

# Podstawy Linuksa Slackware

# slackware®

# linux

Alan Hicks  
Chris Lumens  
David Cantrell  
Logan Johnson

[Copyright](#) © 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005 Slackware Linux, Inc.

1-57176-338-4

## Spis treści

Podstawy Linuksa Slackware.....	1
Przedmowa.....	6
Odbiorcy.....	6
Zmiany od czasu pierwszego wydania.....	7
Organizacja książki.....	7
Konwencje użyte w książce.....	9
Konwencje typograficzne.....	9
Tekst wprowadzany przez użytkownika.....	9
Przykłady.....	9
Podziękowania.....	10
Rozdział 1. Wprowadzenie do Linuksa Slackware.....	10
1.1 Co to jest Linux?.....	10
1.1.1 Słowo na temat GNU.....	10
1.2 Co to jest Slackware?.....	11
1.3 Open Source i Free Software.....	11
Rozdział 2. Pomoc.....	12
2.1 Pomoc dostępna w systemie.....	13
2.1.1. man.....	13
2.1.2 Katalog /usr/doc.....	14
2.1.3 Pliki typu HOWTO i mini-HOWTO.....	14
2.2 Pomoc dostępna w Sieci.....	15
2.2.1 Oficjalna strona internetowa i fora pomocy.....	15
2.2.2 Pomoc dostępna przez e-mail.....	15
2.2.2.1 Listy mailingowe projektu Linux Slackware.....	15
2.2.3 Nieoficjalne strony i fora internetowe.....	16
2.2.3.1 Strony.....	16
2.2.3.2 Internetowe fora online.....	17
2.2.3.3 Grupy usenetowe (NNTP).....	17
Rozdział 3. Instalacja.....	17
3.1 Skąd wziąć Slackware.....	18
3.1.1 Oficjalne komplety CD i zestawy pudełkowe.....	18

3.1.2 Przez Internet.....	18
3.2 Wymagania systemowe.....	19
3.2.1 Grupy pakietów.....	19
3.2.2 Metody instalacji.....	20
3.2.2.1 Z dyskietek.....	20
3.2.2.2 Z CD-ROMu.....	21
3.2.2.3 NFS.....	21
3.2.3 Dyskietka rozruchowa.....	21
3.2.4 Dyskietki root.....	21
3.2.5 Dyskietka dodatkowa.....	22
3.2.6 Tworzenie dyskietek rozruchowych.....	22
3.3 Partycjonowanie.....	22
3.4 Program setup.....	24
3.4.1 Pomoc (HELP).....	26
3.4.2 Mapa klawiatury (KEYMAP).....	27
3.4.3 Dodawanie partycji wymiany (ADDSWAP).....	28
3.4.4 Partycje docelowe (TARGET).....	28
3.4.5 Nośnik źródłowy (SOURCE).....	29
Wybór pakietów (SELECT).....	31
3.4.7 Właściwa instalacja (INSTALL).....	32
3.4.8 Konfiguracja (CONFIGURE).....	33
3.4.8.1 Wybór jądra (INSTALL LINUX KERNEL).....	33
3.4.8.2 Tworzenie dyskietki rozruchowej (MAKE BOOT DISK).....	34
3.4.8.3 Modem (MODEM CONFIGURATION).....	35
3.4.8.4 Strefa czasowa (TIMEZONE CONFIGURATION).....	36
3.4.8.5 Mysz (MOUSE CONFIGURATION).....	37
3.4.8.6 Zegar sprzętowy (HARDWARE CLOCK SET TO UTC?).....	38
3.4.8.7 Czcionka ekranowa (SELECT A SCREEN FONT).....	39
LILO (INSTALL LILO).....	40
3.4.8.9 Konfiguracja sieci (Network).....	41
3.4.8.10 Menedżer okien (SELECT DEFAULT WINDOW MANAGER FOR X).....	42
Rozdział 4. Konfiguracja systemu.....	43
4.1 Przegląd systemu.....	43
4.1.1 Układ systemu plików.....	43
4.1.2 Znajdowanie plików.....	45
4.1.2.1 which.....	45
4.1.2.2 whereis.....	45
4.1.2.3 find.....	46
4.1.2.4 slocate.....	46
4.1.3 Katalog /etc/rc.d.....	46
4.1.3.1 Skrypty uruchomieniowe.....	46
4.1.3.2 Skrypty inicjalizacyjne poziomów pracy.....	47
4.1.3.3 Inicjalizacja sieci.....	48
4.1.3.4 Zgodność z System V.....	48
4.1.3.5 Inne pliki.....	48
4.2 Wybór jądra.....	49
4.2.1 Katalog /kernels na CD-ROMie Slackware.....	49
4.2.2 Kompilacja jądra ze źródeł.....	50
4.2.2.1 Kompilacja jądra Linuksa w wersji 2.4.x.....	50
4.2.2.2 Kompilacja jądra Linuksa w wersji 2.6.x.....	52
4.2.3 Moduły jądra.....	52

Rozdział 5. Konfiguracja sieci.....	53
5.1 Wprowadzenie: netconfig Twoim przyjacielem.....	53
5.2 Konfiguracja sprzętu sieciowego.....	54
5.2.1 Ładowanie modułów obsługi sieci.....	54
5.2.2 Karty LAN (10/100/1000Base-T i Base-2).....	54
5.2.3 Modemy.....	54
5.2.4 PCMCIA.....	55
5.3 Konfiguracja TCP/IP.....	55
5.3.1 DHCP.....	56
5.3.2 Statyczny adres IP.....	57
5.3.3 /etc/rc.d/rc.inet1.conf.....	57
5.3.4 /etc/resolv.conf.....	58
5.3.5 /etc/hosts.....	58
5.4 PPP.....	59
5.4.1 pppsetup.....	59
5.4.2 /etc/ppp.....	59
5.5 Konfiguracja sieci bezprzewodowej.....	59
5.5.1 Sprzętowa obsługa karty bezprzewodowej.....	60
5.5.2 Konfiguracja ustawień bezprzewodowych.....	60
5.5.3 Konfiguracja sieci.....	61
5.6 Sieciowe systemy plików.....	61
5.6.1 SMB/Samba/CIFS.....	62
5.6.2 Sieciowy system plików (Network File System – NFS).....	63
Rozdział 6. Konfiguracja X.....	64
6.1 xorgconfig.....	65
6.2 xorgsetup.....	69
6.3 xinitrc.....	69
6.4 xwmconfig.....	70
6.5 xdm.....	72
Rozdział 7. Uruchamianie.....	74
7.1 LILO.....	74
7.2 LOADLIN.....	77
7.3 Uruchamianie wielu systemów.....	78
7.3.1 Windows.....	78
7.3.1.1 Wykorzystywanie LILO.....	78
7.3.1.2 LOADLIN.....	79
7.3.1.3 Stary patent z Windows NT.....	81
7.3.2 Linux.....	81
Rozdział 8. Powłoka.....	82
8.1 Użytkownicy.....	82
8.1.1 Logowanie.....	82
8.1.2 Root czyli Superużytkownik.....	82
8.2 Wiersz poleceń.....	83
8.2.1 Uruchamianie programów.....	83
8.2.2 Dopasowywanie symboli wieloznacznych.....	83
8.2.3 Przekierowywanie wejścia/wyjścia i potokowanie.....	84
8.3 Bourne Again Shell (bash).....	85
8.3.1 Zmienne środowiskowe.....	85
8.3.2 Uzupełnianie tabulatorem.....	86
8.4 Terminale wirtualne.....	87
8.4.1 Program screen.....	87

Rozdział 9. Struktura systemu plików.....	88
9.2 Prawa dostępu.....	89
9.3 Dowiązania.....	91
9.4 Montowanie urządzeń.....	92
9.4.1 Plik fstab.....	92
9.4.2 Polecenia mount i umount.....	93
9.5 Montowanie udziałów NFS.....	93
Rozdział 10. Praca z plikami i katalogami.....	94
10.1 Nawigacja: ls, cd i pwd.....	94
10.1.1 ls.....	94
10.1.2 cd.....	95
10.1.3 pwd.....	96
10.2 Programy stronicujące: more, less i most.....	96
10.2.1 more.....	96
10.2.2 less.....	96
10.2.3 most.....	97
10.3 Proste wyjście: cat i echo.....	97
10.3.1 cat.....	97
10.3.2 echo.....	97
10.4 Tworzenie: touch i mkdir.....	98
10.4.1 touch.....	98
10.4.2 mkdir.....	98
10.5 Kopiowanie i przenoszenie.....	98
10.5.1 cp.....	98
10.5.2 mv.....	99
10.6 Usuwanie: rm i rmdir.....	99
10.6.1 rm.....	99
10.6.2 rmdir.....	100
10.7 Tworzenie nazw zastępczych plików przez ln.....	100
Rozdział 11. Kontrola procesów.....	101
11.1 Praca w tle.....	101
11.2 Praca na pierwszym planie.....	102
11.3 ps.....	102
11.4 kill.....	105
11.5 top.....	106
Rozdział 12. Podstawowa administracja systemem.....	108
12.1 Użytkownicy i grupy.....	108
12.1.1 Skrypty dołączone do dystrybucji.....	108
12.1.2 Zmiana hasła.....	111
12.1.3 Zmiana informacji o użytkowniku.....	112
12.2 Użytkownicy i grupy dla hardcore'owców.....	112
12.3 Poprawne zamykanie systemu.....	114
Rozdział 13. Podstawowe polecenia sieciowe.....	116
13.1 ping.....	116
13.2 traceroute.....	117
13.3 Narzędzia DNS.....	117
13.1.1 host.....	117
13.3.2 nslookup.....	118
13.3.3 dig.....	118
13.4 finger.....	119
13.5 telnet.....	119

13.5.1	Inne sposoby wykorzystania telnetu.....	120
13.6	Bezpieczna powłoka – The Secure Shell.....	121
13.7	Poczta elektroniczna.....	121
13.7.1	pine.....	121
13.7.2	elm.....	122
13.7.3	mutt.....	123
13.7.4	nail.....	124
13.8	Przeglądarki.....	125
13.8.1	lynx.....	125
13.8.2	links.....	126
13.8.3	wget.....	126
13.9	Klienci FTP.....	127
13.9.1	ftp.....	127
13.9.2	ncftp.....	129
13.10	Rozmowa.....	129
13.10.1	wall.....	129
13.10.2	talk.....	130
13.10.3	ytalk.....	130
Rozdział 14.	Bezpieczeństwo.....	131
14.1	Wyłączanie usług.....	131
14.1.1	Usługi uruchamiane przez inetd.....	132
14.1.2	Usługi uruchamiane ze skryptów inicjalizujących.....	132
14.2	Kontrola dostępu do komputera.....	133
14.2.1	iptables.....	133
14.2.2	tcpwrappers.....	134
14.3	Aktualizowanie systemu.....	135
14.3.1	lista mailingowa slackware-security.....	135
14.3.2	Katalog /patches.....	135
Rozdział 15 –	archiwizacja plików.....	136
15.1	gzip.....	136
15.2	bzip2.....	137
15.3	tar.....	137
15.4	zip.....	139
Rozdział 16.	Edytor vi.....	139
16.1	Uruchamianie vi.....	140
16.2	Tryby.....	141
16.2.1	Tryb poleceń.....	141
16.2.2	Tryb wstawiania.....	143
16.3	Otwieranie plików.....	143
16.4	Zapisywanie plików.....	143
16.5	Wychodzenie z vi.....	144
16.6	Konfiguracja vi.....	144
16.7	Klawisze vi.....	145
Rozdział 17.	Emacs.....	146
17.1	Uruchamianie Emacsa.....	147
17.1.1	Klawisze poleceń.....	147
17.2	emacs – bufory.....	148
17.3	Emacs – tryby.....	148
17.3.1	Otwieranie plików.....	149
17.4	Podstawy edycji.....	149
17.5	Zapisywanie plików.....	150

17.5.1 Wychodzenie z Emacsa.....	151
Rozdział 18. Zarządzanie pakietami w Slackware.....	151
18.1 Omówienie formatu pakietów.....	152
18.2 Programy do zarządzania pakietami.....	152
18.2.1 pkgtool.....	152
18.2.2 installpkg.....	154
18.2.3 removepkg.....	155
18.2.4 upgradepkg.....	156
18.2.5 rpm2tgz/rpm2targz.....	156
18.3 Tworzenie pakietów.....	156
18.3.1 explodepkg.....	157
18.3.2 makepkg.....	157
18.3.3 Skrypty SlackBuild.....	157
18.4 Tworzenie znaczników i plików znaczników (dla programu setup).....	157
Rozdział 19. ZipSlack.....	158
19.1 Co to jest ZipSlack?.....	158
19.1.1 Zalety.....	158
19.1.2 Wady.....	159
19.2 Skąd wziąć ZipSlack.....	159
19.2.1 Instalacja.....	159
19.3 Uruchamianie ZipSlack.....	159
Słowniczek.....	160
Dodatek A. Powszechna Licencja Publiczna GNU.....	167
Tłumaczenie GNU General Public License .....	167
Uwaga!.....	167
Note!.....	167
Powszechna Licencja Publiczna GNU.....	167
Wersja 2, czerwiec 1991.....	167
Preambuła.....	168
Zasady i warunki kopiowania, rozpowszechniania i modyfikowania .....	168
KONIEC ZASAD I WARUNKÓW.....	173
Jak stosować niniejsze Warunki do Twoich nowych programów.....	173

## Przedmowa

## Odbiorcy

System operacyjny Linux Slackware jest rozbudowaną, wszechstronną platformą dla komputerów zgodnych z architekturą Intel. Zaprojektowano go tak, aby był stabilny, bezpieczny i funkcjonalny, zarówno w zastosowaniach serwerowych klasy high-end, jak i w wydajnych stacjach roboczych.

Książka ma pomóc Ci rozpocząć pracę z tym systemem. Nie opisuje ona jednak każdego aspektu dystrybucji Slackware. Celem książki jest pokazanie, jakie możliwości ma system oraz przekazanie podstawowej wiedzy pozwalającej na pracę z nim.

Mamy nadzieję, że w miarę zdobywania doświadczenia w pracy z Linuksem Slackware ta książka będzie dla Ciebie przydatną pomocą. Mamy też nadzieję, że pożyczysz ją wszystkim swoim znajomym, kiedy przyjdą do Ciebie i zapytają o „tego zarąbistego Slacka, którego masz na kompie”.

Nie będziesz czytał tej książki od deski do deski z zapartym tchem. Próbowaliśmy jednak napisać ją w możliwie przystępny i ciekawy sposób. Może ktoś kiedyś będzie chciał zrobić film na jej podstawie. Oczywiście mamy też nadzieję, że wiele się z niej nauczysz i okaże się dla Ciebie przydatna.

No dobra, jedziemy z tym koksem.

## Zmiany od czasu pierwszego wydania

Drugie wydanie to kulminacja lat ciężkiej pracy, którą poświęcili oddani członkowie Projektu Dokumentacji Slackware. Oto najważniejsze zmiany wprowadzone w tym wydaniu:

- Rozdział 3, Instalacja, został zmodyfikowany, dodano nowe zrzuty ekranu programu instalacyjnego, uwzględniono zmiany w zestawach dysków i instalacji z CD.
- Rozdział 4, Konfiguracja systemu, został zaktualizowany przez dodanie informacji o jądrach Linuksa w wersji 2.6.x.
- Rozdział 5, Konfiguracja sieci, rozszerzono o nowe informacje dotyczące Samby, NFS i DHCP. Dodano też sekcję o sieciach bezprzewodowych. Rozdział opisuje też istotne zmiany w konfiguracji sieci w Slackware.
- Rozdział 6, System X Window, gruntownie przepisano, dostosowując go do systemów wykorzystujących Xorg. Teraz rozdział opisuje też graficzny menedżer logowania xdm.
- Rozdział 13, Podstawowe polecenia sieciowe, rozszerzono o informacje dotyczące dodatkowych sieciowych programów użytkowych.
- Rozdział 14, Bezpieczeństwo, to nowy rozdział w tej wersji. Wyjaśnia, jak zabezpieczyć system Linux Slackware.
- Rozdział 17, Emacs, to nowy rozdział w tej wersji. Opisuje, jak używać Emacsa, rozbudowanego edytora dla systemów uniksowych.
- Rozdział 18, Zarządzanie pakietami, zaktualizowano o informacje dotyczące skryptów SlackBuild.
- Wprowadzono również wiele innych zmian, drobnych i istotnych, które odzwierciedlają rozwój Slackware.

## Organizacja książki

Rozdział 1, Wprowadzenie

Zawiera materiał wprowadzający do Linuksa, Slackware, ruchów Open Source i Free Software.

Rozdział 2, Pomoc

Opisuje zasoby pomocy dla Linuksa Slackware dostępne w systemie i w Sieci.

Rozdział 3, Instalacja

Opisuje krok po kroku proces instalacji, ilustrowany zrzutami ekranu.

Rozdział 4, Konfiguracja systemu

Opisuje ważne pliki konfiguracyjne i rekompilację jądra.

Rozdział 5, Konfiguracja sieci

Opisuje, jak podłączyć komputer z Linuksem Slackware do sieci. Dotyczy TCP/IP, PPP/połączenia dodzwaniane, sieci bezprzewodowe itp.

#### Rozdział 6, System X Window

Opisuje, jak skonfigurować graficzny system X Window w Slackware i jak go używać.

#### Rozdział 7, Rozruch

Opisuje proces rozruchu komputera i uruchamiania Linuksa Slackware. Zawiera też informacje na temat podwójnego rozruchu z systemami Microsoft Windows.

#### Rozdział 8, Powłoka

Opisuje rozbudowany, wszechstronny interfejs wiersza poleceń w Linuksie.

#### Rozdział 9, Struktura systemu plików

Opisuje strukturę systemu plików, włączając w to prawa własności, uprawnienia i dowiązania.

#### Rozdział 10, Pliki i katalogi

Zawiera opis poleceń, które służą do operacji na plikach i katalogach z interfejsu wiersza poleceń.

#### Rozdział 11, Kontrola procesów

Opisuje polecenia do zarządzania procesami w systemie Linux, używane do zarządzania wieloma uruchomionymi aplikacjami.

#### Rozdział 12, Podstawowa administracja systemem

Opisuje podstawowe zadania administracyjne, takie jak dodawanie lub usuwanie użytkowników, poprawne zamykanie systemu itp.

#### Rozdział 13, Podstawowe polecenia sieciowe

Opisuje zbiór podstawowych klientów sieciowych dołączonych do Slackware.

#### Rozdział 14, Bezpieczeństwo

Opisuje wiele różnych narzędzi, które umożliwiają zabezpieczenie systemu Slackware, w tym `iptables` i `tcpwrappers`.

#### Rozdział 15, Archiwizacja plików

Opisuje różne programy do archiwizacji i kompresji dostępne w Linuksie.

#### Rozdział 16, vi

Opisuje wszechstronny edytor tekstu `vi`.

#### Rozdział 17, Emacs

Opisuje wszechstronny edytor tekstu `Emacs`.

#### Rozdział 18, Zarządzanie Pakietami Slackware

Opisuje programy do zarządzania pakietami Slackware i proces tworzenia własnych pakietów oraz plików znaczników.

#### Rozdział 19, ZipSlack

Opisuje wersję Linuksa ZipSlack, której można używać z poziomu Windows bez potrzeby instalacji.

#### Dodatek A, Ogólna Licencja Publiczna GNU



Opisuje warunki licencji, na której można kopiować i rozpowszechniać Linuksa Slackware i tę książkę.

## Konwencje użyte w książce

Aby otrzymać spójny, łatwy do czytania tekst, użyliśmy w książce kilku konwencji:

### Konwencje typograficzne

#### Kursywa

*Kursywą* piszemy polecenia, tekst podkreślony, pierwsze wystąpienie terminu technicznego.

#### Stała szerokość

Czcionką o stałej szerokości piszemy komunikaty błędów, polecenia, zmienne środowiskowe, nazwy portów, hostów, użytkowników, grup i urządzeń, zmienne i fragmenty kodu.

#### Wytłuszczenie

Czcionką **wytłuszczoną** piszemy tekst wprowadzany przez użytkownika w przykładach.

### Tekst wprowadzany przez użytkownika

Klawisze są **wytłuszczone**, aby odróżniać je od innego tekstu. Kombinacje klawiszy, które mają być naciskane razem, mają „+” między klawiszami, np.:

#### Ctrl+Alt+Del

To znaczy, że trzeba nacisnąć naraz klawisze **Ctrl**, **Alt** i **Del**.

Klawisze, które mają być naciskane kolejno, są oddzielone przecinkami, np.:

#### Ctrl+X, Ctrl+S

Oznacza, że należy nacisnąć najpierw **Ctrl** i **X** naraz, a następnie **Ctrl** i **S** naraz.

### Przykłady

Przykłady rozpoczynające się od **D:\>** oznaczają polecenie MS-DOS®. O ile nie jest powiedziane inaczej, te polecenia można wykonywać z okna „Wiersza poleceń” w środowiskach Microsoft® Windows®.

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

Przykłady rozpoczynające się od # oznaczają polecenie, które musi w Slackware wykonać superużytkownik. Możesz zalogować się jako `root`, aby wpisać polecenie, albo zalogować się na swoje zwykłe konto, a potem użyć `su(1)`, aby otrzymać uprawnienia superużytkownika.

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

Przykłady rozpoczynające się od % oznaczają, że polecenie powinno się wykonywać z konta zwykłego użytkownika. O ile nie jest powiedziane inaczej, do ustawiania zmiennych środowiskowych i innych poleceń powłoki jest używana składnia powłoki C.

```
% top
```

# Podziękowania

Ten projekt stanowi połączenie miesięcy pracy poświęconej przez wielu oddanych ludzi. Sam nie byłbym w stanie wykonać tej pracy. Wielu zasługuje na nasze podziękowania za swój bezinteresowny wkład: Keith Keller za pracę nad sieciami bezprzewodowymi, Joost Kremers za jego samodzielną, wspaniałą pracę nad sekcją o Emacsie, Simon Williams za rozdział o bezpieczeństwie, Jurgen Philippaerts za podstawowe polecenia sieciowe, Cibao Cu Ali G Colibri za inspirację i kopa na rozpęd. Swoje propozycje i poprawki przysłało też wiele innych osób. Niektórzy z nich to: Jacob Anhoej, John Yast, Sally Welch, Morgan Landry i Charlie Law. Chciałbym też podziękować Keithowi Kellerowi za hostowanie listy dyskusyjnej tego projektu. No i wreszcie chciałbym podziękować Patrickowi J. Volkerdingowi za Linuksa Slackware, Davidowi Cantrellowi, Loganowi Johnsonowi i Chrisowi Lumensowi za pierwszą wersję książki Podstawy Linuksa Slackware. Bez ich pracy ta książka by nie zaistniała. Wkład do projektu, duży i mały, wniosło wielu, wielu innych. Mam nadzieję, że wybaczą mi słabą pamięć i niewymienienie ich.

Alan Hicks, maj 2005

## Rozdział 1. Wprowadzenie do Linuksa Slackware

### 1.1 Co to jest Linux?

W 1991 roku Linus Torvalds zaczął pracę nad własnym projektem – jądrem systemu operacyjnego o nazwie Linux. Stworzył go, ponieważ chciał mieć system operacyjny oparty na Uniksie bez potrzeby wydawania kupy szmalu. Oprócz tego, chciał dokładnie poznać możliwości procesora 386. Linux został udostępniony do powszechnego użytku za darmo, aby każdy mógł poznać go i wносить do niego poprawki, zgodnie z Ogólną Licencją Publiczną (General Public License) (więcej na jej temat znajduje się w sekcji 1.3 oraz Dodatku A). Dziś Linux jest jednym z głównych graczy na rynku systemów operacyjnych. Przeportowano go na wiele architektur, m.in. Alpha firmy HP/Compaq, SPARC i UltraSPARC firmy Sun oraz PowerPC firmy Motorola (używane w komputerach Apple Macintosh i IBM RS/6000). Setki, jeśli nie tysiące programistów na całym świecie pracują nad rozwojem Linuksa. Uruchamia się na nim oprogramowanie serwerów internetowych, takich jak Sendmail, Apache czy BIND. Należy pamiętać, że termin „Linux” oznacza w istocie jądro – rdzeń systemu operacyjnego. Ten rdzeń sprawuje kontrolę nad procesorem komputera, pamięcią, dyskami twardymi i urządzeniami peryferyjnymi. Właśnie to tak naprawdę robi Linux: kontroluje działania wykonywane przez komputer i dba o to, aby programy działały, jak należy. Różne firmy i pojedyncze osoby łączą jądro i inne programy razem, tworząc system operacyjny. Każde takie połączenie nazywa się dystrybucją Linuksa.

#### 1.1.1 Słowo na temat GNU

Projekt jądra Linux był przedsięwzięciem jednej osoby, Linusa Torvaldsa, rozpoczętym w 1991. Jednak, jak powiedział Isaac Newton: „Jeśli widziałem dalej niż inni, to dlatego, że stałem na ramionach gigantów”. Gdy Linus Torvalds dopiero zaczynał pracę nad jądrem, Fundacja Wolnego Oprogramowania (Free Software Foundation) ustaliła już założenia oprogramowania powstającego dzięki współpracy wielu ochotników. Jej członkowie nazwali swój projekt GNU, co oznacza „GNU's Not Unix” (GNU to nie Unix). Oprogramowanie GNU wykorzystywało jądro Linuksa od samego początku. Do skompilowania jądra członkowie fundacji użyli swojego kompilatora `gcc`. Dziś wiele programów GNU, takich jak `gcc`

lub `gnutar`, stanowi podstawę wszystkich ważnych dystrybucji Linuksa. Z tego też powodu wielu zwolenników FSF stanowczo twierdzi, że ich praca powinna być ceniona na równi z jądrem Linuksa. Gorąco walczą o to, aby wszystkie dystrybucje Linuksa nazywały się dystrybucjami GNU/Linux.

Jest to temat wielu prowadzonych w Internecie zażartych dyskusji, który przebija tylko odwieczna święta wojna o to, czy lepszy jest edytor `vi`, czy `emacs`. Celem książki nie jest dolewanie oliwy do ognia w tej dyskusji, ale wyjaśnienie nowicюзom zawłości terminologii. Kiedy widzi się termin GNU/Linux, oznacza on dystrybucję Linuksa. Kiedy widzi się samo słowo Linux, oznacza ono albo jądro, albo dystrybucję. Czasem ciężko się w tym połapać. Z reguły nie używa się terminu GNU/Linux, ponieważ jest trudniejszy do wymówienia.

## **1.2 Co to jest Slackware?**

Slackware jest pierwszą powszechnie używaną dystrybucją Linuksa. Stworzył ją Patrick Volkerding pod koniec 1992 roku i udostępnił światu 17 lipca 1993. Volkerding dowiedział się o Linuksie, kiedy poszukiwał niedrogiego interpretera LISP dla jednego ze swoich projektów. Spośród nielicznych dostępnych w owym czasie dystrybucji Patrick wybrał Linuksa SLS firmy Soft Landing System. Używał tej dystrybucji, poprawiając znalezione błędy. Wreszcie połączył wszystkie poprawki i stworzył z ich wykorzystaniem własną dystrybucję dla siebie i znajomych. Szybko stała się ona popularna. Volkerding nadał jej więc nazwę „Slackware” i udostępnił publicznie. W trakcie pracy dodał do Slackware kilka nowych funkcji, m.in. łatwy w obsłudze program instalacyjny wykorzystujący system menu. Wprowadził również zarządzanie pakietami, które umożliwia użytkownikom proste dodawanie, usuwanie i aktualizowanie programów.

Jest wiele powodów, dla których Slackware pozostaje najstarszą używaną dystrybucją Linuksa: nie próbuje ona naśladować wyglądu i zachowania systemu Windows, jest tak podobna do systemów unikowych, jak to możliwe. Nie próbuje ukrywać działających procesów pod pełnymi bajerów graficznymi interfejsami użytkownika (GUI – Graphical User Interface) typu „pokaż i kliknij”. Zamiast tego daje użytkownikom pełną kontrolę, ponieważ mogą oni bezpośrednio zobaczyć wszystko, co dzieje się w systemie. Rozwój dystrybucji nie jest wyznaczany przez ostateczne terminy – kolejne wersje pojawiają się wtedy, kiedy są gotowe.

Slackware jest dla ludzi, których interesuje nauka i dostrajanie systemu do swoich potrzeb. Prostota i stabilność dystrybucji stanowią o tym, że będzie się używać jej jeszcze przez długie lata. Ma ona opinię mocnego serwera i znakomitej stacji roboczej. Na komputerach ze Slackware może działać praktycznie każdy menedżer okien lub środowisko graficzne, są też maszyny w ogóle bez takiego środowiska. Serwery slackware’owe wykorzystuje się we wszelkich możliwych zastosowaniach w firmach. Użytkownicy Slackware należą do najbardziej zadowolonych użytkowników Linuksa. I kto to mówi. :^)

## **1.3 Open Source i Free Software**

W społeczności linuksowej istnieją dwa główne ruchy ideologiczne. Ruch Wolnego Oprogramowania (Free Software), o którym napiszemy za chwilę, działa na rzecz uwolnienia wszelkiego oprogramowania od ograniczeń własności intelektualnej. Jego zwolennicy uważają, że takie ograniczenia blokują postęp technologiczny i działają przeciwko dobru ogółu. Ruch Otwartych Źródeł (Open Source) ma zasadniczo te same cele, ale prezentuje bardziej pragmatyczne podejście do ich osiągnięcia. Jego zwolennicy opierają swoje argumenty na technicznych i ekonomicznych aspektach nieograniczonego udostępniania kodów źródłowych. Mniej interesują ich moralne i etyczne zasady Ruchu Wolnego Oprogramowania.

Na drugim końcu tego kija są grupy, które chciałyby utrzymywać ściślejszą kontrolę nad ich oprogramowaniem.

Na czele ruchu Wolnego Oprogramowania stoi Fundacja Wolnego Oprogramowania (Free Software Foundation) – organizacja zbierająca fundusze dla projektu GNU. Wolne oprogramowanie jest w dużej mierze oparte na ideologii. Często używa się frazy „free as in speech, not free as in beer” (wolny jak w wolności wypowiedzi, a nie darmowy jak piwo). W skrócie, wolne oprogramowanie jest próbą zagwarantowania pewnych praw zarówno dla użytkowników, jak i programistów. Takie prawa oznaczają możliwość dowolnego użytkowania programu, studiowania i modyfikacji kodów źródłowych, dalszego rozpowszechniania źródeł i dokonanych modyfikacji. Aby zagwarantować te prawa, stworzono Ogólną Licencję Publiczną GNU (GNU General Public License). GPL, mówiąc w uproszczeniu, oznacza, że każdy, kto rozpowszechnia skompilowany program na tej licencji, musi również udostępnić jego kod źródłowy. Wolno mu również dokonywać dowolnych modyfikacji programu, ale te modyfikacje muszą również być dostępne w postaci kodu źródłowego. Gwarantuje to, że po „otwarciu” programu nie można go „zamknąć”, o ile nie wyrażą na to zgody wszyscy autorzy każdego fragmentu oprogramowania, także modyfikacji. Większość programów linuksowych jest rozpowszechniana na licencji GPL.

Należy podkreślić, że GPL nie mówi nic o pieniądzach. Brzmi to dziwnie, ale można pobierać opłaty za wolne oprogramowanie. Określenie „wolny” oznacza swobodę dostępu i modyfikacji kodu źródłowego, a nie cenę, którą płaci się za oprogramowanie. (Jednak, jeśli ktoś sprzedał Ci, lub nawet dał za darmo, skompilowaną wersję programu na licencji GPL, ma obowiązek udostępnić jego kod źródłowy.)

Inną powszechnie stosowaną licencją jest licencja BSD. W przeciwieństwie do GPL, BSD nie wymaga od programistów udostępniania kodów źródłowych. Oprogramowanie rozpowszechniane na licencji BSD pozwala na dystrybucję programów w postaci binarnej lub źródłowej pod tylko kilkoma warunkami. Nie wolno używać nazwisk autorów jako reklamy programu. Licencja zwalnia też autorów z jakiegokolwiek odpowiedzialności za wszelkie szkody wynikłe z użytkowania ich oprogramowania. Wiele programów zawartych w dystrybucji Linux Slackware jest dostępnych na licencji BSD.

Na czele powstałego później ruchu Otwartych Źródeł stoi organizacja Inicjatywa na Rzecz Otwartych Źródeł (Open Source Initiative – OSI). Istnieje ona wyłącznie po to, aby zdobywać wsparcie dla oprogramowania typu open source – czyli oprogramowania dostępnego w postaci kodu źródłowego oraz gotowych programów. Ta organizacja nie ma żadnej konkretnej licencji, wspiera natomiast różne rodzaje innych licencji typu open source.

Założeniem OSI jest, aby coraz więcej firm wspierało oprogramowanie open source. OSI pozwala firmom na tworzenie własnych odmian licencji i certyfikuje je pod względem zgodności z założeniami ruchu. Wiele firm chce udostępniać kody źródłowe, ale nie chcą używać licencji GPL. Ponieważ nie mają możliwości dokonywania znaczących zmian w licencji GPL, OSI daje im możliwość stworzenia własnej licencji i wydaje odpowiedni certyfikat zgodności.

Fundacja Wolnego Oprogramowania i Inicjatywa na Rzecz Otwartych Źródeł pomagają sobie wzajemnie, ale nie są tym samym. FSF używa specyficznej licencji i rozpowszechnia oprogramowanie na tej licencji. OSI poszukuje wsparcia dla wszystkich licencji open source, włączając w to licencję FSF. Czasami założenia obu ruchów dotyczące udostępniania kodów źródłowych stanowią powód do sporów. Jednak fakt, że obie zróżnicowane ideologicznie grupy działają w tym samym celu, daje im należną pozycję i szacunek.

## Rozdział 2. Pomoc

Często zdarza się, że potrzebujesz pomocy dotyczącej jakiegoś polecenia, konfiguracji programu czy uruchomienia jakiegoś urządzenia. Czasem po prostu chcesz lepiej zrozumieć polecenie albo dowiedzieć

się, jakie opcje udostępnia. Na szczęście istnieje wiele sposobów na uzyskanie pomocy. Podczas instalacji Slackware masz możliwość zainstalowania pakietów FAQ i HOWTO z grupy „F”. Poszczególne programy również zawierają pliki pomocy z opisem ich opcji, konfiguracji i użytkowania.

## 2.1 Pomoc dostępna w systemie

### 2.1.1. man

Polecenie `man` (skrót od „manual” – instrukcja, podręcznik) jest tradycyjną formą dokumentacji online w systemach operacyjnych Unix i Linux. Składają się na nią odpowiednio sformatowane pliki, tzw. „strony podręcznika” (`man pages`). Takie strony napisano dla ogromnej większości poleceń, są one rozpowszechniane razem z oprogramowaniem. Wykonanie `man` jakiegopolecenie wyświetli stronę podręcznika dla (a jakże) danego polecenia, w tym przypadku byłby to wymyślony program jakiegopolecenie.

Nietrudno się domyślić, że liczba stron podręcznika szybko się zwiększa z każdym dodanym programem, co nawet dla zaawansowanego użytkownika może być zbyt skomplikowane i pogmatwane. Dlatego strony są podzielone na ponumerowane sekcje. Ten system istnieje i sprawdza się od wielu lat, tak więc często spotkasz się z odsyłaczami do numerów sekcji, gdzie znajdują się strony dotyczące odpowiedniego polecenia, programu czy nawet funkcji bibliotecznej używanej przez programistów.

Na przykład:

Zobaczysz odsyłacz `man(1)`. Taka numeracja oznacza, że polecenie „`man`” jest udokumentowane w sekcji 1 (polecenia użytkownika). Możesz uściślić, że potrzebna jest Ci strona podręcznika dotycząca polecenia `man` z sekcji 1, wpisując `man 1 man`. Podawanie numeru sekcji jest użyteczne, jeśli istnieją strony podręcznika o tej samej nazwie, ale w różnych sekcjach.

**Tabela 2-1. Sekcje stron podręcznika**

Sekcja	Zawartość
Sekcja 1	polecenia użytkownika (tylko wprowadzenie)
Sekcja 2	wywołania systemowe
Sekcja 3	wywołania bibliotek C
Sekcja 4	urządzenia (np. <code>hd</code> , <code>sd</code> )
Sekcja 5	formaty plików i protokoły (np. <code>wtmp</code> , <code>/etc/passwd</code> , <code>nfs</code> )
Sekcja 6	gry (tylko wprowadzenie)
Sekcja 7	konwencje, pakiety makro itp. (np. <code>nroff</code> , <code>ascii</code> )
Sekcja 8	administrowanie systemem (tylko wprowadzenie)

Oprócz `man(1)`, istnieją też polecenia `whatis(1)` i `apropos(1)`. Oba służą łatwiejszemu odnajdywaniu informacji w systemie `man`.

Polecenie `whatis` wyświetla krótki opis poleceń systemowych, podobnie do kieszonkowego słownika czy informatora:

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

Polecenie `apropos` szuka strony `man` zawierającej dane słowo kluczowe:

```
% 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
```

Aby dowiedzieć się więcej na temat tych poleceń, przeczytaj ich strony podręcznika. ;)

## 2.1.2 Katalog `/usr/doc`

Źródła większości programów, które tworzymy, zawierają jakąś dokumentację. Pliki typu `README`, instrukcje obsługi, teksty licencji... wszelkie rodzaje dokumentacji dołączonej do źródeł są instalowane w katalogu `/usr/doc`. Każdy program (zazwyczaj) instaluje własną dokumentację w katalogu:

```
/usr/doc/$program-$wersja
```

gdzie `$program` oznacza nazwę programu, na temat którego poszukujesz informacji, a `$wersja` jest (nie inaczej) odpowiednią wersją programu zainstalowanego w systemie.

Na przykład, aby przeczytać dokumentację dotyczącą polecenia `man(1)`, użyj polecenia `cd`, aby przejść do katalogu:

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

Jeśli lektura strony (albo stron) podręcznika nie da Ci wystarczających informacji lub nie znajdziesz ich w ogóle, w następnej kolejności przeszukaj katalog `/usr/doc`.

## 2.1.3 Pliki typu `HOWTO` i `mini-HOWTO`

Zbiór plików `HOWTO` i `mini-HOWTO` to klasyczny przykład ducha idei społeczności `Open Source`. Te pliki są dokładnie tym, co oznacza ich nazwa: „how to”, czyli „jak to zrobić”. Są to dokumenty i instrukcje opisujące, jak coś wykonać. Jeśli zainstalowałeś zbiór `HOWTO`, pliki `HOWTO` znajdują się w `/usr/doc/Linux-HOWTOs`, a `mini-HOWTO` – w `/usr/doc/Linux-mini-HOWTOs`.

W tej samej grupie pakietów znajduje się zbiór `FAQ`. `FAQ` to akronim od

Frequently  
Asked  
Question

Czyli **Najczęściej Zadawane Pytania**.

Te dokumenty są napisane w stylu „Pytanie – Odpowiedź” na (a to ciekawostka) najczęściej zadawane pytania. FAQ są doskonałym miejscem, w którym można znaleźć szybkie rozwiązanie jakiegoś problemu. Jeśli zdecydujesz się zainstalować FAQ, znajdziesz je w katalogu `/usr/doc/Linux-FAQs`.

Naprawdę warto przeczytać te pliki, kiedy nie jesteś absolutnie pewien, jak coś wykonać. Opisują one zdumiewająco szeroki zakres tematów, najczęściej w zdumiewająco wyczerpującej formie. Znakomita lektura!

## **2.2 Pomoc dostępna w Sieci**

Oprócz dokumentacji dołączonej do systemu operacyjnego Linux Slackware, ogromne zasoby wiedzy i informacji są dostępne w Internecie.

### **2.2.1 Oficjalna strona internetowa i fora pomocy**

Oficjalna strona internetowa  
<http://www.slackware.com>

Oficjalna strona Slackware jest czasem nieaktualna, jednak wciąż są na niej zamieszczane informacje na temat najnowszych wersji dystrybucji. Istniało tam kiedyś aktywne forum pomagające użytkownikom, ale zniszczyła je działalność trolli, wichrzycieli i malkontentów. Utrzymanie porządku na forum zaczęło wymagać zbyt wiele pracy, więc Patrick zamknął je. Archiwum starego forum, dostępne w Sieci do przeszukiwania i czytania, znajduje się pod adresem <http://www.userlocal.com/phorum/>.

Po zamknięciu forum na <http://slackware.com> pojawiło się kilka innych stron, które zaoferowały prowadzenie forów o Slackware. Po długich przemyśleniach Pat zdecydował, że oficjalne forum Linuksa Slackware znajdzie się na [www.linuxquestions.org](http://www.linuxquestions.org). Pod tym adresem jest ono obecnie dostępne.

### **2.2.2 Pomoc dostępna przez e-mail**

Każdy, kto kupi oficjalny zestaw CD, ma prawo do darmowej pomocy przy instalacji, świadczonej przez dewelopera za pośrednictwem poczty elektronicznej. Mimo tego należy pamiętać, że my, deweloperzy (i większość użytkowników) Slackware, prezentujemy „Starą szkołę”. To znaczy, że wolimy pomagać ludziom, którzy są naprawdę zainteresowani i są skorzy pomóc sami sobie. Zawsze będziemy się starać pomóc tym, którzy piszą do nas e-maile z prośbą o pomoc. Zanim jednak wyślesz do nas e-mail, przeszukaj dokumentację i stronę internetową (szczególnie FAQ i ewentualnie fora wymienione poniżej). Prawdopodobnie szybciej uzyskasz pomoc w ten sposób, a czym mniej e-maili będziemy dostawać, tym szybciej będziemy w stanie pomóc naprawdę potrzebującym.

Adres e-mail dla pomocy technicznej to: [support@slackware.com](mailto:support@slackware.com)  
Inne adresy e-mail i informacje kontaktowe są zamieszczone na stronie www.

#### **2.2.2.1 Listy mailingowe projektu Linux Slackware**

Mamy kilka list mailingowych, dostępnych w formie periodyku i tradycyjnej. Oto informacje, jak je zasubskrybować.

Aby zasubskrybować listę mailingową, wystarczy wysłać e-mail na adres:

[majordomo@slackware.com](mailto:majordomo@slackware.com)

z frazą „*subscribe [nazwa listy]*” w treści listu. Poniżej znajduje się wykaz list, nazwy jednej z nich należy użyć w e-mailu.

Archiwa list mailingowych są zamieszczone na stronie Slackware pod adresem:

<http://slackware.com/lists/archive>

*slackware-announce*

Lista *slackware-announce* służy do informowania o nowych wersjach, ważnych uaktualnieniach i innych zdarzeniach.

*slackware-security*

Lista *slackware-security* służy do informowania o ważnych kwestiach dotyczących bezpieczeństwa. Na tę listę są natychmiast wysyłane wszelkie, związane bezpośrednio ze Slackware, ogłoszenia dotyczące exploitów, luk zabezpieczeń itp.

Te listy są również dostępne w formie periodyku (digest). To znaczy, że dostajesz jeden duży, połączony e-mail dziennie zamiast pojedynczych e-maili w ciągu dnia. Jako że zwykły użytkownik nie może wysłać wiadomości na te listy, a ponadto nie ma na nich dużego ruchu, nie zalecam subskrybowania list w formacie periodyku. Jeśli jednak sobie życzysz, taka forma jest dostępna:

*slackware-announce-digest* lub *slackware-security-digest*.

## 2.2.3 Nieoficjalne strony i fora internetowe

### 2.2.3.1 Strony

Google

<http://www.google.com>

Pan i władca wyszukiwarek. Używaj, kiedy absolutnie, bezwzględnie musisz znaleźć wszelkie najdrobniejsze informacje dotyczące jakiegoś problemu. Nie przyjmuj żadnych środków zastępczych.

Google:Linux

<http://www.google.com/linux>

Wyszukiwanie dotyczące Linuksa

Google:BSD

<http://www.google.com/bsd>

Wyszukiwanie dotyczące BSD. Slackware jest tak blisko spokrewniony z oryginalnymi systemami uniksopodobnymi, że większość informacji znaleziona tutaj jest szczegółowa i w prawie 100% ma zastosowanie w Slackware. Często informacje znalezione tutaj są o wiele bardziej techniczne niż wyniki



wyszukiwania linuksowego, które są bardziej zorientowane na sferę Public Relations.

Google:Groups  
<http://groups.google.com>

Wyszukiwanie diamentów mądrości w kopalni postów usenetowych.

userlocal.com  
<http://userlocal.com>

Skarbnica wiedzy, dobrych rad, doświadczenia z pierwszej ręki i interesujących artykułów. Tutaj pojawiają się pierwsze doniesienia o nowinkach w świecie Slackware.

### **2.2.3.2 Internetowe fora online**

[linuxquestions.org](http://linuxquestions.org)  
<http://www.linuxquestions.org/questions/forumdisplay.php?forumid=14>  
Oficjalne forum użytkowników Slackware.

[LinuxISO.org](http://LinuxISO.org) Slackware Forum  
„miejsce, skąd można pobrać Linuksa i gdzie można uzyskać pomoc”

[alt.os.linux.slackware](http://alt.os.linux.slackware) FAQ  
<http://wombat.san-francisco.ca.us/perl/fom>

(od tłum.:) Polskie forum o Slackware: <http://forum.slackware.pl>.

### **2.2.3.3 Grupy usenetowe (NNTP)**

Usenet od dawna jest miejscem, w którym komputerowcy zbierają się i pomagają sobie nawzajem. Istnieje niewiele grup poświęconych Linuksowi Slackware, jednak większość ich użytkowników ma naprawdę dużą wiedzę.

[alt.os.linux.slackware](http://alt.os.linux.slackware)

[alt.os.linux.slackware](http://alt.os.linux.slackware), znane lepiej jako aols (nie mylić z AOL®!) to jedno z najaktywniejszych miejsc, w których można znaleźć pomoc techniczną dotyczącą problemu ze Slackware. Podobnie jak w przypadku każdej grupy usenetowej, kilku niezbyt pomocnych użytkowników („trolli”) może zepsuć przyjemność ciągłymi kłótniami. Nauczenie się, aby ignorować trolli i rozpoznawanie ludzi naprawdę pomocnych to podstawa korzystania z tego cennego źródła wiedzy.

(od tłum.:) Polska grupa usenetowa: [alt.pl.comp.os.linux.slackware](http://alt.pl.comp.os.linux.slackware).

## **Rozdział 3. Instalacja**

Przed rozpoczęciem używania Linuksa Slackware musisz skądś go zdobyć i zainstalować. Samo zdobycie nie jest trudne. Wystarczy kupić oficjalny komplet płyt CD lub pobrać system za darmo z Internetu. Instalacja też jest dość prosta, o ile masz podstawową wiedzę na temat swojego komputera i chcesz nauczyć się czegoś nowego. Sam program instalacyjny w dużej mierze prowadzi Cię krok po kroku. Dzięki temu oprogramowanie można zainstalować i uruchomić w dość krótkim czasie. Mówiąc szczerze,

Slackware szczyli się jednym z najkrótszych czasów instalacji wśród wszystkich dużych dystrybucji Linuksa.

### 3.1 Skąd wziąć Slackware

#### 3.1.1 Oficjalne komplety CD i zestawy pudełkowe

Oficjalne komplety płyt CD z Linuksem Slackware można nabyć w firmie Slackware, Inc. Taki komplet składa się z 4 płyt. Pierwsza zawiera podstawowe oprogramowanie potrzebne do uruchomienia serwerów i systemu X Window. Druga płyta zawiera system typu „live” – jest to CD rozruchowy, z którego można uruchomić Linuksa bezpośrednio. Taki system po uruchomieniu instaluje się w pamięci operacyjnej RAM i stanowi instalację tymczasową, przydatną np. do poznawania Linuksa, odzyskania danych lub przywrócenia działania systemu po błędzie. Ta płyta zawiera również dodatkowe pakiety, takie jak środowiska KDE i GNOME, a także, w katalogu „extra”, pewną liczbę dodatkowych programów, niewymaganych do poprawnego działania systemu. Na trzeciej i czwartej płycie są umieszczone kody źródłowe wszystkich programów zawartych w dystrybucji, jak również oryginalna wersja tej książki.

Możesz także kupić zestaw pudełkowy. Zawiera on 4 płyty CD, drukowaną kopię tej książki okładce i wiele slackware’owych gadżetów, którymi możesz szpanować przed kolegami. Możesz też kupić subskrypcje samych płyt CD po obniżonej cenie.

Preferowaną metodą dokonywania zakupów jest zamówienie w online’owym sklepie Slackware:

<http://store.slackware.com>

Zamówienie można też złożyć telefonicznie lub przez e-mail.

**Tabela 3-1. Informacje kontaktowe ze Slackware, Inc.**

Metoda	Kontakt
telefon	1-(925) 674-0783
www	<a href="http://store.slackware.com">http://store.slackware.com</a>
e-mail	<a href="mailto:orders@slackware.com">orders@slackware.com</a>
poczta zwykła	1164 Claremont Drive, Brentwood, CA 94513

#### 3.1.2 Przez Internet

Linuksa Slackware można pobrać za darmo z Internetu. Można wysłać w takim przypadku e-maile z pytaniami i prośbą o pomoc, ale wyższy priorytet mają osoby, które nabyły oficjalny komplet płyt. Otrzymujemy wiele e-maili, a nasz czas jest ograniczony. Dlatego przed wysłaniem listu przeczytaj najpierw rozdział drugi – Pomoc.

Oficjalna strona internetowa projektu Linux Slackware to:

<http://www.slackware.com/>

Podstawowy serwer FTP Linuksa Slackware to:

<ftp://ftp.slackware.com/pub/slackware/>

Pamiętaj, że jakkolwiek nasz serwer ftp jest dostępny publicznie, nie ma nieograniczonej przepustowości łącza. Aby pobrać Slackware, użyj raczej najbliższego serwera lustrzanego. Niepełna lista takich serwerów znajduje się na naszej stronie pod adresem <http://www.slackware.com/getslack>.

## 3.2 Wymagania systemowe

Aby bezproblemowo zainstalować Slackware, należy posiadać co najmniej:

**Tabela 3-2. Wymagania systemowe**

Sprzęt	Wymagania
Procesor	586
RAM	32 MB
Miejsce na dysku	1 GB
Napęd	CD-ROM 4x

Jeśli masz rozruchową płytę CD, prawdopodobnie nie będziesz potrzebował napędu dyskietek. Oczywiście, jeśli nie posiadasz napędu CD, będziesz potrzebował stacji dyskietek, aby przeprowadzić instalację sieciową. Do instalacji typu NFS jest potrzebna karta sieciowa. Przeczytaj sekcję NFS, aby dowiedzieć się więcej na ten temat.

Wymagania dotyczące miejsca na dysku mogą być mylące. 1 GB z reguły wystarcza dla instalacji minimalnej. Jeśli jednak chcesz przeprowadzić pełną instalację, będziesz potrzebował około 2 GB na system i dodatkowego miejsca na własne pliki. Większość użytkowników nie instaluje pełnego systemu, wielu z nich wystarcza wręcz 100 MB.

Slackware można zainstalować na systemach z mniejszą ilością RAM, mniejszymi dyskami twardymi i słabszymi procesorami, ale wymaga to więcej pracy. Jeśli się jej nie boisz, przeczytaj plik `LOWMEM.TXT` w drzewie plików dystrybucji, aby uzyskać kilka cennych rad na ten temat.

### 3.2.1 Grupy pakietów

Ze względu na zachowanie prostoty, Slackware od początku podzielono na grupy pakietów (software series). Pierwotnie nazywały się one „zestawami dyskietek” (disk sets), ponieważ przystosowano je do instalacji z dyskietek. Teraz grup pakietów używa się do podziału oprogramowania Slackware na poszczególne kategorie. Nie ma już możliwości instalacji z samych dyskietek.

Oto krótki opis każdej grupy pakietów:

**Tabela 3-3. Grupy pakietów**

Grupa	Zawartość
A	Podstawowy system. Zawiera oprogramowanie wystarczające do uruchomienia systemu, edytor tekstu i podstawowy program do komunikacji.

AP	Różne aplikacje niewymagające systemu graficznego X Window.
D	Narzędzia deweloperskie. Kompilatory, debugery, interpretery, strony podręcznika man.
E	Edytor GNU Emacs.
F	Dokumentacja FAQ, HOWTO i inna.
GNOME	Graficzne środowisko użytkownika GNOME
K	Kody źródłowe jądra Linuksa
KDE	K Desktop Environment. Graficzne środowisko użytkownika, z wyglądu i zachowania przypominające systemy MS Windows i Mac OS. Zawiera wymaganą przez KDE bibliotekę Qt.
KDEI	Pakiety lokalizacyjne dla KDE
L	Dynamicznie łączone biblioteki, wymagane przez wiele programów.
N	Programy do pracy w sieci. Demony, programy pocztowe, telnet, czytniki grup usenetowych itp.
T	System formatowania dokumentów teTeX
TCL	Język programowania Tool Command Language. Zawiera Tk, TclX i TkDesk.
X	Podstawowy system X Window
XAP	Aplikacje graficzne, które nie należą do większych środowisk graficznych (np. Ghostscript, Netscape).
Y	Konsolowe gry BSD.

## 3.2.2 Metody instalacji

### 3.2.2.1 Z dyskietek

Kiedyś można było zainstalować całego Linuksa Slackware z dyskietek. Zwiększające się rozmiary pakietów (a dokładniej, niektórych programów) spowodowały zarzucenie tej metody. Możliwość częściowej instalacji z dyskietek istniała jeszcze w wersji 7.1. Prawie w całości można było zainstalować grupy pakietów A i N. Otrzymywało się wówczas podstawowy system, stanowiący bazę dla doinstalowania reszty dystrybucji. Jeśli planujesz instalację z dyskietek (najczęściej w przypadku, gdy masz stary sprzęt), zwykle zaleca się inną metodę lub użycie starszej wersji. Szczególnie popularne są Slackware 4.0 i 7.0.

Uwaga: Potrzebujesz dyskietek, jeśli nie masz rozruchowego dysku CD lub chcesz przeprowadzić instalację typu NFS.

### 3.2.2.2 Z CD-ROMu

Jeśli masz rozruchowy dysk CD, dostępny w oficjalnym komplecie dysków ze Slackware Linux, Inc. (zob. sekcję Skąd wziąć Slackware), instalacja z CD będzie łatwiejsza. Jeśli nie, będziesz musiał uruchomić instalację z dyskietek. Być może będziesz potrzebował specjalnych dyskietek, jeśli masz nietypowy sprzęt, nieobsługiwany przez jądro na rozruchowym CD.

Od wersji Slackware 8.1 używa się nowej metody tworzenia rozruchowych dysków CD. Ta metoda nie działa poprawnie z niektórymi nietypowymi BIOSami (warto zauważyć, że większość rozruchowych dysków z Linuksem ma z tym problemy). W takim przypadku również jest zalecane uruchomienie instalacji z dyskietek.

Przeczytaj sekcje 3.2.3 i 3.2.5, aby dowiedzieć się więcej na temat wyboru i tworzenia potrzebnych dyskietek rozruchowych.

### 3.2.2.3 NFS

NFS (sieciowy system plików – Network File System) to sposób, w jaki udostępnia się systemy plików zdalnym komputerom. Instalacja typu NFS pozwala na zainstalowanie Slackware z innego komputera w sieci. Maszyna, z której się instaluje, musi udostępniać w sieci drzewo dystrybucji Slackware dla komputera, na którym ma być zainstalowany system. Wymaga to oczywiście pewnej wiedzy na temat NFS. Ten temat jest omówiony w sekcji 5.6.

Instalację NFS można przeprowadzić metodą PLIP (przez port równoległy), SLIP i PPP (jakkolwiek nie przez połączenie modemowe). Zaleca się jednak wykorzystanie karty sieciowej, jeśli to możliwe. Instalowanie systemu operacyjnego przez port drukarki trwałoby bardzo, bardzo długo.

