliki

Skrypty opisane poniżej to pozostałe pliki inicjalizacji systemu. Są one zwykle uruchamiane z jednego z głównych skryptów startowych opisanych wyżej. Wszystko więc co musisz zrobić by wprowadzić w nich uaktualnienia to je wyedytować.

`rc.gpm`

> Uruchamia usługi general purpose mouse (mysz ogólnego przeznaczenia :-). Pozwala na kopiowanie i wklejanie w konsoli Linux'owej. Niekiedy gpm stwarza problemy gdy jest używany pod X Window. Gdy takowych doświadczysz w tym środowisku, spróbuj odebrać prawa do wykonywania tego pliku i zatrzymaj serwer gmp.

`rc.font`

> Ładuje czcionki ekranowe dla konsoli.

`rc.local`

> Zawiera polecenia startowe specyficzne dla Twojego systemu. Plik jest pusty zaraz po instalacji, jakoże jest zarezwerowany dla administratorów. Ten skrypt jest uruchamiany po wykonaniu wszystkich innych czynności inicjalizacyjnych.

Aby aktywować skrypt, wszystko co należy wykonać to nadać mu prawa do wykonywania poleceniem `chmod`. Aby go “dezaktywować” trzeba mu te prawa odebrać. Po więcej informacji na temat `chmod` zobacz Sekcja 9.2.

4.2 Wybieranie Jądra

Jądro jest częścią systemu operacyjnego, która umożliwia dostęp do sprzętu, kontrolę procesów i całego systemu. Kernel zawiera procedury obsługi urządzeń więc wybranie odpowiedniego dla Twojego systemu jest ważnym etapem konfiguracji.

Slackware zwiera ponad tuzin skompilowanych już jąder. Każdy z nich zawiera standardowy zestaw sterowników oraz kilka specyficznych tylko dla niego. Możesz używać jednego z prekompilowanych jąder lub sam go skompilować ze źródeł. Cokolwiek wybierzesz musisz się upewnić, że Twój kernel zawiera w sobie obsługę posiadanych przez Ciebie urządzeń.

4.2.1 Katalog `/kernels` na płycie Slackware

Prekompilowane jądra Slackware są dostępne w katalogu `/kernels` na płycie CD lub na serwerze FTP w głównym katalogu Slackware. Dostępne kernele zmieniają się gdy tworzona jest nowa wersja systemu, więc aktualna dokumentacja dostępna jest zawsze w ich katalogu. Dla każdego jądra w katalogu `/kernels` utworzony został podkatalog; każdy z nich ma taką samą nazwę jak odpowiadający mu boot disk. Ponadto w każdym z folderów znajdują się niżej wymienione pliki:

Plik	Do czego służy
`System.map`	Mapa systemu dla danego jądra
`bzImage`	Plik obrazu jądra

Plik	Do czego służy
`config`	Plik konfiguracyjny dla danego kenela

Aby użyć wybranego jądra, skopiuj `System.map` oraz `config` do katalogu `/boot`. Obraz jądra zapisz jako `/boot/vmlinuz`. Następnie uruchom `/sbin/lilo`(8) aby zainstalować LILO dla nowego kernela. Zrestartuj system. Tak wygląda instalowanie nowego jądra.

Nazwy plików z jądrami IDE kończą się na .i. Nie zawierają one obsługi SCSI. Pliki z jądrami ze wsparciem dla SCSI kończą się na .s. Zawierają również obsługę (oprócz SCSI oczywiście) IDE tak jak to ma miejsce w kernelach *.i.

4.2.2 Kompilacja jądra ze źródła

Pytanie “czy muszę kompilować jądro dla swojego systemu?” często zadawane jest przez nowych użytkowników. Odpowiedź jest jedna: być może. Jest kilka przypadków, w których powinno się to uczynic. Większość jednak może używać gotowych kerneli i ładowalnych modułów by uczynić system w pełni funkcjonalnym. Kompilacja będzie konieczna gdy uaktualnia się kernel do wersji nie zawartej w aktualnej wersji dystrybucji, lub gdy patch'owano źródła jądra w celu dodania obsługi jakichś specyficznych urządzeń. Każdy kto posiada system z SMP na pewno zechce skompilować jądro z jego obsługą. Dodatkowo wielu użytkowników dokonuje tego w celu poprawienia wydajności systemu. Można na przykład skompilować jądro z opcjami optymalizacji dla konkretnego procesora.

Utworzenie własnego kernela nie jest aż takie trudne. Pierwszy krok to sprawdzenie czy na systemie zainstalowanio źródła jądra. Należy się upewnić, że zainstalowano pakiety z grupy K. Ważna jest również grupa D, a w szczególności kompilator C, GNU make oraz GNU binutils. W ogólności warto zainstalować całą grupę D jeżeli planuje się rozwijać w jakikolwiek sposób system. Źródła (najaktualniejsze) kernela pozyskać można również z http://www.kernel.org/mirrors.

4.2.2.1 Komplacja jąder wersji 2.4.x

```
% su -
Password:
# cd /usr/src/linux
```

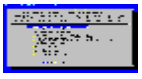
Pierwszą rzeczą jaką należy zrobić to przywrócic źródła do stanu podstawowego. W tym celu wykonujemy poniższe polecenie (zauważ, że warto zachować kopię `.config` - to polecenie tego jednak nie robi):

```
# make mrproper
```

Teraz należy skonfigurować jądro dla swojego systemu. Bieżący kernel oferuje trzy sposoby na dokonanie tego. Pierwszy to klasyczny tekstowy system pytanie-odpowiedź. Zadanych zostaje mnóstwo pytań i na podstawie odpwiedzi budowany jest plik konfiguracyjny. Największy problem polega na tym, że w wypadku pomyłki cały proces należy powtórzyć. Metoda preferowana przez większkość użytkowników to konfiguracja oparta o menu. Ostatnią metodą jest najrzędzie oparte na X'ach. Wybierz tę, która Ci najbardziej odpowiada tj.:

```
# make config           (wersja tekstowa Q&A)
# make menuconfig       (oparte o menu, również tekstowe)
# make xconfig          (wersja korzystająca z X'ów - upewnij się ze je masz!)
```

Rysunek 4-1. Menu konfiguracji jądra (Kernel Configuration Menu)



Nowi użytkownicy prawdopodobnie wybiorą `menuconfig`. Dla każdej części kernela wyświetlane są ekrany pomocy w celu wyjaśnienia ich znaczenia. Po zakończonej konfiguracji wyjdź z programu. Potrzebne pliki zostaną zapisane. Teraz należy przygotować drzewo katalogów ze źródłami do kompilacji:

```
# make dep
# make clean
```

Następny krok to kompilacja. Wydaj poniższe polecenie:

```
# make bzImage
```

Może to zająć pewien czas, który zależy od mocy procesora jaki posiadasz. Podczas procesu kompilacji na ekranie pokazywane będą informacje od kompilatora. Gdy powyższy etap się zakończy należy skompilować fragmenty jądra oznaczone jako moduły:

```
# make modules
```

Teraz należy zainstalować jądro i moduły uprzednio spreparowane. Aby tego dokonać w Slackware następujące polecenia powinny zostać wydane:

```
# mv /boot/vmlinuz /boot/vmlinuz.old
# cat arch/i386/boot/bzImage > /vmlinuz
# mv /boot/System.map /boot/System.map.old
# cp System.map /boot/System.map
# make modules_install
```

Dobrym pomysłem jest edycja pliku `/etc/lilo.conf` w celu dodania sekcji umożliwiającej bootowanie starego jądra na wypadek gdyby coś poszło nie tak. Po wykonaniu tego kroku należy uruchomić `/sbin/lilo` by zainstalować nowy blok bootujący. Teraz można juz zrestartować system i cieszyć się nowym kernelem :-).

4.2.2.2 Jądra Linux'a wersje 2.6.x

Kompilacja jądra w wersji 2.6 różni się tylko nieznacznie od 2.4 czy 2.2. Istotne jest jednak by zrozumieć róznice przed jej rozpoczęciem. Po pierwsze nie ma juz kroku z `make dep` oraz `make clean`. Po drugie podczas procesu kompilacji ograniczeniu uległa ilość wiadomości wyświetlanych na ekranie. Wszystko to powoduje, że kompilacja staje się łatwiejsza do zrozumienia, jeżeli mimo to pojawią się jakieś problemy w jej trakcie zaleca się powrócenie do trybu 'gadatliwego'. Aby tego dokonać należy dodać opcję `V=1`. Spowoduje to pojawienie się wielu informacji podczas kompilacji, które mogą okazać się przydatne osobom rozwijającym jądro lub koledze, który zna się na rzeczy :-). Być może rozwiąże to twój problem.

```
# make bzImage V=1
```

4.2.3 Używanie modułów jądra

Moduły jądra to inna nazwa dla sterowników urządzeń, które można doczepić do działającego jądra. Pozwalają one na poszerzenie listy obsługiwanych przez twój kernel urządzeń bez konieczności zmiany lub ponownej kompilacji jądra.

Moduły mogą być ładowane i usuwanie (wyłączane) w dowolnym czasie podczas pracy systemu. Jest to bardzo użyteczne dla w administratorów wypadku aktualizacji sterowników dokonywanej przez nich. Nowy moduł może zostać skompilowany, stary usunięty, nowy załadowany - i wszystko bez konieczności restartu maszyny.

Moduły przechwywane są w katalogu `/lib/modules/`*`kernel version`*. Mogą być ładowane w czasie

bootowania przez plik `rc.modules`. Jest on opatrzony licznymi komentarzami i przykładami. Aby zobaczyć listę modułów aktywnych w danym momencie użyj polecenia `lsmod`(1):

```
# lsmod
Module                  Size  Used by
parport_pc              7220   0
parport                 7844   0  [parport_pc]
```

W powyższym przykładzie widać, że na mojej maszynie załadowany jest tylko moduł portu równoległego. Aby usunąć moduł, należy użyć polecenia `rmmod`(1). Moduły mogą być ładowane komendą `modprobe`(1) lub `insmod`(1). `modprobe` jest zwykle bezpieczniejszy ponieważ spowoduje on załadowanie nie tylko pożądanego modułu ale również wszystkich, od których on zależy.

Wielu użytkowników nigdy nie będzie zmuszonych do ładowania i usuwania modułów ręcznie. Służy do tego autoloader do zarządzania modułami. Domyślnie Slackware zawiera `kmod` w swoich jądrach. `kmod` to opcja kernela, która uaktywnia automatyczne ładowanie modułów gdy są wymagane. Aby uzyskać więcej informacji na temat `kmod` i jego konfiguracji zobacz `/usr/src/linux/Documentation/kmod.txt`. W tym celu powinieneś mieć zainstalowany pakiet z źródłami jądra lub sciągnąć takowe z http://kernel.org.

Więcej informacji można znaleźć w stronach manuala dla powyższych poleceń oraz w pliku `rc.modules`

Rozdział 5 Konfiguracji sieci

5.1 Wprowadzenie: netconfig twoim przyjacielem.

Kiedy wstępnie zainstalujesz Slackware, program instalacyjny wywola program `netconfig`. `netconfig` sprobuje wykonać za Ciebie następujące czynności:

- Zapyta się o nazwę Twojego komputera i domenę w jakiej się znajduje.
- Zwięźle przedstawi i wyjaśni różne typy adresowania (zdobywania IP [przyp. tłum]), powinformuje kiedy powinno się go użyć, i zapyta, z którego typu adresowania IP chcesz skorzystać do skonfigurowania swojej karty sieciowej:
  - statyczne-IP
  - DHCP
  - Loopback
- Następnie zaproponuje próbę konfiguracji karty sieciowej.

`netconfig` generalnie wykona 80% pracy związanej z konfiguracją podłączenia do sieci LAN. Mocno zaleca się sprawdzenie plików konfiguracyjnych a to z kilku powodów:

1. Nigdy nie należy ufac, że program instalacyjny prawidłowo skonfiguruje Twój komputer. Jeśli używasz programu instalacyjnego powineneś samemu sprawdzić konfigurację.
2. Jeśli nadal uczysz sie Slackware i administracji systemem, podglądanie działającej konfiguracji może okazać się bardzo przydatne. Co najmniej zdobędziesz wiedzę jak powinna ona wygladać. Ponadto pozwoli Ci to naprawić błedy podczas rekonfiguracji systemu w poźniejszym czasie.

5.2 Konfiguracja urządzeń sieciowych

Decydując się na podłaczenie swojej maszymy ze Slackiem do jakiejkolwiek sieci, pierwszą niezbędną rzeczą jest karta sieciowa zgodna z Linuksem. Warto zwrócić szczególną uwagę na wspomnianą kompatybilność. (W tym celu sprawdź Projekt Dokumentacji Linuxa i/lub dokumentacje kernela w celu sprawdzenia statusu Twojej karty.) Zasadniczo będziesz mile zaskoczony liczbą kart sieciowych obsługiwanych przez nowsze jądra. Mowiąc to nadal zachęcam do sprawdzania różnych list zgodności sprzętowej (jak naprzyklad The GNU/Linux Beginners Group Hardware Compatibility Links i The Linux Documentation Project Hardware HOWTO). Dostępne są one w internecie. Trochę dodatkowo spędzonego czasu ma poszukiwaniach może oszczędzić dni, a nawet tygodni poświeconych na rozwiązywaniu problemu w przypadku gdy karta nie jest zgodna z Linuksem.

Kiedy będziesz przeglądać listę zgodności sprzętowej dla Linuksa dostępną przez internet, lub dokumentację kernela zainstalowaną na komputerze dobrym pomysłem będzie zanotowanie, który moduł jądra obsługuje Twoją karte sieciową.

5.2.1 Ładowanie modułów sieci

Moduły jądra, które mają zostać załadowane przy starcie systemu ładowane są z pliku `rc.modules` znajdujacego sie w `/etc/rc.d` lub przez auto-loadera uruchamiango w `/etc/rc.d/rc.hotplug`. Standardowy plik `rc.modules` zawiera sekcję obsługiwanych urządzeń sieciowych. Gdy otworzysz `rc.modules` i spojrzysz do wspomnianej sekcji, zauważysz, że na początku sprawdzana jest “wykonywalność” `rc.netdevice` znajdującego się w `/etc/rc.d/`. Skrypt ten tworzony jest jesli `setup` z powodzeniem ustawi urzadzenie sieciowe podczas instalacji.

We wspomnianym `rc.netdevice` znajduje się lista urządzeń sieciowych. Odnajdź swoje urządzenie i odkomentuj linie z odpowiednim wpisem modprobe, następnie zapisz plik. Uruchomienie `rc.modules` jako `root` powinienno załadowac sterowniki swojego twojego sieciowego (tak samo jak wiele innych modułów, które są wyszczególnione i nie zakomentowane). Zauważ, że niektóre moduły (jak naprzykład sterownik ne2000) wymagaja parametrów; upewnij się, że wybrana została właściwą linie.

5.2.2 Karty LAN (10/100/1000Base-T and Base-2)

Ta grupa obejmuje wszystkie wewnętrzne karty sieciowe PCI i ISA. Ich sterowniki dostarczane są jako ładowalne moduły jądra. `/sbin/netconfig` powinien wyszukac twoją kartę i skonfigurować `rc.netdevice`. Jesli to nie zadziała, najczęstszą przyczyną jest to, że moduł, który próbujesz załadowac dla podanej karty jest niewlasciwy (nie jest nowością, że różne generacja kart tej samej marki i tego samego producenta wymagają różnych modułów). Gdy ten powód nie okaże się prawdziwy, należ odwołać się do dokumentacji modułu by sprawdzić czy nie należy przekazać mu pewnych parametrów podczas inicjalizacji.

5.2.3 Modems

Tak jak karty LAN, modemy moga być obsługiwane przez różne magistrale. Do niedawna większosc modemów była 8 lub 16 bitowymi kartami ISA. Dzięki staraniom Intela i producentów płyt głównych aby pozbyć sie definitywnie standardu ISA, obecnie większość modemów to zewnetrzne urządzenia podłączane do USB lub modemy PCI. Jesli rozważane jest podłączenie modemu w systemie Linux *bardzo* istotne jest dokładnie przemyślenie zakupu. W dzisiejszych czasach większość modemów (szczególnie tych na PCI) to WinModemy. Nie posidają one pewnych podstawowych elementów sprzętowych, których funkcje realizowane są przez CPU i system operacyjny Windows. Oznacza to, że nie posiadają standardowego interfejsu szeregowego, którego PPPD będzie oczekiwał w momencie łączenia z dostawcą internetu (ISP - Internet Service Provider).

Jesli chcesz mieć absolutną pewność, że zakupiony modem będzie współpracował z Linuksem, zaopatrz się w zewnetrzny modem łączacy się z PC przez port szeregowy(serial port). Zagwarantuje to lepszą prace i mniej kłopotów przy instalacji i utrzymaniu, pomimo wymogu zewnetrznego zasilania i wyższej ceny.

Jest kilka stron internetowych, które oferuja sterowniki i pomoc przy konfiguracji urządzeń bazujacych na

WinModemach. Niekótrym uzytkownikom udała się konfiguracja i instalacja sterownikow dla róznych Winmodemow, wliczajac te na chipsetach Lucenta, Conexanta, i Rockwella. Jako że wymagane oprogramowanie dla tych urządzeń nie jest częścią Slackware, i różni się w zależności od modemu, nie bedziemy się zaglebiac w detale.

5.2.4 PCMCIA

Podczas instalacji Slackware, istnieje możliwosc instalacji pakietów pcmcia (w serii pakietów “A”). Zawierają się w nich programy i pliki konfiguracyjne wymagane do pracy z kartami PCMCIA pod Slackware. Trzeba podkreslić, że pakiet pcmcia instaluje tylko ogolne oprogramowanie wymagane do pracy z kartami PCMCIA pod Slackiem. NIE instaluje żadnych sterowników czy modulów. Dostępne moduły i sterowniki znajdą się w katalogu `/lib/modules/`uname -r`/pcmcia`. Być może trzeba będzie trochę poeksperymentowac by znalezc moduł, który bedzie pracować z twoją karta sieciowa.

W celu dokonania konfiguracji należy wyedytować `/etc/pcmcia/network.opts` (dla kart ethernetowych) lub `/etc/pcmcia/wireless.opts` (w przypadku kart wireless (bezprzewodowych)). Tak jak wiekszość plików konfiguracyjnych w Slackware, również wspomniane opatrzone są licznymi komentarzami. Modyfikacje nie powinny nastręczyć wielu trudności.

5.3 Konfiguracja TCP/IP

W tym miejscu twoja karta sieciowa powinna byc fizycznie zainstalowana w komputerze, oraz odpowiedni moduł krenela zaladowany. Nie bedziesz jeszcze w stanie komunikować się przy użyciu karty, ale informacje o urzadzeniu sieciowym możesz uzyskać przez `ifconfig -a`.

```
# ifconfig -a
eth0 Link encap:Ethernet HWaddr 00:A0:CC:3C:60:A4
UP BROADCAST NOTRAILERS RUNNING MULTICAST MTU:1500 Metric:1
RX packets:110081 errors:1 dropped:0 overruns:0 frame:0
TX packets:84931 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:114824506 (109.5 Mb) TX bytes:9337924 (8.9 Mb)
Interrupt:5 Base address:0x8400

lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:2234 errors:0 dropped:0 overruns:0 frame:0
TX packets:2234 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:168758 (164.8 Kb) TX bytes:168758 (164.8 Kb)
```

Jesli poprostu wpiszesz `/sbin/ifconfig` bez sufixu `-a`, nie zobaczysz interfejsu `eth0`, jako że twoja karta sieciowa nie ma jeszcze poprawnego adresu IP.

5.3.1 Ogólnie

Podczas gdy jest wiele róznych sposobów ustawiania sieci, wszystkie z nich moga byc podzielone na 2 typy: statyczne i dynamiczne. Statyczne sieci są budowane tak, że dany węzeł zawsze ma ten sam adres IP. Dynamiczne seci budowane są tak, że adresy IP dla węzłów są kontrolowane przez jeden serwer zwanym serwerem DHCP.

5.3.2 DHCP

DHCP (Dynamic Host Configuration Protocol), jest środkiem, dzięki któremu adres IP może być przydzielony

komputerowi na starcie. Kiedy *klient* DHCP startuje, wysyła żądanie przez sieć lokalną do *serwera* DHCP o przydzielenie adresu IP. Serwer DHCP ma pulę (lub *zakres*) dostepnych adresow IP. Serwer odpowie na to żądanie adresem IP z puli razem z *czasem dzierżawy*. Kiedy czas dzierżawy dla danego adresu IP wygaśnie, klient musi skontaktowac się ponownie z serwerem i powtórzyć negocjacje.

Następnie klient DHCP zaakceptuje adres IP z serwera i skonfiguruje żądany interfejs adresem IP. Jest jeszcze jedena przydatna sztuczka jakiej klienci DHCP używaja przy negocjacjach adresow IP jakie zostana przydzielone. Klient pamieta ostatnio przydzielony adres i prosi serwer o ponowny przydzial tego samego adresu IP przy nastepnych negocjacjach. Jeśli jest to możliwe serwer zrobi to, jeśli nie, przydzielany jest nowy adres. Wiec neogocjacje wygladaja mniej wiecej tak:

Klient: Czy serwer DHCP jest dostepny w sieci LAN?
Serwer: Tak, jest. Ja nim jestem
Klient: Potrzebuje adresu IP
Serwer: Mozesz wziasc 192.168.10.10 na 19200 sekund.
Klient: Dziekuje

Klient: Czy serwer DHCP jest dostępny w sieci LAN?
Serwer:Tak, jest. Ja nim jestem
Klient:Potrzebuje adresu IP. Ostatnio jak rozmawialiśmy, miałem 192.168.10.10; moge go dostać ponownie?
Serwer:Tam mozesz (lub Nie, nie mozesz: wez w zamian 192.168.10.12).
Klient: Dziekuje.

Klientem DHCP w Linuxie jest `/sbin/dhcpcd`. Jeśli otworzysz `/etc/rc.d/rc.inet1` w swoim ulubionym edytorze tekstu, zauważysz, że `/sbin/dhcpcd` jest wywoływany gdzies pośrodku skryptu. `dhcpcd` śledzi także ilość czasu pozostałego na dzierżawę obecnego adresu IP i automatycznie kontaktuje się z serwerem DHCP żądając jej odnowienia jeśli to konieczne. DHCP może również kontrolować pokrewne informacje, jak na przykład którego serwera ntp użyć, jaka trasę obrać, itp.

Ustawianie DHCP w Slackware jest proste. Należy uruchomić `netconfig` i wybrać opcję DHCP. Jeśli masz wiecej niż jedną NIC(Network Interface Card - karta sieciowa) i nie życzysz sobie aby `eth0` było konfigurowane przez DHCP, poprostu wyedytuj plik `/etc/rc.d/rc.inet1.conf` i zmień powiązane zmienne dla twojego NIC na "`YES`".

5.3.3 Statyczne IP

Statyczne adresy IP to niezmienne adresy, które można zmienić "recznie". Używawne są wszędzie tam gdzie administrator nie chce by informacja IP się zmieniała, jak w przypadku wewnetrznych serwerów w LANach, wszystkich serwerów podpietych do Internetu i routerów sieciowych. W przypadku statycznego adresowania IP jest przydzielany i tak pozostawiany. Pozostałe maszyny wiedzą, że zawsze masz określony adres IP i zawsze mogą się z tobą skontaktować używając go.

5.3.4 `/etc/rc.d/rc.inet1.conf`

Jeśli planujesz przypisać adres IP do swojej nowej platformy ze Slackware na pokładzie, mozesz to uczynić zarówno przez skrypt `netconfig`, jak i edycję `/etc/rc.d/rc.inet1.conf`. W tym ostatnim znajdziesz:

```
# Primary network interface card (eth0)
IPADDR[0]=""
NETMASK[0]=""
USE_DHCP[0]=""
DHCP_HOSTNAME[0]=""
```

I na końcu:

```
GATEWAY=""
```

W tym przypadku twoim zadaniem jest jedynie wpisanie poprawnych informacji pomiedzy znaki cudzysłowia. Zmienne te są wywoływane przez `/etc/rc.d/rc.inet1` na starcie systemu aby ustawic sieciówki. Dla kazdej NIC, poporostu wprowadź właściwą informację IP lub przypisz "`YES`" zmiennej `USE_DHCP`. Slackware podniesie interfejsy uwzględniając parametramy jakie zostały wpisane w takiej kolejnosci w jakiej je znajdzie.

Zmienna `DEFAULT_GW` ustawia domyślną trasę (ang. route) dla Slackware. Cała komunikacja pomiędzy twoim a innymi komputerami w Internecie musi przejść przez bramkę (gateway) o ile żadna inna trasa nie jest dla nich określona. Jesli używasz DHCP zazwyczaj nie musisz wpisywac niczego ponieważ DHCP określi jakiej bramki użyć.

5.3.5 `/etc/resolv.conf`

Ok, jeśli masz adres IP, domyślną bramkę, może masz też milion dolarów (podziel sie z nami), ale co z tego skoro nie możesz przypisać nazwy do adresu IP? Nikt nie chce wpisywać `72.9.234.112` do przeglądarki aby odwiedzić `www.slackbook.org`. No i kto poza autorami zapamięta ten adres IP? Należy ustawić DNS, ale jak? Z pomoca przychodzi `/etc/resolv.conf`.

Istnieje prawdopodobieństwo, że masz poprawne opcje w `/etc/resolv.conf`. Jeśli ustawiasz swoje połączenie sieciowe używając DHCP, serwer DHCP powinien zająć się aktualizacją tych plikow za ciebie. (Technicznie serwer DHCP poporstu informuje `dhcpcd` co ma tam wpisac.) Jeśli mimo wszystko trzeba ręcznie zaktualizowac swoją listę DNS, należy wyedytowć plik `/etc/resolv.conf`. Poniżej znajduje się przykład:

```
# cat /etc/resolv.conf
nameserver 192.168.1.254
search lizella.net
```

Pierwsza linijka jest prosta. Dyrektywa nameserver mowi nam o jaki serwer DNS pytac. Z koniecznosci jest tam zawsze adres IP. Możesz miec ich tam tyle ile sobie zażyczysz. Slackware sprawdzi jeden po drugim aż wynik będzie zadowalający.

Druga linia jest trochę bardziej interesująca. Dyrektywa szukania daje liste nazw domen objetych zawsze kiedy DNS wysyła żądanie. Umożliwia to porozumienie się z maszyną tylko dzięki pierwszej części jej FQDN (Fully Qualified Domain Name). Na przykład, jeśli "slackware.com" jest w scieżce wyszukiwania, można dotrzeć do `http://store.slackware.com` przez wpisanie samego `http://store`.

```
# ping -c 1 store
PING store.slackware.com (69.50.233.153): 56 data bytes
64 bytes from 69.50.233.153 : icmp_seq=0 ttl=64 time=0.251 ms
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.251/0.251/0.251 ms
```

5.3.6 `/etc/hosts`

Tak więc DNS działa poprawnie. Co jesli chcemy ominąć nasz serwer DNS, lub dodać wpis DNS dla mszyny, której nie ma na liście DNSa? Slackware dostarcza plik `/etc/hosts`, który zawiera lokalną listę nazw DNS i adresow IP, do których powinny one pasować.

```
# cat /etc/hosts
127.0.0.1           localhost  locahost.localdomain
192.168.1.101       redtail
```

```
172.14.66.32        foobar.slackware.com
```

Jak widać powyżej localhost ma adres IP `127.0.0.1` (zawsze zarezerwowany dla localhosta), redtail przypisany jest do `192.168.1.101`, a `foobar.slackware.com` odpowiada IP `172.14.66.32`.

5.4 PPP

Wciąż wielu ludzi łaczy się z Internetem przez różnego rodzaju połączenia dialup. Najpopularniejszą metodą jest PPP, chociaż SLIP również jest okazjonalnie używany. Skonfigurowanie systemu aby komunikował się ze zdalnym serwerem przez PPP jest całkiem proste. Dołaczyliśmy kilka narzędzi pomocnych przy konfiguracji.

5.4.1 `pppsetup`

Slackware zawiera program nazwany `pppsetup` do konfiguracji systemu aby używać połączenia dialup. Jest podobny w wygladzie i użyciu do programu `netconfig`. Najpierw zaloguj się jako `root`. Następnie wpisz `pppsetup` aby program uruchomić.

Program przedstawi serię pytań, na które oczywiście trzeba odpowiedzieć. Będą one dotyczyć wyboru urządzenia modemu, komendy uruchamiającej, numeru telefonu ISP (Internet Service Provider - dostawca internetu). Niektóre z usatwaień bedą miały wartości domyślne, które w większości przypadków mogą zostać zaakceptowane.

Po zakończeniu konfiguracji utowrzone zostaną programy `ppp-go` i `ppp-off`. Używa się ich do odpowiednio uruchamiania i zatrzymywania połączenia PPP. Programy te znajdą się w `/usr/sbin`. Do ich uruchomienia wymagane są uprawnienia roota.

5.4.2 `/etc/ppp`

Dla większości użytkowników, uruchomienie `pppsetup` wystarczy. Jednakze może zajść potrzeba zmiany wartości uzywanych przez deamona PPP. Wszystkie informacje o konfiguracji trzymane są w `/etc/ppp`. Oto lista czego dotyczą poszczególne pliki:

`ip-down`	Skrypt uruchamiany przez `pppd` po tym jak połączenie jest zakończone.
`ip-up`	Skrypt uruchamiany przez `pppd`, kiedy nastapi udane połaczenie ppp. W tym pliku umieść wszystkie komendy jakie chcesz uruchomić po udanym polaczeniu.
`opcje`	Ogólna konfiguracja opcji dla `pppd`.
`options.demand`	Konfiguracja opcji dla `pppd` kiedy działa w trybie żądania połączenia (ang. demand dialing mode).
`pppscript`	Polecenia wysyłane do modemu.
`pppsetup.txt`	Logi na temat opcji wybranych w czasie uruchamiania `pppsetup`.


Wiekszosci z powyższych plików nie będzie dostępnych do czasu uruchomienia `pppsetup`.

5.5 Wireless

Sieci bezprzewodowe (Wireless) nadal są stosunkowo nową rzeczą w świecie komputerów. Obecnie szybko zdobywają coraz więcej zwolenników kupujących laptopy i chcących surfować w trasie, którzy nie chcą być więźniami ograniczającej skrętki. Tendencja ta nie wydają się słabnąć. Niestety sieci bezprewodowe nie są

jeszcze zbyt dobrze wspierane przez Linuksa jak to ma miejsce w przypadku tradycyjnych sieci kablowych.

Są trzy podstawowe wymagania/kroki do skonfigurowania karty bezprzewodowej wireless 802.11:

1. Obsługa sprzętowa kart bezprzewodowych
2. Konfiguracja karty dla połączenia z access pointem (AP)
3. Konfiguracja sieci

5.5.1 Obsługa sprzetowa

Wsparcie sprzętowe dla kart sieci bezprzewodowych realizowane jest przez jądro systemu, zarówno przez moduły jak i wbudowane funkcje. Ogólnie większość nowszych kart Ethernetowych wspieranych jest przez moduły kernela, wiec nie ma potrzeby ustawiania dodatkowych modułów i ładowania ich przez `/etc/rc.d/rc.modules`. `netconfig` może nie wykryć karty sieci bezprzewodowej, więc prawdopodobnie bedzie trzeba zrobić to ręcznie. Więcej szczegółów na http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/. Są tam informacje o sterownikach jądra dla różnych kart bezprzewodowych.

5.5.2 Konfiguracja ustawień sieci bezprzewodowej

Najważniejsza część tej pracy wykonywana jest przez `iwconfig`, więc jak zwykle zapoznaj się ze stronami manuala dla `iwconfig` by uzyskać więcej informacji.

Po pierwsze, należy skonfigurować swój access point. Wireless access pointy nieznacznie różnią się w swojej terminologii, w sposobie konfiguracji. Może więc zajść potrzeba rekonfiguracji ustawień. Krótko mówiąc potrzebne będą poniższe informacje:

- ID domeny, lub nazwa sieci (ESSID wywoływane przez `iwconfig`)
- Używany kanał WAP
- Ustawienia szyfrowania, wliczajac wszystkie klucze (oryginalnie w zapisie hexadecymalnym)

UWAGA O WEP. WEP jest trochę kłopotliwy, ale znacznie lepszy niz nic. W celu zwiększenia poziomu bezpieczeństwa sieci bezprzewodowej, należy zbadac VPN lub IPSec. Temat ten wykracza niestety poza ramy tego dokumentu. Możesz także skonfigurowac WAPa aby nie rozgłaszał swojej domeny ID/ESSID. Gruntowna dyskusja o polityce sieci bezprzewodowej wykracza poza ten dokument, ale szybkie przeszukanie zasobów Google odkryje wiecej niz możnaby się spodziewać.

W końcu, kiedy zbierzesz wszystkie informacje i zakładając, że używasz `modprobe` do ładowania odpowiednich sterowników jądra, należy wyedytować `rc.wireless.conf` i dodać swoje ustawienia. Plik `rc.wireless.conf` jest trochę bałaganiarski. Najmniej męcząca jest zmiana sekcji z ESSID, KEY, i CHANNEL jeśli wymagane przez kartę. (Spróbuj nie ustawiać CHANNEL - jesli będzie działać to super, jesli nie to ustaw CHANNEL jak nalezy.) Jeśli masz dość odwagi mozesz zmodyfikowac plik tak, żeby tylko niezbędne zmienne były ustawione. Nazwy zmiennych w `rc.wireless.conf` odpowiadają prametrom w `iwconfig`, i są odczytywane przez `rc.wireless` oraz używane odpowiednio przez polecenia `iwconfig`.

Jeśli masz klucz w postaci hexadecymalnej, to świetnie. Prawie na pewno klucze WAPa i `iwconfig` będą zgodne. Jeśli masz tylko wyrażenie(ciąg znaków), nie wiadomo jak WAP przetlumaczy go na klucz hexadecymalny, wiec moze byc potrzebna zgadywanka (lub zdobycie klucza WAP w hex).

Po modyfikacji `rc.wireless.conf`, uruchom `rc.wireless` jako `root`, następnie uruchom `rc.inet1`, znowu jako `root`. Możesz przetestować sieć bezprzewodową standardowymi narzędziami tj `ping` oraz `iwconfig`. Jesli masz interfejs z kablem możesz użyc `ifconfig` aby wyłączyć go w czasie gdy bedziesz testować sieć

bezprzewodową. Możesz także chcieć przetestować swoje zmiany po reboocie.

Teraz wiesz jak edytowac `/etc/rc.d/rc.wireless` dla twojej sieci, przyjrzyjmy się bliżej iwconfig i zobaczmy jak on działa. Nauczy cię to jak szybko i brutalnie ustawiac wifi kiedy znajdziesz się w kafejce internetowej, kawiarence lub przy innym hot spocie i zechcesz posurfowac po sieci.

Pierwszym krokiem jest poinformowanie NICa do jakiej sieci ma się przyłączyć. Należy "`eth0`" zamienić na interfejs sieciowy, którego używa karta bezprzewodowa i zmienic "*`mynetwork`*" na pożądany essid. (Tak, oczywiście wiesz to wszystko :-)) Następnie określ klucz szyfrujacy (jesli istnieje) używany w sieci. Na koncu określ używany kanał (jeśli istnieje taka potrzeba).

```
# iwconfig eth0 essid "mynetwork"
# iwconfig eth0 key XXXXXXXXXXXXXXXXXXXXXXXXXX
# iwconfig eth0 channel n
```

To wszystko o sieciach bezprzewodowych.

5.5.3 Konfiguracja sieci

Przebiega dokładnie tak samo jak sieci kablowych. Poprostu zajrzyj do wczesniejszej sekcji tego rozdzialu.

5.6 Sieciowe systemy plików (NFS)

W tym momencie połączenie TCP/IP powinno działać bez zarzutu. Powinna istnieć możliwość pingownia komputerów w całej sieci lokalnej i jeśli masz własciwie skonfigurowaną bramkę być w stanie pingować komputery w internecie. Jak wiemy celem podłączenia komputera do sieci jest dostęp do informacji. Podczas gdy niektórzy podpinają komputery do sieci dla zabawy, wiele osób chce współdzielić zasoby i drukarki. Chcą mieć dostęp do dokumentacji w internecie i grania w gry sieciowe. Nawet jeżeli na maszynie jest zainstalowany i dzialajacy TCP/IP, jej funkcjonalność będzie wciąż bardzo ograniczona. Aby współdzielić pliki należy przemeiszczac je w tę i z powrotem używając czy to FTP czy SCP. Nie można przeglądać plikow na komputerze z Slackware z otoczenia sieciowego lub "Moich Miejsc Sieciowych" w komputerze z Widnowsem. Przydałby się również dostęp do plików na innych maszynach uniksowych.

Tak więc, chcielibyśmy móc używać *sieciowego systemu plików* (ang. *network file system*) aby umożliwić zrozumiały dostęp do plików na innych komputerach. Program, którego użyjemy do współpracy z informacjami przechowywanymi na naszym komputerze tak naprawde nie potrzebuje wiedziec na jakim komputerze znajduje sie określony plik; musi wiedziec tylko, że ten plik istnieje i jak sie do niego dostac. Wtedy to na system spada odpowiedzialnosc za umożliwienie dostępu do tego pliku przez lokalny i sieciowy system plików. Dwoma najbardziej rozpowszechnionymi sieciowymi systemami plików są SMB (jako implementacja Samby) i NFS.

5.6.1 SMB/Samba/CIFS

SMB (od Server Message Block) jest potomkiem starszego protokołu NetBIOS, który był używany przez IBM w ich LAN Managerze. Microsoft od zawsze był zainteresowany NetBIOSem i jego nastepcami (NetBEUI, SMB i CIFS). Projekt Samba istnieje od 1991, kiedy został napisany aby połączyć IBM PC (z protokołem NetBIOS) z serwerem Uniksowym. Obecnie SMB jest preferowaną metodą współdzielenia plików i drukarek w sieci na calym świecie ponieważ jest obsługiwana przez Windows.

Plikiem konfiguracyjnym samby jest `/etc/samba/smb.conf`; to jeden z najlepiej skomentowanych plików konfiguracyjnych jaki można znaleźć. Przykładowe wartości zostały ustawione tak by móc je wykorzystac w razie potrzeby. Jeśli potrzebujesz większej kontroli - strony manuala dla smb.conf sa niezastąpione. Samba jest bardzo dobrze udokumenotwana (w miejscach, o których mowa wyżej), nie ma potrzeby przytaczać w tym miejscu manuala. Poniżej przedstawiono podstawy.

`smb.conf` jest podzielony na kilka sekcji: jedna sekcja na zasoby, i globalna sekcja dla opcji które mają być używane powszechnie. Niektóre opcje obowiazują tylko w sekcji globalnej; niektore tylko poza sekcją globalna. Pamietaj, że ustawienia w sekcji globalnej mogą zostać “podmienione/nadpisane” przez kazda inna sekcje. Odwołaj się do stron manuala po więcej informacji.