## 3.2.3 Dyskietka rozruchowa

Dyskietka rozruchowa (boot disk) to dyskietka, z której uruchamia się komputer, aby rozpocząć instalację. Zawiera ona skompresowany obraz jądra, które kontroluje sprzęt podczas instalacji, dlatego jest niezbędna (chyba że uruchamiasz komputer z CD, co jest omówione w sekcji CD-ROM). Obrazy dyskietek rozruchowych znajdują się w katalogu `bootdisks/` w drzewie dystrybucji.

Jest więcej dyskietek rozruchowych niż można tu wymienić (około 16). Pełna ich lista, wraz z opisami, znajduje się w drzewie dystrybucji, w pliku `bootdisks/README.TXT`. Większość ludzi używa jednak tylko obrazu dyskietki `bare.i` (dla urządzeń IDE) lub `scsi.s` (dla urządzeń SCSI).

Instrukcje, jak utworzyć dyskietkę z odpowiedniego obrazu, znajdują się w sekcji 3.2.6.

Po uruchomieniu komputera wyświetli się monit o włożenie dyskietki root. Należy to wykonać i kontynuować instalację.

## 3.2.4 Dyskietki root

Dyskietki root (dyskietki główne) zawierają program instalacyjny i system plików używany podczas instalacji. Są one również niezbędne. Obrazy dyskietek root znajdują się w katalogu `rootdisks` w drzewie dystrybucji. Należy utworzyć dwie dyskietki root z obrazów `install.1` i `install.2`. W

tym katalogu są także obrazy `network.dsk`, `pcmcia.dsk`, `rescue.dsk` i `sbootmgr.dsk`.

### 3.2.5 Dyskietka dodatkowa

Dyskietka dodatkowa jest potrzebna w przypadku instalacji typu NFS lub jeśli w komputerze znajdują się urządzenia PCMCIA. Obrazy dyskietek dodatkowych są umieszczone w katalogu `rootdisks` w drzewie dystrybucji. Nazywają się `network.dsk` oraz `pcmcia.dsk`. Ostatnio dodano obrazy innych dyskietek dodatkowych, takie jak `rescue.dsk` i `sbootmgr.dsk`. Dyskietka ratunkowa (`rescue`) to niewielka dyskietka root, która działa na dysku wirtualnym o wielkości 4 MB w pamięci RAM. Jest wyposażona w podstawowe programy sieciowe i edytor `vi`, używany np. do poprawiania niewłaściwych plików konfiguracyjnych na niedziałającym komputerze. Dyskietki `sbootmgr.dsk` używa się do uruchamiania systemu z innych urządzeń. Należy jej użyć, jeśli napęd CD-ROM nie chce uruchomić Slackware z płyty CD. Ta dyskietka wyświetla podczas startu wiele pytań na temat rozruchu i może służyć do obejścia problematycznych BIOSów.

Dyskietka root wyświetli monit o włożenie dyskietki dodatkowej, gdy sama skończy się uruchamiać.

### 3.2.6 Tworzenie dyskietek rozruchowych

Po wybraniu odpowiedniego obrazu trzeba go przenieść na dyskietkę. Ten proces zależy od systemu, którego używasz. Jeśli masz Linuksa (lub jakiś inny uniksopodobny system operacyjny), użyjesz polecenia `dd(1)`. Zakładając, że wybranym obrazem jest `bare.i`, a napęd dyskietek to `/dev/fd0`, składnia polecenia tworzącego dyskietkę `bare.i` wygląda następująco:

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

Jeśli używasz systemu operacyjnego firmy Microsoft, musisz użyć programu `RAWRITE.EXE`. Znajduje się on w drzewie dystrybucji, w tym samym katalogu, co obrazy dyskietek. Zakładając, że wybranym obrazem jest `bare.i`, a napęd dyskietek to `A:`, otwórz okno DOS i wpisz:

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

## 3.3 Partycjonowanie

Po uruchomieniu instalacji w wybrany sposób musisz podzielić dysk na partycje. Partycja to miejsce na dysku, gdzie zostanie utworzony linuksowy system plików i zainstalowany Slackware. Zalecamy utworzenie co najmniej dwóch partycji: jednej dla głównego systemu plików – root (`/`) i jednej dla przestrzeni wymiany (`swap`).

Po zakończeniu uruchamiania się dyskietki root wyświetli się znak zgłoszenia logowania (`login prompt`). Zaloguj się jako `root` (bez hasła). W znaku zgłoszenia powłoki (`shell prompt`) wpisz `cfdisk(8)` lub `fdisk(8)`. Program `cfdisk` ma interfejs bardziej przyjazny dla użytkownika niż standardowy `fdisk`, ale brakuje mu pewnych funkcji. Program `fdisk` w skrócie objaśniamy poniżej.

Rozpocznij od uruchomienia `fdisk` dla Twojego dysku twardego. W Linuksie dyski twarde nie mają przypisanych liter napędów, ale są reprezentowane przez pliki. Pierwszy dysk IDE (pierwszy dysk główny – primary master) to `/dev/hda`, pierwszy dysk podrzędny (primary slave) to `/dev/hdb` itd.

Dyski SCSI mają podobną konwencję nazewniczą, ale w postaci /dev/sdX. Uruchom `fdisk`, przekazując mu informację, którym dyskiem ma się zająć:

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

Jak wszystkie dobre programy uniksowe, `fdisk` ma swój znak zgłoszenia (a co, myślałeś, że menu?). Na początek powinieneś sprawdzić istniejące partycje. Wykonaj to, wpisując `p` po znaku zgłoszenia `fdisk`:

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

Wyświetlą się różnorakie informacje na temat aktualnie istniejących partycji. Większość użytkowników przeznacza osobny dysk w całości dla Linuksa, więc usuwają po prostu wszystkie istniejące na nim partycje, aby utworzyć miejsce dla nowego systemu.



**KONIECZNIE WYKONAJ KOPIĘ ZAPASOWĄ WSZYSTKICH DANYCH, KTÓRE CHCESZ ZACHOWAĆ, A KTÓRE ZNAJDUJĄ SIĘ NA PRZEZNACZONYCH DO USUNIĘCIA PARTYCJACH.**

Ewentualne odzyskanie danych po usunięciu partycji jest bardzo trudne, jeśli nie niemożliwe, więc zawsze rób kopie zapasowe, zanim zaczniesz pracę z partycjami.

W tabeli z informacjami o partycjach widać m.in. numer partycji, jej rozmiar i typ. Są tam też inne informacje, ale na razie się nimi nie przejmuj. Na początek usuń wszystkie partycje na dysku, aby potem utworzyć partycje linuksowe. Aby usunąć partycję, wpisz `d`:

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

Tę czynność powtórz dla każdej partycji. Po usunięciu wszystkich utwórz linuksowe. Załóżmy, że utworzysz jedną partycję dla systemu głównego i jedną dla przestrzeni wymiany. Uwaga: Schematy partycjonowania w systemach uniksowych są tematem wielu zażartych dyskusji. Prawie każdy użytkownik ma swój najlepszy sposób dzielenia. Powinieneś utworzyć co najmniej dwie partycje: jedną dla / i jedną dla wymiany. Z czasem wypracujesz najlepszą dla siebie metodę.

Ja używam dwóch podstawowych schematów. Pierwszy jest przeznaczony dla stacji roboczych. Tworzę 4 partycje: /, /home, /usr/local i wymiany. Pozwala to na reinstalację bądź aktualizację całości systemu w / bez usuwania danych w /home i skompilowanych przeze mnie aplikacji w /usr/local. W drugim schemacie, przeznaczonym dla serwerów, zwykle zastępuję /usr/local przez /var. Wiele serwerów zapisuje informacje na tej partycji, więc oddzielenie jej od / ma pozytywny wpływ na wydajność. Na razie pozostanę jednak przy dwóch partycjach: / i wymiany.

Partycje tworzy się 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
```

Pamiętaj, aby tworzyć partycje podstawowe (primary). Pierwsza będzie partycja wymiany. Mówisz `fdiskowi`, aby utworzył podstawową partycję o numerze 1. Rozpoczynać się ona będzie od cylindra 0. Dla ostatniego cylindra wpisz `+64M`. Otrzymasz 64-megabajtową partycję wymiany (rozmiar partycji wymiany zależy od ilości pamięci RAM. Tradycyjnie tworzy się partycję wymiany dwa razy większą od RAM). Następnie stwórz partycję podstawową o numerze 2. Rozpocznij od pierwszego dostępnego cylindra, a zakończ na ostatnim możliwym:

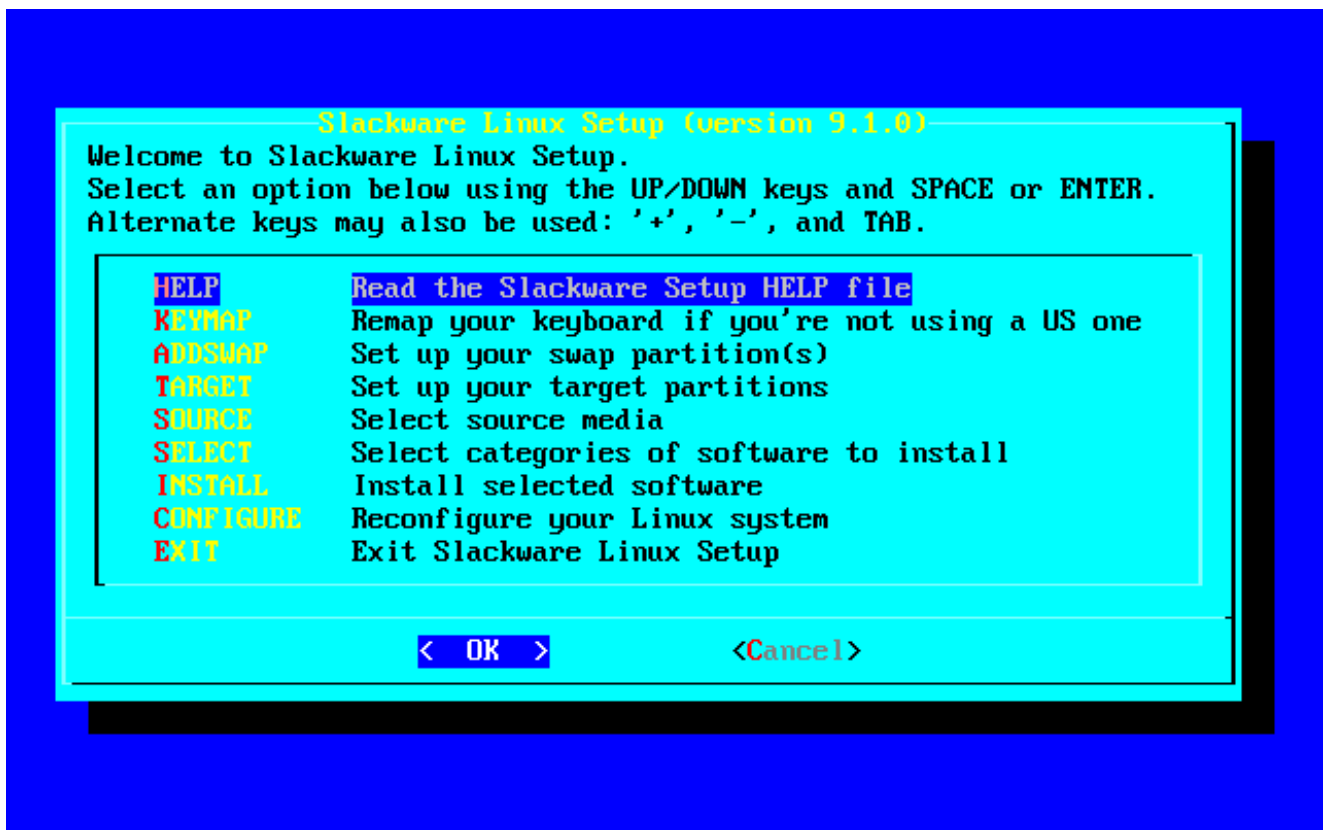
```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
P
Partition number (1-4):2
First cylinder (124-1060, default 124):124
Last cylinder or +size or +sizeM or +sizeK (124-1060, default 1060):1060
```

Prawie skończone. Musisz teraz zmienić typ pierwszej partycji na 82 (linuksowa partycja wymiany – Linux swap). Wpisz `t`, aby zmienić typ, wybierz pierwszą partycję i wpisz `82`. Przed zapisaniem zmian na dysku spójrz ostatni raz na tablicę partycji. Użyj polecenia `p` w `fdisku` do jej wyświetlenia. Jeśli wygląda poprawnie, wpisz `w`. Zapisze to zmiany na dysku i zakończy działanie programu `fdisk`.

### 3.4 Program *setup*

Po utworzeniu partycji możesz instalować Slackware. Następnym krokiem będzie uruchomienie programu instalacyjnego – `setup(8)`. Wystarczy wpisać `setup` po znaku zgłoszenia powłoki. `Setup` to wykorzystujący menu program używany do instalowania pakietów Slackware i konfigurowania systemu.





Witamy w programie instalacyjnym Linuksa Slackware.	
Wybierz opcję poniżej, używając klawiszy strzałek GÓRA/DÓŁ oraz SPACJI i ENTERA.	
Możesz też używać innych klawiszy: '+', '-' i TAB.	
HELP (pomoc)	Przeczytaj plik pomocy programu Slackware setup
KEYMAP	Zmień mapowanie klawiatury, jeśli nie używasz układu amerykańskiego (US)
ADDSWAP	Ustaw partycję (lub partycje) wymiany
TARGET	Ustaw partycje docelowe
SOURCE	Wybierz nośnik źródłowy
SELECT	Wybierz, które kategorie oprogramowania zainstalować
INSTALL	Zainstaluj wybrane oprogramowanie
CONFIGURE	Zrekonfiguruj Linuksa
EXIT	Wyjdź z programu instalacyjnego
<OK>	<Anuluj>

Podczas procesu instalacji przechodzisz krok po kroku przez poszczególne opcje programu setup, w takiej kolejności, w jakiej są wymienione (nie musisz się jej trzymać, ale w innym przypadku najprawdopodobniej instalacja nie powiedzie się). Pozycje w menu wybierasz, używając klawiszy ze strzałkami w górę i w dół. Przyciski „Okay” i „Cancel” wybierasz, używając strzałek w prawo i w lewo. Oprócz tego, każda opcja ma odpowiadający jej klawisz, który jest podświetlony w nazwie. Opcje, które można zaznaczyć (oznaczone przez [X]), można zaznaczać i odznaczać przy użyciu klawisza spacji.

Wszystko to jest oczywiście opisane w sekcji „help” programu setup, ale jak już pisać książkę, to porządnie.

### 3.4.1 Pomoc (HELP)

Jeśli instalujesz Slackware po raz pierwszy, warto spojrzeć na ekran pomocy. Znajdują się tam krótkie opisy każdej części programu setup (podobne do zawartych tutaj, ale nie tak szczegółowe) i instrukcje, jak poruszać się podczas instalacji.

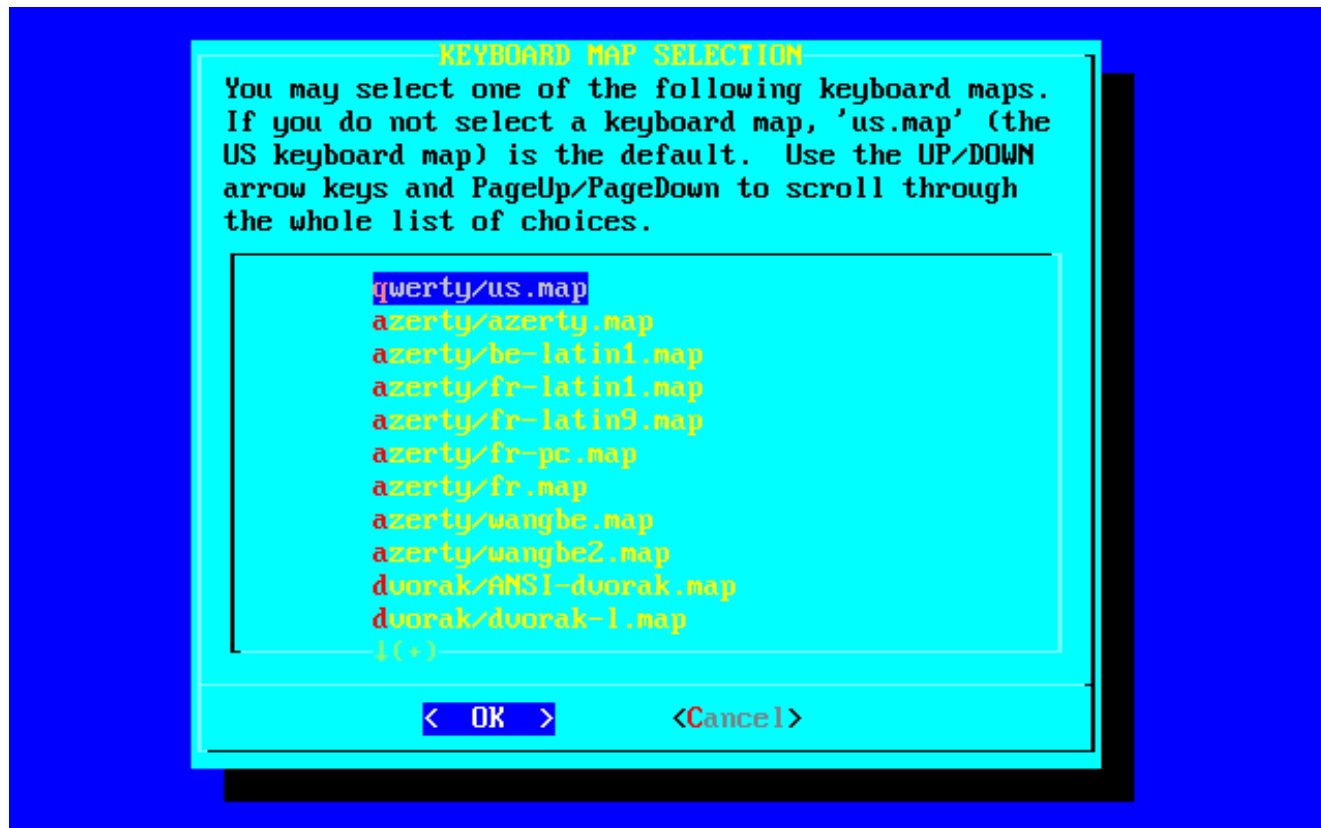


Pomoc Linuksa Slackware	
Na początek trochę pomocy dotyczącej pomocy. Kiedykolwiek podczas instalacji napotkasz przeglądarkę tekstową taką jak ta, możesz się po niej poruszać w ten sposób:	
STRONA W GÓRĘ/SPACJA	Przejdź w dół o jedną stronę
STRONA W DÓŁ/'b'	Przejdź w górę o jedną stronę
ENTER/STRZAŁKA W DÓŁ/'j'	Przejdź w dół o jeden wiersz
STRZAŁKA W GÓRĘ/'k'	Przejdź w górę o jeden wiersz
STRZAŁKA W LEWO/'h'	Przewiń w lewo
STRZAŁKA W PRAWO/'l'	Przewiń w prawo
'0'	Przejdź na początek wiersza
HOME/'g'	Przejdź na początek pliku

END/'G'	Przejdź na koniec pliku
'/'	Wyszukiwanie do przodu
<OK>	

### 3.4.2 Mapa klawiatury (KEYMAP)

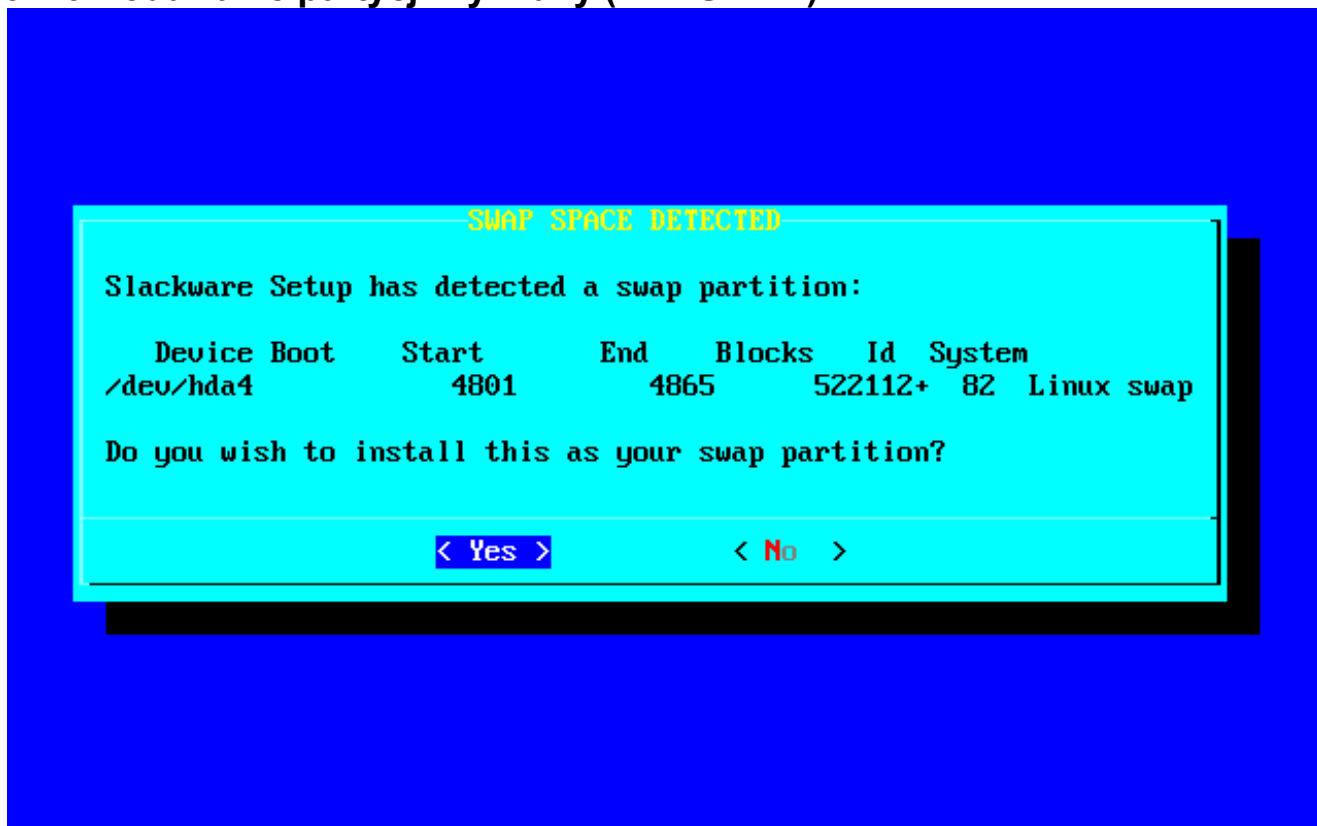
Jeśli potrzebujesz układu klawiszy innego niż „qwerty” używany w Stanach Zjednoczonych, przeczytaj tę sekcję. Możesz tu wybrać różne alternatywne układy klawiatury.



Wybierz jedną z następujących map klawiatury. Jeśli nie wybierzesz mapy klawiatury, zostanie użyta domyślna mapa 'us.map' (mapa klawiatury US). Użyj klawiszy strzałka w górę/w dół lub PageUp/PageDown, aby przewijać listę dostępnych map.

<OK>	<Anuluj>
------	----------

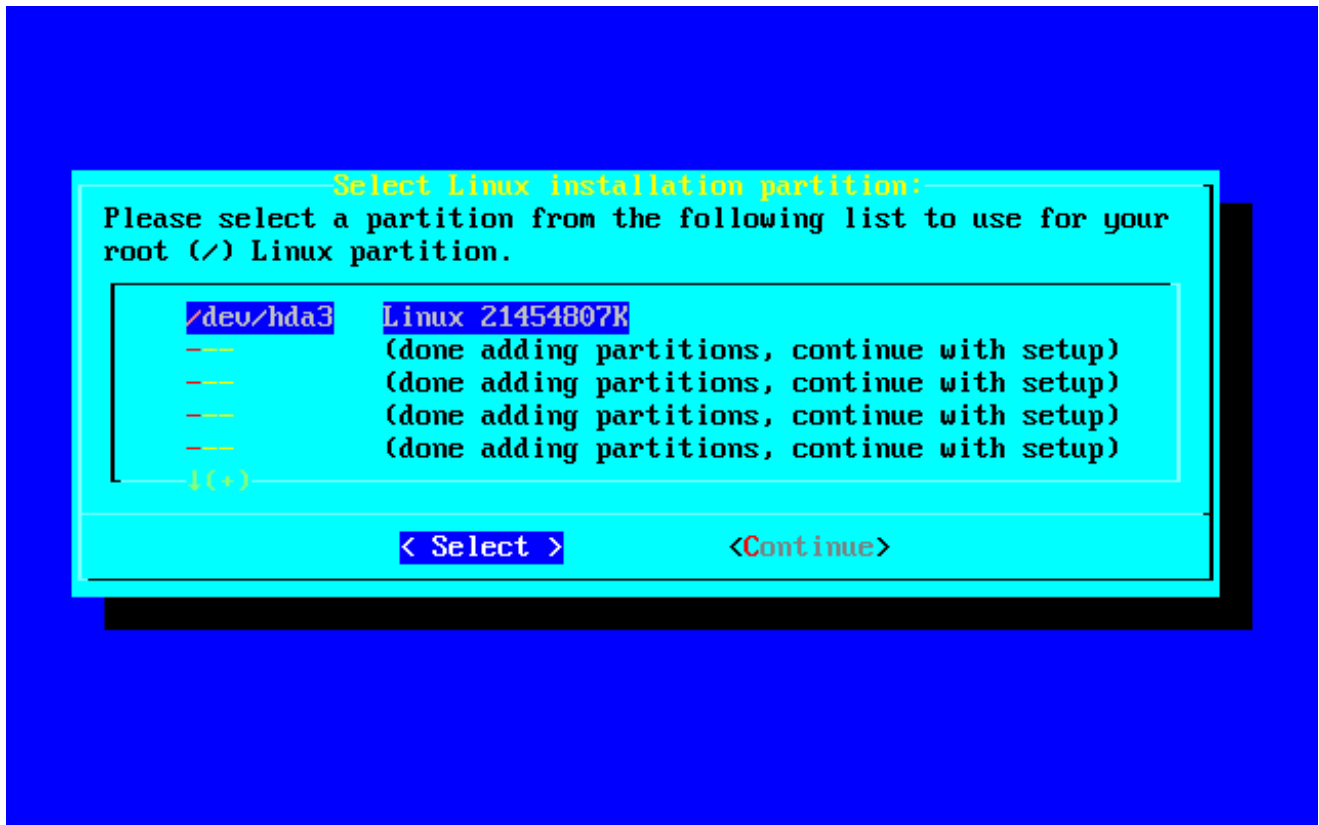
### 3.4.3 Dodawanie partycji wymiany (ADDSWAP)



Program Slackware Setup znalazł partycję wymiany:						
Urządzenie	Rozruch	Początek	Koniec	Bloki	Id	System
/dev/hda4		4801	4865	522112+	82	Linuksowa partycja wymiany
			<Tak>		<Nie>	

Jeśli dodałeś partycję wymiany (sekcja 3.3), teraz możesz ją uruchomić. Partycje wymiany zostaną automatycznie wykryte i wyświetlone, będziesz mógł wybrać jedną do sformatowania i uaktywnienia.

### 3.4.4 Partycje docelowe (TARGET)



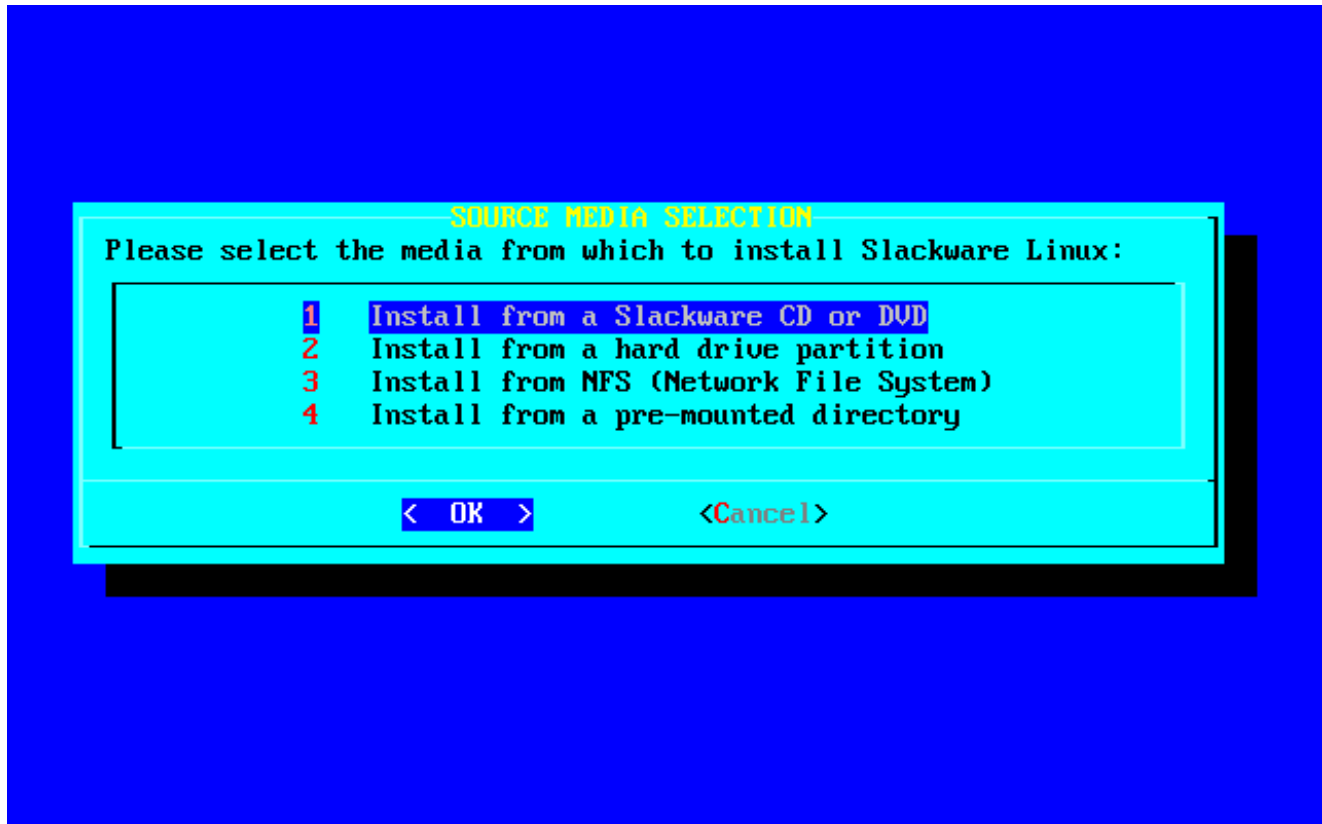
Wybierz z poniższej listy partycję, której chcesz używać jako linuksowej partycji głównej – root (/).	
/dev/hda3	Linux 21454807K
---	(zakończono dodawanie partycji, kontynuuj instalację)
<Wybierz>	<Kontynuuj>

W sekcji partycji docelowych formatujesz inne partycje (nie wymiany) oraz przypisujesz im punkty montowania w systemie plików. Wyświetlona jest lista partycji na dysku twardym. Dla każdej z nich możesz wybrać, czy ją sformatować. W zależności od wersji jądra możesz wybrać między systemem plików reiserfs (domyślnym), ext3, ext2, jfs i xfs. Większość ludzi używa reiserfs lub ext3. W przyszłości prawdopodobnie zostanie dodana obsługa reiserfs4.

Pierwszą opcją w tej sekcji jest wybór partycji, na której zostanie zainstalowany główny system plików (root, czyli /). Następnie będziesz mógł przypisać inne partycje do innych systemów plików (na przykład zdecydować, że trzecia partycja, /dev/hda3, będzie zawierała system plików użytkownika /home. To tylko przykład, wybór należy do Ciebie).

### 3.4.5 Nośnik źródłowy (SOURCE)

W sekcji source wybierasz nośnik źródłowy, z którego instalujesz Slackware. Obecnie są do wyboru cztery źródła: CD-ROM, partycja na dysku twardym, NFS, wcześniej zamontowany katalog.



Wybierz nośnik, z którego będzie zainstalowany Linux Slackware	
1. Zainstaluj z dysku CD/DVD ze Slackware	
2. Zainstaluj z partycji na dysku twardym	
3. Zainstaluj z sieciowego systemu plików (NFS)	
4. Zainstaluj z wcześniej zamontowanego katalogu	
<OK>	<Anuluj>

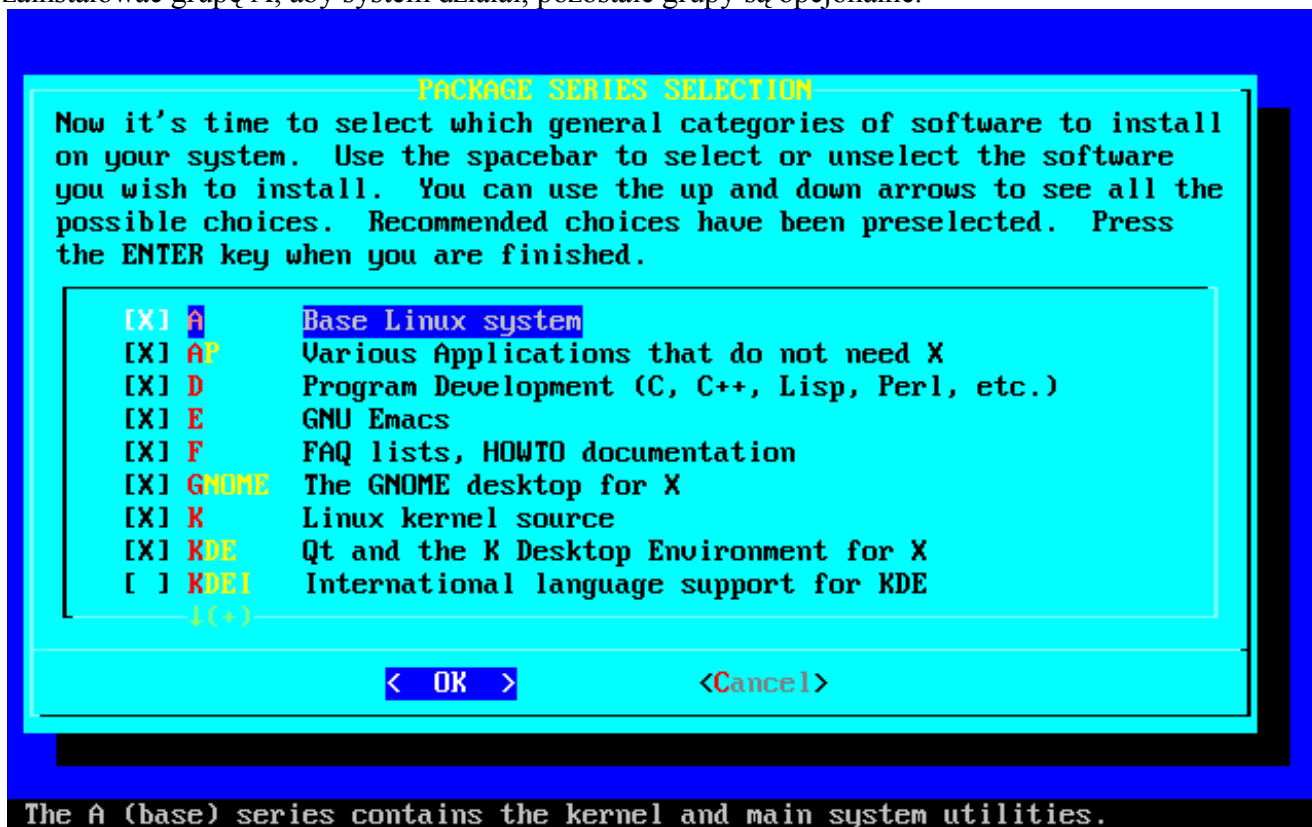
Wybór CD-ROM uruchamia instalację z płyty CD. Wyświetli się okno wyboru między automatycznym skanowaniem w poszukiwaniu napędu CD lub wybraniem go z listy znalezionych napędów. Upewnij się, że w napędzie znajduje się płyta ze Slackware, zanim uruchomisz skanowanie.

Przy wyborze NFS wyświetla się prośba o podanie informacji na temat sieci oraz serwera NFS. Serwer NFS musi zostać uprzednio skonfigurowany. Pamiętaj, że nie możesz używać nazw hostów, musisz używać numerycznych adresów IP, zarówno dla Twojego komputera, jak i serwera NFS (na dysku instalacyjnym nie ma programu tłumaczącego nazwy na adresy). Oczywiście najpierw musisz użyć dyskietki `network.dsk`, aby uruchomić kartę sieciową.

Najwięcej możliwości daje opcja uprzednio zamontowanego katalogu. Można jej użyć, aby zainstalować system z dysków Jaz, systemów NFS zamontowanych przez PLIP i dysków z systemem plików FAT. Zamontuj taki system w wybranej lokalizacji przed uruchomieniem **setup**, a potem podaj odpowiednią lokalizację.

## Wybór pakietów (SELECT)

W tej sekcji wybierasz grupy pakietów, które chcesz zainstalować. Są one opisane w sekcji 3.2.1. Musisz zainstalować grupę A, aby system działał, pozostałe grupy są opcjonalne.



Czas wybrać ogólne kategorie oprogramowania, które mają być zainstalowane. Użyj spacji, aby zaznaczyć lub odznaczyć wybrane oprogramowanie. Przewijając listę za pomocą strzałek w górę/w dół, możesz zobaczyć wszystkie dostępne opcje. Zalecane grupy są wstępnie zaznaczone. Po zakończeniu naciśnij ENTER.

A Podstawowy system Linux

AP Różne aplikacje, które do działania nie potrzebują systemu X

D Programowanie (C, C++, Lisp, Perl itd.)

E Edytor GNU Emacs

F Listy FAQ, dokumentacja HOWTO

GNOME Środowisko graficzne GNOME dla X

K Źródła jądra systemu Linux

KDE Biblioteka Qt i środowisko graficzne K Desktop Environment dla X

KDEI Obsługa różnych języków dla KDE

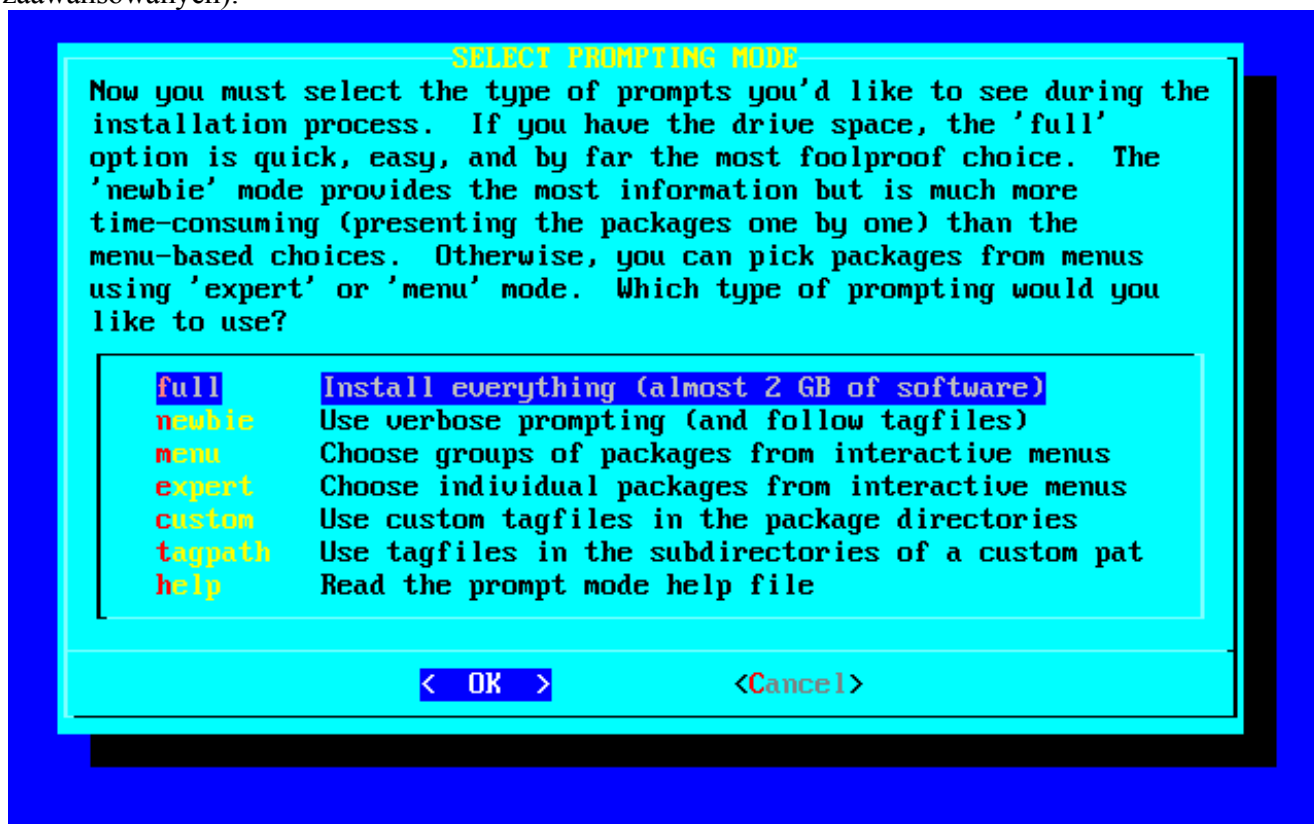
<OK>

<Anuluj>

Grupa A (podstawowa) zawiera jądro i główne programy systemowe.

### 3.4.7 Właściwa instalacja (INSTALL)

Po przejściu przez kroki „target”, „source” i „select” możesz w części *install* wybrać pakiety z zaznaczonych grup. Jeśli nie trzymałeś się proponowanej kolejności, wyświetli się monit o powrót do poprzednich sekcji. Tutaj możesz wybrać spośród sześciu metod instalacji: *full*, (pełnej), *newbie* (dla początkujących), *menu* (z pytaniem o poszczególne niewymagane pakiety), *expert*, *custom* i *tag path* (dla zaawansowanych).



Musisz wybrać typ podpowiedzi, które mają być wyświetlane podczas instalacji. Jeśli masz dużo miejsca na dysku, opcja „full” (pełna) jest najszybsza, najłatwiejsza i najbardziej idiotoodporna. Opcja „newbie” (dla początkujących) wyświetla najwięcej informacji, ale jest o wiele bardziej czasochłonna (wyświetla pakiety jeden po drugim) niż sposoby wybierania oparte na menu. Możesz też wybrać pakiety z menu za pośrednictwem trybów „expert” lub „menu”. Którego typu podpowiedzi przy instalacji chcesz użyć?

full (pełna) Instaluj wszystko (prawie 2 GB oprogramowania)

newbie (dla początkujących) Wyświetlaj szczegółowe informacje (i używaj plików znaczników)

menu Wybierz grupy pakietów z interaktywnych menu

expert (dla zaawansowanych) Wybierz poszczególne pakiety z interaktywnych menu

custom (użytkownika) Użyj własnych plików znaczników w katalogach z pakietami

tagpath (ścieżka do plików znaczników) Użyj plików znaczników w podkatalogach z własnoręcznie podaną ścieżką

help (pomoc) Przeczytaj plik pomocy dotyczący trybów podpowiedzi

<OK>

<Anuluj>



Opcja *full* zainstaluje wszystkie pakiety ze wszystkich grup wybranych w sekcji „select”. Nie będą wyświetlane dalsze monity. Jest to najłatwiejsza metoda instalacji, ponieważ nie musisz podejmować żadnych dodatkowych decyzji. Ta opcja zajmuje oczywiście najwięcej miejsca na dysku.

Opcja *newbie* instaluje wszystkie wymagane pakiety z wybranych grup. Dla pakietów opcjonalnych wyświetli się monit, w którym możesz wybrać „Yes” (Tak), „No” (Nie) lub „Skip” (pomiń). Tak i Nie są oczywiste. Pomiń spowoduje przejście do następnej grupy pakietów. Dodatkowo jest wyświetlany krótki opis każdego pakietu i ilość miejsca wymaganego na dysku, aby pomóc Ci w wyborze. Ta opcja jest zalecana dla początkujących użytkowników, ponieważ wymagane pakiety zostaną na pewno zainstalowane. Z powodu wyświetlanych monitów jest to jednak metoda dość czasochłonna.

*Menu* to szybsza i bardziej zaawansowana wersja opcji *newbie*. Dla każdej grupy jest wyświetlane menu, z którego możesz wybrać różne niewymagane pakiety. Pakiety wymagane nie są widoczne.

Opcja *expert* jest przeznaczona dla zaawansowanych użytkowników. Daje ona pełną kontrolę nad tym, które pakiety są instalowane. Możesz nawet odznaczyć pakiety bezwzględnie wymagane, co skutkuje niepoprawną instalacją. Z drugiej strony, w pełni decydujesz, co znajdzie się w systemie. Po prostu zaznacz pakiety z każdej grupy, które chcesz zainstalować. Ta opcja nie jest zalecana dla początkujących, ponieważ łatwo zrobić sobie kuku.

Opcje *custom* i *tag path* są również dla zaawansowanych. Dają one możliwość instalacji według plików znaczników (tag files). Takie pliki użytkownik tworzy sam w drzewie dystrybucji. Są użyteczne w przypadku szybkiej instalacji na wielu komputerach. Więcej informacji na temat plików znaczników znajduje się w sekcji 18.4.

Po wybraniu metody instalacji nastąpi jedna z rzeczy. Przy opcji *menu* pojawi się ekran z menu, pozwalając wybrać pakiety do zainstalowania. Jeśli wybrałeś *full*, natychmiast rozpocznie się instalacja pakietów na partycji docelowej. Jeśli wybrałeś *newbie*, automatycznie będą instalowane wymagane pakiety, aż instalacja dojdzie do pierwszego pakietu niewymaganego.

Istnieje możliwość, że podczas instalacji zabraknie miejsca na dysku. Jeśli wybrałeś zbyt wiele pakietów, a masz za mało miejsca na partycjach docelowych, masz problem. Najbezpieczniej jest wybierać tylko część pakietów, a pozostałe doinstalowywać w miarę potrzeb. Łatwo to zrobić, używając narzędzi do zarządzania pakietami Slackware. Więcej informacji na ten temat znajduje się w rozdziale 18.

### 3.4.8 Konfiguracja (CONFIGURE)

W tej sekcji dokonujesz podstawowej konfiguracji systemu po zainstalowaniu pakietów. To, co robisz, zależy w dużej mierze od tego, jakie pakiety zainstalowałeś. Na pewno jednak zobaczysz następujące ekrany:

#### 3.4.8.1 Wybór jądra (INSTALL LINUX KERNEL)

Wyświetli się pytanie, które jądro zainstalować. Możesz wybrać jądro z dyskietki startowej, z płyty CD ze Slackware lub z innej dyskietki, którą (zawsze myśl przyszłościowo) przygotowałeś. Możesz też pominąć ten krok, wybierając Skip. W takim przypadku użyte zostanie jądro domyślne i karawana pojedzie dalej.

### INSTALL LINUX KERNEL

In order for your system to boot correctly, a kernel must be installed. If you've made it this far using the installation bootdisk's kernel, you should probably install it as your system kernel (/boot/vmlinuz). If you're sure you know what you're doing, you can also install your choice of kernels from the Slackware CD, or a kernel from a floppy disk. You can also skip this menu, using whatever kernel has been installed already (such as a generic kernel from the A series). Which option would you like?

<b>bootdisk</b>	Use the kernel from the installation bootdisk
<b>cdrom</b>	Use a kernel from the Slackware CD
<b>floppy</b>	Install a zimage or bzimage from a DOS floppy
<b>skip</b>	Skip this menu (use the default /boot/vmlinuz)

< OK >

<Cancel>

Aby system mógł się uruchomić, musisz zainstalować jądro. Jeśli doszedłeś już tak daleko, używając jądra z płyty instalacyjnej, powinieneś zainstalować je jako jądro systemowe (/boot/vmlinuz). Jeśli wiesz, co robisz, możesz zainstalować inne jądro z płyty CD ze Slackware albo z dyskietki. Możesz też pominąć to menu i użyć już zainstalowanego jądra (np. standardowego jądra z grupy A). Której opcji chcesz użyć?

bootdisk (płyta rozruchowa) Użyj jądra z płyty instalacyjnej

cdrom Użyj jednego z jąder z płyty Slackware

floppy (dyskietka) Zainstaluj obraz jądra zimage lub bzimage ze zwykłej DOS-owej dyskietki

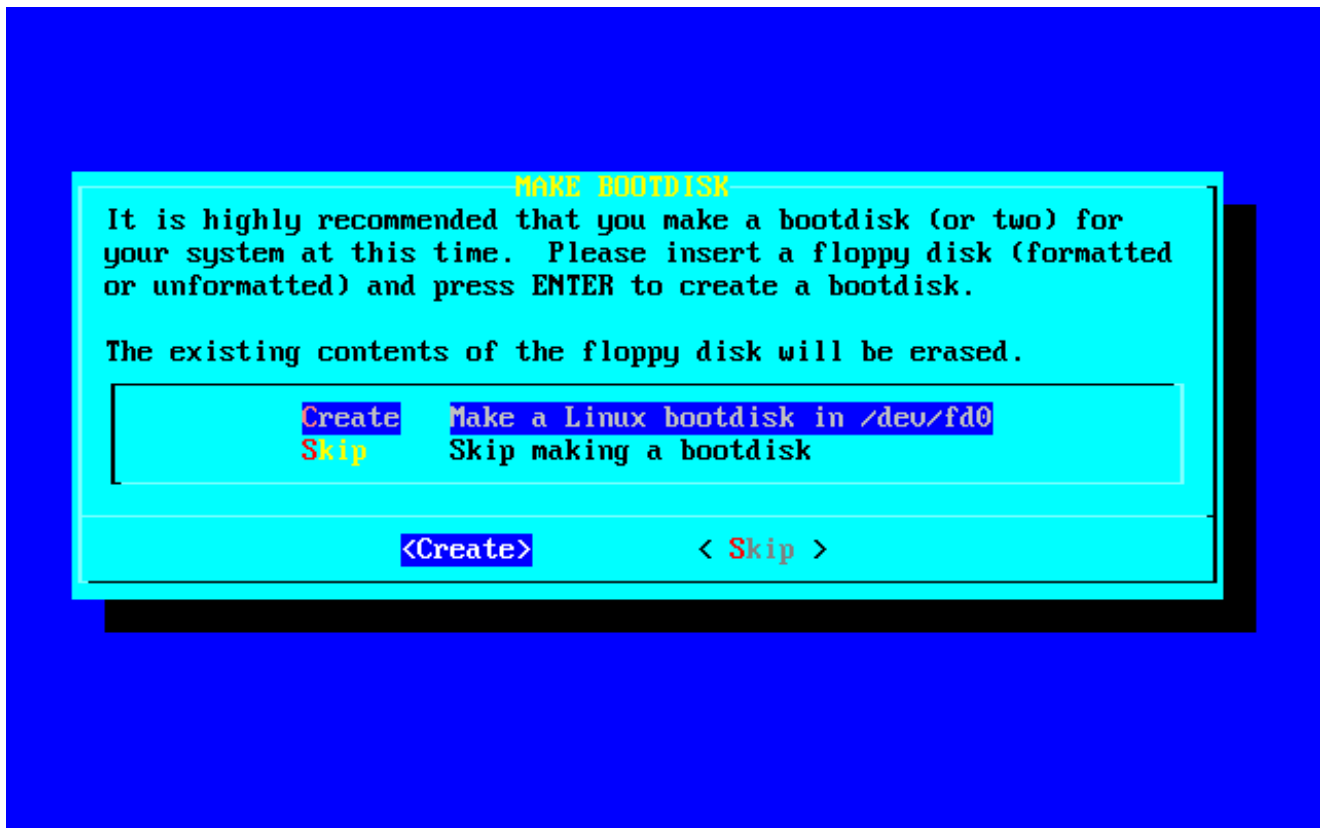
skip (pomiń) Pomiń to menu (użyj domyślnego jądra /boot/vmlinuz)

<OK>

<Anuluj>

### 3.4.8.2 Tworzenie dyskietki rozruchowej (MAKE BOOT DISK)

Utworzenie dyskietki rozruchowej do ewentualnego użycia jej w przyszłości to dobry pomysł. Masz tu możliwość sformatowania dyskietki i utworzenia jednego z dwóch rodzajów dyskietek rozruchowych. Pierwszy rodzaj, *simple* (prosty), jak sama nazwa wskazuje, po prostu zapisuje jądro na dyskietce. Drugi, dający więcej możliwości (i zalecany) sposób to *lilo*, który utworzy dyskietkę z programem rozruchowym *lilo*. Więcej informacji na temat *lilo* znajduje się w sekcji 7.1 – LILO. Możesz też pominąć tworzenie dyskietki, wybierając *continue* (kontynuuj). W takim przypadku dyskietka startowa nie zostanie utworzona.



Zalecane jest w tym momencie utworzenie dyskietki rozruchowej (albo dwóch) systemu. Włóż dyskietkę do stacji (sformatowaną lub nie) i wciśnij ENTER, aby utworzyć dyskietkę rozruchową. Istniejąca zawartość na dyskietce zostanie wymazana.	
Create (utwórz)	Utwórz linuksową dyskietkę rozruchową w /dev/fd0
Skip (Pomiń)	Pomiń tworzenie dyskietki rozruchowej
<Utwórz>	<Pomiń>

### 3.4.8.3 Modem (MODEM CONFIGURATION)

Wyświetli się okno z pytaniem o informacje na temat modemu. Mówiąc dokładniej, zostaniesz zapytany, czy masz modem, a jeśli tak, do którego portu szeregowego jest podłączony.

### MODEM CONFIGURATION

This part of the configuration process will create a `/dev/modem` link pointing to the callout device (`ttyS0`, `ttyS1`, `ttyS2`, `ttyS3`) representing your default modem. You can change this link later if you move your modem to a different port. If your modem is a PCI card, it will probably use `/dev/ttyS4` or higher. Please select the callout device which you would like to use for your modem:

<code>no modem</code>	<code>do not set a /dev/modem link</code>
<code>/dev/ttyS0</code>	(COM1: under DOS)
<code>/dev/ttyS1</code>	(COM2: under DOS)
<code>/dev/ttyS2</code>	(COM3: under DOS)
<code>/dev/ttyS3</code>	(COM4: under DOS)
<code>/dev/ttyS4</code>	PCI modem
<code>/dev/ttyS5</code>	PCI modem
<code>/dev/ttyS6</code>	PCI modem
<code>/dev/ttyS7</code>	PCI modem

< OK >

<Cancel>

W tej części procesu konfiguracji zostanie utworzone dowiązanie `/dev/modem`, wskazujące na urządzenie dzwoniące (`ttyS0`, `ttyS1`, `ttyS2`, `ttyS3`) reprezentujące modem domyślny. Możesz zmienić to dowiązanie później, jeśli podłączysz modem do innego portu. Jeśli modem jest na karcie PCI, prawdopodobnie będzie używał urządzenia `/dev/ttyS4` lub wyższego. Wybierz urządzenie dzwoniące, którego chcesz używać jako modemu:

no modem (brak modemu) nie twórz dowiązania `/dev/modem`

`/dev/ttyS0` (COM1: w systemie DOS)

`/dev/ttyS1` (COM2: w systemie DOS)

`/dev/ttyS2` (COM3: w systemie DOS)

`/dev/ttyS3` (COM4: w systemie DOS)

`/dev/ttyS4` modem PCI

`/dev/ttyS5` modem PCI

`/dev/ttyS6` modem PCI

`/dev/ttyS7` modem PCI

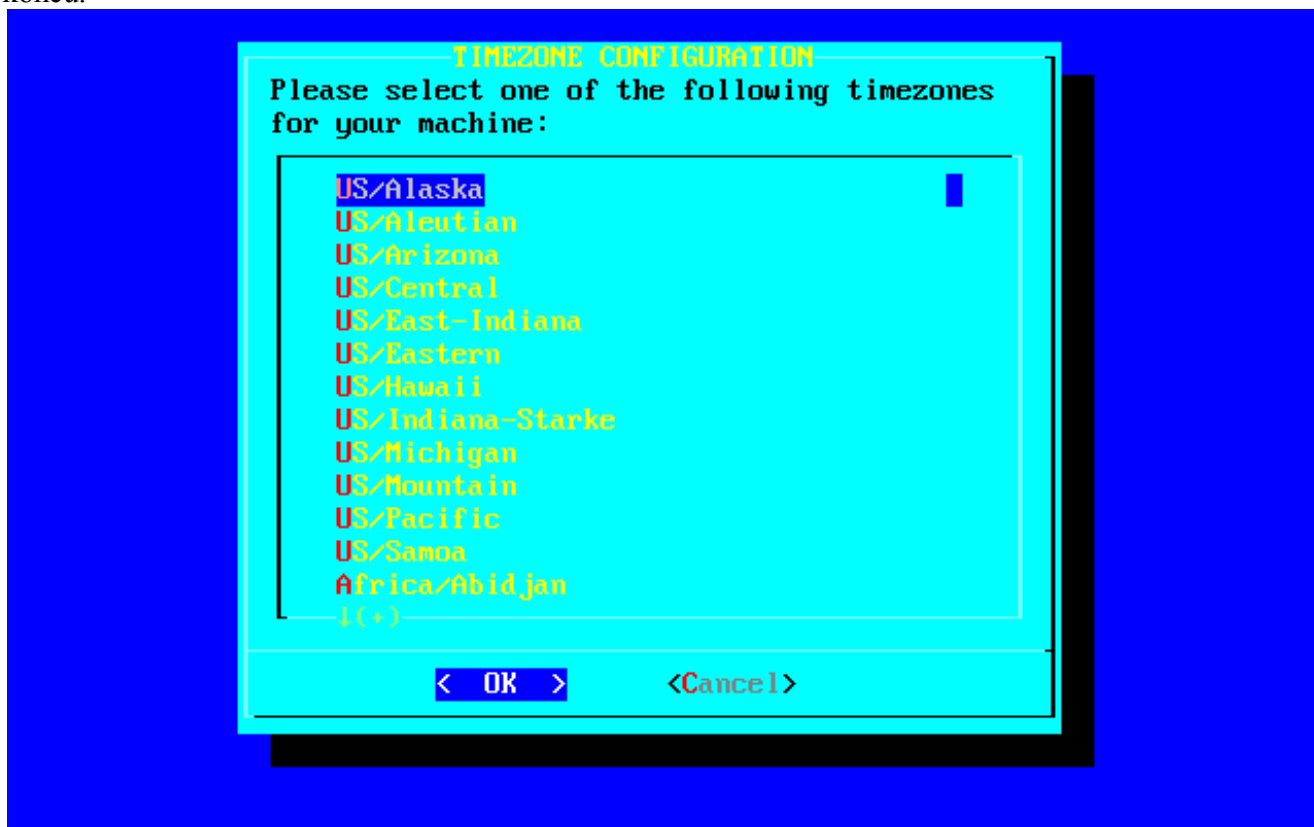
<OK>

<Anuluj>

Kolejne opcje konfiguracji mogą się pojawić lub nie, zależnie od tego, czy zainstalowałeś odpowiednie pakiety.

### 3.4.8.4 Strefa czasowa (TIMEZONE CONFIGURATION)

To dość proste: pojawi się pytanie, w której strefie czasowej się znajdujesz. Jeśli znajdujesz się w strefie Zulu, masz pecha: lista (baardzo długa) jest uporządkowana alfabetycznie, a strefa Zulu jest na samym końcu.



Wybierz jedną z poniższych stref czasowych dla komputera:

<OK>

<Anuluj>

### 3.4.8.5 Mysz (MOUSE CONFIGURATION)

Pojawia się pytanie, jaki rodzaj myszy posiadasz i czy chcesz, aby program `gpm(8)` (obsługa myszy w konsoli tekstowej) był uruchamiany przy starcie.

### MOUSE CONFIGURATION

This part of the configuration process will create a `/dev/mouse` link pointing to your default mouse device. You can change the `/dev/mouse` link later if the mouse doesn't work, or if you switch to a different type of pointing device. We will also use the information about the mouse to set the correct protocol for `gpm`, the Linux mouse server. Please select a mouse type from the list below:

<b>ps2</b>	PS/2 port mouse (most desktops and laptops)
<b>imps2</b>	Microsoft PS/2 Intellimouse
<b>bare</b>	2 button Microsoft compatible serial mouse
<b>ms</b>	3 button Microsoft compatible serial mouse
<b>mman</b>	Logitech serial MouseMan and similar devices
<b>msc</b>	MouseSystems serial (most 3 button serial mice)
<b>pnp</b>	Plug and Play (serial mice that do not work with ms)
<b>usb</b>	USB connected mouse
<b>↓(+)</b>	

< OK >

<Cancel>

W tej części procesu konfiguracji zostanie utworzone dowiązanie `/dev/mouse`, wskazujące na mysz domyślną. Możesz je później zmienić, jeśli mysz nie będzie działać, lub jeśli będziesz używał innego urządzenia wskazującego. Informacje o myszy zostaną też użyte, aby wybrać właściwy protokół dla linuxowego serwera myszy, `gpm`. Wybierz typ myszy z listy poniżej:

ps2 port myszy PS/2 (większość zwykłych komputerów i laptopów)

imps2 (Mysz Microsoft PS/2 Intellimouse)

bare 2-przyciskowa mysz szeregową zgodną z Microsoft

ms 3-przyciskowa mysz szeregową zgodną z Microsoft

mman szeregową mysz Logitech MouseMan i podobne

msc mysz szeregową MouseSystems (większość 3-przyciskowych myszy szeregowych)

pnp Plug and Play (myszy szeregowy, które nie działają z ms)

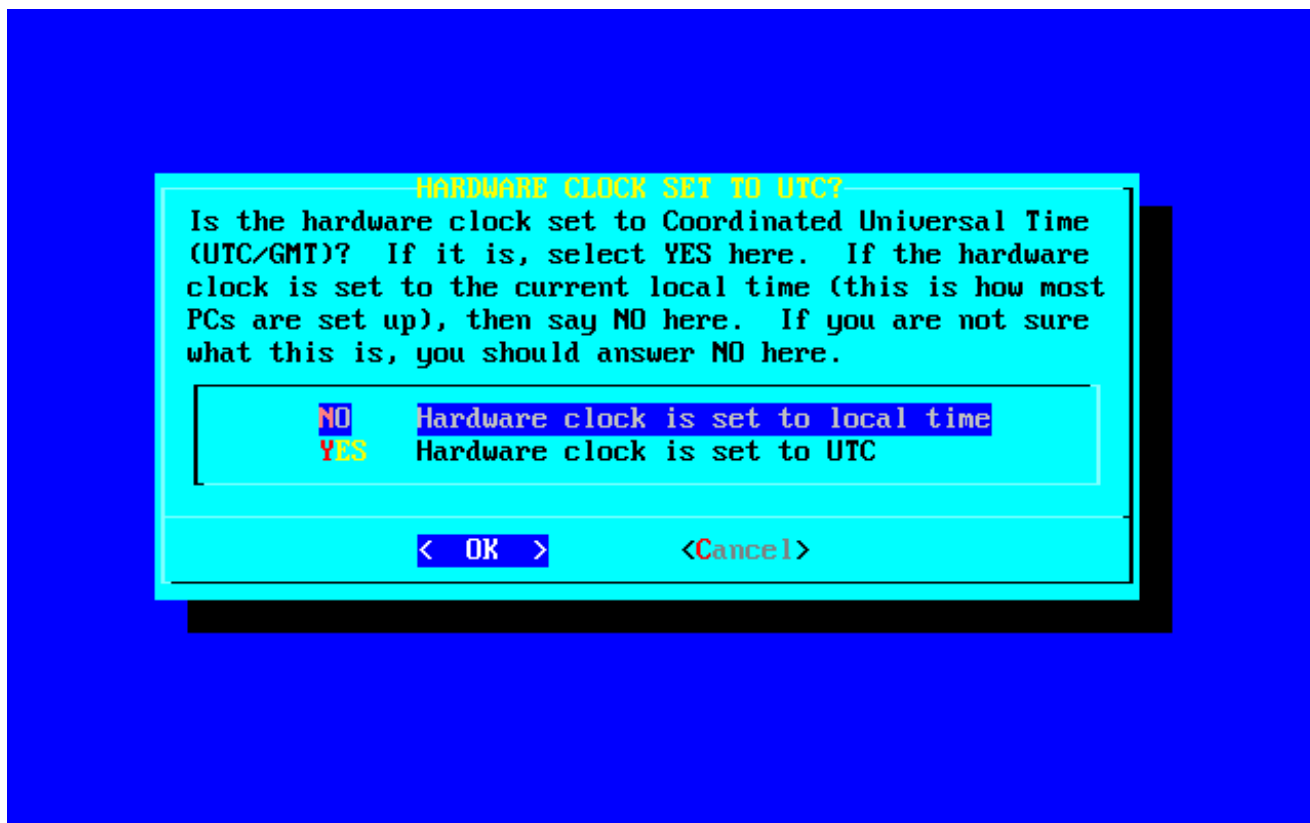
usb Mysz podłączona do portu USB

<OK>

<Anuluj>

### 3.4.8.6 Zegar sprzętowy (HARDWARE CLOCK SET TO UTC?)

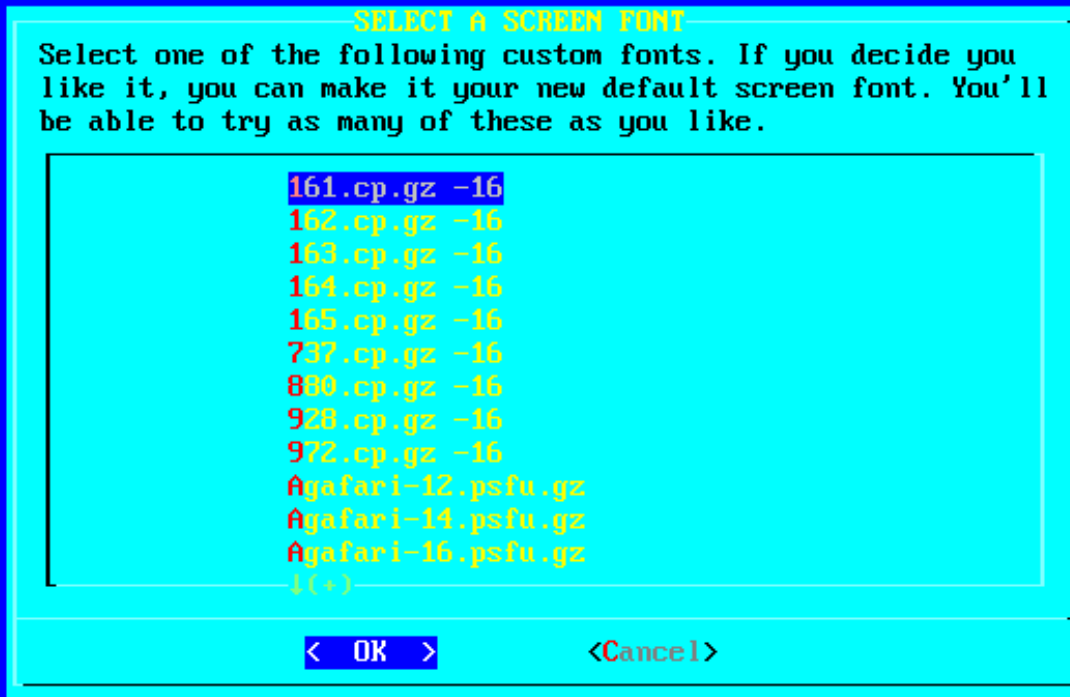
Ta sekcja pyta, czy wbudowany zegar komputera jest ustawiony według czasu uniwersalnego (Coordinated Universal Time – UTC lub GMT). Większość komputerów nie jest, więc z reguły należy wybrać *No* (Nie).



Czy zegar sprzętowy jest ustawiony według czasu uniwersalnego (UTC/GMT)? Jeśli tak, wybierz YES (tak). Jeśli nie jesteś pewien, co to jest, powinieneś wybrać NO (nie).	
NIE Zegar sprzętowy jest ustawiony według czasu lokalnego	
TAK Zegar sprzętowy jest ustawiony według czasu UTC	
<OK>	<Anuluj>

### 3.4.8.7 Czcionka ekranowa (SELECT A SCREEN FONT)

Tutaj możesz wybrać z listy czcionkę, która będzie używana w konsoli.



Wybierz jedną z następujących czcionek. Jeśli Ci się spodoba, możesz jej użyć jako nowej domyślnej czcionki ekranowej. Możesz wybierać tyle razy, ile chcesz.

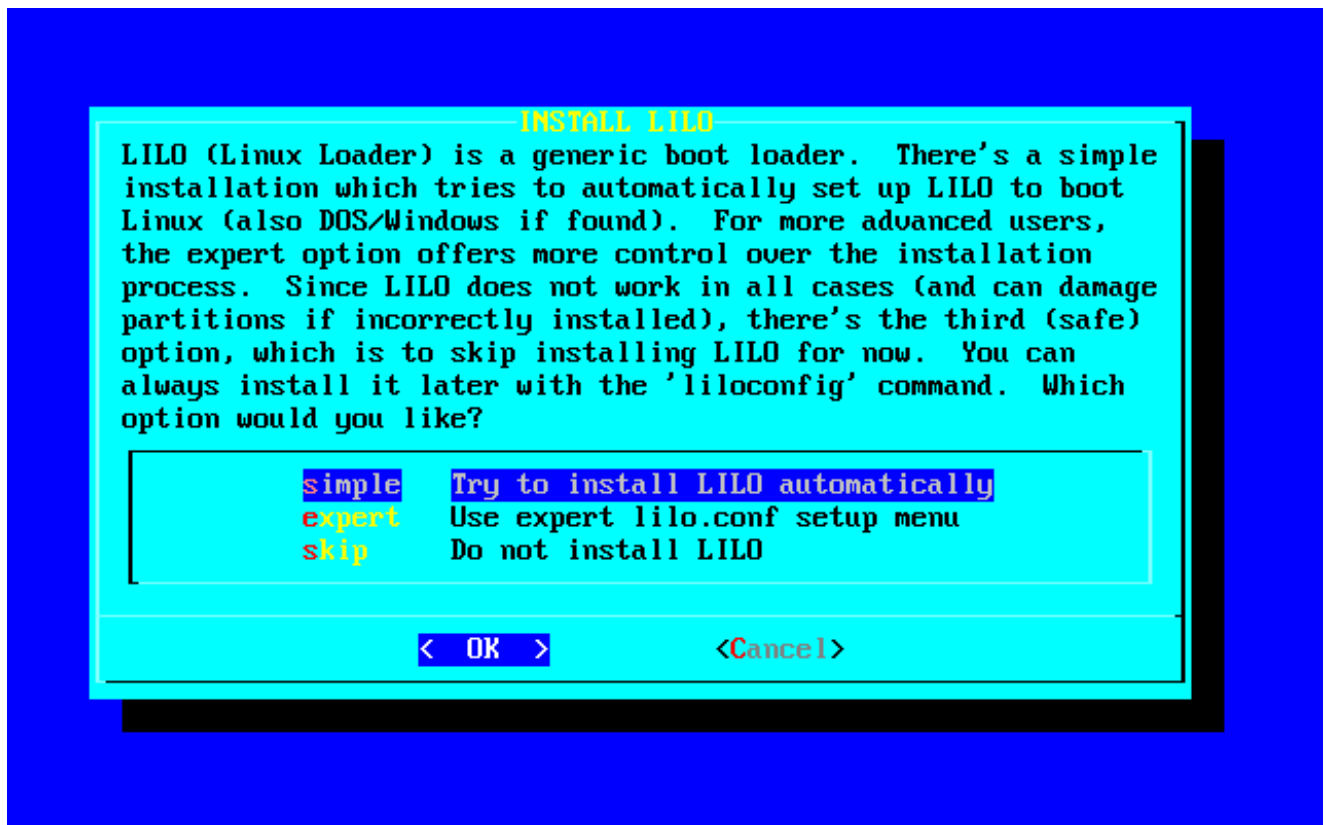
<OK>

<Anuluj>

## LILO (INSTALL LILO)

Tutaj możesz zdecydować, jak zainstalować program ładujący Linuxa podczas startu komputera – LILO (Linux LOader, więcej informacji na jego temat znajduje się w sekcji 7.1 – LILO).





LILO (Linux Loader) to standardowy program ładujący system. Do wyboru jest instalacja prosta, która próbuje automatycznie skonfigurować LILO, aby uruchamiał Linuksa (a także DOS/Windows, jeśli zostaną znalezione). Opcja dla zaawansowanych użytkowników umożliwia większą kontrolę nad procesem instalacji. Jako że LILO czasami nie działa (a w przypadku niewłaściwej instalacji może uszkodzić partycje), jest też trzecia, bezpieczna opcja, która polega na pominięciu instalacji LILO. Możesz go zawsze zainstalować później za pomocą polecenia „liloconfig”. Którą opcję chcesz wybrać?

simple (prosta) Spróbuj zainstalować LILO automatycznie	
expert (dla zaawansowanych) Użyj zaawansowanego menu konfiguracyjnego lilo.conf	
skip (pomiń) Nie instaluj LILO	
<OK>	<Anuluj>

Jeśli Slackware ma być jedynym systemem w komputerze, *simple* powinno wystarczyć. Jeśli chcesz mieć wybór między różnymi systemami, lepsze będzie *expert*. Więcej informacji na ten temat znajduje się w sekcji 7.3. Trzecia opcja, *do not install* (nie instaluj), nie jest zalecana, chyba że wiesz, co robisz i masz naprawdę dobry powód, aby nie instalować LILO. Jeśli używasz *expert*, będziesz miał wybór, gdzie zainstalować LILO. Możesz wybrać główny sektor rozruchowy (MBR – Master Boot Record) dysku twardego, superblok głównej partycji linuksowej lub dyskietkę.

### 3.4.8.9 Konfiguracja sieci (Network)

Konfiguracja sieci jest przeprowadzana przez program `netconfig`. Więcej informacji na ten temat znajduje się w sekcji 5.1.

### 3.4.8.10 Menedżer okien (SELECT DEFAULT WINDOW MANAGER FOR X)

Tutaj wybierasz domyślny menedżer okien dla systemu graficznego X. Więcej informacji na ten temat znajduje się w rozdziale 6.



Wybierz domyślny menedżer okien, używany w systemie X Window. Będzie on określał styl graficznego interfejsu użytkownika komputera. KDE i GNOME mają największe możliwości. Będzie się z nich łatwo korzystało ludziom przyzwyczajonym do Windows lub MacOS. Inne menedżery zużywają mniej zasobów systemowych lub zapewniają inne wyjątkowe funkcje.

xinitrc.kde KDE: środowisko K Desktop Environment

xinitrc.gnome środowisko GNU Network Object Model Environment

xinitrc.xfce łatwe, lekkie i przyjemne środowisko graficzne

xinitrc.blackbox menedżer okien blackbox

xinitrc.fluxbox menedżer okien fluxbox

xinitrc.wmaker WindowMaker

xinitrc.fvwm2 Wirtualny Menedżer Okien (F cośtam) (wersja 2.xx)

xinitrc.fvwm95 FVWM2 podobny z wyglądu i zachowania do Windows

xinitrc.twm menedżer okien Tab (bardzo prosty)

<OK>

<Anuluj>

Ostatnią rzeczą, niezależną od zainstalowanych pakietów, o jaką zostaniesz zapytany, to czy chcesz

ustanowić hasło administratora (`root`). Ze względów bezpieczeństwa jest to zalecane. Oczywiście, jak prawie wszystko w Slackware, zależy to tylko od Ciebie.

## Rozdział 4. Konfiguracja systemu

Zanim rozpoczniesz zaawansowaną konfigurację systemu, powinieneś nauczyć się, jak jest on zorganizowany, jakich poleceń używać do znajdowania plików i programów. Powinieneś też zorientować się, czy potrzebujesz skompilować jądro, i wiedzieć, jak to zrobić. Ten rozdział zaznajomi Cię z organizacją systemu i plikami konfiguracyjnymi. Po przeczytaniu będziesz mógł zająć się konfiguracją na bardziej zaawansowanym poziomie.

### 4.1 Przegląd systemu

Musisz dowiedzieć się, jak jest skonstruowany system linuksowy, zanim zagłębisz się w poszczególne aspekty konfiguracji. Linux bardzo różni się od systemów DOS, Windows czy Macintosh (z wyjątkiem opartego na Uniksie MacOS X). Dzięki temu rozdziałowi zaznajomisz się z układem Linuksa, co ułatwi Ci dostosowanie systemu do własnych potrzeb.

#### 4.1.1 Układ systemu plików

Pierwszą ważną różnicą między Linuksem Slackware a DOS czy Windows jest system plików. Przede wszystkim, nie używa się liter do oznaczania partycji. W Linuksie jest jeden główny katalog. Stanowi on mniej więcej odpowiednik dysku C: w DOSie. Każda partycja jest montowana w podkatalogu należącym do tego głównego. To coś jak dysk twardy, który może zwiększać się w nieskończoność.

Katalog główny nazywa się katalogiem `root` (ang. korzeń) i jest oznaczany pojedynczym ukośnikiem (`/`). Takie rozwiązanie może się wydawać dziwne, ale w istocie bardzo ułatwia życie, kiedy chce się dodać więcej miejsca na dysku. Załóżmy np., że skończyło się miejsce, na którym znajdowała się partycja `/home`. Wielu użytkowników instaluje Slackware, tworząc jeden duży dysk `root`. Jako że partycję można zamontować w każdym katalogu, wystarczy po prostu kupić nowy dysk i zamontować go pod `/home`. W ten sposób „wszczepia się” miejsce do systemu, i to bez zbędnego zachodu.

Oto opis najważniejszych katalogów głównego poziomu w Slackware:

`bin`

Miejsce na podstawowe programy potrzebne do korzystania z systemu. Znajduje się tu powłoka (shell), polecenia do poruszania się w systemie plików (`ls`, `cp` itp.). Katalogu `/bin` z reguły nie modyfikuje się po instalacji. Jeśli tak się dzieje, to raczej w wyniku instalowania zaktualizowanych pakietów.

`boot`

Miejsce na pliki używane przez program rozruchowy LILO. Tego katalogu również z zasady nie modyfikuje się po instalacji. Od wersji Slackware 8.1 mieści się tu jądro. We wcześniejszych wersjach znajdowało się po prostu w `/`, ale powszechną praktyką jest umieszczanie jądra i potrzebnych plików w `/boot`, co daje możliwość uruchamiania wielu systemów.

`dev`

W Linuksie wszystko jest plikiem, nawet urządzenia takie, jak porty szeregowo, dyski twarde i skanery. Aby były one dostępne, istnieją specjalne typy plików – pliki urządzeń (device nodes). Wszystkie znajdują się w katalogu `/dev`. Tak samo jest w wielu innych uniksopodobnych systemach.

`etc`

Tu znajdują się pliki konfiguracyjne systemu, od X Window, przez bazę danych użytkowników, aż po skrypty startowe. Z czasem administrator zaznajomi się z tymi plikami.

`home`

Linux jest systemem wieloużytkownikowym. Każdy użytkownik w systemie ma swoje konto i osobny katalog na prywatne pliki. Nazywa się on katalogiem *domowym* lub *macierzystym* (*home*). Domyślnie katalogi użytkowników znajdują się w `/home`.

`lib`

Biblioteki potrzebne do podstawowego działania systemu, m.in. biblioteka C, ncurses, dynamiczny program ładujący i moduły jądra.

`mnt`

Tutaj są umieszczone tymczasowe punkty montowania dysków twardech i innych mediów, takich jak np. cdromy i dyskietki.

`opt`

Pakiety opcjonalne. Ideą tego katalogu jest instalowanie w nim programów w podkatalogu `/opt/nazwaprogramu`, tak aby łatwiej je było usuwać. Slackware instaluje w `/opt` kilka pakietów (m.in. KDE w `/opt/kde`), ale można tam instalować, co się chce.

`proc`

Katalog szczególny. Nie jest to część faktycznego systemu plików, ale system wirtualny, który pozwala na dostęp do informacji przekazywanych przez jądro. Większość tych informacji jest przekazywana za pośrednictwem „plików” w `/proc`. Niektóre z nich służą także do przesyłania pewnych informacji do jądra. Wykonaj `cat /proc/cpuinfo` i zobacz, co się wyświetli.

`root`

Administrator systemu nazywa się `root`. Katalog domowy `roota` to `/root`, a nie `/home/root`. Powód jest prosty. Co by się stało, gdyby `/home` i `/` były na oddzielnych partycjach, a `/home` nie dałoby się zamontować? `Root` chciałby się zalogować, aby naprawić problem, ale nie mógłby tego wykonać, gdyby jego katalog domowy był na uszkodzonym systemie plików.

`sbin`

Podstawowe programy używane podczas uruchamiania systemu oraz przez `roota`. Zwykli użytkownicy nie korzystają z nich.

`tmp`

Miejsce na zapisywanie tymczasowych danych. Wszyscy użytkownicy mogą zapisywać w tym katalogu i odczytywać z niego.

`usr`

Ten katalog zajmuje najwięcej miejsca w Linuksie. Znajduje się w nim większość pozostałych danych: programy, dokumentacja, kody źródłowe jądra i system X Window. Programy instaluje się najczęściej właśnie w tym katalogu.

`var`

Miejsce na pliki dzienników zdarzeń (log files), dane podręczne (cache data), pliki blokady programów (lock files) i inne dane, które są często modyfikowane.

Teraz wiesz z grubsza, jakie główne katalogi istnieją w systemie i co zawierają. W następnej sekcji nauczysz się, jak łatwo znajdować potrzebne Ci pliki, żebyś nie musiał robić tego ręcznie.

## 4.1.2 Znajdowanie plików

Znasz już podstawową zawartość katalogów. Nie pomaga Ci to jednak w wyszukiwaniu tego, czego potrzebujesz. Mógłbyś oczywiście przeczesać katalog po katalogu, ale są łatwiejsze sposoby. W Slackware istnieją cztery podstawowe polecenia do wyszukiwania plików.

### 4.1.2.1 `which`

Pierwsze z nich to `which(1)`. Używa się go do szybkiego zlokalizowania programu. Przeszukuje ono po prostu ścieżkę systemową (`PATH`), zwraca pierwszy poszukiwany program i ścieżkę do niego. Oto przykład:

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

Jak widać, `bash` znajduje się w katalogu `/bin`. To polecenie jest dość ograniczone, ponieważ przeszukuje tylko katalogi wymienione w `PATH`.

### 4.1.2.2 `whereis`

Polecenie `whereis(1)` działa podobnie do `which`, ale przeszukuje również strony podręcznika i pliki źródłowe. Oto wynik poszukiwania `whereis` dla `bash`:

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

Poznałeś lokalizację nie tylko programu, ale też dokumentacji. To polecenie jednak też jest ograniczone.

Gdybyś chciał znaleźć konkretny plik konfiguracyjny, `which` lub `whereis` nie wystarczyłyby.

#### 4.1.2.3 `find`

Polecenie `find(1)` umożliwia przeszukiwanie systemu według szerokich kryteriów. Możesz używać symboli wieloznacznych w nazwach plików, zakresów czasu modyfikacji lub tworzenia, albo innych właściwości. Na przykład, chcesz przeszukać cały system, aby znaleźć domyślny plik `xinitrc`:

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

Wykonanie tego polecenia trwa długo, ponieważ musi przeczesać całe drzewo katalogów. Jeśli wykonasz je jako zwykły użytkownik, otrzymasz komunikaty o braku dostępu do katalogów dostępnych tylko dla `roota`. Jednak `find` znalazło Twój plik, czyli zadziałało. Gdyby tylko działało szybciej...

#### 4.1.2.4 `slocate`

Polecenie `slocate(1)` przeszukuje cały system, podobnie do `find`. Jednak, zamiast przechodzenia przez wszystkie pliki i katalogi, przeszukuje specyficzną bazę danych. Baza danych jest aktualizowana codziennie rano, dlatego lista plików jest aktualna. Możesz zaktualizować bazę `slocate` ręcznie, używając polecenia `updatedb(1)` (przed wykonaniem `updatedb` musisz tymczasowo zalogować się jako `root` za pomocą polecenia `su`). Oto przykład `slocate` w działaniu:

```
% slocate xinitrc # nie musimy się logować jako root  
/var/X11R6/lib/xinit/xinitrc  
/var/X11R6/lib/xinit/xinitrc.fvwm2  
/var/X11R6/lib/xinit/xinitrc.openwin  
/var/X11R6/lib/xinit/xinitrc.twm
```

Otrzymaliśmy więcej niż szukaliśmy, i to szybko. Używając wymienionych czterech poleceń, powinieneś być w stanie znaleźć wszystko, czego potrzebujesz.

### 4.1.3 Katalog `/etc/rc.d`

W tym katalogu mieszczą się pliki inicjalizacyjne, używane podczas uruchamiania systemu. Slackware korzysta z plików skonstruowanych w stylu BSD, w przeciwieństwie do skryptów typu System V, w których przypadku trudniej zmienić konfigurację bez stosowania specjalnych programów. W skryptach BSD każdy poziom pracy (`runlevel`) ma odrębny plik `rc`. W System V dla każdego poziomu pracy jest przeznaczony odrębny katalog, w każdym z nich znajdują się różne skrypty. Taka struktura jest łatwa w zarządzaniu.

Pliki inicjalizacyjne dzielą się na kilka kategorii. Są to skrypty: uruchomieniowe (`startup`), poziomów pracy (`runlevels`), inicjalizacji sieci i zgodności z System V. Wszystkie pozostałe tradycyjnie wrzucimy do jednego worka o nazwie „inne”.

#### 4.1.3.1 Skrypty uruchomieniowe

Pierwszym programem uruchamianym przez Slackware, oprócz jądra, jest `init(8)`. Odczytuje on plik `/etc/inittab(5)` aby zobaczyć, w jaki sposób uruchomić system. Następnie wykonuje skrypt /

`etc/rc.d/rc.S`, który przygotowuje system do uruchomienia na określonym poziomie pracy. Skrypt `rc.S` uaktywnia pamięć wirtualną, montuje systemy plików, czyści niektóre pliki dzienników, inicjalizuje urządzenia Plug and Play, ładuje moduły jądra, konfiguruje urządzenia PCMCIA i porty szeregowy oraz uruchamia skrypty System V (jeśli je znajdzie). Ten skrypt jest dość rozbudowany, ale w `/etc/rc.d` znajdują się inne skrypty, do których `rc.S` się odwołuje:

`rc.S`

Właściwy skrypt inicjalizujący system.

`rc.modules`

Ładuje moduły jądra, takie jak sterownik karty sieciowej, obsługę PPP i inne. Uruchamia też inny skrypt, `rc.netdevice`, jeśli takowy istnieje.

`rc.pcmcia`

Wyszukuje i konfiguruje urządzenia PCMCIA. Ma zastosowanie głównie w laptopach, które są z reguły wyposażone w modem lub kartę sieciową w tej technologii.

`rc.serial`

Konfiguruje porty szeregowy, uruchamiając odpowiednie polecenia `setserial`.

`rc.sysvinit`

Wyszukuje skrypty System V dla danych poziomów pracy i wykonuje je. Jest to bardziej szczegółowo opisane poniżej.

#### **4.1.3.2 Skrypty inicjalizacyjne poziomów pracy**

Po inicjalizacji samego systemu `init` przechodzi do inicjalizacji poziomów pracy. Poziom pracy oznacza stan, w jakim komputer będzie pracować. Brzmi jak masło maślane? Poziom pracy mówi programowi `init`, czy system ma akceptować logowania wielu użytkowników, czy tylko jednego, czy mają być uruchamiane usługi sieciowe, czy do obsługi logowań będzie używany system X Window, czy program `agetty`(8). Oto opis plików definiujących poszczególne poziomy pracy w Linuksie Slackware:

`rc.0`

Zatrzymanie systemu (poziom pracy 0). Domyślnie jest to dowiązanie symboliczne do `rc.6`.

`rc.4`

Uruchomienie w trybie wieloużytkownikowym (poziom pracy 4), ale z KDM, GDM lub XDM jako menedżerem logowania.

`rc.6`

Zatrzymanie i ponowne uruchomienie systemu (poziom pracy 6).

`rc.K`

Uruchomienie w trybie jednoużytkownikowym (poziom pracy 1).

`rc.M`

Uruchomienie w trybie wieloużytkownikowym (poziomy pracy 2 i 3), ale ze standardowym logowaniem w trybie tekstowym. Jest to domyślny poziom pracy w Slackware.

#### **4.1.3.3 Inicjalizacja sieci**

Poziomy pracy 2, 3 i 4 uruchamiają usługi sieciowe. Oto odpowiedzialne za to pliki:

`rc.inet1`

Jest on tworzony przez program `netconfig` i odpowiada za konfigurację interfejsu sieciowego.

`rc.inet2`

Jest wykonywany po `rc.inet1` i uruchamia usługi sieciowe.

`rc.atalk`

Uruchamia usługi AppleTalk.

`rc.httpd`

Uruchamia serwer stron WWW Apache. Podobnie do niektórych innych skryptów `rc`, można go również używać do zatrzymywania lub restartowania usługi. Przyjmuje on argumenty `start`, `stop` i `restart`.

`rc.news`

Uruchamia serwer grup dyskusyjnych.

#### **4.1.3.4 Zgodność z System V**

Zgodność z System V wprowadzono w Slackware 7.0. Wiele innych dystrybucji Linuksa korzysta z niego zamiast ze stylu BSD. Zasadniczo w tym trybie każdy poziom pracy ma swój podkatalog z odrębnymi skryptami, podczas gdy w BSD każdy poziom ma jeden skrypt.

Plik `rc.sysvinit` wyszukuje skrypty System V w katalogu `/etc/rc.d` i wykonuje je, jeśli należą do odpowiedniego poziomu pracy. Jest to przydatne w przypadku niektórych komercyjnych programów, które instalują pliki typu System V.

#### **4.1.3.5 Inne pliki**

Oto inne skrypty inicjalizacyjne. Z reguły są wykonywane przez jeden z głównych skryptów wymienionych powyżej, więc wystarczy wyedytować ich treść.



`rc.gpm`

Uruchamia ogólną obsługę myszy, która umożliwia kopiowanie i wklejanie w konsoli linuksowej. Czasami program `gpm` sprawia problemy, jeśli używa się go w X Window. Jeśli w trybie graficznym masz problem z obsługą myszy, usuń prawo wykonania dla tego skryptu i zatrzymaj serwer `gpm`.

`rc.font`

Ładuje wybraną przez użytkownika czcionkę ekranową dla konsoli.

`rc.local`

Zawiera polecenia specyficzne dla danego komputera. Od razu po zainstalowaniu jest pusty, ponieważ jest przeznaczony dla lokalnego administratora. Jest wykonywany na końcu, po wszystkich innych skryptach inicjalizacyjnych.

Aby umożliwić wykonanie skryptu, wystarczy nadać mu prawa wykonania poleceniem `chmod`. Aby wyłączyć taką możliwość, trzeba usunąć prawa wykonania. Więcej informacji na temat `chmod` znajduje się w sekcji 9.2.

## 4.2 Wybór jądra

Jądro (ang. kernel) jest częścią systemu operacyjnego, która odpowiada za dostęp do urządzeń, kontroluje procesy i cały system. Obsługuje ono urządzenia sprzętowe, dlatego wybór właściwego jądra jest ważnym krokiem podczas konfiguracji.

W Linuksie Slackware jest dostępnych kilkanaście prekompilowanych jąder do wyboru. Każde z nich ma standardowy zestaw sterowników oraz sterowniki dodatkowe. Możesz używać jądra prekompilowanego bądź skompilować własne ze źródeł. Tak czy owak, musisz być pewien, że w jądrze znajdują się sterowniki sprzętu, jaki posiadasz.

### 4.2.1 Katalog `/kernel`s na CD-ROMie Slackware

Prekompilowane jądra Slackware znajdują się w katalogu `/kernel`s na płycie CD lub w repozytorium FTP w głównym katalogu Slackware. Lista dostępnych jąder zmienia się z wersji na wersję, dlatego właściwym źródłem dokumentacji na ten temat są zawsze odpowiednie pliki w tym katalogu. W `/kernel`s znajdują się podkatalogi dla poszczególnych jąder. Mają taką samą nazwę jak odpowiednia dyskietka rozruchowa. W każdym podkatalogu są następujące pliki:

Plik	Jego przeznaczenie
<code>System.map</code>	mapa systemu dla danego jądra
<code>bzImage</code>	obraz jądra
<code>config</code>	informacja na temat konfiguracji ze źródeł dla danego jądra

Aby użyć jądra, wystarczy skopiować `System.map` i `config` do katalogu `/boot`, a obraz jądra skopiować do pliku `/boot/vmlinuz`. Następnie należy uruchomić `/sbin/lilo(8)`, aby zainstalować LILO dla nowego jądra i uruchomić ponownie system. To cała procedura instalacji nowego

jądra.

Jądra, których nazwy kończą się na „. i”, to jądra IDE. To znaczy, że nie mają obsługi SCSI. Jądra, których nazwy kończą się na „. s”, to jądra SCSI. Zawierają obsługę IDE z jąder . i oraz dodatkowo SCSI.

## 4.2.2 Kompilacja jądra ze źródeł

Nowi użytkownicy Linuksa często zadają pytanie „Czy muszę kompilować jądro”. Odpowiedź brzmi: „to zależy”. Czasami trzeba. W większości przypadków do poprawnego działania systemu wystarczy użyć prekompilowanego jądra i dynamicznych modułów. Rekompilacja jest potrzebna, jeśli chcesz zainstalować zaktualizowaną wersję jądra, której jeszcze nie udostępniamy, albo jeśli zmodyfikowałeś źródła jądra, aby dodać obsługę jakiegoś nietypowego urządzenia nieobecnej w jądrach standardowych. Z pewnością jest potrzebna dla użytkowników komputerów wieloprocessorowych (SMP). Jądra kompilowane własnoręcznie dla danej maszyny z reguły działają też o wiele szybciej. Warto skompilować jądro zoptymalizowane dla danego typu procesora.

Tworzenie własnego jądra nie jest zbyt trudne. Po pierwsze, musisz upewnić się, że masz zainstalowane źródła jądra. Sprawdź, czy zainstalowałeś pakiety z grupy K, a także D – szczególnie kompilator C, GNU make i GNU binutils. Najlepiej zainstalować wszystkie pakiety z D, jeśli zamierzasz tworzyć i kompilować oprogramowanie. Źródła najnowszej wersji jądra możesz też pobrać z <http://www.kernel.org/mirrors>.

### 4.2.2.1 Kompilacja jądra Linuksa w wersji 2.4.x

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

Najpierw należy doprowadzić źródła do stanu podstawowego. Robi się to za pomocą polecenia `make mrproper` (wykonaj kopię zapasową pliku `.config`, ponieważ to polecenie usunie go bez pytania).

```
# make mrproper
```

Teraz skonfiguruj jądro dla swojego systemu. W obecnie dostępnych jądrach są 3 opcje konfiguracji. Pierwsza to oryginalny system tekstowy oparty na pytaniach i odpowiedziach. Zadaje on wiele pytań i na podstawie odpowiedzi tworzy plik konfiguracyjny. Jego wadą jest to, że jeśli udzieliś jakiejś złej odpowiedzi, musisz zacząć od początku. Druga, najpowszechniej wybierana opcja, to system oparty na menu. Trzecia opcja to graficzne narzędzie konfiguracyjne. Wybierz tę, która Ci najbardziej odpowiada i wpisz jedno z poleceń:

```
# make config          (tekstowy system pytań i odpowiedzi)  
# make menuconfig     (tekstowy system wykorzystujący menu)  
# make xconfig        (system graficzny, musisz mieć uruchomiony X Window)
```

Rys. 4-1. Program konfiguracyjny jądra oparty na menu

## Linux Kernel v2.6.11.6 Configuration

```
Linux Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

Code maturity level options --->
General setup --->
Loadable module support --->
Processor type and features --->
Power management options (ACPI, APM) --->
Bus options (PCI, PCMCIA, EISA, MCA, ISA) --->
Executable file formats --->
Device Drivers --->
File systems --->
Profiling support --->
Kernel hacking --->

[+]

<Select> < Exit > < Help >
```

Do nawigacji w menu służą klawisze strzałek. <Enter> wybiera podmenu --->. Podświetlone litery to klawisze skrótu. Wciśnięcie <Y> dołącza funkcje, <N> wyłącza je, <M> powoduje, że będą skompilowane jako moduły. Wciśnij <Esc><Esc>, aby wyjść, <?>, aby uzyskać pomoc, </>, aby wyszukiwać. Legenda: [\*] wbudowana [ ] wyłączona [M] moduł < >

Opcje poziomu dopracowania kodu --->

Konfiguracja ogólna --->

Obsługa modułów dynamicznych --->

Rodzaj i funkcje procesora --->

Opcje zarządzania energią (ACPI, APM) --->

Opcje szyny (PCI, PCMCIA, EISA, MCA, ISA) --->

Formaty plików wykonywalnych --->

Sterowniki urządzeń --->

Systemy plików --->

Obsługa profili --->

Praca nad jądrem --->

<Wybierz>

<Wyjdź>

<Pomoc>

Dla nowicjuszy prawdopodobnie `menuconfig` będzie najwygodniejszy. Poszczególne opcje są objaśnione w ekranach pomocy. Po skonfigurowaniu jądra zakończ program, który zapisze odpowiednie pliki konfiguracyjne. Potem przygotuj drzewo źródeł do utworzenia jądra:

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

Następnie skompiluj jądro poleceniem `make bzImage`:

```
# make bzImage
```

W zależności od szybkości procesora wykonanie może trochę potrwać. Podczas kompilacji na ekranie wyświetlają się komunikaty kompilatora. Po utworzeniu obrazu jądra zbuduj części, które oznaczyłeś jako odrębne moduły:

```
# make modules
```

Teraz zainstaluj jądro i moduły. Oto odpowiednie polecenia:

```
# 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
```

Wyedytuj `/etc/lilo.conf`, aby dodać sekcję ładującą stare jądro, w razie gdyby nowe nie działało. Potem wykonaj `/sbin/lilo`, co zainstaluje nowy sektor rozruchowy. Na koniec uruchom system z nowym jądrem.

#### 4.2.2.2 Kompilacja jądra Linuksa w wersji 2.6.x

Kompilacja jądra serii 2.6 nie różni się wiele od 2.4 czy 2.2, musisz poznać jednak różnice, zanim zaczniesz kompilację. Niepotrzebne są kroki `make dep` i `make clean`, komunikaty wyświetlane podczas kompilacji nie są tak szczegółowe. Dzięki temu proces budowania jest prostszy, ale ma też ograniczenia. Jeśli napotkasz jakieś kłopoty, zaleca się włączenie szczegółowych komunikatów. Wystarczy w tym celu dodać `V=1` do polecenia kompilacji. Dzięki temu uzyskasz więcej informacji, które pomogą deweloperowi jądra lub jakiemuś pomocnemu linuxsiarzowi w rozwiązaniu problemu.

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

### 4.2.3 Moduły jądra

„Moduły jądra” to inna nazwa „sterowników urządzeń”, które można dodawać do jądra podczas jego działania. Pozwala to na dodawanie obsługi nowych urządzeń bez potrzeby wyboru innego jądra lub jego rekompilacji.

Moduły można łączyć i usuwać w każdej chwili, nawet w trakcie pracy systemu. Dzięki temu administratorzy mogą łatwiej aktualizować poszczególne sterowniki. Można skompilować nowy moduł i załadować go od razu po usunięciu starego, bez potrzeby ponownego uruchamiania komputera.

Moduły znajdują się w katalogu `/lib/modules/wersja_jądra`. Można je łączyć podczas startu systemu za pośrednictwem skryptu `rc.modules`. Jest w nim wiele szczegółowych komentarzy i

przykłady ładowania sterowników dla głównych komponentów sprzętowych. Listę aktywnych w danym momencie modułów wyświetla się poleceniem `lsmod(1)`:

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

W tym przykładzie widać, że jest załadowany tylko moduł obsługi portu równoległego. Moduły usuwa się poleceniem `rmmmod(1)`, a ładuje poleceniami `modprobe(1)` lub `insmod(1)`. `Modprobe` jest z reguły bezpieczniejsze, ponieważ oprócz potrzebnego sterownika ładuje też inne, od których dany moduł zależy.

Większość użytkowników nigdy nie musi ręcznie ładować i usuwać modułów. Do zarządzania modułami służy system automatycznego ładowania w jądrze, *kmod*. Domyślnie jest on zawarty w jądrach Slackware. Umożliwia jądru automatyczne ładowanie modułów, kiedy są potrzebne. Więcej informacji na temat *kmod* i jego konfiguracji jest w pliku `/usr/src/linux/Documentation/kmod.txt`. Aby go przeczytać, musisz zainstalować źródła jądra z pakietów lub pobrać je z <http://kernel.org>.

Więcej informacji znajdziesz też na stronach podręcznika dla poszczególnych poleceń oraz w pliku `rc.modules`.

## Rozdział 5. Konfiguracja sieci

### 5.1 Wprowadzenie: *netconfig* Twoim przyjacielem

Podczas instalacji Slackware `setup` uruchomił program `netconfig`. Ten program przeprowadził następujące czynności:

- Zapytał o nazwę komputera i nazwę domeny.
- Krótko wyjaśnił sposoby przydzielania adresu, powiedział, kiedy i którego używać, zapytał, którego z nich użyć do skonfigurowania karty sieciowej:
  - statycznego adresu IP;
  - DHCP;
  - interfejsu zwrotnego (loopback).
- Następnie zaproponował wykrycie karty sieciowej.

`Netconfig` odpowiada za około 80% spraw związanych z konfiguracją sieci, jeśli mu na to pozwolisz. Jednak z paru powodów powinieneś dokładnie przejrzeć plik konfiguracyjny:

1. Nigdy nie bądź pewien, że program skonfiguruje komputer poprawnie. Jeśli użyłeś takiego programu, powinieneś sam przejrzeć wygenerowany plik konfiguracyjny.
2. Jeśli cały czas uczysz się obsługi Slackware i Linuksa jako takiego, przejrzanie działającej konfiguracji może wiele pomóc. Będziesz przynajmniej wiedział, jak wygląda poprawny plik konfiguracyjny, co pozwoli Ci później na rozwiązywanie problemów z błędną konfiguracją systemu.

## 5.2 Konfiguracja sprzętu sieciowego

Gdy zdecydujesz, że chcesz podłączyć komputer ze Slackware do jakiejś sieci, przede wszystkim będziesz potrzebował obsługiwanej przez Linuksa karty sieciowej. Musisz dokładnie sprawdzić, czy karta jest na pewno zgodna z Linuksem (sprawdź w Projekcie Dokumentacji Linuksowej i/lub dokumentacji dołączonej do jądra, aby zobaczyć, jakie karty są obecnie zalecane). Z reguły będziesz pozytywnie zaskoczony ilością sprzętu obsługiwanego przez najnowsze wersje jądra. Niemniej jednak, doradzam przed zakupem sprawdzenie jakiejś listy zgodności sprzętu (takiej jak The GNU/Linux Beginners Group Hardware Compatibility Links (<http://www.eskimo.com/%7Elo/linux/hardwarelinks.html>) czy The Linux Documentation Project Hardware HOWTO (<http://www.linux.org/docs/ldp/howto/Hardware-HOWTO/>)) dostępnej w Internecie. Trochę czasu poświęconego na poszukiwania może zaoszczędzić później kilka dni lub nawet tygodni prób rozwiązywania problemów z kartą, która w ogóle nie jest zgodna z Linuksem.

Przeglądając listę zgodności sprzętu w Internecie lub dokumentację jądra, warto zapisać, którego modułu będziesz potrzebował do obsługi karty sieciowej.

### 5.2.1 Ładowanie modułów obsługi sieci

Moduły jądra potrzebne już przy starcie systemu są ładowane przez plik `rc.modules` w katalogu `/etc/rc.d`, lub przez mechanizm automatycznego ładowania uruchamiany przez `/etc/rc.d/rc.hotplug`. Standardowy plik `rc.modules` zawiera sekcję „Network device support” (obsługa urządzeń sieciowych). Otwórz plik i odszukaj ją. Na początku jest wpis sprawdzający istnienie wykonywalnego skryptu `rc.netdevice` w `/etc/rc.d`. Ten skrypt jest tworzony, jeśli `setup` z powodzeniem wykrył kartę sieciową podczas instalacji.

Pod tym blokiem „if” znajduje się lista urządzeń sieciowych i wpisów `modprobe`. Wszystkie są zakomentowane. Odszukaj swoją kartę sieciową, odkomentuj odpowiedni wiersz `modprobe` i zapisz plik. Jeśli teraz uruchomisz `rc.modules` jako `root`, załaduje się sterownik odpowiedniej karty sieciowej (oraz inne wymienione tam, niezakomentowane moduły). Zwróć uwagę na to, że niektóre moduły (takie jak sterownik `ne2000`) wymagają podania parametrów. Pamiętaj, żeby odkomentować właściwy wpis.

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

Ten opis dotyczy wszystkich wewnętrznych kart sieciowych na złączu ISA i PCI. Sterowniki dla nich znajdują się w modułach jądra i są opisane w poprzedniej części. Program `/sbin/netconfig` powinien wykryć kartę i stworzyć odpowiedni plik `rc.netdevice`. Jeśli tak się nie stało, najprawdopodobniej próbujesz załadować moduł niewłaściwy dla karty. (Często jest tak, że karty tego samego producenta i tej samej marki są zbudowane z wykorzystaniem różnych układów, w zależności od wersji urządzenia. Dlatego wymagają różnych modułów). Jeśli jesteś pewien, że próbujesz ładować właściwy moduł, sprawdź jego dokumentację. Być może potrzebuje on specjalnych parametrów podczas inicjalizacji.

### 5.2.3 Modemy

Podobnie jak karty LAN, istnieją modemy podłączane do różnych szyn. Jeszcze niedawno najczęściej spotykane były 8- i 16-bitowe modemy ISA. Dzięki wysiłkom firmy Intel i producentów płyt głównych, skierowanych na zakończenie produkcji ISA, obecnie większość to modemy zewnętrzne podłączane do portu szeregowego albo USB, lub wewnętrzne modemy PCI. Jeśli chcesz używać modemu pod

Linuksem, **MUSISZ KONIECZNIE** poszukać informacji na ten temat przed jego zakupem, szczególnie jeśli rozważasz kupno urządzenia PCI. Wiele, czy wręcz większość, modemów PCI sprzedawanych obecnie to tzw. winmodemy. Nie ma na nich ważnych układów elektronicznych. Ich funkcje są przerzucone na procesor komputera i sterowniki działające w systemie Windows. Oznacza to, że nie mają one standardowego interfejsu szeregowego, którego będzie poszukiwał program PPPD podczas próby wdzwonienia się do Operatora Internetu (ISP – Internet Service Provider).

Jeśli chcesz być absolutnie pewien, że modem będzie współdziałał z Linuksem, kup modem zewnętrzny podłączany do komputera przez port szeregowy. Będzie on na pewno działał lepiej i będzie ogólnie łatwiejszy w instalacji i utrzymaniu, jakkolwiek wymaga zewnętrznego źródła zasilania i jest droższy.

Istnieją strony internetowe ze sterownikami i pomocą dotyczącą konfiguracji winmodemów. Niektórzy użytkownicy z powodzeniem zainstalowali i skonfigurowali sterowniki winmodemów opartych na układach Lucent, Conexant i Rockwell. Jednak ponieważ oprogramowanie wymagane dla tych urządzeń nie jest dołączone do Slackware, a sterowniki różnią się między sobą, nie będziemy się o nich rozpisywać.

## 5.2.4 PCMCIA

Podczas instalacji Slackware można zainstalować pakiet `pcmcia` (z grupy programów „A”). Zawiera on programy i pliki konfiguracyjne potrzebne, aby używać kart PCMCIA pod Slackware. Należy uwzględnić, że pakiet `pcmcia` zawiera tylko standardowe, podstawowe oprogramowanie. NIE instaluje żadnych sterowników ani modułów. Dostępne moduły znajdują się w katalogu `/lib/modules/`uname -r`/pcmcia`. Prawdopodobnie będziesz musiał poeksperymentować, aby uruchomić swoją kartę sieciową.

Potrzebna będzie edycja pliku `/etc/pcmcia/network.opts` (dla karty Ethernet) lub `/etc/pcmcia/wireless.opts` (dla karty bezprzewodowej). Jak większość plików konfiguracyjnych Slackware, zawierają one dużo wartościowych komentarzy, nie powinieneś więc mieć problemów z przeczytaniem ich i zrozumieniem, jakie zmiany wprowadzić.

## 5.3 Konfiguracja TCP/IP

W tym momencie karta sieciowa powinna być fizycznie zainstalowana w komputerze, a odpowiednie moduły załadowane. Nie możesz jeszcze korzystać z sieci, ale wpisz polecenie `ifconfig -a`, aby uzyskać informacje o karcie:

```
# 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)
```

Gdybyś wpisał samo `/sbin/ifconfig`, bez opcji `-a`, nie wyświetliłyby się informacje o interfejsie `eth0`, ponieważ karta sieciowa nie ma jeszcze przydzielonego adresu IP ani trasowania pakietów.

Istnieje wiele sposobów na stworzenie i konfigurację sieci. Można jednak wyróżnić dwa główne rodzaje: statyczny i dynamiczny. W sieciach statycznych każdy węzeł (ang. *node*, w żargonie komputerowców – każde urządzenie z własnym adresem IP) ma zawsze ten sam adres IP. W sieciach dynamicznych kontrolą i przydzielaniem adresów IP zajmuje się jeden serwer, zwany serwerem DHCP.

### 5.3.1 DHCP

DHCP (Dynamic Host Configuration Protocol – dynamiczny protokół konfiguracji hosta) to sposób, w jaki adres IP może być przydzielany komputerowi podczas jego uruchamiania. Kiedy uruchamia się *klient* DHCP, wysyła on w sieci lokalnej (LAN – Local Area Network) żądanie przydzielenia adresu do *serwera* DHCP. Serwer DHCP dysponuje *pulą* (ang. *pool*, zwaną też *zakresem* – *scope*) dostępnych adresów. Odpowie żądającej maszynie, przydzielając IP z puli razem z *czasem dzierżawy* (ang. *lease time*). Kiedy czas dzierżawy wygaśnie, klient musi skontaktować się z serwerem i ponownie negocjować adres.

Następnie klient zaakceptuje przyznany adres i skonfiguruje interfejs sieciowy z jego wykorzystaniem. Klienci DHCP używają, oprócz tego, pewnego cwanego patentu podczas negocjacji adresów. Klient pamięta adres, który był mu ostatnio przyznany, i podczas kolejnej negocjacji wyśle do serwera żądanie przydzielenia tego samego adresu. Jeśli to możliwe, serwer to wykona. Jeśli nie, przydzieli inny adres. Negocjacja wygląda mniej więcej tak:

*Klient:* Czy w sieci lokalnej jest dostępny serwer DHCP?

*Serwer:* Tak, jestem.

*Klient:* Potrzebuję adresu IP.

*Serwer:* Proszę, oto adres 192.168.10.10 na 19200 sekund.

*Klient:* Dziękuję.

*Klient:* Czy w sieci lokalnej jest dostępny serwer DHCP?

*Serwer:* Tak, jestem.

*Klient:* Potrzebuję adresu IP. Ostatnio dostałem 192.168.10.10. Mogę poprosić o niego znowu?

*Serwer:* Tak, oczywiście (albo: Nie, ale jest dostępny 192.168.10.12).

*Klient:* Dziękuję.



Linuksowym klientem DHCP jest `/sbin/dhclient`. Jeśli otworzysz `/etc/rc.d/rc.inet1` w swoim ulubionym edytorze tekstów, zauważysz odwołanie do `/sbin/dhclient` mniej więcej w połowie skryptu. Ten program wywoła negocjację opisaną powyżej. `Dhclient` kontroluje również upływ czasu dzierżawy i automatycznie skontaktuje się z serwerem DHCP w odpowiednim czasie, aby odnowić adres. DHCP może też kontrolować inne rzeczy, takie jak używany serwer ntp czy trasowanie pakietów.

Ustawienie DHCP w Slackware jest proste. Wystarczy uruchomić `netconfig` i wskazać DHCP, gdy wyświetli się odpowiednia opcja. Jeśli masz więcej niż jedną kartę sieciową i nie chcesz, aby `eth0` była konfigurowana przez DHCP, wyedytuj `/etc/rc.d/rc.inet1.conf`, zmieniając odpowiednią zmienną dla danej karty sieciowej na „*YES*” (tak).

### 5.3.2 Statyczny adres IP

Styczne adresy IP to adresy przyznane na stałe, które można zmienić tylko ręcznie. Używa się ich, kiedy administrator nie chce zmieniać IP, np. w przypadku wewnętrznych serwerów w sieci LAN, serwerów podłączonych do Internetu czy ruterów. Styczne adresy IP przypisuje się tylko raz. Inne komputery wiedzą, że dana maszyna ma zawsze ten sam adres IP i mogą się z nią skontaktować zawsze za jego pośrednictwem.

### 5.3.3 `/etc/rc.d/rc.inet1.conf`

Jeśli planujesz przypisanie adresu komputerowi ze Slackware, możesz to zrobić albo za pośrednictwem skryptu `netconfig`, albo wyedytować `/etc/rc.d/rc.inet1.conf`. W tym pliku znajdują się wpisy:

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

i dalej, bliżej końca:

```
GATEWAY=""
```

Wystarczy po prostu umieścić odpowiednie informacje między cudzysłowami. Do tych zmiennych będzie się odwoływał `/etc/rc.d/rc.inet1`, konfigurując karty sieciowe podczas startu systemu. Dla każdej karty trzeba wpisać właściwe dane IP lub „*YES*” przy `USE_DHCP`. Slackware uruchomi poszczególne interfejsy sieciowe zgodnie z zamieszczonymi informacjami, w kolejności, w jakiej występują.

Zmienna `DEFAULT_GW` ustawia domyślną trasę pakietów w Slackware. Wszelka komunikacja między Twoim komputerem a innymi komputerami w Internecie musi przejść przez tę bramę, jeśli nie poda się innej trasy. Jeśli używasz DHCP, nie musisz raczej nic tu wpisywać, bo serwer DHCP sam poda odpowiednią bramę.

### 5.3.4 /etc/resolv.conf

No dobra, masz już adres IP, masz domyślną bramę, możesz sobie mieć nawet 10 milionów dolarów (podziel się). Ale co z tego, skoro nie możesz tłumaczyć nazw komputerów na adresy IP? Nikt nie będzie wpisywał 72.9.234.112 w pasku adresów przeglądarki, żeby wejść na [www.slackbook.org](http://www.slackbook.org). No i komu oprócz autorów książki chciałoby się zapamiętywać ten adres IP? Musimy ustawić serwer DNS, ale jak? Tu do głosu dochodzi `/etc/resolv.conf`.

Być może masz już odpowiednie opcje w `/etc/resolv.conf`. Jeśli konfigurujesz sieć przez DHCP, serwer DHCP powinien zająć się aktualizacją tego pliku (technicznie rzecz biorąc, serwer po prostu mówi klientowi `dhcpcd`, co w nim wpisać, a klient to wykonuje). Jeśli jednak musisz zmienić listę serwerów DNS ręcznie, będziesz musiał sam zmodyfikować `/etc/resolv.conf`. Oto przykład:

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

Pierwszy wiersz jest prosty. Dyrektywa `nameserver` mówi, które serwery DNS odpytywać. Z konieczności są to zawsze adresy IP. Może ich być dowolna ilość. Slackware będzie sprawdzał jeden po drugim, aż napotka pasującą opcję.

Drugi wiersz jest bardziej interesujący. Dyrektywa `search` podaje listę nazw domen, z których komputer ma korzystać, kiedy wykonuje zapytanie do serwera DNS. Dzięki temu można skontaktować się z komputerem, podając tylko pierwszą część FQDN (Fully Qualified Domain Name – pełna nazwa domeny). Jeśli np. w ścieżce wyszukiwania (`search`) znajduje się „`slackware.com`”, możesz się skontaktować z `http://store.slackware.com`, wysyłając zapytanie o samo „`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
```

### 5.3.5 /etc/hosts

Dobra, mamy już działający serwer DNS. A jeśli chcemy go obejść lub dodać wpis dla maszyny, której serwer DNS nie ma na liście? Slackware ma lubiany przez wielu plik `/etc/hosts`, który zawiera lokalną listę nazw DNS i odpowiadających im adresów IP.

```
# cat /etc/hosts
127.0.0.1          localhost localhost.localdomain
192.168.1.101     redtail
172.14.66.32      foobar.slackware.com
```

Widać, że komputer lokalny (`localhost`) ma adres `127.0.0.1` (zawsze zarezerwowany dla komputera lokalnego), maszyna `redtail` ma adres `192.168.1.101`, a `foobar.slackware.com` – `172.14.66.32`.