Może zajść potrzeba wprowadzenia poprawek w `smb.conf` by odzwierciedlał on ustawienia twojej sieci LAN. Sugeruję modyfikacje elementów wymienionych poniżej:

```
[global]
# workgroup = NT-Domain-Name or Workgroup-Name, eg: LINUX2
workgroup = MYGROUP
```

Zmień nazwę grupy roboczej aby odzwierciedlała nazwę domeny w jakiej się znajdujesz.

```
# server string is the equivalent of the NT Description field
server string = Samba Server
```

Bedzie to nazwa twojego komputera ze Slackiem wsyświetlana w katalogu z otoczeniem sieciowym..

```
# Security mode. Most people will want user level security. See
# security_level.txt for details. NOTE: To get the behaviour of
# Samba-1.9.18, you'll need to use "security = share".
security = user
```

W większości przypadków istotne będzie ustawienie poziomu bezpieczeństwa. Z reguły jest to poziom użytkownika (user level).

```
# You may wish to use password encryption. Please read
# ENCRYPTION.txt, Win95.txt and WinNT.txt in the Samba
# documentation.
# Do not enable this option unless you have read those documents
encrypt passwords = yes
```

Jeśli nie włączone jest szyfrowanie haseł, nie bedziesz możliwości używać Samby z NT4.0, Win2k, WinXP, i Win2003. Wcześniejsze systemy operacyjne Windows nie wymagały szyfrowania do udostepniania plików.

SMB jest protokolem z uwierzytelnianianiem oznaczajacym, że musisz dostarczyć własciwą nazwę użytkownika i hasło aby skorzystać z usługi. Mowimy serwerowi Samby jakie nazwy użytkowników i hasła sa poprawne używajac komendy `smbpasswd`. `smbpasswd` (przy pomocy opcji wywołania) pozwala na dodanie maszyny albo tradycyjnego uzytkownika. (SMB wymaga podania nazwy NETBIOS komputerów które mają zostać dodane

```
Dodawanie uzytkownika do pliku /etc/samba/private/smbpasswd
# smbpasswd -a user
Dodawanie komputera do pliku /etc/samba/private/smbpasswd.
# smbpasswd -a -m machine
```

To bardzo ważne aby dodawana nazwa użytkownika lub komputera istniala w pliku `/etc/passwd`. Można to osiagnac przez prostą komendę `adduser`. Zauważ, ze kiedy używasz polecenia `adduser` aby dodac nazwę komputera musi ona zawierac znak dolara przy nazwie komputera (“`$`”). *Nie* powinno być to robione przez `smbpasswd`. `smbpasswd` dodaje znak dolara samoczynnie. Nie dodanie tego znaku w przypadku `adduser` zakończy się błędem dodawania nazwy komputera do Samby.

```
# adduser machine$
```

5.6.2 Network File System (NFS)

NFS (ang. Network File System) został oryginalnie napisany przez Suna dla ich implementacji Uniksa w Solarisie. Mimo, że jest znacznie łatwiej z niego korzystać (w porównaniu do SMB), jest też znacząco mniej bezpieczny. Podstawową luka w bezpieczenstwie NFS jest łatwosc podszycia sie pod ID użytkownika i grupy z jednego komputera na drugi. NFS jest pozbawiony autoryzacji. Przyszłe wersje protokołu NFS bedą miały wzmocnione bezpieczeństwo, ale nie jest to obecne w momencie pisania tego rozdziału.

Za konfigurację NFS odpowiedzialny jest plik `/etc/exports`. Kiedy załadujesz domyślny plik `/etc/exports` do edytora, zobaczysz pusty plik z dwiema liniami komentarza u góry. Należy dodać po linijce dla każdego katalogu, który ma zostać wyeksportowany. Dodatkowo w każdej takiej linii trzeba wylistować klienckie stacje robocze, które mają mieć do niego dostęp. Dla przykladu jesli chcemy eksportowac katalog `/home/foo` do stacji roboczej Bar, musimy tylko dodać linię:

```
/home/foo Bar(rw)
```

do naszego `/etc/exports`. Poniżej znajdziesz przykład ze strony manuala dla pliku `exports`:

```
# sample /etc/exports file
/               master(rw) trusty(rw,no_root_squash)
/projects       proj*.local.domain(rw)
/usr            *.local.domain(ro) @trusted(rw)
/home/joe       pc001(rw,all_squash,anonuid=150,anongid=100)
/pub            (ro,insecure,all_squash)
```

Jak widać opcji jest dośc dużo, jednak po przeanalizowaniu powyższego przykładu większość z nich powinna być zrozumiała.

NFS działa przy założeniu, że dany użytkownik na jednym komputerze ma te samo ID na wszystkich komputerach w obrębie sieci. Kiedy próbuje odczytac czy zapisać z klienta NFS na serwer NFS, przekazuje UID jako część żądania odczytu/zapisu. UID jest traktowany tak samo jakby żądanie odczytu/zapisu pochodzilo z komputera lokalnego. Jak się można domyślić, jeżeli ktoś byłby w stanie arbitralnie ustawić sobie UID mógłby dokonać baaaardzo złych rzeczy. Jako częściową zaporę przeciw temu, każdy katalog jest montowany z opcją `root_squash`. Zabieg ten powoduje mapowanie UIDów użytkowników podających się za roota na *różne* UIDy. Chroni to przed niepożądanym dostępem z prawami roota do katalogów wyeksportowanych. `root_squash` wydaje się być włączone domyslnie, jednak twórcy zalecają wyszczególnienie tej opcji w `/etc/exports`.

Możesz także eksportowac katalog bezpośrednio z linii poleceń na serwerze używając komendy `exportfs`:

```
# exportfs -o rw,no_root_squash Bar:/home/foo
```

Ta linia eksportuje katalog `/home/foo` do komputera "`Bar`" z uprawnieniami odczyt/zapis. Dodatkowo serwer NFS nie będzie nada opcji `root_squash`, co oznacza ze każdy użytkownik na Bar z UIDem równym "0" (UID roota) bedzie mieć te same prawa jak root na serwerze. Składnia wygląda dziwnie (normalnie składnia `computer:/katalog/plik`, oznacza, że odwołujesz się do pliku w katalogu na danym komputerze).

Wiecej informacji znajdziesz na stronach manuala `exportfs` .

Rozdział 6 Konfiguracja X'ów

Poczynając od Slackware 10.0, środowisko X Window dostarczane jest przez Xorg. X'y odpowiedzialne są za interfejs graficzny użytkownika. W przeciwieństiwe do Windows czy MacOS są niezależne od systemu operacyjnego .

System X Window implementowany jest przez wiele programów uruchamianych przez użytkownika. Dwa główne komponenty to serwer oraz menedżer okien (window manager). Serwer dostarcza niskopoziomowych

funkcji operujących na kartach video, więc jest zależny od konkretnego systemu. Menedżer okien znajduje się na samym szczycie serwera i dostarcza interfejsu użytkownika. Zaletą takiego podziału jest możliwość posiadania wielu interfejsów graficznych - wystarczy zmienić menedżer okien.

Konfiguracja X'ów może być niekiedy bardzo złożonym zadaniem. Powodem tego jest mnogość kart video dostępnych na architektury PC. W większości różnią się one używanym interfejsem programistycznym. Na szczęście wiele dzisiejszych kart obsługuje podstawowe standardy znane jako VESA. Jeżeli twoja karta znajduje się pośród nich, będziesz mógł używać X'ów. Wystarczy wpisać `startx` z lini poleceń.

W przypadku problemów z działaniem powyższego polecenia lub chęci wykorzystania specjalnych możliwości karty (takich jak akceleracja sprzętowa czy renderowanie sprzętowe 3-D) konieczna okaże się rekonfiguracja X'ów.

By skonfigurować X'y należy utworzyć plik `/etc/X11/xorg.conf` i go wyedytować. Zawiera on mnóstwo szczegółowych informacji na temat sprzętu wideo, myszy oraz monitora. Jest to złożony plik konfiguracyjny ale na szczęście jest kilka programów, które pomagają w jego spreparowaniu. Wspomnimy o kilku z nich w tym miejscu.

6.1 `xorgconfig`

Jest to prosty(!!!!) program oparty o menu, które przypomina to z instalacji systemu. Jego działanie polega na ustawieniu najlepszych wartości początkowych dla pliku `/etc/X11/xorg.conf` (w oparciu o posiadany sprzęt). Powinien on być dobrą bazą dla większości systemów. Co więcej powinien być poprawny i umożliwić działanie X'ów bez dokonywania w nim modyfikacji.

Jest to program konfiguracyjny przeznaczony dla zaawansowanych administratorów systemu. Poniżej przedstawione zostało jego działanie na przykładzie. Najpierw należy program uruchomić:

```
# xorgconfig
```

Wyświetlona zostanie pełnoekranowa informacja o `xorgconfig`. By przejść do następnego kroku należy wcisnąć **`ENTER`**. Polecenie `xorgconfig` poprosi cię o zweryfikowanie poprawności zmiennej `PATH`. Powinno być dobrze więc wciskamy **`ENTER`**.

Rysunek 6-1. `xorgconfig` Konfiguracja myszy

Z wyświetlonego menu wybierz swój typ myszy. Jeżeli nie uwzględniono twojej myszki na porcie szeregowym, wybierz protokół Microsoftu - jest najbardziej powszechny i prawdopodobnie zadziała. Następnie `xorgconfig` zapyta cię o chęć używania `ChordMiddle` oraz `Emulate3Buttons`. Szczegółowe opisy wymienionych opcji ujrzysz na ekranie. Należy wybrać je gdy środkowy przycisk myszy nie działa pod X'ami lub gdy twoja mysz posiada tylko dwa przyciski (`Emulate3Buttons` pozwala na symulację środkowego przycisku poprzez naciskanie prawego i lewego jednocześnie). Następnie wpisz nazwę swojego urządzenia myszy. Domyślnie jest to `/dev/mouse`. Powinno działać jakoże dowiązanie zostało skonfigurowane podczas instalacji Slackware. Jeżeli używasz GMP (Linux'owy serwer myszy) w trybie powtarzania (repeater mode), możesz ustawić typ myszy jako `/dev/gpmdata`. X'y będą otrzymywały informacje o myszy poprzez `gpm`. W niektórych przypadkach (szczególnie z myszy szynowej [busmice]) może to działać lepiej lecz większość użytkowników nie praktykuje tego.

`xorgconfig` zapyta cię o uaktywnienie specjalnych skrótów klawiszowych (key bindings). Jeżeli ich potrzebujesz odpowiedz “**`y`**”. Dla większości jednak opcja ta nie jest użyteczna. W przypadku wątpliwości należy również wbrać “**`n`**”.

Rysunek 6-2. `xorgconfig` Horizontal Sync

W następnej sekcji ustawiony zostanie zakres odświeżania dla twojego monitora. By rozpocząć jego konfigurację wciśnij **ENTER**. Na ekranie pojawi się lista typów monitorów - wybierz jeden z nich. Należy zwrócić szczególną uwagę na to by nie przekroczyć wartości określonych w specyfikacji monitora (istnieje ryzyko uszkodzenia sprzętu).

Rysunek 6-3. `xorgconfig` Vertical Sync

Określ zares odświeżania pionowego dla twojego monitora. Informacje na ten temat powinny być w insrtukcji monitora. xorgconfig poprosi o wpisanie łańcucha identyfikującego typ monitora w pliku `xorg.conf`. Wpisz tam cokolwiek (lub też nic).

Rysunek 6-4. `xorgconfig` Video Card

W następnym kroku istnieje możliwość przejrzenia bazy danych typów kart video. Należy to uczynić - odpowadamy "**y**" na zadane pytanie, po czym wybieramy odpowiednią kartę z listy. Jeżeli nie ma tam dokładnie takiej jaka znajduje się w twoim systemie postaraj się odnaleźć kartę z używającą chipsetu twojej karty.

Następnie należy poinformować `xorgconfig` o ilośći RAM'u na karcie video. `xorgconfig` będzie także próbował wyciągnąć jakiś opis karty - jeżeli masz ochotę możesz go umieścić w trzech wyświetlonych linijkach.

Kolejny krok to ustawienie rozdzielczości, która będzie używana. Ponownie wybranie wartości domyślnych powinno wystarczyć na początek. Później można wyedytować `/etc/X11/xorg.conf` i ustawić tryb 1024x768 (lub inny) jako domyślny.

W tym momencie `xorgconfig` zapyta Cię czy zachować plik konfiguracyjny. Po potwierdzeniu proces konfiguracji zostanie zakończony i X'y można będzie odpalić poleceniem `startx`.

6.2 `xorgsetup`

Drugim sposobem na konfigurację X'ów jest program `xorgsetup`, dostarczony razem z Dystrybucją.

Aby go uruchomić należy zalogować się jako root, a następne wpisać:

```
# xorgsetup
```

Jeżeli w systemie znajduje się plik `/etc/X11/xorg.conf` (chociażby dlatego, że X'y są już skonfigurowane) zostanie wyświetlone pytanie czy go zachować przed kontynuacją. Nazwa oryginalnego pliku zostanie zmieniona na `/etc/X11/xorg.conf.backup`.

6.3 xinitrc

Program `xinit`(1) jest faktycznie odpowiedzialny za uruchomienie X'ów. Jest on wywoływany przez `startx`(1), więc być może nie jest to takie oczywiste. W pliku konfiguracyjnym tego programu zawarte są informacje na temat programów (w szczególności menedżerów okien) uruchamianych podczas startu X'ów.

`xinit` sprawdza najpierw katalog domowy użytkownika w poszukiwaniu pliku `.xinitrc`. W wypadku powodzenia korzysta z niego. Jeżeli pliku nie ma używany jest (domyślny dla całego systemu) `/var/X11R6/lib/xinit/xinitrc`. Poniżej przedstawiono przykładowy plik `.xinitrc`:

```
#!/bin/sh
# $XConsortium: xinitrc.cpp,v 1.4 91/08/22 11:41:34 rws Exp $

userresources=$HOME/.Xresources
usermodmap=$HOME/.Xmodmap
sysresources=/usr/X11R6/lib/X11/xinit/.Xresources
sysmodmap=/usr/X11R6/lib/X11/xinit/.Xmodmap

# merge in defaults and keymaps

if [ -f $sysresources ]; then
    xrdb -merge $sysresources
fi

if [ -f $sysmodmap ]; then
    xmodmap $sysmodmap
fi

if [ -f $userresources ]; then
    xrdb -merge $userresources
fi

if [ -f $usermodmap ]; then
    xmodmap $usermodmap
fi

# start some nice programs

twm &
xclock -geometry 50x50-1+1 &
xterm -geometry 80x50+494+51 &
xterm -geometry 80x20+494-0 &
exec xterm -geometry 80x66+0+0 -name login
```

Wszystkie te bloki “if” są po to aby połączyć ustawienia konfiguracyjne z innych plików. Interesująca cześć pliku znajduje się na końcu gdzie uruchamiane są różne programy. W powyższym przykładzie sesja X'ów uruchomiona zostanie z menedżerem okien `twm`(1), zegarem i trzema terminalami. Zwróć uwagę na ostatnią linijkę pliku. Polecenie `exec` przed ostatnim uruchomieniem `xterm` zamienia aktualnie działającego shell'a (wykonującego skrypt) z tym z `xterm`(1). W rezultacie wyjście z `xterm` zakończy sesję X'ów.

Aby dopasować start X'ów skopiuj domyślny plik `/var/X11R6/lib/xinit/xinitrc` do `~/.xinitrc` i go wyedytuj dodając programy, które uważasz za potrzebne. W moim przypadku wygląda to prosto:

```
# Start the window manager:
exec startkde
```

Zauważ, że jest kilka plików `xinitrc.*` w katalogu `/var/X11R6/lib/xinit`. Powodem tego jest mnogość menedżerów okien i interfejsów graficznych użytkownika (GUI). Wybierz ten, który ci najbardziej odpowiada.

6.4 `xwmconfig`

Przez długie lata Unix był używany prawie wyłącznie jako system operacyjny pod serwery, z wyjątkiem superszybkich profesjonalnych stacji roboczych. Systemy Unix'owe nie były wcześniej używane przez “domowych użytkowników” i interfejsy użytkownika odzwierciedlały ten stan rzeczy. Były dość “toporne”, zaprojektowane do uruchamiania niewielu wówczas istniejących aplikacji graficznych (takich jak np. CAD i programy do obróbki obrazów). Zarządzanie plikami i systemem odbywało się praktycznie całkowicie przy pomocy linii poleceń. Różni dostawcy (Sun Microsystems, Silicon Graphics itd.) sprzedawali stacje robocze z naciskiem na spoisty “wygląd”, ale duża różnorodność narzędzi GUI używana przez developerów doprowadziła do nieuniknionego rozłamu w jednolitości desktopów. Nawet pasek przewijania nie musiał

wyglądać tak samo w dwóch różnych aplikacjach. Menu pojawiało się w prawie dowolnych miejscach. Przyciski (buttons) i checkbox'y także się różniły między sobą w zależności od aplikacji. Z kolorami było podobnie a dodatkowo najczęściej nie było możliwości ich zmiany. Do czasu gdy użytkownikami byli głównie zawodowi technicy, nie miało to wielkiego znaczenia.

Z nastaniem ery darmowych systemów Unix-owych liczba i różnorodność aplikacji graficznych spowodowała znaczący rozwój X'ów. Większość użytkowników przyzwyczaiło się jednak wcześniej do wyglądu i stylów które zostały zaproponowane przez Windows Microsoft'u czy MacOS Apple'a. To jego brak w aplikacjach opartych o X'y był głównym czynnikiem ograniczającym popularność systemu. W odpowiedzi na ten problem zapoczątkowane zostały dwa projekty open source: The K Desktop Environment (KDE) oraz GNU Network Object Model Environment (Gnome). Każdy z nich posiada niezliczoną liczbę aplikacji poczynając od pasków zadań przez menedżery plików, gry a kończąc na pakietach biurowych. Wszystkie napisane przy użyciu tych samych narzędzi do tworzenia interfejsów (GUI) i standardów, które powodują, że “środowisko” (desktop) sprawia wrażenie spójnego i funkcjonalnego.

Różnice w KDE i GNOME są dość subtelne. Każdy wygląda inaczej bo narzędzia do tworzenia GUI są różne. KDE bazowany jest na bibliotece Qt dostarczonej przez Troll Tech AS, podczas gdy GNOME używa GTK, zestawu narzędzi pierwotnie stworzonych dla programu GIMP (The GNU Image Manipulation Program). Jako niezależne projekty KDE i GNOME mają różnych programistów i projektantów. Każda z grup ma swój styl i swoją filozofię. Rezultat jednakże jest generalnie taki sam: spójne, zintegrowane środowisko i kolekcja aplikacji. Funkcjonalność, użyteczność i estetyka KDE i GNOME'a z powodzeniem może rywalizować z tym co spotyka się na innych systemach operacyjnych.

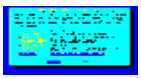
Najlepsze z tego wszystkiego jest to, że wymienione środowiska są darmowe. Oznacza to, że możesz mieć dowolne z nich lub nawet oba jednocześnie. Wybór należy do Ciebie.

Oprócz GNOME i KDE Slackware dostarcza dużą kolekcję menedżerów okien. Niektóre zaprojektowane są w taki sposób by emulować inny system operacyjny czy to w kwestii szybkości czy możliwości dopasowania do swoich potrzeb. Oczywiście można zainstalować dowolną ich ilość (albo wszystkie). Zachęcamy do eksperymentów.

Aby móc łatwo wybrać używane środowisko, Slackware zawiera program `xwmconfig`, który może być użyty do wybrania menedżera okien bądź środowiska graficznego. Aby go uruchomić należy wpisać:

```
% xwmconfig
```

Rysunek 6-5. Konfiguracja Środowiska Graficznego przy pomocy `xorgconfig`



Po uruchomieniu programu, pojawi się lista wszystkich zainstalowanych menedżerów okien i desktopów. Należy poprostu wybrać jeden. Odnosi się to każdego użytkownika w systemie (o ile domyślnie ustawiony menedżer mu nie odpowiada).

Następnie uruchamiamy X'y i jesteśmy gotowi do pracy.

6.5 `xdm`

Linux staje się coraz bardziej popularnym systemem na komputery domowe. Wielu użytkowników uzna za wygodne by maszyna bootowała się bezpośrednio do środowiska graficznego. Aby tego dokonać należy poinformować Slackware aby bootował się prosto do X'ów oraz wybrać odpowiedni menedżer logownania. Slackware dostarcza trzech takich narzędzi: `xdm`(1), `kdm`, oraz `gdm`(1).

`xdm` to graficzny menedżer logowania dostarczany razem z systemem X.org. Jest on obecny w każdym systemie lecz nie jest tak funkcjonalny jak programy alternatywne. `kdm` to menedżer dostarczany z KDE, `gdm` natomiast z

GNOME. Każdy z nich pozwoli na zalogowanie się jako dowolny użytkownik i wybór menedżera okien/desktopu.

Niestety Slackware nie oferuje żadnego ładnego programu takiego jak `xwmconfig` służącego do wybrania menedżera logowania. Jeżeli więc zainstalowane zostały wszystkie trzy, należy wyedytować plik konfiguracyjny w celu dokonania wyboru. Najpierw jednek omówione zostanie bootowanie od razu do graficznego środowiska.

Aby od razu po bootowaniu uruchamiane były X'y, należy wybrać runlevel 4. Runlevel'e są sposobem na poinformowanie `init`(8)'a by wykonał określone czynności podczas startu systemu. Dokonuje się tego poprzez edycję pliku konfiguracyjnego `init`'a - `/etc/inittab`.

```
# These are the default runlevels in Slackware:
## (Domyślne runlevele w slackware)
#   0 = halt
#   1 = single user mode
#   2 = unused (but conRysunekd the same as runlevel 3)
#   3 = multiuser mode (default Slackware runlevel)
#   4 = X11 with KDM/GDM/XDM (session managers)
#   5 = unused (but conRysunekd the same as runlevel 3)
#   6 = reboot

# Default runlevel. (Do not set to 0 or 6)
## Domyślny runlevel. (Nie ustawiać na 0 lub 6)
id:3:initdefault:
```

Aby kazać Slackware bootować się od razu do środowiska graficznego należy porpostu zmienić 3 na 4.

```
  # Default runlevel. (Do not set to 0 or 6)
  id:4:initdefault:
```

Teraz domyślnym runlevelem dla Slackware będzie 4. Wykonany zostanie plik `/etc/rc.d/rc.4`. W nim uruchomione zostaną X'y i wybrany menedżer logowania. W jaki sposób więc wybiera się ten ostatni? Jest kilka sposobów aby tego dokonać. Najpierw jednak warto przyjrzeć się plikowi `rc.4`.

```
  # Try to use GNOME's gdm session manager:
  if [ -x /usr/bin/gdm ]; then
    exec /usr/bin/gdm -nodaemon
  fi

  # Not there?  OK, try to use KDE's kdm session manager:
  if [ -x /opt/kde/bin/kdm ]; then
    exec /opt/kde/bin/kdm -nodaemon
  fi

  # If all you have is XDM, I guess it will have to do:
  if [ -x /usr/X11R6/bin/xdm ]; then
    exec /usr/X11R6/bin/xdm -nodaemon
  fi
```

Jak widać powyżej `rc.4` najpierw sprawdza czy `gdm` ma prawa do wykonywania. Jeżeli tak to go uruchamia. Drugi na liście jest `kdm` i w końcu `xdm`. Jeden ze sposobów na wybranie menedżea logowania jest usunięcie pozostałych. Należy do tego celu użyć narzędzia `removepkg` i usunąć cały pakiet. Aby dowiedzieć się więcej o `removepkg` zobacz Rozdział 18.

Opcjonalnie można zabrać prawa do wykonywania z plików które nie będą używane. Do tego celu służy polecenie `chmod` szerzej omawiane w Rozdział 9.

```
# chmod -x /usr/bin/gdm
```

Ostatnim sposobem jest zakomentowanie odpowiednich linii `rc.4`.

```
# Try to use GNOME's gdm session manager:
# if [ -x /usr/bin/gdm ]; then
#   exec /usr/bin/gdm -nodaemon
# fi

# Not there?  OK, try to use KDE's kdm session manager:
if [ -x /opt/kde/bin/kdm ]; then
  exec /opt/kde/bin/kdm -nodaemon
fi

# If all you have is XDM, I guess it will have to do:
if [ -x /usr/X11R6/bin/xdm ]; then
  exec /usr/X11R6/bin/xdm -nodaemon
fi
```

Każda linia zaczynająca się znakiem hash (#) uznawana jest jako komentarz i jest ignorowana przez powłokę. W powyższym przykładzie nawet jeżeli `gdm` byłby zainstalowany i wykonywalny shell (w tym wypadku `bash`) nie będzie tego sprawdzał.

Rozdział 7 Booting

Uruchamianie Linuksa może byc proste, ale też w pewnych wypadkach może sprawiać trudności. Wielu użytkowników instaluje na swoich komputerach Slackware jako jedyny system operacyjny - po włączeniu komputer jest gotów do działania. Jednak gdy na dysku znajduje się więcej niż jeden system operacyjny, to uruchomienie komputera może okazać się niezłą harówką. Z pomocą przychodzą LILO oraz Loadlin - zawarte w dystrybucji Slackware programy do uruchamiania Slackware. LILO moze działać zainstalowane na jednej z partycji na dysku twardym, w MBR dysku twardego albo też na dyskietce, co czyni go bardzo wszechstronnym narzędziem. Loadlin pracuje w środowisku DOS: kończy jego pracę i uruchamia Linuksa.

Innym znanym narzędziem do uruchamiania Linuksa jest GRUB. Nie jest to program włączony do dystrybucji Slackware ani też oficjalnie przez nią wspierany, dlatego też nie będziemy go tutaj omawiać. Slackware opiera się na programach dobrze już przetestowanych i dzialających pewnie, i tylko takie trafiają do dystrybucji. GRUB co prawda sprawuje się nieźle i oferuje pewne możliwości których nie ma LILO, niemniej jednak LILO od lat spełnia rzetelnie wszystkie zasadnicze zadania bootloader'a. Będąc młodszym, GRUB nie miał jeszcze dość czasu, by zapracować sobie na takie zaufanie jakim od lat cieszy się LILO. Jeżeli masz zamiar używać GRUB'a (być może znalazłeś go w innej dystrybucji Linuksa) - zapoznaj się z jego dokumentacją.

Ten rozdział zawiera opis używania programów LILO oraz Loadlin, dwóch bootloader'ow włączonych do dystrybucji Slackware. Ponadto wyjaśnia pewne typowe zasady bootowania, gdy na dysku twardym znajdują się dwa systemy operacyjne oraz opisuje jak poprawnie skonfigurować LILO powyższym przypadku.

7.1 LILO

LILO (skrot od: LInux LOader) to najpopularniejsze narzędzie do uruchamiania Linuksa. Posiada wiele opcji konfiguracyjnych i może być w łatwy sposób użyte do uruchamiania innych systemów operacyjnych.

Linux Slackware zawiera program `liloconfig` - narzędzie z graficznym interfejsem służące do konfiguracji LILO. Program ten po raz pierwszy jest uruchamiany podczas instalacji, jednak można wywołać go w dowolnym momencie użytkowania systemu poprzez podanie komendy `liloconfig`.

LILO czyta swoje ustawienia z pliku `/etc/lilo.conf`(5). Plik ten nie jest jednak czytany przy każdym uruchomieniu systemu - LILO odwołuje się do niego jedynie wtedy, gdy jest instalowane. LILO musi być przeinstalowane za każdym razem kiedy tylko zmieniony zostanie plik konfiguracyjny. Wiele problemów z LILO bierze się właśnie stąd, iż po dokonaniu zmian w pliku `lilo.conf` samo LILO nie zostaje ponownie przeładowane. `liloconfig` pomaga w łatwy sposób zbudować plik konfiguracyjny; jeżeli jednak wolisz ręcznie edytować plik `lilo.conf`, to przeinstalowanie LILO wymaga jedynie wpisania `/sbin/lilo` (jako `root`) w linii

poleceń.

Kiedy po raz pierwszy uruchomisz `liloconfig`, będzie ono wyglądało mniej więcej tak:

Rysunek 7-1. `liloconfig`

Jeżeli po raz pierwszy instalujesz LILO, najlepiej wybierz opcję "simple" (instalacja prosta). Jednakże jeżeli znasz już trochę LILO oraz Linuksa, opcja "expert" zapewne będzie dla ciebie lepsza. Wybranie opcji "simple" rozpoczyna proces konfiguracji LILO.

Jeżeli w jądro systemu jest wkompilowana obsługa framebuffer'a, `liloconfig` spyta się o rozdzielczość jakiej chcesz używac. Jest to także rozdzielczość używana przez framebuffer serwera graficznego XFree86. Jeżeli wolisz by konsola nie uruchamiała się w rozdzielczościach innych niż standardowe, wybierz opcję "normal" by ustawić zwykły rozmiar konsoli 80x25.

Następnym krokiem przy konfiguracji LILO jest ustalenie, gdzie chcesz je zainstalować. Prawdopodobnie jest to najważniejsza decyzja. Lista poniżej opisuje możliwe miejsca instalacji:

`Root`

> Root - opcja ta instaluje LILO na początku partycji Linuksa. Jest to opcja najbezpieczniejsza jeżeli tylko posiadasz inne systemy operacyjne na swoim komputerze, gdyż zapewnia, że żaden inny bootloader nie zostanie przypadkowo nadpisany. Minusem jest fakt, że LILO zostanie załadowane tylko wtedy, gdy dysk ten jest pierwszym dyskiem w komputerze. Jest to powód, dla ktorego wielu użytkowników tworzy maleńką partycję `/boot`, widoczną jako pierwszy dysk w ich systemie. Sposób ten pozwala jądru oraz plikom LILO być zainstalowanymi na początku dysku, gdzie LILO umie je znaleźć. Poprzednie wersje LILO zawierały niesławny błąd znany jako “1024 cylinder limit”. LILO było niezdolne do uruchomienia kerneli na partycjach umieszczonych poza 1024-tym cylindrem dysku twardego. Kolejne edycje LILO wyeliminowały ten problem.

`Floppy`

> Ta metoda jest jeszcze bezpieczniejsza niż poprzednia. Tworzy bootowalną dyskietkę, której możesz używać do uruchamiania swojego systemu. Utrzymuje to bootloadera całkowicie "z dala" od dysku twardego, tak więc chcąc uruchomić Linuksa - uruchamiasz komputer z tej własnie dyskietki. Złe strony takiego podejścia do sprawy są oczywiste: dyskietki miewają swoje humory, często się psują. Ponadto, bootloader nie jest już na stałe zawarty w komputerze. Jeżeli stracisz dyskietkę, będziesz musiał stworzyć nastepną by uruchomić system.

`MBR`

> Tej opcji będziesz używał, gdy Slackware jest jedynym systemem operacyjnym na twoim dysku, bądź też gdy będziesz używał LILO by wybrać, który z zainstalowanych systemow uruchomić. Jest to najbardziej preferowana metoda instalacji LILO i zadziała na praktycznie każdym komputerze.
>
> Wybranie tej opcji nadpisze inny bootloader, który znajduje się w MBR dysku twardego.

Po wyborze miejsca instalacji, `liloconfig` zapisze plik konfiguracyjny oraz zainstaluje LILO. I gotowe - LILO może już uruchamiać twój system. Jeżeli wybrałeś opcję "expert", zobaczyłeś specjalne menu. Pozwala ono odpowiednio dostroić plik `/etc/lilo.conf`: dodać kolejne systemy operacyjne do menu startowego LILO oraz przekazywać jądru specjalne parametry podczas ładowania systemu. Menu "expert" wygląda tak:

Rysunek 7-2. `liloconfig` Expert Menu

Jakakolwiek by nie była konfiguracja twojego systemu, ustawienie działającego bootloader'a nie jest trudne. `liloconfig` sprawia, że jest to wręcz dziecinnie proste.

7.2 LOADLIN

LOADLIN to alternatywny sposób bootowania systemu dostępny w Slackware. Jest to program DOS'owy i może być użyty do uruchomienia Linuksa spod działającego DOS'a. Do poprawnego uruchomienia Linuksa LOADLIN wymaga, by jądro systemu znajdowało się na partycji DOS'owskiej.

Podczas instalacji LOADLIN zostanie skopiowany do domowego katalogu root'a jako plik .ZIP. Nie ma dla niego żadnego procesu instalacji - należy jedynie skopiować na partycję DOS'a plik jądra Linuksa (zwykle `/boot/vmlinuz`) oraz plik LOADLIN z katalogu root'a.

LOADLIN okaże się najbardziej użyteczny, gdy planujesz utworzyć menu wyboru systemu operacyjnego na partycji DOS'a. Menu takie dodane do pliku `AUTOEXEC.BAT` pozwala wybierać między uruchomieniem Linuksa lub DOS'a. Wybór Linuksa spowoduje uruchomienie najpierw programu LOADLIN, który sam już zajmie się uruchomieniem Slackware'a. Przykładowy plik `AUTOEXEC.BAT` pod Windows 95, tworzący wystarczające menu, może wyglądać tak:

```
@ECHO OFF
SET PROMPT=$P$G
SET PATH=C:\WINDOWS;C:\WINDOWS\COMMAND;C:\
CLS
ECHO Wybierz system do uruchomienia:
ECHO.
ECHO [1] Slackware Linux
ECHO [2] Windows 95
ECHO.
CHOICE /C:12 "Wybrano: -> "
IF ERRORLEVEL 2 GOTO WIN
IF ERRORLEVEL 1 GOTO LINUX
:WIN
CLS
ECHO Uruchamiam Windows 95...
WIN
GOTO END
:LINUX
ECHO Uruchamiam Slackware Linux...
CD \LINUX
LOADLIN C:\LINUX\VMLINUZ ROOT=<partycja z jądrem Linuksa> RO
GOTO END
:END
```

Oczywiście, w swoim systemie jako partycję na której znajduje się jądro Linuksa podasz, np. `/dev/hda2`. Zawsze możesz użyć LOADLIN'a z linii poleceń w sposób analogiczny do przedstawionego przed chwilą. Dokumentacja LOADLIN'a zawiera wiele pomocnych przykładów.

7.3 Dual-booting

Wielu użytkowników na swoich komputerach posiada oprócz Slackware jeszcze jeden system operacyjny. Poniżej opisaliśmy typowe sposoby uruchamiania komputera w przypadku, gdy na dysku znajdują się dwa systemy operacyjne.

7.3.1 Windows

Komputer z zainstalowanymi systemami MS Windows oraz Linuksem to chyba najczęściej spotykany układ. Wybór systemu operacyjnego do uruchomienia można zrealizować na wiele sposobów, jednak ta sekcja opisze dwa z nich.

Zdarza się, że użytkownik instalujący dwa systemy operacyjne przygotuje sobie dokładny plan w którym wszystko działa idealnie, jednak nieświadomie instaluje systemy w złej kolejności. Ważną sprawą jest zrozumienie, iż dla właściwego współdziałania systemy powinny być instalowane w ściśle określonym porządku. Linux zawsze pozwala kontrolować to, co zostanie zapisane do MBR dysku twardego (o ile w ogóle chcemy coś zapisywać). Dlatego też sensownie jest instalować Linuksa na samym końcu. Windows, dla odmiany, powinien być zainstalowany jako pierwszy, gdyż jest to system który bez względu na wszystko i tak nadpisze MBR własnymi danymi, nie zważając na to co mógł zapisać tam instalowany wcześniej Linuks.

7.3.1.1 Jak używać LILO

Większość ludzi chce używać LILO do wyboru między uruchomieniem Linuksa a Windowsa. Jak powiedziano wyżej, należy w takim wypadku najpierw zainstalować Windows, a następnie Linuksa.

Powiedzmy, że masz 40 GB dysk twardy i jest to jedyny dysk w twoim komputerze. Przyjmijmy też, że połowę dysku chcesz przeznaczyć na Windows'a, a drugą połowę przydzielić Linuksowi. Przykład ten posłuży nam do pokazania, jak rozwiązać problem uruchamiania Linuksa i Windowsa.

```
20GB   Windows boot (C:)
1GB   Linux root (/)
19GB  Linux /usr (/usr)
```

Potrzebne też będzie przydzielenie adekwatnej przestrzeni na partycję wymiany dla Linuksa (ang. Linux swap). Niepisana reguła mówi, by użyć w tym celu obszaru dysku o rozmiarze równym podwojonej ilości pamięci RAM zainstalowanej w systemie. Tak więc system z 64MB RAM'u będzie miał 128MB plik wymiany, itp. Odpowiedni rozmiar pliku wymiany jest tematem wielu gorących dyskusji na IRC'u oraz Usenet'cie. Nie istnieje “jedyna prawdziwa” zasada wyboru rozmiaru pliku wymiany, niemniej jednak stosowanie wyżej wspomnianej reguły powinno być wystarczające.

Gdy ustaliłeś już podział dysku na partycje, rozpocznij instalację Windows'a. Gdy ten będzie już działał, przyjdzie czas na instalację Linuksa oraz LILO. W naszym przykładzie do poprawnej instalacji LILO należy wybrać tryb "expert".

Rozpoczynamy konfigurację LILO. Najlepszym wyjściem będzie zainstalowanie LILO w obszarze MBR dysku twardego tak, by możliwy był wybór między dwoma zainstalowanymi systemami. W menu wybieramy dodanie obu partycji: Linuksa i Windowsa (bądź DOS'a). Teraz można już instalować LILO.

Przy następnym uruchomieniu komputera LILO powinno wyświetlić menu pozwalające na wybór pomiędzy zainstalowanymi systemami. Wybierz nazwę systemu do uruchomienia (nazwy te zostały ustalone podczas konfiguracji LILO).

LILO posiada ogromne możliwości. Nie jest bootloaderem uruchamiającym jedynie Linuksa bądź DOS'a - potrafi uruchomić praktycznie każdy system. Wiele przydatnych informacji dostarczają strony podręcznika systemowego man dotyczące `lilo`(8) oraz `lilo.conf`(5).

A co zrobić, jeżeli LILO nie działa? Istnieją powody, dla których na pewnych konkretnych maszynach LILO nie uruchomi się. Na szczęście, istnieje inny sposób uruchamiania Linuksa i Windowsa.

7.3.1.2 Używanie LOADLIN

Metoda ta może być użyta gdy LILO nie uruchamia się na twoim systemie, bądź też gdy po prostu nie chcesz używać LILO. Jest także idealna dla osób, które często przeinstalowują Windows, gdyż przy każdej reinstalacji

Windows nadpisuje MBR swoimi danymi, w konsekwencji niszcząc pliki LILO. Taki problem nie istnieje, gdy używa się LOADLIN'a. Ale metoda ta ma też minus: może posłużyć do uruchomienia jedynie Linuksa.

Jeżeli planujesz używanie LOADLIN'a, nie obowiązuje cię podana wyżej kolejność instalacji systemów. Instaluj je w dowolnej kolejności. Niemniej jednak uważaj, co zapisujesz w obszar MBR dysku twardego. LOADLIN wymaga by partycja Windowsa była bootowalna. Tak więc podczas instalacji Slackware'a upewnij się, że pominąłeś instalację LILO.

Po zainstalowaniu obydwu systemów operacyjnych przekopiuj plik `loadlinX.zip` (gdzie `X` jest numerem wersji, np. `16a`) z katalogu domowego root'a na partycję Windowsa. Ponadto trzeba też skopiować tam obraz jądra Linuksa. Obydwie czynności należy wykonać spod Linuksa. Poniższy przykład ilustruje, jak to zrobić:

```
# mkdir /win
# mount -t vfat /dev/hda1 /win
# mkdir /win/linux
# cd /root
# cp loadlin* /win/linux
# cp /boot/vmlinuz /win/linux
# cd /win/linuz
# unzip loadlin16a.zip
```

Wykonanie powyższych poleceń spowoduje utworzenie katalogu `C:\LINUX` na partycji Windowsa (oczywiście przy założeniu, że partycja ta pod linuksem widoczna jest jako `/dev/hda1`) oraz przekopiuje tam rzeczy potrzebne do działania LOADLIN'a. Gdy wszystko jest już zrobione, uruchom ponownie komputer. Czas na ostatni etap konfiguracji LOADLIN'a.

Pod Windowsem przejdź do linii poleceń DOS'a. Przede wszystkim trzeba sprawić, by system nie uruchamiał się w trybie graficznym. Wykonaj następujące polecenia:

```
C:\> cd \
C:\> attrib -r -a -s -h MSDOS.SYS
C:\> edit MSDOS.SYS
```

W tak otwartym pliku dodaj następującą linijkę:

```
BootGUI=0
```

Zachowaj plik i opuść edytor. Teraz dodamy do pliku `C:\AUTOEXEC.BAT` menu pozwalające na wybór systemu do uruchomienia. Poniższe linie pokazują, jak może wyglądać przykładowy fragment pliku `AUTOEXEC.BAT` odpowiedzialny za nasze menu:

```
cls
echo System Boot Menu
echo.
echo 1 - Linux
echo 2 - Windows
echo.
choice /c:12 "Wybrano: -> "
if errorlevel 2 goto WIN
if errorlevel 1 goto LINUX
:LINUX
cls
echo "Uruchamiam Linuksa..."
cd \linux
loadlin c:\linux\vmlinuz root=/dev/hda2 ro
goto END
:WIN
cls
echo "Uruchamiam Windows'a..."
win
goto END
:END
```

Kluczową rolę pełni linia w której wywoływany jest LOADLIN. Podajemy tam obraz jądra do uruchomienia, wskazujemy partycję Linuksa oraz żądamy zamontowania jej w trybie tylko do odczytu.

Narzędzia niezbędne do wykonania powyższych dwóch metod dostępne są w dystrybucji Slackware Linux. Jest wiele innych bootloaderów na rynku, jednak te dwa są wystarczające w znakomitej większości wypadków.

7.3.1.3 Przestarzałe hackowanie Windows NT

Przedstawimy teraz najmniej chyba znany sposób na dual-booting. W dawnych czasach LILO nie potrafiło uruchomić Windowsa NT, wymagając od użytkowników Linuksa hackowania NTLDR, który to ze swej strony sprawiał znacznie więcej problemów niż uruchamianie Windowsów z serii 9x. Dziś te czasy to historia, LILO od wielu lat bezproblemowo uruchamia systemy Windows NT/2000/XP/2003. Tak więc hackowanie Windows NT ma sens jedynie wtedy, gdy używasz naprawdę starej maszyny. Oto kolejne kroki tej metody:

1. Zainstaluj Windows NT
2. Zainstaluj Linuksa. Upewnij się, że LILO jest zainstalowane na partycji Linuksowej.
3. Zapisz pierwsze 512 bajtów partycji linuksowej do pliku na partycji Windowsa NT
4. Edytuj `C:\BOOT.INI` pod Windows NT, dodając partycję Linuksa.

Zainstalowanie Windowsa NT nie powinno sprawiać problemów, gorzej może być z Linuksem. Od tego momentu zaczyna się ciekawa zabawa. Pobranie pierwszych 512 bajtów partycji linksowej jest łatwiejsze niż się wydaje (musisz być pod Linuksem, by tego dokonać). Przy założeniu, iż twoją partycją linuksową jest `/dev/hda2`, wykonaj te komendy:

```
# dd if=/dev/hda2 of=/tmp/bootsect.lnx bs=1 count=512
```

Gotowe. Teraz trzeba przekopiować nowoutworzony plik bootsect.lnx na partycję Windowsa NT. I tu pojawia się kolejny problem: Linuks nie posiada stabilnego wsparcia dla systemu plików NTFS. Tak więc, jeżeli instalowałeś Windows NT na partycji NTFS, będziesz musiał najpierw skopiować ten plik na dyskietkę i za jej pomocą przenieść go pod Windows NT. Jeżeli Windows NT znajduje się na partycji FAT, możesz bezproblemowo podmontować ją pod Linuksem i skopiować plik bezpośrednio. Jakiego sposobu byś nie użył, cel jest jeden: skopiowanie pliku `/tmp/bootsect.lnx` z partycji linuksowej do `C:\BOOTSECT.LNX` na partycji Windows NT.

Ostatnim krokiem jest dodanie odpowiedniej opcji do menu startowego Windowsa NT. Pod tym systemem uruchom linię poleceń i wykonaj następujące polecenia:

```
C:\WINNT> cd \
C:\> attrib -r -a -s -h boot.ini
C:\> edit boot.ini
```

Dopisz tę linię na końcu pliku:

```
C:\bootsect.lnx="Slackware Linux"
```

Zapisz zmiany i opuść edytor. Kiedy ponownie uruchomisz Windows NT, zmodyfikowane menu pozwoli Ci na uruchomienie Linuksa.

7.3.2 Linux

Tak, ludzie naprawdę instalują po kilka Linuksów na swoich dyskach. Jest to zdecydowanie najłatwiejszy

dual-booting. Po prostu, używając LILO, dodaj więcej odpowiednich wpisów do pliku `/etc/lilo.conf`. To wszystko, co trzeba tu zrobić.

Rozdział 8 The Shell

W środowisku graficznym interfejs systemu jest tworzony przez program odpowiedzialny za okna, menu, paski przewijania itp. W środowisku konsoli za tworzenie interfejsu użytkownika odpowiedzialna jest powłoka. Powłoka interpretuje polecenia użytkownika przekazywane do systemu oraz, w ogólności, robi wiele pożytecznych rzeczy. Tuż po zalogowaniu się do systemu (zostało ono opisane poniżej w tym rozdziale) użytkownik dostaje "do swej ręki" powłokę i może rozpocząć pracę. Ten rozdział należy traktować jako wprowadzenie do shella. Przybliża najbardziej znaną powłokę: bash (the Bourne Again SHell). Dla uzyskania informacji bardziej szczegółowych niż przedstawione w tym wprowadzeniu, przejrzyj stronę manuala `bash`(1).

8.1 Użytkownicy

8.1.1 Logowanie

Tak więc po udanym uruchomieniu komputera widzisz coś mniej więcej takiego:

```
Welcome to Linux 2.4.18
Last login: Wed Jan   1 15:59:14 -0500 2005 on tty6.
darkstar login:
```

Cóż... nikt dotąd nie wspominał o logowaniu. I co to takiego ten darkstar? Spokojnie, najprawdopodobniej nie stworzyłeś przypadkowego połączenia międzyprzestrzennego ze sztucznym księżycem Imperium. (Obawiam się także, iż protokół obsługi takiego połączenia nie jest aktualnie obsługiwany przez jądro Linuksa. Być może jądra serii 2.8 poprawią to jakże oczywiste niedopatrzenie). Pewne jest za to, iż darkstar to nazwa jednego z naszych komputerów oraz ta właśnie nazwa ustawiona jest domyślnie w świeżo zainstalowanych systemach. Jeżeli ustaliłeś podczas instalacji własną nazwę dla swojego komputera, to ją właśnie teraz widzisz zamiast tajemniczego darkstar.

Słówko o loginie... Jeżeli jest to pierwsze uruchomienie systemu, zaloguj się jako `root`. Pojawi się pytanie o hasło; jeżeli zosatało ono uswawione podczas instalacji, wpisz je. Jeżeli nie, po prostu naciśnij ENTER. Gotowe - logowanie do systemu zakończone!

8.1.2 Root: Superużytkownik

OK - kim, lub raczej *czym* jest `root`? I dlaczego posiada konto w *twoim* systemie?

Śpieszymy z wyjaśnieniem. W świecie Uniksów oraz systemów im pokrewnych (jak np. Linux) istnieje podział na użytkowników oraz użytkowników. Bardziej szczegółowo wyjaśnimy to już w najbliższym czasie, na razie należy jedynie stwierdzić, iż `root` to użytkownik nad użytkownikami; `root` w systemie jest wszechmogący i wszechwiedzący, i każdy słucha się `roota`. Inna sytuacja jest niedopuszczalna. `root` jest nazywany "superużytkownikiem". I prawidłowo. No i najlepsze w tym wszystkim jest to, iż `root` to własnie *ty*.

Fajnie, co nie?

Być może jeszcze to nie takie oczywiste: ale tak, to naprawdę świetna sprawa, gdyż jako root masz prawo do robienia absolutnie wszystkiego, na co ma ochotę (włącznie z niszczeniem wszystkiego). Proponujemy teraz przeskoczyć do Sekcja 12.1.1. Przeczytawszy jak dodać nowego użytkownika do systemu, zalecamy wykonanie tego i zalogowanie się na tak utworzone konto. Mądrość wieków mówi, że logować się na konto superużytkownika należy jedynie wtedy, gdy jest to absolutnie niezbędne. Chodzi o to, by jak najbardziej

zminimalizować prawdopodobieństwo przypadkowego zepsucia czegoś.

Przy okazji: jeżeli jesteś już zalogowany na jakimś koncie i zachodzi potrzeba uzyskania uprawnień `roota` nie ma problemu. Po prostu wydaj komendę su(1). Zostanie wyświetlone pytanie o hasło `roota`. Jako superużytkownik będziesz zalogowany tak długo, aż nie wylogujesz się - poleceniem `exit` lub `logout`. Możesz też za pomocą polecenia `su` zalogować się na konto innego użytkownika, oczywiście przy założeniu iż znasz hasło tego użytkownika. I tak wpisanie np. `su boogi` spowodowałoby przejście na moje konto.

Ciekawostka: `root` może zalogować się na dowolne konto. I nie potrzebuje do tego znajomości hasła.

8.2 Linia poleceń

8.2.1 Uruchamianie programów

Trudno cokolwiek zrobić na komputerze bez możliwości uruchamiania programów. Oczywiście zawsze można użyć komputera w charakterze podpórki do czegoś bądź też przytrzymać nim drzwi tak, by się nie otwierały, niektórzy też włączą komputer tylko i wyłącznie by posłuchać delikatnego szumu pracującej maszyny - ale to wszystko do czego zdolny jest komputer bez programów. I wszyscy zgodzimy się z tym, iż bycie szumiącą podstawką pod drzwi to nie jest to, co przyniosło komputerom osobistym popularność, którą dziś się cieszą.

Pamiętasz jak wspominaliśmy o tym, iż w Linuksie praktycznie wszystko jest przedstawiane pod postacią plików? Ta sama zasada obowiązuje programy. Każde polecenie, które wykonasz (a które nie jest wbudowane w powłokę), urzęduje gdzieś na dysku w postaci pliku. Programy uruchamia się podając ścieżkę dostępu do nich.

Dla przykładu przypomnij sobie polecenie `su` przedstawione w poprzednim rozdziale. Program ten znajduje się w katalogu `/bin`; uruchomi go wpisanie `/bin/su`.

Dlaczego więc działa wpisanie tylko `su`? Przecież nigdzie nie podajemy, że program ten jest w katalogu `/bin`. Równie dobrze mógłby znajdować się on chociażby w `/usr/local/share`, czyż nie? W jaki sposób powłoka *wie*, gdzie szukać? Odpowiedzią na to pytanie jest istnienie zmiennej środowiskowej `PATH` (z ang. ścieżka); większość powłok posiada albo taką właśnie zmienną, albo coś pełniącego jej rolę. W ogólności zmienna `PATH` zawiera listę katalogów przeglądanych w poszukiwaniu programów, których uruchomienia żądasz. Tak więc, kiedy wydajesz komendę `su`, twoja powłoka przegląda kolejne katalogi szukając programu o nazwie `su`. Uruchamiany jest pierwszy znaleziony. Tak dzieje się zawsze, gdy wydasz komendę nie podawszy pełnej ścieżki dostępu. Jeżeli otrzymasz komunikat ““`Command not found`”” (z ang. 'polecenie nie znalezione') to znaczy, że pożądany program, znajduje się w katalogu nie wymienionym w zmiennej `PATH` (bądź też w ogóle nie istnieje na dysku...). `PATH` jest jedną z wielu zmiennych środowiskowych. Zmiennym tym bliżej przyjrzymy się w Sekcja 8.3.1.

I na koniec jeszcze jedna uwaga: “`.`” jest skrótowym określeniem bieżącego katalogu. Oznacza to, iż np. gdy jesteś wewnątrz `/bin`, to wpisanie `./su` będzie miało dokładnie ten sam skutek, co podanie pełnej ścieżki dostępu, czyli `/bin/su`.

8.2.2 Wildcards - znaki uniwersalne

Praktycznie każda powłoka rozpoznaje pewne znaki jako podstawienia albo skróty zastępujące dowolne ciągi znaków. Są to tzw. znaki uniwersalne (z ang. “wildcards”; w polskiej literaturze często tłumaczone jako “znaki globalne”). Dwa najbardziej znane to `*` oraz `?`. Tradycyjnie, znak `?` służy do zastępowania dowolnego innego znaku. Dla przykładu: załóżmy, iż mamy katalog z trzema plikami: `ex1.txt`, `ex2.txt` oraz `ex3.txt` i chcemy przekopiować (używając polecenia `cp`, poznanego w Sekcja 10.5.1) całą trójkę do katalogu, powiedzmy `/tmp`. Tak więc cierpliwie w linii poleceń wpisujemy `cp ex1.txt ex2.txt ex3.txt /tmp`... ale to dużo roboty. O wiele łatwiej jest napisać `cp ex?.txt /tmp`; znak uniwersalny `?` stanowi dopasowanie do występujących w nazwach plików znaków “1”, “2” oraz “3” i podczas tego kopiowania będzie zastąpiony właśnie nimi.

Twierdzisz, że i to *ciągle* zbyt długo? Masz rację. To okropne; przecież mamy prawo pracy chroniące nas przed takimi rzeczami. Na szczęście, mamy też możliwość używania drugiego znaku specjalnego, czyli `*`. Jak wspomniano, `*` zastępuje "dowolny ciąg znaków" (wliczając w to także brak znaków). Tak więc, jeżeli wspomniane chwilę temu trzy pliki są jedynymi w tym katalogu, to możemy po prostu napisać `cp * /tmp` i w ten sposób jednym rzutem skopiujemy całą trójkę. Przypuśćmy teraz, iż w danym katalogu znajdują się jeszcze dwa pliki, pierwszy o nazwie `ex.txt` oraz drugi, `hejaz.txt`. Razem z wcześniej wspomnianą trójką chcemy jeszcze skopiować plik `ex.txt`, nie chcemy jednak kopiować `hejaz.txt`; `cp ex* /tmp` wykona nasze żądanie.

`cp ex?.txt /tmp` skopiuje oczywiście trzy znane nam już pliki; w nazwie `ex.txt` nie ma znaku, który mógłby być zastąpiony przez `?`, tak więc plik ten zostanie pominięty...

Następne powszechnie używane znaki specjalne to para nawiasów kwadratowych: `[ ]`. Wszystkie znaki umieszczone wewnątrz nich będą kolejno podstawione, by znaleźć pasujące do wzorca dopasowania. Brzmi niezrozumiale? Spokojna głowa, już wyjaśniamy na przykładzie. Przypuśćmy, że mamy katalog zawierający 8 plików: `a1`, `a2`, `a3`, `a4`, `aA`, `aB`, `aC`, oraz `aD` i chcemy spośród nich wybrać tylko te kończące się cyframi; `[ ]` zrobią to dla nas.

```
% ls a[1-4]
a1 a2 a3 a4
```

W powyższym przykładzie zażądaliśmy dopasowania wszystkich wartości pomiędzy 1 a 4. Gdy będziemy chcieli wyświetlić np. `a1`, `a2` oraz `a4` to żądane wartości oddzielmy przecinkami:

```
% ls a[1,2,4]
a1 a2 a4
```

OK, z cyframi problemów nie ma; nasuwa się pytanie "Dobra, a co z literami?" Linux jest wrażliwy na wielkość znaków (z ang. case-sensitive). Oznacza to, iż `a` oraz `A` to dwa całkiem różne znaki, powiązane ze sobą jedynie w twoim umyśle. Duże litery zawsze występują przed małymi, tak więc `A` oraz `B` znajdują się przed `a` oraz `b`. Kontynuując poprzedni przykład, jeżeli chcemy wylistować pliki `a1` oraz `A1`, wystarczy odpowiednio użyć nawiasów `[ ]`:

```
% ls [A,a]1
A1 a1
```

Uważaj, by nie pomylić łącznika "-" z przecinkiem, gdyż możesz otrzymać wynik inny od zamierzonego:

```
% ls [A-a]1
A1 B1 C1 D1 a1
```

Oczywiście można też używać łączników i przecinków w sposób kombinowany:

```
% ls [A,a-d]
A1 a1 b1 c1 d1
```

8.2.3 Przekierowywanie wejścia/wyjścia

Tu zaczyna się coś naprawdę świetnego!

```
% ps > blargh
```

Widziałeś kiedyś coś takiego? Wiesz, co to znaczy? To ja, uruchamiam sobie `ps` by wyświetlić listę procesów uruchomionych w systemie (polecenie `ps` zostało opisane w Sekcja 11.3). Ale to przecież nic specjalnego. Gwoździem programu jest przekierowanie (z ang. "redirection") `> blargh`, mówiące powłoce "weź wyjście z

`ps` i zapisz je do pliku `blargh`". Proste i przydatne, prawda? Ale przygotuj się, przed nami jeszcze ciekawsze rzeczy.

```
% ps | less
```

Powyższe kieruje strumień wyjściowy (ang. "piping") programu `ps` na wejście programu `less`, dzięki czemu mogę je sobie przejrzeć bez pośpiechu.

```
% ps >> blargh
```

Oto trzeci powszechnie znany i używany rodzaj przekierowania; robi dokładnie to samo, co ">", z tą drobną różnicą, że ">>" dopisze wyjście `ps` do pliku `blargh`, jeżeli plik istnieje. Jeżeli nie - zostanie tak jak przy użyciu ">" utworzony. (Użycie ">" gdy plik `blargh` istnieje, spowoduje nadpisanie jego zawartości.)

Jest też jeszcze operator "<", który oznacza "pobierz na wejście to, co stoi za operatorem" - ale nie jest on już tak często używany.

```
% fromdos < dosfile.txt > unixfile.txt
```

Przekierowywanie staje się naprawdę zabawne, kiedy zaczynasz mieszać różne warianty:

```
% ps | tac >> blargh
```

Taki ciąg poleceń uruchomi `ps`, odwróci kolejność linii na jego wyjściu oraz dopisze je w tak zmienionej kolejności do pliku `blargh`. Możesz wymieszać przekierowania w takiej ilości jak tylko chcesz; jednakże nie zapominaj, iż zawsze są one interpretowane od lewej do prawej.

Przeczytaj strony manuala dla `bash`(1) by dowiedzieć się więcej na temat przekierowań.

8.3 BASH - The Bourne Again Shell

8.3.1 Zmienne środowiskowe

LinuX to złożona bestia. Trzeba w nim zwracać uwagę na wiele rzeczy, cała masa detali wpływa na codzienną pracę twojego systemu oraz interakcję z programami (o wielu z tych detali być może nie wiesz, że istnieją). Nikt nie chce przekazywać masy opcji do każdego uruchamianego programu, mówić jakiego terminala się używa, jaka jest nazwa komputera, jak powinien wyglądać znak zachęty...

Dla ułatwienia życia użytkownicy otrzymują przydatne coś, zwane środowiskiem. Środowisko definiuje warunki, w których uruchamiają się programy; niektóre z tych warunków są zmiennymi, które może definiować wedle własnych potrzeb użytkownik (jest to prawidłowość właściwa tylko Linuksom). Praktycznie każda powłoka ma swoje własne zmienne środowiskowe (jeżeli nie ma - to najpewniej jest to powłoka niezbyt przydatna). Przyjrzymy się teraz pewnym poleceniom, które bash udostępnia do manipulowania swoimi zmiennymi środowiskowymi.

Polecenie `set` wpisane bez żadnych parametrów wypisze wszystkie aktualnie zdefiniowane zmienne środowiskowe, wraz z przyporządkowanymi im wartościami. Tak jak większość poleceń wbudowanych (ang. "built-ins") w powłokę `bash`, potrafi też robić więcej rzeczy - po podaniu odpowiednich parametrów. Tego jednak tematu nie będziemy rozwijać; dla zrozumienia istoty rzeczy wystarczy przeczytać stronę manuala `bash`(1). Przykład 8-1 pokazuje fragment wyjścia polecenia `set` uruchomionego na komputerze jednego z autorów (mowa o autorach oryginału angielskiego - przyp. tłum.). Zwróćcie uwagę na zmienną `PATH`, opisaną kilka akapitów wcześniej. Programy umieszczone w wylistowanych tam katalogach mogą być uruchamiane poprzez proste wpisanie tylko ich nazwy.

Przykład 8-1. Wypisanie zmiennych środowiskowych przy pomocy polecenia `set`

```
% set
PATH=/usr/local/lib/qt/bin:/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:
/usr/openwin/bin:/usr/games:.:/usr/local/ssh2/bin:/usr/local/ssh1/bin:
/usr/share/texmf/bin:/usr/local/sbin:/usr/sbin:/home/logan/bin
PIPESTATUS=([0]="0")
PPID=4978
PS1='\h:\w\$ '
PS2='> '
PS4='+ '
PWD=/home/logan
QTDIR=/usr/local/lib/qt
REMOTEHOST=ninja.tdn
SHELL=/bin/bash
```

```
% unset VARIABLE
```

`unset` usunie podaną jako parametr zmienną, razem z jej wartością; `bash` zapomni, iż zmienna taka w ogóle kiedykolwiek istniała. (Nie martw się - dopóki jest to zmienna zdefiniowana na potrzeby jednej konkretnej sesji powłoki, to w innej sesji najprawdopodobniej zostanie zdefiniowana ponownie).

```
% export VARIABLE=some_value
```

Poznaj bardzo przydatne polecenie `export`. Używając go, nadajesz zmiennej środowiskowej `VARIABLE` wartość "`some_value`"; jeżeli zmienna `VARIABLE` nie istniała wcześniej - to teraz już istnieje. Jeżeli do `VARIABLE` była już przypisana jakaś wartość... cóż, właśnie została zmieniona. Nie brzmi to dobrze, gdy np. próbujesz jedynie dodać katalog do zmiennej `PATH`. W takim wypadku najlepiej będzie napisać:

```
% export PATH=$PATH:/some/new/directory
```

Zwróć uwagę na użycie w powyższym poleceniu zapisu `$PATH`. Jeżeli chcesz, by `bash` zinterpretował zmienną (operował nie na jej nazwie, lecz na wartości w niej przechowywanej) to dodaj znak `$` przed nazwę zmiennej. Dla przykładu: polecenie `echo $PATH` w moim przypadku wypisze:

```
% echo $PATH
/usr/local/lib/qt/bin:/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:
/usr/openwin/bin:/usr/games:.:/usr/local/ssh2/bin:/usr/local/ssh1/bin:
/usr/share/texmf/bin:/usr/local/sbin:/usr/sbin:/home/logan/bin
```

8.3.2 Dopełnianie za pomocą tabulatora

Uwaga! Fajny rozdział :-)

1. Linia poleceń oznacza dużo pisania.
2. Pisanie to praca.
3. Pracy nikt nie lubi.

Z punktów 3. i 2. można łatwo wywnioskować, że (4) nikt nie lubi pisania. Na szczęście `bash` ratuje nas przed zaistnieniem stwierdzenia (5) (nikt nie lubi linii poleceń).

W jaki sposób `bash` wywiązuje się z tego cudownego zadania, spytacie. Otóż, oprócz omówionego już uzupełniania za pomocą znaków uniwersalnych, `bash` oferuje uzupełnianie za pomocą klawisza tabulacji.

Owo uzupełnianie działa mniej więcej tak: wpisujesz nazwę pliku; może trzeba będzie go znaleźć w twojej zmiennej `PATH`, a może podajesz nazwę jednoznacznie. Jedyne co musisz zrobić to podanie takiej ilości znaków

z nazwy pliku, by możliwa była jego jednoznaczna identyfikacja. Wtedy, po naciśnięciu klawisza tabulacji, `bash` odnajdzie żądany plik i wpisze za ciebie resztę jego nazwy!

Czas na przykład. `/usr/src` zawiera dwa podkatalogi: `/usr/src/linux` oraz `/usr/src/sendmail`. Chcesz obejrzeć zawartość `/usr/src/linux`. Tak więc wpisujesz po prostu `ls /usr/src/l`, naciskasz klawisz **`TAB`**, a wtedy `bash` uzupełnia resztę nazwy: `ls /usr/src/linux`.

Przypuśćmy teraz, że są tam takie dwa katalogi: `/usr/src/linux` oraz `/usr/src/linux-old`; jeżeli wpiszę `/usr/src/l`, po czym nacisnę **`TAB`**, `bash` uzupełni tyle, ile będzie w stanie uzupełnić, w wyniku czego otrzymam `/usr/src/linux`. Mogę na tym poprzestać, mogę też ponownie nacisnąć **`TAB`** - w drugim przypadku `bash` pokaże listę katalogów pasujących do podanego przeze mnie do tej pory wzorca.

Tak więc - o wiele mniej pisania (i za to lubimy linię poleceń). A nie mówiłem, że to będzie fajny rozdział?

8.4 Wirtualne terminale

Zdarzyło ci się może kiedyś, że będąc w środku robienia jednej rzeczy, nagle masz ochotę/musisz zrobić coś innego? Możesz oczywiście zakończyć aktualnie uruchomiony program i uruchomić kolejny, ale przecież używasz systemu wieloużytkownikowego, czyż nie? I możesz zalogować się jednocześnie tyle razy, ile tylko zapragniesz, prawda? Tak więc, dlaczego ograniczać się do robienia tylko jednej rzeczy, skoro można wykonać ich kilka równolegle?

Takie ograniczenie da się zwalczyć. Co prawda nie podłącza się kilku klawiatur do jednego komputera, podobnie jest z myszami i, poniekąd, monitorami; zresztą wielu z nas nie chciałoby tego nawet. Tak więc sprzęt nie przynosi żądanego rozwiązania. Pozostaje oprogramowanie. Linux staje na wysokości zadania, oferując równoległy dostęp do wielu konsol (tzw. “wirtualnych terminali”, z ang. “virtual terminals”, w skrócie “VTs”).

Poprzez naciśnięcie kombinacji **`Alt`** oraz klawisza funkcyjnego możesz przełączać się między konsolami; każdy klawisz funkcyjny odpowiada jednej z nich. Slackware domyślnie oferuje 6 konsol. **Alt**+**F2** przeniesie cię do drugiej, **Alt**+**F3** do trzeciej, itd.

Pozostałe klawisze funkcyjne są zarezerwowane dla sesji X-ów. Wszystkie uruchomione okienka używają swojego własnego VT, począwszy od siódmego (**Alt**+**F7**) w górę. Gdy jesteś pod X-ami, kombinacja **Alt**+**Klawisz funkcyjny** zostaje zastąpiona przez **Ctrl**+**Alt**+**Klawisz funkcyjny**; tak więc, by powrócić spod X-ów do konsoli (bez zamykania sesji X-ów), wystarczy wcisnąć **Ctrl**+**Alt**+**F3** (przeniesie cię to do trzeciej konsoli). Powrót pod X-y nastąpi po wciśnięciu **Alt**+**F7** (przy założeniu, iż używana była pierwsza sesja X-ów).

8.4.1 Screen

A co z sytuacjami, gdy nie mamy dostępu do większej ilości konsol? Co wtedy? Na szczęście Slackware zawiera wspaniały menedżer okien, całkiem trafnie nazwany `screen` (z ang. ekran). `screen` jest emulatorem terminala, posiadającym jego pewne cechy. Wykonanie polecenia `screen` wypisze krótkie wprowadzenie, po czym przełączy nas do konsoli. Będąc emulatorem wirtualnego terminala, `screen` posiada swoje własne komendy. Każda z nich zaczyna się od klawiszy **Crtl**+**A**. Dla przykładu: **Ctrl**+**A**+**C** stworzy otworzy nową konsolę (C - skrót od ang. create “utwórz”). **Ctrl**+**A**+**N** przełączy użytkownika do następnego otwartego terminala (N - z ang. next “następny”). **Ctrl**+**A**+**P** powoduje powrót do poprzedniego terminala (P - ang. previous “poprzedni”).

`screen` oferuje także odłączanie i ponowne dołączanie swoich sesji, co jest szczególnie przydatne podczas sesji zdalnych przez `ssh` oraz `telnet`, (więcej na temat tej dwójki - już niedługo). **Ctrl**+**A**+**D** odłączy aktualnie aktywny ekran. Wykonanie `screen -r` wypisze wszystkie aktualnie uruchomione ekrany, do których można się ponownie podłączyć.

```
% screen -r
```

```
There are several suitable screens on:
     1212.pts-1.redtail      (Detached)
     1195.pts-1.redtail      (Detached)
     1225.pts-1.redtail      (Detached)
     17146.pts-1.sanctuary   (Dead ???)
Remove dead screens with 'screen -wipe'.
Type "screen [-d] -r [pid.]tty.host" to resume one of them.
```

Uruchomienie `screen -r 1212` podłączy ponownie pierwszy z wypisanych ekranów. Wspomniałem już, jak użyteczne jest to w przypadku sesji zdalnych. Jeżeli byłem zalogowany do zdalnego serwera Slackware poprzez `ssh` i moje połączenie zostało przypadkowo zerwane z jakiegoś powodu (choćby przez chwilowy brak prądu), to wszystko co robiłem momentalnie znika. Może to być bardzo nieprzyjemne. Jednakże używanie `screen`'a zabezpiecza przed odłączeniem sesji, jeżeli połączenie zostanie przerwane. Gdy tylko połączenie z serwerem zostanie przywrócone, mogę ponownie podłączyć się do uruchomionego wcześniej ekranu i podjąć pracę w miejscu, w którym została ona przerwana.

Rozdział 9 Struktura Systemu Plików

We wcześnejszych rozdziałach omówiona została struktura katalogów w Slackware Linux oraz metody poszukiwania plików. W tym przekonasz się, że system plików to nie tylko układ katalogów.

Linux jest systemem wieloużytkownikowym w każdym aspekcie - również systemu plików. System przechowuje informacje o tym kto jest właścicielem jakiego pliku a kto może go tylko odczytywać(lub nie). Dowiązania do plików oraz montowania NFS to inne niepowtarzalne własności systemu plików, które wraz z tymi świadczącymi o wielużytkownikowości omówione zostaną w tym rozdziale.

9.1 Właściciel pliku

System plików przechowuje informacje o tym jaki użytkownik i jaka grupa jest właścicielem każdego pliku i katalogu w systemie. Najprostszym sposobem na uzyskanie tej informacji jest wykonanie polecenia `ls` :

```
% ls -l /usr/bin/wc
-rwxr-xr-x   1 root     bin    7368 Jul 30  1999 /usr/bin/wc
```

W tym momencie interesują nas trzecia i czwarta kolumna. Zawierają one informacje o właścicielu i grupie posiadającej dany plik. Jak widać powyżej są to: użytkownik “`root`” i grupa “`bin`”

By zmienić właściciela pliku używane jest polecenie `chown`(1) (skrót od “change owner - zmień właściciela”). Grupa zmieniana jest komendą `chgrp`(1) (skrót od “change group - zmień grupę”). Przykładowo aby zmienić właściciela pliku na `daemon`, należy wykonać `chown`:

```
# chown daemon /usr/bin/wc
```

Zmiana grupy na “`root`”, wymaga polecenia `chgrp`:

```
# chgrp root /usr/bin/wc
```

Można również użyc `chown` do wykonania powyższych czynności jednocześnie:

```
# chown daemon:root /usr/bin/wc
```

W powyższym przykładzie możliwe jest użycie kropki zamiast dwukropka - rezultat będzie taki sam (zaleca się

jednak używanie dwukropka jako lepszej formy). Użycie kropki jest przestarzałe i może zostać usunięte z przyszłych wersji `chown` by pozwolić na używanie tego znaku w nazwach użytkowników. Jest to bardzo popularna praktyka na serwerach Windows Exchange - stało się powszechne używanie adresów e-mail'owych takich jak np. `mr.jones@example.com`. W Slackware administratorom nie zaleca się nadawania takich nazw ponieważ niektóre skrypty wciąż używają kropki by wskazać użytkownika i grupę pliku. We wpomnianym przykładzie `chmod` zinterpretuje `mr.jones` jako użytkownika “mr” i grupę “jones”.

Praca z plikami a w szczególności zmiany właściciela czy grupy to bardzo ważne aspekty w Linuksie. Nawet gdy w systemie jest jeden użytkownik często wymagana jest korekcja praw dla plików czy węzłów urządzeń (device nodes).

9.2 Prawa dostępu

Prawa dostępu to kolejna ważna kwestia w aspekcie wieloużytkownikowego systemu plików. Dzięki nim można decydować kto może czytać, wykonywać czy pisać do pliku.

Informacja o prawach przechowywana jest jako czterocyfrowa liczba w systemie ósemkowym. Wyróżnione są prawa: właściciela, grupy i reszty użytkowników (każde z nich to jedna cyfra). Czwarta cyfra używana jest do przechowywania informacji specjalnych takich jak set user ID, set group ID, oraz sticky bit. Wartości oktalne przypiane prawom (sa również odpowiedniki literowe wyświetlane przez programy takie jak `ls` i akceptowane przez `chmod`) zostały opisane poniżej:

Tabela 9-1. Wartości oktalne praw dostępu

Rodzaj prawa	Wartość oktalna	Odpowienik literowy
“sticky” bit	1	t
set user ID	4	s
set group ID	2	s
read(odczyt)	4	r
write(zapis)	2	w
execute(wykonywanie)	1	x

Prawa mogą być przypisywane grupom. Na przykład aby nadać prawa “read”(odczytu) i “write”(zapisu) należy użyć cyfry “6” w części praw dotyczącej grupy.

Domyślne prawa `bash`'a to:

```
% ls -l /bin/bash
-rwxr-xr-x   1 root     bin  477692 Mar 21 19:57 /bin/bash
```

Pierwszy myślnik będzie zastąpiony literą “d” gdy do czynienia mamy z katalogiem. Trzy części praw (patrząc od lewej: właściciela, grupy, reszty) wyświetlone są dalej. Jak widać właściciel pliku ma prawa odczytu, zapisu i wykonywania (`rwx`). Grupa ma tylko prawo odczytu i wykonania (`r-x`). Reszta użytkowników również ma prawo do odczytu i wykonania.

Jak ustawić prawa dla pliku tak by były one takie jak w przypadku `bash`'a? Najpierw utwórzmy przykładowy plik:

```
% touch /tmp/example
```

```
% ls -l /tmp/example
-rw-rw-r---  1 david    users    0 Apr 19 11:21 /tmp/example
```

Następnie użyta zostanie komenda chmod(1) (co znaczy “change mode”) by ustawić do niego prawa. Teraz należy dodać wartości oktalne odpowiadające wymaganym prawom. Dla właściciela będą to: odczyt, zapis i wykonanie co w sumie daje wartość 7. Odczyt i wykonanie zsumuje się do 5. Teraz należy złożyć cyfry razem i przekazać do chmod:

```
% chmod 755 /tmp/example
% ls -l /tmp/example
-rwxr-xr-x   1 david    users    0 Apr 19 11:21 /tmp/example
```

Nasuwa się pytanie “Dlaczego poprostu nie utworzyć pliku z wymaganymi prawami?” Odpowiedź jest prosta. bash posiada pewne wbudowane wartości (built-ins) domyślne nazwane umask. Jest tak w przypadku większości powłok Unix'owych. Wartości te kontrolują jakie prawa nadawane będą nowo tworzonym plikom. Inne wartości domyślne bash'a omówione zostały w Sekcja 8.3.1. Używanie umask wymaga pewnej praktyki. Działa podobnie do chmod tyle, że na odwrót. Określa się wartości oktalne, które nie mają być nadawane nowym plikom. Domyślna wartość umask to 0022.

```
% umask
0022
% umask 0077
% touch tempfile
% ls -l tempfile
-rw-------  1 david    users    0 Apr 19 11:21 tempfile
```

Przeczytaj strony manuala dla bash'a by uzyskać więcej informacji.

Aby nadać specjalne prawa poleceniem chmod, należy dodać ich wartości oktalne i umieścić je w pierwszej kolumnie. na przykład by nadać set user ID oraz set group ID, użyta zostanie cyfra 6 w pierwszej kolumnie:

```
% chmod 6755 /tmp/example
% ls -l /tmp/example
-rwsr-sr-x   1 david    users    0 Apr 19 11:21 /tmp/example
```

Obok wartości ósemkowych podczas używania polecenia chmod można posłużyć się ich odpowiednikami literowymi.

Owner - właściciel	u
Group - grupa	g
World - reszta	o
All of the above - wszystkie z powyższych	a

Aby jednak dokonać tego co zrobiliśmy wcześniej należy wpisać kilka linijek poleceń:

```
% chmod a+rx /tmp/example
% chmod u+w /tmp/example
% chmod ug+s /tmp/example
```

Niektórzy wolą literki zamiast wartości oktalnych. W każdym razie obie metody są równoważne.

Format z użyciem cyfr ósemkowych jest często szybszy i powszechnie używany w skryptach powłoki. Litery są jednak w niektórych przypadkach bardzo przydatne. Na przykład nie ma łatwego sposobu (w przypadku cyfr) na zmiane określonych praw właścicielowi z zachowoaniem pozostałych w niezmienionej postaci.

```
% ls -l /tmp/
-rwxr-xr-x   1 alan    users    0 Apr 19 11:21 /tmp/example0
-rwxr-x---   1 alan    users    0 Apr 19 11:21 /tmp/example1
----r-xr-x   1 alan    users    0 Apr 19 11:21 /tmp/example2
% chmod g-rwx /tmp/example?
-rwx---r-x   1 alan    users    0 Apr 19 11:21 /tmp/example0
-rwx------   1 alan    users    0 Apr 19 11:21 /tmp/example1
-------r-x   1 alan    users    0 Apr 19 11:21 /tmp/example2
```

Wspomnieliśmy wcześniej o specjalnych prawach set user ID i set group ID. Wyjaśnimy co to takiego. W normalnym przypadku gdy uruchamiamy program, wykonuje się on z takimi prawami jakie ma dany użytkownik. To samo odnosi się do grupy. Przy pomocy prawa set user ID mozna zmusić program by zawsze wykonywał się z prawami właściciela (np. “root”'a). Analogicznie jest w przypadku set group ID.

Należy zachować ostrożność z tymi opcjami, gdyż mogą one potencjalnie powodować dziury w zabezpieczeniach. Jeżeli dla programów, których właścicielem jest `root` zostają nadane powyższe prawa, każdy może wykonywać je jako `root`. Ponieważ `root` nie ma ograniczeń w systemie od razu widać jakie skutki może to spowodować. W skócie używanie set user ID oraz set group ID nie jest złe ale trzeba robić to z rozwagą.

9.3 Dowiązania

Dowiązania to wskaźniki pomiędzy plikami. Dzięki dowiązaniom możliwe jest posiadanie pliku w wielu miejscach i o wielu nazwach. Wyróżniane są dwa typy dowiązń (linków): twarde i symboliczne (miękkie).

Twarde dowiązania to różne nazwy konkretnego pliku. Istnieją tylko w obrębie jednego systemu plików (???partycji???). Plik usunięty zostanie w momencie usunięcia wszystkich takich linków. W niektórych przypadkach stosowanie takich dowiązań jest uzasadnione jednak większość użytkowników preferuje linki symboliczne.

Dowiązanie symboliczne (dowiązanie miękkie) może wskazywać na plik poza systemem plików, na którym się znajduje. Właściwie jest to mały plik zawierający potrzebne informacje. Dzięki temu można tworzyć dowiązania nawet do katalogów. Częstą praktyką jest posiadanie `/var/tmp` jako linka do `/tmp`. Dowiązania tego typu można tworzyć i usuwać bez naruszania rzeczywistego pliku.