## 5.4 PPP

Wielu użytkowników ciągle jeszcze łączy się z Internetem za pośrednictwem połączenia dodzwanianego. Najpowszechniejszą metodą łączenia jest PPP, czasami także SLIP. Skonfigurowanie systemu tak, aby mógł porozumiewać się z serwerem przez PPP jest całkiem proste. Dołączyliśmy do Slackware w tym celu kilka pomocnych narzędzi.

### 5.4.1 pppsetup

W Slackware znajduje się program o nazwie `pppsetup`, który konfiguruje dodzwaniane konto w systemie. Wygląda i działa podobnie, jak `netconfig`. Aby go uruchomić, upewnij się, że jesteś zalogowany jako `root`. Uruchom program, wpisując `pppsetup`.

Program zada szereg pytań, na które podasz właściwe odpowiedzi – informacje o modemie i ciągu inicjalizacyjnym modemu, numer telefonu operatora Internetu. Niektóre pozycje będą ustawione domyślnie, co w większości przypadków jest odpowiednie.

Po uruchomieniu `pppsetup` utworzy dwa programy – `ppp-go` i `ppp-off`. Używa się ich odpowiednio do uruchamiania i zatrzymywania połączenia PPP. Znajdują się one w `/usr/sbin`, a do ich wykonania potrzebne są prawa `roota`.

### 5.4.2 /etc/ppp

Większości użytkowników wystarczy uruchomienie `pppsetup`. Czasami jednak trzeba dostosować niektóre wartości używane przez demona PPP. Wszystkie opcje konfiguracyjne znajdują się w `/etc/ppp`. Oto lista i przeznaczenie poszczególnych plików:

<code>ip-down</code>	Ten skrypt jest wykonywany przez <code>pppd</code> po zakończeniu połączenia PPP.
<code>ip-up</code>	Ten skrypt jest wykonywany przez <code>pppd</code> po nawiązaniu połączenia PPP. Wpisz tu polecenia, które mają być wykonywane po nawiązaniu połączenia.
<code>options</code>	Ogólne opcje konfiguracyjne dla <code>pppd</code> .
<code>options.demand</code>	Ogólne opcje konfiguracyjne dla <code>pppd</code> przy uruchamianiu w trybie dzwonienia na żądanie ( <code>demand dialling</code> ).
<code>pppscript</code>	Polecenia wysyłane do modemu.
<code>pppsetup.txt</code>	Zapis tego, co wpisałeś po uruchomieniu <code>pppsetup</code> .

Uwaga: Większość z tych plików zostanie utworzona dopiero po uruchomieniu `pppsetup`.

## 5.5 Konfiguracja sieci bezprzewodowej

Sieci bezprzewodowe są nowinką w świecie komputerów. Zdobywają jednak coraz większą popularność, ponieważ coraz więcej ludzi kupuje laptopy i chcą oni pracować w sieci bez zawracania sobie głowy kablami. Ten trend utrzymuje się cały czas. Niestety, Linux nie obsługuje sieci bezprzewodowych tak

dobrze, jak tradycyjnych sieci kablowych.

Aby skonfigurować bezprzewodową kartę sieciową 802.11, potrzebne są trzy rzeczy:

1. Sprzętowa obsługa karty bezprzewodowej.
2. Konfiguracja karty tak, aby mogła się podłączyć do punktu dostępowego.
3. Konfiguracja sieci.

### 5.5.1 Sprzętowa obsługa karty bezprzewodowej

Obsługą takiego sprzętu zajmuje się jądro. Sterownik jest albo wbudowany w jądro, albo dostępny jako moduł. Większość sterowników do nowych kart ma postać modułów, więc wystarczy określić potrzebny moduł i załadować go przez `/etc/rc.d/rc.modules`. Program `netconfig` może mieć problemy z wykryciem karty bezprzewodowej, więc będziesz potrzebował określić moduł sam. Więcej informacji na temat sterowników jądra dla różnych kart bezprzewodowych znajduje się na stronie [http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/).

### 5.5.2 Konfiguracja ustawień bezprzewodowych

Większości ustawień dokonuje się za pomocą programu `iwconfig`. Przeczytaj stronę podręcznika `iwconfig`, aby dowiedzieć się więcej.

Najpierw musisz skonfigurować punkt dostępowy (access point). Takie punkty mają różną terminologię i sposoby konfiguracji, będziesz je więc musiał odpowiednio przystosować do posiadanej karty. Ogólnie rzecz biorąc, potrzebujesz co najmniej następujących informacji:

- identyfikator domeny (domain ID) lub nazwę sieci (w `iwconfig` nazywa się to ESSID)
- kanał, którego używa WAP
- ustawienia szyfrowania, włączając w to klucze (najlepiej w postaci heksadecymalnej).



**INFORMACJA NA TEMAT WEP.** WEP ma wiele wad, ale jest lepsze niż nic. Jeśli wymagasz od swojej sieci bezprzewodowej większego stopnia bezpieczeństwa, zainteresuj się VPN lub IPSec, które jednak wykraczają poza zakres tej książki. Warto również skonfigurować WAP, aby nie ogłaszał swojego ID/ESSID. Polityka zabezpieczania sieci bezprzewodowych wykracza poza zakres sekcji, ale odrobina szukania na Google da Ci więcej niż chciałeś wiedzieć.

Zakładamy, że używasz `modprobe` do ładowania odpowiednich sterowników i że zgromadziłeś potrzebne informacje. Wyedytuj teraz `rc.wireless.conf`, dodając odpowiednie ustawienia. Ten plik jest cokolwiek nieporządkowany. Najprościej zmodyfikować tylko standardową sekcję, wpisując ESSID i KEY, ewentualnie CHANNEL, jeśli karta go wymaga (warto sprawdzić, czy karta działa bez wpisywania CHANNEL. Jeśli tak, to dobrze, jeśli nie, ustaw go). Jeśli jesteś odważny, możesz ustawić tylko niezbędne zmienne. Nazwy zmiennych w `rc.wireless.conf` odpowiadają parametrom `iwconfig`, są odczytywane przez skrypt `rc.wireless` i przekazywane do odpowiednich poleceń `iwconfig`.

Jeśli klucz (key) jest w postaci heksadecymalnej, to świetnie. Możesz być prawie pewien, że WAP i `iwconfig` będą go akceptować. Jeśli masz tylko ciąg znaków, nie będziesz na pewno wiedział, jak WAP przełoży go na klucz heksadecymalny. Będziesz musiał zgadywać (albo zdobyć klucz WAP w

formie heksadecymalnej).

Po zmodyfikowaniu `rc.wireless.conf` uruchom `rc.wireless` jako `root`, a następnie uruchom `rc.inet1`, też jako `root`. Przetestuj połączenie bezprzewodowe za pomocą standardowych narzędzi, takich jak `ping` i `iwconfig`. Jeśli masz też interfejs przewodowy, wyłącz go, używając `ifconfig`, aby połączenia nie przeszkadzały sobie wzajemnie. Sprawdź też, czy wszystko działa po ponownym uruchomieniu systemu.

Wiesz już, jak zmodyfikować `/etc/rc.d/rc.wireless`. Przyjrzyjmy się teraz działaniu `iwconfig`. Dzięki temu będziesz wiedział, jak na szybko skonfigurować wifi, kiedy jesteś w kawiarni internetowej, restauracji albo innym punkcie dostępowym typu „hot spot” i chcesz podłączyć się do Sieci.

Najpierw powiedz karcie bezprzewodowej, do jakiej sieci się podłączyć. Zastąp „eth0” przez odpowiedni numer używany przez kartę wifi i zmień „mojasieć” na właściwy ESSID. Tak, wiemy, że nie jesteś aż takim głabem. Następnie ustaw klucz szyfrujący używany w sieci (jeśli jest). Na koniec wybierz właściwy kanał (jeśli jest taka potrzeba).

```
# iwconfig eth0 essid "mojasieć"
# iwconfig eth0 key XXXXXXXXXXXXXXXXXXXXXXXXXXXX
# iwconfig eth0 channel n
```

To tyle, jeśli chodzi o część bezprzewodową.

### 5.5.3 Konfiguracja sieci

Przeprowadza się ją tak samo, jak konfigurację sieci przewodowej, co jest opisane we wcześniejszych sekcjach rozdziału.

## 5.6 Sieciowe systemy plików

Działa już połączenie TCP/IP z siecią. Możesz pingować komputery w sieci lokalnej i w Internecie, o ile poprawnie skonfigurowałeś bramę. Jak wiadomo, sens podłączenia komputera do sieci leży w dostępie do informacji. Niektórzy konfigurują sieć dla zabawy. Większość jednak chce współdzielić pliki i drukarki, mieć dostęp do informacji w Internecie czy grać w gry online. Zainstalowany i działający w Slackware protokół TCP/IP to dobry środek na drodze do osiągnięcia celu, ale sam TCP/IP daje tylko bardzo podstawową funkcjonalność. Aby współdzielić pliki, trzeba przenosić je między komputerami przy użyciu FTP lub SCP. Nie można przeglądać plików na komputerze ze Slackware za pomocą ikony Otoczenie sieciowe czy Moje miejsca sieciowe z poziomu maszyn z Windows. Chcielibyśmy też móc bezproblemowo przeglądać pliki na innych komputerach uniksowych.

Innymi słowy, potrzebujemy *sieciowego systemu plików* (network file system), aby mieć przezroczysty, niezakłócony dostęp do plików na innych komputerach. Programy do obróbki informacji, których używamy, tak naprawdę nie muszą wiedzieć, na jakim komputerze jest przechowywany dany plik. Muszą tylko wiedzieć, że on istnieje i jak się do niego dostać. Zadaniem systemu operacyjnego jest więc zarządzanie takim dostępem za pomocą lokalnych i sieciowych systemów plików. Dwa najpowszechniej używane sieciowe systemy plików to SMB (zaimplementowany w systemie Samba) oraz NFS.

## 5.6.1 SMB/Samba/CIFS

SMB (Server Message Block – blok komunikatów serwera) to potomek starszego protokołu (NetBIOS), pierwotnie używanego przez IBM w produkcie LAN Manager. Firma Microsoft zawsze interesowała się NetBIOS i jego następcami (NetBEUI, SMB i CIFS). Projekt Samba istnieje od 1991. Początkowo napisano go, aby połączyć IBM PC, na którym był uruchomiony NetBIOS, z serwerem uniksowym. Obecnie praktycznie wszędzie w cywilizowanym świecie preferuje się SMB jako metodę współdzielenia plików i usług drukowania w sieci, ponieważ obsługuje ją Windows.

Plik konfiguracyjny Samby, `/etc/samba/smb.conf`, należy do plików konfiguracyjnych, które mają najlepsze komentarze i dokumentację. Umieszczono w nim przykładowe konfiguracje udziałów, które można łatwo dostosować do swoich potrzeb. Jeśli wymagasz lepszej kontroli dostępu, strona podręcznika dla `smb.conf` to lektura obowiązkowa. Jako że Sambę opisano najlepiej w wymienionych źródłach, nie będziemy tu przytaczali dokumentacji. Krótko jednak opiszemy podstawy.

Plik `smb.conf` jest podzielony na wiele sekcji: jedna sekcja na udział oraz sekcja globalna, w której wpisuje się ustawienia wspólne dla wszystkich udziałów. Niektóre ustawienia działają tylko, jeśli są umieszczone w sekcji globalnej, niektóre tylko poza nią. Pamiętaj, że nad wpisami w sekcji globalnej mają pierwszeństwo wpisy w innych sekcjach. Przeczytaj strony podręcznika, aby dowiedzieć się więcej.

Najprawdopodobniej zmodyfikujesz `smb.conf` zgodnie z ustawieniami swojej sieci lokalnej. Proponuję zmienić następujące pozycje:

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

Zmień nazwę grupy roboczej (`workgroup`), aby była zgodna z nazwą lokalnej grupy roboczej lub nazwy domeny.

```
# server string is the equivalent of the NT Description field
server string = Serwer Samby
```

Serwer Samby będzie nazwą, pod jaką komputer ze Slackware będzie widoczny w windowsowym folderze Otoczenie sieciowe (lub Moje miejsca sieciowe).

```
# 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
```

Prawie na pewno należy zaimplementować zabezpieczenia na poziomie użytkownika (`user level security`) w Slackware.

```
# 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 włączysz opcji szyfrowania haseł (encrypt passwords), nie będzie możliwe używanie Samby z NT 4.0, Win2k, WinXP i Win2003. Wcześniejsze wersje Windows nie wymagają szyfrowania haseł do współdzielenia plików.

SMB jest protokołem uwierzytelnianym (authenticated). To znaczy, że aby używać tej usługi, trzeba podać poprawną nazwę użytkownika i hasło. Do przekazywania serwerowi samby, jakie nazwy użytkowników i hasła są poprawne, używa się polecenia smbpasswd. To polecenie przyjmuje kilka przełączników, za których pomocą można dodawać zwykłych użytkowników lub nazwy maszyn. SMB wymaga dodania nazw komputerów NETBIOS jako nazw maszyn, co ogranicza liczbę komputerów, z których można się logować.

```
Adding a user to the /etc/samba/private/smbpasswd file.  
# smbpasswd -a user  
Adding a machine name to the /etc/samba/private/smbpasswd file.  
# smbpasswd -a -m machine
```

Należy pamiętać, że dana nazwa użytkownika lub maszyny musi już istnieć w /etc/passwd. Aby dodać do niego nazwę, wystarczy użyć polecenia adduser. Uwaga: dodając za pomocą adduser nazwę maszyny, należy dodać znak dolara („,\$”) do nazwy maszyny. Nie powinno się tego robić przez smbpasswd – to polecenie dodaje znak dolara samo. Jeśli nie dodasz „,\$” przy adduser, wystąpi błąd podczas dodawania nazwy maszyny do samby.

```
# adduser nazwamaszyny$
```

## 5.6.2 Sieciowy system plików (Network File System – NFS)

Sieciowy system plików NFS początkowo stworzyła firma Sun na potrzeby własnej implementacji Uniksa – Solaris. Jest on o wiele łatwiejszy w instalacji i uruchomieniu w porównaniu z SMB, ale jest też o wiele mniej bezpieczny. Podstawowym zagrożeniem bezpieczeństwa w NFS jest łatwość podszywania się pod numery innych użytkowników i grup podczas komunikacji między komputerami. NFS to protokół niewierzytelniany. Projektuje się przyszłe wersje NFS, w których poziom bezpieczeństwa będzie większy, ale w momencie pisania nie były one powszechnie stosowane.

Konfiguracja NFS jest zapisana w pliku /etc/exports. Po otwarciu domyślnego pliku /etc/exports w edytorze zobaczysz pusty plik z dwoma wierszami komentarza na górze. Musisz dodać wiersz dla każdego katalogu, który chcesz wyeksportować, wraz z listą klienckich stacji roboczych, które będą miały do niego dostęp. Aby np. wyeksportować katalog /home/foo do stacji roboczej Bar, wystarczy dodać:

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

do pliku /etc/exports. Poniżej jest zamieszczony przykład ze strony podręcznika 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 widzisz, dostępne są różne opcje, ale większość z tych przedstawionych na przykładzie powinna być zrozumiała.

NFS działa przy założeniu, że dany użytkownik na jednym komputerze w sieci ma ten sam identyfikator użytkownika (User ID – UID) na wszystkich komputerach w sieci. Podczas próby zapisu lub odczytu z klienta NFS na serwerze NFS jako część żądania zapisu/odczytu jest przesyłany UID. Ten UID jest traktowany w taki sam sposób, jakby żądanie zapisu/odczytu pochodziło z lokalnego komputera. Jak widzisz, ktoś mógłby dowolnie podać dany UID podczas uzyskiwania dostępu do zdalnego systemu. Wówczas mogłyby się zdarzyć, i zdarzyłyby się, Niedobre Rzeczy (tm). Częściowym zabezpieczeniem jest montowanie każdego katalogu z opcją *root\_squash*. Mapuje to UID każdego użytkownika podającego się za *roota* na inny UID, uniemożliwiając dostęp z prawami *roota* do plików i katalogów w wyeksportowanym katalogu. Ta opcja jest z reguły domyślnie włączona jako środek bezpieczeństwa, ale autorzy zalecają podanie jej tak czy owak w pliku */etc/exports*.

Możesz też wyeksportować katalog bezpośrednio z wiersza poleceń na serwerze, używając do tego polecenia *exportfs*:

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

Ten wiersz eksportuje katalog */home/foo* do komputera „Bar” i daje mu prawa zapisu/odczytu. Ponadto, serwer NFS nie będzie wywoływał *root\_squash*, co oznacza, że każdy użytkownik komputera *Bar* z UID równym „0” (UID *roota*) będzie miał na serwerze takie same przywileje, jak *root*. Składnia faktycznie wygląda dziwnie (zwykle, podając katalog w postaci *komputer:/katalog/plik*, odwołujesz się do pliku lub katalogu na danym komputerze).

Więcej informacji na ten temat znajduje się na stronie podręcznika dla pliku *exports*.

## Rozdział 6. Konfiguracja X

Od wersji Slackware-10.0 w dystrybucji znajduje się środowisko graficzne X Window dostarczone przez Xorg. X jest odpowiedzialne za graficzny interfejs użytkownika. Jest ono niezależne od systemu operacyjnego, w przeciwieństwie do Windows czy MacOS.

System X Window jest zaimplementowany w postaci wielu programów, działających w przestrzeni użytkownika. Dwa główne komponenty to serwer i menedżer okien. Serwer oferuje niskopoziomowe funkcje w celu interakcji z kartą graficzną, dlatego jest specyficzny dla danego systemu. „Na czubku” serwera znajduje się menedżer okien, który udostępnia interfejs użytkownika. Zaletą takiego rozwiązania jest fakt, że możesz używać wielu różnych graficznych interfejsów, zmieniając po prostu menedżer okien.

Konfiguracja X może być trudna i skomplikowana. Powodem tego jest duża liczba różnych kart wideo dostępnych dla architektury PC, z których większość ma różne interfejsy programowania. Na szczęście



większość kart obsługuje podstawowy standard wideo, tzw. VESA. Jeśli Twoja karta go obsługuje, będziesz mógł uruchomić X poleceniem startx od razu po zainstalowaniu systemu.

Jeśli to nie zadziała, lub jeśli chciałbyś używać zaawansowanych funkcji zwiększających wydajność, np. sprzętowej akceleracji lub sprzętowego renderowania 3-D, będziesz musiał zrekonfigurować X.

Aby skonfigurować X, musisz utworzyć plik `/etc/X11/xorg.conf`. Zawiera od wiele szczegółowych informacji na temat Twojej karty wideo, myszy i monitora. Ten plik jest dość złożony, ale na szczęście istnieje kilka programów pomagających w jego utworzeniu. Opiszemy tu kilka z nich.

## 6.1 xorgconfig

Jest to prosty, wykorzystujący menu program, podobny do programu instalacyjnego Slackware. Mówi on po prostu serwerowi X, aby sprawdził możliwości karty, a następnie stworzył najlepszy wstępny plik w oparciu o informacje, które uzyska. Wygenerowany plik `/etc/X11/xorg.conf` powinien stanowić dobry punkt wyjściowy dla większości systemów (i powinien działać bez potrzeby edycji).

To tekstowy program konfiguracyjny X, przeznaczony dla zaawansowanych administratorów systemu. Oto przykładowe działanie programu. Na początek, uruchom go:

```
# xorgconfig
```

Wyświetli się cały ekran informacji o `xorgconfig`. Aby kontynuować, wciśnij **ENTER**. Program zapyta, czy właściwie skonfigurowałeś zmienną `PATH`. Powinna być dobra, więc wciśnij **ENTER**.

Rys. 6-1. Konfiguracja myszy w `xorgconfig`

```
First specify a mouse protocol type. Choose one from the following list:

1.  Auto
2.  SysMouse
3.  MouseSystems
4.  PS/2
5.  Microsoft
6.  Busmouse
7.  IMPS/2
8.  ExplorerPS/2
9.  GlidePointPS/2
10. MouseManPlusPS/2
11. NetMousePS/2
12. NetScrollPS/2
13. ThinkingMousePS/2
14. AceCad

The recommended protocol is Auto. If you have a very old mouse
or don't want OS support or auto detection, and you have a two-button
or three-button serial mouse, it is most likely of type Microsoft.

Enter a protocol number:
```

Najpierw wybierz protokół myszy. Wybierz jeden z listy poniżej:

Zalecany protokół to Auto. Jeśli masz bardzo starą mysz lub nie chcesz włączać obsługi przez system albo autowykrywania, a także jeśli masz dwuprzyciskową lub trójprzyciskową mysz szeregową, jest ona najprawdopodobniej typu Microsoft.

Wpisz numer protokołu:

Wybierz mysz z przedstawionego menu. Jeśli Twoja mysz szeregową nie jest wymieniona, wybierz protokół Microsoft – jest on najbardziej powszechny i prawdopodobnie będzie działał. Następnie `xorgconfig` zapyta o opcję *ChordMiddle* i *Emulate3Buttons*. Będą one szczegółowo opisane na ekranach. Użyj ich, jeśli środkowy przycisk myszy nie działa w X, albo jeśli masz dwuprzyciskową mysz (*Emulate3Buttons* pozwala na symulację zachowania środkowego przycisku poprzez jednoczesne naciśnięcie obu przycisków). Dalej wpisz nazwę urządzenia myszy. Domyślny wybór, `/dev/mouse`, powinien działać, ponieważ to dołączenie zostało utworzone podczas instalacji Slackware. Jeśli masz uruchomiony serwer myszy GPM w trybie repeatera, możesz ustawić typ myszy jako `/dev/gpmdata`, aby X otrzymywało informacje o myszy za pośrednictwem gpm. W niektórych przypadkach, szczególnie jeśli masz mysz w standardzie Bus Mouse, może to działać lepiej. Większość użytkowników nie powinna jednak tego robić.

`Xorgconfig` zapyta o włączenie specjalnych przypisań klawiszy. Jeśli ich potrzebujesz, wpisz „y”. Większość ludzi może wpisać „n” - wpisz to, jeśli nie jesteś pewien.

### Rys. 6-2 Synchronizacja pozioma w `xorgconfig`

```
You must indicate the horizontal sync range of your monitor. You can either
select one of the predefined ranges below that correspond to industry-
standard monitor types, or give a specific range.

It is VERY IMPORTANT that you do not specify a monitor type with a horizontal
sync range that is beyond the capabilities of your monitor. If in doubt,
choose a conservative setting.

    hsync in kHz; monitor type with characteristic modes
  1  31.5; Standard VGA, 640x480 @ 60 Hz
  2  31.5 - 35.1; Super VGA, 800x600 @ 56 Hz
  3  31.5, 35.5; 8514 Compatible, 1024x768 @ 87 Hz interlaced (no 800x600)
  4  31.5, 35.15, 35.5; Super VGA, 1024x768 @ 87 Hz interlaced, 800x600 @ 56 Hz
  5  31.5 - 37.9; Extended Super VGA, 800x600 @ 60 Hz, 640x480 @ 72 Hz
  6  31.5 - 48.5; Non-Interlaced SUGA, 1024x768 @ 60 Hz, 800x600 @ 72 Hz
  7  31.5 - 57.0; High Frequency SUGA, 1024x768 @ 70 Hz
  8  31.5 - 64.3; Monitor that can do 1280x1024 @ 60 Hz
  9  31.5 - 79.0; Monitor that can do 1280x1024 @ 74 Hz
 10  31.5 - 82.0; Monitor that can do 1280x1024 @ 76 Hz
 11  Enter your own horizontal sync range

Enter your choice (1-11):
```

Musisz podać zakres synchronizacji poziomej monitora. Wybierz jeden ze zdefiniowanych zakresów, odpowiednich dla standardowych typów monitorów, lub podaj własny zakres.

Jest NIEZWYKLE WAŻNE, abyś nie podawał typu monitora z zakresem synchronizacji poziomej przewyższającym możliwości Twojego monitora. W razie wątpliwości wybierz bezpieczne ustawienia.

Synchronizacja pozioma w kHz; typ monitora z charakterystycznymi trybami

1 31.5; Standard VGA, 640x480 @ 60 Hz

2 31.5 – 35.1; Super VGA, 800x600 @ 56 Hz

3 31.5, 35.5; Zgodna z 8514, 1024x768 @ 87 z przeplotem (bez 800x600)

4 31.5, 35.15, 35.5; Super VGA, 1024x768 @ 87 Hz z przeplotem, 800x600 @ 56 Hz

5 31.5 – 37.9; Rozszerzona Super VGA, 800x600 @ 60 Hz, 640x480 @ 72 Hz

6 31.5 – 48.5; SVGA bez przeplotu, 1024x768 @ 60 Hz, 800x600 @ 72 Hz

7 31.5 – 57.0; SVGA o wysokiej częstotliwości, 1024x768 @ 70 Hz

8 31.5 – 64.3; Monitor osiągający 1280x1024 @ 60 Hz

9 31.5 – 79.0; Monitor osiągający 1280x1024 @ 74 Hz

10 31.5 – 82.0; Monitor osiągający 1280x1024 @ 76 Hz

11 Wpisz własny zakres synchronizacji poziomej

Wpisz wybrany numer (1-11):

W następnej sekcji wpisujesz zakres synchronizacji monitora. Aby rozpocząć konfigurację monitora, wciśnij ENTER. Zobaczysz listę typów monitora – wybierz jeden z nich. Uważaj, aby nie przekroczyć specyfikacji monitora, bo może on ulec uszkodzeniu.

### Rys. 6-3 Synchronizacja pionowa `xorgconfig`

```
You must indicate the vertical sync range of your monitor. You can either
select one of the predefined ranges below that correspond to industry-
standard monitor types, or give a specific range. For interlaced modes,
the number that counts is the high one (e.g. 87 Hz rather than 43 Hz).
```

```
1 50-70
2 50-90
3 50-100
4 40-150
5 Enter your own vertical sync range
```

```
Enter your choice: █
```

Musisz podać zakres synchronizacji pionowej monitora. Wybierz jeden ze zdefiniowanych zakresów poniżej, które odpowiadają standardowym typom monitora, lub wpisz własny adres. Dla trybów z przeplotem właściwy będzie zakres wysoki (np. 87 Hz, a nie 43 Hz).

1 50-70

2 50-90

3 50-100

4 40-150

5 Wpisz własny zakres synchronizacji pionowej

Wpisz wybrany numer:

Wpisz zakres synchronizacji pionowej monitora (powinieneś go znaleźć w instrukcji obsługi monitora). Xorgconfig zapyta też o ciągi znaków identyfikujące typ monitora w pliku xorg.conf. W tych 3 wierszach możesz wpisać to, co chcesz (albo nie wpisywać nic).

#### Rys 6-4. Karta wideo w xorgconfig

```
0 * Generic VESA compatible          -
1 * Generic VGA compatible          -
2 * Unsupported VGA compatible      -
3 ** 3DLabs, TI (generic)           [glint] -
4 ** 3Dfx (generic)                 [tdfx]  -
5 ** ATI (generic)                  [ati]   -
6 ** ATI Radeon (generic)           [radeon]-
7 ** ATI Rage 128 based (generic)   [r128]  -
8 ** Alliance Pro Motion (generic)  [apm]   -
9 ** Ark Logic (generic)            [ark]   -
10 ** Chips and Technologies (generic) [chips] -
11 ** Cirrus Logic (generic)        [cirrus]-
12 ** Cyrix MediaGX (generic)       [cyrix] -
13 ** DEC TGA (generic)             [tga]   -
14 ** Intel i740 (generic)          [i740]  -
15 ** Intel i810 (generic)          [i810]  -
16 ** Linux framebuffer (generic)   [fbdev] -
17 ** Matrox Graphics (generic)     [mga]   -

Enter a number to choose the corresponding card definition.
Press enter for the next page, q to continue configuration.
```

Wpisz numer odpowiadający danej definicji karty. Wciśnij enter, aby przejść do następnej strony, q, aby kontynuować konfigurację.

Teraz możesz przejrzeć bazę danych typów kart wideo. Zapewne będziesz chciał to zrobić, więc wpisz „y” i wybierz kartę z poniższej listy. Jeśli nie widzisz dokładnie takiej samej karty, wybierz taką, która ma taki sam układ, to powinno działać poprawnie.

Następnie powiedz xorgconfig, ile pamięci RAM masz na karcie. Xorgconfig będzie też chciał, abyś wpisał tekst opisujący kartę wideo. W tych trzech wierszach możesz wpisać opis.

Kolejne pytanie dotyczy rozdzielczości ekranu, których chcesz używać. Na początek wystarczy trzymać się zaproponowanych rozdzielczości domyślnych. Później możesz wyedytować plik `/etc/X11/xorg.conf` i zmienić kolejność trybów, aby 1024x768 (lub inny wybrany tryb) był domyślny.

Na koniec `xorgconfig` zapyta, czy chcesz zapisać bieżący plik konfiguracyjny. Po odpowiedzeniu „yes” (tak) plik konfiguracyjny X zostanie zapisany i proces konfiguracji się zakończy. Teraz możesz uruchomić X poleceniem `startx`.

## 6.2 *xorgsetup*

Drugi sposób konfiguracji X to `xorgsetup`, automagiczny program konfiguracyjny dołączony do Slackware.

Aby go uruchomić, zaloguj się jako `root` i wpisz:

```
# xorgsetup
```

Jeśli w systemie istnieje plik `/etc/X11/xorg.conf` (bo X było już wcześniej konfigurowane), wyświetli się pytanie, czy zapisać kopię zapasową istniejącego pliku konfiguracyjnego przed kontynuacją. Nazwa oryginalnego pliku zostanie zmieniona na `/etc/X11/xorg.conf.backup`.

## 6.3 *xinitrc*

Program `xinit(1)` to właściwy program uruchamiający X. Jest on wywoływany przez polecenie `startx(1)`, więc pewnie nie zwróciłeś na to uwagi (i raczej nie jest Ci to potrzebne). Plik konfiguracyjny `xinit` jest jednak istotny, ponieważ opisuje, jakie programy (szczególnie menedżer okien) są uruchamiane po starcie X. `Xinit` najpierw sprawdza, czy w katalogu domowym jest plik `.xinitrc`. Jeśli tak, jest on wykonywany. Jeśli nie, zostaje użyty domyślny plik systemowy `/var/X11R6/lib/xinit/xinitrc`. Oto przykładowy taki plik:

```

#!/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
    xrdp -merge $sysresources
fi

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

if [ -f $userresources ]; then
    xrdp -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 bloki „if” służą dołączaniu ustawień konfiguracyjnych z innych plików. Najbardziej interesująca jest końcowa część pliku, która uruchamia różne programy. Taka sesja X uruchamia menedżer okien `twm(1)`, zegar i trzy okna terminala. Zwróć uwagę na słowo `exec` przed ostatnim wywołaniem `xterm`. Zastępuje ono aktualnie działającą powłokę (tę, która wykonała skrypt `xinitrc`) poleceniem `xterm(1)`. Gdy użytkownik zamknie to okno `xterm`, zakończy sesję X.

Aby dostosować sposób uruchamiania X do własnych potrzeb, skopiuj domyślny `/var/X11R6/lib/xinit/xinitrc` do `~/xinitrc` i zmodyfikuj go, zmieniając odpowiednie wiersze. Końcówka mojego pliku wygląda tak:

```

# Start the window manager:
exec startkde

```

Zauważ, że w `/var/X11R6/lib/xinit/` znajduje się kilka plików `xinitrc.*`, które odpowiadają konkretnym menedżerom okien i graficznym interfejsom użytkownika. Wybierz ten, który chcesz.

## 6.4 `xwmconfig`

Przez lata Unix był używany wyłącznie jako system operacyjny serwerów, z wyjątkiem superwydajnych profesjonalnych stacji roboczych. Tylko użytkownicy o zainteresowaniach technicznych byli skłonni do używania uniksopodobnego systemu operacyjnego, co widać po graficznym interfejsie użytkownika (GUI). GUI miały tendencję do bycia bardzo prostymi, zaprojektowanymi do uruchamiania kilku z natury graficznych aplikacji, takich jak programy do renderowania obrazów i CAD. Większość czynności związanych z zarządzaniem systemem i plikami wykonywało się z wiersza poleceń. Wielu producentów (Sun Microsystems, Silicon Graphics itp.) sprzedawało stacje robocze, które próbowały zachowywać spójny wygląd i zachowanie (look and feel), ale różnorodność zestawów narzędzi do tworzenia GUI, używanych przez programistów, prowadziła do zakłócenia tej spójności. Np. w dwóch aplikacjach pasek przewijania był inny. Menu pojawiały się w różnych miejscach. Programy miały różne przyciski i okna wyboru. Kolory różniły się i z reguły były wewnętrznie zakodowane w każdym zestawie narzędzi. Dopóki używali tego głównie ludzie zainteresowani komputerami, nie miało to wielkiego znaczenia.

Wraz z pojawieniem się darmowych uniksopodobnych systemów i rosnącą liczbą graficznych aplikacji, X zyskało dużą grupę użytkowników. Większość z nich jest oczywiście przyzwyczajona do spójnego wyglądu i zachowania pulpitu graficznego, takiego jak w Microsoft Windows czy Apple MacOS. Brak takiej spójności stanowił przeszkodę w upowszechnieniu się X. Dlatego rozpoczęto pracę nad dwoma projektami: środowiskiem K Desktop Environment (KDE) oraz GNU Network Object Model Environment (GNOME). Każde z tych środowisk ma szeroki wybór aplikacji, od pasków zadań i menedżerów plików po gry i pakiety biurowe, napisane z użyciem tego samego zestawu narzędzi do tworzenia GUI i ściśle zintegrowane, aby stworzyć jednolite, spójne środowisko pracy.

Różnice między KDE i GNOME są w sumie niewielkie. Różnią się one od siebie, ponieważ każde używa innego zestawu narzędzi GUI. KDE stworzono z wykorzystaniem biblioteki Qt firmy Troll Tech AS, natomiast GNOME używa GTK, zestawu narzędzi pierwotnie napisanego dla programu do obróbki grafiki The GNU Image Manipulation Program (w skrócie GIMP). Będąc oddzielnymi projektami, KDE i GNOME mają różnych projektantów i programistów, którzy hołdują różnym stylom programowania i filozofiom. W obu przypadkach efekt ich prac jest taki sam: spójne, ściśle zintegrowane środowisko pracy i zbiór aplikacji. Funkcjonalność, użyteczność i czyste piękno KDE i GNOME nie mają sobie równych wśród innych systemów operacyjnych.

Oczywiście najlepsze jest to, że oba te zaawansowane środowiska są darmowe. To znaczy, że możesz zainstalować jedno lub oba (tak, naraz). Wybór należy do ciebie.

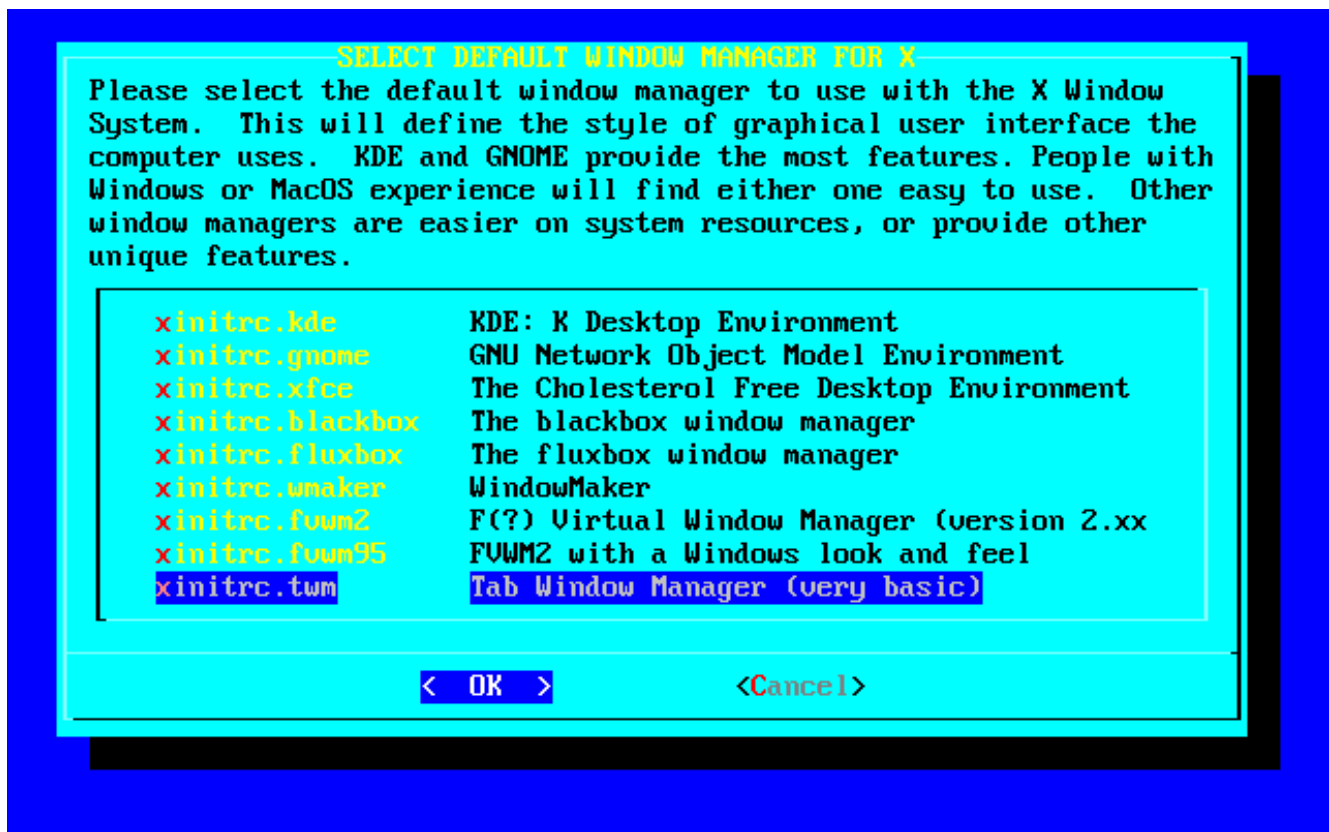
Oprócz KDE i GNOME, w Slackware jest dostępna duża liczba menedżerów okien. Niektóre zaprojektowano tak, aby emulowały (naśladowały) inne systemy operacyjne, inne tak, aby łatwo je było konfigurować, jeszcze inne tak, aby działały szybko. Jest ich dość sporo. Możesz oczywiście zainstalować tyle, ile chcesz, zapoznać się z nimi i wybrać ten, który Ci się najbardziej podoba.

Aby ułatwić wybór, w Slackware znajduje się program `xwmconfig`, którego można użyć do wyboru środowiska lub menedżera okien. Wpisz:

```
% xwmconfig
```

### Rys. 6-5 Konfiguracja pulpitu w `xorgconfig`

Wyświetli się lista wszystkich zainstalowanych środowisk i menedżerów okien. Wybierz z listy ten, który chcesz. Ten program będzie musiał uruchomić każdy użytkownik systemu, ponieważ każdy może chcieć używać różnych środowisk albo menedżerów okien, a nie każdemu będzie się podobał wybór domyślny, dokonany podczas instalacji.



Potem po prostu uruchom X i hulaj dusza, piekła nie ma.

## 6.5 xdm

W miarę rosnącej użyteczności Linuksa jako systemu na desktopach, wielu użytkowników chce, aby ich komputer po włączeniu od razu uruchamiał środowisko graficzne. Aby to osiągnąć, musisz ustawić Slackware tak, aby bezpośrednio uruchamiał X i wybrać graficzny menedżer logowania. W Slackware są dostępne trzy graficzne menedżery logowania: `xdm(1)`, `kdm` i `gdm(1)`.

Graficzny menedżer logowania `xdm` jest dostarczany wraz z systemem X.org. Dzięki temu jest zawsze dostępny, ale nie ma tak wielu funkcji, jak inne menedżery. Z KDE jest dostępny menedżer `kdm`, a z GNOME – `gdm`. Niezależnie od tego, który wybierzesz, będziesz mógł zalogować się jako dowolny użytkownik i wybrać środowisko lub menedżer, którego chcesz używać.

Slackware nie zawiera, niestety, takiego prostego programu, jak `xwmconfig`, który ułatwia wybór menedżera logowania. Jeśli więc zainstalowane są wszystkie trzy, będziesz musiał trochę poedytować pliki, aby dokonać wyboru. Najpierw jednak opiszemy, jak uruchomić środowisko graficzne po włączeniu komputera.

Aby uruchomić X po włączeniu komputera, musisz uruchomić go na poziomie pracy 4. Poziomy pracy to sposób przekazywania programowi `init(8)` różnych opcji po uruchomieniu systemu operacyjnego. W tym celu trzeba wyedytować plik konfiguracyjny `init - /etc/inittab`.



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

# Default runlevel. (Do not set to 0 or 6)
id:3:initdefault:
```

Aby Slackware uruchamiał się od razu ze środowiskiem graficznym, wystarczy zmienić 3 na 4.

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

Teraz Slackware będzie się uruchamiał na poziomie pracy 4 i wykonywał skrypt `/etc/rc.d/rc.4`. Ten plik uruchamia X i wywołuje wybrany menedżer okien. Więc jak się je wybiera? Istnieje kilka sposobów, wyjaśnię je po tym, jak się przyjrzymy plikowi `rc4`.

```
# 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, `rc.4` najpierw sprawdza, czy `gdm` jest plikiem wykonywalnym, a jeśli tak jest, uruchamia go. Kolejny na liście jest `kdm`, następnie `xdm`. Jednym ze sposobów wyboru menedżera logowania jest usunięcie tych menedżerów, których nie chcesz używać, za pomocą `removepkg`. Więcej informacji na temat `removepkg` znajduje się w rozdziale 18.

Możesz też usunąć prawa wykonywania z plików, których nie chcesz używać. Polecenie `chmod` jest omówione w rozdziale 9.

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

No i wreszcie możesz po prostu zakomentować wiersze dotyczące menedżerów logowania, których nie chcesz używać.

```

# 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

```

Wszystkie wiersze poprzedzone znakiem krzyżyka (#) są traktowane jak komentarze i powłoka pominię je. Dlatego, nawet jeśli plik `gdm` istnieje i jest wykonywalny, powłoka (w tym przypadku `bash`), nie będzie go sprawdzać.

## Rozdział 7. Uruchamianie

Proces uruchamiania Linuksa może być łatwy lub trudny. Większość użytkowników instaluje Slackware, włącza komputer i tyle. A czasem samo uruchomienie systemu może być ciężką, mozolną pracą. W większości przypadków najlepiej działa program rozruchowy LILO. Linux Slackware udostępnia dwa programy rozruchowe: LILO i Loadlin. LILO działa z partycji dysku twardego, głównego sektora rozruchowego lub dyskietki, dzięki czemu jest bardzo wszechstronne. Loadlin działa z wiersza poleceń DOS, wyłączając DOS i uruchamiając Linuksa.

Innym popularnym programem rozruchowym jest GRUB. Nie jest on jednak włączony do Slackware ani oficjalnie wspierany. Programy wchodzące w skład tej dystrybucji trzymają się zasady „wypróbowane i działające”. Wprawdzie GRUB działa dobrze i ma pewne funkcje niedostępne w LILO, jednak LILO zajmuje się rozruchem systemu w sposób stabilny i od dawna sprawdzony. GRUB jest nowszy i nie został wystarczająco przetestowany. Ponieważ nie dołączono go do Slackware, nie będziemy go tu opisywać. Jeśli chcesz używać GRUBa (być może został zainstalowany z inną dystrybucją Linuksa i chcesz używać go do uruchamiania wielu systemów), przeczytaj jego dokumentację.

Ta sekcja opisuje LILO i Loadlin, które są dołączone do Slackware. Wyjaśnia też kilka typowych scenariuszy uruchamiania wielu systemów i pomaga w ich konfiguracji.

### 7.1 LILO

Najczęściej używanym programem ładującym system jest LILO, czyli LInux LOader. Ma on wiele opcji konfiguracji i łatwo go używać do uruchamiania więcej niż jednego systemu.

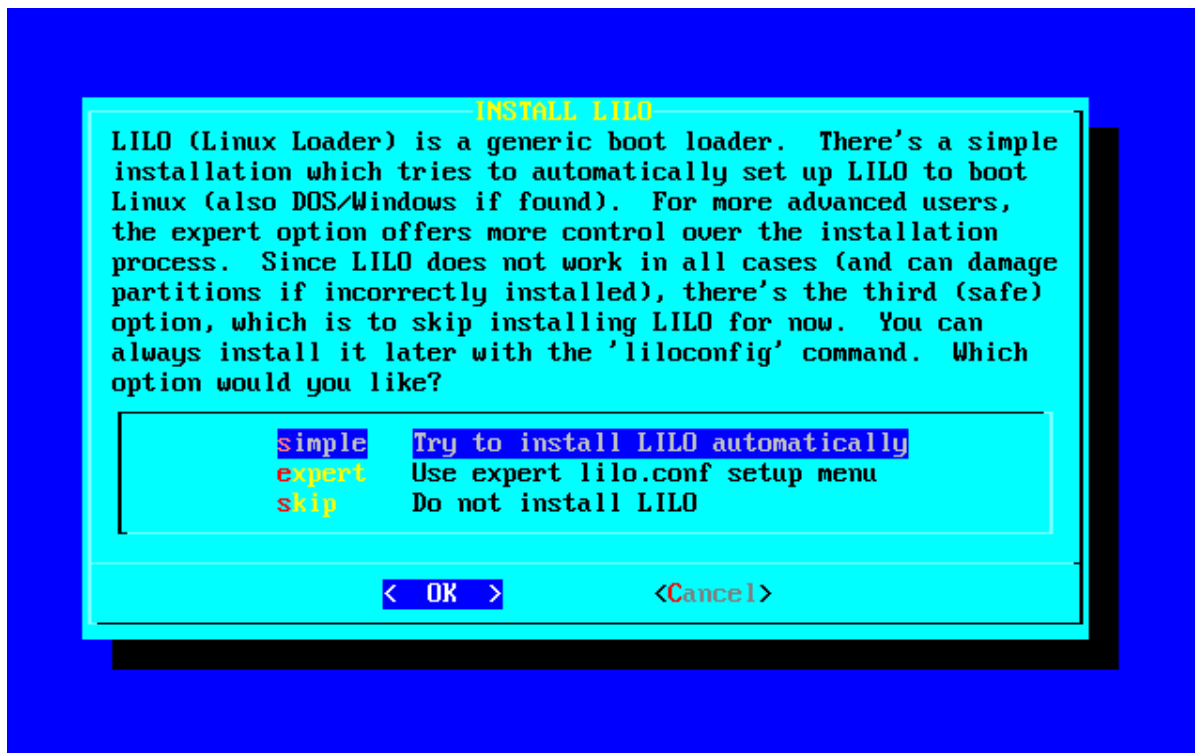
Do Slackware jest dołączony konfigurator LILO – `liloconfig`. Wykorzystuje on system menu. Jest uruchamiany po raz pierwszy w czasie instalacji, ale można go uruchomić w każdym momencie, wpisując w wierszu poleceń `liloconfig`.

LILO odczytuje konfigurację z pliku `/etc/lilo.conf(5)`. Nie dzieje się to podczas każdego uruchamiania systemu, ale podczas każdej instalacji LILO. LILO trzeba ponownie zainstalować w sektorze rozruchowym za każdym razem, gdy zmieni się jego konfigurację. Wiele błędów LILO bierze się z tego, że zmieniono konfigurację w `lilo.conf`, ale nie uruchomiono `lilo`, aby wprowadzić

zmiany. `liloconfig` pomaga w tworzeniu pliku konfiguracyjnego, aby instalacja LILO była możliwa. Jeśli chcesz modyfikować `/etc/lilo.conf` ręcznie, reinstalacja LILO polega wyłącznie na wpisaniu `/sbin/lilo` (jako `root`) w wierszu poleceń i wciśnięciu `enter`.

Podczas pierwszego uruchomienia `liloconfig` ekran wygląda tak:

Rys. 7-1. `liloconfig`



Jeśli nie instalowałeś wcześniej LILO, wybierz „simple” (instalacja prosta). Jeśli jesteś obeznany z Linuksem i LILO, opcja „expert” (dla zaawansowanych) będzie szybsza. Po wybraniu „simple” rozpocznie się konfiguracja LILO.

Jeśli w jądro jest wkompilewana obsługa bufora ramek, `liloconfig` zapyta, jakiej rozdzielczości wideo używać. Jest to rozdzielczość używana również przez bufor ramek serwera X. Jeśli nie chcesz, aby konsola działała w specyficznym trybie wideo, wybierz opcję „normal”, czyli standardowy tryb tekstowy 80x25.

Następne pytanie jest najważniejsze i dotyczy miejsca, w którym LILO ma być zainstalowany. Poniżej są objaśnione dostępne opcje:

#### *Root*

Ta opcja zainstaluje LILO na początku partycji `root`. Jest ona najbezpieczniejsza, jeśli masz zainstalowane inne systemy operacyjne. Daje pewność, że nie zostaną nadpisane inne programy rozruchowe. Wadą jest to, że LILO będzie się w stanie uruchomić tylko, jeśli dysk z Linuksem jest pierwszym dyskiem w komputerze. Dlatego wielu ludzi tworzy niedużą partycję `/boot` na początku dysku. Dzięki temu jądro i LILO mogą być zainstalowane na początku dysku, gdzie LILO

może je znaleźć. Starsze wersje LILO miały niesławną wadę znaną jako „ograniczenie 1024 cylindra”. Program LILO nie był w stanie załadować jądra, jeśli znajdowało się ono na partycji powyżej 1024 cylindra. W nowszych wersjach LILO problem został wyeliminowany.

### Floppy

Ta metoda jest jeszcze bezpieczniejsza niż poprzednia. Tworzy ona dyskietkę rozruchową, z której można uruchomić Linuksa. Program rozruchowy nie jest wówczas w ogóle instalowany na dysku twardym, dzięki czemu możesz uruchomić dyskietkę tylko wtedy, kiedy chcesz używać Slackware. Zasadniczą wadą tej metody jest nietrwałość dyskietek, które z zasady są podatne na uszkodzenia. Ponadto, jako że program rozruchowy nie znajduje się w komputerze, jeśli stracisz dyskietkę, będziesz musiał utworzyć nową, aby uruchomić system.