Dowiązania nie mają swojego zbioru praw dostępu czy właściciela - przejmują te, które ma plik, na który wskazują. W Slackware częściej używa się dowiązań miękkich. Przykład:

```
% ls -l /bin/sh
lrwxrwxrwx   1 root     root     4 Apr  6 12:34 /bin/sh -> bash
```

Powłoka `sh` w Slackware to tak na prawdę `bash`. Usuwanie dowiązań odbywa się przy pomocy polecenia `rm`. Komenda `ln` używana jest do ich tworzenia. Obie one są omówione szczegółowo w Rozdział 10.

Bardzo ważne jest ostrożne używanie dowiązań. Pewnego razu pracowałem na maszynie, która wykonywała kopie bezpieczeństwa na taśmach. Problem polegał na tym, że dwa dowiązania w systemie wskazywały na katalogi nadrzędne. W rezultacie na taśmie zapis trwał do momentu wyczerpania miejsca. Generalnie, wykonuje się pewne testy, które chronią przed takimi sytuacjami, jednak nasz przypadek był szczególny.

9.4 Montowanie urządzeń

W Sekcja 4.1.1, wspominaliśmy, że wszystkie urządzenia i napędy w systemie tworzą jeden wielki system plików. Różne partycje dyskowe, CD-ROM'y i stacje dyskietek umieszczone są w tym samym drzewie katalogów. Aby jednak dodać je do systemu plików by mieć do nich dostęp należy użyć polecenia `mount`(1) (i `umount`(1) by je odłączyć).

Niektóre urządzenia są automatycznie montowane podczas bootowania systemu. Są one wymienione w pliku `/etc/fstab`. Jeżeli więc jakieś urządzenie ma być montowane przy starcie należy tam umieścić wpis. W innym przypadku trzeba będzie używać powyższych poleceń.

9.4.1 `fstab`

Przyjrzyjmy się przykładowemu plikowi `/etc/fstab` :

```
% cat /etc/fstab
/dev/sda1       /               ext2        defaults        1   1
/dev/sda2       /usr/local      ext2        defaults        1   1
/dev/sda4       /home           ext2        defaults        1   1
/dev/sdb1       swap            swap        defaults        0   0
/dev/sdb3       /export         ext2        defaults        1   1
none            /dev/pts        devpts      gid=5,mode=620  0   0
none            /proc           proc        defaults        0   0
/dev/fd0        /mnt            ext2        defaults        0   0
/dev/cdrom      /mnt/cdrom      iso9660     ro              0   0
```

Pierwsza kolumna to nazwa urządzenia. W powyższym przykładzie widać pięć partycji na dwóch dyskach SCSI, dwa specjalne systemy plików nie wymagające urządzenia, stację dyskietek i napęd CD-ROM. Druga kolumna punkty montażu. Musi to być nazwa katalogu (wyjątkiem jest partycja wymiany). Trzecia kolumna to typ systemu plików na urządzeniu. Dla partycji Linux'owych będzie to `ext2` (second extended filesystem). W przypadku CD-ROM'ów `iso9660`, a partycji Windowsowych `msdos` lub `vfat`.

Czwarta kolumna to lista opcji które odnoszą się do zamontowanego systemu plików. Wartość defaults jest w większości przypadków wystarczająca. Jakkolwiek urządzenia tylko do odczytu powinny być oznaczone flagą `ro`(od read-only). Jest wiele innych opcji, których można użyć. Strony manuala dla `fstab`(5) dostarczą odpowiednich informacji na ten temat. Ostatnie dwie kolumny są wykorzystywane przez `fsck` i inne programy, które manipulują urządzeniami. Oczywiście polecamy strony manuala dla tego polecenia.

Plik `fstab` w większej części zostanie utworzony przez program setup podczas instalacji.

9.4.2 `mount` i `umount`

Dołączanie urządzeń do systemu plików jest łatwe. Wszystko co trzeba zrobić to użyć polecenia `mount` z kilkoma opcjami. Zadanie można uprościć dodając wpis w pliku `/etc/fstab`. Na przykład by zamontować CD-ROM, przy założeniu, że plik `fstab`wygląda jak w poprzednim podrozdziale należy wykonać:

```
% mount /cdrom
```

Ponieważ w `fstab` istnieje wpis dla tego punktu montażu, `mount` wie jakich opcji należy użyć. Gdyby go nie było cała procedura wyglądałaby trochę inaczej:

```
% mount -t iso9660 -o ro /dev/cdrom /cdrom
```

Powyższa linia zawiera te same informacje, które zawarte zostały w przykładowym pliku `fstab`, dla przypomnienia zostaną one pokrótce wyjaśnione. Opcja `-t iso9660` to rodzaj systemu plików na montowanym urządzeniu. `-o ro` informuje o trybie montowania (tylko do odczytu). Trzecia opcja - `/dev/cdrom` to nazwa urządzenia a `/cdrom` to punkt montażu (katalog).

W Linuxie, przed wyjęciem dyskietki, płytki czy innego urządzenia(oczywiście zamontowanego) należy je najpierw odmontować. W tym celu należy posłużyć się komendą `umount`. Nie pytaj gdzie się podziało “n” - my tego nie wiemy :-). Jako argument należy użyć nazwy urządzenia bądź punktu montażu. Na przykład by odmontować CD-ROM z poprzedniego przykładu, poniższe dwa polecenia będą poprawne:

```
# umount /dev/cdrom
# umount /cdrom
```

9.5 Montowanie NFS

NFS - Network Filesystem (Sieciowy system plików). Nie jest on naprawdę częścią rzeczywistego systemu plików, ale może być użyty w celu dodania mu pewnej funkcjonalności.

Duże środowiska Unix'owe często dzielą miedzy sobą te same programy, zbiory katalogów domowych i skrzynek mailowych. Trudne byłoby utrzymanie identycznej kopii na każdej z maszyn. Problem ten rozwiązuje NFS, dzięki któremu można dzielić zasoby, jednocześnie wygląda to tak jakby były one dostępne na każdej stacji roboczej z osobna.

Zobacz Sekcja 5.6.2 oraz storny manuala dla `exports`(5), `nfsd`(8) oraz `mountd`(8) by uzyskać więcej informacji na powyższy temat.

Rozdział 10 Pliki i katalogi

Linuks stara się być tak mocno uniksowy, jak tylko się da. Tradycyjnie, Unixy są zorientowane na obsługę przez linię poleceń. W Slackware istnieje graficzny interejs użytkownika, jednak linia poleceń nadal pełni najważniejszą rolę w zarządzaniu systemem - ważne jest więc zapoznanie się z jej podstawowymi komendami do zarządzania plikami.

Ten rozdział opisuje najbardziej znane polecenia do zarządzania plikami i pokazuje przykładowe tych poleceń użycie. Oczywiście takich komend istnieje cała masa, ale garstka przedstawiona tutaj w zupełności wystarczy na początku znajomości z Linuksem. Ponadto, przedstawione tu komendy są opisane raczej krótko - więcej detali podanych jest na odpowiednich stronach manuala.

10.1 Nawigacja : `ls`, `cd`, oraz `pwd`

10.1.1 `ls`

Ta komenda wypisuje zawartość katalogu. Użytkownicy DOSa oraz Windowsa dostrzegą jej podobieństwo do polecenia `dir`. Wywołane bez podania żadnych parametrów, `ls`(1) wylistuje pliki z katalogu bieżącego. By zobaczyć co znajduje się w katalogu głównym wykonaj następujące polecenia:

```
% cd /
% ls
bin   cdr    dev  home  lost+found  proc  sbin   tmp  var
boot  cdrom  etc  lib   mnt         root  suncd  usr  vmlinuz
```

Wielu użytkowników taka lista wprawia w zakłopotanie ponieważ nie można jednoznacznie stwierdzić na jej podstawie co jest plikiem, a co katalogiem. I tak większość użytkowników woli wywoływać `ls` dodając identyfikator typu do każdego wywołania, np:

```
% ls -FC
bin/   cdr/    dev/  home/  lost+found/  proc/  sbin/   tmp/  var/
boot/  cdrom/  etc/  lib/   mnt/         root/  suncd/  usr/  vmlinuz
```

Katalogi otrzymają znak ukośnika (slash) na końcu swojej nazwy, pliki wykonywalne gwiazdkę, itd.

Polecenie `ls` może też być wykorzystane do uzyskania innych informacji o plikach. Dla przykładu: chcąc

obejrzeć czasy utworzenia [od tłumacza: wg manuala jest to czas ostatniej modyfikacji - domyślnie], właścicieli oraz prawa dostępu należy wywołać:

```
% ls -l
drwxr-xr-x   2 root     bin          4096 May  7 09:11 bin/
drwxr-xr-x   2 root     root         4096 Feb 24 03:55 boot/
drwxr-xr-x   2 root     root         4096 Feb 18 01:10 cdr/
drwxr-xr-x  14 root     root         6144 Oct 23 18:37 cdrom/
drwxr-xr-x   4 root     root        28672 Mar  5 18:01 dev/
drwxr-xr-x  10 root     root         4096 Mar  8 03:32 etc/
drwxr-xr-x   8 root     root         4096 Mar  8 03:31 home/
drwxr-xr-x   3 root     root         4096 Jan 23 21:29 lib/
drwxr-xr-x   2 root     root        16384 Nov  1 08:53 lost+found/
drwxr-xr-x   2 root     root         4096 Oct  6 12:47 mnt/
dr-xr-xr-x  62 root     root            0 Mar  4 15:32 proc/
drwxr-x--x  12 root     root         4096 Feb 26 02:06 root/
drwxr-xr-x   2 root     bin          4096 Feb 17 02:02 sbin/
drwxr-xr-x   5 root     root         2048 Oct 25 10:51 suncd/
drwxrwxrwt   4 root     root       487424 Mar  7 20:42 tmp/
drwxr-xr-x  21 root     root         4096 Aug 24 03:04 usr/
drwxr-xr-x  18 root     root         4096 Mar  8 03:32 var/
```

Przypuśćmy, że chcesz też wylistować pliki ukryte znajdujące się w aktualnym katalogu. Dokonać można tego przez wywołanie:

```
% ls -a
.              bin   cdrom  home        mnt   sbin   usr
.             boot  dev    lib         proc  suncd  var
.pwrchute_tmp  cdr   etc    lost+found  root  tmp    vmlinuz
```

Pliki o nazwie zaczynającej się od kropki (ang. 'dot files') są plikami ukrytymi i nie zostają wyświetlone przy wywołaniu `ls`. By je ujrzeć, należy wywołać `ls` z opcją `-a`.

Jest jeszcze wiele różnych opcji, o których przeczytasz na stronach podręcznika systemowego. Nie zapomnij też, że wywołując `ls` można mu przekazać wiele wymieszanych ze sobą parametrów.

10.1.2 `cd`

Polecenie `cd` służy do zmiany bieżącego katalogu. Użycie: po prostu wpisujesz `cd` i ścieżkę dostępu do katalogu, do którego chcesz przejść. Przykłady:

```
darkstar:~$ cd /bin
darkstar:/bin$ cd usr
bash: cd: usr: No such file or directory
darkstar:/bin$ cd /usr
darkstar:/usr$ ls
bin
darkstar:/usr$ cd bin
darkstar:/usr/bin$
```

Zauważ, że pominięcie pierwszego slash'a powoduje próbę przejścia do katalogu wewnątrz bieżącego folderu. Ponadto wywołanie `cd` bez opcji przeniesie cię do twojego katalogu domowego.

Polecenie `cd` różni się nieznacznie od innych komend, gdyż jest to polecenie wbudowane w powłokę (ang. 'builtin shell command'). Polecenia takie zostały omówione w Sekcja 8.3.1. W tej chwili pewnie niewiele ci to mówi. Wiedz jednak, że z tego powodu np. nie ma dla tego polecenia strony manuala. W zamian można użyć pomocy powłoki, w taki sposób:

```
% help cd
```

Wyświelona zostanie lista dostępnych opcji dla polecenia `cd` oraz sposób ich użycia.

10.1.3 pwd

pwd to komenda pokazująca twoje aktualne położenie. Użycie: po prostu wpisz pwd; dla przykładu:

```
% cd /bin
% pwd
/bin
% cd /usr
% cd bin
% pwd
/usr/bin
```

10.2 Filtry: more, less, oraz most

10.2.1 more

more(1) to przykład czegoś, co nazywamy filtrem. Zdarza się, że wyjście (ang. output) poszczególnych komend jest zbyt duże, by zmieścić się na jednym ekranie. Komendy same z siebie nie wiedzą, jak podzielić wyniki swych działań na kilka oddzielnych ekranów. Zostawiają tę pracę filtrom.

more dzieli wyjście na osobne ekrany i wstrzymuje się z wyświetlaniem kolejnych, póki nie naciśniesz spacji. Naciśnięcie klawisza enter spowoduje przesunięcie ekranu o jedną linię. A oto i odpowiedni przykład:

```
% cd /usr/bin
% ls -l
```

Wynik na pewno nie zmieści się na ekranie. By zapobiec przewijaniu kolejnych ekranów wyjścia, po prostu przepuść je przez more:

```
% ls -l | more
```

W powyższym przykładzie użyliśmy operatora przekierowania (uzyskuje się go, wciskając shift oraz backslash). Operator przekierowania (ang. pipe) oznacza "weź wyjście ls oraz przepuść je przez more. Przez more można przepuścić praktycznie wszystko, nie tylko wynik działania ls. Więcej o przekierowaniach zostało już napisane w Sekcja 8.2.3.

10.2.2 less

Polecenie more jest całkiem użyteczne, ale często zdarzy się, że trzeba wrócić do ekranu, który został przewinięty. more nie oferuje cofania się. Oferuje je za to komenda less(1). Jest używana w ten sam sposób co polecenie more, tak więc powyższe przykłady pasują także i tutaj. Polecenie less (ang. mniej) oferuje nam więcej niż more (ang. więcej). Joost Kremers po angielsku ujął to tak:

> less is more, but more more than more is, so more is less less, so use more less if you want less more.

* Nie dająca się dosłownie przetłumaczyć gra słów, po polsku znacząca mniej więcej: less to dużo więcej niż more, stąd more to mniejsze less, tak więc używajcie więcej less jeżeli chcecie mniej more.

10.2.3 most

Tam, gdzie nie wystarcza funkcjonalność `more` i `less`, do akcji wkracza `most`(1). `less` potrafi więcej niż `more`, zaś `most` potrafi jeszcze więcej. I tak, gdy dwa pierwsze filtry są w stanie wyświetlić na raz zaledwie jeden plik, to `most` umie wyświetlać dowolną ich liczbę tak długo, dopóki okno pojedynczego pliku jest długie na co najmniej dwie linie. `most` posiada wiele opcji - po ich opis odsyłamy na stronę manuala.

10.3 Proste wypisywanie zawartości pliku: `cat` oraz `echo`

10.3.1 `cat`

`cat`(1) to skrót od angielskiego słowa “concatenate”, czyli 'połączyć, związać'. I faktycznie - początkowo był to program zaprojektowany do scalania plików tekstowych w jeden, może jednak być użyty do wielu innych celów.

By połączyć dwa lub więcej plików w jeden, wystarczy po prostu wpisać nazwy kolejnych plików po komendzie `cat` oraz przekierować tak powstałe wyjście do pliku. `cat` operuje na standardowych wejściu i wyjściu, tak więc najlepiej jest użyć konsolowych znaków przekierowania. Dla przykładu:

```
% cat file1 file2 file3 > bigfile
```

Polecenie to pobierze zawartości plików `file1`, `file2`, oraz `file3` i zapisze wszystko razem w pliku `bigfile`.

Można też używać `cat` do wyświetlenia zawartości plików. Np. wielu użytkowników używa `cat` przepuszczając przy dłuższych plikach wyjście poprzez `more` albo `less` jak zaprezentowano poniżej:

```
% cat file1 | more
```

Spowoduje to wyświetlenie zawartości `file1` oraz przepuszczenie jej poprzez `more`, dzięki czemu widać będzie tylko jeden ekran w danej chwili.

Inne, dobrze znane zastosowanie `cat` to kopiowanie plików. Możesz przekopiować każdy plik przy pomocy `cat` np. w taki sposób:

```
% cat /bin/bash > ~/mybash
```

Tutaj plik `/bin/bash` zostaje skopiowany do twojego katalogu domowego. Nowy plik otrzymuje nazwę `mybash`

`cat` posiada wiele użytecznych zastosowań, zaś przykłady omówione wyżej stanowią zaledwie kilka z nich. `cat` obszernie korzysta ze standardowych wejścia i wyjścia, dlatego też jest idealny do używania w skryptach powłoki, jako część bardziej złożonych komend.

10.3.2 `echo`

Polecenie `echo`(1) wyświetla zadany tekst na ekranie. Tekst do wyświetlenia określasz tuż po wpisaniu komendy `echo`. Domyślnie, `echo` wyświetla wprowadzony tekst, po czym drukuje za nim znak nowej linii. Przejście kursora do nowej linii można jednak ominąć - wystarczy dodać parametr `-n`. Parametr `-e` spowoduje, że `echo` poszuka w znaków specjalnych w ciągu i wykona je.

10.4 Tworzenie: `touch` oraz `mkdir`

10.4.1 `touch`

Polecenie touch(1) jest używane do zmiany czasu ostatniego dostępu oraz czasu ostatniej modyfikacji pliku. Jeżeli żądany plik nie istnieje, to polecenie touch utworzy nowy pusty plik o podanej nazwie. By ustawić w pliku czas ostatniego dostępu na ten, który jest w danej chwili w systemie, wystarczy napisać:

```
% ls -al file1
-rw-r--r--    1 root     root        9779 Feb  7 21:41 file1
% touch file1
% ls -al file1
-rw-r--r--    1 root     root        9779 Feb  8 09:17 file1
```

touch posiada kilka opcji, pozwalających na wybór, który czas pliku poddany zostanie modyfikacji. Wszystkie z nich szczegółowo opisują strony manuala.

10.4.2 mkdir

mkdir(1) służy do stworzenia nowego katalogu. Wystarczy tylko określić jego nazwę podczas wywoływania mkdir. By mówić niewiele a konkretnie - przedstawiamy przykładzik, w którym tworzymy nowy katalog w katalogu aktualnym:

```
% mkdir hejaz
```

Można też od razu sprecyzować ścieżkę dostępu:

```
% mkdir /usr/local/hejaz
```

Parametr -p informuje mkdir, by utworzyć też katalogi wyższych rzędów. Np. wywołanie powyższej komendy zakończy się porażką, jeżeli katalog /usr/local nie istnieje. Opcja -p sprawi, że w takim wypadku zostaną utworzone katalogi /usr/local oraz /usr/local/hejaz:

```
% mkdir -p /usr/local/hejaz
```

10.5 Kopiowanie i przenoszenie plików

10.5.1 cp

W Linuksie za kopiowanie plików odpowiada polecenie cp(1). Użytkownicy DOSa zauważą podobieństwa tej nazwy do nazwy polecenia copy. Polecenie cp posiada wiele opcji, dlatego też przed jego użyciem warto zajrzeć na strony manuala.

Najoczywistszy sposób użycia polecenia cp to oczywiście skopiowanie pliku z jednego miejsca w drugie. Przykładzik:

```
% cp hejaz /tmp
```

Spowoduje to skopiowanie pliku hejaz: z katalogu w którym aktualnie się znajdujemy do katalogu /tmp.

Wielu użytkowników chciałoby przy kopiowaniu pozostawić niezmienione czasy pliku, jak w poniższym przykładzie:

```
% cp -a hejaz /tmp
```

Takie użycie polecenia cp zapewnia, że czasy dostępu, modyfikacji itp. pozostaną niezmienione przy

kopiowaniu.

By rekursywnie skopiować zawartość jednego katalogu do drugiego, trzeba wykonać następujące polecenie:

```
% cp -R mydir /tmp
```

Zaowocuje to skopiowaniem katalogu *mydir* do katalogu `/tmp`.

Jeżeli życzysz sobie skopiować katalog bądź plik, ale chcesz pozostawić bez zmian wszystkie jego czasy oraz prawa dostępu, użyj `cp -p`.

```
% ls -l file
-rw-r--r--    1 root     vlad            4 Jan  1 15:27 file
% cp -p file /tmp
% ls -l /tmp/file
-rw-r--r--    1 root     vlad            4 Jan  1 15:27 file
```

`cp` posiada jeszcze wiele innych opcji, szczegółowo opisanych na stronach manuala.

10.5.2 `mv`

`mv`(1) przenosi pliki z jednego miejsca w drugie. Brzmi raczej łatwo, co nie?

```
% mv oldfile /tmp/newfile
```

`mv` posiada kilka użytecznych opcji, opisanych w manualu. Niemniej jednak bardzo rzadko bywają one używane w praktyce.

10.6 Usuwanie: `rm` oraz `rmdir`

10.6.1 `rm`

`rm`(1) usuwa pliki oraz drzewa katalogów. Użytkownicy DOSa skojarzą tę nazwę z poleceniami `del` oraz `deltree`. Nieodpowiednio użyte `rm` może stać się bardzo groźną bronią. Oczywiście zazwyczaj możliwe jest odzyskanie pliku utraconego przez przypadkowe skasowanie, niemniej jednak może być to bardzo skomplikowane (czasem też kosztowne); temat odzyskiwania plików wykracza poza ramy tej książki.

By usunąć pojedynczy plik, podaj jego nazwę zaraz po wpisaniu rm:

```
% rm file1
```

Jeżeli nie posiadasz uprawnień do zapisywania pliku, możesz otrzymać komunikat o braku dostępu. By mimo wszystko usunąć plik dodaj opcję `-f`:

```
% rm -f file1
```

By usunąć katalog wraz z zawartością, należy użyć opcji `-r` oraz `-f` razem. Pokażemy, w jaki sposób usunąć zawartość całego dysku twardego. Oczywiście - nie chcesz tego zrobić. Niemniej jednak obiecany przykład:

```
# rm -rf /
```

Bądź bardzo ostrożny gdy używasz `rm`; możesz sobie strzelić bardzo brzydkiego samobója. Polecenie to

posiada kilka opcji, które szczegółowo są opisane w podręczniku systemowym.

10.6.2 `rmdir`

`rmdir`(1) usuwa katalogi z dysku. Istnieje pewien wymóg: usuwany w ten sposób katalog musi być pusty. Składnia jest prosta:

```
% rmdir <directory>
```

Oto przykład, jak można usunąć katalog `hejaz` z katalogu bieżącego:

```
% rmdir hejaz
```

Gdy usuwamy katalog nieistniejący, `rmdir` nie omieszka o tym wspomnieć. Można też oczywiście podać pełną ścieżkę dostępu do katalogu do usunięcia, jak w przykładzie:

```
% rmdir /tmp/hejaz
```

W tym przykładzie próbowaliśmy usunąć katalog `hejaz`, znajdujący się w katalogu `/tmp`.

Można też usuwać katalog wraz z katalogami znajdującymi się nad nim - poprzez użycie opcji `-p`.

```
% rmdir -p /tmp/hejaz
```

Powyższe najpierw spróbuje usunąć katalog `hejaz` z wnętrza `/tmp` - jeżeli taka operacja się powiedzie, to zostanie podjęta próba usunięcia katalogu `/tmp`. I tak dalej; `rmdir` będzie kontynuował swą pracę do wyczyszczenia całego wspomnianego drzewa katalogów, bądź też do pierwszego błędu.

10.7 Tworzenie dowiązań między plikami: `ln`

`ln`(1) jest użytecznym poleceniem do tworzenia dowiązań (linków, skrótów) do plików. Dowiązania takie mogą być twarde (ang. "hard link"), bądź też symboliczne (z ang. "symbolic link" lub "soft link"). Różnice między tymi dwoma rodzajami dowiązań zostały już omówione w Sekcja 9.3. Gdy chcesz utworzyć symboliczny link do katalogu `/var/media/mp3` oraz umieścić go w swoim katalogu domowym, wykonasz takie polecenie:

```
% ln -s /var/media/mp3 ~/mp3
```

Parametr `-s` informuje `ln`, że ma utworzyć link symboliczny. Kolejną opcją jest plik źródłowy, na który będzie pokazywał link; na końcu podajemy nazwę samego skrótu. W powyższym przykładzie utworzyliśmy w katalogu domowym plik o nazwie `mp3`, wskazujący na `/var/media/mp3`. Oczywiście możesz nazwać link jak tylko zechcesz - poprzez zmianę ostatniej opcji.

Tworzenie dowiązania twardego jest jeszcze prostsze - wszystko, co musisz uczynić to pominięcie opcji `-s`. Dowiązania twarde mogą nie potrafić wskazywać na katalogi bądź też nie będą w stanie objąć całego systemu plików. By utworzyć twarde dowiązanie `/usr/bin/email` do pliku `/usr/bin/mutt`, po prostu wpisz co następuje:

```
# ln /usr/bin/mutt /usr/bin/email
```

Rozdział 11 Sterowanie procesami

Każdy uruchomiony w systemie program nazywany jest procesem. Procesem takim jest zarówno każdy z malutkich demonów, uruchomionych przy starcie systemu, jak też i potężny menedżer okien X Window. Każdy proces działa w systemie jako konkretny użytkownik. Te, uruchomione przy starcie systemu zwykle działają jako `root` albo `nobody`. Procesy uruchamiane przez ciebie będą widoczne w systemie jako ty. Procesy innych użytkowników będą widoczne jako inni użytkownicy.

Posiadasz pełną kontrolę nad procesami, które uruchomiłeś ty sam. `root` posiada dodatkowo kontrolę nad wszystkimi procesami, także tymi uruchomionymi przez innych użytkowników. Procesy mogą być kontrolowane i monitorowane przez wiele programów oraz poleceń powłoki.

11.1 Uruchamianie programów w tle

Programy uruchomione z linii poleceń wypisują swoje wyjście na ekran. Pozwala to śledzić i kontrolować ich pracę. Jednakże istnieją sytuacje, w których wolałbyś uruchomić program tak by nie zajmował on terminala. Taki sposób nazywa się uruchamianiem w tle; istnieje na to kilka sposobów.

Pierwszy sposobów to dodanie znaku ampersanda w linii poleceń, gdy uruchamiasz program. Dla przykładu załóżmy, że chcesz uruchomić działający w linii poleceń odtwarzacz mp3 `amp` by słuchać muzyki z katalogu pełnego plików mp3, ale potrzebujesz też wykonać inną czynność na tym samym terminalu. Następujące polecenie uruchomi `amp` w tle:

```
% amp *.mp3 &
```

Program uruchomi się normalnie, a ty powrócisz do linii poleceń.

Drugi sposób pozwala na odesłanie procesu w tło podczas jego działania. Oczywiście, najpierw trzeba taki program uruchomić. Kiedy już działa, naciskasz klawisze **Control**+**z** - powoduje to uśpienie procesu; domyślnie przerwane jest wtedy jego działanie: momentalnie zostaje zatrzymany, możesz jednak przywrócić go do pracy kiedy tylko zechcesz. Gdy uśpiłeś proces, powracasz do linii poleceń. Możesz przenieść proces w tło, wpisując:

```
% bg
```

Teraz uśpiony proces działa w tle.

11.2 Foregrounding

Jeżeli potrzebujesz wpłynąć na działanie procesu aktualnie działającego w tle, to możesz go przywrócić na pierwszy plan. Jeżeli w danej chwili w tle uruchomiony jest tylko jeden proces, to przywróci go wpisanie:

```
% fg
```

Przywrócony w ten sposób program, który nie skończył jeszcze swojego działania, przejmie kontrolę nad terminalem - nie zostaniesz przywrócony do linii poleceń. Czasem jednak program zakończy swe działanie będąc jeszcze w tle; w takim wypadku otrzymasz informację podobną do tej poniżej:

```
[1]+  Done              /bin/ls $LS_OPTIONS
```

Jest to informacja o tym, że proces uruchomiony w tle (w tym wypadku: `ls` - nieszczególnie interesujące) zakończył już swe działanie.

Możliwe jest też uruchomienie w tle kilku programów równolegle. W takiej sytuacji trzeba wiedzieć, który proces chcemy przywrócić na pierwszy plan. Zwykłe wpisanie `fg` przywróci proces przeniesiony w tło jako ostatni. A co w sytuacji, gdy mamy już całkiem sporą listę programów działających w tle? Na taką sytuację bash zawiera polecenie wypisujące wszystkie takie procesy. Nazywa się ono `jobs` i daje wynik tego typu:

```
% jobs
[1]   Stopped                 vim
[2]-  Stopped                 amp
[3]+  Stopped                 man ps
```

Wypisane zostają wszystkie procesy będące uruchomione w tle. Jak już pewnie zauważyłeś - wszystkie są zatrzymane (stopped). Oznacza to, że wszystkie zostały uśpione. Numer w pierwszej kolumnie oznacza identyfikator (w skrócie: ID) procesu będącego w tle. ID ze znakiem plus (`man ps`) oznacza, że proces będzie przywrócony, jeżeli wpiszesz tylko `fg`.

Jeżeli chcesz przywrócić `vim`, wpisz:

```
% fg 1
```

i `vim` zostanie przywrócony na konsolę. Przenoszenie programów w tło jest szczególnie użyteczne, jeżeli posiadasz tylko jeden terminal w połączeniu zdalnym. Możesz wtedy uruchomić w tle równocześnie wiele programów, wedle uznania przełączając kolejne z nich na pierwszy plan.

11.3 ps

Tak więc wiesz już, jak przełączać uruchomione programy w tło i z powrotem przywracać je na pierwszy plan. Wiesz także, że cały czas w systemie uruchomionych jest wiele różnych procesów. Ale jak zdobyć ich listę? Służy do tego polecenie `ps`(1). Polecenie to posiada całą masę opcji; my skupimy się tylko na tych najważniejszych. By uzyskać pełną ich listę, przejrzyj strony manuala. O samym manualu powiedziano więcej w rozdziale Sekcja 2.1.1.

Proste wpisanie `ps` zaowocuje wypisaniem spisu wszystkich programów uruchomionych na twoim terminalu, włączając w to programy działające na pierwszym planie (w tym także powłokę, której używasz oraz, rzecz jasna, samo `ps`). Wylistowane zostaną także procesy, które uruchomiłeś w tle. Często może być to dość krótka lista:

Rysunek 11-1. Przykładowe wyjście dla `ps`

```
% ps
   PID TTY          TIME CMD
  7923 ttyp0    00:00:00 bash
  8059 ttyp0    00:00:00 ps
```

Pomijając niewielką liczbę procesów, informacja powyższa jest całkiem typowa. Zawsze uzyskasz te same kolumny, gdy użyjesz ps, niezależnie od tego jak wiele procesów jest uruchomionych. Więc teraz: co to wszystko znaczy?

Zaczynając od lewej: `PID` oznacza identyfikator procesu (ang. *process ID*). Każdy działający proces posiada swój własny, unikalny identyfikator którym jest liczba z przedziału pomiędzy 1 a 32767. Każdemu nowemu procesowi jest przydzielany pierwszy wolny PID. Kiedy proces kończy swoje działanie normalnie (albo też kiedy zostaje "zabity" - o tym przeczytasz w następnym podrozdziale), jego PID zostaje zwolniony. Kiedy maksymalny PID zostaje osiągnięty, następnemu procesowi przydzielony zostanie najniższy aktualnie wolny PID.

Kolumna TTY określa, w którym terminalu uruchomiony został proces. Wykonanie polecenia ps wypisze jedynie procesy uruchomione na aktualnym terminalu, tak więc przy wszystkich procesach pojawi się ten sam numer TTY. I tak np. powyżej widać przy obu procesach, iż są uruchomione na ttyp0. Wskazuje to, iż są one uruchomione zdalnie albo też w trybie graficznym X Window.

Kolejna kolumna: TIME określa, jak wiele czasu pracy procesora zajmuje dany proces. Jej wartość jest różna od czasu, który upłynął od uruchomienia procesu. Trzeba pamiętać, że Linuks to system wielozadaniowy. W danej chwili uruchomionych jest wiele procesów; każdy z nich otrzymuje część czasu pracy procesora. Tak więc, kolumna TIME pokazuje, ile mniej więcej czasu dany proces już pracuje. Jeżeli w tej kolumnie pokazują się wartości większe niż kilka minut - można zacząć podejrzewać, że coś jest nie tak.

I ostatnia kolumna, CMD, pokazująca o którym to programie jest właściwie mowa. Pokazuje jedynie nazwy samych programów, nie mówiąc nic na temat opcji z którymi programy były wywołane. By uzyskać tego typu informacje, trzeba uruchomić ps z jedną z jego wielu opcji. Omówimy to krótko.

Możesz otrzymać kompletną listę programów działających w systemie używając kombinacji kilku opcji. Zaowocuje to zapewne dość długą listą pozycji (równe 70 na komputerze, na którym piszę teraz to tłumaczenie), tak więc przytnę trochę wyjście:

```
% ps -ax
  PID TTY      STAT   TIME COMMAND
    1 ?        S      0:03 init [3]
    2 ?        SW     0:13 [kflushd]
    3 ?        SW     0:14 [kupdate]
    4 ?        SW     0:00 [kpiod]
    5 ?        SW     0:17 [kswapd]
   11 ?        S      0:00 /sbin/kerneld
   30 ?        SW     0:01 [cardmgr]
   50 ?        S      0:00 /sbin/rpc.portmap
   54 ?        S      0:00 /usr/sbin/syslogd
   57 ?        S      0:00 /usr/sbin/klogd -c 3
   59 ?        S      0:00 /usr/sbin/inetd
   61 ?        S      0:04 /usr/local/sbin/sshd
   63 ?        S      0:00 /usr/sbin/rpc.mountd
   65 ?        S      0:00 /usr/sbin/rpc.nfsd
   67 ?        S      0:00 /usr/sbin/crond -l10
   69 ?        S      0:00 /usr/sbin/atd -b 15 -l 1
   77 ?        S      0:00 /usr/sbin/apmd
   79 ?        S      0:01 gpm -m /dev/mouse -t ps2
   94 ?        S      0:00 /usr/sbin/automount /auto file /etc/auto.misc
  106 tty1     S      0:08 -bash
  108 tty3     SW     0:00 [agetty]
  109 tty4     SW     0:00 [agetty]
  110 tty5     SW     0:00 [agetty]
  111 tty6     SW     0:00 [agetty]
  [output cut]
```

Większość tych procesów zostaje uruchomiona jeszcze przy starcie większości systemów. Dokonałem pewnych zmian w swoim systemie, tak więc u ciebie spis ten będzie zapewne trochę inny. Niemniej jednak, większość z tych procesów zobaczysz także u siebie. Jak widzisz, opcje podane przy wywołaniu ps powodują wyświetlenie opcji wywołania wszystkich procesów. Ostatnimi czasy, wrażliwość jądra systemu na ptrace sprawiła, że stworzono poprawkę dzięki której nie są podawane opcje wywołania dla wielu różnych działających w systemie procesów. Przykładem są te wylistowane wyżej w nawiasach pozycje o PID'ach od 108 do 110. Powyższy wypis pokazuje też jedną kolumnę więcej niż poprzedni oraz przynosi ciut więcej ciekawych informacji.

Jako pierwsze zauważysz na pewno, iż większa część procesów została uruchomiona na konsoli numer “?”. Takie procesy nie są związane z żadnym konkretnym terminalem. Najczęściej dzieje się tak w wypadku demonów, które to właśnie są procesami nie wiążącymi się z żadnym konkretnym terminalem. Bardziej znane demony to np. sendmail, BIND, apache oraz NFS. Typowym ich zadaniem jest oczekiwanie na żądanie wysłane przez klienta oraz udzielenie odpowiedzi na nie.

Druga rzecz: pojawia się nowa kolumna STAT. Pokazuje ona status danego procesu. S (skrót od ang. "sleeping") oznacza uśpiony: proces w tym stanie oczekuje na jakieś zdarzenie. Z oznacza proces-zombie. Proces-zombie to proces, którego rodzic zakończył swe działanie ("umarł"), pozostawiając nie zakończony proces potomny. Nie

jest to dobra rzecz. `D` oznacza proces który jest uśpiony tak mocno, że nie można go "obudzić" - procesy w takim stanie często nie reagują nawet na komendę SIGKILL (więcej o SIGKILL przeczytasz już niedługo w podrozdziale `kill`). `W` oznacza stronicowanie. Proces martwy jest oznaczony przez `X`. Proces z literką `T` jest albo śledzony (ang. "traced"), albo zatrzymany. `R` oznacza, iż proces jest gotów do wykonania.

Jeżeli chcesz zobaczyć jeszcze więcej informacji na temat uruchomionych procesów, spróbuj tego:

```
% ps -aux
 USER       PID %CPU %MEM    VSZ  RSS TTY      STAT START   TIME COMMAND
 root         1  0.0  0.0    344   80 ?        S    Mar02   0:03 init [3]
 root         2  0.0  0.0      0    0 ?        SW   Mar02   0:13 [kflushd]
 root         3  0.0  0.0      0    0 ?        SW   Mar02   0:14 [kupdate]
 root         4  0.0  0.0      0    0 ?        SW   Mar02   0:00 [kpiod]
 root         5  0.0  0.0      0    0 ?        SW   Mar02   0:17 [kswapd]
 root        11  0.0  0.0   1044   44 ?        S    Mar02   0:00 /sbin/kerneld
 root        30  0.0  0.0   1160    0 ?        SW   Mar02   0:01 [cardmgr]
 bin         50  0.0  0.0   1076  120 ?        S    Mar02   0:00 /sbin/rpc.port
 root        54  0.0  0.1   1360  192 ?        S    Mar02   0:00 /usr/sbin/sysl
 root        57  0.0  0.1   1276  152 ?        S    Mar02   0:00 /usr/sbin/klog
 root        59  0.0  0.0   1332   60 ?        S    Mar02   0:00 /usr/sbin/inet
 root        61  0.0  0.2   1540  312 ?        S    Mar02   0:04 /usr/local/sbi
 root        63  0.0  0.0   1796   72 ?        S    Mar02   0:00 /usr/sbin/rpc.
 root        65  0.0  0.0   1812   68 ?        S    Mar02   0:00 /usr/sbin/rpc.
 root        67  0.0  0.2   1172  260 ?        S    Mar02   0:00 /usr/sbin/cron
 root        77  0.0  0.2   1048  316 ?        S    Mar02   0:00 /usr/sbin/apmd
 root        79  0.0  0.1   1100  152 ?        S    Mar02   0:01 gpm
 root        94  0.0  0.2   1396  280 ?        S    Mar02   0:00 /usr/sbin/auto
 chris      106  0.0  0.5   1820  680 tty1     S    Mar02   0:08 -bash
 root       108  0.0  0.0   1048    0 tty3     SW   Mar02   0:00 [agetty]
 root       109  0.0  0.0   1048    0 tty4     SW   Mar02   0:00 [agetty]
 root       110  0.0  0.0   1048    0 tty5     SW   Mar02   0:00 [agetty]
 root       111  0.0  0.0   1048    0 tty6     SW   Mar02   0:00 [agetty]
 [output cut]
```

To już całkiem sporo informacji. Generalnie, dowiadujemy się tu który użytkownik uruchomił dany proces, jak wiele zasobów systemowych proces zajmuje (kolumny %CPU, %MEM, VSZ oraz RSS), oraz kiedy proces został uruchomiony. Oczywiście, taka ilość informacji będzie bardzo pomocna dla administratora systemu. Ale pojawia się tu drobny problem: część tak obfitej informacji przekracza szerokość ekranu, przez co zostaje wyświetlona niepełna jej część. Opcja `-w` zmusi `ps` do przełamania długich linii.

Niezbyt to piękne, ale spełnia swoje zadanie. Teraz masz już potężną rozpiskę dla każdego procesu. Ale można usyskać o procesach jeszcze więcej informacji - sprawdź bardzo dogłębną stronę manuala na temat komendy `ps`. Co by jednak nie było, opcje podane powyżej są najbardziej popularne i to ich właśnie będziesz używał najczęściej.

11.4 `kill`

Okazjonalnie programy mogą zachowywać się źle i potrzebne będzie siłowe przywrócenie ich do porządku. Program do tego typu zadań administracyjnych nazywa się `kill`(1) i może być użyty do manipulowania procesami na kilka różnych sposobów. Najbardziej oczywisty sposób użycia komendy `kill` (ang. "zabij") - to zabicie procesu. Będziesz potrzebował tej możliwości wtedy, gdy program wymknął się spod kontroli i zużywa zbyt wiele zasobów systemowych, bądź też po prostu masz już dość jego działania.

By zabić proces, musisz znać jego PID albo nazwę. By uzyskać PID, użyj polecenia `ps` poznanego w poprzednim podrozdziale. Dla przykładu, by zakończyć proces 4747, wydasz następującą komendę:

```
% kill 4747
```

Zauważ, że musisz być właścicielem procesu by móc go zabić. Jest to wymóg bezpieczeństwa. Będąc w stanie zakańczać procesy innych użytkowników, mógłbyś narobić niezłego zamieszania. Oczywiście, `root` może zabić dowolny proces uruchomiony w systemie.

Jest też pewna odmiana polecenia `kill` zwana `killall`(1). Program ten robi dokładnie to, na co wskazuje jego nazwa: zabija wszystkie procesy mające określoną nazwę. Jeżeli chcesz zakończyć pracę wszystkich aktualnie uruchomionych edytorów `vim`, wydasz następującą komendę:

```
% killall vim
```

Każda uruchomiona przez ciebie kopia `vim'a` zostanie w tym momencie zakończona. Użycie powyższej komendy przez `root'a` spowoduje zabicie wszystkich `vim'ów` uruchomionych przez dowolnego użytkownika. Podsuwa to pomysł na wykopanie każdego (także i siebie) z systemu:

```
# killall bash
```

Czasem zwykły kill nie wystarczy. Niektóre procesy nie posłuchają się nakazu wydanego tym poleceniem i nie umrą na rozkaz. Do takich opornych trzeba będzie użyć bardziej konkretnych argumentów. Jeżeli oporny PID 4747 nie odpowiedział na twoje żądanie zakończenia go, możesz użyć następującego polecenia:

```
% kill -9 4747
```

To prawie na pewno zabije proces 4747. Można to samo wykonać dla `killall`. Taka forma polecenia przesyła do procesu inny sygnał. Zwykły `kill` przesyła do procesu sygnał `SIGTERM` (od ang. terminate 'zakańczać'), mówiący by proces zakończył to, co robi, posprzątał po sobie oraz zakończył się. `kill -9` wysyła sygnał `SIGKILL` (kill), po prostu zakańczający proces. Proces nie dostaje wtedy zezwolenia na posprzatanie po sobie, tak więc czasem niedobre rzeczy - jak np. błędy danych - mogą się pojawić przy wysłaniu sygnału `SIGKILL`. Do twojej dyspozycji jest cała lista sygnałów. Możesz ją uzyskać wpisując:

```
% kill -l
  1) SIGHUP     2) SIGINT    3) SIGQUIT   4) SIGILL
  5) SIGTRAP    6) SIGABRT   7) SIGBUS    8) SIGFPE
  9) SIGKILL   10) SIGUSR1  11) SIGSEGV  12) SIGUSR2
 13) SIGPIPE   14) SIGALRM  15) SIGTERM  17) SIGCHLD
 18) SIGCONT   19) SIGSTOP  20) SIGTSTP  21) SIGTTIN
 22) SIGTTOU   23) SIGURG   24) SIGXCPU  25) SIGXFSZ
 26) SIGVTALRM 27) SIGPROF  28) SIGWINCH 29) SIGIO
 30) SIGPWR
```

Przy wywołaniu `kill` musi zostać użyty numer, zaś z `killall` może zostać użyty minus wraz z nazwą sygnału “SIG”. Oto kolejny przykład:

```
% killall -KILL vim
```

Ostatni sposób użycia `kill` to zrestartowanie procesu. Wysłanie sygnału `SIGHUP` spowoduje w większości przypadków, że procesy ponownie przeczytają swoje pliki konfiguracyjne. Przydaje się to wtedy, gdy właśnie dokonaliśmy edycji tych plików.

11.5 `top`

I na sam koniec tego rozdziału: komenda, dzięki której możesz oglądać odświeżane na bieżąco informacje o procesach uruchomionych w twoim systemie. Komenda ta nazywa się `top`(1), a wywołuje się ją tak:

```
% top
```

Takie użycie spowoduje wyświetlenie pełnoekranowej informacji o procesach działających w systemie oraz kilku ogólnych informacji o nim samym. Zawarte tam będą dane o średnim obciążeniu systemu (load average), liczbie procesów, stanie procesora, wolnej pamięci oraz pewne detale odnośnie samych procesów, takie jak

PID, właściciel, priorytet, zużycie CPU oraz pamięci, czas działania i nazwa programu.

```
 6:47pm  up 1 day,  18:01,  1 user,  load average: 0.02, 0.07, 0.02
61 processes: 59 sleeping, 2 running, 0 zombie, 0 stopped
CPU states:  2.8% user,  3.1% system,  0.0% nice, 93.9% idle
Mem:   257992K av,  249672K used,    8320K free,   51628K shrd,  78248K buff
Swap:   32764K av,     136K used,   32628K free,                 82600K cached

PID USER     PRI  NI  SIZE  RSS  SHARE  STAT  LIB %CPU %MEM  TIME COMMAND
 112 root     12   0 19376  18M   2468  R       0  3.7  7.5 55:53 X
4947 david    15   0  2136 2136   1748  S       0  2.3  0.8  0:00 screenshot
3398 david     7   0 20544  20M   3000  S       0  1.5  7.9  0:14 gimp
4946 root     12   0  1040 1040    836  R       0  1.5  0.4  0:00 top
 121 david     4   0   796  796    644  S       0  1.1  0.3 25:37 wmSMPmon
 115 david     3   0  2180 2180   1452  S       0  0.3  0.8  1:35 wmaker
4948 david    16   0   776  776    648  S       0  0.3  0.3  0:00 xwd
   1 root      1   0   176  176    148  S       0  0.1  0.0  0:13 init
 189 david     1   0  6256 6156   4352  S       0  0.1  2.4  3:16 licq
4734 david     0   0  1164 1164    916  S       0  0.1  0.4  0:00 rxvt
   2 root      0   0     0    0      0  SW      0  0.0  0.0  0:08 kflushd
   3 root      0   0     0    0      0  SW      0  0.0  0.0  0:06 kupdate
   4 root      0   0     0    0      0  SW      0  0.0  0.0  0:00 kpiod
   5 root      0   0     0    0      0  SW      0  0.0  0.0  0:04 kswapd
  31 root      0   0   340  340    248  S       0  0.0  0.1  0:00 kerneld
  51 root      0   0    48   48     32  S       0  0.0  0.0  0:00 dhcpcd
  53 bin       0   0   316  316    236  S       0  0.0  0.1  0:00 rpc.portmap
  57 root      0   0   588  588    488  S       0  0.0  0.2  0:01 syslogd
```

Program ten nazywa się `top` (ang. "szczyt"), gdyż procesy zużywające najwięcej mocy procesora znajdą się na samej górze listy. Ciekawostką jest to, że top będzie umieszczony jako pierwszy na liście najmniej aktywnych (oraz pewnych aktywnych) systemów ze względu na swoją zajętość procesora. Niemniej jednak `top` jest całkiem użytecznym programem, pozwalającym stwierdzić który z procesów zachowuje się nieprawidłowo i powinien zostać zakończony.

Ale przypuśćmy, że chcesz jedynie listę swoich własnych procesów, bądź też procesów uruchomionych przez innego użytkownika. Takie procesy niekoniecznie muszą być w czołówce tych najbardziej "procesorożernych". Opcja `-u` pozwala na podanie nazwy użytkownika (bądź jego UID) oraz monitorowanie procesów będących własnością tylko tego UID.

```
% top -u boogi
  PID USER      PR  NI  VIRT RES   SHR S %CPU %MEM   TIME+   COMMAND
 3622 alan      13   0 11012  10m 6956 S  1.0  2.1   0:03.66 gnome-terminal
 3739 alan      13   0  1012 1012  804 R  0.3  0.2   0:00.06 top
 3518 alan       9   0  1312 1312 1032 S  0.0  0.3   0:00.09 bash
 3529 alan       9   0   984  984  848 S  0.0  0.2   0:00.00 startx
 3544 alan       9   0   640  640  568 S  0.0  0.1   0:00.00 xinit
 3548 alan       9   0  8324 8320 6044 S  0.0  1.6   0:00.30 gnome-session
 3551 alan       9   0  7084 7084 1968 S  0.0  1.4   0:00.50 gconfd-2
 3553 alan       9   0  2232 2232  380 S  0.0  0.4   0:00.05 esd
 3555 alan       9   0  2552 2552 1948 S  0.0  0.5   0:00.10 bonobo-activati
 3557 alan       9   0  2740 2740 2224 S  0.0  0.5   0:00.05 gnome-smproxy
 3559 alan       9   0  6496 6492 5004 S  0.0  1.3   0:00.31 gnome-settings-
 3565 alan       9   0  1740 1740 1440 S  0.0  0.3   0:00.28 xscreensaver
 3568 alan       9   0  7052 7052 4960 S  0.0  1.4   0:02.28 metacity
 3572 alan       9   0 11412  11m 7992 S  0.0  2.2   0:01.58 gnome-panel
 3574 alan       9   0 12148  11m 8780 S  0.0  2.4   0:00.64 nautilus
 3575 alan       9   0 12148  11m 8780 S  0.0  2.4   0:00.00 nautilus
 3576 alan       9   0 12148  11m 8780 S  0.0  2.4   0:00.00 nautilus
```

Jak widzisz, aktualnie działają u mnie `X`, `top`, oraz `gnome-terminal` (w którym to piszę te słowa) i jeszcze cała masa powiązanych z X-ami procesów, które to pożerają najwięcej czasu pracy procesora. Powyższa metoda jest świetna, gdy trzeba sprawdzić jak ciężko użytkownicy pracują w twoim systemie.

`top` pozwala także na monitorowanie procesów według ich PID'ów, ignorowanie bezczynnych procesów oraz procesów-zombie, a także na wiele wiele innych rzeczy. Najlepszym miejscem na uzyskanie informacji o tych opcjach jest strona manuala dla polecenia `top`.

Rozdział 12 Podstawy administracji systemem

Spokojnie, spokojnie... Znam twoje myśli: “Nie jestem administratorem systemu! Nigdy nawet nie myślałem, by nim zostać! ”

Prawda jest taka, że jesteś administratorem na każdym komputerze, na którym posiadasz hasło `root'a`. Może to być zwykły komputer biurowy, na którym urzęduje jeden bądź dwóch użytkowników, albo też może to być duży serwer z paroma setkami użytkowników. Niezależnie od tego, z pewnością przydatna okaże się wiedza jak zarządzać użytkownikami oraz jak poprawnie zamykać system. Te zadania nie wyglądają na szczególnie skomplikowane, ale znajdzie się kilka drobiazgów na które trzeba uważać.

12.1 Users and Groups

Jak wspomniano w Rozdział 8, na co dzień nie powinieneś się logować do systemu jako `root`. Zamiast tego najlepiej stworzyć sobie konto zwykłego użytkownika do użytku codziennego, zaś konta root'a używać jedynie do wykonywania zadań administracyjnych. By stworzyć w systemie nowego użytkownika, możesz albo użyć narzędzi dostarczanych w tym celu razem ze Slackware, albo też bezpośrednio edytować odpowiednie pliki.

12.1.1 Supplied Scripts

Najłatwiejszy sposób zarządzania użytkownikami i grupami to sposób wspierany przez gotowe skrypty i programy. Slackware zawiera programy: `adduser`, `userdel`(8), `chfn`(1), `chsh`(1) oraz `passwd`(1), służące do zarządzania kontami użytkowników. Polecenia `groupadd`(8), `groupdel`(8) oraz `groupmod`(8) służą do zarządzania grupami. Za wyjątkiem komend `chfn`, `chsh` oraz `passwd`, wszystkie wymienione programy są generalnie uruchamiane jedynie przez `root'a`; stąd też są zlokalizowane w katalogu `/usr/sbin`. `chfn`, `chsh` oraz `passwd` mogą być uruchomione przez każdego; znajdują się w `/usr/bin`.

Nowi użytkownicy mogą zostać dodani do systemu za pomocą polecenia `adduser`. Zaczniemy od przejścia przez całą procedurę, pokazując wszystkie zapytania kierowane do nas przez program, oraz pokrótce wyjaśniając ich znaczenie. Domyślna odpowiedź znajduje się w nawiasach i może być wybrana dla prawie każdego zapytania, dopóki oczywiście nie będziesz chciał czegoś zmienić.

```
# adduser
Login name for new user []: jellyd
```

Jest to nazwa której użytkownik będzie używał do logowania się. Tradycyjnie loginy są co najwyżej ośmioznakowe oraz pisane małymi literami. (Nic jednak nie stoi na przeszkodzie, by wpisać więcej znaków bądź użyć cyfr - ale unikaj tego dopóki nie masz dość dobrego powodu).

Możesz też podać login jako parametr w linii poleceń:

```
# adduser jellyd
```

W tym drugim przypadku, po ustaleniu loginu, adduser zapyta się o ID użytkownika:

```
User ID ('UID') [ defaults to next available ]:
```

To właśnie na podstawie ID użytkownika (w skrócie UID) ustala się prawa własności w Linuksie. Każdy użytkownik posiada swój unikalny numer, począwszy od 1000 w Slackware. Możesz sam ustalić numer dla nowego użytkownika, może to zrobić za ciebie adduser - poprzez przydzielenie pierwszego wolnego identyfitakora.

```
Initial group [users]:
```

Wszyscy użytkownicy są umieszczani domyślnie w grupie `users`. Możesz zapewne chcieć umieścić nowego użytkownika w innej grupie, ale nie jest to polecane dopóki naprawdę nie masz absolutnej pewności, że wiesz co robisz.

```
Additional groups (comma separated) []:
```

To pytanie pozwala na dodanie użytkownika dodatkowo do innych grup. Możliwa jest przynależność jednego użytkownika do kilku grup na raz w tym samym czasie. Jest to bardzo użyteczne, gdy masz już konkretnie zorganizowane grupy do robienia takich rzeczy jak np. modyfikowanie plików stron internetowych, granie w gry itp. Dla przykładu: niektóre lokalizacje [sites] definiują grupę `wheel` jako jedyną mogącą używać komendy `su`. Albo też drugi przykład: domyślnie podczas instalacji Slackware zostaje utworzona grupa `sys`, która posiada uprawnienia do odgrywania dźwięków za pomocą zamontowanej karty dźwiękowej.

```
Home directory [/home/jellyd]
```

Katalog domowy domyślnie zostanie umieszczony wewnątrz `/home`. Jeżeli zarządzasz naprawdę dużym systemem, to możliwe jest iż przeniosłeś katalogi domowe w inne miejsce (bądź inne miejsca). Ten krok pozwala ci określić, gdzie znajdzie się katalog domowy aktualnie dodawanego użytkownika.

```
Shell [ /bin/bash ]
```

`bash` jest powłoką domyślną dla Slackware i będzie odpowiedni dla większości osób. Jeżeli nowy użytkownik jest wychowankiem Uniksa, to być może będzie bardziej zaznajomiony z inną powłoką. Możesz zmienić domyślną powłokę na inną już teraz, albo też pozwolić użytkownikom zmienić ją później przy użyciu polecenia `chsh`.

```
Expiry date (YYYY-MM-DD) []:
```

Kontom można przydzielić czas wygaśnięcia z nastaniem pewnej daty. Domyślnie wygaśnięcie konta nie jest ustawione. Można jednak to zmienić, gdy zachodzi potrzeba - dostawcy internetu mogą na przykład chcieć utworzyć konto tracące ważność konkretnego dnia, gdy jego użytkownik nie opłacił abonamentu.

```
New account will be created as follows:
---------------------------------------
Login name:         jellyd
UID:                [ Next available ]
Initial group:      users
Additional groups:  [ None ]
Home directory:     /home/jellyd
Shell:              /bin/bash
Expiry date:        [ Never ]
```

No i gotowe.. jeżeli chcesz zrezygnować, naciśnij **Control**+**C**. W przeciwnym wypadku wciśnij **`ENTER`** by kontynuować tworzenie konta.

Widzisz teraz wszystkie informacje odnośnie nowego konta, które wprowadziłeś, oraz masz możliwość przerwania procesu tworzenia nowego konta użytkownika. Jeżeli stwierdzisz jakiś błąd w danych już wprowadzonych, wciśnij **Control**+**C** i rozpocznij proces tworzenia konta od początku. Jeżeli wszystko gra - wciśnięcie **`enter'a`** utworzy nowe konto.

```
Creating new account...

Changing the user information for jellyd
Enter the new value, or press return for the default
```

```
Full Name []: Jeremy
Room Number []: Smith 130
Work Phone []:
Home Phone []:
Other []:
```

Wszystkie te informacje są opcjonalne. Nie musisz wprowadzać tu niczego, jeżeli nie masz na to ochoty, zaś użytkownik może zmienić zawarte tu dane w każdej chwili, korzystając z polecenia `chfn`. Niemniej jednak pomocne może okazać się wprowadzenie chociaż imienia oraz numeru telefonu, w razie gdybyś później musiał skontaktować się z tą osobą.

```
Changing password for jellyd
Enter the new password (minimum of 5, maximum of 127 characters)
Please use a combination of upper and lower case letters and numbers.
New password:
Re-enter new password:
Password changed.

Account setup complete.
```

Teraz musisz wprowadzić hasło dla nowego usera. Generalnie, jeżeli nowy użytkownik nie jest przy tym fizycznie obecny, wpisuje się po prostu standardowe hasło i informuje się go, by zmienił je sobie na coś bardziej bezpiecznego.

Ustalenie hasła: Posiadanie dobrego hasła to pierwszy krok w skutecznej obronie przeciw włamaniom. Z pewnością nie chcesz być posiadaczem hasła łatwego do odgadnięcia, ponieważ takie hasło ułatwiłoby tylko włamanie do twojego systemu. Idealne, bezpieczne hasło będzie przypadkowym ciągiem znaków, zarówno małych jak i dużych liter, cyfr oraz pozostałych znaków alfanumerycznych. (Użycie w haśle znaku tabulacji może nie być zbyt mądrym pomysłem, ale to zależy jeszcze od systemu w którym będziesz się logował). Istnieje wiele programów generujących za ciebie losowe hasła; bardzo łatwo znaleźć je w internecie.

Nie zapomnij o zdrowym rozsądku: nie twórz hasła z daty czyichś urodzin, powszechnie znanej frazy, czegoś znalezionego na biurku bądź czegokolwiek, co jest w łatwy sposób związanego z tobą. Hasło typu “bezpieczne1” bądź też każde inne które zobaczyłeś na papierze albo w internecie także jest złe.

Usuwanie użytkowników także nie należy do trudnych procesów. Wystarczy wywołanie komendy `userdel` oraz podać jako parametr nazwę konta do usunięcia. Wcześniej należy jeszcze sprawdzić, czy dany użytkownik nie jest zalogowany w systemie, oraz czy nie ma procesów uruchomionych jako ten użytkownik. Poza tym pamiętaj także, iż po usunięciu konta wszystkie informacje na temat hasła użytkownika znikają bezpowrotnie.

```
# userdel jellyd
```

Komenda ta usuwa denerwującego użytkownika `jellyd` z systemu. Nie ma więcej drania! :) Informacje o użytkowniku zostały usunięte z plików `/etc/passwd`, `/etc/shadow` oraz `/etc/group`, niemniej jednak nie został usunięty katalog domowy użytkownika.

Jeżeli za jednym zamachem chcesz usunąć tak użytkownika, jak i jego konto - wydaj takie polecenie:

```
# userdel -r jellyd
```

Tymczasowa dezaktywacja konta zostanie opisana w następnym podrozdziale o hasłach, jako że taka czynność pociąga za sobą konieczność zmiany hasła użytkownika. Zmiany innych informacji o kontach opisane są w Sekcja 12.1.3.

Programy dodające i usuwające grupy są bardzo proste. `groupadd` po prostu dodaje do pliku `/etc/group` kolejny wpis z unikalnym ID grupy, zaś `groupdel` usuwa zadaną grupę. W twoich rękach pozostaje edycja

pliku /etc/group, by dodawać użytkowników do poszczególnych grup. A teraz, dla przykładu utwórzmy grupę cvs:

```
# groupadd cvs
```

...oraz usuńmy ją:

```
# groupdel cvs
```

12.1.2 Changing Passwords

Program passwd zmienia hasła poprzez modyfikację pliku /etc/shadow. W pliku tym przechowywane są wszystkie hasła systemu w zakodowanym formacie. By zmienić swoje hasło, wpisz:

```
% passwd
Changing password for chris
Old password:
Enter the new password (minumum of 5, maximum of 127 characters)
Please use a combination of upper and lower case letters and numbers.
New password:
```

Jak zauważyłeś, zostałeś poproszony o podanie swojego starego hasła. Podczas wpisywania nie pojawi się ono na ekranie, podobnie zresztą jak podczas logowania. Po wpisaniu starego hasła zostaniesz poproszony o podanie nowego. passwd sprawdzi hasło swoimi testami i poinformuje, jeżeli nie przejdzie ono testu jakości. Jeżeli chcesz, możesz zignorować to ostrzeżenie. Zostaniesz poproszony o powtórne wpisanie nowego hasła - tak dla pewności.

Będąc root'em możesz także zmieniać hasła innych użytkowników:

```
# passwd ted
```

Przejdziesz wtedy przez taką samą procedurę jak opisana wyżej, za wyjątkiem wpisywania starego hasła (kolejny fajny przywilej posiadania uprawnień root'a...).

Jeżeli zachodzi taka konieczność, to można tymczasowo zablokować konto, oraz uczynić je ponownie dostępnym gdy tylko zechcemy. Blokowanie i odblokowywanie konta następują przy użyciu polecenia passwd. By zablokować konto, jako root wpisz:

```
# passwd -l david
```

Spowoduje to zmianę hasła użytkownika na coś, co nigdy nie będzie mogło być dopasowane do żadnej z zakodowanych wartości (parametr "-l" to skrót od ang. locked 'zamknięty' - przyp. tłum.). Konto odblokowuje się poleceniem:

```
# passwd -u david
```

Teraz konto david'a działa już normalnie (skrót "-u" powstał od ang. unlocked 'odblokowany' - przyp. tłum.). Założenie blokady na konto może okazać się użyteczne, gdy dany użytkownik nie przestrzega zasad ustalonych przez ciebie, bądź też wysłał zbyt wiele kopii xeyes(1) na twój pulpit...

12.1.3 Modyfikacja danych użytkownika

Są dwie rzeczy, które użytkownicy mogą zmienić w każdej chwili: są to ich powłoka, oraz informacje podawane

o nich przez `finger`. Slackware Linux używa `chsh` (change shell 'zmień powłokę') oraz `chfn` (change finger) do modyfikacji tych wartości.

Użytkownik może wybrać sobie każdą powłokę spośród tych wypisanych w pliku `/etc/shells`. Dla większości osób najodpowiedniejszym shell'em będzie `/bin/bash`. Niektórzy jednak, będąc przyzwyczajeni do innych powłok używanych na swoich komputerach w pracy bądź w szkole, będą chcieli używać tego co już znają. By zmienić powłokę, należy wpisać `chsh`:

```
% chsh
Password:
Changing the login shell for chris
Enter the new value, or press return for the default
        Login Shell [/bin/bash]:
```

Po wpisaniu hasła podaj pełną ścieżkę dostępu do nowego shell'a. Upewnij się przedtem, że znajduje się on na liście w pliku `/etc/shells`(5). `root` może zmienić powłokę każdemu użytkownikowi wywołując `chsh` z nazwą użytkownika jako parametrem.

Informacje podawane przez `finger` to dodatkowe informacje o użytkowniku, takie jak imię i nazwisko, numery telefonów oraz pokoju. Mogą być one zmodyfikowane poleceniem `chfn`; przechodzi się wtedy tą samą procedurę, co przy tworzeniu konta. I jak zwykle, `root` może zmienić te dodatkowe informacje każdemu użytkownikowi.

12.2 Users and Groups, the Hard Way

Oczywiście możliwe jest dodawanie, modyfikacja oraz usuwanie użytkowników bez użycia gotowych skryptów i programów dostarczonych w dystrybucji Slackware. I nie jest to trudne, chociaż po zapoznaniu się z tym procesem stwierdzisz pewnie, że o wiele łatwiej jest posłużyć się w tym przypadku skryptami. Niezależnie jednak od tego ważne jest, by wiedzieć jak przechowywane są informacje o twoim koncie, w przypadku gdyby trzeba było je odzyskać, nie mając pod ręką narzędzi omówionych przed chwilą.

Na początek zajmijmy się dodaniem informacji o nowym użytkowniku do plików: `/etc/passwd`(5), `/etc/shadow`(5) oraz `/etc/group`(5). Plik `passwd` przechowuje pewne informacje o użytkownikach twojego systemu, ale ich hasła są przechowywane gdzie indziej. Kiedyś hasła znajdowały się właśnie w tym pliku, ale dawno już zarzucono ten sposób ze względów bezpieczeństwa. Plik passwd musi być czytelny dla każdego użytkownika, ale też nikt nie chce by nawet zakodowane hasło było dostępne każdemu, gdyż nie byłoby niemożliwością odkodowanie go. Zamiast tego, zakodowane hasła są przechowywane w pliku `shadow`, który może być czytany tylko przez root'a, zaś w pliku `passwd` hasło każdego jest zapisane jako “`x`”. Plik `group` stanowi spis wszystkich grup oraz tego, kto do nich należy.

Do bezpiecznej modyfikacji pliku `/etc/passwd` możesz użyć polecenia `vipw`, zaś komendy `vigr` użyj do równie bezpiecznej edycji pliku `/etc/group`. Użyj `vipw -s` do bezpiecznej edycji pliku `/etc/shadow`. (drobne wyjaśnienia odnośnie słowa “bezpieczny” - w tym kontekście oznacza ono, iż nikt inny nie będzie miał możliwości modyfikacji pliku w czasie, gdy edytujesz go ty. Jeżeli jesteś jedynym administratorem w swoim systemie, to najprawdopodobniej jesteś bezpieczny, ale i tak warto wyrabiać w sobie dobre nawyki od samego początku).

Przyjrzyjmy się teraz plikowi `/etc/passwd` oraz zobaczmy, jak dodać nowego użytkownika. Typowy wpis w `passwd` wygląda tak:

```
chris:x:1000:100:Chris Lumens,Room 2,,:/home/chris:/bin/bash
```

Każda linia stanowi wpis odnośnie pojedynczego użytkownika, zaś każde pole każdej linii oddzielone jest od pozostałych dwukropkiem. Polami są login, zakodowane hasło (czyli “`x`” dla każdego użytkownika w Slackware, odkąd Slackware przechowuje właściwe hasła w pliku shadow), ID użytkownika, ID grupy,

opcjonalne informacje fingera (oddzielane przecinkami), katalog domowy oraz powłoka. By dodać nowego użytkownika od ręki, dodaj nową linię na końcu pliku, wpisując odpowiednie informacje.

Informacje które właśnie dodałeś muszą spełniać określone kryteria - inaczej nowy użytkownik może mieć problemy z zalogowaniem się. Po pierwsze, upewnij się że pole z hasłem zawiera `x` oraz że zarówno login, jak i ID użytkownika są unikalne. Przydziel użytkownika do grupy, albo do 100 (grupa “users” w Slackware), albo do innej zdefiniowanej przez ciebie (użyj numeru grupy, nie jej nazwy). Przydziel użytkownikowi właściwy katalog domowy (który za chwilę utworzysz) oraz powłokę (pamiętaj, musi być ona na liście w pliku `/etc/shells`).

Teraz zajmiemy się dodaniem odpowiedniego wpisu w pliku /etc/shadow, który jest miejscem przechowywania zakodowanych haseł. Typowa linijka tego pliku wygląda tak:

```
chris:$1$w9bsw/N9$uwLr2bRER6YyBS.CAEp7R.:11055:0:99999:7:::
```

Ponownie, każda linia dotyczy jednej osoby, zawierając informacje w polach ograniczonych dwukropkami. Pola te to kolejno: login, zakodowane hasło, liczba dni które upłynęły od początku "Epoki" (czyli od 1. stycznia 1970r.) do dnia w którym ostatnio zmieniono hasło, liczba dni po upływie których hasło musi ponownie zostać zmienione, liczba dni przed upływem ważności hasła gdy użytkownik zostanie powiadomiony o tym fakcie, liczba dni po upływie ważności hasła gdy konto zostanie zablokowane, liczba dni od początku "Epoki" po których konto zostanie zablokowane, oraz pole zarezerwowane.

Jak widzicie, większość tych informacji dotyczy czasu ważności konta. Jeżeli nie macie zamiaru tworzyć kont, które wygasną po określonym czasie, to wystarczy wam wypełnić zaledwie kilka pól specjalnymi wartościami. W przeciwnym wypadku potrzeba będzie kilku wyliczeń i podjęcia pewnych decyzji, zanim zdecydujecie się wypełnić te pola. Dla nowego użytkownika w pole z hasłem wpisz przypadkowy ciąg znaków; nie martw się, że hasło będzie nieprawidłowe - i tak zmienisz je za chwilę. Jedyny znak, którego nie możesz użyć w tym zapisie to dwukropek. Pozostaw pole “dni od ostatniej zmiany hasła” puste. W zamian wpisz `0`, `99999` oraz `7` tak, jak widać w przykładzie powyżej, resztę pól pozostawiając pustymi.

(Dla tych, którzy myślą że zobaczywszy zakodowaną postać mojego hasła są w stanie włamać się do mojego systemu: śmiało! Jeżeli umiecie złamać to hasło, to znacie hasło do naszego testowego systemu. Taka informacja jest chyba całkiem użyteczna :))

Wszyscy zwykli użytkownicy w typowym systemie Slackware są członkami grupy “`users`”. Jeżeli chcesz dodać nową grupę, bądź też dodać użytkownika do większej ilości grup, będziesz musiał zmodyfikować plik `/etc/group`. Oto przykładowy wpis w tym pliku:

```
cvs::102:chris,logan,david,root
```

Kolejne jego pola to nazwa grupy, jej hasło, ID oraz lista członków do niej należących (oddzielonych przecinkami). Tworzenie nowej grupy do proste dodanie nowej linijki z unikalnym ID grupy oraz listą wszystkich użytkowników, którzy mają do niej należeć. Wszyscy użytkownicy dopisani do nowej grupy i zalogowani już w systemie, muszą się wylogować i zalogować ponownie, by zmiany odniosły pożądany efekt.

W tym miejscu dobrym pomysłem będzie użycie poleceń `pwck` oraz `grpck` do sprawdzenia, czy zmiany wprowadzone ręcznie przed chwilą są spójne. Jako pierwsze, wydaj komendy `pwck -r` oraz `grpck -r`; opcja `-r` nie służy do wprowadza zmian, wypisuje jedynie zmiany o których dokonanie zostaniesz poproszony, gdy wywołasz te komendy bez parametrów. Możesz użyć tak otrzymanego wyjścia programu by zdecydować, czy zachodzi potrzeba modyfikacji plików bądź uruchomienia `pwck` lub `grpck`, czy też można już pozostawić zmodyfikowane pliki bez konieczności wprowadzania poprawek.

W tym miejscu powinieneś też użyć polecenia `passwd` by przydzielić odpowiednie hasło użytkownikowi. Gdy to już zrobione, użyj komendy `mkdir` by utworzyć nowy katalog domowy wg lokacji, którą wprowadziłeś do pliku `/etc/passwd`, oraz polecenia `chown` do zmiany właściciela tego katalogu na nowego użytkownika.

Usuwanie użytkownika to prosta sprawa usunięcia wszystkich wpisów dotyczących tego użytkownika. Usuń odpowiednie wpisy z plików `/etc/passwd` oraz `/etc/shadow` a także usuń login użytkownika ze wszystkich grup w pliku `/etc/group`. Jeżeli chcesz, możesz też usunąć katalog domowy użytkownika, pliki z pocztą oraz wpisy crontab'a (o ile takowe istnieją).

Usuwanie grup wygląda podobnie: usuń wpis o grupie z pliku `/etc/group`.

12.3 Shutting Down Properly

To bardzo ważne, byś wyłączał swój system właściwie. Zwykłe odłączenie zasilania za pomocą włącznika może poskutkować poważnym uszkodzeniem systemu plików. Gdy system jest włączony, wiele plików jest w użyciu, nawet gdy ty sam nic nie robisz. Pamiętasz z pewnością, że w tle uruchomionych jest wiele procesów. Zarządzają one systemem oraz posiadają wiele otwartych plików. Gdy nagle znika zasilanie, pliki te nie zostają poprawnie zamknięte i mogą zostać uszkodzone. W zależności od tego, które pliki uległy uszkodzeniu, system może stać się nawet całkowicie bezużyteczny! W każdym jednak wypadku uszkodzenia plików zmuszony będziesz przebrnąć przez długą procedurę sprawdzenia systemu plików przy ponownym uruchomieniu.

Jeżeli postawiłeś swój system na partycji z journaling filesystem jak na przykład ext3 albo reiserfs, będziesz częściowo chroniony przed uszkodzeniami systemu plików, zaś sprawdzanie systemu plików przy ponownym uruchomieniu będzie krótsze niż gdybyś używał partycji bez journalingu (np. ext2). Jakkolwiek posiadanie takiego zabezpieczenia nie jest usprawiedliwieniem przy nieprawidłowym wyłączaniu systemu! System plików z journalingiem jest zaprojektowany, by chronić pliki przed wypadkami nie podlegającymi siłą rzeczy pod twoją kontrolę - ale do ochrony przed twoim lenistwem!.

Tak więc, gdy chcesz wyłączyć lub ponownie uruchomić komputer, musisz zrobić do właściwie. Jest na to kilka sposobów; wybierz ten, który uznasz za najbardziej zabawny/wymagający najmniej pracy. Jako, że zamykanie i ponowne uruchomienie są do siebie dosyć podobne, większość sposobów na wyłączenie może też zostać użyte przy restarcie.

Pierwsza i chyba najpopularniejsza metoda to użycie programu `shutdown`(8). `shutdown` może być użyty do wyłączenia bądź restartu komputera o określonej godzinie, może także wyświetlić zalogowanym użytkownikom odpowiednie do swych działań komunikaty.

Najbardziej podstawowe użycie programu `shutdown` do wyłączenia komputera to:

```
# shutdown -h now
```

W tym wypadku nie wyślemy użytkownikom żadnej wiadomości; zobaczą standardową wiadomość wysyłaną przez `shutdown`. “`now`” to czas, kiedy chcemy wyłączyć komputer, zaś “`-h`” oznacza że chcemy wyłączyć komputer (skrót od ang. halt 'zatrzymać'). Nie jest to zbyt przyjazny sposób działania w systemach z wieloma użytkownikami, za to sprawdzi się dobrze na twoim domowym komputerze. W większych systemach lepszą metodą jest danie wszystkim ostrzeżenia o mającym nastąpić wyłączeniu z pewnym wyprzedzeniem:

```
# shutdown -h +60
```

Powyższa komenda spowoduje wyłączenie komputera za godzinę (60 minut), a taki czas z pewnością będzie wystarczający w zwykłym systemie z wieloma użytkownikami. Ważniejsze systemy powinny mieć wyłączenie zaplanowane na długo przed samym faktem, zaś administrator powinien umieścić informację o planowanym zamknięciu systemu w odpowiednich na ogłaszanie tego typu informacji miejscach (e-mail, tablica ogłoszeń, `/etc/motd`, gdziekolwiek...).

Ponowne uruchomienie systemu następuje przy użyciu tej samej komendy, jednak zamiast “`-h`” użyj “`-r`”:

```
# shutdown -r now
```

Możesz użyć tego samego zapisu z komendą `shutdown -r` co przy użyciu `shutdown -h`. Jest jeszcze wiele innych rzeczy, które możesz zrobić za pomocą `shutdown` przy ustalaniu kiedy zatrzymać/zrestartować komputer; przejrzyj strony manuala w poszukiwaniu szczegółów.

Drugi sposób na wyłączenie/restart komputera to użycie poleceń `halt`(8) oraz `reboot`(8). Jak sama nazwa wskazuje, `halt` niezzwłocznie wyłączy komputer, zaś `reboot` uruchomi system ponownie. (`reboot` aktualnie to jedynie link symboliczny do `halt`). Są wywoływane w ten sposób:

```
# halt
# reboot
```

Niskopoziomowym sposobem na zamknięcie systemu jest "rozmowa" bezpośrednio z `init`'em. Wszystkie inne metody są po prostu wygodniejszą wersją tego sposobu, ale też jeżeli chcesz możesz sam bezpośrednio powiedzieć to, używając polecenia `telinit`(8) (zauważ, że w nazwie występuje tylko jedno “l”). Użyte `telinit` poinformuje `init` na który poziom (runlevel) przejść, co spowoduje uruchomienie odpowiedniego skryptu. Skrypt ten zabije bądź też stworzy procesy, zależnie od uruchamianego runlevelu. Działa to przy zamknięciu oraz resetowaniu systemu, gdyż oba posiadają odpowiednie runlevele.

```
# telinit 0
```

Poziom 0 oznacza zatrzymanie systemu. Powiedzenie `init`'owi by przeszedł na poziom 0 spowoduje zabicie wszystkich procesów, odmontowanie systemów plików oraz zatrzymanie komputera. Taki sposób zamykania systemu jest w pełni prawidłowy. W większości laptopów oraz nowszych komputerów spowoduje to też całkowite wyłączenie komputera.

```
# telinit 6
```

Przejście na poziom 6 oznacza zresetowanie maszyny. Wszystkie procesy zostaną zabite, systemy plików odmontowane, komputer uruchomi się ponownie. Podobnie jak powyższa, tak i ta metoda jest całkowicie akceptowalna i prawidłowa.

Dla ciekawskich: kiedy przełączasz system na poziom 0 lub 6, czy to przez wykonanie komendy `shutdown`, `halt`, bądź `reboot`, wykonywany jest skrypt `/etc/rc.d/rc.6`. (Skrypt `/etc/rc.d/rc.0` to tylko link symboliczny do `/etc/rc.d/rc.6`.) Możesz dostosować plik do swoich własnych upodobań - ale bądź pewien, iż wprowadzone zmiany przetestowałeś szczególnie ostrożnie!

Jest jeszcze jeden sposób na ponowne uruchamianie komputera. Wszystkie poprzednie wymagają bycia zalogowanym jako `root`. Ale metoda, o której mowa, nie stawia tego wymogu. Ważne jest tylko, by mieć fizyczny dostęp do klawiatury. Jednoczesne wciśnięcie **Control**+**Alt**+**Delete** spowoduje natychmiastowe ponowne uruchomienie maszyny. (Tak nawiasem mówiąc, naciśnięcie **Control**+**Alt**+**Delete** uruchomi skrypt `shutdown`). Sposób ten nie zawsze zadziała, gdy używasz serwera graficznego X Window - być może przed jego użyciem będziesz musiał wcisnąć **Control**+**Alt**+**F1** (bądź inny klawisz funkcyjny) by przełączyć się na konsolę "nie x-ową"l

I na koniec informacja o pliku, który całkowicie kontroluje każdy aspekt uruchomienia i wyłączenia systemu; jest to plik `/etc/inittab`(5). Ogólnie rzecz biorąc nie ma potrzeby modyfikacji tego pliku, ale wgląd w niego może dać ci wiedzę, dlaczego pewne rzeczy działają tak a nie inaczej. Jak zwykle polecam lekturę stron manuala.