### MBR

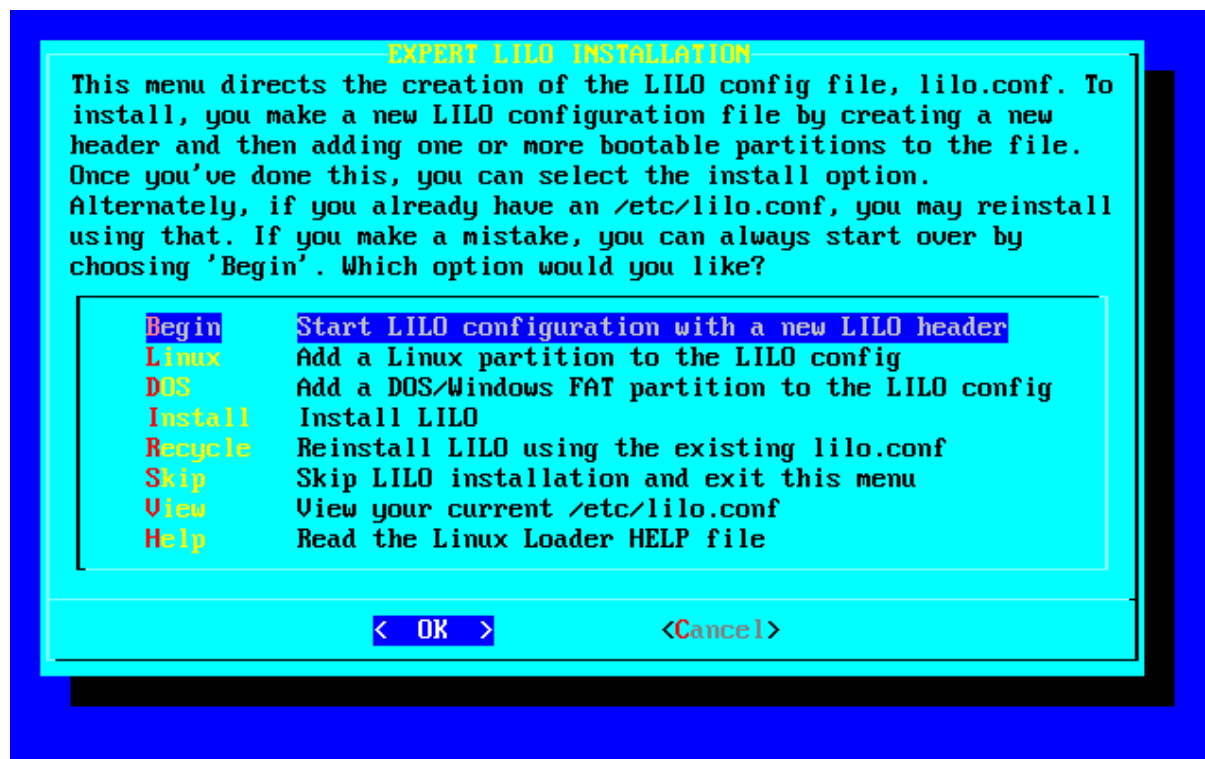
Jest to odpowiednia metoda, jeśli Slackware jest jedynym systemem w komputerze, lub jeśli zamierzasz używać LILO do wybierania systemu podczas startu. Jest to preferowana metoda instalacji LILO, która będzie działać na prawie każdym komputerze.



Ta opcja nadpisze inne programy zainstalowane w głównym sektorze rozruchowym.

Po wybraniu lokalizacji `liloconfig` utworzy plik konfiguracyjny i zainstaluje LILO. Jeśli wybierzesz opcję „expert”, wyświetli się specjalne menu. Pomoże ono w dostosowaniu `/etc/lilo.conf`, dodaniu innych systemów operacyjnych do menu rozruchowego, ustawieniu LILO, aby przekazywało parametry do jądra podczas startu. Oto ekran menu „expert”:

Rys. 7-2. Ekran menu „expert”



To menu kieruje utworzeniem pliku konfiguracyjnego LILO, lilo.conf. Aby go zainstalować, tworzysz nowy plik konfiguracyjny LILO, tworząc nowy nagłówek, a następnie dodając kolejne partycje rozruchowe. Potem możesz wybrać opcję install (instaluj). Jeśli plik /etc/lilo.conf już istnieje, możesz go wykorzystać do reinstalacji. Jeśli się pomyliłeś, możesz zawsze zacząć od nowa, wybierając begin (rozpocznij). Którą opcję chcesz wybrać?	
Begin (rozpocznij) Uruchom konfigurację LILO z nowym nagłówkiem LILO	
Linux Dodaj partycję linuxową do pliku konfiguracyjnego LILO	
DOS Dodaj partycję DOS/Windows FAT do pliku konfiguracyjnego LILO	
Install (instaluj) Zainstaluj LILO	
Recycle (Przeinstaluj) Zainstaluj LILO ponownie, używając istniejącego lilo.conf	
Skip (pomiń) Pomiń instalację LILO i wyjdź z tego menu	
View (przeglądaj) Przeglądaj obecną zawartość /etc/lilo.conf	
Help (pomoc) Przeczytaj plik pomocy dotyczący LILO	
<OK>	<Anuluj>

Niezależnie od systemu, stworzenie działającej konfiguracji programu rozruchowego jest dzięki `liloconfig` proste jak konstrukcja cepa.

## 7.2 LOADLIN

Inną opcją uruchamiania Linuksa Slackware jest LOADLIN. LOADLIN to program DOSowy, którego można użyć do uruchomienia Linuksa spod działającego systemu DOS. Jądro Linuksa musi znajdować się na partycji z DOSem, aby LOADLIN je załadował i uruchomił system.

Podczas instalacji LOADLIN jest kopiowany do domowego katalogu `roota` w postaci archiwum `.ZIP`. Nie ma automatycznego procesu instalacji. Należy ręcznie skopiować jądro Linuksa (zwykle `/boot/vmlinuz`) i plik LOADLIN z tego katalogu na partycję DOS.

LOADLIN przydaje się, jeśli chcesz utworzyć menu startowe na partycji DOS. Możesz je dodać do `AUTOEXEC.BAT`, co pozwoli na wybór między systemami. Wybór opcji Linux spowoduje uruchomienie LOADLIN, który załaduje Slackware. Oto plik `AUTOEXEC.BAT` dla Windows 95, który powinien wystarczyć:

nu:

```

@ECHO OFF
SET PROMPT=$P$G
SET PATH=C:\WINDOWS;C:\WINDOWS\COMMAND;C:\
CLS
ECHO Wybierz system operacyjny:
ECHO.
ECHO [1] Linux Slackware
ECHO [2] Windows 95
ECHO.
CHOICE /C:12 "Wybor? -> "
IF ERRORLEVEL 2 GOTO WIN
IF ERRORLEVEL 1 GOTO LINUX
:WIN
CLS
ECHO Uruchamianie Windows 95...
WIN
GOTO END
:LINUX
ECHO Uruchamianie Linuksa Slackware...
CD \LINUX
LOADLIN C:\LINUX\VMLINUX ROOT=<urządzenie z partycja root> RO
GOTO END
:END

```

Urządzenie z partycją `root` trzeba podać jako nazwę urządzenia linuksowego, np. `/dev/hda2`. `LOADLIN` można też zawsze uruchomić z wiersza poleceń. Używa się go tak samo, jak w powyższym przykładzie. Wiele przykładów zastosowania `LOADLIN` znajduje się w dołączonej do niego dokumentacji.

## 7.3 Uruchamianie wielu systemów

Wielu użytkowników konfiguruje komputery tak, aby było możliwe uruchamianie Linuksa Slackware i innego systemu operacyjnego. Jeśli masz problemy z właściwą konfiguracją takiego rozwiązania, poniżej opisaliśmy kilka typowych scenariuszy.

### 7.3.1 Windows

Konfiguracja maszyny z MS Windows i Linuksem to prawdopodobnie najczęstszy scenariusz. Istnieje wiele sposobów podobnej konfiguracji, ta sekcja opisuje dwa z nich.

Podczas instalacji wielu systemów często zdarza się, że użytkownik dokładnie zaplanował, co gdzie umieścić, ale pokręciła mu się kolejność czynności. Trzeba pamiętać, że systemy operacyjne należy instalować w określonej kolejności, bo inaczej możliwość uruchamiania wielu systemów nie będzie działać poprawnie. Linux zawsze daje pełną kontrolę nad tym, co, i czy w ogóle, jest instalowane w MBR. Dlatego zaleca się instalację Linuksa jako ostatniego. Windows należy instalować na początku, ponieważ zawsze zapisuje on swój program rozruchowy w MBR, nadpisując wszystkie dane, jakie Linux ewentualnie tam wcześniej zapisał.

#### 7.3.1.1 Wykorzystywanie LILO

W większości przypadków możliwością wyboru Linuksa lub Windows podczas startu zarządza LILO. Jak wspomniano powyżej, należy najpierw zainstalować Windows, potem Linuksa.

Załóżmy, że masz dysk twardy IDE 40 GB i jest to jedyny dysk w komputerze. Połowę pojemności chcesz przeznaczyć dla Windows, połowę dla Linuksa. Może pojawić się problem z uruchomieniem Linuksa.

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

Musisz też przeznaczyć pewną ilość miejsca na linuksową partycję wymiany (swap). Niepisana zasada mówi, że partycja wymiany powinna być dwa razy większa od pojemności pamięci RAM. Jeśli masz RAM 64 MB, partycja powinna mieć wielkość 128 MB itd. Właściwy rozmiar swap to przedmiot wielu zażartych dyskusji w Usenecie i na IRCu. Tak naprawdę nie ma złotej zasady, ale powyższa powinna wystarczyć.

Po podzieleniu dysku na partycje zainstaluj Windows. Gdy już będzie działał, zainstaluj Linuksa. Konfiguracja LILO będzie wymagała specjalnych zabiegów, więc wybierz tryb „expert” przy instalacji LILO.

Uruchom konfigurację LILO. Zainstaluj go w MBR, aby można go było użyć do wybierania systemu. W menu dodaj partycje z Linuksem i Windows (lub DOSem). Następnie zainstaluj LILO.

Uruchom komputer ponownie. Program LILO powinien się załadować i wyświetlić menu wyboru zainstalowanych systemów operacyjnych. Wybierz odpowiednią nazwę (nazwy zostały zdefiniowane podczas konfiguracji LILO).

LILO udostępnia wiele opcji konfiguracyjnych. Nie ogranicza się do uruchamiania Linuksa lub DOSa. Można z niego uruchomić prawie każdy system. Więcej informacji znajduje się na stronach podręcznika `lilo(8)` i `lilo.conf(5)`.

A jeśli LILO nie działa? Może się tak zdarzyć na niektórych komputerach. Na szczęście istnieją inne sposoby na uruchamianie Linuksa i Windows.

### **7.3.1.2 LOADLIN**

Można użyć tej metody, jeśli LILO nie działa lub w ogóle nie chcesz go instalować. Jest ona idealna dla tych, którzy często reinstalują Windows. Windows nadpisuje MBR podczas każdej instalacji, zamazując LILO. Dzięki LOADLIN nie ma takiego problemu. Największą wadą jest to, że za pomocą LOADLIN można ładować tylko Linuksa.

Przy LOADLIN można instalować systemy w dowolnej kolejności. Uważaj, aby nie instalować świadomie nic w MBR, ponieważ LOADLIN wymaga, aby partycja z Windows była partycją startową. Podczas instalacji Slackware pomiń krok instalacji LILO.

Po zainstalowaniu obu systemów skopiuj plik `loadlinX.zip` („X” oznacza numer wersji, np. *16a*) z domowego katalogu `roota` na partycję Windows. Skopiuj tam też jądro Linuksa. Musisz to robić, kiedy uruchomiony jest Linux. Oto przykładowe polecenia:

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

To utworzy katalog C:\LINUX na partycji Windows (zakładając, że jest to /dev/hda1) i skopiuje do niego pliki potrzebne dla LOADLIN. Następnie musisz przeładować komputer i uruchomić Windows, aby skonfigurować menu startowe.

Z poziomu Windows uruchom okno wiersza poleceń DOS. Najpierw musimy się upewnić, że system nie uruchamia się od razu w trybie graficznym

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

Dodaj do tego pliku następujący wiersz:

```
BootGUI=0
```

Zapisz plik i zakończ działanie edytora. Następnie zmodyfikuj AUTOEXEC.BAT, dodając menu startowe. Oto przykład odpowiednich wpisów:

```
cls
echo Menu wyboru systemow
echo.
echo 1 - Linux
echo 2 - Windows
echo.
choice /c:12 "Wybor? -> "
if errorlevel 2 goto WIN
if errorlevel 1 goto LINUX
:LINUX
cls
echo "Uruchamianie Linuksa..."
cd \linux
loadlin c:\linux\vmlinuz root=/dev/hda2 ro
goto END
:WIN
cls
echo "Uruchamianie Windows..."
win
goto END
:END
```

Kluczowym wierszem jest wpis uruchamiający LOADLIN. Mówimy mu, aby załadował jądro, gdzie jest partycja root i aby była ona początkowo zamontowana w trybie tylko do odczytu.

Te dwa narzędzia są dostarczane z Linuksem Slackware. Istnieje wiele innych programów rozruchowych,



ale wyżej wymienione powinny wystarczyć dla większości komputerów z dwoma systemami.

### 7.3.1.3 Stary patent z Windows NT

Jest to najrzadziej spotykana sytuacja uruchamiania 2 systemów. W zamierzonych czasach LILO nie potrafił uruchomić Windows NT. To wymagało od użytkowników Linuksa przerobienia NTLDR, co z kolei było o wiele bardziej problematyczne niż wybór między Linuksem a Windows 9x. Pamiętaj, że poniższe instrukcje są przestarzałe i nie mają już zastosowania. LILO już od wielu lat jest w stanie uruchomić Windows NT/2000/XP/2003. Jeśli jednak masz stary komputer, możesz potrzebować takiego patentu.

1. Zainstaluj Windows NT.
2. Zainstaluj Linuksa. Upewnij się, że LILO jest umieszczony w superbloku partycji linuksowej.
3. Skopiuj pierwsze 512 bajtów linuksowej partycji `root` na partycję Windows NT.
4. Zmodyfikuj `C:\BOOT.INI` w Windows NT, aby dodać opcję uruchamiania Linuksa.

Instalacja Windows NT i Linuksa powinna być prosta. Dalej zaczynają się schody. Skopiowanie pierwszych 512 bajtów partycji linuksowej jest prostsze niż to brzmi. Trzeba to wykonać z poziomu Linuksa. Zakładając, że linuksowa partycja `root` to `/dev/hda2`, wpisz następujące polecenie:

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

To jest to. Teraz skopiuj `bootsect.lnx` na partycję Windows NT. To kolejny problem. Linux nie ma stabilnej obsługi zapisu w systemie plików NTFS. Jeśli dysk z Windows NT sformatowałeś podczas instalacji jako NTFS, musisz skopiować ten plik na dyskietkę FAT i odczytać ją pod Windows NT. Jeśli partycja z NT jest sformatowana jako FAT, wystarczy ją zamontować pod Linuksem i normalnie skopiować plik. Tak czy owak, plik `/tmp/bootsect.lnx` musi zostać skopiowany z Linuksa do `C:\BOOTSECT.LNX` na dysku z Windows NT.

Na koniec trzeba dodać opcję menu startowego Windows NT. Z poziomu NT uruchom okno wiersza poleceń i wpisz:

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

Na końcu pliku `boot.ini` dodaj wiersz

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

Zapisz plik i zakończ pracę edytora. Po ponownym uruchomieniu Windows NT pojawi się w menu opcja wyboru Linuksa. Jej wybranie spowoduje uruchomienie Slackware.

## 7.3.2 Linux

Tak, ludzie tak robią. Jest to najprostszy scenariusz. Wystarczy zmodyfikować `/etc/lilo.conf`, dodając odpowiednie wpisy, i do uruchamiania różnych Linuksów używać LILO.

# Rozdział 8. Powłoka

W środowisku graficznym interfejs jest zapewniany przez program, który tworzy okna, paski przewijania, menu itp. W środowisku tekstowym interfejs stanowi „powłoka” (ang. „shell”), która interpretuje polecenia i generalnie umożliwia korzystanie z systemu. Natychmiast po zalogowaniu (co jest opisane w tym rozdziale) użytkownicy znajdują się w powłoce i mogą zająć się swoimi sprawami. Ten rozdział stanowi wprowadzenie do powłoki, szczególnie do najczęściej używanej w Linuksie powłoki Bourne Again Shell (bash). Więcej informacji na opisane tu tematy znajduje się na stronie podręcznika `bash(1)`.

## 8.1 Użytkownicy

### 8.1.1 Logowanie

No dobra. Uruchomiłeś komputer i patrzysz na coś podobnego do tego:

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

Hmm, nikt nic nie mówił o żadnym loginie. I co to jest ta mroczna gwiazda (ang. darkstar)? Nie martw się. Raczej nie otwarłeś przez pomyłkę hiperprzestrzennego łącza komunikacyjnego z imperialną Gwiazdą Śmierci. (Obawiam się, że jądro Linuksa nie obsługuje jeszcze protokołu komunikacji hiperprzestrzennej. Może będzie dostępna w wersji 2.8.) Darkstar to nazwa jednego z naszych komputerów. Używamy jej jako domyślnej nazwy każdego komputera z nowo zainstalowanym Slackware. Jeśli podczas instalacji nadałeś swojej maszynie inną nazwę, będzie ona widoczna zamiast darkstar.

Co do logowania... jeśli logujesz się po raz pierwszy, musisz się zalogować jako `root`. Wyświetli się monit o wprowadzenie hasła. Wpisz hasło, które ustaliłeś podczas instalacji. Jeśli nie masz żadnego hasła, po prostu wciśnij enter. Gotowe!

### 8.1.2 Root czyli Superużytkownik.

Fajnie. Kim lub *czym* jest ten cały „`root`”? I co robi jego konto w *twoim* systemie?

W świecie Uniksa i podobnych systemów operacyjnych (a więc i Linuksa) istnieją użytkownicy... i użytkownicy. Bardziej szczegółowo opiszemy to później. Najważniejsze jest jednak wiedzieć, że `root` stoi ponad wszystkimi innymi użytkownikami. Jest wszechpotężny i wszechwiedzący, *wszyscy* go słuchają. Nie wolno go nie słuchać. `Root` jest tym, czym go nazywamy – superużytkownikiem. A teraz najlepsze. `Rootem` jesteś *Ty*.

Zarabicie, co nie?

Jeśli nie jesteś tego pewien, odpowiemy: tak, to naprawdę zarabicie. Haczyk polega na tym, że `rootowi` z natury wolno zniszczyć wszystko, czego zapragnie. Możesz przejść do rozdziału 12.1.1, przeczytać o dodawaniu użytkowników, zalogować się jako inny użytkownik i pracować na jego koncie. Mądrość ludowa głosi, że należy logować się jako superużytkownik tylko wtedy, kiedy jest to naprawdę potrzebne, aby zminimalizować możliwość uszkodzenia czegoś przez pomyłkę.

Tak przy okazji, jeśli chcesz stać się `rootem`, podczas gdy jesteś zalogowany jako inny użytkownik, żaden problem. Użyj polecenia `su(1)`. Wyświetli się monit o podanie hasła i zostaniesz `rootem` do momentu, kiedy wylogujesz się poleceniem `exit` lub `logout`. Za pomocą `su` możesz zostać każdym użytkownikiem, jeśli znasz jego hasło. Na przykład, `su logan` sprawi, że staniess się mną.

Uwaga: `root` może stać się każdym użytkownikiem bez potrzeby podawania jego hasła.

## 8.2 Wiersz poleceń

### 8.2.1 Uruchamianie programów

Cieżko cokolwiek osiągnąć bez uruchamiania programów. Możesz użyć komputera jako podpórki albo przycisku do papieru. Niektóre komputery wydają przecudny mruczący dźwięk, kiedy są włączone, ale tak naprawdę bez programów są nieprzydatne. Chyba wszyscy zgodzimy się, że mruczająca podpórka to nie zastosowanie, które przyniosło komputerom osobistym tak wielką popularność.

Pamiętasz, jak pisaliśmy, że prawie wszystko w Linuksie jest plikiem? To samo dotyczy programów. Każdy program, który uruchamiasz (a który nie jest wbudowany w powłokę), znajduje się gdzieś w postaci pliku. Uruchamiasz go, podając po prostu pełną ścieżkę do niego.

Na przykład, pamiętasz polecenie `su` z poprzedniej sekcji? Tak naprawdę znajduje się ono w katalogu `/bin`. `/bin/su` działa tak samo dobrze.

To dlaczego działa wpisywanie samego `su`? Przecież nie napisaliście, że jest w `/bin`. Mogłoby równie dobrze być w `/usr/local/share`, prawda? Skąd system to *wiedział*? Odpowiedź leży w zmiennej środowiskowej `PATH` (ścieżka). Większość powłok ma taką `PATH` (lub coś podobnego). `PATH` jest zasadniczo listą katalogów, w których system szuka programów do uruchomienia. Kiedy uruchomiłeś `su`, powłoka sprawdziła katalogi wymienione na liście w poszukiwaniu wykonywalnego pliku o nazwie `su` i uruchomiła pierwszy, który znalazła. Dzieje się tak za każdym razem, kiedy uruchamiasz program bez podania pełnej ścieżki. Jeśli wyświetla się błąd `Command not found` (Nie znaleziono polecenia), oznacza to, że programu, który chciałeś uruchomić, nie ma w `PATH`. (Oczywiście dzieje się tak, kiedy programu w ogóle nie ma...) Zmienne środowiskowe są omówione dokładniej w sekcji 8.3.1.

Pamiętaj, że „.” (kropka) jest skrótem oznaczającym katalog bieżący. Jeśli więc jesteś akurat w katalogu `/bin`, napisanie `./su` zadziała jak wpisanie pełnej ścieżki.

### 8.2.2 Dopasowywanie symboli wieloznacznych

Prawie każda powłoka rozpoznaje pewne znaki jako zastępniki lub skróty oznaczające „wszystko pasuje”. Takie znaki nazywają się „symbolami wieloznacznymi” (ang. wildcards). Najczęściej używane są `*` i `?`. Tradycyjnie `?` oznacza pojedynczy znak. Na przykład: jesteś w katalogu, gdzie znajdują się trzy pliki: `ex1.txt`, `ex2.txt` i `ex3.txt`. Chcesz je wszystkie skopiować do katalogu `/tmp` (za pomocą polecenia `cp`, opisanego w sekcji 10.5.1). Możesz wpisać `cp ex1.txt ex2.txt ex3.txt /tmp`, ale to kupa roboty. O wiele prościej wpisać `cp ex?.txt /tmp`. `?` będzie pasował do każdego znaku „1”, „2” i „3”, każdy z nich będzie podstawiony po kolei.

Co mówisz? *Jeszcze* za dużo roboty? Masz rację. To okropne. Istnieją odpowiednie regulacje prawne

przeciwdziałające takiemu wyzyskowi. Na szczęście jest jeszcze \*. Jak już wspomnieliśmy, \* oznacza „dowolna liczba znaków”, także 0 znaków. Jeśli więc te trzy pliki są jedynymi w katalogu, do skopiowania wszystkich naraz wystarczy `cp * /tmp`. Załóżmy jednak, że są tam też pliki `ex.txt` i `hejaz.txt`. Chcesz skopiować `ex.txt`, ale nie `hejaz.txt`; `cp ex* /tmp` załatwi sprawę.

`cp ex?.txt /tmp` skopiowałoby tylko trzy pliki wspomniane na początku. W `ex.txt` nie ma znaku pasującego do `?`, a więc pozostałby nieskopiowany.

Inne często używane symbole wieloznaczne to para nawiasów kwadratowych `[]`. Wszystkie znaki znajdujące się wewnątrz `[]` zostaną podstawione w ich miejsce. Namieszaliśmy? Nie jest tak źle. Załóżmy, że mamy katalog z 8 plikami: `a1`, `a2`, `a3`, `a4`, `aA`, `aB`, `aC` i `aD`. Chcemy znaleźć tylko pliki kończące się cyframi. `[]` to zrobią.

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

A jeśli chcemy tylko `a1`, `a2` i `a4`? W poprzednim przykładzie użyliśmy – dla oznaczenia wszystkich wartości od 1 do 4. Możemy też oddzielać poszczególne pozycje przecinkami.

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

Wiem, co teraz myślisz. „A co z literami?” Linux rozróżnia wielkie i małe litery. To znaczy, że `a` i `A` to dwa zupełnie różne znaki, tylko my kojarzymy je ze sobą. Wielkie litery idą zawsze przed małymi, tak więc `A` i `B` znajdują się przed `a` i `b`. Kontynuując poprzedni przykład, jeśli chcemy plików `a1` i `A1`, szybko je znajdziemy przy użyciu `[]`.

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

Zauważ, że użycie myślnika zamiast przecinka dałoby niewłaściwy wynik.

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

Można też łączyć myślniki i przecinki.

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

## 8.2.3 Przekierowywanie wejścia/wyjścia i potokowanie

(Teraz coś naprawdę zarąbistego)

```
% ps > blargh
```

Wiesz, co to znaczy? To ja wykonałem `ps`, żeby zobaczyć, jakie procesy są teraz uruchomione. Program `ps` jest opisany w sekcji 11.3. Ale to nie jest zarąbista część. Zarąbista część to

> *blargh*, która oznacza w skrócie „weź to, co wyświetlił na ekranie program `ps` i zapisz w pliku o nazwie `blargh`”. Ale dalej będzie jeszcze zarąbiściej.

```
% ps | less
```

To bierze „wyjście” czyli wyświetlony wynik `ps` i „potokuje”, czyli „przepuszcza” je przez `less`, tak że można je dowolnie przewijać w górę i w dół.

```
% ps >> blargh
```

To trzecie najczęściej używane przekierowanie. Robi to samo, co „>”, tyle że „>>” doda wynik polecenia `ps` na końcu pliku `blargh`, jeśli taki plik istnieje. Jeśli nie, zostanie utworzony, jak przy „>”. („>” zamaże poprzednią zawartość pliku `blargh`.)

Istnieje też operator „<”, który oznacza „pobierz dane wejściowe z...” Używa się go trochę rzadziej.

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

Zabawa z przekierowywaniem zaczyna się na dobre, kiedy łączy się kilka operatorów.

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

To uruchomi program `ps`, odwróci kolejność wierszy w jego wyniku i dopisze je na końcu `blargh`. Można łączyć dowolnie wiele operatorów, pamiętaj tylko, że są interpretowane od lewej do prawej.

Więcej informacji na temat przekierowywania znajduje się na stronie podręcznika `bash(1)`.

## 8.3 Bourne Again Shell (*bash*)

### 8.3.1 Zmienne środowiskowe

Linux to skomplikowane bydlę. Trzeba pamiętać o wielu drobiazgach, które mają znaczenie dla normalnej pracy z programami (niektórych z tych drobiazgów prawdopodobnie nigdy nie będziesz musiał poznać). Nikt nie spamięta kupy różnych opcji każdego programu ani rodzaju terminalu, który przekazuje się do uruchamianego programu, nazwy komputera, wyglądu znaku zgłoszenia...

Wszystkim tym zawiaduje mechanizm znany jako środowisko. Środowisko określa, w jakich warunkach działają programy. Część tych warunków jest zmienna. Użytkownicy mogą je zmieniać i dostosowywać, jak to w systemach linuksowych bywa. Praktycznie każda powłoka ma swoje zmienne środowiskowe (jeśli nie ma, prawdopodobnie do niczego się nie nadaje). Oto opis poleceń, które powłoka `bash` udostępnia w celu manipulowania zmiennymi środowiskowymi.

Polecenie `set`, użyte bez opcji, wyświetla wszystkie obecnie zdefiniowane zmienne oraz ich wartości. Podobnie jak większość wbudowanych poleceń `bash`, może też (przy wykorzystaniu różnych parametrów) wykonywać inne rzeczy. Szczegóły są na stronie podręcznika `bash(1)`. Oto przykładowy fragment wyniku wykonania polecenia `set` na jednym z komputerów autora. Zwróć uwagę na opisaną wcześniej zmienną `PATH`. Programy znajdujące się w jednym z tych katalogów uruchamia się, po prostu

wpisując nazwę pliku.

### Przykład 8-1. Wyświetlanie zmiennych środowiskowych poleceniem `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 ZMIENNA
```

Polecenie `unset` usuwa zmienną podaną jako parametr oraz jej wartość. Powłoka zapomina, że taka zmienna kiedykolwiek istniała. (Nie przejmuj się, jeśli nie jest to zmienna ustalona tylko w danej sesji powłoki, zostanie zapewne ponownie zdefiniowana w nowej sesji.)

```
% export ZMIENNA=jakaś_wartość
```

Polecenie `export` jest bardzo przydatne. Dzięki niemu nadajesz zmiennej środowiskowej o nazwie `ZMIENNA` wartość równą „`jakaś_wartość`”. Jeśli `ZMIENNA` nie istniała, zostanie utworzona. Jeśli istniała i miała inną wartość, cóż... już nie ma tej wartości. Nie jest to szczególnie dobre, jeśli tylko próbujesz dodać jakiś katalog do `PATH`. W takim przypadku wykonaj:

```
% export PATH=$PATH:/jakiś/nowy/katalog
```

Zwróć uwagę na `$PATH`. Kiedy chcesz, żeby `bash` zinterpretowała zmienną (czyli zastąpiła nazwę wartością), dodaj `$` przed nazwą zmiennej. Na przykład, `echo $PATH` wyświetli wartość `PATH`, w moim przypadku:

```
% 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 Uzupełnianie tabulatorem

(Teraz znowu coś zarąbistego)

1. Interfejs wiersza poleceń oznacza dużo klepania w klawisze.
2. Klepanie oznacza pracę.
3. Nikt nie lubi pracy.

Na podstawie punktu 2. i 3. można dopisać punkt 4. – nikt nie lubi klepania w klawisze. Na szczęście

dzięki `bash` nie ma punktu 5. (nikt nie lubi interfejsu wiersza poleceń).

No i jak niby `bash` wykonuje za nas pracę klepania w klawisze? Oprócz symboli wieloznacznych, o których pisaliśmy wcześniej, `bash` ma funkcję „uzupełniania tabulatorem” (ang. `tab completion`).

Uzupełnianie tabulatorem działa tak: Wpisujesz nazwę pliku. Być może jest on w `PATH`, może piszesz pełną ścieżkę. Wystarczy tylko wpisać kilka pierwszych znaków z nazwy, aby wiadomo było, że chodzi o dany plik. Po wciśnięciu klawisza tabulatora `bash` domyśli się, o jaki plik chodzi i dokończy za Ciebie wpisywanie!

Czas na przykład. W katalogu `/usr/src` są dwa podkatalogi: `/usr/src/linux` i `/usr/src/sendmail`. Chcę zobaczyć zawartość `/usr/src/linux`. Wpisuję więc `ls /usr/src/l`, wciskam klawisz **TAB**, a `bash` dopełnia resztę i daje `ls /usr/src/linux`.

Załóżmy, że mamy dwa katalogi: `/usr/src/linux` i `/usr/src/linux-stare`. Jeśli wpiszę `/usr/src/l` i wcisnę **TAB**, `bash` uzupełni tyle, ile będzie mogła, i wyświetli się `/usr/src/linux`. Mogę poprzestać na tym albo wcisnąć **TAB** jeszcze raz, wtedy `bash` wyświetli listę katalogów, które pasują do dotychczas wpisanej nazwy.

To oznacza mniej klepania w klawisze (co z kolei oznacza, że ludzie mogą polubić wiersz poleceń). Mówiłem, że to zarąbiste.

## 8.4 Terminale wirtualne

Wyobraź sobie, że akurat pracujesz nad czymś i chcesz zacząć robić coś innego. Możesz porzucić obecne zadanie i zająć się innym, ale to przecież system wieloużytkownikowy, nieprawdaż? Możesz zalogować się wielokrotnie w tym samym czasie, wiesz? Po cóż więc miałbyś robić tylko jedną rzecz naraz?

Nie musisz tak robić. Co prawda większość użytkowników nie może i nie chce mieć wielu klawiatur, myszy i monitorów podłączonych do jednego komputera. Tak naprawdę, to nie kwestia sprzętu. Zadaniem oprogramowania, a więc Linuksa, jest stworzenie „terminali wirtualnych”, czyli „VT” (ang. `virtual terminals`).

Wciskając **Alt** i jeden z klawiszy funkcyjnych, przełączasz się między terminalami wirtualnymi. Każdy klawisz funkcyjny odpowiada jednemu z nich. Slackware domyślnie umożliwia logowanie na 6 VT. **Alt+F2** przełącza na drugi, **Alt+F3** na trzeci itd.

Pozostałe klawisze funkcyjne są zarezerwowane dla sesji X. Każda sesja X zajmuje jeden VT, rozpoczynając od siódmego (**Alt+F7**) w górę. Kiedy jesteś w X, zastępujesz kombinację **Alt+Fn** przez **Ctrl+Alt+Fn**. Na przykład, będąc w X i chcąc przejść do logowania tekstowego (bez opuszczania X), wciskasz **Ctrl+Alt+F3**, aby przełączyć się na trzeci VT. (**Alt+F7** powróci do X, zakładając, że masz uruchomioną tylko jedną sesję X.)

### 8.4.1 Program `screen`

A jeśli terminale wirtualne nie są dostępne? Na szczęście Slackware ma wspaniały menedżer ekranów, nazwany odpowiednio `screen` (ekran). `Screen` to emulator terminala, posiadający właściwości terminali wirtualnych. Po uruchomieniu `screen` wyświetla krótki tekst informacyjny i tworzy terminal.

W przeciwieństwie do standardowych VT, `screen` ma własny zestaw poleceń. Wszystkie polecenia `screen` poprzedza się kombinacją `Ctrl+A`. Na przykład, `Ctrl+A+N` tworzy nową sesję terminala, `Ctrl+A+N` przelacza do następnego terminala, `Ctrl+A+P` – do poprzedniego.

`Screen` umożliwia również odłączanie się i ponowne podłączanie do sesji `screen`, co jest szczególnie przydatne w przypadku zdalnych sesji poprzez `ssh` lub `telnet` (więcej na ich temat jest napisane później). `Ctrl+A+D` odłącza od bieżącej sesji `screen`. Wpisanie `screen -r` wyświetla wszystkie aktualnie uruchomione sesje, 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.
```

Wpisanie `screen -r 1212` spowoduje podłączenie do pierwszej wymienionej sesji. Wspomniałem wcześniej, że przydaje się to w sesjach zdalnych. Na przykład, gdybym był zalogowany do zdalnego serwera Slackware przez `ssh` i moje połączenie zostało przerwane z jakiegoś powodu, np. lokalnej awarii zasilania, wszystko, co robiłem, zostałoby unicestwione. Mogłoby to się marnie skończyć dla serwera. `Screen` zapobiega podobnym sytuacjom, ponieważ gdy połączenie zostaje zerwane, sesja jest automatycznie odłączana. Po przywróceniu połączenia mogę się do niej podłączyć ponownie i kontynuować pracę.

## Rozdział 9. Struktura systemu plików

Omówiliśmy już strukturę katalogów w Linuksie Slackware. Potrafisz znaleźć pliki i katalogi. Ale system plików to więcej niż struktura katalogów.

Linux to wieloużytkownikowy system operacyjny. Każdy aspekt systemu, także system plików, uwzględnia istnienie wielu użytkowników. System przechowuje informacje, np. kto jest właścicielem danego pliku i kto może go odczytać. Istnieją też specyficzne pojęcia, takie jak dowiązania i udziały NFS. W tej sekcji wyjaśniamy je oraz inne aspekty systemu wieloużytkownikowego.

### 9.1 Prawa własności

System plików zawiera informacje o prawach własności każdego pliku i katalogu, m.in. jaki użytkownik i jaka grupa są właścicielami danego pliku. Najprościej wyświetlić te informacje poleceniem `ls`:

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

Interesują nas kolumny trzecia i czwarta. Zawierają one nazwę użytkownika i nazwę grupy, którzy są właścicielami pliku. Widzimy, że są to: użytkownik „root” i grupa „bin”.

Prawa własności pliku zmienia się poleceniami: `chown(1)` (od „change owner” – „zmień właściciela”) i



chgrp(1) (od „change group” – „zmiń grupę”). Aby zmienić właściciela pliku na użytkownika „daemon”, użyjemy chown:

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

Aby zmienić grupę właścicieli na „root”, użyjemy chgrp:

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

Aby zmienić naraz właściciela i grupę, użyjemy chown:

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

W powyższym przykładzie można użyć kropki zamiast dwukropka. Efekt będzie ten sam, jednak zaleca się używanie dwukropka. Wykorzystanie kropki jest przestarzałe i prawdopodobnie zostanie usunięte z przyszłych wersji chown, aby umożliwić istnienie nazw użytkowników zawierających kropkę. Tego typu nazwy użytkowników są powszechnie stosowane w serwerach Windows Exchange i najczęściej spotyka się je w adresach e-mail, np. *wilus.bramkowski@przyklad.com*. W Slackware administratorzy powinni unikać takich nazw, ponieważ niektóre skrypty używają kropki do rozróżnienia właściciela i grupy. W naszym przykładzie polecenie chmod zinterpretowałoby *wilus.bramkowski* jako użytkownika „wilus” i grupę „bramkowski”.

Prawa własności to bardzo ważna część systemu plików Linuksa, nawet jeśli jesteś jedynym użytkownikiem. Czasami będziesz potrzebował zmieniać prawa własności plików i urządzeń.

## 9.2 Prawa dostępu

Prawa dostępu to kolejny ważny aspekt systemu wieloużytkownikowego. Decydują one, kto może dany plik odczytać, uruchomić i zmienić jego zawartość.

Informacje o prawach dostępu są zapisane jako cztery liczby w systemie ósemkowym. Każda oznacza poszczególny zestaw praw. Istnieją prawa właściciela, prawa grupy i prawa globalne (ang. world). Czwartą liczbą ósemkową oznacza specyficzne prawo, takie jak ustawiony numer użytkownika (set user ID), ustawiony numer grupy (set group ID) oraz tzw. bit „kleisty” (ang. sticky). Oto wartości ósemkowe przypisane do trybów praw dostępu (mają także odpowiednie oznaczenia literowe, które można wyświetlić przez `ls` i zmienić przez `chmod`):

**Tabela 9.1** Ósemkowe wartości praw dostępu

Typ prawa	Wartość ósemkowa	Oznaczenie literowe
bit „kleisty”	1	t
set user ID	4	s
set group ID	2	s
odczyt	4	r
zapis	2	w

wykonanie	1	x
-----------	---	---

Dla każdej grupy praw można dodać wartości ósemkowe. Na przykład, aby ustawić prawa grupy do odczytu i zapisu, należy użyć „6” w części dotyczącej praw grupy.

Domyślne prawa powłoki bash to:

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

Pierwszy myślnik będzie zastąpiony przez „d”, jeśli będą wyświetlane informacje o katalogu. W następnej kolejności są trzy grupy praw (właściciel, grupa, globalne). Widać, że właściciel ma prawo do odczytu, zapisu i wykonania (*rwx*), a grupa ma tylko prawa odczytu i wykonania (*r-x*). Wszyscy pozostali mają prawa odczytu i wykonania (*r-x*).

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

Użyjemy `chmod(1)` (change mode – zmień tryb), aby ustawić prawa dostępu do przykładowego pliku. Dodajmy wartości ósemkowe dla żądanych praw. Aby właściciel miał prawo odczytu, zapisu i wykonania, łączna wartość wyniesie 7. Wartość prawa odczytu i wykonania wyniesie 5. Połączmy je razem i przekażmy jako parametr do `chmod`:

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

Zastanawiasz się może „Dlaczego nowo utworzony plik nie miał od razu takich praw?” Odpowiedź jest prosta. Powłoka `bash` ma ciekawą wbudowaną funkcję, tzw. `umask`. Tę funkcję ma większość powłok uniksowych. Decyduje ona, jakie prawa są przyznawane nowo tworzonemu plikom. W sekcji 8.3.1 omówiliśmy pokrótce niektóre wbudowane funkcje `bash`, `umask` trzeba trochę poznać. Działa podobnie jak `chmod`, ale na odwrót. Ustawia się wartości ósemkowe, jakich nowo tworzone pliki nie mają mieć. Domyślna wartość `umask` to `0022`.

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

Więcej informacji na ten temat znajduje się na stronie podręcznika `bash`.

Aby ustawić prawa specjalne, dodaje się wartości ósemkowe i umieszcza w pierwszej kolumnie. Na przykład, aby plikowi nadać set user ID i set group ID, użyj 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
```

Jeśli nie za bardzo rozumiesz wartości ósemkowe, możesz w `chmod` użyć liter. Odpowiednie prawa są reprezentowane przez:

Właściciel	u
Grupa	g
Globalne	o
Wszyscy powyżej razem	a

Aby wykonać powyższe przykłady, użylibyśmy kilku poleceń:

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

Niektórzy użytkownicy wolą litery od wartości ósemkowych. Tak czy owak, zestaw praw będzie ten sam.

Format ósemkowy jest z reguły szybszy. Najczęściej używa się go w skryptach powłoki. Czasem jednak litery dają więcej możliwości. Na przykład, trudno zmienić jedną grupę praw, zachowując inne grupy, przy użyciu wartości ósemkowych. Użycie liter jest trywialnie proste.

```
% 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
```

Parokrotnie wspomnieliśmy set user ID i set group ID. Zastanawiasz się pewnie, co to jest. Zwykle, kiedy uruchamiasz program, jest on uruchamiany na twoim koncie użytkownika. To znaczy, że ma takie same prawa, jak użytkownik. To samo dotyczy grupy. Program jest wykonywany z prawami twojej grupy. Używając set user ID, możesz wymusić, aby program był zawsze uruchamiany z prawami swojego właściciela (takiego jak „root”). Set group ID oznacza to samo, ale dla grupy.

Należy ostrożnie używać tych praw specjalnych. Set user ID i set group ID mogą stanowić poważne zagrożenie dla bezpieczeństwa systemu. Jeśli set user ID ma wiele programów, których właścicielem jest root, pozwalasz każdemu uruchamiać je właśnie jako root. Root nie jest niczym ograniczony w systemie, widzisz więc, że stanowi to problem z bezpieczeństwem. Krótko mówiąc, set user ID i set group ID nie są złe, ale należy używać ich z głową.

### 9.3 Dowiązania

Dowiązania (ang. links) to wskaźniki do plików. Dzięki nim jeden plik może istnieć w wielu miejscach i mieć wiele nazw. Istnieją dwa typy dowiązań: twarde i miękkie.

Dowiązania twarde to różne nazwy tego samego pliku. Mogą istnieć tylko wewnątrz tego samego

systemu plików i być usunięte tylko, kiedy z systemu jest usunięta prawdziwa nazwa pliku. W niektórych przypadkach jest to przydatne, jednak większość użytkowników woli dowiązania miękkie, gdyż są bardziej wszechstronne.

Dowiązanie miękkie, zwane też dowiązaniem symbolicznym, może wskazywać na plik poza swoim systemem plików. Jest to w istocie mały plik z odpowiednimi informacjami. Można dodawać i usuwać dowiązania symboliczne, nie zmieniając właściwego pliku. Jako że dowiązanie symboliczne to niewielki plik zawierający własne informacje, może nawet wskazywać na katalog. Często np. `/var/tmp` jest dowiązaniem symbolicznym do `/tmp`.

Dowiązania nie mają zestawu praw własności ani dostępu. Odzwierciedlają tylko prawa plików, na które wskazują. W Slackware używa się głównie dowiązań symbolicznych. Oto typowy przykład:

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

Powłoka w `sh` w Slackware to tak naprawdę `bash`. Dowiązania usuwa się poleceniem `rm`, a tworzy `ln`. Te polecenia są omówione dokładniej w rozdziale 10.

Trzeba uważać na dowiązania symboliczne. Kiedyś pracowałem z komputerem, który miał notoryczne problemy z conocnym wykonywaniem kopii zapasowej na taśmie. Problemem okazały się dwa dowiązania symboliczne do katalogów, jednego pod drugim. Oprogramowanie do tworzenia kopii zapasowych zapisywało wciąż te same katalogi na taśmie, aż kończyło się na niej miejsce. Z reguły seria testów zapobiega tworzeniu dowiązań symbolicznych w takiej sytuacji, ale to był specyficzny komputer.

## 9.4 Montowanie urządzeń

Jak już omówiliśmy w sekcji 4.1.1, wszystkie napędy i urządzenia stanowią jeden wielki system plików. Partycje dysków twardych, CD-ROMy i dyskietki znajdują się w tym samym drzewie. Aby podłączyć je do systemu plików i móc z nich korzystać, używa się poleceń `mount(1)` i `umount(1)`.

Niektóre urządzenia są montowane automatycznie podczas uruchamiania komputera. Są one wymienione w pliku `/etc/fstab`. Wszystko, co ma być montowane automatycznie, ma wpis w tym pliku. Inne urządzenia wymagają wpisania odpowiedniego polecenia za każdym razem, kiedy chcesz z nich skorzystać.

### 9.4.1 Plik `fstab`

Oto przykład pliku `/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
```

W pierwszej kolumnie widnieje nazwa urządzenia. W tym przypadku, urządzenia to pięć partycji rozłożonych na dwóch dyskach twardych SCSI, dwa specjalne systemy plików niewymagające urządzenia, dyskietka i napęd CD-ROM. Druga kolumna zawiera punkty montowania urządzeń. Musi to być nazwa katalogu, z wyjątkiem partycji wymiany (swap). W trzeciej kolumnie jest typ systemu plików na urządzeniu. Dla zwykłych linuksowych systemów plików będzie to *ext2* (second extended filesystem), dla napędów CD-ROM – *iso9660*, a dla urządzeń z Windows – *msdos* lub *yfat*.

Czwarta kolumna to opcje stosowane do każdego montowanego systemu. Opcje domyślne (ang. defaults”) są w większości przypadków odpowiednie. Jednak urządzenia tylko do odczytu powinny mieć ustawioną flagę „ro”. Istnieje wiele takich opcji. Są opisane na stronie podręcznika *fstab*(5). Ostatnich dwóch kolumn używa program *fsck* i inne, które operują na urządzeniach. Przeczytaj również odpowiednią stronę podręcznika.

Podczas instalacji Linuksa Slackware program *setup* tworzy większość wpisów w pliku */etc/fstab*.

## 9.4.2 Polecenia *mount* i *umount*

Podłączanie nowego urządzenia do systemu plików jest łatwe. Wystarczy polecenie *mount* z kilkoma opcjami. Użycie *mount* może być tym łatwiejsze, jeśli urządzenie ma odpowiedni wpis w */etc/fstab*. Przypuśćmy, że chcę zamontować CD-ROM, a mój *fstab* wygląda, jak powyżej. Wystarczy, że wpiszę:

```
% mount /mnt/cdrom
```

W *fstab* istnieje wpis dla takiego punktu montowania, a więc *mount* wie, jakich opcji użyć. Gdyby nie istniał taki wpis, musiałbym sam użyć kilku opcji:

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

Takie polecenie zawiera wszystkie opcje z przykładowego *fstab*, ale i tak je omówię. Opcja *-t iso9660* to typ systemu plików na urządzeniu do zamontowania. W tym przypadku jest to *iso9660*, najczęściej używany w CD-ROMach. Opcja *-o ro* mówi *mount*, żeby zamontować urządzenie w trybie tylko do odczytu. */dev/cdrom* to nazwa urządzenia do zamontowania, a */mnt/cdrom* to miejsce, w którym ma ono być zamontowane.

Przed wyjęciem dyskietki, CD-ROMu lub innego obecnie zamontowanego urządzenia wymiennego trzeba je odmontować. Robi się to poleceniem *umount*. Nie pytaj, gdzie uciekło „n”, bo nie wiem. Jako argument do *umount* możesz przekazać albo urządzenie, albo punkt montowania. Jeśli np. chcesz odmontować CD-ROM z poprzedniego przykładu, oba polecenia zadziałają analogicznie:

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

## 9.5 Montowanie udziałów NFS

NFS to skrót od Network Filesystem, czyli sieciowy system plików. Nie jest to część faktycznego systemu plików, ale może stanowić jego rozszerzenie.

W dużych środowiskach uniksowych często współdzielili się te same programy, katalogi domowe i kolejki poczty. NFS rozwiązuje problem istnienia identycznej kopii na każdym komputerze. Można używać NFS do współdzielenia zbioru katalogów domowych między wieloma stacjami roboczymi. Takie stacje montują wówczas udziały NFS tak, jakby znajdowały się one na dyskach lokalnych.

Więcej informacji na ten temat znajduje się w sekcji NFS w sekcji 5.6.2 oraz na stronach podręcznika `exports(5)`, `nfsd(8)` i `mountd(8)`.

## Rozdział 10. Praca z plikami i katalogami

Linux ma w założeniu być tak uniksopodobny, jak to możliwe. Uniksowe systemy operacyjne zawsze używały interfejsu wiersza poleceń. W Slackware co prawda istnieje interfejs graficzny, ale za pomocą wiersza poleceń ciągle sprawuje się większość kontroli nad systemem. Dlatego należy znać podstawowe polecenia służące do zarządzania plikami.

W następujących sekcjach są opisane najczęściej stosowane polecenia do zarządzania plikami wraz z przykładami ich zastosowań. Istnieje wiele innych poleceń, ale tyle wystarczy na początek. Ponadto zamieszczone opisy są krótkie. Więcej informacji znajduje się na odpowiednich stronach podręcznika.

### 10.1 Nawigacja: `ls`, `cd` i `pwd`

#### 10.1.1 `ls`

To polecenie wyświetla listę plików w katalogu. Użytkownicy Windows i DOS zauważą podobieństwo do polecenia `dir`. Użyte bez opcji, `ls(1)` wyświetla pliki w bieżącym katalogu. Aby zobaczyć zawartość katalogu głównego (`root`), można użyć:

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

Częstym problemem jest fakt, że ciężko odróżnić katalog od pliku. Użytkownicy czasem chcą, aby `ls` dodawało identyfikator typu do wyniku:

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

Nazwy plików kończą się ukośnikiem, pliki wykonywalne gwiazdką itd.

`ls` może też wyświetlić inne informacje. Na przykład, aby zobaczyć daty utworzenia plików, ich właścicieli i prawa dostępu, użyj długiego listingu:

```
% 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/
```

Jeśli chcesz wyświetlić pliki ukryte, użyj:

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

Pliki, których nazwa zaczyna się od kropki (tzw. „dot files” czyli „kropliki”) to pliki „ukryte” podczas wykonywania `ls`. Można je wyświetlić, przekazując opcję `-a`.

O wiele więcej opcji jest opisanych na stronie podręcznika. Pamiętaj, że opcje można łączyć.

## 10.1.2 cd

Polecenia `cd` używa się do zmiany bieżącego katalogu. Wystarczy wpisać `cd` i odpowiednią ścieżkę. Oto 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 bez poprzedzającego ukośnika `cd` próbuje przejść do katalogu leżącego w bieżącym katalogu. To polecenie użyte bez żadnych opcji przechodzi do katalogu domowego użytkownika.

Polecenie `cd` nie jest podobne do innych poleceń. Jest to wbudowane polecenie powłoki. Takie polecenia są opisane w sekcji 8.3.1. Prawdopodobnie nie widzisz na razie różnicy. Zasadniczo oznacza to, że `cd` nie ma strony podręcznika. Musisz zamiast tego użyć pomocy powłoki:

```
% help cd
```

To wyświetla opcje `cd` i instrukcje, jak je stosować.

### 10.1.3 `pwd`

Polecenie `pwd` wyświetla katalog, w którym się aktualnie znajdujesz. Wystarczy wpisać samo `pwd`:

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

## 10.2 Programy stronicujące: `more`, `less` i `most`

### 10.2.1 `more`

Program `more(1)` (ang. więcej) to tzw. program stronicujący (ang. pager). Często wynik jakiegoś polecenia nie mieści się na jednym ekranie. Polecenia nie wiedzą, jak podzielić swój wynik na poszczególne ekrany. Zostawiają to zadanie programowi stronicującemu.

Polecenie `more` dzieli wyjście na pojedyncze ekrany i przechodzi do kolejnych po naciśnięciu spacji. Po wciśnięciu klawisza `enter` wynik przesuwa się o jeden wiersz. Oto przykład:

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

Wynik będzie się przez chwilę przewijał na ekranie. Aby podzielić go na poszczególne ekrany, wystarczy potokować go przez `more`:

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

Znak potokowania (`shift + odwrotny ukośnik`) oznacza „weź wyjście polecenia `ls` i przekaz je do `more`”. Przez `more` można potokować nie tylko wyjście `ls`, ale praktycznie wszystko. Potokowanie jest też opisane w sekcji 8.2.3.

### 10.2.2 `less`

Polecenie `more` jest wygodne, często jednak zdarza się przewinąć ekran za daleko. W `more` nie ma możliwości powrotu do poprzedniego ekranu. Taką funkcjonalność ma za to program `less(1)` (ang. mniej). Używa się go tak samo jak `more`, a więc powyższe przykłady też pasują. Tak więc `less` (mniej) to więcej niż `more` (więcej). Joost Kremers ujmuje to (mniej więcej) tak:

`less` oznacza więcej, ale więcej `more` niż `more`, dlatego `more` jest mniej `less`, więc używaj więcej `less`, jeśli chcesz mniej `more`.



## 10.2.3 most

Polecenie `most(1)` (ang. *najwięcej*) ma jeszcze więcej funkcji niż `less` i `more`. `less` to więcej niż `more`, a `most` to więcej niż `less`. Inne programy stronicujące potrafią wyświetlić tylko jeden plik naraz, za to `most` jest w stanie wyświetlić dowolną liczbę plików, pod warunkiem, że okno każdego pliku ma co najmniej dwa wiersze. `most` ma wiele opcji, które są opisane na stronie podręcznika.

## 10.3 Proste wyjście: `cat` i `echo`

### 10.3.1 `cat`

Polecenie `cat(1)` to skrót od „concatenate” – konkatelować, czyli łączyć. Zaprojektowano je początkowo do łączenia wielu plików tekstowych w jeden, ale ma wiele innych zastosowań.

Aby połączyć dwa lub więcej plików w jeden, po prostu wymienia się te pliki po wpisaniu `cat`, a wyjście przekierowuje do nowego pliku. `cat` operuje na standardowym wejściu i standardowym wyjściu, więc należy użyć znaków przekierowywania powłoki. Oto przykład:

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

To polecenie łączy treść plików `file1`, `file2` i `file3` razem, a następnie wysyła wynik do pliku `bigfile`.

Można użyć `cat` do wyświetlania treści plików. Często „`cat`”uje się pliki tekstowe przez `more` lub `less`, np.:

```
% cat file1 | more
```

To wyświetli plik `file1` i przepuści je przez `more`, dzięki czemu naraz będzie się wyświetlał jeden ekran.

Innym typowym zastosowaniem `cat` jest kopiowanie plików. Można skopiować dowolny plik, np.:

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

Powłoka `/bin/bash` jest kopiowana do katalogu domowego i nazwana „`mojapowloka`”.

Powyższe przykłady to jedne z wielu zastosowań `cat`. Jako że `cat` w dużej mierze korzysta ze standardowego wejścia i wyjścia, idealnie nadaje się do skryptów powłoki lub innych złożonych poleceń.

### 10.3.2 `echo`

Polecenie `echo(1)` wyświetla dany tekst na ekranie. Jako parametr podaje się po `echo` ciąg znaków, który ma być wyświetlony. Domyślnie `echo` wyświetli ciąg, a po nim znak nowego wiersza. Aby nie przechodzić do nowego wiersza, używa się opcji `-n`. Opcja `-e` sprawia, że `echo` poszukuje tzw. „znaków ucieczki” (ang. *escape characters*) w ciągu i wykonuje je.

## 10.4 Tworzenie: *touch* i *mkdir*

### 10.4.1 *touch*

Polecenia *touch*(1) używa się do zmiany znacznika czasu (ang. timestamp) pliku. Można zmienić znaczniki dostępu i modyfikacji. Jeśli plik nie istnieje, *touch* utworzy plik o podanej nazwie i zerowej długości. Aby zaznaczyć w pliku bieżący czas systemowy, używa się:

```
% 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* ma kilka opcji decydujących o tym, który znacznik zmodyfikować, jakiego czasu użyć itp. Więcej szczegółów znajduje się na stronie podręcznika.