Rozdział 13 Podstawowe komendy sieciowe

Sieć składa się z komputerów połączonych razem. Może być prosta w postaci kilku maszyn połączonych kablem w twoim domu czy biurze, a może też to być skomplikowana, rozległa sieć akademicka. Siecią jest też Internet. Kiedy twój komputer jest częścią sieci, masz dostęp do jej systemów bezpośrednio, lub poprzez specjalne usługi jak np. poczta elektroniczna czy strony www.

Jest sporo różnych programów sieciowych, których możesz używać. Niektóre z nich są przydatne do diagnostyki - aby sprawdzić, czy wszystko działa poprawnie. Inne (jak na przykład klienci poczty e-mail czy przeglądarki internetowe) są przydatne do wykonywania pracy i bycia w kontakcie z innymi osobami.

13.1 `ping`

`ping`(8) wysyła pakiet ICMP `ECHO_REQUEST` do wybranego hosta. Jeśli host odpowie, otrzymasz pakiet ICMP spowrotem. Brzmi groźnie? Po prostu możesz uruchomić “ping” dla danego adresu IP aby sprawdzić, czy dana maszyna działa. Jeśli nie ma odpowiedzi to znaczy, że coś jest nie tak. Oto przykładowa rozmowa pomiędzy dwoma użytkownikami Linuksa:

> *Użytkownik A*: Loki jest znowu wyłączony.
> *Użytkownik B*: Jesteś pewien?
> *Użytkownik A*: Tak, próbowałem go pingować, ale nie odpowiada.

To właśnie takie przypadki sprawiają, że `ping` jest od czasu do czasu użyteczną komendą. Pozwala ona na szybkie sprawdzenie czy dana maszyna jest włączona i podłączona do sieci. Podstawowa składnia to:

```
% ping www.slackware.com
```

Jest oczywiście wiele innych opcji. Sprawdź stronę manuala `ping`(1) aby uzyskać więcej informacji.

13.2 `traceroute`

Slackware'owa komenda `traceroute`(8) to bardzo przydatne narzędzie diagnostyczne. `traceroute` wyświetla kolejne hosty, przez które przechodzi pakiet zanim trafi do celu. Możesz sprawdzić, ile “skoków” dzieli cię od strony Slackware'a wpisując poniższą komendę:

```
% traceroute www.slackware.com
```

Każdy kolejny host zostanie wyświetlony, wraz z jego czasem odpowiedzi. Oto przykładowy wynik wykonania komendy:

```
% traceroute www.slackware.com
traceroute to www.slackware.com (204.216.27.13), 30 hops max, 40 byte packets
1  zuul.tdn (192.168.1.1)  0.409 ms  1.032 ms  0.303 ms
2  207.171.227.254 (207.171.227.254)  18.218 ms  32.873 ms  32.433 ms
3  border-sf-2-0-4.sirius.com (205.134.230.254) 15.662 ms 15.731 ms 16.142 ms
4  pb-nap.crl.net (198.32.128.20)  20.741 ms  23.672 ms  21.378 ms
5  E0-CRL-SFO-03-E0X0.US.CRL.NET (165.113.55.3) 22.293 ms 21.532 ms 21.29 ms
6  T1-CDROM-00-EX.US.CRL.NET (165.113.118.2)  24.544 ms  42.955 ms 58.443 ms
7  www.slackware.com (204.216.27.13)  38.115 ms  53.033 ms  48.328 ms
```

`traceroute` jest podobne do komendy `ping` w tym, że używa pakietów ICMP. Istnieje też wiele innych możliwości oferowanych przez `traceroute`. Opcje te są dokładnie wytłumaczone w manualu.

13.3 Narzędzia DNS

Domain Name Service (w skrócie: DNS) jest tym magicznym protokołem, który umożliwia twojemu komputerowi zamieniać nic nie znaczące nazwy domen jak np. www.slackware.com na znaczące adresy IP jak np. `64.57.102.34`. Komputery nie mogą przesyłać pakietów do www.slackware.com, ale mogą przesyłać pakiety do adresu IP tej domeny. DNS daje nam wygodną metodę zapamiętywania adresów maszyn. Bez DNS'a musieli byśmy pamiętać jaki adres IP należy do jakiego komputera, i to zakładając, że ów adres IP się nie zmieni. Oczywistym jest, że używanie nazw domen dla komputerów jest lepsze, ale jak przyporządkować domenie jej adres IP?

13.3.1 `host`

`host`(1) zrobi to za nas. `host` jest używany do przyporządkowania nazwie jej adresu IP. To bardzo szybkie i proste narzędzie bez rozbudowanej funkcjonalności.

```
% host www.slackware.com
www.slackware.com is an alias for slackware.com.
slackware.com has address 64.57.102.34
```

A co jeśli z jakiegoś powodu chcemy uzyskać nazwę domeny mając adres IP; co wtedy?

13.3.2 `nslookup`

`nslookup` jest wypróbowanym programem, który "zwietrzał" przez lata. `nslookup` został zdeprecjonowany i może zostać usunięty z przyszłych wydań. Nie ma nawet strony manuala dla tego programu.

```
% nslookup 64.57.102.34
Note:  nslookup is deprecated and may be removed from future releases.
Consider using the `dig' or `host' programs instead.  Run nslookup with
the `-sil[ent]' option to prevent this message from appearing.
Server:         192.168.1.254
Address:        192.168.1.254#53

Non-authoritative answer:
www.slackware.com       canonical name = slackware.com.
Name:   slackware.com
Address: 64.57.102.34
```

13.3.3 `dig`

Nowy program "w zagrodzie", wyszukiwarka informacji o domenie (ang. domain information groper), w skrócie `dig`(1), jest programem docelowym do wyszukiwania informacji DNS. `dig` potrafi wydobyć z serwera DNS praktycznie każde dane włączając w to reverse lookups, rekordy A, CNAME, MX, SP, i TXT. `dig` posiada wiele opcji lini poleceń i jeśli jeszcze się nie zaznajomiłeś z nimi, powinieneś przeczytać jego obszerną stronę manuala.

```
% dig @192.168.1.254 www.slackware.com mx

; <<>> DiG 9.2.2 <<>> @192.168.1.254 www.slackware.com mx
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26362
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION Sekcja:
;www.slackware.com.             IN      MX

;; ANSWER Sekcja:
www.slackware.com.      76634   IN      CNAME   slackware.com.
slackware.com.          86400   IN      MX      1 mail.slackware.com.

;; AUTHORITY Sekcja:
```

```
slackware.com.          86400   IN      NS      ns1.cwo.com.
slackware.com.          86400   IN      NS      ns2.cwo.com.

;; ADDITIONAL Sekcja:
ns1.cwo.com.            163033  IN      A       64.57.100.2
ns2.cwo.com.            163033  IN      A       64.57.100.3

;; Query time: 149 msec
;; SERVER: 192.168.1.254#53(192.168.1.254)
;; WHEN: Sat Nov  6 16:59:31 2004
;; MSG SIZE  rcvd: 159
```

To powinno pokazać ci ideę działania komendy `dig`. “@192.168.1.254” to serwer dns który zamierzasz użyć. “www.slackware.com” to domena, którą zamierzasz sprawdzić natomiast “mx” oznacza typ wykonywanego testu. Powyższe zapytanie wykazało, że e-mail wysyłany na serwer `www.slackware.com` zostanie przekierowany na serwer `mail.slackware.com`.

13.4 `finger`

`finger`(1) wyświetli informację o danym użytkowniku. Podajesz tej komendzie nazwę użytkownika bądź jego e-mail a ona połączy się z odpowiednim serwerem i zwróci nazwę użytkownika, biuro, numer telefonu i inne informacje. Oto przykład:

```
% finger johnc@idsoftware.com
```

`finger` potrafi wydobyć nazwę użytkownika, status poczty, numery telefonów, oraz pliki określane jako “dot plan” i “dot project”. Oczywiście zwrócona informacja różni się w zależności od użytego serwera `finger`. Ten dostarczony wraz ze Slackware domyślnie zwraca następujące informacje:

- Nazwa użytkownika
- Numer pokoju
- Domowy numer telefonu
- Numer telefonu w pracy
- Status logowania
- Status poczty
- Zawartość pliku `.plan` z katalogu domowego użytkownika.
- Zawartość pliku `.project` z katalogu domowego użytkownika.

Pierwsze cztery pozycje z powyższej listy mogą zostać zmodyfikowane komendą `chfn`. Zapisuje ona te wartości w pliku `/etc/passwd`. Aby zmienić informację w twoim pliku `.plan` lub `.project`, po prostu otwórz je w swoim ulubionym edytorze tekstowym. Muszą się znajdować w twoim katalogu domowym i musza być nazwane `.plan` oraz `.project`.

Wielu użytkowników wykonuje `finger` swojego konta z obcej maszyny aby sprawdzić szybko czy nie ma nowej poczty. Można też w ten sposób sprawdzić plan czy projekt użytkownika.

Jak wiele komend, `finger` posiada opcje. Sprawdź stronę manuala aby dowiedzieć się o opcjach które mogą się tobie przydać.

13.5 `telnet`

Ktoś kiedyś powiedział, że `telnet`(1) jest najfajniejszą rzeczą jaką kiedykolwiek widział w komputerach. Możliwość zdalnego zalogowania się i pracy na innym komputerze to coś, co wyróżnia Unix'a i Unixopodobne systemy operacyjne od innych systemów.

`telnet` umożliwia ci zalogowanie się do komputera tak, jakbyś siedział przy terminalu. Kiedy twój login i hasło zostaną zweryfikowane, wyświetlony zostanie znak zachęty (ang. shell prompt). Od teraz możesz robić wszystko, na co pozwala konsola tekstowa. Pisać emaile, czytać grupy newsowe, porządkować pliki, etc. Jeśli pracujesz w Xach i połączysz się przez `telnet` do innego komputera, możesz wykonywać programy Xów na zdalnej maszynie a wyświetlając je na własnym komputerze.

Aby zalogować się na zdalnej maszynie, użyj następującej składni:

```
% telnet <hostname>
```

Jeśli host odpowie, otrzymasz ekran logowania. Podaj swój login i hasło. To wszystko. Teraz jesteś w powłoce (ang. shell). Aby zakończyć sesję telnetu, użyj komendy `exit` względnie `logout`

`telnet` nie szyfruje przesyłanych informacji. Wszystko, włącznie z hasłami jest wysyłane jako zwykły tekst. Nie poleca się używania komendy `telnet` w Internecie. Zamiast niego rozważ użycie `Secure Shell`. Szyfruje ona wszystko co wysyła i jest dostępna za darmo.

13.5.1 Inne użycie telnetu

Jako że przekonaliśmy cię do nieużywania protokołu telnet do logowania się na zdalną maszynę, pokażemy kilka innych, użytecznych sposobów użycia komendy `telnet`.

Możesz np. użyć `telnet` do połączenia się z hostem na konkretnym porcie.

```
% telnet <host> [port]
```

To może być przydatne jeśli potrzebujesz szybko sprawdzić konkretną usługę, mając pełną kontrolę nad komendami i chcąc zobaczyć co konkretnie się dzieje. Możesz tą drogą interaktywnie testować bądź używać serwera SMTP, POP3, HTTP, itp.

Na poniższym przykładzie możesz zobaczyć, jak wykonać `telnet` na serwer HTTP na port 80 i zdobyć z niego kilka podstawowych informacji.

Rysunek 13-1. Telnetowanie do serwera HTTP

```
% telnet store.slackware.com 80
Trying 69.50.233.153...
Connected to store.slackware.com.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 25 Apr 2005 20:47:01 GMT
Server: Apache/1.3.33 (Unix) mod_ssl/2.8.22 OpenSSL/0.9.7d
Last-Modified: Fri, 18 Apr 2003 10:58:54 GMT
ETag: "193424-c0-3e9fda6e"
Accept-Ranges: bytes
Content-Length: 192
Connection: close
Content-Type: text/html

Connection closed by foreign host.
%
```

Możesz zrobić to samo dla innych czysto tekstowych protokołów, o ile wiesz z jakim portem należy się

połączyć i jakie są ich komendy.

13.6 SSH (ang. The Secure shell)

Dzisiaj SSH jest tak popularny, jak kiedyć był `telnet`. `ssh` pozwala na połączenie się ze zdalną maszyną i wykonywanie programów tak, jak by były one fizycznie obecne; aczkolwiek `ssh` szyfruje wszystkie dane wędrujące pomiędzy dwoma komputerami więc nawet jeśli inni przechwycą je, nie będą w stanie ich zrozumieć. Typowe połączenie ssh wygląda jak niżej.

```
% ssh carrier.lizella.net -l alan
The authenticity of host 'carrier.lizella.net (192.168.1.253)' can't be
established.
RSA key fingerprint is 0b:e2:5d:43:4c:39:4f:8c:b9:85:db:b2:fa:25:e9:9d.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'carrier.lizella.net' (RSA) to the list of
known hosts.
Password: password
Last login: Sat Nov  6 16:32:19 2004 from 192.168.1.102
Linux 2.4.26-smp.
alan@carrier:~$ ls -l MANIFEST
-rw-r--r--  1 alan users 23545276 2004-10-28 20:04 MANIFEST
alan@carrier:~$ exit
logout
Connection to carrier.lizella.net closed.
```

Tutaj widzisz jak wykonuję połączenie ssh do `carrier.lizella.net`, i sprawdzam uprawnienia do pliku `MANIFEST`.

13.7 email

Poczta elektroniczna jest jedną z najpopularniejszych usług Internetu. W 1998 zarejestrowano więcej wysłanych listów elektronicznych niż tradycyjnych. Jest to jak widać szeroko stosowane i przydatne narzędzie.

W Slackware, dostarczamy standartowy serwer pocztowy i kilka klientów poczty e-mail. Wszystkie aplikacje klienckie omówione poniżej są tekstowe. Wielu użytkowników Windows może być przeciwnych temu, ale zobaczysz, że tekstowy klient jest bardzo wygodny, zwłaszcza kiedy sprawdzasz e-mail zdalnie. Jeśli chcesz, jest wiele graficznych klientów e-mail jak np. Kmail z pakietu KDE. Jeśli chcesz używać jednego z nich sprawdź jego menu Pomoc.

13.7.1 `pine`

`pine`(1) is not `elm` (ang. pine nie jest elm'em). Taka jest geneza nazwy. Uniwersytet w Waszyngtonie stworzył własny program do news'ów i poczty elektronicznej w wyniku zapotrzebowania na prosty czytnik mail dla studentów. `pine` jest jednym z najpopularniejszych klientów poczty elektronicznej używanych dzisiaj i jest dostępny dla niemal każdej odmiany systemu Unix czy nawet Windows.

Rysunek 13-2. Główne menu Pine

Zobaczysz menu komend i wiersz przycisków na dole. `pine` jest w rzeczy samej złożonym programem, więc nie będziemy omawiać tutaj każdej jego możliwości.

Aby zobaczyć, co jest w twojej poczcie przychodzącej (ang. inbox), wciśnij **`i`**. Twoje wiadomości zostaną wylistowane z ich datami, autorem i tematem. Podświetl wiadomość którą chcesz i wciśnij **`enter`** aby ją

zobaczyć. Naciścięcie **r** pozwoli na utworzenie odpowiedzi na tę wiadomość. Kiedy napiszesz już odpowiedź, użyj **Ctrl**+**X** aby ją wysłać. Możesz też nacisnąć **i** aby powrócić do spisu wiadomości.

Jeśli chcesz usunąć wiadomość, naciścij **d**. Zaznaczy to podświetloną wiadomość do skasowania. `pine` usuwa wiadomości kiedy opuszczasz program. `pine` także pozwala ci trzymać twoją pocztę w folderach. Możesz uzyskać listę folderów wciskając **l**. Mając otworzoną wiadomość, wciśnij **s** aby zachować ją w innym folderze Program zapyta o nazwę folderu, w którym ma zapisać wiadomość.

`pine` oferuje bardzo wiele możliwości; powinieneś spojrzeć na stronę manuala w celu usyskania więcej informacji. Będzie ona zawierała najnowsze informacje o programie.

13.7.2 `elm`

`elm`(1) jest kolejnym popularnym tekstowym klientem poczty elektronicznej. Choć nie tak przyjazny dla użytkownika jak `pine`, był on dostępny znacznie dłużej.

Rysunek 13-3. Elm main screen

Domyślnie, jesteś w swojej skrzynce odbiorczej. Wiadomości są wylistowane wraz z ich numerami, datą, nadawcą i tematem. Użyj przycisków strzałek aby podświetlić konkretną wiadomość. Naciśnij **Enter** aby przeczytać wiadomość.

Aby stworzyć nową wiadomość, wciśnij **m** w głównym ekranie. Przycisk **d** zaznaczy wiadomość do skasowania. Natomiast **r** odpowie na wiadmomość którą aktualnie czytasz. Wszystkie te klawisze są wyświetlone na dole ekranu wraz ze znakiem zachęty.

Strona manuala opisuje `elm` w większych szczegółach, więc prawdopodobnie będziesz musiał po nią sięgnąć zanim zaczniesz używać `elm'a`.

13.7.3 `mutt`

“Wszystkie czytniki e-mail są do d... . Ten trochę mniej.” `mutt` swój oryginalny interfejs zawdzięcza programowi `elm` z dodanymi opcjami i możliwościami z innych popularnych klientów poczty, dając w rezultacie hybrydę.

Kilka cech programu `mutt`:

- wsparcie dla kolorów
- wątki wiadomości
- wsparcie dla MIME i PGP/MIME
- wsparcie dla pop3 oraz imap
- wsparcie dla różnych formatów skrzynki (mbox, MMDF, MH, maildir)
- *wysoko* personalizowalny

Rysunek 13-4. Główny ekran programu Mutt

Jeśli szukasz klienta e-mail który pozowli ci na totalną kontrolę nad wszystkim, to `mutt` spodoba ci się. Wszystkie domyślne opcje mogą być spersonalizowane, skróty klawiaturowe mogą zostać zmienione. Możesz też dodać makra.

Prawdopodobnie będziesz chciał spojrzeć na stronę manuala do pliku `muttrc`, która powie ci, jak skonfigurować wszystko. Możesz też przejrzeć dołączony przykładowy plik `muttrc`.

13.7.4 `nail`

`nail`(1) jest sterowanym z lini poleceń klientem poczty. Jest bardzo prymitywny i oferuje praktycznie nic jeśli chodzi o interfejs. Aczkolwiek, mailx jest bardzo poręczny, jeśli chcesz coś szybko wysłać, napisać korespondencję seryjną, przetestować swoją instalację MTA czy coś podobnego. Zauważ, że Slackware tworzy link symboliczny do komendy `nail` jako `/usr/bin/mail` i `/usr/bin/mailx`. Wszystkie te trzy komendy uruchamiają ten sam program. Faktem jest, że najczęściej zobaczysz odwołanie do `nail` jako `mail`.

Podstawowa składnia to:

```
% mailx <subject> <to-addr>
```

`mailx` czyta treść wiadomości ze standartowego wejścia. Tak więc możesz przekierować jakiś plik potokiem na tę komendę aby wysłać jego zawartość, albo też napisać tekst i nacisnąć **Ctrl**+**D** kiedy skończysz pisać.

Oto przykład wysłania pliku źródłowego do innej osoby.

```
% cat randomfunc.c | mail -s "Here's that function" asdf@example.net
```

Strona manuala wyjaśnia więcej, co `nail` potrafi robić, więc pewnie będziesz musiał tam zajrzeć zanim użyjesz programu.

13.8 Przeglądarki

Pierwszą rzeczą jaka nasuwa się na myśl kiedy słyszy się słowo Internet jest “surfowanie po necie”, albo też przeglądanie stron internetowych używając do tego przeglądarek internetowych. Jest to prawdopodobnie najczęściej stosowana usługa Internetu przez przeciętnego użytkownika.

Slackware dostarcza zarówno popularne graficzne przeglądarki internetowe w katalogu “XAP”, jak też przeglądarki tekstowe w katalogu “N”. Przyjrzymy się po krótce najczęściej stosowanym opcjom poniżej.

13.8.1 `lynx`

`lynx`(1) jest tekstową przeglądarką. Jest to szybka metoda wyszukania czegoś w Internecie. Czasem grafika niepotrzebnie przeszkadza jeśli wiesz czego szukasz.

Aby uruchomić `lynx`, po prostu wpisz w konsoli: `lynx`.

```
% lynx
```

Rysunek 13-5. Domyślna strona startowa lynx'a

Możesz podać stronę, którą ma na starcie otworzyć `lynx`:

```
% lynx http://www.slackware.com
```

`lynx` wyświetla klawisze komend wraz z ich funkcją na dole ekranu. Strzałki w górę i w dół służą do poruszania się po dokumencie, **`Enter`** uruchamia podświetlony link natomiast **`strzałka w lewo`** wraca do poprzedniej strony. Naciśnięcie **`d`** ściągnie na dysk aktualnie podświetlony plik. Komenda **`g`** wyświetla dialog Idź, w którym możesz podać URL do wyświetlenia.

Jest wiele innych komend w programie `lynx`. Możesz sięgnąć po strony manuala, lub też wcisnąć **`h`** aby uzuskać ekran pomocy lynxa na którym znajdziesz dodatkowe informacje.

13.8.2 `links`

Tak jak `lynx`, `links` jest tekstową przeglądarką, gdzie cała nawigacja odbywa się z klawiatury. Aczkolwiek, kiedy naciśniesz klawisz **`Esc`**, uaktywni się bardzo wygodne menu na górze ekranu. To gwarantuje łatwość użycia, bez konieczności nauki wszystkich skrótów klawiszowych. Ludzie nie używający na co dzień przeglądarek tekstowych docenią tę opcję.

`links` wydaje się mieć lepszą obsługę ramek i tabel w porównaniu z `lynx`.

Rysunek 13-6. Links z otwartym menu plik

13.8.3 `wget`

`wget`(1) jest narzędziem linii poleceń które ściąga pliki z danego URL. Chociaż nie jest właściwą przeglądarką, `wget` jest przede wszystkim używany do skopiowania całości lub części strony do przeglądania offline, albo do szybkiego ściągania pojedyńczych plików z serwerów HTTP lub FTP. Podstawowa składnia to:

```
% wget <url>
```

Możesz także podać opcje. Np. poniższa komenda ściągnie stronę Slackware:

```
% wget --recursive http://www.slackware.com
```

`wget` stworzy katalog `www.slackware.com` i tam zapisze pliki, tak jak jest to na stronie.

`wget` może też ściągać pliki z FTP; po prostu podaj adres FTP zamiast HTTP.

```
% wget ftp://ftp.gnu.org/gnu/wget/wget-1.8.2.tar.gz
--12:18:16--  ftp://ftp.gnu.org/gnu/wget/wget-1.8.2.tar.gz
           => `wget-1.8.2.tar.gz'
Resolving ftp.gnu.org... done.
Connecting to ftp.gnu.org[199.232.41.7]:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done.   ==> PWD ... done.
==> TYPE I ... done. ==> CWD /gnu/wget ... done.
==> PORT ... done.   ==> RETR wget-1.8.2.tar.gz ... done.
Length: 1,154,648 (unauthoritative)

100%[====================================>] 1,154,648     209.55K/s    ETA 00:00

12:18:23 (209.55KB/s) - `wget-1.8.2.tar.gz' saved [1154648]
```

`wget` posiada wiele innych opcji, które robią z niego dobry program do specyficznych skryptów (mirroring stron itp.). Na stronie manuala znajdziesz dodatkowe informacje.

13.9 Klienci FTP

FTP to skrót od File Transfer Protocol (protokół transferu plików). pozwala na wysyłanie i odbieranie plików pomiędzy dwoma komputerami. Jest serwer FTP i klient FTP. W tej sekcji omówimy klienta.

Dla ciekawości, “client” to ty. “serwer” to komputer który odpowie na twe żądanie FTP i pozwoli na zalogowanie. Będziesz ściągał z i wysyłał pliki na serwer. Klient nie potrafi przyjmować połączeń FTP, potrafi jedynie łączyć się z serwerami FTP.

13.9.1 `ftp`

Aby połączyć się z serwrem FTP, po prostu uruchom komendę `ftp`(1) i podaj host:

```
% ftp <hostname> [port]
```

Jeśli na hoście uruchomiony jest serwer FTP, zapyta on o nazwę użytkownika i hasło. Możesz zalogować się jako ty, lyb też jako osoba anonimowa - “anonymous”. Anonimowe strony FTP są bardzo popularne zwłaszcza wśród archiwów oprogramowania. Dla przykładu, aby ściągnąć Slackware Linux z FTP, musisz użyć anonimowego FTP.

Kiedy już się połączysz, zobaczysz znak zachęty `ftp>`. Są specjalne komendy dla FTP, ale są one podobne do standardowych komend. Poniżej jest lista podstawowych komend wraz z ich przeznaczeniem:

Tabela 13-1. komendy `ftp`

Komenda	Działanie
`ls`	Listuje pliki
`cd <nazwakatalogu>`	Zmienia katalog
`bin`	Ustawia binarny tryb transferu
`ascii`	Ustawia tryb ASCII transferu
`get <nazwapliku>`	Ściąga plik
`put <nazwapliku>`	Wysyła plik
`hash`	Przełącza wskaźnik statystyk znaku hash
`tick`	Przełącza wskaźnik licznika bajtów
`prom`	Przełącza na tryb interaktywny dla ściągania
`mget <maska>`	Ściąga plik lub grupę plików; dozwolone są znaki wildcard
`mput <maska>`	Wysyła plik lub grupę plików; dozwolone są znaki wildcard
`quit`	Wylogowuje z serwera FTP

Możesz także użyć poniższych komend, które raczej nie potrzebują wytłumaczenia: `chmod`, `delete`, `rename`, `rmdir`. Kompletną listę komend wraz z ich znaczeniem otrzymasz wpisując **`help`** lub **`?`**.

FTP jest dosyć prostym w użytkowaniu progrem, ale nie posiada interfejsu użytkownika, do którego wielu z nas jest dziś przyzwyczajonych. Strona manuala omawia opcje linii poleceń komendy `ftp`(1).

```
ftp> ls *.TXT
200 PORT command successful.
```

```
150 Opening ASCII mode data connection for /bin/ls.
-rw-r--r--   1 root     100         18606 Apr  6  2002 BOOTING.TXT
-rw-r--r--   1 root     100         10518 Jun 13  2002 COPYRIGHT.TXT
-rw-r--r--   1 root     100           602 Apr  6  2002 CRYPTO_NOTICE.TXT
-rw-r--r--   1 root     100         32431 Sep 29 02:56 FAQ.TXT
-rw-r--r--   1 root     100        499784 Mar  3 19:29 FILELIST.TXT
-rw-r--r--   1 root     100        241099 Mar  3 19:12 PACKAGES.TXT
-rw-r--r--   1 root     100         12339 Jun 19  2002 README81.TXT
-rw-r--r--   1 root     100         14826 Jun 17  2002 SPEAKUP_DOCS.TXT
-rw-r--r--   1 root     100         15434 Jun 17  2002 SPEAK_INSTALL.TXT
-rw-r--r--   1 root     100          2876 Jun 17  2002 UPGRADE.TXT
226 Transfer complete.
ftp> tick
Tick counter printing on (10240 bytes/tick increment).
ftp> get README81.TXT
local: README81.TXT remote: README81.TXT
200 PORT command successful.
150 Opening BINARY mode data connection for README81.TXT (12339 bytes).
Bytes transferred: 12339
226 Transfer complete.
12339 bytes received in 0.208 secs (58 Kbytes/sec)
```

13.9.2 `ncftp`

`ncftp`(1) (wymawiany "Nik-F-T-P") jest alternatywnym do tradycyjnego ftp klientem i jest dostarczany ze Slackware. To nadal program tekstowy, ale oferuje więcej ulepszeń niż `ftp`, włączając:

- Autouzupełnianie klawiszem Tab
- Posiada zakładki
- Bardziej liberalne używanie znaków wildcard
- Historia komend

Domyślnie, `ncftp` spróbuje zalogować się anonimowo na serwer który podasz. Możesz wymusić na `ncftp` aby ukazał znak zachęty opcją "`-u`". Kiedy zalogujesz się, możesz używać tych samych komend jak w `ftp`, ale zauważysz ładniejszy interfejs, który działa bardziej jak `bash`.

```
ncftp /pub/linux/slackware > cd slackware-current/
Please read the file README81.TXT
  it was last modified on Wed Jun 19 16:24:21 2002 - 258 days ago
CWD command successful.
ncftp ...ware/slackware-current > ls
BOOTING.TXT              FAQ.TXT                  bootdisks/
CHECKSUMS                FILELIST.TXT             extra/
CHECKSUMS.asc            GPG-KEY                  isolinux/
CHECKSUMS.md5            PACKAGES.TXT             kernels/
CHECKSUMS.md5.asc        PRERELEASE_NOTES         pasture/
COPYING                  README81.TXT             rootdisks/
COPYRIGHT.TXT            SPEEKUP_DOCS.TXT         slackware/
CRYPTO_NOTICE.TXT        SPEEK_INSTALL.TXT        source/
CURRENT.WARNING          Slackware-HOWTO
ChangeLog.txt            UPGRADE.TXT
ncftp ...ware/slackware-current > get README81.TXT
README81.TXT:                                        12.29 kB  307.07 kB/s
```

13.10 Rozmowa z innymi ludźmi

13.10.1 `wall`

`wall`(1) jest szybką metodą napisania wiadomości do innych użytkowników systemu. Podstawowa składnia:

```
% wall [plik]
```

Zawartość pliku [plik] zostanie wyświetlone na terminalach wszystkich zalogowanych aktualnie użytkowników. Jeśli nie podasz pliku, wall będzie czytał ze standartowego wejścia, więc po prostu wpiszesz wiadomość i zakończysz ją sekwencją **Ctrl**+**d**.

`wall` nie ma wielu możliwości, oprócz np. powiadomienia innych użytkowników, że robisz poważne prace w systemie czy też nawet zamierzasz go zrestartować, tak więc mają czas aby zachować ich pracę i wylogować się :)

13.10.2 `talk`

`talk`(1) pozwala rozmawiać dwóm osobom. Dzieli on ekran w połowie w poziomie. Aby zacząć rozmowę z innym użytkownikiem użyj komendy:

```
% talk <osoba> [nazwatty]
```

Rysunek 13-7. Dwie osoby w sesji `talk`

Jeśli podasz tylko nazwę użytkownika, zostanie to zrozumiane, że chodzi o lokalnego użytkownika, tacy też tylko zostaną wzięci pod uwagę. Nazwa tty jest wymagana jeśli chcesz zadzwonić pod specyficzny terminal (jeśli użytkownik jest zalogowany na więcej niż jednym). Wymagane informacje dla `talk` mogą być zdobyte przez komendę `w`(1).

`talk` może też porozumiewać się z użytkownikami na obcych hostach. Jako nazwę użytkownika po prostu podajesz jego adres e-mail. `talk` spróbuje połączyć się z tym userem na tym hoście.

`talk` jest w pewnym sensie ograniczony. Można nim rozmawiać tylko we dwóch, i to w trybie half-duplex.

13.10.3 `ytalk`

`ytalk`(1) jest wstecznie kompatybilnym zamiennikiem dla `talk`. Jest dostarczony z systemem Slackware jako komenda `ytalk`. Składnia jest podobna ale ma kilka różnic:

```
% ytalk <nazwauzytkownika>[#nazwatty]
```

Rysunek 13-8. Dwaj użytkownicy w sesji `ytalk`

Nazwa użytkownika i terminalu podaje się jak w talk tyle że musisz podać je razem rozdzielając znakiem hash (#).

ytalk oferuje wiele zalet:

- Można romawiać w więcej niż 2 osoby.
- Posiada menu opcji wywoływane klawiszem **`Esc`**.
- Możesz używać shella podczas trwania sesji.

- Plus inne...

Jeśli jesteś administratorem serwera, będziesz musiał upewnić się, czy port `ntalk` jest włączony w `/etc/inetd.conf`. `ytalk` potrzebuje tego do poprawnej pracy.

Rozdział 14 Bezpieczeństwo

Bezpieczeństwo każdego systemu jest bardzo ważne; może zapobiec atakom ludzi na twoj komputer, jak również ochronić ważne dane. Rozdział ten jest w całości o tym jak rozpocząć zabezpieczanie Slackware przed niepożądanymi interwencjami z zewnątrz. Zapamiętaj że to tylko wprowadzenie do zabezpieczania systemu; zabezpieczanie to proces nie stan.

14.1 Wyłączanie usług

Pierwszym krokiem po instalacji Slackware powinno być wyłączenie wszystkich usług, których nie potrzebujesz. Każda usługa może potencjalnie powodować zagrożenie bezpieczeństwa, więc jest to ważne aby uruchamiać tak mało usług jak tylko to możliwe. Usługi sa uruchamiane głównie z dwóch miejsc - `inetd` i skryptów startowych (ang. init scripts)

14.1.1 Usługi uruchamiane z `inetd`

Wiele daemonów dostarczanych z Slackware uruchamiane jest z `inetd`(8). `inetd` jest daemonem, który nasłuchuje na wszystich portach używanych przez usługi, które zostały skonfigurowane by przez niego być uruchamiane. W momencie nadejścia połączenia tworzona jest instancja odpowiedniego daemona. Daemony uruchamiane przez `inetd` mogą być wyłączone przez zakomentowanie odpowiednich linii w `/etc/inetd.conf`. Aby to zrobić otwórz plik w ulubionym edytorze (np. `vi`). Zobaczysz linijki podobne do tej:

```
telnet stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
```

Możesz wyłączyć tą usługę i wszstkie inne, których nie potrzebujesz, komentując je (dodając symbol `#` (hash) na początku linii). Powyższa linia będzie wyglądać:

```
#telnet stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
```

Po zrestartowaniu `inetd`, usługa ta będzie wyłączona. Możesz zrestartować `inetd` komendą:

```
# kill -HUP $(cat /var/run/inetd.pid)
```

14.1.2 Usługi uruchamiane przez skrypty startowe

Pozostałe usługi uruchamiane podczas startu systemu, wywoływane są przez skrypty startowe (które znajdują się w `/etc/rc.d/`). Można je wyłączyć na dwa różne sposoby - przez usunięcie praw do wykonywania odpowiedniego skryptu lub zakomentowanie odpowiedniej linii w skrypcie.

Na przykład, SSH uruchamiane jest przez własny skrypt (`/etc/rc.d/rc.sshd`). Możesz go wyłączyć przez:

```
# chmod -x /etc/rc.d/rc.sshd
```

W przypadku gdy usługa nie ma swojego skryptu startowego, wyłączenie jej może odbyć się przez

zakomentowanie odpowiednich linii. Na przykład daemon portmap jest uruchamiany przez poniższe wywołanie w pliku `/etc/rc.d/rc.inet2`:

```
# This must be running in order to mount NFS volumes.
# Start the RPC portmapper:
if [ -x /sbin/rpc.portmap ]; then
  echo "Starting RPC portmapper:  /sbin/rpc.portmap"
  /sbin/rpc.portmap
fi
# Done starting the RPC portmapper.
```

Można ho wyłączyć przez dodanie symbolu # na początku linii, które jeszcze go nie posiadają, na przykład:

```
# This must be running in order to mount NFS volumes.
# Start the RPC portmapper:
#if [ -x /sbin/rpc.portmap ]; then
#  echo "Starting RPC portmapper:  /sbin/rpc.portmap"
#  /sbin/rpc.portmap
#fi
# Done starting the RPC portmapper.
```

Zmiany te odniosą sukces po restarcie albo po przełączeniu sie spowrotem na runlevel 3 lub 4 (po wcześniejszym wejściu na runlevel 1). Możesz to zrobic wpisując w konsoli (po zmianie runlevela konieczne jest ponowne logowanie):

```
# telinit 1
# telinit 3
```

14.2 Kontrola dostępu do hosta

14.2.1 `iptables`

`iptables` jest programem konfiguracyjnym filtra pakietów dla Linuksa 2.4 i powyżej. Jądro 2.4 (2.4.5, żeby być dokładnym) zostało pierwszy raz wprowadzone do Slackware (jako opcja) w wersji 8.0 i ustawione jako domyślne w Slackware 8.1. Sekcja ta porusza jedynie podstawy iptables; warto odwiedzić http://www.netfilter.org/ by uzyskać więcej informacji. Komendy poniżej opisane wprowadzane są do `/etc/rc.d/rc.firewall`, który musi mieć ustawione prawa wykonywania aby odniosły efekt przy starcie systemu. Zauważ że niepoprawna konfiguracja `iptables` może zablokować twój komputer. Jeżeli choć trochę wątpisz w swoje umiejętności w tej dziedzinie, upewnij się, że masz lokalny dostęp do maszyny.

Pierwszą rzeczą jaką ludzie powinni zrobić jest ustawienie domyślnej polityki dla każdego wchodzącego łańcucha na DROP:

```
# iptables -P INPUT DROP
# iptables -P FORWARD DROP
```

Kiedy wszystko jest odrzucane (denied), możesz zacząć dopuszczać. Pierwszą rzeczą do dopuszczenia jest cały ruch dla ustanowionych już sesji.

```
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Aby nie przeszkadzać aplikacjom, które komunikują się używając adresu loopback, zazwyczaj dobrze jest dodać taką regułę:

```
# iptables -A INPUT -s 127.0.0.0/8 -d 127.0.0.0/8 -i lo -j ACCEPT
```

Reguła ta dopuszcza ruch wchodzący i wychodzący z 127.0.0.0/8 (127.0.0.0 - 127.255.255.255) poprzez interfejs loopback (`lo`). Podczas tworzenia reguł, warto dobrze się zastanowić. Może się zdażyć, że regułka będzie zezwalać na cos co w żadnym wypadku nie będzie pożądane. Z drugiej strony - jeżeli reguły zezwalają na zbyt mało - musi ich być dużo :-).

Następną rzeczą do zrobienia będzie umożliwienie dostepu do określonych usług działających na komuterze. Na przykład chcąc uruchomić serwer www, należy użyć regułki podobnej do tej:

```
# iptables -A INPUT -p tcp --dport 80 -i ppp0 -j ACCEPT
```

Umożliwi to dostęp z każdego komputera na 80 port twojej maszyny przez interfejs `ppp0`. Możesz chcieć ograniczyć dostep do usługi tak żeby tylko okreśone komputery miały taki dostep. Poniższa reguła ta umożliwia dostęp do usługi z hosta `64.57.102.34`:

```
# iptables -A INPUT -p tcp -s 64.57.102.34 --dport 80 -i ppp0 -j ACCEPT
```

Dopuszczanie ruchu ICMP może być użyteczne w celach diagnostycznych. Aby to umożliwić musisz użyć regułki:

```
# iptables -A INPUT -p icmp -j ACCEPT
```

Wiekszość osób zechce ustawić Network Address Translation (NAT) na komputerach bramach, tak żeby pozostałe komputery w ich sieci miały dostep do Internetu poprze nie. W tym celu należy użyć następującej regułki:

```
# iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
```

Ważne jest włączenie IP forwarding. Można to zrobić czasowo używając nstepującego polecenia:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Aby aktywować IP forwarding na stałe (np. tak, żeby zmiany zachowały sie po restarcie) należy wyedytować `/etc/rc.d/rc.inet2` i zmienić nastepująca linie:

```
IPV4_FORWARD=0
```

...na:

```
IPV4_FORWARD=1
```

Więcej o NAT dowiesz się z NAT HOWTO.

14.2.2 `tcpwrappers`

`tcpwrappers` kontrolują dostęp do daemonów bardziej na poziomie aplikacji, niż na poziomie IP. Może to zapewnić dodatkową warstwę bezpieczeństwa w czasie gdy kontrola dostępu z poziomu IP (np. Netfilter) nie funkcjonuje poprawnie. Może się tak stać w przypadku rekompilacji jądra jeżeli zapomni się wkompilować obsługi iptables. Ochrona na poziomie IP zawiedzie jednak tcpwrappers wciąż będzie pełnić swoją funkcję.

Dostęp do usług chronionych przez tcpwrappers może być kontrolowany przy użyciu `/etc/hosts.allow` i `/etc/hosts.deny`.

Większość ludzi będzie mieć tylko jedną linię w swoim pliku `/etc/hosts.deny` - domyślnie odmowa dostępu do wszystkich daemonów. Będzie ona wyglądać następująco:

```
ALL : ALL
```

Kiedy to już jest zrobione, możesz skoncentrować się na umożliwianiu dostępu do usług z określonych hostów, domen czy zakresów IP. Można tego dokonać w pliku `/etc/hosts.allow`.

Wielu użytkowników zacznie od zezwolenia na połączenia z `localhosta`. Można to osiągnąć używając:

```
ALL : 127.0.0.1
```

Aby umożliwić dostęp do SSHd z `192.168.0.0/24`, możesz użyć jednej z podanych reguł:

```
sshd : 192.168.0.0/24
sshd : 192.168.0.
```

Możliwe jest także zastrzeżenie dostępu do hosta z określonych domen. Można to zrobić używając następujacej reguły:

```
sshd : .slackware.com
```

14.3 Aktualizacja systemu

14.3.1 `slackware-security` listy mailingowe

Kiedy problemy bezpieczeństwa dotykają Slackware, wysyłany jest email do wszystkich zapisanych na liscie mailingowej `slackware-security@slackware.com`. Raporty są wysyłane w przypadku znalezienia usterek w Slackware, wyłączając oprogramowanie zawarte w `/extra` i `/pasture`. Wysyłane informacje zawierają szczegóły na temat dostepnych uaktualnień lub czynności jakich należy się podjąć w celu usunięcia usterki.

W Sekcja 2.2.2 dowiesz się jak zapisać się na listę mailingową.

14.3.2 Katalog `/patches`

Zawsze kiedy wypuszczane są uaktualnienia paczek dla Slackware (zazwyczaj w celu naprawienia problemów z bezpieczeństwem, dla juz opublikowanej wersji Slackware), umieszczane są one w katalogu `/patches`. Pełna ścieżka do tych uaktualnień zależy od serwera lustrzanego jakiego używasz, ale wygląda mniej więcej tak: `/path/to/slackware-x.x/patches/`.

Przed instalacją paczek dobrym nawykiem jest sprawdzenie sumy `md5sum` paczki. `md5sum`(1) jest narzędziem wywoływanym z linii poleceń, które tworzy “unikalny” ciąg (mathematical hash). Jeśli pojedyńczy bit pliku został zmieniony, wygeneruje to inną wartość md5sum.

```
% md5sum package-<ver>-<arch>-<rev>.tgz
6341417aa1c025448b53073a1f1d287d  package-<ver>-<arch>-<rev>.tgz
```

Należy go porównać z odpowiednim wpisem w `CHECKSUMS.md5` w katalogu `slackware-`*`$VERSION`* (również w katalogu `/patches` dla uaktualnień) lub w mailu z listy `slackware-security`.

Jeśli masz plik z wartościami md5sum, możesz użyć go zamiast źródła z opcją `-c md5sum`.

```
# md5sum -c CHECKSUMS.md5
./ANNOUNCE.10_0: OK
./BOOTING.TXT: OK
./COPYING: OK
./COPYRIGHT.TXT: OK
./CRYPTO_NOTICE.TXT: OK
./ChangeLog.txt: OK
./FAQ.TXT: FAILED
```

Jak widzisz, każdy plik którego suma `md5sum` jest zgodna, jest wypisany z "`OK`" podczas gdy pliki, które mają inna wartość sumy są oznaczone "`FAILED`".

Rozdział 15 Pliki archiwów

15.1 `gzip`

`gzip`(1) jest programem kompresującym GNU. Kompresuje pojedyncze pliki. Podstawowe użycie jest następujace: bierze pojedyńczy plik i kompresuje go:

```
% gzip filename
```

Wynikowy plik będzie nosić nazwę *`filename`*`.gz` i zazwyczaj będzie mniejszy niż plik wejsciowy. Zauważ że `filename.gz` zastąpi `filename`. Oznacza to że `filename` przestanie istnieć, pomimo że gzippowana kopia pozostanie. Zwykły tekst kompresuje się bardzo ładnie, podczas gdy obrazy jpeg, mp3, i inne podobne pliki nie skompresują się zbytnio ponieważ już są skompresowane. Podstawowe użycie jest wynikiem pomiędzy końcowym rozmiarem pliku a czasem kompresji. Maksymalną kompresję można osiagnąć przez:

```
% gzip -9 filename
```

Zajmie to więcej czasu ale wynik będzie tak mały jak tylko pozwala na to `gzip`. Używajac mniejszych wartości jako opcji spowoduje szybszą kompresje, ale mniejszy poziom kompresji.

Dekompresje pliku gzipped można przeprowadzić dwoma poleceniami, które w rzeczywistości są tym samym programem. `gzip` rozpakuje każdy plik z rozpoznawalnym rozszerzeniem. Rozszerzenie to może być jednym z: `.gz`, `-gz`, `.z`, `-z`, `.Z`, or `-Z`. Pierwsza metoda to wywolanie `gunzip`(1) na pliku, na przykład:

```
% gunzip filename.gz
```

Pozostawi to rozpakowaną wersję pliku wejściowego w bieżącym katalogu i rozszerzenie `.gz` zosatnie usunięte z nazwy pliku. `gunzip` jest tak naprawdę częścią `gzip` i jest identyczny do `gzip -d`. Jak naprzykład `gzip` jest często nazywane `gunzip`, jako nazwa która lepiej brzmi. :^)

15.2 `bzip2`

`bzip2`(1) jest alternatywnym programem kompresującym instalowanym w Slackware Linux. Używa odmiennego od `gzip` algorytmu kompresji. Głowną zaletą `bzip2` jest rozmiar skompresowanego pliku. `bzip2` prawie zawsze kompresuje lepiej niż `gzip`. W niektórych przypadkach daje to niesamowicie małe pliki. Moze to być ogromna zaleta dla osób używających połączeń modemowych. Pamietaj także że kiedy pobierasz plik z publicznego ftp kultura internetowa (netykieta) zobowiazuje do pobierania pliku `.bz2` zamiast `.gz`, jako oznakę zrozumienia dla hojności osób utrzymujących hosting na serwerze.

Wadą `bzip2` jest to że zajmuje więcej zasobów CPU niż `gzip`. Oznacza to że bzippowanie pliku generalnie

zabiera więcej i zużywa wiecej CUP niż gzippowanie tego samego pliku. Kiedy rozważasz którego programu kompresujacego użyć, musisz zrównoważyc prędkość z poziomem kompresji i wybrać to co będzie ważniejsze.

Używanie `bzip2` jest niemalże identyczne z użyciem `gzip`, więc nie poświecimy wiele czasu na omawianiu tego. Podobnie jak `gunzip`, `bunzip2` jest identyczny do `bzip2 -d`. Podstawowa różnica w praktycznym użyciu jest to że `bzip2` używa rozszerzenia `.bz2`.

```
% bzip2 filename
% bunzip2 filename.bz2
% bzip2 -9 filename
```

15.3 `tar`

`tar`(1) to the GNU tape archiver. Zbiera kilka plików lub katalogów i tworzy jesten duży plik. Pozwala to na kompresje całych katalogów co jest niemożliwe używając `gzip` czy `bzip2`. `tar` ma wiele opcji, które są wyszczególnione na stronach manuala. Sekcja ta omawia tylko najpopularniejsze opcje komendy `tar`.

Najpopularniejszym użyciem komendy `tar` jest dekompresja i rozpakowywanie paczek które pobrałes ze strony intrenetowej czy serwera ftp. Większość z nich posiada rozszerzenie `.tar.gz`. Jest ono popularnie zwane “tarball”. Oznacz to że kilka plików zostało zarchiwizowanych używając `tar` i skompresowanych używając `gzip`. Możesz także znaleźć je jako pliki `.tar.Z`. To jest to samo tylko że użyto starszego systemu unixowego.

Alternatywnie mozesz znalezc pliki `.tar.bz2`. Zrodla kernela sa tak rozprowadzane poniewaz sa mniejsze. Jak pewnie sie domyslasz jest to kikla plikow zarchiwizowanych `tar` a nastepnie zbzippowany.

Możesz dostać się do wszystkich plików w tym archiwum używając `tar` i częsci z argumentów polecenia. Rozpakowywując tarballa użyj flagi `-z`, która oznacza żeby najpierw przepuścić plik przez `gunzip` i rozpkować go (dekompresja). Najpopularniejszym sposobem dekompresji tarballa jest:

```
% tar -xvzf filename.tar.gz
```

To całkiem sporo opcji. Więc co one oznaczają? `-x` oznacza rozpakować. To ważne, nakazuje `tar` dokładnie co ma zrobić z plikiem wejsciowym. W tym wypadku będziemy [we'll be splitting it back up into all the files that it came from]. `-v` oznacza że postęp będzie wyświetlany na standardowym wyjściu. Wyświetli wszystkie pliki które będą rozpakowywane. Jest całkiem normalne opuszczanie tej opcji, jeśli coś jest nudzące. Alternatywnie użyć opcję `-vv` dla listowania jeszcze większej ilości informacji o każdym z plików. Opcja `-z` nakazuje `tar` uruchomienie `filename.tar.gz` najpierw przez `gunzip`. I nareszcie opcja `-f` mówi ona `tar` że następnym wyrażeniem dla komendy będzie plik na którym ma operować.

Jest kilka innych sposobów na zapisanie tych poleceń. Na starszych systemach ze starszą kopią GNU `tar`, możesz znaleźć zapis:

```
% gunzip filename.tar.gz | tar -xvf -
```

Komenda ta dokona dekompresji pliku, a następnie wyśle wyjście do `tar`. Podczas gdy `gzip` zapisze swój wynik na standardowe wyjście o ile mu tak każesz, Strumień wyśle go do `tar` dla rozpakowania (unarchiving). “-” oznacza nakaz operowania na standardowym wejściu. Rozpakuje strumień danych otrzymany z `gzip` i zapisze to na dysku.

Kolejnym sposobem zapisania pierwszej komendy jest pominięcie myślnika przed opcjami:

```
% tar xvzf filename.tar.gz
```

Możesz także natknąć się na archiwa zbzzipowane. Wersja `tar` dostarczona z Slackware Linux poradzi sobie z

tym tak samo jak z archiwum zgzipowanymi. Zamiast opcji `-z` w linii opcji, użyjesz `-j`:

```
% tar -xvjf filename.tar.bz2
```

Ważne jest to aby zaznaczyć że `tar` umieści rozpakowany plik w obecnym katalogu. Więc jeśli masz archiwum w `/tmp` które chcesz zdekompresować do swojego katalogu domowego, jest kilka opcji. Popierwsze archiwum musi być przeniesione do twojego katalogu domowego i następnie uruchomione przez `tar`. Po drugie możesz określić ścieżkę do pliku archiwum w lini poleceń. Po trzecie możesz użyc opcji `-C` do “odpalenia” tarballa w określonym katalogu.

```
% cd $HOME
% cp /tmp/filename.tar.gz .
% tar -xvzf filename.tar.gz

% cd $HOME
% tar -xvzf /tmp/filename.tar.gz

% cd /
% tar -xvzf /tmp/filename.tar.gz -C $HOME
```

Wszystkie powyższe wyrażenia są równoznaczne. W każdym z przypadków archiwum jest rozpakowywane w twoim katalogu domowym i oryginalne nieskompresowane archiwum pozostaje na swoim miejscu.

Co za frajda jest z rozpakowywania tych archiwów jeśli nie możesz ich tworzyć? Prawdę mówiąc `tar` radzi sobie z tym. W większości przypadków wystarczy zastąpić opcje “`-x`” opcją “`-c`”.

```
% tar -cvzf filename.tar.gz .
```

W tym poleceniu opcja `-c` nakauje `tar` utowrzenie archiwum podczas gdy opcja `-z` uruchamia na wynikowym pliku archiwum `gzip` aby je skompresować. `filename.tar.gz` jest plikiem jaki chcesz utworzyć.

Określanie opcji “`-f`” nie jest niezbędne, ale jest dobrym zwyczajem. Bez tego `tar` zapisze na standardowym wyjściu które jest zwykle dla strumieniowania (piping) `tar` do kolejnego programu jak naprzykład:

```
% tar -cv filename.tar . | gpg --encrypt
```

Komenda ta tworzy nieskompresowane archiwum tar bierzącego katalogu, przepuszcza tarballa przez `gpg` który szyfruje i kompresuje tarballa, czyniąc go teoretycznie niemożliwym do odczytania przez kogokolwiek innego niż osobę znającą sekretny klucz.

15.4 `zip`

W końcu, są dwa narzędzia które mogą być użyte na plikach zip. Są one bardzo popularne w świecie Windows, więc Linux posiada programy do radzenia sobie z nimi. Program kompresujący nazywa się `zip`(1), a dekompresujący `unzip`(1).

```
% zip foo *
```

Stworzy to plik `foo.zip`, który będzie zawierać wszytskie pliki z bieżącego katalogu. `zip` doda rozszerzenie `.zip` automatycznie, więc nie musisz dodwać go przy nazwie pliku. Możesz także rekursyjnie przez bierzący katalog, zipujac wszystkie katalogi znajdujace sie w sąsiedztwie:

```
% zip -r foo *
```

Dekompresja plików jest równie prosta.

```
% unzip foo.zip
```

Rozpakuje to wszystkie pliki włącznie z katalogami w pliku `foo.zip`.

Narzędzie `zip` ma kilka zaawansowanych opcji do tworzenia samorozpakowywujących się archiwów, pozostawiania plików kontroli rozmairu kompresowanego pliku, wyświetlania co się dzieje i wiele innych. Zobacz strony manuala dla `zip` i `unzip` aby dowiedzieć się jak używać tych opcji.

Rozdział 16 Vi

`vi`(1) to standardowy Unix'owy edytor tekstów. W dzisiejszych czasach opanowanie go nie jest już tak istotne jak niegdyś. Warto jednak się z nim zapoznać gdyż ma on wielkie możliwości. Istnieje kilka wersji (lub jak kto woli klonów) `vi`. Są nimi: `elvis`, `vile`, oraz `vim`. Przynajmniej jeden z nich jest dostępny na chyba wszystkich rodzajach Unix'a i oczywiście Linux'a. Każda z wymienionych wersji edytora zawiera podstawowy zestaw komend, więc wystarczy zapoznać się z jednym z nich a opanowanie kolejnego nie powinno sprawić problemów. Różnorodność edytorów dostarczanych w dzisiejszych czasach z dystrybucjami Linux'a i Unix'a spowodowała znaczny spadek zainteresowania programem `vi`. Wciąż jednak pozostaje on najbardziej uniwersalnym edytorem na platformach Unix'owych.

`vi` zawiera znaczną liczbę udogodnień włączając w to podświetlanie składni, formatowanie kodu, bardzo rozbudowany mechanizm znajdź-i-zamień, makra i wiele wiele więcej. Wszystko to czyni go szczególnie atrakcyjnym dla programistów, webmasterów itp. Dla administratorów istotna jest możliwość integracji z powłoką.

W Slackware domyślną wersją `vi` jest `elvis`. Inne klony tj. `vim` oraz `gvim` są również dostępne po instalacji odpowiednich pakietów. `gvim` to X'owa wersja `vim`'a z paskiem menu i okienkami dialogowymi.

16.1 Uruchamianie vi

Edytor `vi` może być uruchomiony z lini poleceń na kilka sposobów. Najprostszy z nich to:

```
% vi
```

Rysunek 16-1. Sesja vi

W ten sposób zostanie uruchomiony `vi` z pustym buforem. Edytor w tym momencie jest w trybie wydawania poleceń (“command mode”). Więcej na temat trybów dowiesz się w Sekcja 16.2. Aby opuścić `vi` należy wpisać:

```
:q
```

Zakładając, że nie dokonano żadnych zmian w pliku nastąpi zakończenie pracy i wyjście. Gdyby jednak plik został zmodyfikowany, edytor wyświetli ostrzeżenie i podpowie jak zlekceważyć wprowadzone zmiany. Wykonuje się to zwykle poprzez dodanie wykrzyknika po “**q**”:

```
:q!
```

Wykrzyknik oznacza w większości przypadków wymuszenie akcji. Ta i inne kombinacje klawiszy zostaną omówione szczegółowo w dalszej części rozdziału.

`vi` można uruchomić również z już istniejącym plikiem. Na przykład aby otworzyć `/etc/resolv.conf` należy wywołać:

```
% vi /etc/resolv.conf
```

Edytor może być również otwarty na konkretnej linijce istniejącego pliku. Jest to szczególnie użyteczne dla programistów. W informacji o błędzie często podany jest numer linii, na której program poległ. Na przykład można wystartować `vi` na linii 47 pliku `/usr/src/linux/init/main.c`:

```
% vi +47 /usr/src/linux/init/main.c
```

`vi` wyświetli podany plik i umieści kursor na pożądanej linijce. W przypadku gdy podana zostanie linijka spoza końca pliku edytor umieści kursor na ostatniej istniejącej. Jak wspomniano jest to szczególnie przydatne dla programistów, którzy unikają w ten sposób konieczności przeszukiwania pliku.

16.2 Tryby

`vi` może działać w kilku trybach. W każdym z nich wykonuje się określone zadania. Podczas startu `vi`, użytkownik znajduje się w trybie wydawania poleceń. Można w nim - jak sama nazwa wskazuje - wydawać polecenia służące do manipulacji tekstem, zapisywania, otwierania i poruszania się po pliku jak i zmiany trybu pracy. Edycja tekstu dokonywana jest w trybie wprowadzania (“insert mode”). Zmiany trybu dokonuje się szybko przy pomocy skrótów klawiszowych opisanych poniżej.

16.2.1 Tryb wydawania poleceń (Command Mode)

W tym trybie edytor znajduje się bezpośrednio po uruchomieniu. Nie można w nim bezpośrednio wprowadzać tekstu czy też dokonywać edycji pliku. Tym niemniej można manipulować tekstem, przeszukiwać, ładować i zapisywać pliki. Poniżej przedstawiamy wprowadzenie do trybu. Po pełniejszy spis poleceń zajrzyj do Sekcja 16.7.

Prawdopodonie najczęściej używanym poleceniem w omawianym trybie jest przejście do trybu wprowadzania. Dokonuje się tego poprzez naciśnięcie klawisza **i**. Kursor zmieni kształt i u dołu ekranu pojawi się -- *INSERT* --(w wersji polskiej -- *WPROWADZANIE* --). Należy zauważyć, że nie dzieje się to we wszystkich klonach `vi`. Od tego momentu wszystkie wciśnięte klawisze są zapisywane do aktualnego bufora i wyświetlane na ekranie. Aby powrócić do trybu wydawania poleceń, należy wcisnąć **ESCAPE**.

Tryb wydawania poleceń służy też do poruszania się po pliku. W niektórych systemach można używać strzałek (kursorów), w innych tylko tradycyjnej kombinacji klawiszy “**hjkl**”. Oto ich znaczenie:

h	przesuń w lewo o jeden znak
j	przesuń w dół o jeden znak
k	przesuń w górę o jeden znak
l	przesuń w prawo o jeden znak

Poprostu wciśnij klawisz by poruszyć kursor. Jak się później okaże klawisze te mogą być kombinowane z liczbami w celu polepszenia efektywności.

Wiele z komend używanych w trybie wydawania poleceń rozpoczyna się znakiem dwukropka. Na przykład wyjście to **:q**. Dwukropek informuje, że jest to polecenie podczas gdy "**q**" oznacza zakończenie pracy. Inne polecenia składają się z opcjonalnej liczby i następującej po niej litery. Nie zaczynają się dwukropkiem i są generalnie używane do manipulacji tekstem.

Dla przykładu, by usunąć całą linijkę (tę, na której znajduje się kursor) z pliku należy wpisać **dd**. Wydanie polecenia **4dd** spowoduje natomiast skasowanie wiersza pod kursorem i trzech kolejnych. Generalnie liczba informuje `vi` ile razy wykonać daną akcję.

Liczby można łączyć też z innymi poleceniami, na przykład by poruszać się o kilka znaków w określonym kierunku. Wpisane **10k** spowoduje przeniesienie o 10 wierszy w górę ekranu.

Trybu wydawania poleceń używać można również do kopiowania, wklejania i wczytywania nowego pliku do aktualnego bufora. Aby skopiować tekst należy użyć klawisza **y** (ang. yank). **yy** spowoduje skopiowanie linii a poprzedzenie polecenia liczbą wymusi akcję dla większej ilości wierszy. Komenda **p** odpowiedzialna jest za umieszczenie skopiowanego tekstu w kolejnej linijce (licząc od pozycji kursora).

Wycinanie tekstu dokonywane jest przez sekwencję **dd**, natomiast polecenie **p** może być użyte by wkleić wycięty tekst. Wczytanie tekstu z innego pliku polega na wydaniu komendy **:r** i po spacji nazwy pliku. Jego zawartość zostanie wklejona poczynając od linii poniżej kursora. Niektóre klony `vi` oferują nawet uzupełnianie nazw plików jak to ma miejsce w powłoce.

Tryb wydawania poleceń umożliwia zarówno proste wyszukiwanie jak i skomplikowane komendy typu znajdź-i-zamień korzystające z wyrażeń regularnych. Omówienie tych ostatnich przekracza ramy tej publikacji, dlatego skupimy się w tym miejscu na podstawowych aspektach wyszukiwania.

Trywialne wyszukiwanie realizowane jest poprzez wpisanie (w omawianym trybie) **/** i pożądanego fragmentu tekstu. `vi` rozpocznie poszukiwanie od miejsca, w którym znajduje się kursor aż do końca pliku. Należy zauważyć, że odnalezione zostaną także niedokładne fragmenty. Na przykład wpisanie słowa "*the*" spowoduje zatrzymanie na słowach takich jak "*then*", "*therefore*" itp.

W momencie gdy `vi` znajdzie pierwsze wystąpienie poszukiwanego wyrazu, można nakazać mu poszukać kolejnych. W tym celu należy ponownie wpisać **/** i wcisnąć enter. Aby wyszukiwać wstecz (względem pozycji kursora) znak **/** trzeba zastąpić **?** (i wprowadzić po nim poszukiwane słowo).

16.2.2 Tryb wprowadzania - Insert Mode

W tym trybie następuje wprowadzanie tekstu. Jak już wcześniej wspomniano wejście do omawianego trybu następuje poprzez wciśnięcie klawisza **i** (w trybie Command). Od tego momentu wszystko co zostanie wpisane pojawi się na ekranie i w buforze. Powrót do trybu wydawania poleceń osiąga się poprzez wciśnięcie klawisza **ESCAPE**

Podmienianie (zastępowanie) tekstu może być dokonane na kila sposobów. W trybie Command wpisanie **r** pozwoli na podmianę jednego znaku (znajdującego się pod kursorem). Klawisz **R** działa analogicznie, z różnicą, że nadpisać można dowolną ilość znaków. Ponownie - powrót do trybu wydawania poleceń - odbywa się przez naciśnięcie **ESCAPE**.

Jest jeszcze jedna ciekawa metoda, która pozwala na przełączanie między wprowadzaniem znaków a nadpisywaniem. Aby z niej skorzystać należy wcisnąć **INSERT** w trybie wydawania poleceń. Od tej chwili wspomniany klawisz służy jako przełącznik między wprowadzaniem tekstu a ich nadpisywaniem.

16.3 Otwieranie plików

`vi` pozwala na otwieranie plików w trybie wydawania poleceń jak i bezpośrednio przy uruchamianiu programu - z lini poleceń powłoki. Przykładowo by otworzyć plik `/etc/lilo.conf` w edytorze należy wydać polecenie:

```
:e /etc/lilo.conf
```

Gdy we wcześniej edytowanym pliku zostaną dokonane zmiany i nie zostaną one zapisane `vi` wyświetli ostrzeżenie przed otwarciem kolejnego bufora. Generalnie można ich uniknąć używając wykrzyknika (**:e!**).

Aby ponownie otworzyć ten sam plik należy posłużyć się poleceniem **e!**. Jest to przydatna komenda w wypadku wprowadzenie niechcianych zmian do pliku.

Niektóre klony `vi` (na przykład `vim`) pozwalają na otwieranie kilku buforów jednocześnie. Na przykład, chcąc otworzyć plik `09-vi.sgml` (znajdującego się w katalogu domowym) w momencie edycji innego można wpisać:

```
:split ~/09-vi.sgml
```

Nowy plik pokaże się w górnej połowie ekranu, wcześniej otwarty w dolnej. Jest wiele poleceń służących do manipulacji dzieleniem ekranu. Najlepszym pomysłem jest posłużeniem się manualem konkretnego edytora. Należy zauważyć, że niektóre klony w ogóle nie zezwalają na dzielenie ekranu.

16.4 Zapisywanie plików

Pliki zapisywać można na kilka sposobów. Aby np. zapisać aktualny bufor do pliku `randomness` można posłużyć się poleceniem:

```
:w randomness
```

Gdy plik raz już był zachowany, wprowadzone zmiany można zapisać poprostu przez wpisanie **:w**. Po wykonaniu tej akcji edytor wraca do trybu wydawania poleceń. Aby zapisać plik i zakończyć pracę edytora (bardzo częsta operacja) należy użyć komendy **:wq**.

Zachowywanie zmian w plikach tylko-do-odczytu wymaga użycia znaku wykrzyknika:

```
:w!
```

Są jednak przypadki, gdy pliku nie można zapisać (np. gdy należy on do innego użytkownika). Gdy taki wystąpi `vi` poinformuje o zainstniałym fakcie. Jedyne co pozostaje, to zakończyć pracę i wyedytować go jako `root` lub właściciel.

16.5 Zamykanie vi

Jednym ze sposobów na wyjście z `vi` jest wydanie komendy **:wq** (która spowoduje zapisanie zmian przed zakończeniem). Wyjście bez zapisywania dokonywane jest przez polecenie **:q** lub (częściej) **:q!**. Ostatni sposób przydaje się gdy dokonano zniam w pliku lecz nie chce się ich zachowywać.

Na wypadek nieoczekiwanych błędów zarówno `vim` jak i `elvis` posiadają mechanizmy zabezpieczające. Oba edytory zachowują bufory w tymczasowych plikach. Ich nazwy są podobne do edytowanych plików - najczęściej poprzedzone kropką co czyni je ukrytymi.

Taki tymczasowy plik usuwany jest za każdym razem, gdy edytor kończy działanie w normalnych warunkach. Oznacza to, że w przypadku awarii tymczasowa kopia będzie znajdować się na dysku. Przy ponownej edycji zostanie wyświetlone zapytanie co zrobić w zaistniałym przypadku. Bardzo często większość niezapisanej pracy może zostać odzyskana. `elvis` dodatkowo wyśle mail'a informującego, że kopia zapasowa istnieje.

16.6 Konfiguracja vi

Klony `vi` mogą być konfigurowane na kilka sposobów.

Znaczna ilość komend konfiguracyjnych może zostać wprowadzona podczas pracy (w trybie wydawania poleceń). W zależności od wybranego edytora można uaktywnić cechy pomocne podczas programowania (podświetlanie składni, wcinanie kodu itp.), ustawić makra, zadania automake i wiele więcej.

Prawie wszystkie polecenia mogą być umieszczone w pliku konfiguracyjnym, w katalogu domowym. `elvis` używa `.exrc`, `vim` - `.vimrc`. Większość poleceń wydanych w trybie Command można umieścić w tych plikach.

Omawianie wszystkich opcji konfiguracji i różnic pomiędzy edytorami to obszerny temat. Aby uzyskać więcej informacji warto przeczytać strony manuala lub odwiedzić stronę internetową ulubionego edytora. `vim` posiada również bardzo obszerną pomoc dostępną podczas pracy z edytorem (polecenie **:help**). Ciekawą pozycją jest również książka wydawnictwa O'Reilly - *Learning the `vi` Editor* Lamb'a i Robbins'a.

Znaczna ilość popularnych programów w Linux'ie otwiera pliki tekstowe w `vi` jako domyślnym edytorze. (Na przykład podczas edycji plików crontab.) Aby to zmienić należy ustawić zmienną `VISUAL` tak by wskazywała na preferowany edytor. Informację, jak tego dokonać znajdują się w rozdziale 8 w części Zmienne środowiskowe. Aby mieć pewność, że wybrany edytor zawsze będzie używany jako domyślny warto umieścić odpowiednie linijki w pliku `.bash_profile` lub `.bashrc`.

16.7 Klawiszologia Vi

Ten podrozdział ma pełnić funkcję “szybkiej pomocy” na temat najpopularnieszych poleceń `vi`. Niektóre z poniższych komend zostały już omówione, inne nie.

Tabela 16-1. Poruszanie się

Operacja	Klawisz
lewo, dół, góra, prawo	**h**, **j**, **k**, **l**
Na koniec linijki	**$**
Na początek linijki	^
Na koniec pliku	**G**
Na początek pliku	**:1**
Skok do linijki 69	**:69**

Tabela 16-2. Edycja

Operacja	Klawisz
Usuwanie wiersza	**dd**
Usuwanie pięciu linii	**5dd**
Nadpisywanie znaku	**r**
Usuwanie znaku	**x**
Usuwanie 10 znaków	**10x**
Cofnięcie ostatniej akcji	**u**
Złączenie linijek	**J**
Zamień występienie słowa old na new, w całym pliku	**%s'old'new'g**

Tabela 16-3. Przeszukiwanie

Operacja	Klawisz
Znajdź “asdf”	**/asdf**
Szukaj wstecz ciągu “asdf”	**?asdf**
Powtórz ostatnie wyszukiwanie wprzód	**/**
Powtórz ostatnie wyszukiwanie wstecz	**?**
Powtórz ostatnie wyszukiwanie w tym samym kier.	**n**
Powtórz wyszukiwanie w przeciwnym kierunku	**N**

Tabela 16-4. Zapisywanie i wychodzenie

Operacja	Klawisz
Wyjście	**:q**
Wyjście bez zachowywania zmian	**:q!**
Wyjście z zapisaniem zmian	**:wq**
Zapisanie, bez wyjścia	**:w**
Przeładowanie pliku	**:e!**
Zapisanie pliku jako `asdf`	**:w asdf**
Otwarcie pliku `hejaz`	**:e hejaz**
Wczytanie pliku `asdf`do bufora	**:r asdf**
Wczytanie wyniku działania polecenia `ls` do bufora	**:r !ls**

Rozdział 17 Emacs

Podczas gdy `vi` (wraz ze swoimi klonami) jest bez wątpienia najbardziej wszechobecnym edytorem w systemach uniksopodobnych, Emacs trzyma się dobrze na drugiej pozycji. Zamiast używać różnych “wersji (modułów) [modes]”, jak w `vi` używa kombinacji klawiszy **Control** i **Alt** do wprowadzania poleceń, w większości przypadków podobnie jak możesz używać kombinacji klawiszy **Control** i **Alt** w wordzie i napewno w wielu aplikacjach do uruchamiania określonych funkcji. (Należy podkreślić że polecenia rzadko odpowiadają; więc wiele nowoczesnych plikacji używa **Ctrl**-**C**/ **X**/ **V** dla kopiowania, wycinania i wklejania, Emacs używa innych klawiszy i aktualnie nieco innych mechanizmów dla nich.)

Również niepodobnie do `vi`, który jest doskonałym edytorem i niczym więcej, Emacs jest programem z prawie nieograniczonymi możliwościami. Emacs jest (w większości) napisany w Lips, który jest bardzo potężnym językiem programowania i ma specyficzne własności, jak każdy program napisany w nim jest zarazem automatycznie kompilatorem Lisp. Oznacza to że użytkownik może rozwijać i dosłownie pisać kompletnie nowe programy “w Emacs”.

W rezultacie Emacs nie jest już tylko edytorem. Dostępne jest wiele różnych paczek z dodatkami dla Emacsa (mogą być dołączone do źródeł programu) gwarantuje to cały zakres funkcjonalności. Wiele z nich jest przeznaczonych do edycji tekstu, co jest podstawowym zadaniem Emacsa, ale na tym sie nie kończy. Na przykład jest kilka arkuszy kalkulacyjnych dla Emacsa, bazy danych, gry, klienci maili i newsów (najsłynniejszy to Gnus) itd.

Są dwie główne wersje Emacsa: GNU Emacs (to wersja którą masz razem z Slackwarem) i XEmacs. Litera *NIE* oznacza wersji Emacs działającego pod xami. Wprawdzie zarówno Emacs jak i XEmacs działają pod konsolą i pod Xami. XEmacs powstał jako projekt uparządkowania kodu Emacsa. Obecnie obie wersje są aktywnie rozwijane i jest spora interakcja (współpraca) między oboma zespołami developerskimi. Obecnie dla rozdziału nie ma znaczenia czy używasz Emacs czy XEmacs, różnice pomiędzy nimi nie są istotne dla zwykłego użytkownika.

17.1 Uruchamianie Emacsa

Emacs może zostać uruchomiony z powłoki przez wpisanie `emacs`. Kiedy używasz X-ów Emacs (zazwyczaj) uruchomi się we własnym okienku, zazwyczaj z paskiem menu na górze, gdzie możesz znaleźć najważneijsze funkcje. Na starcie Emacs pokazuje wiadomość powitalną i po odczekaniu paru sekund przerzuci się do *scratch* buffer. (Zobacz Sekcja 17.2.)

Możesz także uruchomić Emacs na istniejącym pliku, pisząc

```
% emacs /etc/resolv.conf
```

Nakaże to Emacsowi załadowanie określonego pliku kiedy będzie się uruchamiać z pominięciem wiadomości powitalnej.

17.1.1 Klawisze poleceń

Jak wspomniano wyżej, Emacs używa kombinacji klawiszy **Control** i **Alt**jako poleceń. Typowa konwencja to zapis **litera**-**C** i **litera**-**M**, odpowiednio. Więc **C**-**x** oznacza **Control**+**x**, i **M**-**x** oznacza **Alt**+**x**. (Litera **M** używana jest zamiast A poznieważ oryginalnie klawisz nie nazywał się **Alt** lecz **Meta**. Klawisz **Meta** znikł ze wszytskich klawiatur komputerowych, a jego miejsce zajął klawisz**Alt**.)

Wiele poleceń Emacsa zawiera sekwencję kombincaji klawiszy. Na przykład, **C**-**x** **C**-**c** (to jest **Control**-**x** dalej **Control**-**c**) opuszczenie Emacsa, **C**-**x** **C**-**s** zapisanie bierzącego pliku. Zapamiętaj że **C**-**x** **C**-**b** *NIE* oznacza to samo co **C**-**x** **b**. Poprzednio oznaczone jako **Control**-**x** natepnie **Control**-**b**, podczas gdy litera oznacza **Control**-**x** nastepnie tylko '**b**'.

17.2 Bufory

W Emacsie podstawą jest koncepcja “buforów”. Każdy plik który otwierasz jest ładowany do własnego bufora. Ponadto Emacs ma kilka specjalnych buforów, które nie zawierają plików tylko służą do innych celów. Takie specjalne bufory mają nazwy zaczynające się i kończące gwiazdką. Na przykład burof który pokazuje się przy pierwszym uruchomieniu Emacsa jest tak nazwany *scratch* buffer. W *scratch* buffer, możesz pisać tekst w normalny sposób, ale tekst nie jest zapisywany kiedy zamykasz Emacsa.

Jest jeszcze jeden specjalny bufor o którym musisz wiedzieć i jest to minibuffer. Bufor ten zawiera tylko jedną linie i jest zawsze widoczny: jest to ostatnia linia okna Emacsa, poniżej paska statusu dla bieżącego bufora. Minibuffer jest miejscem gdzie Emacs pokazuje wiadomości od użytkownika i miejscem gdzie wykonywane są polecenia od użytkownika. Na przykład kiedy otwierasz plik Emacs zapyta o jego nazwe w minibufferze.

Przełanczanie między jednym a drugin buforem może byc wykonane przez polecenie **C**-**x** **b**. Poprosi cie to o nazwe bufora (nazwą bufora jest zazwyczaj nazwa pliku który edytujesz) i zaproponuje domyślny wybór, który jest z reguły buforem utworzonym przed przełączeniem lub utowrzeniem bieżącego burofa. Wciskając poprostu *Enter* przełączysz się właśnie do tego bufora.

Jeśli chcesz przełączyć się do innego bufora niż domyślny oferowany przez Emacsa wpisz poprostu jego nazwę. Zauważ że możesz używać tak zwanego **Tab**-completion (uzupełniania nazwy przez wciśnięcie klawisza tabulacji [tłum.]): napisz pierwsze pare liter nazwy bufora i wciśnij **Tab**; Emacs dokończy nazwe buforu. Uzupełnianie **Tab** działa wszędzie w Emacsie gdzie ma ono sens.

Można uzyskać listę otwartych buforów przez wciśnięcie **C-x C-b**. Polecenie to zazwyczaj podzieli ekran na dwie części, wyświetlając bufor na którym pracujesz na górnej połowie i nowy bufor zwany *Listą Buforów* w dolnej części. Bufor ten zawiera listę wszystkich buforów, ich roamiar i tryb oraz pliki jeśli istnieją, to bufory ich są wyświetlone (jak to zwią w Emacsie). Możesz wyczyścić ten ekran przez wciśnięcie **C-x 1**.


Pod X-ami, lista buforów dostępna jest także w menu bufora w pasku menu.

17.3 Tryby

Każdy bufor w Emacsie ma towarzyszący mu tryb. Tryb ten ma całkiem inną idee niż tryby w `vi`: tryb informuje w jakim rodzaju bufora jesteś. Na przykład jest tryb tekstu dla normalnych plików tekstowych, ale jest również tryb-c dla edycji programów w C, tryb-sh dla edycji skryptów powłoki, tryb-latex dla edycji **LaTeX** pliki, tryb-mail dla edycji emaili i wiadomości, itp. Tryby wyposażone są w specjalnie ulepszenia i funkcje użyteczne dla typów plików jakie są edytowane. Jest nawet możliwe ponowne zdefiniowanie klawiszy i poleceń dla trybów. Na przykład w trybie tekstowym klawisz Tab poprostu przeskakuje do następnego znaku tabulacji i zatrzymuje się, ale w wielu trybach języków programowania klawisz **Tab** robi wcięcie w bieżącej linii odpowiednio do głębokości bluku z liniami.

Tryby wymienione powyżej są nazwywane trybami głównymi. Każdy bufor ma dokładnie jeden główny tryb. Dodatkowo bufor może mieć jeden lub więcej podrzędnych trybów. Podrzędny tryb zapewnia dodatkowe właściwości które moga być użyteczne do określonch zadań. Na przykład jeśli wciśniesz klawisz **INSERT** wywołasz tryb nadpisywania, którego nie chcesz. Jest także tryb auto-uzupełniania który jest kombinacją tryby tekstowego i trybu latex: powoduje on że każda linia która zostanie wpisana będzie automatycznie zawinięta kiedy linia osiągnie określona liczbę znaków. Bez trybu auto-uzupełniania musisz wpisać **M-q** aby uzupełnić paragraf. (Co możesz użyć do formatowania paragrafu jak już wyedytujesz jakiś tekst i nie wygląda on załadnie.)

17.3.1 Otwieranie plików

Aby otowrzyć plik w Emacsie wciśnij