### 10.4.2 *mkdir*

*mkdir*(1) tworzy nowy katalog. Przy uruchamianiu *mkdir* podaje się nazwę katalogu. W taki sposób tworzy się katalog *hejaz* w bieżącym katalogu:

```
% mkdir hejaz
```

Można też podać ścieżkę:

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

Opcja *-p* mówi *mkdir*, aby utworzyło też odpowiednie katalogi nadrzędne. Powyższy przykład się nie uda, jeśli nie istnieje katalog */usr/local*. Opcja *-p* utworzy zarówno */usr/local*, jak i */usr/local/hejaz*:

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

## 10.5 Kopiowanie i przenoszenie

### 10.5.1 *cp*

Polecenie *cp*(1) służy do kopiowania plików. Działa podobnie jak DOSowe *copy*. *cp* ma wiele opcji, warto więc przeczytać jego stronę podręcznika.

Najczęściej używa się *cp* do kopiowania pliku z jednej lokalizacji do drugiej, np.:

```
% cp hejaz /tmp
```

To skopiuje plik `hejaz` z bieżącego katalogu do katalogu `/tmp`.

Wielu użytkowników chce zachować znaczniki czasu, tak jak tu:

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

Dzięki temu znaczniki czasu w kopii nie ulegają modyfikacji.

Aby rekursywnie skopiować zawartość jednego katalogu do drugiego, używa się:

```
% cp -R jakiskatalog /tmp
```

To skopiuje katalog `jakiskatalog` do `/tmp`.

Aby skopiować dokładnie pliki i katalogi, zachowując ich prawa własności i dostępu oraz znaczniki czasu, używa się `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
```

Więcej opcji `cp` jest dokładnie opisanych na stronie podręcznika.

## 10.5.2 mv

Polecenie `mv(1)` przenosi pliki z jednego miejsca do drugiego. Brzmi prosto, czyż nie?

```
% mv staryplik /tmp/nowyplik
```

To polecenie ma kilka przydatnych opcji wiersza poleceń, które są dokładnie opisane na stronie podręcznika. W praktyce `mv` prawie nigdy nie używa się z opcjami wiersza poleceń.

## 10.6 Usuwanie: rm i rmdir

### 10.6.1 rm

`rm(1)` usuwa pliki i drzewa katalogów, podobnie jak DOSowe `del` i `deltree`. `rm` może być bardzo niebezpieczne, jeśli używa się go nierozważnie. Wprawdzie czasami jest możliwe przywrócenie usuniętego pliku, ale jest to skomplikowane (i potencjalnie kosztowne) i wykracza poza zakres tej książki.

Aby usunąć plik, wystarczy podać jego nazwę poleceniu `rm`:

```
% rm plik1
```

Jeśli nie masz do pliku prawa zapisu, wyświetli się błąd o braku dostępu. Aby wymusić usunięcie pliku, użyj opcji `-f`:

```
% rm -f plik1
```

Aby usunąć cały katalog, użyj `-r` i `-f` razem. To doskonały przykład, jak usunąć całą zawartość dysku twardego. Nie chcesz tego robić, niemniej jednak, oto polecenie:

```
# rm -rf /
```

Uważaj podczas wykonywania `rm`. Łatwo możesz zrobić sobie kuku. Inne opcje podawane w wierszu poleceń są opisane na stronie podręcznika.

## 10.6.2 `rmdir`

`rmdir(1)` usuwa katalogi. Katalog musi być pusty, zanim będzie go można usunąć. Składnia to po prostu:

```
% rmdir <katalog>
```

Oto przykład na usunięcie katalogu `hejaz` w bieżącym katalogu:

```
% rmdir hejaz
```

Jeśli katalog do usunięcia nie istnieje, `rmdir` wyświetli odpowiednią informację. Można podać pełną ścieżkę do katalogu:

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

Takie polecenie spróbuje usunąć podkatalog `hejaz` z katalogu `/tmp`.

Można usunąć katalog wraz z jego katalogami nadrzędnymi, używając opcji `-p`:

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

Najpierw `rmdir` spróbuje usunąć `hejaz` wewnątrz `/tmp`. Jeśli mu się uda, spróbuje usunąć sam `/tmp`. Polecenie kontynuuje działanie, dopóki nie napotka pierwszego błędu lub nie usunie całego drzewa.

## 10.7 Tworzenie nazw zastępczych plików przez `ln`

Polecenia `ln(1)` używa się do tworzenia dowiązań między plikami. Mogą to być dowiązania twarde lub miękkie (symboliczne). Różnica jest omówiona w sekcji 9.3. Aby utworzyć dowiązanie symboliczne do katalogu `/var/media/mp3` i umieścić je w katalogu domowym, użyj:

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

Opcja `-s` mówi `ln`, aby utworzyć dowiązanie symboliczne. Następnie mamy docelowy plik lub katalog, a na końcu nazwę dowiązania. W tym przypadku zostanie utworzony plik o nazwie `mp3` w katalogu domowym, który będzie wskazywał na `/var/media/mp3`. Dowiązanie może mieć dowolną nazwę, wystarczy ją podać jako ostatnią opcję.

Tworzenie dowiązania twardego jest równie łatwe. Wystarczy wywołać `ln` bez opcji `-s`. Jednak dowiązania twarde nie mogą zwykle wskazywać na katalogi lub pliki na innej partycji albo zdalnym systemie plików. Aby utworzyć dowiązanie twarde `/usr/bin/email` do `/usr/bin/mutt`, wpisz:

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

## Rozdział 11. Kontrola procesów

Każdy uruchomiony program jest nazywany procesem. Procesem może być zarówno system X Window, jak i programy systemowe (tzw. demony, ang. daemons), które są uruchamiane podczas startu komputera. Każdy proces działa z prawami jakiegoś użytkownika. Procesy uruchamiane podczas startu zwykle działają jako użytkownik `root` lub `nobody`. Procesy uruchomione przez Ciebie działają jako Ty. Procesy uruchomione przez innych użytkowników – jako oni.

Sprawujesz kontrolę nad procesami uruchomionymi przez siebie. Ponadto, `root` kontroluje wszystkie procesy, także te uruchomione przez innych użytkowników. Działanie procesów można kontrolować i monitorować za pomocą kilku programów oraz poleceń powłoki.

### 11.1 Praca w tle

Programy uruchamiane z wiersza poleceń działają na pierwszym planie. Dzięki temu można zobaczyć komunikaty programu i przekazywać mu polecenia. Czasem jednak chcesz, aby uruchomiony program nie zajmował terminala. Nazywa się to uruchamianiem programów w tle. Można to osiągnąć na kilka sposobów.

Po pierwsze, można dodać znak „ampersand” (&) na końcu wiersza poleceń podczas wywoływania programu. Na przykład, chcesz uruchomić konsolowy odtwarzacz plików `mp3 amp`, aby odtworzyć pliki z katalogu zawierającego ich kolekcję. Nie chcesz jednak zajmować terminala, bo masz coś na nim do zrobienia. Oto polecenie, które uruchomi `amp` w tle:

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

Program uruchomi się w tle i przywróci znak zgłoszenia powłoki.

Po drugie, można program przenieść w tło podczas jego działania. Najpierw go uruchom. Gdy już działa, wcisnij **Ctrl+z**. To zawiesza, czyli pauzuje proces. Natychmiast przestaje działać, ale może być uruchomiony znowu w każdej chwili. Po zawieszeniu procesu wracasz do znaku zgłoszenia. Teraz przenieś proces w tło, wpisując:

```
% bg
```

Proces, który był zawieszony, działa znowu w tle.

## 11.2 Praca na pierwszym planie

Jeśli chcesz się komunikować z procesem działającym w tle, musisz przywołać go na pierwszy plan. Jeśli proces w tle jest tylko jeden, wystarczy polecenie:

```
% fg
```

Jeśli jego działanie jeszcze się nie zakończyło, zajmie on terminal i nie powróci do znaku zgłoszenia. Czasami program kończy działanie podczas pracy w tle. W takim przypadku wyświetli się coś podobnego do tego:

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

To znaczy, że działający w tle proces (w tym przypadku `ls` – nieszczególnie ciekawy) zakończył się.

W tle może działać wiele procesów naraz. Wówczas musisz wiedzieć, który z nich chcesz przywołać. Wpisując `fg`, przywołasz proces, który był przeniesiony w tło jako ostatni. A jeśli masz całą listę takich procesów? Na szczęście w powłoce jest dostępne polecenie `jobs` wyświetlające wszystkie takie procesy:

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

Pokazuje ono listę wszystkich procesów w tle. Widać, że wszystkie są zatrzymane (stopped), czyli zawieszone. Liczba po lewej stronie jest czymś w rodzaju numeru identyfikacyjnego (ID) poszczególnych procesów. ID ze znakiem plusa obok (*man ps*) to proces, który zostanie przywołany na pierwszy plan po wpisaniu `fg`.

Aby przywołać `vim`, wpisz:

```
% fg 1
```

wtedy `vim` powróci na konsolę. Przenoszenie procesów w tło jest przydatne, jeśli masz tylko jeden terminal dostępny poprzez połączenie telefoniczne. Na takim terminalu możesz uruchomić kilka procesów, przełączając się między nimi co jakiś czas.

## 11.3 ps

Wiesz już, jak przełączać się między procesami uruchomionymi z wiersza poleceń. Wiesz też, że w tym samym czasie jest uruchomionych wiele procesów. Jak wyświetlić je wszystkie? Używając polecenia `ps` (1). Ma ono wiele opcji, omówimy tylko najważniejsze. Pełny opis znajduje się na stronie podręcznika. Strony podręcznika są omówione w sekcji 2.1.1.

Wpisanie samego `ps` wyświetli listę programów uruchomionych na danym terminalu. Będzie ona zawierać procesy pierwszoplanowe (w tym program samej powłoki) i działające w tle, a także samo polecenie `ps`. Często lista nie jest długa:

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

Nie ma tu zbyt wiele procesów, ale wyświetlone informacje są typowe. Wynik `ps` jest zawsze pogrupowany w odpowiednie kolumny, niezależnie od liczby uruchomionych procesów. Co one wszystkie oznaczają?

*PID* (*process ID*) to numer identyfikacyjny procesu. Każdy działający proces ma swój niepowtarzalny identyfikator, zawierający się w przedziale od 1 do 32767. Każdy nowo uruchomiony proces otrzymuje pierwszy wolny PID. Kiedy proces kończy działanie (lub jest unicestwiany, co zostało opisane w następnej sekcji), zwalnia swój PID. Gdy maksymalna liczba PID zostanie osiągnięta, jako następny PID zostanie wykorzystany pierwszy dostępny najmniejszy numer.

Kolumna *TTY* oznacza, na którym terminalu jest uruchomiony proces. Jako że `ps` bez opcji wyświetla tylko programy na bieżącym terminalu, w kolumnie *TTY* jest ta sama wartość. Jak widać, oba procesy są uruchomione na terminalu `ttyp0`. To znaczy, że albo działają zdalnie, albo z jakiegoś terminala X.

Kolumna *TIME* określa, ile czasu procesora zajmuje uruchomiony proces. Różni się to od rzeczywistego czasu działania procesu. Pamiętaj, że Linux to system wielozadaniowy. Przez cały czas jest w nim uruchomionych wiele procesów. Każdemu z nich jest przydzielana część czasu procesora. Tak więc kolumna *TIME* powinna pokazywać o wiele mniej czasu dla każdego procesu, niż w rzeczywistości on działa. Jeśli wartość *TIME* jest większa niż kilka minut, prawdopodobnie coś tu nie gra.

Wreszcie kolumna *CMD* pokazuje właściwy program, jaki jest uruchomiony. Wyświetlana jest tylko nazwa programu, bez opcji wiersza poleceń itp. Aby je wyświetlić, użyj odpowiedniej opcji `ps`. Omówimy to w skrócie.

Używając odpowiednio połączonych opcji, możesz wyświetlić wszystkie uruchomione procesy. Lista będzie prawdopodobnie długa (gdy piszę to zdanie na laptopie, ma ona 55 wierszy), więc obetnę 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]
[obcięte wyjście]

```

Większość z tych procesów jest z reguły uruchamiana podczas startu. Ponieważ dokonałem u siebie kilku modyfikacji, u Ciebie wynik może wyglądać inaczej. Większość procesów zobaczysz jednak również na swoim komputerze. Jak widać, są wyświetlone opcje wiersza poleceń przekazane do procesów. Ostatni błąd jądra, wykryty w funkcji `ptrace`, poskutkowało wprowadzeniem poprawki, która uniemożliwia wyświetlenie opcji dla wielu procesów. Obecnie są wyświetlane w nawiasach kwadratowych pod PID 108 do 110. Poprawka wprowadziła również kilka innych kolumn i interesujących informacji.

Zauważ, że większość procesów jest uruchomiona na tty „?”. To oznacza, że nie są one podłączone do żadnego terminala. Najczęściej są to demony, czyli procesy działające bez podłączania do konkretnych terminali. Najczęściej spotykane demony to sendmail, BIND, apache i NFS. Działają one w ten sposób, że „nasłuchują” w oczekiwaniu na jakieś zgłoszenie klienta, a po otrzymaniu zgłoszenia przekazują mu jakieś informacje.

Widać tu nową kolumnę: *STAT*. Pokazuje ona status procesu. *S* oznacza „sleeping” - w stanie uśpienia. Taki proces oczekuje na jakieś zdarzenie. *Z* oznacza proces zombie. Procesy zombie to takie procesy, których procesy macierzyste zostały unicestwione, pozostawiając swoje procesy potomne. Nie wróży to nic dobrego. *D* oznacza proces, który znajduje się w stanie nieprzerwywalnego uśpienia. Takich procesów często nie daje się unicestwić, nawet przekazując im sygnał SIGKILL. Więcej na temat sygnału SIGKILL znajduje się sekcji dotyczącej polecenia `kill`. *W* oznacza stronicowanie. Proces martwy jest oznaczony literą *X*. Proces *T* jest śledzony lub zatrzymany. *R* oznacza, że proces można uruchomić.

Jeszcze więcej informacji uzyskasz, używając:



```

% 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]
[obcięte wyjście]

```

To dopiero kupa informacji. Widać dodatkowo, który użytkownik uruchomił proces, ile zasobów systemowych proces używa (kolumny %CPU, %MEM, VSZ i RSS), a także czas uruchomienia. Taka ilość informacji przydaje się administratorowi systemu. Jeszcze jedno: informacje wychodzą poza prawą krawędź ekranu, więc nie można jej przeczytać. Użyj opcji `-w` polecenia `ps`, aby wymusić zawijanie wierszy.

Nie wygląda to ładnie, ale spełnia swoje zadanie. Wyświetlają się pełne informacje o każdym procesie. Istnieją też inne, opisane szczegółowo na stronie podręcznika `ps`. Powyższe opcje są jednak najpowszechniejsze i ich będziesz używał najczęściej.

## 11.4 kill

Czasami programy nie działają poprawnie i trzeba przywrócić je do porządku. Używa się do tego polecenia `kill(1)`. Za jego pomocą można manipulować procesami na kilka sposobów. Najbardziej oczywistym jest zabicie (ang. `kill` – zabić), czyli wyłączenie procesu (zwane też unicestwieniem). Jest to potrzebne, jeśli program wymknął się spod kontroli i zużywa zbyt wiele zasobów systemowych, albo jeśli po prostu Ci się znudził.

Aby zabić proces, musisz znać jego PID lub nazwę. Do zdobycia PID użyj `ps`, tak jak zostało to opisane w poprzedniej sekcji. Aby zabić proces o numerze np. 4747, wpisz:

```
% kill 4747
```

Musisz być właścicielem procesu, aby go zabić. To funkcja bezpieczeństwa. Gdybyś było możliwe zabijanie procesów innych użytkowników, mogłoby to prowadzić do różnego rodzaju niewłaściwych postępowań. Oczywiście `root` może zabić każdy proces.

Istnieje odmiana polecenia `kill` o nazwie `killall(1)` (ang. zabić wszystkie). Działa dokładnie tak, jak się nazywa: zabija wszystkie aktualnie działające procesy o podanej nazwie. Na przykład, aby zabić wszystkie procesy `vim`, wpisz:

```
% killall vim
```

Wszystkie procesy `vim` zostaną zakończone. Wykonanie tego polecenia jako `root` spowoduje wyłączenie wszystkich procesów `vim` u wszystkich użytkowników. Oto ciekawy sposób, w jaki można usunąć wszystkich (włączając w to siebie samego) z systemu:

```
# killall bash
```

Czasami zwykle `kill` nie działa, ponieważ niektóre procesy nie reagują na nie. Trzeba użyć mocniejszej formy. Jeśli ten podły PID 4747 nie odpowiedział na żądanie wyłączenia, wpisz:

```
% kill -9 4747
```

Proces 4747 zostanie prawie na pewno zabity. Tej samej opcji można użyć z `killall`. Działa to w ten sposób, że do procesu jest wysłany szczególny sygnał. Standardowe `kill` wysyła sygnał zakończenia *SIGTERM* (ang. terminate – zakończyć), które mówi procesowi, aby zakończył pracę, posprzątał po sobie i wyłączył się. `kill -9` wysyła *SIGKILL*, czyli sygnał zabicia, który go po prostu wyłącza. Proces nie ma szansy zakończyć się w czysty sposób, co czasem prowadzi do uszkodzenia danych. Oto pełna lista sygnałów. Można ją wyświetlić poleceniem:

```
% 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
```

Z `kill` należy użyć numeru, z `killall` – części nazwy bez poprzedzającego „SIG”. Oto inny przykład:

```
% killall -KILL vim
```

Polecenia `kill` można też użyć do zrestartowania procesu. Większość procesów reaguje na *SIGHUP*, odczytując ponownie swoje pliki konfiguracyjne. Jest to szczególnie przydatne, jeśli zmodyfikowałeś taki plik i chcesz, aby program odczytał zmienioną wersję.

## 11.5 top

Na koniec opiszemy polecenie, którego używa się, aby wyświetlić stale aktualizowane informacje o uruchomionych procesach. To polecenie nazywa się `top(1)` i uruchamia się je tak:

```
% top
```

Wyświetli to ekran pełen informacji o uruchomionych procesach, a także ogólne dane o systemie. Są to: przeciętne obciążenie, liczba procesów, status procesora, ilość wolnej pamięci oraz szczegóły dotyczące procesów, takie jak PID, właściciel, priorytet, zużycie procesora i pamięci, czas od włączenia 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
```

Polecenie nazywa się `top` (ang. szczyt, wierzchołek, góra), ponieważ programy zużywające najwięcej czasu CPU są wyświetlone na górze. Co jest ciekawe, to fakt, że samo `top` będzie wyświetlone jako pierwsze na większości mało obciążonych (czasem też na obciążonych) komputerach, ponieważ samo zużywa dużo czasu procesora. Niemniej jednak, `top` jest bardzo użyteczne do określenia, który program wymknął się spod kontroli i trzeba go zabić.

Jak wyświetlić procesy należące tylko do Ciebie albo tylko do innego użytkownika? Przecież mogą nie mieścić się na liście najbardziej aktywnych programów. Dzięki opcji `-u` możesz określić nazwę lub numer użytkownika (UID) i monitorować tylko te procesy, których właścicielem jest dany UID.

```
% top -u alan
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 widać, w tej chwili mam uruchomiony system X, `top`, `gnome-terminal` (w którym właśnie piszę) i wiele innych procesów związanych z X, które zużywają najwięcej czasu CPU. W ten sposób można mierzyć, jak intensywnie pracują użytkownicy.

Można też monitorować procesy według PID, ignorować procesy beczynne i zombie. Te i inne opcje są opisane na stronie podręcznika `top`.

## Rozdział 12. Podstawowa administracja systemem

Zaraz zaraz... Wiem, co myślisz: „Nie jestem administratorem systemu! Nawet nie chcę nim być!”

W rzeczywistości jesteś administratorem każdego komputera, do którego znasz hasło `roota`. Może to być domowy komputer z jednym czy dwoma użytkownikami albo wielki serwer z setkami użytkowników. Tak czy owak, musisz wiedzieć, jak zarządzać kontami użytkowników i jak bezpiecznie zamykać system. Wydaje się, że to łatwe, ale trzeba pamiętać o paru rzeczach,

### 12.1 Użytkownicy i grupy

Jak napisaliśmy w rozdziale 8, nie powinno się korzystać z konta `root` do codziennej pracy. Powinieneś utworzyć konto zwykłego użytkownika i korzystać z niego na co dzień, a konta `root` używać tylko do czynności administracyjnych. Do stworzenia użytkownika możesz użyć narzędzi dołączonych do Slackware lub ręcznie zmodyfikować plik z hasłem.

#### 12.1.1 Skrypty dołączone do dystrybucji

Najprościej zarządza się użytkownikami i grupami za pomocą skryptów i programów dostępnych w dystrybucji. W Slackware są to: `adduser`, `userdel(8)`, `chfn(1)`, `chsh(1)` i `passwd(1)` dla użytkowników i `groupadd(8)`, `groupdel(8)`, `groupmod(8)` dla grup. Z wyjątkiem `chfn`, `chsh` i `passwd` używa ich tylko `root`. Dlatego są umieszczone w `/usr/sbin`. Natomiast `chfn`, `chsh` i

passwd może uruchomić każdy, dlatego są w /usr/bin.

Użytkowników dodaje się poleceniem `adduser`. Opiszemy poszczególne kroki procedury, pokazując wszystkie zadawane pytania i krótko je objaśniając. Domyślne odpowiedzi są w nawiasach kwadratowych i z reguły są odpowiednie, chyba że naprawdę chcesz coś zmienić.

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

Login name (nazwa logowania) to nazwa, z jaką użytkownik będzie się logował w systemie. Zazwyczaj nazwy mają 8 znaków lub mniej i są pisane małymi literami. (Można użyć więcej niż 8 znaków, a także cyfr, ale tylko jeśli ma się naprawdę ważny powód).

Nazwa logowania może być podana jako argument w wierszu poleceń:

```
# adduser wilcio
```

W obu przypadkach następnym będzie pytanie o numer identyfikacyjny użytkownika – UID (user ID).

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

Linux tak naprawdę określa prawa własności według numerów użytkowników, a nie ich nazw. Każdemu użytkownikowi jest przypisany niepowtarzalny numer, w Slackware zaczynają się one od 1 000. Można przyznać konkretny UID lub pozwolić `adduser` na przypisanie pierwszego wolnego numeru.

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

Wszyscy użytkownicy są domyślnie przypisani do grupy `users`. Jeśli nie wiesz, co robisz, nie zmieniaj tego przydziału na inną grupę.

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

Tutaj możesz dodać użytkownika do innych grup. Można należeć do kilku grup naraz. Przydaje się to, jeśli utworzysz grupy użytkowników mogących modyfikować strony www, grać w gry itp. Często np. tworzy się grupę `wheel`, której członkowie jako jedyni mogą używać polecenia `su`. W domyślnej instalacji Slackware istnieje grupa `sys` dla użytkowników, którzy mogą odtwarzać dźwięki za pomocą karty dźwiękowej.

```
Home directory [/home/wilcio]
```

Katalogi domowe użytkowników domyślnie znajdują się w katalogu `/home`. Jeśli system jest rozbudowany, być może katalogi domowe są w innym miejscu (albo miejscach). W tym kroku możesz wybrać lokalizację katalogu domowego.

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

`bash` jest domyślną powłoką w Linuksie Slackware. Wystarcza ona większości ludzi. Jeśli nowy użytkownik ma doświadczenie z systemami uniksowymi, być może zna inną powłokę. Można ją zmienić teraz lub użytkownik zmieni ją sobie później, używając `chsh`.

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

Można ustawić konta tak, aby traciły ważność po upływie określonego czasu. Domyślnie nie ma daty wygaśnięcia ważności konta, ale możesz to zmienić. Jest to przydatne, jeśli ktoś jest Operatorem Internetu (ISP – Internet Service Provider) i chce, aby czyjeś konto wygasało po upływie jakiejś daty, chyba że użytkownik zapłaci za następny rok.

```
New account will be created as follows:
```

```
-----  
Login name:          wilcio  
UID:                [ Next available ]  
Initial group:      users  
Additional groups:  [ None ]  
Home directory:    /home/wilcio  
Shell:              /bin/bash  
Expiry date:       [ Never ]
```

```
This is it... if you want to bail out, hit Control-C.  Otherwise, press  
ENTER to go ahead and make the account.
```

To podsumowanie dotychczas wybranych informacji i możliwość anulowania tworzenia konta. Jeśli wpisałeś coś źle, wciśnij **Ctrl+C**, aby anulować tworzenie. Po wciśnięciu **ENTER** natomiast konto zostanie utworzone.

```
Creating new account...
```

```
Changing the user information for wilcio  
Enter the new value, or press return for the default  
  Full Name []: Wlodzimierz  
  Room Number []: Bramkowski 130  
  Work Phone []:  
  Home Phone []:  
  Other []:
```

Pozostałe informacje są opcjonalne. Nie musisz nic wpisywać. Można je zmienić później, używając **chfn**. Jeśli jednak planujesz późniejsze kontakty z daną osobą, warto wpisać przynajmniej imię, nazwisko i numer telefonu.

```
Changing password for wilcio  
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.
```

Następnie musisz wpisać hasło dla nowego użytkownika. Jeśli nie jest on obecny przy tworzeniu konta, wymyśl jakieś proste hasło i przekazaj użytkownikowi, aby zmienił je później na bezpieczniejsze.

*Uwaga na temat wyboru hasła:* Bezpieczne hasło to pierwsza linia obrony przed włamaniem. Hasło nie powinno być łatwe do odgadnięcia, ponieważ w ten sposób włamywacze komputerowi mają ułatwione zadanie. Idealne hasło to ciąg przypadkowych znaków, takich jak wielkie i małe litery, cyfry i znaki specjalne (tabulator nie jest akurat dobrym znakiem, zależy to od komputerów, z których się logujesz). W Internecie jest dostępnych wiele programów generujących hasła z przypadkowych znaków.

Ogólnie rzecz biorąc, użyj zdrowego rozsądku. Nie używaj daty urodzin, powszechnie znanych słów, nazwy czegoś, co znajduje się na biurku ani czegokolwiek, co się z Tobą kojarzy. „bezpiecznehaslo1” (lub inne hasło wydrukowane albo widoczne online) też nie jest dobre.

Usuwanie użytkowników jest proste. Wystarczy uruchomić `userdel` z nazwą użytkownika jako argumentem. Upewnij się, że użytkownik jest wylogowany i nie są uruchomione żadne jego procesy. Pamiętaj, że po usunięciu użytkownika znikają bezpowrotnie informacje o hasle.

```
# userdel wilcio
```

To usunie `wilcio` z systemu. Dobry sposób na pozbycie się denerwującego użytkownika :). Usuwane są wpisy z `/etc/passwd`, `/etc/shadow` i `/etc/group`, ale katalog domowy pozostaje.

Aby usunąć również katalog domowy, wpisz:

```
# userdel -r wilcio
```

Tymczasowa blokada konta jest opisana w kolejnej sekcji, jako że tymczasowe zablokowanie wymaga zmiany hasła użytkownika. Zmiana innych informacji o użytkowniku jest opisana w sekcji 12.1.3.

Programy do usuwania i dodawania grup są proste. `groupadd` dodaje nowy wpis do `/etc/group` z niepowtarzalnym numerem identyfikacyjnym grupy (group ID – GID). `groupdel` usuwa podaną grupę. Możesz też modyfikować `/etc/group`, dodając użytkowników do innych grup. Aby np. dodać grupę `cvs`, wpisz:

```
# groupadd cvs
```

Aby ją usunąć:

```
# groupdel cvs
```

## 12.1.2 Zmiana hasła

Program `passwd` zmienia hasło, modyfikując plik `/etc/shadow`. W tym pliku są przechowywane zaszyfrowane hasła wszystkich użytkowników. Aby zmienić własne hasło, wpisz:

```
$ passwd
Changing password for chris
Old password:
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:
```

Jak widać, pokazuje się monit o wpisanie starego hasła. Nie pojawi się ono na ekranie podczas wpisywania (podobnie jak podczas logowania). Następnie `passwd` prosi o podanie nowego hasła, sprawdza, czy jest odpowiednie i wyświetla ostrzeżenie, jeśli nie jest. Możesz je zignorować. Potem pojawia się prośba o wpisanie nowego hasła po raz drugi dla potwierdzenia.

Root może zmieniać hasła innych użytkowników:

```
# passwd ted
```

Procedura jest podobna, ale nie musisz wpisywać starego hasła. (Jedna z wielu zalet bycia `rootem`...)

Możesz tymczasowo zablokować konto i odblokować je po pewnym czasie. Robi się to za pomocą `passwd`. Aby zablokować konto, wykonaj jako `root`:

```
# passwd -l david
```

Zmieni to hasło użytkownika `david` na coś, co nigdy nie pasuje do zaszyfrowanej wartości. Konto odblokowuje się przez:

```
# passwd -u david
```

Teraz konto `david` jest aktywne. Blokowanie konta przydaje się, jeśli użytkownik naruszył ustalone zasady korzystania z systemu albo wyeksportował bardzo dużą kopię programu `xeyes(1)` na twój pulpit X.

### 12.1.3 Zmiana informacji o użytkowniku

Użytkownik może w każdej chwili zmienić dwie rzeczy: powłokę i informacje używane przez program `finger`. Linux Slackware używa do tego `chsh` (change shell – zmień powłokę) i `chfn` (change finger – zmień informacje używane przez `finger`).

Można wybrać dowolną powłokę wymienioną w `/etc/shells`. `/bin/bash` wystarcza w większości przypadków. Niektórzy wolą powłokę, której używają w pracy czy szkole. Aby zmienić powłokę, użyj `chsh`:

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

Po podaniu hasła wpisz pełną ścieżkę do powłoki. Przedtem sprawdź, czy jest wymieniona w `/etc/shells(5)`. `root` może zmienić powłokę każdego użytkownika, uruchamiając `chsh` z jego nazwą jako argumentem.

Informacje dla programu `finger` są opcjonalne. Są to: imię i nazwisko, telefon, numer pokoju. Można je zmienić za pomocą `chfn`, przechodząc przez te same kroki, co podczas tworzenia użytkownika. Oczywiście `root` może je zmienić dla każdego.

## 12.2 Użytkownicy i grupy dla hardcore'owców

Można dodawać, usuwać i zmieniać użytkowników oraz grupy bez skryptów i programów. Nie jest to trudne, ale po przeczytaniu tej sekcji raczej stwierdzisz, że skrypty są prostsze. Musisz jednak wiedzieć, w jaki sposób są przechowywane informacje o koncie, w razie gdybyś musiał je kiedyś odtworzyć, a nie miał dostępu do narzędzi Slackware.

Najpierw dodamy użytkownika do `/etc/passwd(5)`, `/etc/shadow(5)` i `/etc/groups(5)`. W pliku `passwd` są przechowywane niektóre informacje o użytkownikach, ale (o dziwo) nie ich hasła. Takie rozwiązanie zostało dawno zarzucone ze względów bezpieczeństwa. Prawo do odczytu `passwd`



muszą mieć wszyscy użytkownicy, ale zaszyfrowane hasła nie powinny być dostępne do odczytu dla każdego. Potencjalni włamywacze mogliby spróbować odszyfrować zaszyfrowane hasła. Zamiast tego, hasła w zaszyfrowanej postaci są przechowywane w pliku `shadow`, do którego prawo odczytu ma tylko `root`. W polu hasła w `passwd` jest tylko „x”. W pliku `group` są wymienione wszystkie grupy i ich członkowie.

Aby bezpiecznie modyfikować `/etc/passwd`, użyj polecenia `vipw`, do `/etc/group` – `vigr`, do `/etc/shadow` – `vispw -s`. („Bezpiecznie” oznacza w taki sposób, że podczas edycji nikt inny nie ma dostępu do pliku. Jeśli jesteś jedynym administratorem, jesteś raczej bezpieczny, ale warto wyrabiać dobre nawyki od samego początku.)

Zobaczmy, jak jest skonstruowany plik `/etc/passwd` i dowiedzmy się, jak dodać w nim nowego użytkownika. Oto typowy wpis w `/etc/passwd`:

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

Każdy wiersz jest wpisem dotyczącym jednego użytkownika. Pola w każdym wierszu są oddzielone dwukropkami. Pola to: nazwa użytkownika, zaszyfrowane hasło (lub „x” w Slackware, ponieważ Slackware używa haseł typu „shadow”), UID, GID, opcjonalne informacje dla `finger` (oddzielone przecinkami), katalog domowy i powłoka. Aby dodać użytkownika ręcznie, trzeba na końcu pliku dodać nowy wiersz z odpowiednimi informacjami.

Takie informacje muszą spełniać pewne wymagania, w przeciwnym przypadku użytkownik będzie miał problemy z zalogowaniem się. Po pierwsze, upewnij się, że w polu hasła jest „x”, a nazwa użytkownika i jego UID są niepowtarzalne. Przypisz użytkownikowi grupę 100 (grupa „users” w Slackware) lub inną domyślną grupę (użyj GID, a nie nazwy grupy). Przydziel poprawny katalog domowy (utworzysz go później) i powłokę (właściwe powłoki są wymienione w `/etc/shells`).

Następnie dodaj wpis w `/etc/shadow`, który zawiera zaszyfrowane hasło. Oto typowy wpis:

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

Tak samo jak w `passwd`, każdy wiersz odpowiada osobnemu użytkownikowi, poszczególne pola są oddzielone dwukropkami. Są to po kolei: nazwa użytkownika; zaszyfrowane hasło; liczba dni, które upłynęły od początku tzw. Epoki (1 stycznia 1970) do czasu ostatniej zmiany hasła; liczba dni, które muszą upłynąć, aby hasło mogło zostać zmienione znowu; dni, po których hasło musi być zmienione; dni przed czasem wygaśnięcia konta, kiedy użytkownik jest powiadamiany o nadchodzącym wygaśnięciu; dni od początku Epoki, kiedy konto jest nieaktywne, pole zarezerwowane.

Jak widzisz, większość pól dotyczy wygaśnięcia konta. Jeśli nie chcesz, aby konto wygasło, wystarczy, że wypełnisz niektóre pola odpowiednimi wartościami. W przeciwnym razie będziesz musiał trochę pomyśleć i wyliczyć właściwe liczby. W polu hasła nowego użytkownika wpisz jakieś przypadkowe znaki. Nie przejmuj się hasłem, bo i tak je za chwilę zmienisz. Jedynym znakiem, jakiego nie możesz użyć w polu hasła, jest dwukropek. Pole „dni od ostatniej zmiany hasła” pozostaw puste. Wpisz dalej odpowiednio 0, 99999 i 7, tak jak w przykładzie, a inne pola też zostaw puste.

(Jeśli myślisz, że podałem swoje prawdziwe, zaszyfrowane hasło i chcesz się włamać do mojego systemu, wal śmiało. Jeśli potrafisz złamać hasło, będziesz znał hasło do testowego systemu zabezpieczonego zaporą sieciową. Na pewno Ci się to przyda :))

W typowym systemie Slackware wszyscy zwykli użytkownicy są członkami grupy `users`. Jeśli jednak chcesz utworzyć nową grupę lub dodać użytkownika do innych grup, zmodyfikuj plik `/etc/groups`. Oto typowy wpis:

```
cvs::102:chris,logan,david,root
```

Kolejne pola to: nazwa grupy, hasło grupy, GID, i członkowie grupy oddzieleni przecinkami. Aby utworzyć nową grupę, wystarczy dodać na końcu pliku nowy wiersz z niepowtarzalnym GID i wszystkimi użytkownikami, którzy mają do niej należeć. Jeśli taki użytkownik istnieje i jest zalogowany, musi wylogować się i zalogować ponownie, aby wprowadzić zmiany w życie.

Warto użyć poleceń `pwck` i `grpck`, aby sprawdzić, czy dokonane zmiany są spójne. Najpierw wywołaj `pwck -r` i `grpck -r`. Opcja `-r` nie dokonuje żadnych zmian, za to wyświetla listę ewentualnych zmian, jakich polecenie by dokonało, gdyby było uruchomione bez niej. Użyj wyniku, aby zdecydować, czy potrzebna jest dalsza modyfikacja plików, uruchomienie `pwck` i `grpck` bez `-r`, czy też można zostawić zmienione pliki.

Teraz uruchom `passwd`, aby przydzielić nowemu użytkownikowi hasło. Następnie za pomocą `mkdir` utwórz katalog domowy w takiej lokalizacji, jak podana w `/etc/passwd`. Na koniec przez `chown` zmień właściciela tego katalogu na nowego użytkownika.

Aby usunąć użytkownika, wystarczy usunąć jego wpisy: wiersze z `/etc/passwd` i `/etc/shadow` oraz nazwę użytkownika ze wszystkich grup w `/etc/group`. Możesz też usunąć jego katalog domowy, kolejkę poczty i wpis `crontab` (jeśli istnieją).

Grupy usuwa się podobnie: usuń odpowiedni wpis z `/etc/group`.

## 12.3 Poprawne zamykanie systemu

Poprawne zamykanie systemu jest niezwykle ważne. Wyłączenie komputera przez wciśnięcie przycisku zasilania może spowodować poważne uszkodzenie systemu plików. Kiedy komputer jest włączony, stale używane są pewne pliki, nawet jeśli sam nic nie robisz. Pamiętaj, że cały czas jest uruchomionych wiele procesów w tle. Te procesy zarządzają systemem i utrzymują wiele otwartych plików. Jeśli wyłączy się zasilanie, pliki nie zostaną poprawnie zamknięte, co może je uszkodzić. W zależności od tego, które z nich uległy uszkodzeniu, system może przestać nadawać się do użytku! Nawet jeśli nie, procedura sprawdzania systemu plików po ponownym uruchomieniu komputera jest długotrwała.

Uwaga: Jeśli Twój komputer wykorzystuje system plików z kronikowaniem (zwanym też księgowaniem), taki jak `ext3` lub `reiserfs`, chroni to częściowo przed uszkodzeniem plików. Czas sprawdzania po ponownym uruchomieniu komputera jest krótszy niż w przypadku systemu bez kronikowania, np. `ext2`. Takie zabezpieczenie nie jest jednak wymówką dla nieprawidłowego zamykania systemu! Systemy plików z kronikowaniem mają chronić przed uszkodzeniami wynikłymi nie z Twojej winy, ale nie chronią przed uszkodzeniami, których sprawcą jest Twoje lenistwo.

Jeśli chcesz wyłączyć komputer lub uruchomić go ponownie, musisz właściwie zamykać system. Można to zrobić na kilka sposobów, wybierz ten, który uznasz za najciekawszy (albo niewymagający wiele pracy). Procedury zamykania i restartowania są podobne, tak więc większość sposobów ma zastosowanie i przy tym, i przy tym.

Pierwszą i prawdopodobnie najpopularniejszą metodą jest użycie polecenia `shutdown(8)`. Można za jego pomocą wyłączać lub restartować system w zadanym czasie, wyświetlając jednocześnie komunikat o zbliżającym się zamknięciu dla wszystkich zalogowanych użytkowników.

Najczęściej używa się `shutdown` do wyłączenia komputera w ten sposób:

```
# shutdown -h now
```

W tym przypadku nie wysyłamy żadnego specjalnego komunikatu, użytkownicy zobaczą domyślny komunikat programu. „*Now*” (ang. teraz) to czas, kiedy komputer ma być wyłączony, a „*-h*” oznacza zatrzymanie systemu. Nie jest to dobre rozwiązanie w przypadku systemów z zalogowanymi wieloma użytkownikami, ale jest odpowiednie dla komputera domowego. W systemach z wieloma użytkownikami lepszym rozwiązaniem jest przekazanie komunikatu z niewielkim wyprzedzeniem:

```
# shutdown -h +60
```

Spowoduje to zamknięcie systemu za godzinę (60 minut), co powinno w zupełności wystarczyć, jeśli z systemu korzysta wiele osób. Czas zamknięcia systemów o istotnym znaczeniu powinien być zaplanowany z większym wyprzedzeniem. Komunikat o planowanym zamknięciu powinien być podany za pośrednictwem odpowiedniego kanału (e-mail, tablica ogłoszeń, plik `/etc/motd` itp.).

Ponowne uruchomienie komputera wywołuje się tym samym poleceniem, ale z opcją „*-r*” zamiast „*-h*”:

```
# shutdown -r now
```

Czas podaje się tak samo, jak przy wyłączaniu komputera. Polecenie `shutdown` ma wiele innych opcji kontrolujących sposób i czas wyłączania/restartowania komputera. Są one opisane na stronie podręcznika.

Innym sposobem wyłączania są polecenia `halt(8)` i `reboot(8)`. Jak sama nazwa wskazuje, `halt` (ang. zatrzymać) natychmiast zatrzyma pracę systemu operacyjnego, a `reboot` (ang. przeładować) uruchomi komputer ponownie (`reboot` to tak naprawdę dowiązanie symboliczne do `halt`). Wywołuje się je tak:

```
# halt  
# reboot
```

Niskopoziomowym sposobem wyłączania lub restartowania jest bezpośrednia komunikacja z programem `init`. Inne metody to tylko wygodniejsze odmiany takiej komunikacji. Można jednak przekazać informacje do `init` bezpośrednio, używając polecenia `telinit(8)` (uwaga na pisownię – tylko jedno „*l*”). To polecenie mówi `init`, na jaki poziom pracy ma przejść, co spowoduje wykonanie odpowiedniego skryptu. Skrypt zabija lub uruchamia procesy odpowiednie dla danego poziomu. Ten sposób działa z wyłączaniem i restartowaniem, ponieważ są to specjalne poziomy pracy.

```
# telinit 0
```

Poziom pracy 0 to tryb zatrzymania. Polecenie przejścia w ten tryb przekazane do `init` spowoduje, że wszystkie procesy zostaną zabite, systemy plików odmontowane, a komputer zatrzymany. Jest to całkowicie poprawna metoda zamykania systemu, w przypadku laptopów i nowszych komputerów domowych najczęściej również wyłącza maszynę całkowicie.

```
# telinit 6
```

Poziom pracy 6 to tryb restartowania. Polecenie przejścia w ten tryb przekazane do `init` spowoduje, że wszystkie procesy zostaną zabite, systemy plików odmontowane, a komputer uruchomiony ponownie. Jest to całkowicie poprawna metoda restartowania systemu.

Dla ciekawskich: podczas przełączania systemu na poziom pracy 0 lub 6, niezależnie czy przez `shutdown`, `halt`, czy `reboot`, jest uruchamiany skrypt `/etc/rc.d/rc.6` (`/etc/rc.d/rc.0` to dowiązanie symboliczne do `/etc/rc.d/rc.6`). Możesz dostosować ten plik do swoich potrzeb, ale upewnij się, że dobrze przetestowałeś wprowadzone zmiany!

Istnieje jeszcze jedna metoda restartowania systemu. Wszystkie pozostałe wymagają, aby wykonywał je `root`. Możesz jednak uruchomić komputer ponownie nawet, jeśli nie jesteś `rootem`. Wystarczy, że będziesz miał fizyczny dostęp do klawiatury. Użycie kombinacji **Control+Alt+Delete** (tzw. „salut Trzech Króli”) spowoduje natychmiastowy restart (w istocie po wciśnięciu **Ctrl+Alt+Del** jest wywoływane polecenie `shutdown`). Salut nie zawsze działa w X – musisz użyć **Ctrl+Alt+F1** (albo innego klawisza funkcyjnego), aby przełączyć się na terminal tekstowy przed zastosowaniem tej kombinacji.

Głównym plikiem konfiguracyjnym, który kontroluje wszelkie aspekty uruchamiania i zamykania systemu, jest `/etc/inittab(5)`. Najczęściej nie trzeba go modyfikować, ale jego przeczytanie może wyjaśnić, dlaczego pewne rzeczy działają tak, a nie inaczej. Oczywiście zawsze przeczytaj odpowiednie strony podręcznika.

## Rozdział 13. Podstawowe polecenia sieciowe

Sieć składa się z kilku komputerów połączonych razem. Sieć może być niewielka, np. parę komputerów w domu albo biurze, albo bardzo rozbudowana, tak jak duża sieć uniwersytecka lub nawet cały Internet. Kiedy komputer jest częścią sieci, masz dostęp do innych maszyn bezpośrednio lub za pomocą usług, takich jak poczta elektroniczna lub strony WWW.

Istnieje wiele programów służących do pracy w sieci. Jedne są przydatne do testowania, czy wszystko dobrze działa. Inne (jak czytniki poczty lub przeglądarki WWW) przydają się do pracy i utrzymywania kontaktów z ludźmi.

### 13.1 ping

Polecenie `ping(8)` wysyła pakiet ICMP *ECHO\_REQUEST* (żądanie odpowiedzi) do danego komputera w sieci. Jeśli komputer odpowie, otrzymasz pakiet ICMP z powrotem. Dziwnie to brzmi? Możesz np. „spingować” adres IP, aby zobaczyć, czy komputer pod tym adresem jest włączony. Jeśli nie ma odpowiedzi, wiesz, że coś jest nie tak. Oto typowa rozmowa dwóch linuksiarzy:

*Użytkownik A:* Loki znowu zdechl.

*Użytkownik B:* Jesteś pewien?

*Użytkownik A:* Tjaa, pingowałem go, ale nie odpowiada.

Właśnie w takich sytuacjach `ping` okazuje się niezmiernie przydatne jako narzędzie codziennego użytku. Za jego pomocą można szybko sprawdzić, czy jakiś komputer jest włączony i podłączony do sieci. Oto przykład podstawowego wykorzystania:

```
% ping www.slackware.com
```

Istnieje oczywiście wiele opcji polecenia `ping(1)`, które są opisane na stronie podręcznika.

## 13.2 traceroute

Polecenie `traceroute(8)` w Slackware jest niezwykle przydatnym narzędziem diagnostyki sieci. `traceroute` wyświetla każdy komputer w sieci, przez który przechodzi pakiet danych, dążąc do celu. Możesz np. zobaczyć, o ile „przeskoków” (ang. „hop”) jesteś oddalony od strony `www Slackware`:

```
% traceroute www.slackware.com
```

Wyświetli się każdy komputer wraz z czasem odpowiedzi. Oto przykład:  
ut:

```
% 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 `ping`, ponieważ używa pakietów ICMP. Można go użyć z wieloma opcjami, opisanymi na stronie podręcznika.

## 13.3 Narzędzia DNS

System nazw domen (ang. Domain Name System – DNS) to magiczny protokół, dzięki któremu komputer zamienia nieznaczące dla niego nazwy domen, takie jak [www.slackware.com](http://www.slackware.com), na pełne znaczenia adresy IP, jak 64.57.102.34. Komputery nie potrafią kierować pakietów do [www.slackware.com](http://www.slackware.com), ale potrafią je kierować do adresu IP tej domeny. Dzięki temu łatwo jest zapamiętać komputery. Bez DNS musielibyśmy pamiętać, które adresy IP należą do jakich komputerów, i to zakładając, że IP się nie zmieniają. Nadawanie nazw komputerom jest wygodniejsze, ale jak przypisuje się te nazwy do konkretnych adresów IP?

### 13.1.1 host

Zajmuje się tym polecenie `host(1)`. Używa się go do przypisywania (mapowania) nazw do adresów IP. Jest to szybki i prosty, mało rozbudowany program.

```
% host www.slackware.com
www.slackware.com is an alias for slackware.com.
slackware.com has address 64.57.102.34
```

A jeśli chcemy przypisać adres IP do nazwy domeny?

## 13.3.2 nslookup

nslookup to stary jak świat i sprawdzony program. Obecnie jest uznany za przestarzały i prawdopodobnie zostanie usunięty z nowszych wersji Slackware. Nie ma nawet do niego strony podręcznika.

```
% 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

Najbardziej rozbudowanym programem jest dig(1) (domain information groper – wyszukiwarka informacji o domenie). To narzędzie, które potrafi wydobyć najwięcej informacji z serwera DNS, od wyszukiwań odwrotnych (ang. reverse lookups) do wpisów A, CNAME, MX, SP i TXT. Ma najwięcej opcji wiersza poleceń, więc jeśli go nie znasz, przeczytaj szczegółową stronę podręcznika.

```
% 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 SECTION:
;www.slackware.com.          IN      MX

;; ANSWER SECTION:
www.slackware.com.         76634  IN      CNAME   slackware.com.
slackware.com.            86400  IN      MX      1 mail.slackware.com.

;; AUTHORITY SECTION:
slackware.com.            86400  IN      NS      ns1.cwo.com.
slackware.com.            86400  IN      NS      ns2.cwo.com.

;; ADDITIONAL SECTION:
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
```

Tak właśnie działa dig. „@192.168.1.245” określa, którego serwera DNS użyć. [www.slackware.com](http://www.slackware.com) to nazwa domeny, o której informacji poszukuję, a „mx” to zastosowany typ wyszukiwania. Z powyższego wyniku wiem, że e-mail wysłany do [www.slackware.com](http://www.slackware.com) zostanie przesłany do „mail.slackware.com” w celu dostarczenia.

## 13.4 finger

Polecenie `finger(1)` (ang. palec) wyświetla informacje o danym użytkowniku. Podaje się mu jako argumenty nazwę użytkownika lub jego adres, a `finger` próbuje skontaktować się z odpowiednim serwerem i wydobyć od niego takie informacje, jak nazwa użytkownika, biuro, telefon itp. Oto przykład:

```
% finger johnc@idsoftware.com
```

`finger` może zwrócić w odpowiedzi nazwę użytkownika, status poczty elektronicznej, numery telefonów oraz pliki tzw. „dot plan” lub „dot project”. Oczywiście zwracane informacje zależą od serwera `finger`. Domyślnie serwer ze Slackware wyświetla:

- nazwę użytkownika
- numer pokoju
- numer telefonu domowego
- numer telefonu do pracy
- status zalogowania
- status poczty elektronicznej
- zawartość pliku `.plan` w katalogu domowym użytkownika
- zawartość pliku `.project` w katalogu domowym użytkownika

Pierwsze cztery pozycje można zdefiniować za pomocą polecenia `chfn`. Zapisuje ono te wartości w pliku `/etc/passwd`. Aby zmienić informacje w `.plan` lub `.project`, po prostu zmodyfikuj je w swoim ulubionym edytorze tekstu. Muszą one znajdować się w katalogu domowym i nazywać się dokładnie `.plan` i `.project`.

Wielu użytkowników „fingeruje” swoje konto ze zdalnego komputera, aby sprawdzić, czy mają nową pocztę. Możesz też zobaczyć obecny plan pracy użytkownika lub projekt, nad którym pracuje.

Podobnie jak inne polecenia, `finger` ma wiele opcji opisanych na stronie podręcznika.

## 13.5 telnet

Ktoś kiedyś powiedział, że `telnet(1)` to najbardziej zaręczista rzecz, jaką w życiu widział w komputerach. Możliwość zdalnego logowania się na inne komputery i wykonywania na nich pracy to coś, co odróżnia systemy uniksowe i uniksopodobne od pozostałych.

Dzięki `telnetowi` możesz zalogować się do komputera, tak jakbyś siedział przed terminalem. Po zweryfikowaniu nazwy użytkownika i hasła otrzymujesz dostęp do powłoki. Możesz na niej robić wszystko, co można robić w powłoce tekstowej: pisać listy elektroniczne, czytać grupy dyskusyjne, zarządzać plikami itd. Jeśli masz uruchomiony X i „telnetujesz” się na inną maszynę, możesz uruchamiać programy X na zdalnym komputerze, a wyświetlać je na swoim.

W ten sposób logujesz się na zdalnym komputerze:

```
% telnet <nazwacomputera>
```

Jeśli komputer odpowie, wyświetli się znak zgłoszenia logowania. Wpisz nazwę użytkownika i hasło. To wszystko, masz dostęp do powłoki. Aby zakończyć sesję telnet, użyj polecenia `exit` lub `logout`.

Uwaga: telnet nie przesyła informacji w formie zaszyfrowanej. Wszystko jest wysyłane w postaci czystego tekstu, nawet hasła. Używanie telnetu w Internecie nie jest zalecane. Zamiast tego powinno się używać SSH (Secure Shell – bezpieczna powłoka), która szyfruje wszelkie przesyłane dane i jest dostępna za darmo.

### 13.5.1 Inne sposoby wykorzystania telnetu

Przekonaaliśmy cię, żeby nie używać telnetu do logowania się na zdalnym komputerze. Pokażemy teraz jego kilka innych, przydatnych zastosowań.

Można go użyć, aby podłączyć się do komputera przez dany port.

```
% telnet <nazwacomputera> [port]
```

Jest to przydatne, kiedy chcesz przetestować działanie jakiejś usługi, a potrzebujesz pełnej kontroli nad poleceniami i tym co się dzieje. W ten sposób możesz interaktywnie testować lub wykorzystywać serwer SMTP, POP3, HTTP itp.

W ten sposób możesz zatelnetować do serwera HTTP na port 80 i uzyskać kilka podstawowych informacji:

```
% 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 z innymi protokołami wykorzystującymi czysty tekst, jeśli wiesz, do których portów się podłączyć i jakie polecenia wpisywać.



## 13.6 Bezpieczna powłoka – The Secure Shell

Bezpieczna powłoka (SSH) cieszy się dziś takim poważaniem, jak niegdyś telnet. ssh(1) pozwala na podłączenie się do zdalnego komputera i wykonywanie na nim programów tak, jak na lokalnej maszynie. Jednak ssh szyfruje wszelkie dane przesyłane między oboma komputerami, więc nawet jeśli ktoś przechwyci konwersację, nie będzie w stanie jej zrozumieć. Oto typowa sesja ssh:

```
% 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.
```

W ten sposób połączyłem się przez ssh z carrier.lizella.net, aby sprawdzić prawa własności i dostępu do pliku MANIFEST.

## 13.7 Poczta elektroniczna

Poczta elektroniczna (e-mail, email) to jedna z najpopularniejszych rzeczy w Internecie. W 1998 roku liczba wiadomości elektronicznych przekroczyła liczbę listów wysłanych zwykłą pocztą. Jest ona bardzo użyteczna i powszechnie stosowana.

W Slackware jest dostępny standardowy serwer poczty oraz kilka klientów. Wszystkie poniżej opisane klienty są tekstowe. Wielu użytkowników Windows krzywi się na ten widok, ale klienty tekstowe są bardzo wygodne, zwłaszcza jeśli sprawdzasz pocztę na zdalnym komputerze. Nie ma się czego bać, jest wiele klientów graficznych, jak np. Kmail dostępny w KDE. Jeśli chcesz z któregoś korzystać, przeczytaj dołączoną do niego pomoc.

### 13.7.1 pine

Jak to się mówi, pine(1) (ang. sosna) to nie elm (ang. wiąz). Ten program stworzono na Uniwersytecie Waszyngtońskim dla studentów, którzy potrzebowali prostego klienta poczty elektronicznej i grup dyskusyjnych. Jest to obecnie jeden z najpowszechniej używanych czytników poczty, dostępny dla różnych systemów uniksowych, a nawet dla Windows.

#### Rys. 13-2. Główny ekran Pine

```

PINE 4.58      MAIN MENU                               Folder: INBOX  2 Messages

?      HELP          - Get help using Pine
C      COMPOSE MESSAGE - Compose and send a message
I      MESSAGE INDEX - View messages in current folder
L      FOLDER LIST   - Select a folder to view
A      ADDRESS BOOK  - Update address book
S      SETUP         - Configure Pine Options
Q      QUIT          - Leave the Pine program

Copyright 1989-2003. PINE is a trademark of the University of Washington.
[Folder "INBOX" opened with 2 messages]
? Help          P PrevCmd          R RelNotes
0 OTHER CMDS > [Compose] N NextCmd        K KBLock

```

Widać menu poleceń oraz rząd klawiszy poleceń na dole. Jest to dość rozbudowany program, dlatego nie będziemy opisywać każdej funkcji.

Aby zobaczyć zawartość skrzynki z pocztą przychodzącą (inbox), wciśnij `i`. Wyświetlą się wiadomości wraz z datą wysłania, autorem i tematem. Podświetl tę, którą chcesz przeczytać, i wciśnij `enter`. Gdy wciśniesz `r`, uruchomi się opcja odpowiadania na list. Po napisaniu odpowiedzi wciśnij **Ctrl+X**, aby ją wysłać. Ponownie wciśnij `i`, aby powrócić do listy wiadomości.

Jeśli chcesz skasować wiadomość, wciśnij `d`. Podświetlona wiadomość zostanie oznaczona jako przeznaczona do usunięcia. Pine usuwa takie wiadomości, gdy kończysz pracę z nim. Możesz też zapisywać wiadomości w folderach. Lista folderów wyświetli się po wciśnięciu `l`. Na liście wiadomości wciśnij `s`, aby zapisać wiadomość w innym folderze. Wyświetli się pytanie o nazwę folderu, w którym wiadomość ma być zapisana.

Pine ma mnóstwo funkcji. Koniecznie przeczytaj jego stronę podręcznika. Zawiera ona najświeższe informacje o programie.

### 13.7.2 e1m

e1m(1) to kolejny popularny tekstowy klient poczty elektronicznej. Nie jest tak łatwy w obsłudze, jak pine, ale jest o wiele starszy.

Rys 13-3. Główny ekran e1m

```
Mailbox is '/var/mail/root' with 2 messages [ELM 2.5 PL6]

  1  Sep 17  Patrick J. Volkerd (174)  Welcome to Linux (Slackware 9.1)!
0  2  Sep 17                               (45)  Register with the Linux counter pr

You can use any of the following commands by pressing the first character:
d)delete or u)ndelete mail, m)ail a message, r)eply or f)orward mail, q)uit
To read a message, press <return>.  j = move down, k = move up, ? = help

Command:
```

Domyślnie przechodzisz od razu do skrzynki wiadomości przychodzących. Wyświetla się lista wiadomości wraz z ich numerami, datą, autorem i tematem. Użyj klawiszy strzałek, aby podświetlić wiadomość i wciśnij `enter`, aby ją odczytać.

Aby utworzyć nową wiadomość, wciśnij `m` na głównym ekranie. Klawisz `d` oznaczy wiadomość jako „do usunięcia”. Klawisz `r` uruchomi pisanie odpowiedzi na aktualnie czytany list. Wszystkie te klawisze są wyświetlone na dole ekranu wraz ze znakiem zgłoszenia.

Przed korzystaniem z programu `elm` przeczytaj jego stronę podręcznika.

### 13.7.3 mutt

„Wszystkie klienty poczty są do bani. Ten jest trochę mniej do bani”. Oryginalny interfejs `mutt` jest oparty na `elmie`, z dodanymi funkcjami obecnymi w innych programach tego typu. Efektem jest hybryda `mutt`.

Oto niektóre z funkcji `mutt`:

- obsługa kolorów
- sortowanie wiadomości według wątków
- obsługa MIME i PGP/MIME
- obsługa POP3 i IMAP
- obsługa wielu formatów skrzynki pocztowej (mbox, MMDF, MH, maildir)
- *duże* możliwości dostosowania do własnych potrzeb

Rys. 13-4. Główny ekran `mutt`

```
q:Quit d:Del u:Undel s:Save m:Mail r:Reply g:Group ?:Help
1 0 F Sep 17 To root ( 23) Register with the Linux counter project
2 + Sep 17 Patrick J. Volk ( 153) Welcome to Linux (Slackware 9.1)!