```
C-x C-f
```

Emacs poprosi o nazwę pliku uzupełniając czasem domyślną ścieżkę (którą zazwyczaj jest `~/`). Po wpisaniu nazwy pliku (możesz użyć uzupełnienia **Tab**) i wciśnięciu **ENTER**, Emacs otworzy plik w nowym buforze i wyświetli jego zawartość na ekranie.


Emacs automatycznie utorzy nowy bufor, nie załaduje go do bieżącego buforu.

Chąc utworzyć nowy plik w emacsie nie można poprostu pisać. Najpierw trzeba utworzyć bufor dla niego i nazwę pliku dla niego. Robi się to wciskając **C-x C-f** i wpisując nazwę pliku, tak jak otwieranie istniejącego pliku. Emacs zauwarzy że plik który wpisano nie istnieje i utworzy nowy bufor i raport “(Nowy plik)” w głównym buforze.

Kiedy wpiszesz **C-x C-f** a następnie wprowadzisz nazwe katalogu zamiast nazwy pliku, Emacs utworzy nowy bufor w którym znajdzie się lista wszystkich plików z tego katalogu. Można przesuwać kursor aby znaleźć pożądany plik, a Emacs go otworzy. (W rzeczywistości jest znacznie więcej akcji jakie możesz tam wykonać, jak kasowanie, zmiana nazwy i przenoszenie plików. Emacs teraz jest w trybie dired-mode, który jest podstawowym prostym menadżerem plików.)

Kiedy wpiszesz **C-x C-f** i nagle zmienisz zdanie, możesz wpisać **C-g** aby anulować akcje. **C-g** działa prawie wszędzie gdzie potrzeba anulować rozpoczętą akcje lub polecenie a nie chce się aby doszło ono do końca.

17.4 Podstawowa edycja

Kiedy otworzy się plik, można oczywiście poruszać się po nim kursorami. **Klawisze kursora** i **PgUp**, **PgDn** robią to czego od nich się oczekuje. **Home** i **End** przeskakują na początek i koniec linii. (W satrszych wersjach, przeskoczą na początek lub koniec bufora.) Jednakże, są też komba klawiszy **Control** i **Meta** (**Alt**) które przenoszą kursor dookoła. Ponieważ nie potrzebujesz przenosić rąk na inne części klawiatury aby to zrobić, jest to znacznie szybsze kiedy przywykniesz do tego. Najważniejsze polecenia wypisane są w Tabela 17-1.

Tabela 17-1. Podstawowe polecenia edycji w Emacsie

Polecenie	Rezultat
C-b	cofa o jeden znak
C-f	jeden znak do przodu
C-n	jedna linia w dół
C-p	jedna linia w górę
C-a	idź na początek linii
C-e	idź na koniec linii
M-b	cofnij o jedno słowo
M-f	jedno słowo do przodu
M-}	idź paragraf do przodu
M-{	idź jeden paragraf w tył
M-a	jendo zdanie do przodu
M-e	jedno zdanie w tył
C-d	skasuj znak pod kursorem
M-d	kasuj do końca słowa
C-v	idź w dół o jedną strone (np. PgDn)
M-v	idź w góre o jedną stronę (np. PgUp)
M-<	idź na początek buforu
M->	idź na koniec bufora
C-_	cofnij ostatnią zmianę (może być powtarzane); zauważ że aktualnie trzeba wpisać **Shift**+**Control**+**hyphen** aby to osiągnąć.
C-k	kasuj do końca linii
C-s	szukaj do przodu
C-r	szukaj do tyłu

Zauważ, że wiele poleceń **Meta** są adekwatne do poleceń **Control** c poza tymi które operują na większych jednostkach: gdzie **C-f** idzie do przodu o jeden znak, **M-f** idzie do przodu o całe słowo, itp.

Zauważ że **M-<** i **M->** wymagają wciśnięcia **Shift**+**Alt**+**przecinek** i **Shift**+**Alt**+**dot** odpowiednio, od < i > są **Shift**+**przecinek** i **Shift**+**kropka**. (Oczywiście dopóki nie masz innego układu klawiatury niż standardowy US.)

Zuważ że **C-k** kasuje (zabija jak to sie potocznie mówi) cały tekst za kursorem do końca linii, ale nie kasuje samej linii (np. nie kasuje końca nowej linii). Jedynie kasuje linie jeśli nie ma tekstu w pobliżu kursora. Innymi słowy aby skasować kompletną linie, musisz umieścić kursor na początku linii i wcisnąć dwukrotnie: **C-k** raz aby skasować tekst w linii, a drugi raz aby skasować linie.

17.5 Zapisywanie plików

Aby zapisać plik, wciśnij

```
C-x C-s
```

Emacs nie zapyta o nazwę pliku, bufor zostanie zapisany w pliku z którego został wczytany. Jeśli chcesz zapisać plik w innym pliku, wpisz

```
C-x C-w
```

Kiedy zapisze się plik pierwszy raz w danej sesji, Emacs zapisze starą wersję pliku w pliku zapasowym, który ma tą samą nazwę zakończoną tyldą: więc jeśli edytujesz plik “`cars.txt`”, Emacs utworzy kopie zapasową “`cars.txt~`”.

Ten plik zapasowy jest kopią pliku który został otwarty. Kiedy nie pracujesz, Emacs regularnie wykonuje kopie auto-zapisu wykonywanej pracy, do pliku z znakami hash: `#cars.txt#`. Ta kopia jest kasowana kiedy zapisuje się plik przy pomocy C-x C-s.

Kiedy skończy się edycje pliku, możesz zabić bufor wpisując

```
C-x k
```

Emacs następnie zapyta który bufor ma zabić, który bufor jest obecnie domyslnym i który możesz wybrać wciskając**ENTER**. Jeśli nie zapisałeś jeszcze pliku, Emacs zapyta czy napewno ma zabić bufor.

17.5.1 Wychodzenie z Emacsa

Kiedy już zupełnie skończysz z Emacs, możesz wpisać

```
C-x C-c
```

To wyłączy Emacsa. Jeśli masz niezapisane pliki, Emacs powiadomi o tym i zapyta czy ma je zapisać. Jeśli odpowie się nie, Emacs poprosi o ostatnie potwierdzenie i wyłączy się.

Rozdział 18 Zarządzenie pakietami w Slackware

Pakiet (potocznie zwany paczką - przyp. tłum.) to zbiór powiązanych ze sobą programów gotowych do zainstalowania. Gdy ściągasz z internetu kod źródłowy programu, to przed instalacją musisz go jeszcze skonfigurować i skompilować. W pakietach te dwie rzeczy zostały już za ciebie zrobione. Jedyne co ty musisz zrobic to instalacja takiej paczki. Inną zaletą korzystania z pakietów jest fakt, iż bardzo łatwo je usunąć bądź zaktualizować, gdy tylko sobie zażyczysz. W Slackware znajdziesz wszystkie niezbędne programy do zarządzania paczkami. Możesz dzięki nim w łatwy sposób instalować, usuwać, uaktualniać pakiety oraz tworzyć je i sprawdzać ich poprawność.

Krąży mit (od momentu wypuszczenia przez RedHat menedżera pakietów), że Slackware nie posiada własnego narzędzia do zarządzania pakietami. Nie istnieje chyba drugie równie dalekie od prawdy stwierdzenie. Slackware od zawsze zawierał menedżer pakietów, nawet wtedy gdy nie istniał jeszcze RedHat. Co prawda nie tak rozbudowany bądź wszechobecny jak ten do rpm (bądź też deb), `pkgtool` oraz programy z nim powiązane nie ustępują wyżej wymienionym. Prawda o `pkgtool` nie jest taka, że ten program nie istnieje - on poprostu nie zajmuje się sprawdzaniem zależności.

Większa część społeczności linuksowej jest przekonana, iż menedżer pakietów musi z definicji zawierać sprawdzanie zależności. Cóż, tak naprawdę żadna definicja tego nie wymaga, zaś Slackware najczęściej tego nie robi. Oczywiście nie oznacza to, że pakiety w Slackware nie posiadają powiązań; po prostu menedżer pakietów nie sprawdza ich. Sprawdzenie takich zależności pozostaje zadaniem dla administratora systemu i jest to sposób, który preferujemy.

18.1 Format paczek

Zanim zaczniemy uczyć się narzędzi, warto najpierw zapoznać się z formatem pakietów Slackware. Pakiety te to proste archiwa tar, które zostały skompresowane programem `gzip`. Są one zbudowane w ten sposób, by mogły zostać rozpakowane do katalogu głównego.

Oto fikcyjny program oraz przykładowa paczka z nim:

```
./
usr/
usr/bin/
usr/bin/makehejaz
usr/doc/
usr/doc/makehejaz-1.0/
usr/doc/makehejaz-1.0/COPYING
usr/doc/makehejaz-1.0/README
usr/man/
usr/man/man1
usr/man/man1/makehejaz.1.gz
install/
install/doinst.sh
```

Menedżer paczek rozpakuje taki plik do katalogu głównego by zainstalować go. W bazie danych tego pakietu zostanie utworzony wpis zawierający zawartość paczki tak, by mogła być ona później zaktualizowana bądź usunięta.

Zwróć uwagę na podkatalog `install/`. Jest to podkatalog o specjalnym znaczeniu, który może zawierać skrypt poinstalacyjny o nazwie `doinst.sh`. Jeżeli menedżer paczek znajdzie taki plik, to uruchomi go po zainstalowaniu pakietu.

Inne skrypty także mogą zostać umieszczone w paczce; więcej o nich w podrozdziale Sekcja 18.3.2 poniżej.

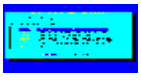
18.2 Narzędzia do obsługi pakietów

Istnieją cztery podstawowe narzędzia do zarządzania paczkami w systemie. Zajmują się one instalacją, usuwaniem oraz aktualizacją pakietów.

18.2.1 pkgtool

`pkgtool`(8) to program obsługiwany za pomocą menu, który pozwala na instalację oraz usuwanie pakietów. Menu główne pokazane jest w Rysunek 18-1.

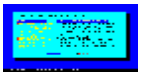
Rysunek 18-1. Pkgtool - menu główne.



Proponowana jest instalacja z katalogu aktualnego bądź innego, albo też z dyskietki. Wybierz metodę instalacji, a pkgtool przeszuka wskazaną lokację w poszukiwaniu paczek do zainstalowania.

Możesz też przejrzeć listę pakietów już zainstalowanych, jak pokazano w Rysunek 18-2.

Rysunek 18-2. Pkgtool - przeglądanie pakietów



Jeżeli chcesz usunąć paczkę, wybierz opcję “remove” (ang. usuń) - otrzymasz listę pakietów zainstalowanych w twoim systemie. Zaznacz te przeznaczone do usunięcia, po czym wybierz OK. `pkgtool` pozbędzie się ich.

Część użytkowników bardziej prferuje to narzędzie niż inne - wywoływane z linii poleceń. Niemniej jednak trzeba zauważyć, że te drugie oferują o wiele więcej opcji. Ponadto też możliwość uaktualniania pakietów jest dostępna jedynie poprzez narzędzia linii poleceń.

18.2.2 installpkg

`installpkg`(8) zajmuje się instalacją nowych paczek w systemie. Składnia jest następująca:

```
# installpkg opcja nazwa_pakietu
```

`installpkg` posiada trzy opcje, z czego można użyć tylko jednej na raz.

Tabela 18-1. `installpkg` Options

Opcja	Działanie opcji
-m	Przeprowadza operację "makepkg" w bieżącym katalogu.
-warn	Pokazuje, co stanie się gdy zainstalujesz wskazaną paczkę. Opcja szczególnie użyteczna na serwerach - dzięki niej dokładnie zobaczysz efekt instalacji jeszcze przed jej wykonaniem.
-r	Zainstaluj rekursywnie wszystkie paczki z katalogu bieżącego i wszystkich katalogów w nim zawartych. W nazwie paczki można użyć znaków uniwersalnych (ang. wildcards), które zostaną wykorzystane jako maska wyszukiwania przy rekursywnej instalacji.

Jeżeli do zmiennej środowiskowej `ROOT` wpiszesz ścieżkę będzie ona użyta przez `installpkg`, jako katalog główny (ang. root directory). Jest to użyteczne przy przygotowywaniu nowych dysków/napędów, na których zostanie umieszczony wspomniany katalog. Typowo będą one zamontowane w `/mnt` lub gdzie indziej wyłączając `/`.

Baza danych o zainstalowanych pakietach przechowywana jest w `/var/log/packages`. Wpisy takie to w rzeczywistości zwykłe pliki tekstowe, po jednym dla każdej paczki. Jeżeli pakiet posiadał skrypt poinstalacyjny, to zapisany on został w `/var/log/scripts/`.

Możesz podać nazwy kilku pakietów, bądź też użyć w nazwie znaków uniwersalnych. Miej świadomość tego, że `installpkg` nie poinformuje w żaden sposób o tym, że właśnie nadpisuje jakąś już zainstalowaną paczkę. Po prostu zainstaluje nową nadpisując pliki starej. Jeżeli chcesz mieć pewność, że stare pliki z poprzedniej paczki zostały bezpiecznie usunięte, użyj `upgradepkg`.

18.2.3 removepkg

`removepkg`(8) odpowiada za usuwanie pakietów z systemu. Oto jego składnia:

```
# removepkg opcja nazwa_pakietu
```

Dla `removepkg` zostały przewidziane cztery opcje; w danej chwili może być użyta tylko jedna z nich.

Tabela 18-2. `removepkg` Opcje

Opcja	Efekt
-copy	Pakiet zostaje skopiowany do specjalnego (ang. preserved) katalogu paczek. Tworzy to drzewo katalogów oryginalnego pakietu bez usuwania go.
-keep	Zachowuje pliki tymczasowe, tworzone podczas usuwania. Tak naprawdę potrzebne jedynie przy debuggowaniu.
-preserve	Pakiet zostaje usunięty, ale jednocześnie kopiowany jest do specjalnego katalogu paczek.
-warn	Pokazuje, co stałoby się, gdyby usunąć pakiet.

Jeżeli do zmiennej środowiskowej `ROOT` wpiszesz ścieżkę będzie ona użyta przez `removepkg`, jako katalog główny (ang. root directory). Jest to użyteczne przy przygotowywaniu nowych dysków/napędów, na których zostanie umieszczony wspomniany katalog. Typowo będą one zamontowane w `/mnt` lub gdzie indziej wyłączając `/`.

`removepkg` przegląda też inne zainstalowane paczki i usuwa pliki używane tylko i wyłącznie przez pakiet wskazany przez ciebie do usunięcia. Nie zapomina też o przeczytaniu zawartości skryptu poinstalacyjnego danej paczki oraz usunięciu wszystkich linków symbolicznych przez niego utworzonych.

Podczas procesu usuwania wyświetlany jest raport o statusie. Po usunięciu informacje o paczce zostają zapisane w `/var/log/removed_packages`, zaś skrypt poinstalacyjny jest przeniesiony do `/var/log/removed_scripts`.

Podobnie jak przy `installpkg`, tak i tu możesz określić kilka pakietów, bądź też użyć znaków uniwersalnych w nazwie pakietu.

18.2.4 upgradepkg

`upgradepkg`(8) uaktualni wskazany pakiet Slackware. Składnia tego polecenia wygląda tak:

```
# upgradepkg nazwa_pakietu
```

albo tak:

```
# upgradepkg nazwa_starego_pakietu%nazwa_nowego_pakietu
```

`upgradepkg` pracuje w ten sposób, że najpierw instaluje nowy pakiet, zaś dopiero potem usuwa stary, tak by nieużywane już pliki nie zaśmiecały systemu. Jeżeli nazwa aktualizowanego pakietu zmieniła się, użyj znaku procentu do określenia starego pakietu (tego, który aktualnie jest już zainstalowany w twoim systemie) oraz nowego (tego, do którego zainstalowany pakiet ma być uaktualniony).

Jeżeli do zmiennej środowiskowej `ROOT` wpiszesz ścieżkę będzie ona użyta przez `upgradepkg`, jako katalog główny (ang. root directory). Jest to użyteczne przy przygotowywaniu nowych dysków/napędów, na których zostanie umieszczony wspomniany katalog. Typowo będą one zamontowane w `/mnt` lub gdzie indziej wyłączając `/`.

Wykonanie `upgradepkg` nie zawsze powiedzie się w pełni. Należy zawsze zachować swoje pliki konfiguracyjne. Jeżeli zostaną one usunięte bądź nadpisane, będzie można użyć ich kopii przy pracach naprawczych.

I tak jak w wypadku `installpkg` oraz `removepkg`, można znowu podać nazwy kilku pakietów oraz używać

znaków uniwersalnych w nazwie pakietu.

18.2.5 `rpm2tgz/rpm2targz`

RedHat Package Manager (Menedżer Pakietów w RedHat Linux) to popularny menedżer pakietów, łatwo dziś dostępnych. Wielu dystrybutorów oprogramowania oferuje swoje produkty w postaci RPM. Nie jest to dla Slackware natywny format, dlatego też nie polecamy polegać na nim. Niemniej jednak pewne rzeczy są dostępne tylko w postaci RPM (także niektóre ze źródeł).

Slackware zawiera program konwertujący pakiety RPM na jego rodzimy format `.tgz`. Pozwala to na rozpakowanie paczki (np. przy użyciu `explodepkg`) do katalogu tymczasowego oraz przetestowanie jej zawartości.

Program `rpm2tgz` stworzy Slackware'ową paczkę o rozszerzeniu `.tgz`, zaś `rpm2targz` utworzy archiwum o rozszerzeniu `.tar.gz`.

18.3 Tworzenie paczek

Stworzenie paczki dla Slackware może być trudne albo proste. Nie ma na to konkretnej metody. Jedynym wymogiem jest, by pakiet ten był plikiem starowanym (przy użyciu programu tar) i spakowanym programem gzip, oraz jeżeli jest tam skrypt poinstalacyjny, to musi się on znaleźć w `/install/doinst.sh`.

Jeżeli jesteś zainteresowany tworzeniem paczek dla swojego systemu albo dla sieci, którą zarządzasz, powinieneś rzucić okiem na różne skrypty w drzewie katalogów źródeł Slackware. Jest tam kilka metod, których używamy do tworzenia paczek.

18.3.1 `explodepkg`

`explodepkg`(8) dokona tego samego, co `installpkg` by rozpakować pakiet, ale nie zainstaluje go ani też nie uczyni żadnych wpisów odnośnie niego w bazie danych zainstalowanych pakietów. Po prostu polecenie to rozpakowuje dany pakiet do bieżącego katalogu.

Jeżeli spojrzysz na katalog ze źródłami Slackware, zobaczysz w jaki sposób używamy tego polecenia dla roboczych pakietów. Paczki takie zawierają szkielet tego, jak wyglądać będzie końcowa wersja pakietu. Zawierają wszystkie wymagane nazwy plików (nierzadko są to tylko puste pliki), uprawnienia oraz właścicieli. Skrypt budujący paczkę przekieruje zawartość paczki z katalogu źródłowego do katalogu budowania paczki.

18.3.2 `makepkg`

`makepkg`(8) stworzy z aktualnego katalogu poprawną paczkę dla Slackware. Przeszuka drzewo katalogów w poszukiwaniu linków symbolicznych oraz doda do skryptu poinstalacyjnego specjalny blok tworzący je podczas instalacji paczki. Dodatkowo otrzymamy ostrzeżenia o plikach o zerowej długości występujących w drzewie paczki.

Polecenie to jest zwykle uruchamiane, gdy stworzyłeś już katalog ze swoim pakietem.

18.3.3 SlackBuild Scripts

Pakiety Slackware są budowane w różny sposób, zależnie od potrzeby. Nie wszystkie pakiety programów są napisane przez ich twórców tak, by kompilowały się w jednakowy sposób. Wiele z nich posiada opcje czasu kompilacji, które nie są zawarte w paczkach, których używa Slackware. Być może zechcesz wykorzystać

niektóre z ich funkcjonalności; w takim wypadku należy skompilować swój własny pakiet. Na szczęście dla wielu paczek w Slackware możesz znaleźć skrypty "SlackBuild" w kodzie źródłowym pakietów.

Tak więc czym jest SlackBuild-skrypt? Skrypt SlackBuild to wykonywalny skrypt powłoki, który uruchamiasz jako `root` by skonfigurować, skompilować oraz stworzyć paczkę dla Slackware. Możesz bez ograniczeń modyfikować te skrypty w katalogu źródłowym oraz uruchamiać je by stworzyć własne wersje standardowych paczek Slackware.

18.4 Tworzymy tagi oraz tagfiles (dla instalatora)

Instalator Slackware zajmuje się instalacją paczek z oprogramowaniem w twoim systemie. Są w nim pliki mówiące instalatorowi, które paczki muszą być zainstalowane, które są opcjonalne oraz które zostały domyślnie wybrane do zainstalowania.

Plik z takimi informacjami znajduje się w pierwszym katalogu danej grupy programów i nazywany jest "tagfile". Zawiera listę pakietów na konkretnym dysku instalacyjnym oraz ich status, tj:

Tabela 18-3. Tagfile - możliwe statusy

Opcja	Znaczenie
ADD	Pakiet jest wymagany do prawidłowego działania systemu
SKP	Pakiet zostanie automatycznie pominięty
REC	Pakiet nie jest wymagany, niemniej zaleca się zainstalowanie go
OPT	Pakiet opcjonalny

Składnia jest prosta:

```
nazwa_pakietów: status
```

Jedna linia - jeden pakiet. Oryginalne tagfiles dla każdej grupy programów są przechowywane jako tagfiles.org. Tak więc, jeżeli za bardzo namieszałeś przy zmianach plików tagfiles, możesz odzyskać ich oryginalną postać.

Wielu administratorów preferuje tworzenie swoich własnych tagfiles oraz uruchamianie instalatora z opcją “full”. Instalator przeczyta wtedy plik tagfiles oraz przeprowadzi instalację wg jego zawartości. Jeżeli wybrana zostanie opcja REC lub OPT, pojawi się okienko dialogowe (podczas instalacji) by zapytać użytkownika czy dany pakiet ma zostać zainstalowany. Zaleca się używanie ADD i SKP do przeprowadzania zautomatyzowanych instalacji.

Nie zapomnij sprawdzić, czy tagfiles twojego autorstwa znajdują się w tym samym miejscu co oryginały. Możesz też ustalić własną ścieżkę dostępu do tagfiles.

Rozdział 19 ZipSlack

19.1 Co to jest ZipSlack?

ZipSlack jest specjalną wersją Slackware Linux. Jest to kopia już zainstalowanego Slacka gotowa do uruchomienia z partycji DOS lub Windows. To podstawowa instalacja, nie zawiera wszystkiego co wchodzi w skład Slackware.

ZipSlack wzięło swoją nazwe formy dystrybucji - dużego pliku ZIP. Użytkownicy DOSa i Windowsa z pewnością znają te pliki. Są to skompresowane archiwa. ZipSlack zawiera wszystko czego potrzeba do

uruchomienia i pracy ze Slackware.

Ważne aby zaznaczyć, że ZipSlack jest znacząco inny od typowej instalacji. Pomimo tej samej funkcjonalności i zawartości tych samych programów ich docelowi odbiorcy i funkcje różnią się. Parę zalet i wad ZipSlacka omówiono poniżej.

Jeszcze jedna uwaga - zawsze sprawdzaj dokumentację dołączoną do ZipSlacka. Zawiera ona najnowsze informacje odnosnie instalacji, botowania i ogólnego użycia produktu.

19.1.1 Zalety

- Nie wymaga partycjonowanie dysku twardego.
- Doskonały sposób na nauke Slackware Linux bez przechodzenia przez proces instalacyjny.

19.1.2 Wady

- Używa DOSowego formatu plików, który jest wolniejszy niż natywny system Linuksa.
- Nie będzie działać z Windows NT.

19.2 Pozyskiwanie ZipSlacka

Pozyskanie ZipSlacka jest proste. W przypadku zakupienia oryginalnego zestawu Slackware Linux CD, będzie on do niego dołączony. Odszukaj płytkę zawierającą katalog `zipslack` i umieść go w swoim napędzie CD-ROM. To zazwyczaj trzeci lub czwarty dysk, ale zawsze warto sprawdzić opis na nośniku.

Jesli chcesz pobrać ZipSlack, musisz najpierw odwiedzić naszą stronę “Get Slack” w poszukiwanu najświeższych informacji:

http://www.slackware.com/getslack/

ZipSlack jest częścią każdej dystrybucji Slackware. Zlokalizuj wydanie, które cię interesuje i prejdź do katalogu na stronie FTP. Najświeższą wersję możesz znaleźć pod adresem:

ftp://ftp.slackware.com/pub/slackware/slackware/

ZipSlacka znajdziesz w podkatalogu `/zipslack`. ZipSlack oferowany jest jako jeden wielki plik `.ZIP` lub cześci o rozmiarach dyskietki. Części znajdują się w katalogu `/zipslack/split`.

Nie ograniczaj się tylko do pliku `.ZIP`. Warto także pobrac dokumentacje i wszystkie obrazy boot jakie znajdują się w katalogu.

19.2.1 Instalacja

Po ściągnięciu wymaganych plików należy rozpakować archiwum `.ZIP`. Upewnij się, że używasz 32-bitowego unzippera. Rozmiary i nazwy plików w archiwum są za duże dla 16-bitowego unzippera.

ZipSlack jest zaprojektowany do rozpakowania bezpośrednio na dysk w głównym katalogu (jak naprzyklad `C:` lub `D:`). Katalog `\LINUX` zostanie utworzony i bedzie zawierał aktualną instalację. W tym katalogu znajdziesz również pliki niezbędne do bootowania systemu.

Po rozpakowaniu plików powinien zostać utworzony katalog `\LINUX` na dysku, który został wybrany (my użyjemy `C:`).

19.3 Botowanie ZipSlacka

Jest kilka sposobów na bootowanie ZipSlacka. Najpopularniejszym jest użycie dołączonego LINUX.BAT w celu startu systemu z poziomu DOS (lub z trybu DOS pod Windows 9x). Plik ten musi zostać zmodyfikowany przed odpaleniem.

Rozpocznij od otwarcia `C:\LINUX\LINUX.BAT` w twoim ulubionym edytorze. Na początku pliku znajdziesz duży komentarz. Wyjaśnia on co należy wyedytować w pliku (oraz co zrobic jeśli bootujesz z zewnetrznego napędu ZIP). Nie przejmuj się jeśli nie zrozumiesz zmiennej `root=`. Jest kilka przykładów więc nie krępuj sie wybrać jeden i spróbować. Jeśli nie zadziała, możesz wyedytowac plik znowu komentując linie i wybierając inną opcję.

Po odkomentowaniu pożądanych linii (przez skasowanie “rem” na jej początku), zapisz plik i wyjdz z edytora. Przejdź do DOSu.

Tryb MS-DOS w Windows 9x NIE jest dobrym pomysłem.

Wpisz `C:\LINUX\LINUX.BAT` aby wystartować system. Jeśli wszystko pójdzie dobrze, powinien pojawić się monit powitalny.

Zaloguj się jako `root`, bez hasła. Warto je ustawić i dodać konto dla siebie. W tym miejscu możesz odwoływać sie do innych sekcji tej książki.

Jeśli plik `LINUX.BAT` nie zadziała należy odwołać się do dołączonego pliku `C:\LINUX\README.1ST` w celu przejrzenia opcji bootowania.

Słownik

Account

> All of the information about a user, including username, password, finger information, UID and GID, and home directory. To create an account is to add and define a user.

Background

> Any process that is running without accepting or controlling the input of a terminal is said to be running in the background.

Boot disk

> A floppy disk containing an operating system (in our case, the Linux kernel) from which a computer can be started.

Compile

> To convert source code to machine-readable “binary” code.

Daemon

> A program designed to run in the background and, without user intervention, perform a specific task (usually providing a service).

Darkstar

The default hostname in Slackware; your computer will be called darkstar if you do not specify some other name.

One of Patrick Volkerding's development machines, named after “Dark Star”, a song by the Grateful Dead.

Desktop Environment

A graphical user interface (GUI) that runs atop the X Window System and provides such features as integrated applications, cohesive look-and-feel between programs and components, file and window management capabilities, etc. A step beyond the simple window manager.

Device driver

A chunk of code in the kernel that directly controls a piece of hardware.

Device node

A special type of file in the `/dev` filesystem that represents a hardware component to the operating system.

DNS

Domain Name Service. A system in which networked computers are given names which translate to numerical addresses.

Domain name

A computer's DNS name, excluding its host name.

Dot file

In Linux, files which are to be hidden have filenames beginning with a dot ('.').

Dotted quad

The format of IP addresses, so called because it consists of four numbers (range 0-255 decimal) separated by periods.

Dynamic loader

When programs are compiled under Linux, they usually use pieces of code (functions) from external libraries. When such programs are run, those libraries must be found and the required functions loaded into memory. This is the job of the dynamic loader.

Environment variable

A variable set in the user's shell which can be referenced by that user or programs run by that user within that shell. Environment variables are generally used to store preferences and default parameters.

Epoch

A period of history; in Unix, “The Epoch” begins at 00:00:00 UTC January 1, 1970. This is considered the “dawn of time” by Unix and Unix-like operating systems, and all other time is calculated relative to this date.

Filesystem

A representation of stored data in which “files” of data are kept organized in “directories”. The filesystem is the nearly universal form of representation for data stored to disks (both fixed and removable).

Foreground

A program that is accepting or controlling a terminal's input is said to be running in the foreground.

Framebuffer

A type of graphics device; in Linux, this most often refers to the software framebuffer, which provides a standard framebuffer interface to programs while keeping specific hardware drivers hidden from them. This layer of abstraction frees programs of the need to speak to various hardware drivers.

FTP

The File Transfer Protocol. FTP is a very popular method of transferring data between computers.

Gateway

A computer through which data on a network is transferred to another network.

GID

Group Identifier. The GID is a unique number attributed to a group of users.

Group

Users in Unix belong to “groups”, which can contain many other users and are used for more general access control than the existence of users alone can easily allow.

GUI

Graphical User Interface. A software interface that uses rendered graphical elements such as buttons, scrollbars, windows, etc. rather than solely text-based input and output

Home directory

A user's “home directory” is the directory the user is placed in immediately upon logging in. Users have full permissions and more or less free reign within their home directories.

HOWTO

A document describing “how to” do something, such as conRysunek a firewall or manage users and groups. There is a large collection of these documents available from the Linux Documentation Project.

HTTP

The Hypertext Transfer Protocol. HTTP is the primary protocol on which the World Wide Web operates.

ICMP

Internet Control Message Protocol. A very basic networking protocol, used mostly for pings.

Kernel

The heart of an operating system. The kernel is the part that provides basic process control and interfaces with the computer's hardware.

Kernel module

A piece of kernel code, usually a driver of some sort, that can be loaded and unloaded from memory separately from the main body of the kernel. Modules are handy when upgrading drivers or testing kernel settings, because they can be loaded and unloaded without rebooting.

Library

A collection of functions which can be shared between programs.

LILO

The LInux LOader. LILO is the most widely-used Linux boot manager.

LOADLIN

LOADLIN is a program that runs under MS DOS or Windows and boots a Linux system. It is most commonly used on computers with multiple operating systems (including Linux and DOS/Windows, of course).

Man Sekcja

Pages in the standard Unix online manual ("man") are grouped into Sekcjas for easy reference. All C programming pages are in Sekcja 3, system administration pages in Sekcja 5, etc.

MBR

The Master Boot Record. A reserved space on a hard drive where information on what to do when booting is stored. LILO or other boot managers can be written here.

Motif

A popular programming toolkit used in many older X programs.

MOTD

Message of the Day. The motd (stored in Linux in `/etc/motd` is a text file that is displayed to all users upon logging in. Traditionally, it is used by the system administrator as a sort of “bulletin board” for communicating with users.

Mount point

An empty directory in a filesystem where another filesystem is to be “mounted”, or grafted on.

Nameserver

A DNS information server. Nameservers translate DNS names to numerical IP addresses.

Network interface

A virtual representation of a network device provided by the kernel. Network interfaces allow users and programs to talk to network devices.

NFS

The Network Filesystem. NFS allows the mounting of remote filesystems as if they were local to your computer and thus provides a transparent method of file sharing.

Octal

Base-8 number system, with digits 0-7.

Pager

An X program that allows the user to see and switch between multiple “desktops”.

Partition

A division of a hard drive. Filesystems exist on top of partitions.

PPP

Point-to-Point Protocol. PPP is used mainly for connecting via modem to an Internet Service Provider.

Process

A running program.

Root directory

Represented as “/”, the root directory exists at the top of the filesystem, with all other directories branching out beneath it in a “file tree”.

Root disk

The disk (usually fixed) on which the root directory is stored.

Routing table

The set of information the kernel uses in “routing” network data around. It contains such tidbits as where your default gateway is, which network interface is connected to which network, etc.

Runlevel

The overall system state as defined by init. Runlevel 6 is rebooting, runlevel 1 is “single user mode”, runlevel 4 is an X login, etc. There are 6 available runlevels on a Slackware system.

Secure shell

An encrypted (thus secure) method of logging in remotely to a computer. Many secure shell programs are available; both a client and server are needed.

Service

The sharing of information and/or data between programs and computers from a single “server” to multiple “clients”. HTTP, FTP, NFS, etc. are services.

Shadow password suite

The shadow password suite allows encrypted passwords to be hidden from users, while the rest of the information in the `/etc/passwd` file remains visible to all. This helps prevent brute-force attempts at cracking passwords.

Shell

Shells provide a commandline interface to the user. When you're looking at a text prompt, you're in a shell.

Shell builtin

A command built into the shell, as opposed to being provided by an external program. For instance, `bash` has a `cd` builtin.

Signal

Unix programs can communicate between each other using simple “signals”, which are enumerated and usually have specific meanings. `kill -l` will list the available signals.

SLIP

Serial Line Interface Protocol. SLIP is a similar protocol to PPP, in that it's used for connecting two machines via a serial interface.

Software package

A program and its associated files, archived and compressed into a single file along with any necessary scripts or information to aid in managing the installation, upgrade, and removal of those files.

Software series

A collection of related software packages in Slackware. All KDE packages are in the “kde” series, networking packages in the “n” series, etc.

Source code

The (more or less) human-readable code in which most programs are written. Source code is compiled into “binary” code.

Standard Error (stderr)

The Unix-standard output stream for errors. Programs write any error messages on stderr, so that they can be separated from normal output.

Standard Input (stdin)

The Unix-standard input stream. Data can be redirected or piped into a program's stdin from any source.

Standard Output (stdout)

The Unix-standard output stream. Normal text output from a program is written to stdout, which is separate from the error messages reported on stderr and can be piped or redirected into other programs' stdin or to a file.

Subnet

An IP address range that is part of a larger range. For instance, 192.168.1.0 is a subnet of 192.168.0.0 (where 0 is a mask meaning “undefined”); it is, in fact, the “.1” subnet.

Superblock

In Linux, partitions are discussed in terms of blocks. A block is 512 bytes. The superblock is the first 512 bytes of a partition.

Supplemental disk

In Slackware, a floppy disk used during installation that contains neither the kernel (which is on the boot disk) nor the root filesystem (which is on the root disk), but additional needed files such as network modules or PCMCIA support.

Suspended process

A process which has been frozen until killed or resumed.

Swap space

Disk space used by the kernel as “virtual” RAM. It is slower than RAM, but because disk space is cheaper, swap is usually more plentiful. Swap space is useful to the kernel for holding lesser-used data and as a fallback when physical RAM is exhausted.

Symbolic link

A special file that simply points to the location of another file. Symbolic links are used to avoid data duplication when a file is needed in multiple locations.

Tagfile

A file used by the Slackware `setup` program during installation, which describes a set of packages to be installed.

Terminal

A human-computer interface consisting of at least a screen (or virtual screen) and some method of input (almost always at least a keyboard).

Toolkit, GUI

A GUI toolkit is a collection of libraries that provide a programmer with code to draw “widgets” such as

scrollbars, checkboxes, etc. and construct a graphical interface. The GUI toolkit used by a program often defines its “look and feel”.

UID

User Identifier. A unique number that identifies a user to the system. UIDs are used by most programs instead of usernames because a number is easier to deal with; usernames are generally only used when the user has to see things happen.

VESA

Video Electronics Standards Association. The term “VESA” is often used to denote a standard specified by said Association. Nearly all modern video adapters are VESA-compliant.

Virtual terminal

The use of software to simulate multiple terminals while using only a single set of input/output devices (keyboard, monitor, mouse). Special keystrokes switch between virtual terminals at a single physical terminal.

Window manager

An X program whose purpose is to provide a graphical interface beyond the simple rectangle-drawing of the X Window System. Window managers generally provide titlebars, menus for running programs, etc.

Working directory

The directory in which a program considers itself to be while running.

Wrapper program

A program whose sole purpose is to run other programs, but change their behavior in some way by altering their environments or filtering their input.

X server

The program in the X Window System which interfaces with graphics hardware and handles the actual running of X programs.

X Window System

Network-oriented graphical interface system used on most Unix-like operating systems, including Linux.

Dodatek A. The GNU General Public License

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

A.1. GPL a sprawa polska ;-)

Ponieważ FSF (Free Software Foundation) nie opublikowała polskiej wersji językowej licencji GNU GPL my również nie będziemy jej publikować.

Dla osób które nie czują się na siłach aby przyswoić sobie warunki licencji w angielskiej wersji językowej zamieszaczmy link do polskiego **nieoficjalnego** tłumaczenia: Powszechna Licencja Publiczna GNU http://www.gnu.org.pl/text/licencja-gnu.html

---

A.2. Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

A.3. TERMS AND CONDITIONS

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

1. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it,

either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

2. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

3. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Sekcja 1 above, provided that you also meet all of these conditions:

   a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

   These requirements apply to the modified work as a whole. If identifiable Sekcjas of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those Sekcjas when you distribute them as separate works. But when you distribute the same Sekcjas as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

   Thus, it is not the intent of this Sekcja to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

   In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

4. You may copy and distribute the Program (or a work based on it, under Sekcja 2) in object code or executable form under the terms of Sekcjas 1 and 2 above provided that you also do one of the following:

   a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sekcjas 1 and 2 above on a medium customarily used for software interchange; or,

b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sekcjas 1 and 2 above on a medium customarily used for software interchange; or,

c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with SubSekcja b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

5. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

6. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

7. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

8. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this Sekcja is held invalid or unenforceable under any particular circumstance, the balance of the Sekcja is intended to apply and the Sekcja as a whole is intended to apply in other circumstances.

   It is not the purpose of this Sekcja to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this Sekcja has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This Sekcja is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

9. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

10. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

    Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

11. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

12. NO WARRANTY

    BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

13. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

A.4. How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and

a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
   Copyright (C) <year>  <name of author>

   This program is free software; you can redistribute it and/or modify
   it under the terms of the GNU General Public License as published by
   the Free Software Foundation; either version 2 of the License, or
   (at your option) any later version.

   This program is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   GNU General Public License for more details.

   You should have received a copy of the GNU General Public License
   along with this program; if not, write to the Free Software
   Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA   02111-1307 USA
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year  name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

```
  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.