-----Mutt: /var/mail/root [Msgs:2 Old:1 9.5K]----- (date/date)----- (all)-----
```

Jeśli szukasz klienta poczty elektronicznej, w którym możesz kontrolować absolutnie wszystko, `mutt` jest dla Ciebie. Możesz zmienić wszystkie ustawienia domyślne, skróty klawiaturowe, a nawet napisać własne makra.

Przeczytaj stronę podręcznika `muttrc`, która opisuje wszystkie opcje konfiguracji. Przejrzyj też dołączony do programu przykładowy plik `muttrc`.

### 13.7.4 `nail`

`nail(1)` to klient poczty obsługiwany z wiersza poleceń. Jest raczej prymitywny i ciężko nazwać jego interfejs użytkownika wygodnym. Przydaje się, kiedy potrzebujesz szybko wysłać jakiś email, napisać skrypt do masowej wysyłki poczty, przetestować MTA itp. Zwróć uwagę, że Slackware tworzy dowiązania symboliczne do `nail` jako `/usr/bin/mail` i `/usr/bin/mailx`. Wszystkie trzy polecenia wywołują ten sam program. Bardzo często mówi się o `mail`, mając na myśli `nail`.

Oto podstawowa składnia wiersza poleceń:

```
% mailx <temat> <adres>
```

`Mailx` czyta treść wiadomości ze standardowego wejścia. Można więc „catnąć” jakiś plik do tego polecenia jako argument albo po prostu wpisać tekst i wcisnąć **Ctrl+D** po wpisaniu wiadomości.

W taki sposób można np. przesłać do kogoś plik ze źródłami programu:

```
% cat randomfunc.c | mail -s "To te źródła" asdf@example.net
```

Więcej możliwości `mail` jest opisanych na stronie podręcznika, warto więc ją przeczytać.

## 13.8 Przeglądarki

Kiedy ludzie słyszą słowo „Internet”, pierwszą rzeczą, o której myślą, jest „surfowanie po Sieci”. Albo „Oglądanie stron WWW przy użyciu przeglądarki internetowej”. Z punktu widzenia przeciętnego użytkownika jest to najpowszechniejszy sposób wykorzystania Internetu.

W Slackware znajdują się popularne graficzne przeglądarki www w grupie „XAP”, a także przeglądarki tekstowe w grupie „N”. Pokrótkie omówimy kilka najpopularniejszych opcji.

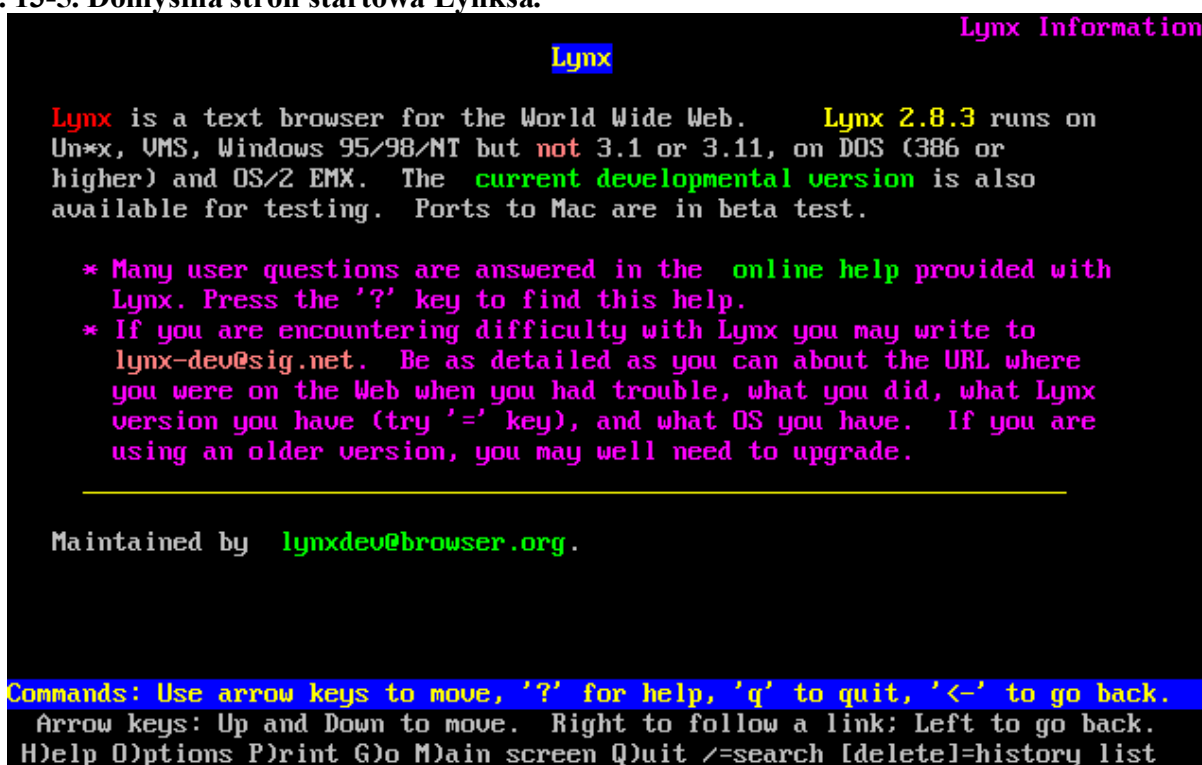
### 13.8.1 lynx

`lynx(1)` jest tekstową przeglądarką internetową. Stanowi doskonały sposób na szybkie dotarcie do informacji w Internecie. Czasami grafika po prostu przeszkadza, kiedy dokładnie wiesz, czego szukasz.

Aby go uruchomić, po prostu wpisz `lynx` w wierszu poleceń:

```
% lynx
```

Rys. 13-5. Domyślna stron startowa Lynksa.



```

                                     Lynx Information
                                     Lynx
Lynx is a text browser for the World Wide Web.   Lynx 2.8.3 runs on
Un*x, VMS, Windows 95/98/NT but not 3.1 or 3.11, on DOS (386 or
higher) and OS/2 EMX. The current developmental version is also
available for testing.  Ports to Mac are in beta test.

* Many user questions are answered in the online help provided with
  Lynx. Press the '?' key to find this help.
* If you are encountering difficulty with Lynx you may write to
  lynx-dev@sig.net. Be as detailed as you can about the URL where
  you were on the Web when you had trouble, what you did, what Lynx
  version you have (try '=' key), and what OS you have. If you are
  using an older version, you may well need to upgrade.

-----
Maintained by lynxdev@browser.org.

Commands: Use arrow keys to move, '?' for help, 'q' to quit, '<' to go back.
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

Możesz podać, jaką stronę `lynx` ma otworzyć:

```
% lynx http://www.slackware.com
```

Na dole ekranu `lynx` wyświetla klawisze poleceń wraz z ich krótkim opisem. Strzałki w górę i w dół

przewijają dokument. Enter wybiera podświetlony odsyłacz, a strzałka w lewo przechodzi do poprzednio odwiedzonej strony. Klawisz d pobiera podświetlony plik na dysk, g wyświetla okno „Go” (Idź), w którym można wpisać URL do otwarcia.

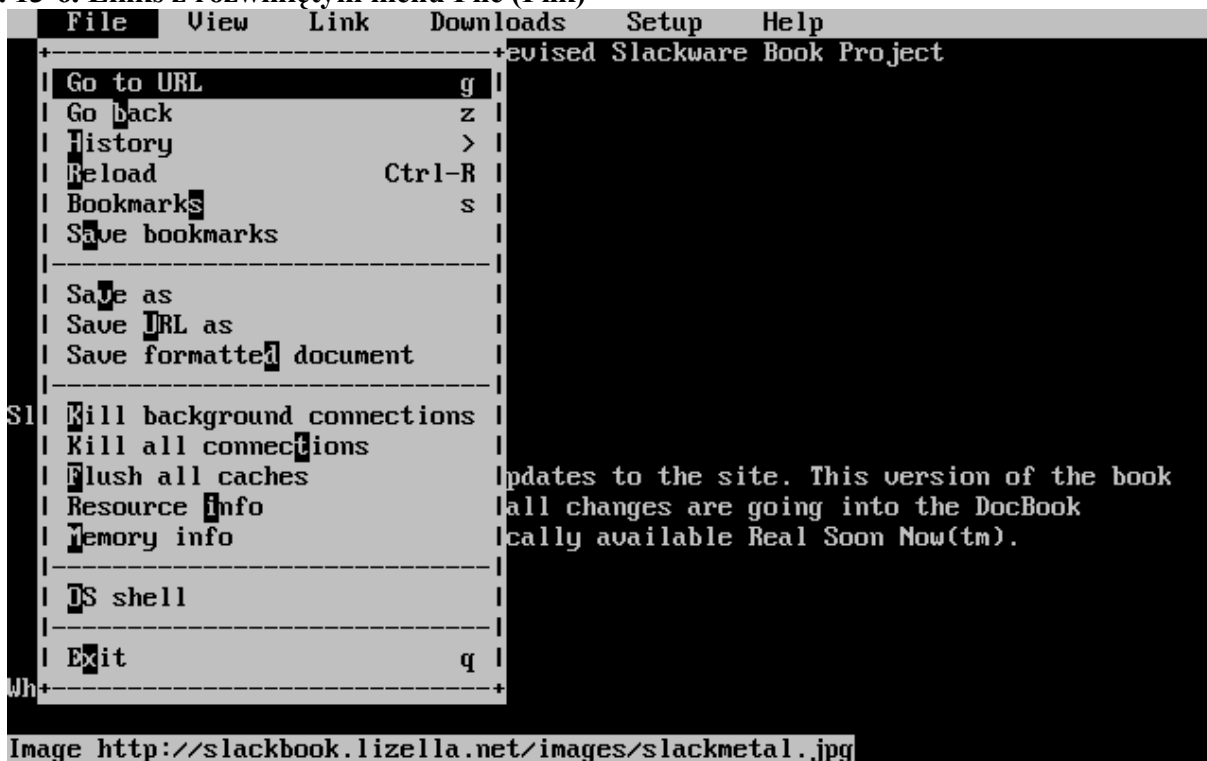
Lynx ma wiele innych poleceń. Przeczytaj jego stronę podręcznika lub wciśnij h, aby wyświetlić ekran pomocy.

### 13.8.2 links

Podobnie jak lynx, links jest przeglądarką działającą w trybie tekstowym. Nawiguje się w niej za pomocą klawiatury. Po wciśnięciu Esc na górze ekranu pojawia się wygodne rozwijane menu. Dzięki temu obsługa programu jest prosta i nie trzeba uczyć się wszystkich skrótów klawiaturowych. Osoby nieużywające przeglądarek tekstowych na co dzień z pewnością docenią tę funkcję.

Links ma lepszą obsługę ramek i tabel niż lynx.

Rys. 13-6. Links z rozwiniętym menu File (Plik)



### 13.8.3 wget

wget(1) to obsługiwany z wiersza poleceń program, który służy do pobierania plików z podanego URL. Nie jest to przeglądarka stron WWW. Używa się go raczej do pobrania części lub całości jakiejś strony internetowej w celu przeglądania w trybie offline albo do szybkiego pobierania pojedynczych plików z serwerów HTTP lub FTP. Oto podstawowa składnia:

```
% wget <url>
```

Możesz przekazywać opcje do wget. W ten sposób pobierzesz całą witrynę Slackware:

```
% wget --recursive http://www.slackware.com
```

wget utworzy katalog [www.slackware.com](http://www.slackware.com) i umieści w nim pliki, tak jak oryginalna strona.

wget może też pobierać pliki z serwerów FTP. Wystarczy podać URL 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]
```

To polecenie ma wiele opcji, które są szczególnie przydatne w pisaniu skryptów związanych ze stronami (np. robieniem kopii lustrzanych itp.) Przeczytaj stronę podręcznika.

## 13.9 Klienty FTP

FTP to skrót od File Transfer Protocol (protokół transferu plików). Umożliwia on przesyłanie plików między komputerami. Istnieją serwery i klienty FTP. W tej sekcji omawiamy klienty.

Dla ciekawskich: „klient” to Ty. „Serwer” to komputer, który odpowiada na Twoje żądanie FTP i pozwala Ci się zalogować. Możesz pobierać (ang. download) pliki z serwera i wysyłać je (ang. upload) na serwer. Sam klient nie przyjmuje połączeń FTP, może się tylko podłączyć do serwerów.

### 13.9.1 ftp

Aby podłączyć się do serwera FTP, wpisz polecenie `ftp(1)`, podając nazwę komputera:

```
% ftp <nazwakomputera> [port]
```

Jeśli na komputerze jest uruchomiony serwer FTP, zapyta on o nazwę użytkownika i hasło. Możesz zalogować się jako Ty sam lub jako użytkownik anonimowy – „anonymous”. Anonimowe serwery FTP są szczególnie często wykorzystywane jako archiwa z oprogramowaniem. Aby np. pobrać Linuksa Slackware z FTP, musisz użyć anonimowego FTP.

Po połączeniu będziesz miał dostęp do znaku zgłoszenia `ftp:ftp>`. FTP ma specjalne polecenia, ale są one podobne do standardowych. Oto niektóre z nich:

Tabela 13-1. Polecenia `ftp`

Polecenie	Cel
-----------	-----

ls	wyświetl listę plików
cd <katalog>	przejdź do katalogu
bin	ustaw binarny tryb transferu
ascii	ustaw tekstowy tryb transferu
get <plik>	pobierz plik z serwera
put <plik>	wyślij plik na serwer
hash	przełącz wyświetlanie znaku „#” dla każdego przetransferowanego bloku danych
tick	przełącz wskaźnik licznika bajtów
prom	przełącz tryb interaktywny pobierań
mget <maska>	pobierz plik lub grupę plików, można używać symboli wieloznacznych
mput <maska>	wyślij plik lub grupę plików, można używać symboli wieloznacznych
quit	wyloguj się z serwera FTP

Istnieją również polecenia, które same się wyjaśniają: `chmod`, `delete` (usuń), `rename` (zmień nazwę), `rmdir`. Pełna lista poleceń z ich wyjaśnieniem wyświetla się po wpisaniu `help` lub `?`.

Ftp jest prostym programem, ale nie ma typowego interfejsu użytkownika, do jakiego przywykło wiele osób. Wiele opcji `ftp(1)` jest opisanych na stronie podręcznika.

```
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) (wymawiane „nik-ef-te-pe”) to dostępna w Slackware alternatywa dla tradycyjnego klienta ftp. Również działa w trybie tekstowym, ale ma więcej funkcji niż ftp, m.in.:

- uzupełnianie tabulatorem
- plik z zakładkami
- szersze możliwości używania symboli wieloznacznych
- historię poleceń

Domyślnie ncftp próbuje zalogować się anonimowo do podanego serwera. Można wymusić znak zgłoszenia logowania za pomocą opcji „-u”. Po zalogowaniu polecenia są te same, co w ftp, ale interfejs jest lepszy i przypomina 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

### 13.10.1 wall

Polecenie wall(1) to szybki sposób, aby przekazać jakąś wiadomość wszystkim zalogowanym użytkownikom. Używa się go tak:

```
% wall [plik]
```

Zawartość pliku [plik] zostanie wyświetlona na terminalach wszystkich aktualnie zalogowanych użytkowników. Jeśli nie podasz pliku, wall będzie czytać ze standardowego wejścia, możesz więc po prostu wpisać tekst i wcisnąć **Ctrl+D**.

Wall nie ma zbyt wielu funkcji. Nadaje się co najwyżej do oznajmienia użytkownikom, że zamierzasz przeprowadzić jakieś ważne prace administracyjne albo zrestartować system, aby mieli czas na zapisanie wyników swojej pracy i wylogowanie się :).

## 13.10.2 talk

Polecenie `talk(1)` (ang. rozmawiać) umożliwia rozmowę dwóch użytkowników. Dzieli ono poziomo ekran na dwie części. Aby rozpocząć rozmowę, wpisz:

```
% talk <użytkownik> [nazwa_tty]
```

Rys. 13-7. Dwaj użytkownicy podczas sesji `talk`



Jeśli podasz tylko użytkownika, `talk` zakłada, że żądasz rozmowy z użytkownikiem lokalnym i tylko takich użytkowników poszukuje. Nazwa terminala (`tty`) jest wymagana, jeśli chcesz rozmawiać na konkretnym terminalu z kimś, kto jest zalogowany kilkakrotnie. Informacje potrzebne dla `talk` możesz uzyskać za pomocą polecenia `w(1)`.

`Talk` umożliwia też rozmowę ze użytkownikiem na zdalnym komputerze. Jako nazwę użytkownika podaje się wówczas adres email. `Talk` spróbuje skontaktować się ze zdalnym użytkownikiem na tej maszynie.

`Talk` ma ograniczone funkcje. Umożliwia rozmowę tylko dwóch ludzi i działa w trybie half-duplex.

## 13.10.3 ytalk

Program `ytalk(1)` to zgodny wstecz zastępnik `talka`. Jest dostępny w Slackware jako polecenie `ytalk`. Składnia jest podobna, ale różni się paroma szczegółami:

```
% ytalk <użytkownik>[#nazwa_tty]
```

Rys. 13-8. Dwaj użytkownicy podczas sesji `ytalk`

```
-----= YTalk version 3.1.1 =-----
test

-----= root@192.168.1.132#tty1 =-----
this is a test
```

Nazwa użytkownika i terminal są takie same, jak w `talk`, ale trzeba je połączyć znacznikiem „#”. `Ytalk` ma wiele zalet:

- obsługuje więcej niż 2 rozmówców
- ma menu opcji, które można wywołać klawiszem Esc
- można wyjść z powłoki, nie zakańczając sesji
- i o wiele więcej...

Jeśli jesteś administratorem serwera, upewnij się, że port `ntalk` jest włączony w `/etc/inetd.conf`. `Ytalk` potrzebuje go do działania.

## Rozdział 14. Bezpieczeństwo

Bezpieczeństwo każdego systemu jest niezwykle istotne. Pomaga chronić ważne dane, powstrzymuje ludzi przed przeprowadzaniem ataków z Twojego komputera. W tym rozdziale są opisane podstawowe czynności zmierzające do zabezpieczenia Linuksa Slackware przed włamywaczami domorosłymi (script kiddies), naprawdę groźnymi (crackerami) i przed wściekłymi chomikami. Pamiętaj, że to tylko początek zabezpieczania systemu. Bezpieczeństwo to proces, a nie stan.

### 14.1 Wyłączanie usług

Pierwszym krokiem po zainstalowaniu Slackware powinno być wyłączenie wszystkich niepotrzebnych usług. Każda z nich stanowi potencjalne zagrożenie dla bezpieczeństwa systemu, dlatego należy uruchamiać tylko niezbędne. Usługi są uruchamiane z dwóch miejsc: demona `inetd` i skryptów inicjalizujących.

## 14.1.1 Usługi uruchamiane przez inetd

Wiele demonów w Slackware jest uruchamianych przez `inetd`. Sam `inetd` też jest demonem, który nasłuchuje na wszystkich portach używanych przez usługi skonfigurowane tak, aby były uruchamiane właśnie przez niego. `Inetd` uruchamia instancję odpowiedniego demona za każdą próbą połączenia sieciowego. Demony uruchamiane przez `inetd` wyłącza się, zakomentowując właściwe wiersze w `/etc/inetd.conf`. Po otwarciu tego pliku w ulubionym edytorze tekstów (np. `vi`) zobaczysz wpisy podobne do tego:

```
telnet stream tcp      nowait  root    /usr/sbin/tcpd  in.telnetd
```

Możesz wyłączyć tę usługę (i wszystkie inne, których nie potrzebujesz), zakomentowując je (tj. dodając krzyżyk „#” na początku wiersza). Zakomentowany wpis wygląda tak:

```
#telnet stream tcp      nowait  root    /usr/sbin/tcpd  in.telnetd
```

Po zrestartowaniu `inetd` ta usługa będzie wyłączona. `Inetd` restartuje się poleceniem:

```
# kill -HUP $(cat /var/run/inetd.pid)
```

## 14.1.2 Usługi uruchamiane ze skryptów inicjalizujących

Pozostałe usługi uruchamiane podczas startu komputera są wywoływane ze skryptów inicjalizujących w katalogu `/etc/rc.d`. Można je wyłączyć na dwa sposoby: albo zdejmując prawo wykonania danego skryptu, albo zakomentować odpowiednie wiersze w skrypcie.

Na przykład, demon SSH jest uruchamiany przez własny skrypt, `/etc/rc.d/rc.sshd`. Można go wyłączyć tak:

```
# chmod -x /etc/rc.d/rc.sshd
```

Jeśli jakaś usługa nie ma odrębnego skryptu, trzeba ją wyłączyć, zakomentowując odpowiednie wiersze wewnątrz skryptu. Na przykład, demon `portmap` jest uruchamiany przez następujące wpisy w `/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 go wyłączyć, dodając symbole `#` na początku wierszy, które ich jeszcze nie mają:

```
# 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 zostaną wprowadzone w życie po zrestartowaniu komputera lub po wyjściu z poziomu pracy 3 lub 4 i powrocie do niego. Aby to zrobić, wpisz w konsoli (po przejściu do poziomu pracy 1 będziesz się musiał znowu zalogować):

```
# telinit 1
# telinit 3
```

## 14.2 Kontrola dostępu do komputera

### 14.2.1 iptables

iptables to program do konfiguracji filtrowania pakietów w jądrach Linuksa 2.4 i wyższych. Jądro 2.4 (dokładniej 2.4.5) zostało wprowadzone po raz pierwszy (jako opcja) w Slackware 8.0, a jako domyślne w 8.1. Ta sekcja opisuje tylko jego podstawy. Więcej szczegółów jest na stronie <http://www.netfilter.org>. Polecenia można wpisać do skryptu `/etc/rc.d/rc.firewall`, który musi mieć ustawione prawo wykonania, aby zasady filtrowania były wprowadzane w życie podczas startu systemu. Pamiętaj, że niewłaściwe polecenia iptables mogą całkowicie odciąć Cię od komputera. O ile nie jesteś w 100% pewny swoich umiejętności, zawsze upewnij się, że masz bezpośredni dostęp do swojej maszyny.

Pierwszą rzeczą, jaką robi większość ludzi, jest ustawienie domyślnej polityki łańcuchów przychodzących na DROP (blokowanie):

```
# iptables -P INPUT DROP
# iptables -P FORWARD DROP
```

Po zablokowaniu wszelkiego ruchu przychodzącego można zacząć odblokowywać niektóre jego rodzaje. Na początek trzeba zezwolić na ruch dla sesji, które zostały już otwarte:

```
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Aby nie blokować działania aplikacji, które korzystają z adresu pętli zwrotnej, warto dodać:

```
# iptables -A INPUT -s 127.0.0.0/8 -d 127.0.0.0/8 -i lo -j ACCEPT
```

Ta zasada zezwala na każdego rodzaju ruch do i z adresów 127.0.0.1/8 (127.0.0.1-127.255.255.255) na urządzeniu interfejsu pętli zwrotnej (loopback – `lo`). Podczas tworzenia zasad należy być tak dokładnym, jak to możliwe, aby być pewnym, że nie zezwalamy na jakieś niewłaściwe akcje. Jednak zasady, które zezwalają na zbyt mało, oznaczają więcej zasad i więcej pisania.

Następnie trzeba zezwolić na dostęp do serwerów uruchomionych na komputerze. Jeśli np. chcesz

prowadzić serwer WWW, wpisz coś podobnego do:

```
# iptables -A INPUT -p tcp --dport 80 -i ppp0 -j ACCEPT
```

To zezwala na dostęp z każdego komputera do portu 80 na Twojej maszynie za pośrednictwem interfejsu ppp0. Możesz ograniczyć dostęp do serwera tylko dla niektórych komputerów. To zezwala na dostęp do usługi WWW z adresu 64.57.102.34:

```
# iptables -A INPUT -p tcp -s 64.57.102.34 --dport 80 -i ppp0 -j ACCEPT
```

Zezwalanie na pakiety ICMP jest przydatne do celów diagnostycznych. Umożliwia to zasada: is:

```
# iptables -A INPUT -p icmp -j ACCEPT
```

Wiele osób uruchamia tzw. maskaradę, czyli translację adresów sieciowych (Network Address Translation – NAT) na komputerze pełniącym rolę bramki, aby inne maszyny w sieci miały za jej pośrednictwem dostęp do Internetu. Oto odpowiednia zasada:

```
# iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
```

Musisz też włączyć przekazywanie IP (IP forwarding). Tymczasowo robi się to poleceniem:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Aby przekazywanie IP było włączone na stałe (tzn. włączane po restarcie), musisz zmodyfikować /etc/rc.d/rc.inet2 za pomocą swojego ulubionego edytora, zmieniając wiersz:

```
IPV4_FORWARD=0
```

... na taki:

```
IPV4_FORWARD=1
```

Więcej informacji o NAT znajduje się w NAT HOWTO ([http://www.netfilter.org/documentation/HOWTO/NAT\\_HOWTO.txt](http://www.netfilter.org/documentation/HOWTO/NAT_HOWTO.txt)).

## 14.2.2 tcpwrappers

Program tcpwrappers kontroluje dostęp do demonów na poziomie aplikacji, a nie IP. Stanowi on dodatkowy poziom zabezpieczeń, kiedy kontrola dostępu na poziomie IP (np. Netfilter) nie działa poprawnie (bo np. podczas kompilacji jądra zapomniałeś włączyć obsługę iptables – tcpwrappers dalej będzie pomagał chronić Twój system).

Dostęp do usług chronionych przez tcpwrappers kontrolują pliki /etc/hosts.allow i /etc/hosts.deny.

Większość osób ma tylko jeden wpis w `/etc/hosts.deny`, który domyślnie zabrania dostępu do wszystkich demonów:

```
ALL : ALL
```

Po jego wpisaniu możesz skoncentrować się na przyznawaniu dostępu dla poszczególnych komputerów, domen lub zakresów IP. Robi się to w pliku `/etc/hosts.allow`, który ma taki sam format.

Najczęściej akceptuje się wszystkie połączenia z lokalnego komputera:

```
ALL : 127.0.0.1
```

Dostęp do demona `sshd` z adresów `192.168.0.0/24` można umożliwić na jeden z dwóch sposobów:

```
sshd : 192.168.0.0/24
sshd : 192.168.0.
```

Można również umożliwiać dostęp tylko dla komputerów w danej domenie. Oto odpowiednia zasada (pamiętaj, że jej działanie zależy od tego, czy odwrotny wpis DNS łączącego się komputera jest zaufany, nie zalecam używania jej na komputerach podłączonych do Internetu):

```
sshd : .slackware.com
```

## 14.3 Aktualizowanie systemu

### 14.3.1 lista mailingowa *slackware-security*

Kiedykolwiek pojawia się problem z bezpieczeństwem, który dotyczy Slackware (np. luka w oprogramowaniu dołączonym do Slackware, z wyjątkiem programów w `/extra i /pasture`), do wszystkich subskrybentów listy *slackware-security@slackware.com* jest wysyłany email. Takie emaile zawierają informacje, skąd pobrać zaktualizowane wersje oprogramowania lub jak rozwiązać problem w inny sposób (o ile taki istnieje).

Sposób zapisania się na listy mailingowe Slackware jest opisany w sekcji 2.2.2.

### 14.3.2 Katalog `/patches`

Kiedykolwiek są udostępniane zaktualizowane pakiety dla danej wersji Slackware (w przypadku oficjalnych wersji dystrybucji najczęściej tylko usuwające lukę w bezpieczeństwie), są one umieszczone w katalogu `/patches`. Pełna ścieżka do poprawionych pakietów (tzw. „łat”, ang. patches) zależy od serwera lustrzanego, ale ma formę `/ścieżka/do/slackware-x.x/patches/`.

Przed ich zainstalowaniem należy zweryfikować sumę kontrolną `md5sum` pakietu. Program `md5sum(1)` jest uruchamiany z wiersza poleceń. Tworzy on niepowtarzalny numer matematycznej funkcji mieszającej (ang. hash). Zmiana nawet tylko jednego bitu w pliku spowoduje wygenerowanie

innej wartości md5sum.

```
% md5sum pakiet-<wersja>-<architektura>-<podwersja>.tgz  
6341417aa1c025448b53073a1f1d287d pakiet-<wersja>-<architektura>-<podwersja>.tgz
```

Porównaj tę sumę z odpowiednim wpisem w pliku CHECKSUMS.md5 w głównym katalogu slackware-\$WERSJA (dla łat znajduje się on w /patches) lub w emailu wysłanym na listę slackware-security.

Jeśli masz plik z wartościami md5sum, możesz podać go do programu md5sum jako argument za pośrednictwem opcji -c.

```
# 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, pliki, które md5sum uzna za poprawne, są oznaczone jako „OK” (w porządku). Pliki, które są niepoprawne, są oznaczone jako „FAILED” (błędne). (Tak, wiem, że to była obelga dla Twojej inteligencji. Jak ty ze mną wytrzymujesz?)

## Rozdział 15. Archiwizacja plików

### 15.1 gzip

gzip(1) to program kompresujący GNU. Kompresuje on pojedyncze pliki. Podstawowa składnia to:

```
% gzip nazwapliku
```

Plik wynikowy będzie miał nazwę nazwapliku.gz i najczęściej mniejszy rozmiar niż plik wejściowy. Zauważ, że nazwapliku.gz zastąpi nazwapliku. To znaczy, że nazwapliku przestanie istnieć, mimo że kopia skompresowana gzipem będzie dostępna. Zwykle pliki tekstowe kompresują się w dużym stopniu. Obrazy jpeg, pliki mp3 itp. nie będą się dobrze kompresować, ponieważ same w sobie są już skompresowane. Podstawowa składnia stanowi kompromis między rozmiarem pliku a czasem kompresji. Tak można osiągnąć maksymalny stopień kompresji:

```
% gzip -9 nazwapliku
```

Czas kompresji będzie dłuższy, ale wynikowy plik będzie możliwie najmniejszy. Mniejsze wartości spowodują, że plik będzie się kompresował szybciej, ale nie w tak dużym stopniu.

Skompresowane pliki można zdekompresować na dwa sposoby, które tak naprawdę wykorzystują jeden program. Gzip potrafi zdekompresować każdy plik, którego rozszerzenie rozpozna. Znane rozszerzenia to: .gz, -gz, .z, -z, .Z, -Z. Pierwsza metoda to wywołanie gunzip(1) z nazwą pliku jako



argumentem:

```
% gunzip nazwapliku.gz
```

Pozostawi to zdekompresowany plik wejściowy w bieżącym katalogu, a rozszerzenie `.gz` zostanie usunięte. `Gunzip` jest częścią `gzip` i działa identycznie z `gzip -d`. Dlatego `gzip` często wymawia się jak `gunzip`, bo tak fajniej brzmi. :^)

## 15.2 bzip2

Program `bzip2(1)` to alternatywny program kompresujący w Slackware. Używa innego algorytmu kompresji niż `gzip`, co ma wady i zalety. Główną zaletą `bzip2` jest rozmiar pliku wynikowego. `Bzip2` prawie zawsze kompresuje lepiej niż `gzip`. Niekiedy różnica może być bardzo duża. Szczególnie korzystają na tym osoby z wolnymi łączami modemowymi. Pamiętaj, że kiedy pobierasz oprogramowanie z publicznego serwera ftp, netykieta radzi, aby pobierać wersje `bz2`, a nie `gz`. Dzięki temu serwer hostowany (utrzymywany) przez dobrych ludzi jest mniej obciążony.

Wadą `bzip2` jest większe zużycie CPU niż w przypadku `gzip`. „Bzipowanie” pliku z reguły trwa dłużej i zajmuje więcej zasobów procesora niż „gzipowanie”. Podczas wybierania programu do kompresji rozważ, co jest ważniejsze: szybkość czy wielkość pliku wynikowego.

`Bzip2` ma prawie identyczną składnię jak `gzip`, więc nie będziemy się nad nią rozwodzić. Podobnie jak `gunzip`, `bunzip2` jest identyczne z `bzip2 -d`. Podstawową różnicą jest inne rozszerzenie - `.bz2`.

```
% bzip2 nazwapliku
% bunzip2 nazwapliku.bz2
% bzip2 -9 nazwapliku
```

## 15.3 tar

`tar(1)` to program GNU do archiwizacji na taśmach. Łączy on pliki i katalogi w jeden duży plik. Dzięki temu można skompresować całe drzewo katalogu, co jest niemożliwe przy użyciu samego `bzip2` lub `gzip`. `Tar` ma wiele opcji, które są opisane na stronie podręcznika. W tej sekcji opisujemy tylko podstawowe zastosowania.

Najczęściej używa się `tar` do zdekompresowania i zdearchiwizowania pakietu pobranego ze strony WWW lub serwera ftp. Większość takich plików ma rozszerzenie `.tar.gz`. Potocznie nazywa się to „tarball”. To znaczy, że pewna liczba plików została zarchiwizowana przy użyciu `tar`, a następnie skompresowana przez `gzip`. Czasem plik ma rozszerzenie `.tar.Z`, co oznacza to samo, ale spotyka się w starszych systemach uniksowych.

Możesz też spotkać pliki z rozszerzeniem `.tar.bz2`. W ten sposób są dystrybuowane źródła jądra Linuksa, ponieważ plik do pobrania ma mniejszy rozmiar. Nietrudno się domyślić, że są to pliki zarchiwizowane przez `tar` i skompresowane przez `bzip2`.

Dostęp do plików wewnątrz archiwum jest możliwy dzięki użyciu polecenia `tar` z odpowiednimi opcjami. Dearchiwizacja „tarballa” odbywa się z wykorzystaniem opcji `-z`, która mówi, że archiwum ma

być najpierw zdekompresowane gzipem. Oto najczęstszy sposób dekompresji tarballa:

```
% tar -xvzf filename.tar.gz
```

Co znaczą te wszystkie opcje? Ważna jest opcja `-x` oznaczająca rozpakowanie, ponieważ mówi tarowi, co dokładnie zrobić z plikiem wejściowym. W tym przypadku – rozdzielić z powrotem na pliki i katalogi, z których się oryginalnie składał. `-v` oznacza tryb szczegółowy (tzw. „gadatliwy” – ang. verbose).

Wyświetla listę wszystkich dearchiwizowanych plików. Można jej nie używać, po prostu informacje na ekranie będą nudniejsze. Można też użyć opcji `-vv`, która wyświetla jeszcze więcej szczegółów na temat każdego pliku. Opcja `-z` nakazuje najpierw przepuścić *nazwapliku.tar.gz* przez `gunzip`. Wreszcie opcja `-f` mówi tarowi, że następnym ciągiem w wierszu poleceń jest nazwa pliku, na którym ma operować.

To samo polecenie można wykonać na inne sposoby. W starszych systemach, w których nie ma w miarę nowego GNU tar, pisze się:

```
% gunzip nazwapliku.tar.gz | tar -xvf -
```

Takie polecenie zdekompresuje plik i wyśle wyjście do tar. Gzip wypisuje wynik działania na standardowe wyjście, jeśli mu się tak każe, tak też będzie w tym przypadku. Potokowanie przesyła wyjście do tar celem dearchiwizacji. Opcja „-” nakazuje operować na standardowym wejściu. Zdearchiwizuje to strumień danych otrzymany od `gzip` i zapisze go na dysku.

Można też zapisać polecenie, nie używając myślnika:

```
% tar xvzf nazwapliku.tar.gz
```

Jeśli archiwum jest skompresowane przez `bzip2`, można użyć tar w ten sam sposób, jak z archiwum `gzip`. Zamiast `-x` podaje się `-j`:

```
% tar -xvjf nazwapliku.tar.bz2
```

Należy pamiętać, że tar umieszcza zdearchiwizowane pliki w bieżącym katalogu. Jeśli np. masz archiwum w `/tmp`, a chcesz je zdekompresować do katalogu domowego, masz kilka możliwości. Po pierwsze, możesz przenieść archiwum do katalogu domowego i uruchomić tar. Po drugie, możesz podać ścieżkę do archiwum w wierszu poleceń. Po trzecie, za pomocą opcji `-C` możesz rozpakować archiwum do konkretnego 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 trzy sposoby działają tak samo. Archiwum zostaje rozpakowane do katalogu domowego, a

oryginalny niezdekompresowany plik pozostanie nietknięty.

Po co komuś rozpakowywanie archiwów, jeśli nie może ich tworzyć? To również potrafi `tar`. W większości przypadków wystarczy zastąpić „-x” przez „-c”.

```
% tar -cvzf nazwapliku.tar.gz .
```

Opcja `-c` mówi, aby `tar` utworzyło archiwum, `-z` powoduje, że zostanie ono skompresowane przez `gzip`, `nazwapliku.tar.gz` to nazwa tworzonego archiwum.

Opcja `-f` nie zawsze jest konieczna, lepiej jej jednak zawsze używać. Bez niej `tar` pisze na standardowe wyjście, czego z reguły używa się przy potokowaniu do innego programu:

```
% tar -cv nazwapliku.tar . | gpg --encrypt
```

Takie polecenie tworzy nieskompresowane archiwum bieżącego katalogu i przepuszcza wyjście przez `gpg`, które z kolei szyfruje archiwum i kompresuje je. Dzięki temu utworzony „tarball” jest praktycznie niemożliwy do odczytania dla nikogo, kto nie zna tajnego klucza.

## 15.4 zip

Do obsługi archiwów `zip`, często spotykanych w systemach Windows, istnieją w Linuksie dwa programy: program do kompresji o nazwie `zip(1)` oraz do dekompresji – `unzip(1)`.

```
% zip foo *
```

To utworzy plik `foo.zip`, który będzie zawierał wszystkie pliki z bieżącego katalogu. Program dodaje rozszerzenie `.zip` automatycznie, więc nie trzeba dodawać go do nazwy pliku. Można też przejść rekursywnie przez bieżący katalog, „zipując” wszystkie jego podkatalogi:

```
% zip -r foo *
```

Dekompresja jest bardzo łatwa:

```
% unzip foo.zip
```

Rozpakuje to wszystkie pliki i katalogi z archiwum `foo.zip`.

Programy do obsługi `zip` mają wiele zaawansowanych opcji, które służą do tworzenia samorozpakowujących się archiwów, pomijania plików, kontrolowania rozmiaru archiwum, wyświetlania, co się stanie po rozpakowaniu itp. Przeczytaj strony podręcznika `zip` i `unzip`, aby dowiedzieć się więcej.

## Rozdział 16. Edytor vi

`vi(1)` to standardowy uniksowy program do edycji tekstu. Opanowanie jego obsługi nie jest tak

niezbędne, jak niegdyś, ale cały czas się opłaca. Istnieje wiele odmian (tzw. „klonów”) tego programu: `vi`, `elvis`, `vile` i `vim`. Jedna z tych wersji jest dostępna w praktycznie każdym systemie uniksowym oraz w Linuksie. Każda z nich ma podstawowy zestaw funkcji i poleceń, dzięki czemu nauczenie się jednej ułatwia poznanie innych. Istnieje wiele innych edytorów w świecie Uniksa i Linuksa, dlatego wiele osób nie używa już `vi`. Cały czas jednak pozostaje on najbardziej uniwersalnym edytorem tekstu w systemach uniksowych i uniksopodobnych. Dzięki opanowaniu `vi` zawsze, siedząc przed maszyną uniksową, będziesz czuł się pewnie, mając do dyspozycji wspaniały edytor.

`Vi` ma wiele funkcji, m.in.: podświetlanie składni, formatowanie kodu, rozbudowany mechanizm wyszukiwania i zastępowania, makra itd. Te funkcje stanowią o szczególnej przydatności dla programistów, twórców stron WWW itp. Administratorzy systemów cenią sobie automatyzm i integrację z powłoką.

W Linuksie Slackware domyślnym edytorem `vi` jest `elvis`. Po zainstalowaniu odpowiednich pakietów są również dostępne inne wersje: `vim` i `gvim`. `Gvim` to graficzna wersja `vim`, która posiada paski narzędzi, odłączalne menu i okna dialogowe.

## 16.1 Uruchamianie `vi`

`vi` można uruchomić z wiersza poleceń na kilka sposobów. Oto najprostszy:

```
% vi
```

Rys. 16-1. Sesja `vi`

```
*/
void
TabReset(Tabs tabs)
{
    int i;

    for (i = 0; i < TAB_ARRAY_SIZE; ++i)
        tabs[i] = 0;

    for (i = 0; i < MAX_TABS; i += 8)
        TabSet(tabs, i);
}

tabs.c
if ((tmp = TypeCallocN(Char, length)) == 0)
    SysError(ERROR_SREALLOC);
*sbufaddr = tmp;

for (i = k = 0; i < nrow; i++) {
    k += BUF_HEAD;
    for (j = BUF_HEAD; j < old_max_ptrs; j++) {
        memcpy(tmp, base[k++], ncol);
        tmp += ncol;
    }
}

screen.c
```

To polecenie uruchamia `vi` z pustym buforem. W tym momencie widzisz prawie pusty ekran. Teraz `vi` jest w tzw. „trybie poleceń” i oczekuje, aż coś wykonasz. Więcej na temat trybów `vi` jest w sekcji 16.2. Aby zakończyć działanie `vi`, wpisz:

```
:q
```

Jeśli w pliku nie dokonałeś żadnych zmian, `vi` się zakończy. Jeśli zmodyfikowałeś plik, `vi` wyświetli ostrzeżenie o zmianach i powie, jak je zignorować. Aby zignorować zmiany, należy dodać wykrzyknik po „`q`”:

```
:q!
```

Wykrzyknik zwykle oznacza wymuszenie jakiejś akcji. Ta i inne kombinacje klawiszy są omówione później.

Można uruchomić `vi` z istniejącym plikiem jako argumentem. Np. plik `/etc/resolv.conf` otwiera się tak:

```
% vi /etc/resolv.conf
```

Można też uruchomić `vi` na konkretnym wierszu pliku. Jest to szczególnie przydatne dla programistów, jeśli komunikat o błędzie programu zawiera numer wiersza, na którym program się wysypał. Aby np. otworzyć `vi` na wierszu 47 pliku `/usr/src/linux/init/main.c`, wpisz:

```
% vi +47 /usr/src/linux/init/main.c
```

`Vi` otworzy podany plik i umieści kursor w danym wierszu. Jeśli podasz numer wiersza większy niż ilość wierszy pliku, kursor zostanie umieszczony w ostatnim wierszu. Jest to użyteczne dla programistów, ponieważ mogą przejść bezpośrednio do miejsca wystąpienia błędu i nie muszą przeszukiwać całego pliku.

## 16.2 Tryby

`Vi` ma różne tryby działania, które stosuje się do różnych zadań. Po uruchomieniu `vi` znajduje się w trybie poleceń. Możesz tu wydawać różne polecenia manipulujące tekstem, poruszać się po pliku, zapisywać, wychodzić z programu, zmieniać tryby. Edycji tekstu dokonujesz w trybie wstawiania. Między trybami przełączasz się za pomocą kombinacji klawiszy omówionych poniżej.

### 16.2.1 Tryb poleceń

Na początku `vi` znajduje się w trybie poleceń. Nie możesz teraz wpisywać ani modyfikować tekstu. Możesz jednak nim manipulować, wyszukiwać, wychodzić z programu, zapisywać plik, otwierać nowe pliki itd. Omówimy teraz tylko podstawy trybu wstawiania. Poszczególne polecenia są opisane w sekcji 16.7.

Najczęściej używanym poleceniem jest przejście do trybu wstawiania. Służy do tego klawisz `i`. Kursor zmienia kształt, a na dole ekranu wyświetla się `--INSERT--` (niektóre klony `vi` nie wyświetlają trybu). Teraz wszystkie klawisze, które wciśniesz, są wprowadzane do bieżącego bufora i wyświetlane na ekranie. Aby wrócić do trybu poleceń, wciśnij `Esc`.

W trybie poleceń możesz poruszać się po pliku. W niektórych systemach poruszasz się za pomocą strzałek, w innych używasz tradycyjnych klawiszy „**hjkl**”. Oto lista zastosowań tych klawiszy:

**h**     przejdź o jeden znak w lewo  
**j**     przejdź o jeden znak w dół  
**k**     przejdź o jeden znak w górę  
**l**     przejdź o jeden znak w prawo

Wciśnięcie klawisza spowoduje odpowiednie przejście kursora. Później opiszemy, jak łączyć te klawisze z numerami, aby poruszać się efektywniej.

Wiele poleceń wydawanych w tym trybie rozpoczyna się od dwukropka. Na przykład, wcześniej omówione wychodzenie z edytora - **:q**. Dwukropek oznacza po prostu polecenie, a **q** oznacza wyjście. Inne polecenia to opcjonalny numer, a po nim litera. Nie wstawia się przed nimi dwukropka, służą głównie do manipulacji tekstem.

Na przykład, do usunięcia jednego wiersza z pliku służy **dd**. Usunie to wiersz, w którym znajduje się kursor. Polecenie **4dd** usuwa wiersz z kursorem i trzy kolejne. Ogólnie rzecz biorąc, numer oznacza, ile razy **v i** ma wykonać daną akcję.

Można łączyć numer z klawiszami przesuwania, aby poruszać się o więcej niż jeden znak. Na przykład, **10k** przesuwa kursor o 10 wierszy w górę.

Trybu poleceń używa się też do wycinania, wklejania i wstawiania tekstu, wczytywania innych plików do bieżącego bufora. Tekst kopiuje się klawiszem **y** (**y** oznacza *yank*, dosłownie „wyszarpować”). Bieżący wiersz kopiuje się przez **yy**, można to poprzedzić numerem określającym liczbę wierszy do skopiowania. Po przejściu do miejsca, w które ma być skopiowany tekst, wciśnij **p**. Tekst zostanie wklejony pod bieżącym wierszem.

Tekst można wyciąć poleceniem **dd**, a wycięty tekst wstawić **p**. Wczytywanie tekstu z innego pliku jest proste. Wpisz **:r**, a po spacji nazwę pliku, który chcesz wstawić. Zawartość tego pliku zostanie wstawiona do bieżącego bufora pod wierszem, w którym znajduje się kursor. Bardziej rozbudowane klony **v i** mają funkcję uzupełniania tabulatorem, podobnie jak powłoka.

Ostatnie opisane zastosowanie trybu poleceń to wyszukiwanie. Można albo po prostu wyszukiwać tekst, albo stosować zaawansowane mechanizmy wyszukiwania i zastępowania, wykorzystujące wyrażenia regularne. Wyrażenia regularne wychodzą poza zakres tej książki, więc skupimy się na prostym wyszukiwaniu.

Wyszukiwanie uruchamia się, wciskając **/** i wpisując tekst do wyszukania. **v i** przeszukuje plik od kursora w dół, zatrzymując się na pierwszym znalezionym tekście. Niedokładne dopasowania również spowodują zatrzymanie wyszukiwania. Np. wyszukiwanie „*piw*” będzie zatrzymywało się na „*piwnica*”, „*piwo*” itd., ponieważ wszystkie te słowa pasują do „*piw*” na początku wyrazu.

Po znalezieniu pierwszego pasującego tekstu możesz kontynuować wyszukiwanie, wciskając **/** i **enter**. Możesz też szukać w górę pliku, używając **?** zamiast **/**. Aby np. przeszukać plik od kursora w górę, szukając „*piw*”, wpisz **?piw**.

## 16.2.2 Tryb wstawiania

Wpisywania i zastępowania tekstu dokonuje się w trybie wstawiania. Jak napisaliśmy wcześniej, przechodzi się do niego z trybu poleceń klawiszem **i**. Wówczas wpisywany tekst jest umieszczany w bieżącym buforze. Po wciśnięciu **Esc** `vi` wraca do trybu poleceń.

Tekst można zastępować na kilka sposobów. Wciśnięcie **r** w trybie poleceń umożliwia zastąpienie znaku pod kursorem. Wystarczy wpisać inny znak, a zastąpi on ten znajdujący się pod kursorem. Zaraz potem `vi` powróci do trybu poleceń. Po wciśnięciu **R** możesz zmienić więcej znaków. Aby wyjść z trybu zastępowania, wciśnij **Esc**, dzięki czemu wrócisz do trybu poleceń.

Jest jeszcze jeden sposób na przechodzenie między trybem wstawiania i zastępowania. Będąc w trybie poleceń, wciśnij klawisz **Insert**. `vi` przejdzie w tryb wstawiania. Gdy jesteś w trybie wstawiania, klawisz **Insert** służy jako przełącznik między wstawianiem a zastępowaniem. Wciśnięcie go włącza tryb zastępowania, wciśnięcie ponowne – tryb wstawiania.

## 16.3 Otwieranie plików

Pliki w `vi` można otworzyć, podając je jako argument w wierszu poleceń powłoki albo bezpośrednio w trybie poleceń edytora. Aby np. otworzyć `/etc/lilo.conf`:

```
:e /etc/lilo.conf
```

Jeśli dokonałeś zmian w bieżącym buforze i nie zapisałeś ich, `vi` będzie marudzić. Możesz otworzyć plik bez zapisywania bieżącego bufora, wpisując **:e!**, spację i nazwę pliku. Generalnie ostrzeżenia `vi` można zignorować, pisząc wykrzyknik po poleceniu.

Jeśli chcesz wczytać ponownie otwarty plik, wpisz **e!**. Jest to szczególnie przydatne, jeśli coś namieszałeś i chcesz otworzyć plik na nowo.

Niektóre klony `vi` (np. `vim`) umożliwiają wiele otwartych buforów naraz. Aby np. otworzyć plik `09-vi.sgml` w moim katalogu domowym, podczas gdy inny plik jest otwarty, wpisałbym:

```
:split ~/09-vi.sgml
```

Nowy plik jest wyświetlony na górnej połowie ekranu, stary na dolnej. Do pracy z podzielonym ekranem służy wiele poleceń, wiele z nich przypomina te używane w edytorze `Emacs`. Ich opis znajdziesz na stronie podręcznika `vi`. Wiele klonów nie obsługuje dzielenia ekranu, więc ta funkcja może być niedostępna.

## 16.4 Zapisywanie plików

Pliki w `vi` można zapisać na kilka sposobów. Aby zapisać bieżący bufor do pliku `costam`, wpisz:

```
:w costam
```

Po pierwszym zapisaniu pliku ponownych zapisów dokonujesz poleceniem **:w**. Wszelkie zmiany zostaną

zachowane w pliku. Po zapisaniu wracasz do trybu poleceń. Jeśli chcesz zapisać plik i wyjść z `vi` (bardzo często tak jest), wpisz `:wq`. To powoduje, że `vi` zapisuje plik, kończy działanie i wraca do powłoki.

Czasami próbujesz zapisać plik, który ma prawa tylko do odczytu. Aby zapisać do takiego pliku, dodaj wykrzyknik do polecenia zapisu:

```
:w!
```

Bywają też przypadki, kiedy zapis do pliku jest niemożliwy (np. edytujesz plik, którego właścicielem jest inny użytkownik). Wówczas `vi` powie Ci, że nie może zapisać pliku. Jeśli naprawdę chcesz go zmodyfikować, musisz robić to jako `root`, albo (najlepiej) właściciel tego pliku.

## 16.5 Wychodzenie z `vi`

Najczęściej wychodzi się z `vi` poleceniem `:wq`, które zapisuje bieżący bufor na dysku przed zamknięciem programu. Można też wyjść bez zapisywania, wpisując `:q`, lub (powszechnie stosowane) `:q!`. Tego ostatniego używa się, jeśli zmodyfikowałeś plik, ale nie chcesz zachować zmian.

Czasami komputer z uruchomionym `vi` lub sam program ulegają awarii. Jednak zarówno `elvis`, jak i `vim` mają funkcje minimalizujące uszkodzenie otwartych buforów. Oba edytory zapisują otwarte bufory w tymczasowych plikach. Taki plik zwykle ma nazwę identyczną z oryginałem, ale ma kropkę na początku. Dzięki temu plik jest ukryty.

Plik tymczasowy jest usuwany, gdy edytor zostaje normalnie zamknięty. To znaczy, że tymczasowa kopia będzie istnieć w razie awarii. Gdy będziesz chciał wrócić do edytowania pliku, wyświetli się pytanie o to, jakie działanie podjąć. W większości przypadków jest możliwe odzyskanie większości niezapisanej pracy. `Elvis` ponadto wyśle Ci email (o dziwo, z Graceland :) z wiadomością, że istnieje kopia zapasowa.

## 16.6 Konfiguracja `vi`

Każdy klon `vi` można skonfigurować na kilka sposobów.

Aby dostosować `vi` do własnych potrzeb, można wpisać odpowiednie polecenia, gdy edytor jest uruchomiony w trybie poleceń. W zależności od wersji, możesz uruchomić funkcje pomagające w programowaniu (podświetlanie składni, automatyczne wcięcia itp.), stworzyć makra automatyzujące pewne czynności, włączyć zastępowanie danych ciągów innym tekstem i o wiele więcej.

Większość tych poleceń można wpisać w pliku konfiguracyjnym w katalogu domowym. `Elvis` wymaga pliku `.exrc`, a `vim` - `.vimrc`. Prawie wszystkie polecenia wpisywane w trybie poleceń można umieścić w pliku konfiguracyjnym. Są to informacje o konfiguracji, podstawienia tekstu, makra itp.

Wszystkie te opcje i różnice między poszczególnymi wersjami programu to dość złożone zagadnienie. Więcej informacji znajdziesz na stronie podręcznika i stronie WWW wybranego edytora. Niektóre edytory (np. `vim`) mają wbudowaną pomoc, którą możesz uruchomić poleceniem `:help` lub podobnym. Warto również przeczytać książkę wydawnictwa O'Reilly *Learning the vi Editor* autorstwa Lindy Lamb i Arnolda Robbinsa (od tłum.: wydanie polskie – *Edytor vi*, wydawnictwo Helion, 2001).



Wiele programów linuksowych otwiera pliki tekstowe właśnie w `vi`. Na przykład, edytowanie plików `crontab` domyślnie uruchamia je w `vi`. Jeśli nie podoba Ci się `vi` i chcesz używać innego edytora jako domyślnego, musisz ustawić zmienną środowiskową `VISUAL` na właściwy edytor. Informacje o ustawianiu zmiennych środowiskowych znajdują się w sekcji Zmienne środowiskowe w rozdziale 8. Aby upewnić się, że wybrany edytor będzie domyślny przy każdym logowaniu, dodaj ustawienie `VISUAL` w pliku `.bash_profile` lub `.bashrc`.

## 16.7 Klawisze `vi`

Ta sekcja to krótka lista ważniejszych poleceń `vi`. Niektóre zostały opisane wcześniej, większość jest nowa.

**Tabela 16-1. Poruszanie się po pliku**

Działanie	klawisz
lewo, dół, prawa, góra	<b>h, j, k, l</b>
na koniec wiersza	<b>\$</b>
na początek wiersza	<b>^</b>
na koniec pliku	<b>G</b>
na początek pliku	<b>:1</b>
do wiersza 47	<b>:47</b>

**Tabela 16-2. Edycja**

Działanie	klawisz
usunięcie wiersza	<b>dd</b>
usunięcie 5 wierszy	<b>5dd</b>
zastąpienie znaku	<b>r</b>
usunięcie znaku	<b>x</b>
usunięcie 10 znaków	<b>10x</b>
cofnięcie ostatniej operacji	<b>u</b>
połączenie bieżącego i następnego wiersza	<b>J</b>
zastąpienie starego tekstu nowym w całym pliku	<b>%s'stary'nowy'g</b>

**Tabela 16-3 Wyszukiwanie**

Działanie	klawisz
wyszukiwanie ciągu „asdf”	<b>/asdf</b>
wyszukiwanie ciągu „asdf” wstecz”	<b>?asdf</b>
powtórzenie ostatniego wyszukiwania do przodu	<b>/</b>
powtórzenie ostatniego wyszukiwania do tyłu	<b>?</b>
powtórzenie ostatniego wyszukiwania w tym samym kierunku	<b>n</b>

powtórzenie ostatniego wyszukiwania w przeciwnym kierunku	<b>N</b>
-----------------------------------------------------------	----------

**Tabela 16-4. Zapisywanie i wychodzenie**

Działanie	klawisz
wyjscie z programu	<b>:q</b>
wyjscie bez zapisywania	<b>:q!</b>
zapisanie i wyjście	<b>:wq</b>
zapisanie bez wychodzenia	<b>:w</b>
ponowne wczytanie otwartego pliku	<b>:e!</b>
zapisanie bufora do pliku <code>asdf</code>	<b>:w asdf</b>
otwarcie pliku <code>hejaz</code>	<b>:e hejaz</b>
wczytanie pliku <code>asdf</code> do bufora	<b>:r asdf</b>
wczytanie wyjścia <code>ls</code> do bufora	<b>:r !ls</b>

## Rozdział 17. Emacs

`vi` (oraz jego klony) to bez wątpienia edytor zajmujący pierwsze miejsce w systemach uniksowych. Emacs zajmuje drugie miejsce. W przeciwieństwie do `vi`, nie używa się w nim „trybów”. Do wydawania poleceń używa się kombinacji klawiszy z **Control** i **Alt**, podobnie jak w procesorach tekstów i innych aplikacjach. Należy jednak podkreślić, że kombinacje w `emacsie` różnią się od standardowych. Większość aplikacji używa np. **Ctrl+C/X/V** do kopiowania/wycinania/wklejania. Emacs używa innych kombinacji i tak naprawdę innego mechanizmu do realizacji tych funkcji.

`vi` jest (doskonałym) edytorem, ale tylko edytorem. Emacs natomiast nadaje się do praktycznie wszystkiego. Większość `Emacsa` napisano w języku Lisp. To język dający bardzo wiele możliwości. Jego szczególną cechą jest fakt, że programy napisane w tym języku stają się automatycznie swoimi własnymi kompilatorami. Dzięki temu użytkownicy mogą dowolnie rozszerzać funkcjonalność `Emacsa`, a wręcz pisać nowe programy „w `Emacsie`”.

Dzięki temu Emacs nie jest tylko edytorem. Istnieje wiele dodatkowych rozszerzeń dla `Emacsa` (wiele dołączonych do źródeł programu), które dodają wszelkiego rodzaju funkcje. Większość z nich dotyczy edycji tekstu, co oczywiście jest podstawowym przeznaczeniem `Emacsa`, ale nie ograniczają się do tego. Istnieją np. arkusze kalkulacyjne, bazy danych, gry, klienty poczty i grup dyskusyjnych (najważniejszy z nich to `Gnus`) itp.

Są dostępne dwie wersje `Emacsa`: GNU Emacs (dołączony do Slackware) oraz XEmacs. Ten drugi to *NIE* jest wersja `Emacsa` działająca w X. Obie wersje działają zarówno w konsoli, jak i w X. Projekt XEmacs rozpoczęto w celu uporządkowania kodu `Emacsa`. Obecnie trwają aktywne prace nad rozwojem obu odmian, a zespoły deweloperów w znacznym stopniu współdziałają ze sobą. Na poziomie tego rozdziału nie jest istotne, której wersji używasz, ponieważ z punktu widzenia zwykłego użytkownika różnice między nimi są nieistotne.

## 17.1 Uruchamianie Emacsa

Emacs uruchamia się z wiersza poleceń, wpisując po prostu „emacs”. Jeśli jesteś w X, Emacs uruchomi się (zwykle) we własnym oknie X, z paskiem menu na górze, udostępniającym najważniejsze funkcje. Po uruchomieniu Emacs wyświetla wiadomość powitalną, a po paru sekundach przechodzi do bufora „scratch” (więcej w sekcji 17.2).

```
File Edit Options Buffers Tools Help
Welcome to GNU Emacs, one component of the GNU/Linux operating system.

Get help          C-h (Hold down CTRL and press h)
Emacs manual      C-h r
Emacs tutorial    C-h t          Undo changes      C-x u
Buy manuals       C-h C-m        Exit Emacs        C-x C-c
Browse manuals    C-h i
Activate menubar  F10 or ESC ` or M-`
(`C-' means use the CTRL key. `M-' means use the Meta (or Alt) key.
If you have no Meta key, you may instead type ESC followed by the character.)

GNU Emacs 23.0.0.2 (i686-pc-linux-gnu, X toolkit, Xaw3d scroll bars)
of 2005-03-25 on slackware
Copyright (C) 2004 Free Software Foundation, Inc.

GNU Emacs comes with ABSOLUTELY NO WARRANTY; type C-h C-w for full details.
Emacs is Free Software--Free as in Freedom--so you can redistribute copies
of Emacs and modify it; type C-h C-c to see the conditions.
Type C-h C-d for information on getting the latest version.

If an Emacs session crashed recently, type M-x recover-session RET
to recover the files you were editing.

----- GNU Emacs -----
For information about the GNU Project and its goals, type C-h C-p.
```

Można też uruchomić Emacs z zadany plikiem:

```
% emacs /etc/resolv.conf
```

Emacs po uruchomieniu załaduje wybrany plik, pomijając wiadomość powitalną.

### 17.1.1 Klavisze poleceń

Jak wspomnieliśmy powyżej, Emacs używa kombinacji z klawiszami **Control** i **Alt** do wydawania poleceń. Konwencja ich zapisywania to **C-litera** i **M-litera**. Tak więc **C-x** oznacza **Control+x**, a **M-x** oznacza **Alt+x**. (Litery **M** używa się zamiast **A**, ponieważ kiedyś ten klawisz nazywał się **Meta**, a nie **Alt**. Klawisz **Meta** obecnie nie spotyka się już na klawiaturach, a w Emacsie jego funkcje przejął **Alt**.)

Wiele poleceń Emacs składa się z kombinacji i sekwencji klawiszy. Na przykład, **C-x C-c** (czyli **Control+x**, a następnie **Control+c**) kończy pracę z Emacsem. **C-x C-s** zapisuje bieżący plik. Pamiętaj, że **C-x C-b** to *NIE* to samo, co **C-x b**. To pierwsze oznacza kolejno kombinacje **Control+x** i **Control+b**, to drugie oznacza **Control+x** i samo **b**.

## 17.2 emacs – bufory

Jednym z podstawowych pojęć w Emacsie jest „bufor”. Każdy otwarty plik jest ładowany do odrębnego bufora. Emacs ma też inne, specjalne bufory, które nie zawierają pliku, ale są używane do innych rzeczy. Nazwy tych buforów specjalnych zwykle zaczynają się i kończą gwiazdką. Na przykład, bufor wyświetlany podczas uruchomienia Emacs’a nazywa się „\*scratch\*” (bufor szkicu). Możesz w tym buforze normalnie wpisywać tekst, ale nie zostanie on zapisany przy wychodzeniu z edytora.

Kolejnym ważnym buforem specjalnym jest tzw. minibufor. Zawiera on tylko jeden wiersz i jest zawsze widoczny na dole ekranu. To ostatni wiersz w oknie Emacs’a, pod paskiem statusu bieżącego bufora. Emacs wyświetla w minibuforze informacje dla użytkownika, wpisuje się tam też informacje wymagane przez niektóre polecenia. Na przykład, kiedy chcesz otworzyć plik, Emacs poprosi o wpisanie jego nazwy w minibuforze.

Między buforami przełącza się kombinacją **C-x b**. Wyświetla to prośbę o wpisanie nazwy bufora (nazwa bufora to z reguły nazwa edytowanego pliku) i proponuje wybór domyślny, czyli bufor, w którym byłeś przed przełączeniem się lub utworzeniem nowego. Wciśnij *Enter*, aby przejść do tego domyślnego bufora.

Jeśli chcesz przełączyć się do bufora innego niż domyślny, wpisz jego nazwę. Możesz użyć funkcji uzupełniania tabulatorem: wpisz pierwsze kilka znaków nazwy bufora i wciśnij **Tab**. Emacs uzupełni nazwę bufora. Uzupełnianie tabulatorem działa w Emacsie wszędzie tam, gdzie jego użycie ma sens.

Listę otwartych buforów wyświetla **C-x C-b**. Rozdziela to ekran na dwie części. W górnej jest zawartość bieżącego bufora, w dolnej otwiera się nowy bufor o nazwie „\*Buffer List\*”. Zawiera on listę wszystkich buforów, ich rozmiarów, trybów i plików, które są „odwiedzane” (w terminologii Emacs’a) przez bufory. Podział ekranu usuwa się przez **C-x l**.

Uwaga: W X lista buforów jest również dostępna pod pozycją Bufor na pasku menu.

## 17.3 Emacs – tryby

Każdy bufor w Emacsie ma tzw. tryb (ang. mode). Tryby w Emacsie różnią się od tych w vi. Tryb oznacza, w jakim rodzaju bufora jesteś. Na przykład, dla zwykłych plików tekstowych jest tryb tekstowy (text-mode), dla edycji programów w C jest tryb C (C-mode), dla skryptów powłoki sh-mode, dla plików **LaTeX** latex-mode, dla edycji wiadomości poczty elektronicznej i grup dyskusyjnych – mail-mode itd. W każdym trybie są dostępne szczególne funkcje i możliwości dostosowania, odpowiednie dla rodzaju edytowanego pliku. Można nawet w trybach redefiniować klawisze i kombinacje. Na przykład, w text-mode klawisz **Tab** przechodzi po prostu do następnego znaku tabulacji. Natomiast w wielu trybach programowania **Tab** tworzy wcięcie bieżącego wiersza na głębokość danego bloku, w którym znajduje się wiersz.

Te tryby, tj. text-mode, sh-mode, latex-mode, mail-mode itp. to tryby główne (ang. major modes). Każdy bufor ma jeden tryb główny. Bufor może też mieć jeden lub więcej trybów dodatkowych (minor modes). Tryby dodatkowe dodają funkcje użyteczne przy wykonywaniu niektórych zadań. Na przykład, po wciśnięciu **Insert** przechodzisz w tryb nadpisywania, którego potrzebujesz. Istnieje też tryb autowypełniania (auto-fill-mode), przydatny w połączeniu z text-mode i latex-mode. Powoduje on, że każdy wiersz jest automatycznie zawijany po przekroczeniu określonej liczby znaków. Bez tego trybu

trzeba wciskać **M-q**, aby wypełnić akapit. (Możesz też tego użyć, aby ponownie sformatować akapit, w którym wpisałeś jakiś tekst, powodując zmianę układu.)

### 17.3.1 Otwieranie plików

Aby otworzyć plik w Emacsie, wpisz:

```
C-x C-f
```

Emacs zapyta o nazwę pliku, wypełniając za Ciebie część ścieżki domyślnej (zwykle ~/). Po wpisaniu nazwy (możesz użyć uzupełniania tabulatorem) i wciśnięciu **Enter** Emacs otworzy plik w nowym buforze i wyświetli jego zawartość na ekranie.

Uwaga: Emacs automatycznie utworzy nowy bufor, nie zastąpi zawartości bieżącego.

Aby w Emacsie utworzyć nowy plik, nie możesz po prostu zacząć pisać. Najpierw musisz utworzyć nowy bufor i podać nazwę pliku. Wystarczy wpisać **C-x C-f**, tak jak przy otwieraniu istniejącego pliku. Emacs zauważy, że plik nie istnieje, utworzy nowy bufor, a w minibuforze wyświetli komunikat „New file” (Nowy plik).

Jeśli wpiszesz **C-x C-f** i podasz nazwę katalogu zamiast pliku, Emacs utworzy nowy bufor z listą wszystkich plików w tym katalogu. Możesz przejść kursorem do poszukiwanego pliku i wcisnąć **Enter**, aby go otworzyć. (Możesz tu wykonać wiele innych działań, np. usunąć plik, przenieść go lub zmienić jego nazwę. Emacs jest teraz w trybie katalogów (dired-mode), czyli działa jak menedżer plików.)

Jeśli po wpisaniu **C-x C-f** zmieniłeś zdanie, możesz anulować swoje działanie poleceniem **C-g**. **C-g** działa prawie zawsze, kiedy chcesz anulować jakieś działanie lub polecenie, które rozpocząłeś, ale się rozmyśliłeś.

### 17.4 Podstawy edycji

Możesz poruszać się po otwartym pliku za pomocą kursora. **Klawisze kursora** oraz **PgUp** (strona w górę) i **PgDn** (strona w dół) tak właśnie działają. **Home** i **End** przesuwały kursor odpowiednio na początek i koniec wiersza (w starszych wersjach przechodziły na początek lub koniec bufora). Są też kombinacje z klawiszami **Control** i **Meta (Alt)**, które przesuwały kursor. Dzięki temu, że nie musisz odrywać palców, aby przejść do innej części klawiatury, o wiele szybciej się z nich korzysta, gdy się do nich przywyknie. W tabeli 17-1 przedstawiamy najważniejsze kombinacje.

Tabela 17-1. Najważniejsze polecenia edycji w Emacsie

Polecenie	Wynik
<b>C-b</b>	przejście o jeden znak wstecz
<b>C-f</b>	przejście o jeden znak naprzód
<b>C-n</b>	przejście o jeden wiersz w dół
<b>C-p</b>	przejście o jeden wiersz w górę
<b>C-a</b>	przejście na początek wiersza
<b>C-e</b>	przejście na koniec wiersza

<b>M-b</b>	przejdź o jedno słowo wstecz
<b>M-f</b>	przejdź o jedno słowo naprzód
<b>M-}</b>	przejdź o jeden akapit naprzód
<b>M-{'</b>	przejdź o jeden akapit wstecz
<b>M-a</b>	przejdź o jedno zdanie wstecz
<b>M-e</b>	przejdź o jedno zdanie naprzód
<b>C-d</b>	usunięcie znaku pod kursorem
<b>M-d</b>	usunięcie znaków od kursora do końca bieżącego słowa
<b>C-v</b>	przesunięcie o jeden ekran w dół (tak jak <b>PgDn</b> )
<b>M-v</b>	przesunięcie o jeden ekran w górę (tak jak <b>PgUp</b> )
<b>M-&lt;</b>	przejdź na początek bufora
<b>M-&gt;</b>	przejdź na koniec bufora
<b>C-_</b>	cofnięcie ostatniej zmiany (można je powtarzać); pamiętaj, że musisz tak naprawdę wcisnąć <b>Shift + Control + myślnik</b>
<b>C-k</b>	usunięcie znaków od kursora do końca wiersza
<b>C-s</b>	wyszukiwanie naprzód
<b>C-r</b>	wyszukiwanie wstecz

Wiele kombinacji z **Meta** działa tak, jak kombinacje z **Control**, ale działają na większych fragmentach: **C-f** przesuwa kursor o jeden znak naprzód, **M-f** przesuwa o jedno słowo naprzód itd.

Pamiętaj też, że **M-<** i **M->** wymagają wpisania odpowiednio **Shift + Alt + przecinka** i **Shift+Alt+kropki**, ponieważ znaki **<** i **>** są dostępne po wcisnięciu **Shift+przecinek** i **Shift+kropka** (o ile nie masz układu klawiatury innego niż standardowy US).

Zwróć także uwagę, że **C-k** usuwa (zabija, jak powszechnie się mówi) tekst od kursora do końca wiersza, ale nie usuwa samego wiersza (czyli nie usuwa znaku nowego wiersza). Cały wiersz jest usuwany tylko wtedy, kiedy po kursorze nie było tekstu. Innymi słowy, aby usunąć cały wiersz, trzeba umieścić kursor na jego początku i wcisnąć **C-k** dwukrotnie: raz, aby usunąć tekst, drugi raz, aby usunąć sam wiersz.

## 17.5 Zapisywanie plików

Aby zapisać plik, wpisz:

```
C-x C-s
```

Emacs nie zapyta o nazwę pliku, po prostu zapisze bufor do tego pliku, który załadował. Jeśli chcesz zapisać tekst w innym pliku, wpisz:

```
C-x C-w
```

Kiedy zapisujesz plik po raz pierwszy w sesji, Emacs zapisuje starą wersję w pliku kopii zapasowej, o takiej samej nazwie jak oryginał, z dodaną tyldą. Jeśli np. edytujesz plik „piwo.txt”, Emacs utworzy kopię zapasową o nazwie „piwo.txt~”.

Ten plik jest kopią właśnie otwartego pliku. Podczas pracy Emacs tworzy ponadto regularnie kopie w pliku o nazwie z krzyżykami: #piwo.txt#. Zostanie on usunięty po zapisaniu pliku przez C-x C-s.

Po zakończeniu edycji pliku możesz zabić jego bufor za pomocą:

```
C-x k
```

Emacs zapyta, który bufor chcesz zabić, proponując domyślnie bieżący – możesz go wybrać, wciskając **Enter**. Jeśli nie zapisałeś jeszcze pliku, wyświetli się pytanie, czy na pewno zabić bufor.

### 17.5.1 Wychodzenie z Emacsa

Po zakończeniu pracy z Emacsem wpisz:

```
C-x C-c
```

Kończy to pracę edytora. Jeśli istniały jakieś niezapisane pliki, Emacs będzie się po kolei pytał, czy je zapisać. Jeśli odpowiesz „nie” na któryś z nich, Emacs zada ostateczne pytanie o potwierdzenie i zakończy działanie.

## Rozdział 18. Zarządzanie pakietami w Slackware

Pakiet z oprogramowaniem to grupa połączonych programów gotowych do zainstalowania. Kiedy pobierasz kod źródłowy, musisz go skonfigurować, skompilować i zainstalować ręcznie. W pakiecie już za Ciebie tę pracę wykonano. Wystarczy go tylko zainstalować. Inną zaletą pakietów jest ich łatwość usuwania lub aktualizacji. Slackware zawiera programy do obsługi wszystkich zadań związanych z zarządzaniem pakietami. Instalacja, usuwanie, aktualizacja, tworzenie i sprawdzanie zawartości pakietów jest proste.

Istnieje mit, że od kiedy Red Hat wprowadził swój program do zarządzania pakietami – RedHat Package Manager, w Slackware nie ma podobnego programu. Oczywiście to bzdura. Linux Slackware miał program do zarządzania pakietami jeszcze przed pojawieniem się Red Hata. Ten program, `pkgtool`, nie jest, co prawda, tak rozbudowany i powszechny jak `rpm` (albo `deb`), ale wraz z towarzyszącymi programami również doskonale nadaje się do instalacji pakietów. Tak naprawdę, problemem `pkgtool` jest fakt, że nie dokonuje on żadnego sprawdzania zależności.

Wiele osób należących do społeczności linuksowej uważa, że menedżer pakietów musi z założenia sprawdzać zależności. Nie jest to oczywiście prawda, bo Slackware tego nie robi. Nie to, że pakiety Slackware nie mają zależności. Po prostu `pkgtool` ich nie sprawdza. Sprawdzanie zależności pozostawiono administratorowi systemu, zgodnie z duchem Slackware.

## 18.1 Omówienie formatu pakietów

Zanim nauczysz się używać narzędzi, powinieneś poznać format pakietów Slackware. Taki pakiet to po prostu archiwum `tar`, skompresowane przez `gzip`. Pakiety tworzy się tak, aby rozpakowywać je w katalogu głównym.

Oto przykładowy, fikcyjny program i zawartość jego pakietu:

```
./
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
```

Aby zainstalować pakiet, system obsługi pakietów rozpakuje go w katalogu głównym. Do bazy danych pakietów zostaną dodane opisy zawartości pakietu, aby można go było później zaktualizować lub usunąć.

Zwróć uwagę na podkatalog `install/`. Jest to specjalny katalog, w którym znajduje się skrypt poinstalacyjny o nazwie `doinst.sh`. Jeśli system obsługi pakietów go znajdzie, wykona go po zainstalowaniu pakietu.

W pakiecie mogą znajdować się też inne skrypty. Jest to omówione w sekcji 18.3.2.

## 18.2 Programy do zarządzania pakietami

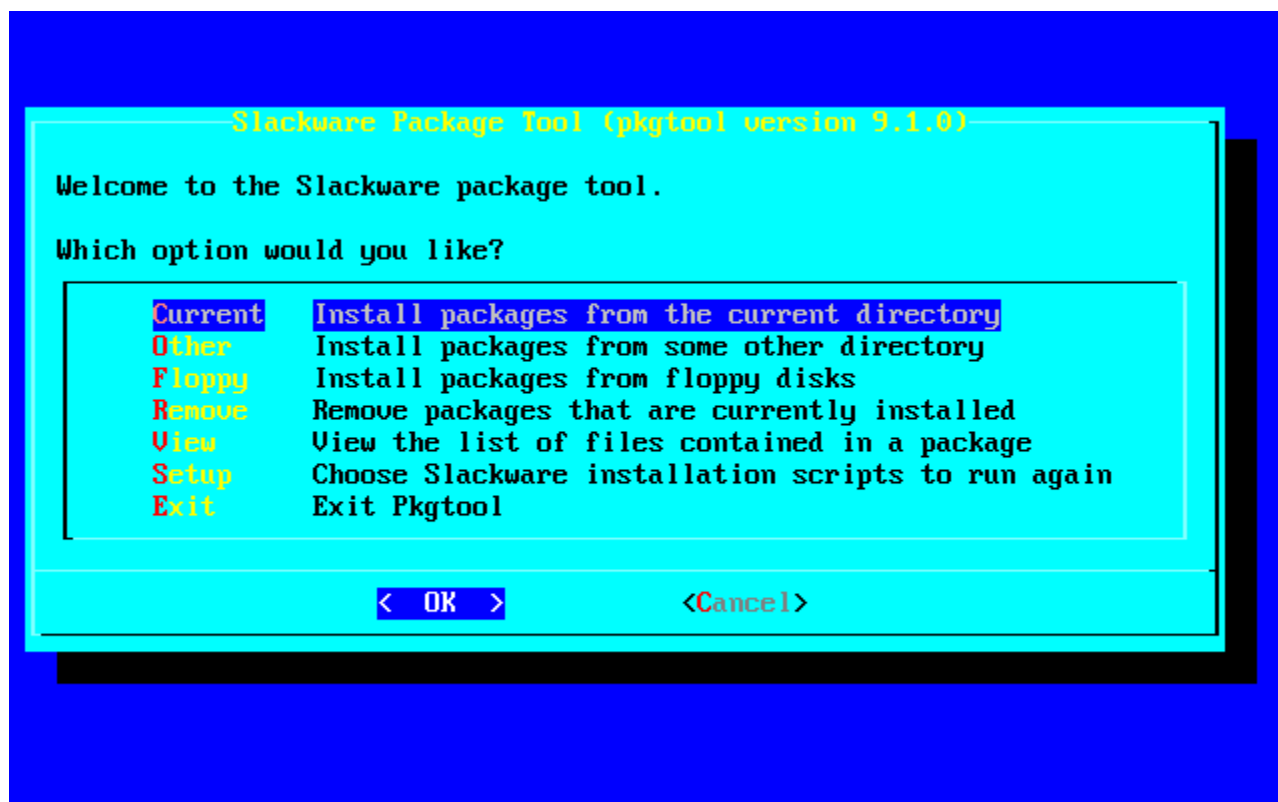
Do zarządzania pakietami są przeznaczone cztery zasadnicze narzędzia, służące do instalowania, usuwania i aktualizacji.

### 18.2.1 pkgtool

`pkgtool(8)` to wykorzystujący menu program, który umożliwia instalowanie i usuwanie pakietów. Oto jego menu główne:

**Rys. 18-1 Menu główne `pkgtool`**





Witamy w narzędziu do zarządzania pakietami Slackware.

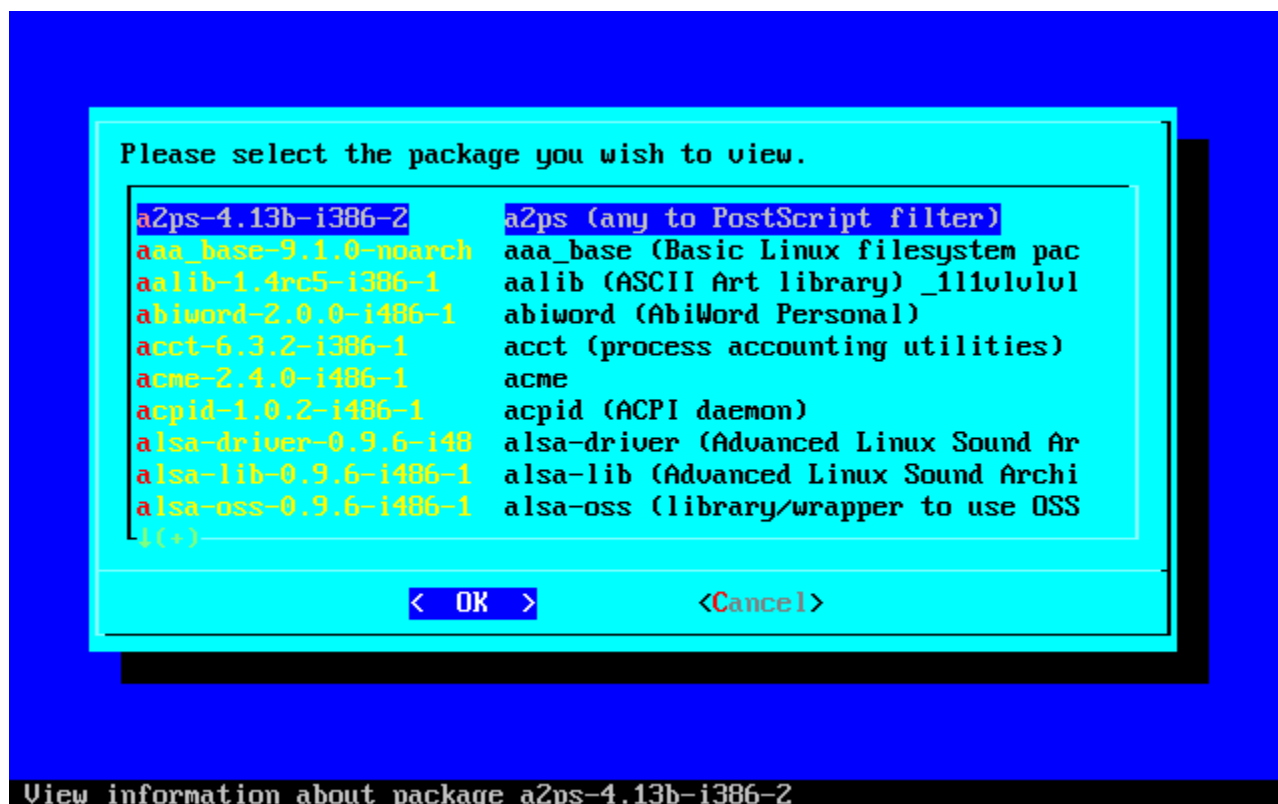
Którą opcję chcesz wybrać?

Current (bieżący)	Zainstaluj pakiety z bieżącego katalogu
Other (inny)	Zainstaluj pakiety z innego katalogu
Floppy (dyskietka)	Zainstaluj pakiety z dyskietek
Remove (usuń)	Usuń zainstalowane obecnie pakiety
View (Przeglądaj)	Przeglądaj listę plików zawartych w pakiecie
Setup (Konfiguruj)	Wybierz skrypty instalacyjne Slackware, które mają być uruchomione ponownie
Exit (Wyjdź)	Wyjdź z pkgtool
	<OK>
	<Anuluj>

Można wybrać instalację z bieżącego katalogu, innego katalogu lub dyskietek. Wystarczy wybrać metodę, a `pkgtool` przejrzy lokalizację w poszukiwaniu pakietów.

Możesz też przejrzeć listę zainstalowanych pakietów:

**Rys. 18-2. Tryb przeglądania listy pkgtool**



Aby usunąć pakiety, wybierz opcję Remove. Wyświetli się lista zainstalowanych pakietów z polami wyboru. Zaznacza te, które chcesz usunąć i wybierz OK, a `pkgtool` je usunie.

Niektórzy wolą `pkgtool` od programów uruchamianych z wiersza poleceń. Takie programy, z drugiej strony, mają o wiele więcej opcji. Ponadto aktualizować pakiety można tylko za pomocą programów z wiersza poleceń.

## 18.2.2 `installpkg`

Nowe pakiety instaluje się poleceniem `installpkg(8)`. Oto składnia:

```
# installpkg option package_name
```

`Installpkg` ma trzy opcje, można użyć tylko jednej naraz.

**Tabela 18-1. Opcje `installpkg`**

Opcja	Efekt
-m	Przeprowadza operację <code>makepkg</code> na bieżącym katalogu
-warn	Pokazuje, co stałoby się po zainstalowaniu podanego pakietu. Jest to przydatne szczególnie w systemach produkcyjnych, abyś przed instalacją zobaczył, co dokładnie by się stało.

-r

Rekursywnie instaluje wszystkie pakiety w bieżącym katalogu i jego podkatalogach. Można używać symboli wieloznacznych jako maski wyszukiwania przy rekursywnej instalacji.

Jeśli przed `installpkg` ustawisz zmienną środowiskową `ROOT`, zostanie ona użyta jako katalog główny. Przydaje się to do konfigurowania nowych napędów tak, aby był na nich katalog główny. Zwykle montuje się je w `/mnt` albo innym katalogu niż `/`.

Wpis dotyczący zainstalowanego pakietu znajduje się w bazie danych w `/var/log/packages`. Taki wpis to w istocie zwykły plik tekstowy, jeden dla każdego pakietu. Jeśli w pakiecie znajdował się skrypt poinstalacyjny, jest zapisywany do `/var/log/scripts`.

Możesz podawać kilka pakietów lub użyć symboli wieloznacznych. Pamiętaj, że `installpkg` nie ostrzega przed nadpisaniem zainstalowanego pakietu. Po prostu zastąpi stare pliki nowymi. Jeśli chcesz mieć pewność, że pliki starego pakietu zostały bezpiecznie usunięte, użyj `upgradepkg`.

### 18.2.3 `removepkg`

Do usuwania zainstalowanych pakietów z systemu służy `removepkg(8)`. Oto składnia:

```
# removepkg option package_name
```

`Removepkg` ma cztery opcje, można użyć tylko jednej naraz.

Tabela 18-2. Opcje `removepkg`

Opcja	Efekt
<code>-copy</code>	Pakiet jest kopiowany do katalogu z zachowanymi pakietami. Tworzy to drzewo oryginalnego pakietu bez usuwania go.
<code>-keep</code>	Zapisuje pliki tymczasowe tworzone podczas deinstalacji. Użyteczne w praktyce wyłącznie do celów debugowania.
<code>-preserve</code>	Pakiet jest usuwany, ale jednocześnie kopiowany do katalogu z zachowanymi pakietami.
<code>-warn</code>	Pokazuje, co stałoby się po usunięciu pakietu.

Jeśli przed `removepkg` ustawisz zmienną środowiskową `ROOT`, zostanie ona użyta jako katalog główny. Przydaje się to do konfigurowania nowych napędów tak, aby był na nich katalog główny. Zwykle montuje się je w `/mnt` albo innym katalogu niż `/`.

`Removepkg` przegląda inne zainstalowane pakiety i usuwa tylko pliki należące wyłącznie do danego pakietu. Skanuje również skrypt poinstalacyjny i usuwa dowiązania symboliczne przez niego utworzone.

Podczas procesu deinstalacji jest wyświetlany raport o postępie. Po usunięciu wpis jest przesuwany z

bazy danych do `/var/log/removed_packages`, a skrypt poinstalacyjny do `/var/log/removed_scripts`.

Podobnie jak przy `installpkg`, możesz podać kilka pakietów lub użyć symboli wieloznacznych.

## 18.2.4 `upgradepkg`

`upgradepkg(8)` aktualizuje zainstalowany pakiet Slackware. Oto składnia:

```
# upgradepkg package_name
```

albo

```
# upgradepkg old_package_name%new_package_name
```

`Upgradepkg` działa tak, że najpierw instaluje nowy pakiet, a potem usuwa stary, tak aby nie pozostały w systemie żadne pliki należące do starego. Jeśli nazwa aktualizowanego pakietu się zmieniła, użyj składni ze znakiem procenta do podania starego pakietu (tego zainstalowanego) i nowego (do którego aktualizujesz).

Jeśli przed `upgradepkg` ustawisz zmienną środowiskową `ROOT`, zostanie ona użyta jako katalog główny. Przydaje się to do konfigurowania nowych napędów tak, aby był na nich katalog główny. Zwykle montuje się je w `/mnt` albo innym katalogu niż `/`.

`Upgradepkg` może nie działać poprawnie. Zawsze wykonaj kopię zapasową plików konfiguracyjnych przed jego użyciem. Jeśli zostaną usunięte albo nadpisane, kopie są potrzebne do naprawy.

Podobnie jak przy `installpkg` i `removepkg`, możesz podać kilka pakietów lub użyć symboli wieloznacznych.

## 18.2.5 `rpm2tgz/rpm2targz`

Powszechnie wykorzystywanym systemem zarządzania pakietami jest obecnie Red Hat Package Manager (RPM). Wielu producentów oprogramowania oferuje swoje produkty w formacie RPM. Nie jest to rodzimy format Slackware, dlatego uważamy, że nie można na nim polegać. Niektóre programy jednak (nawet źródła) są dostępne jedynie w tym formacie.

Istnieje program konwertujący pakiety RPM do slackware'owego formatu `.tgz`. Możesz potem rozpakować taki pakiet (np. za pomocą `explodepkg`) do tymczasowego katalogu i przejrzeć jego zawartość.

Program `rpm2tgz` tworzy pakiet Slackware z rozszerzeniem `.tgz`, `rpm2targz` tworzy archiwum z rozszerzeniem `.tar.gz`.

## 18.3 Tworzenie pakietów

Tworzenie pakietów Slackware może być jednocześnie łatwe i trudne. Nie ma konkretnej metody tworzenia pakietu. Jedyne wymagania są takie, że musi on być w formacie `.tgz`, a ewentualny skrypt

poinstalacyjny musi być w `/install/doinst.sh`.

Jeśli chcesz tworzyć pakiety dla swojego systemu lub sieci, którą administrujesz, przejrzyj skrypty do tworzenia pakietów w drzewie źródeł Slackware. My używamy kilku metod do tworzenia pakietów.

### 18.3.1 `explodepkg`

`explodepkg(8)` wykonuje te działania, co `installpkg` w celu rozpakowania pakietu, ale go nie instaluje naprawdę i nie tworzy wpisu w bazie danych. Po prostu rozpakowuje zawartość do bieżącego katalogu.

Po przejrzaniu drzewa źródeł Slackware zobaczysz, jak używamy tego polecenia dla pakietów „szkieletowych” (ang. framework). Zawierają one szkielet układu końcowego pakietu. Są tam potrzebne nazwy plików (pliki mają zerową długość), prawa dostępu i własności. Skrypt tworzący „catuje” zawartość pakietu z katalogu źródłowego do katalogu tworzenia.

### 18.3.2 `makepkg`

`makepkg(8)` pakuje zawartość bieżącego katalogu i tworzy pakiet Slackware. Polecenie wyszukuje w drzewie dowiązania symboliczne i dodaje do skryptu poinstalacyjnego blok, który będzie je tworzył podczas instalacji. Ostrzega też o istnieniu plików o zerowej długości w drzewie pakietu.

To polecenie uruchamia się z reguły po utworzeniu drzewa pakietu.

### 18.3.3 Skrypty `SlackBuild`

Z konieczności pakiety Slackware tworzy się różnymi metodami. Programiści nie tworzą wszystkich pakietów tak, aby kompilowały się one w ten sam sposób. Wiele z nich skompilowano z takimi opcjami, które nie są zawarte w normalnych slackware'owych pakietach. Jeśli potrzebujesz jakiejś funkcji, będziesz musiał skompilować pakiet samodzielnie. Na szczęście w kodzie źródłowym wielu pakietów znajdują się skrypty `SlackBuild`.

Co to są skrypty `SlackBuild`? Są to wykonywalne skrypty powłoki, które uruchamia się jako `root`, aby skonfigurować, skompilować i utworzyć pakiety Slackware. Możesz dowolnie modyfikować te skrypty w katalogu ze źródłami i uruchamiać je, aby utworzyć własne wersje domyślnych pakietów Slackware.

## 18.4 Tworzenie znaczników i plików znaczników (dla programu `setup`)

Program instalacyjny Slackware (`setup`) zajmuje się instalowaniem oprogramowania w systemie. Istnieją pliki mówiące `setupowi`, które pakiety muszą być zainstalowane, które są opcjonalne, a które są wybrane jako domyślne.

Plik znaczników (ang. `tagfile`) znajduje się w katalogu pierwszej grupy programów i nazywa się `tagfile`. Zawiera on listę pakietów w danym zestawie dysków i ich status. Oto możliwe statusy:

**Tabela 18-3. Opcje statusu `tagfile`**

Opcja	Znaczenie
-------	-----------

ADD	pakiet jest niezbędny do prawidłowego działania systemu
SKP	pakiet zostanie pominięty automatycznie
REC	pakiet nie jest niezbędny, ale zalecany
OPT	pakiet jest opcjonalny

Format pliku jest prosty:

```
nazwa_pakiety: status
```

Jeden pakiet w wierszu. Oryginalne pliki znaczników dla każdej grupy programów są zapisywane jako `tagfile.org`. Jeśli źle utworzysz własny, możesz przywrócić oryginał.

Wielu administratorów tworzy własne pliki znaczników, uruchamia instalator i wybiera opcję „full”. Setup odczytuje te pliki i przeprowadza instalację zgodnie z ich zawartością. Jeśli użyłeś REC lub OPT, wyświetla się okno dialogowe z pytaniem, czy zainstalować dany pakiet. Dlatego zaleca się używanie tylko ADD i SKP przy pisaniu plików dla automatycznej instalacji.

Upewnij się, że zapisałeś pliki znaczników w tej samej lokalizacji co oryginały. Jeśli masz własne pliki, możesz też podać własną ścieżkę do nich.

## Rozdział 19. ZipSlack

### 19.1 Co to jest ZipSlack?

ZipSlack to specjalna wersja Linuksa Slackware. Jest to zainstalowany Slackware przeznaczony do uruchamiania z partycji DOS lub Windows. Jest to instalacja minimalna, nie ma w niej wszystkich programów dostępnych w Slackware..

Nazwa ZipSlack pochodzi od formy dystrybucji – jednego dużego pliku .ZIP. Użytkownicy Windows i DOS najprawdopodobniej znają ten format. Są to skompresowane archiwa. Archiwum ZipSlack zawiera wszystko, co potrzeba, aby mieć działający Slackware.

Należy podkreślić, że ZipSlack różni się od typowej instalacji Slackware. Jakkolwiek działają tak samo i mają te same programy, różnią się docelową grupą odbiorców i funkcjami. Poniżej omawiamy ważniejsze wady i zalety ZipSlack.

Jeszcze jedno: Zawsze przeczytaj dokumentację w aktualnym katalogu ZipSlack. Znajdują się tam najświeższe informacje dotyczące instalacji, uruchamiania i użytkowania tych produktów.

#### 19.1.1 Zalety

- Nie wymaga repartycjonowania dysku twardego.
- Stanowi doskonały sposób zapoznania się z Linuksem Slackware bez potrzeby „przebijania się” przez proces instalacji.

## 19.1.2 Wady

- Używa systemu plików DOS, który jest wolniejszy niż rodzimy system plików Linuksa.
- Nie działa z Windows NT.

## 19.2 Skąd wziąć ZipSlack

Zdobycie ZipSlack jest proste. Jeśli kupiłeś oficjalny komplet Slackware CD, już go masz. Znajdź CD z katalogiem `zipslack` i włóż go do napędu CD-ROM. Z reguły jest to dysk 3 lub 4, ale etykieta dysku, który często się zmienia, jest ważniejsza niż ta dokumentacja.

Jeśli chcesz pobrać ZipSlack z Internetu, sprawdź naszą stronę „Get Slack”, gdzie znajdują się aktualne informacje na temat pobierania:

<http://www.slackware.com/getslack/>

ZipSlack stanowi część każdej wersji Slackware. Wybierz odpowiednią wersję i przejdź do odpowiedniego katalogu FTP. Katalog najnowszej wersją znajduje się pod adresem:

<ftp://ftp.slackware.com/pub/slackware/slackware/>

ZipSlack jest w podkatalogu `/zipslack`. ZipSlack ma postać jednego dużego pliku `.ZIP` lub bloków o wielkości jednej dyskietki. Bloki są w katalogu `/zipslack/split`.

Nie pobieraj wyłącznie plików `.ZIP`. Pobierz też dokumentację i pliki obrazów startowych.

### 19.2.1 Instalacja

Po pobraniu odpowiednich komponentów rozpakuj plik `.ZIP` (lub pliki, jeśli pobrałeś bloki). Używaj 32-bitowego programu do rozpakowywania zip. Rozmiary i nazwy plików w archiwum nie są obsługiwane przez programy 16-bitowe. 32-bitowe programy to na przykład WinZip i PKZIP dla Windows.

ZipSlack zaprojektowano tak, aby rozpakować go do katalogu głównego napędu (np. `C:` lub `D:`). Utworzony zostanie tam katalog `\LINUX`, który zawiera instalację Slackware. W tym katalogu znajdują się również pliki niezbędne do uruchomienia systemu.

Katalog `\LINUX` pojawia się po rozpakowaniu na wybranym napędzie (od tej chwili zakładamy, że będzie to `C:`).

## 19.3 Uruchamianie ZipSlack

ZipSlack można uruchomić na kilka sposobów. Najczęściej używa się dołączonego pliku `LINUX.BAT`, aby uruchomić Slackware spod DOSa (lub spod trybu DOS w Windows 9x). Przed uruchomieniem plik trzeba zmodyfikować, dopasowując go do danego komputera.

Otwórz plik `C:\LINUX\LINUX.BAT` w wybranym edytorze. Na górze jest obszerny komentarz objaśniający, co i jak zmienić (a także co zrobić, jeśli uruchamiasz komputer z zewnętrznego napędu

Zip). Nie przejmuj się, jeśli nie rozumiesz ustawienia `root=`. Jest tam kilka przykładów, wybierz jeden i przetestuj. Jeśli nie zadziała, wyedytuj plik znowu, zakomentowując wiersz, który odkomentowałeś i wybierając inny.

Wiersze odkomentowuje się, usuwając „`rem`” z ich początku. Po odkomentowaniu zapisz plik, wyjdź z edytora i przejdź w tryb DOS.

Okno DOS wewnątrz Windows 9x NIE działa.

Wpisz `C:\LINUX\LINUX.BAT`, aby uruchomić Slackware. Jeśli wszystko zadziała, zobaczysz znak zgłoszenia logowania.

Zaloguj się jako `root`, z pustym hasłem. Załóż hasło `roota`, a także konto zwykłego użytkownika. W tym momencie możesz przeczytać inne rozdziały książki, aby dowiedzieć się więcej o działaniu i użytkowaniu systemu.

Jeśli metoda z `LINUX.BAT` nie zadziałała, przeczytaj plik `C:\LINUX\README.1ST`, w którym znajdują się informacje na temat innych metod uruchamiania.

## Słowniczek

### **Bezpieczna powłoka (Secure shell, ssh)**

Metoda logowania do zdalnego komputera wykorzystująca szyfrowanie (i dlatego bezpieczna). Istnieje wiele programów bezpiecznej powłoki. Potrzebne są zarówno klient, jak i serwer)

### **Biblioteka (Library)**

Zbiór funkcji, które mogą współdzielić różne programy.

### **Brama (Gateway)**

Zwana też „bramką”. Komputer, przez który dane w sieci są transferowane do innej sieci

### **Bufor ramek (Framebuffer)**

Rodzaj urządzenia graficznego. W Linuksie najczęściej oznacza programowy bufor ramek, który udostępnia standardowy interfejs bufora ramek programom, zachowując ukryte przed nimi specyficzne sterowniki urządzeń. Ta warstwa abstrakcji uwalnia programy z konieczności porozumiewania się z różnymi sterownikami urządzeń.

### **Darkstar**

Domyślna nazwa komputera w Slackware. Komputer będzie nazywał się darkstar, jeśli nie podasz innej nazwy.

Jeden z komputerów Patricka Volkerdinga, na którym pracuje; nazwa pochodzi od piosenki „Dark Star” zespołu „The Grateful Dead”

### **Demon (Daemon)**

Program stworzony tak, aby działał w tle i, bez interwencji użytkownika, wykonywał konkretne działania (zwykle świadczył usługi).

### **DNS**



Usługa nazw domen (Domain Name Service). System, w którym komputerom znajdującym się w sieci są nadawane nazwy, które tłumaczy się na adresy numeryczne.

### **Dowiązanie symboliczne (Symbolic link)**

Zwane też dowiązaniem miękkim. Jest to specjalny plik, który wskazuje na lokalizację innego pliku. Dowiązań symbolicznych używa się do tego, aby uniknąć duplikacji danych, gdy jeden plik jest potrzebny w kilku miejscach.

### **Dynamiczny program ładujący (Dynamic loader)**

Programy skompilowane w Linuksie zwykle używają fragmentów kodu (funkcji), które znajdują się w zewnętrznych bibliotekach. Przy uruchomieniu programów te biblioteki muszą zostać znalezione a wymagane funkcje załadowane do pamięci. Tym zajmuje się właśnie dynamiczny program ładujący.

### **Dysk główny (Root disk)**

Dysk (zwykle stały), na którym jest zapisany katalog główny.

### **Dyskietka dodatkowa (Supplemental disk)**

Dyskietka używana podczas instalacji Slackware, która nie zawiera jądra (znajdującego się na dyskietce rozruchowej) ani głównego systemu plików (na dysku głównym), ale dodatkowe pliki, takie jak moduły sieciowe lub obsługę PCMCIA.

### **Dyskietka rozruchowa (Boot disk)**

Zwana też często dyskietką startową, zawiera system operacyjny (w tym przypadku jądro Linuksa), z której można uruchomić komputer.

### **Epoka (Epoch)**

Okres w historii. W Uniksie „Epoka” („The Epoch”) zaczyna się 1 stycznia 1970, o godzinie 00:00:00 UTC. Unix i systemy uniksopodobne uważają to za „świt czasu”. Inne czasy są liczone względem tej daty.

### **FTP**

Protokół transferu plików (File Transfer Protocol). FTP jest popularną metodą przenoszenia danych między komputerami.

### **GID**

Identyfikator grupy (Group Identifier). GID jest niepowtarzalnym numerem przypisanym danej grupie użytkowników.

### **Grupa (Group)**

Użytkownicy w Uniksie należą do „grup”, które mogą zawierać wielu innych użytkowników. Używa się ich do bardziej ogólnej kontroli dostępu, niż pozwala na to obecność samych użytkowników.

### **Grupa programów (Software series)**

Zbiór powiązanych pakietów oprogramowania w Slackware. Wszystkie pakiety KDE są w grupie „kde”, pakiety do pracy w sieci w grupie „n” itp.

### **GUI**

Graficzny interfejs użytkownika (Graphical User Interface). Interfejs programowy, który używa elementów graficznych, takich jak przyciski, paski przewijania, okna itp., zamiast opartego

wyłącznie na tekście wejścia i wyjścia.

## **HOWTO**

Dokument opisujący, jak coś zrobić („how to...” - jak zrobić...), np. jak skonfigurować zapórę sieciową albo jak zarządzać użytkownikami i grupami. Duży zbiór tych dokumentów jest na stronie Linux Documentation Project – <http://www.tldp.org>, a polskie tłumaczenia na <http://www.jtz.org.pl>.

## **HTTP**

Protokół transferu hipertekstu (Hypertext Transfer Protocol). HTTP jest podstawowym protokołem, na którym opiera się działanie sieci WWW.

## **ICMP**

Protokół komunikatów kontrolnych Internetu (Internet Control Message Protocol). Bardzo prosty protokół sieciowy, używane głównie do pingowania.

## **Jądro (Kernel)**

Serce systemu operacyjnego. Jądro zapewnia podstawową kontrolę procesów i komunikuje się ze sprzętem komputerowym.

## **Katalog bieżący (Working directory)**

Uruchomiony program zakłada, że znajduje się w tym katalogu podczas działania.

## **Katalog domowy (Home directory)**

Zwany też katalogiem macierzystym. Jest to katalog, w którym użytkownik znajduje się natychmiast po zalogowaniu. W ramach swoich katalogów domowych użytkownicy mają pełne prawa i mniejszą lub większą władzę.

## **Katalog główny (Root directory)**

Reprezentowany przez „/” Znajduje się on na szczycie systemu plików, wszystkie inne pliki w „drzewie plików” wychodzą z niego.

## **Kod źródłowy (Source code)**

Zrozumiały (mniej lub bardziej) dla człowieka kod, w którym jest napisana większość programów. Kod źródłowy kompiluje się do kodu „binarnego”.

## **Kompilować (Compile)**

Przetwarzać kod źródłowy programu do kodu „binarnego”, który potrafi przeczytać komputer.

## **Konto (Account)**

Wszystkie informacje o użytkowniku, w tym nazwa użytkownika, hasło, informacja dla programu finger, numery UID i GID oraz katalog domowy. Utworzenie konta polega na dodaniu i zdefiniowaniu użytkownika.

## **Kroplik (Dot file)**

Nazwy plików w Linuksie, które mają być ukryte, zaczynają się od kropki ('.').

## **LILO**

Program ładujący Linuksa (Linux LOader). LILO jest najpowszechniej używanym menedżerem ładowania Linuksa.

## **LOADLIN**

Program LOADLIN działa pod MS DOSem lub Windows i uruchamia Linuksa. Najczęściej używa się go na komputerach, na których jest zainstalowany więcej niż jeden system operacyjny (w tym, oczywiście, Linux i DOS/Windows).

## **MBR**

Główny sektor rozruchowy (Master Boot Record). Zarezerwowane miejsce na dysku twardym, na którym są zapisane informacje, co robić przy uruchomieniu komputera. Można tam zapisać LILO albo inne menedżery ładowania.

## **Menedżer okien (Window manager)**

Program X, którego zadaniem jest tworzenie graficznego interfejsu użytkownika bardziej rozbudowanego niż proste rysowanie prostokątów w systemie X Window. Menedżery okien zwykle udostępniają paski tytułów, menu dla uruchomionych programów itp.

## **Moduł jądra (Kernel module)**

Fragment kodu jądra, zwykle jakiś sterownik urządzenia, który może być załadowany do pamięci i usunięty z niej niezależnie od głównego jądra. Moduły są użyteczne podczas aktualizacji urządzeń lub testowania ustawień jądra, ponieważ można je ładować i usuwać bez ponownego uruchamiania komputera.

## **MOTD**

Wiadomość dnia (Message Of The Day). Motd (zapisywana w Linuksie w `/etc/motd`) to plik tekstowy, którego zawartość jest wyświetlana każdemu użytkownikowi w momencie zalogowania się. Zwykle używa jej administrator systemu w charakterze „tablicy informacyjnej” służącej do komunikacji z użytkownikami.

## **Motif**

Popularny zestaw narzędzi programistycznych, używany w wielu starszych programach X.

## **Nazwa domeny (Domain name)**

Nazwa DNS komputera, z wyłączeniem nazwy samego komputera.

## **NFS**

Sieciowy System Plików (Network Filesystem). NFS pozwala na montowanie zdalnych systemów plików tak, jakby znajdowały się one na lokalnym komputerze, dzięki temu zapewniając przezroczystą metodę współdzielenia plików.

## **Notacja dziesiętna z kropkami (Dotted quad)**

Format adresu IP. Nazwany w ten sposób, ponieważ zawiera cztery liczby (w zakresie od 0 do 255, w systemie dziesiętnym) oddzielone kropkami.

## **Pager**

Program w X, który umożliwia podgląd i przełączanie się między wieloma „pulpitami”.

## **Pakiet haseł typu „shadow” (Shadow password suite)**

Pakiet shadow pozwala na przechowywanie zapisanych haseł niedostępnych dla użytkowników, podczas gdy pozostałe informacje z pliku `/etc/passwd` mogą odczytać wszyscy. Pomaga to zapobiegać próbom złamania haseł metodą „brute force” („na siłę”).

**Pakiet oprogramowania (Software package)**

Program i powiązane z nim pliki, zarchiwizowane i skompresowane do pojedynczego pliku, wraz z niezbędnymi skryptami lub informacjami pomocnymi przy instalacji, aktualizacji lub usuwaniu tych plików.

**Partycja (Partition)**

Część dysku twardego. Na partycji istnieją systemy plików.

**Pierwszy plan (Foreground)**

O programie, który przyjmuje lub kontroluje wejście terminala, mówi się, że działa na pierwszym planie.

**Plik znaczników (Tagfile)**

Plik używany przez program instalacyjny Slackware – `setup` – który opisuje zestaw pakietów do zainstalowania.

**Podsieć (Subnet)**

Zakres adresów IP, który jest częścią szerszego zakresu. Np. 192.168.1.0 to podsieć 192.168.0.0 (gdzie 0 to maska oznaczająca „niezdefiniowany”); jest to faktycznie podsieć „.1”.

**Powłoka (Shell)**

Powłoka udostępnia użytkownikowi interfejs wiersza poleceń. Kiedy patrzysz na tekstowy znak zgłoszenia (prompt), to znaczy, że jesteś w powłoce.

**Poziom pracy (Runlevel)**

Ogólny stan systemu zdefiniowany przez `init`. Poziom pracy 6 to ponowne uruchomienie komputera, poziom 1 to „tryb jednonazwiskowy”. poziom 4 to logowania X itp. W Slackware jest dostępnych 6 poziomów pracy.

**PPP**

Protokół punkt-punkt (Point-to-Point protocol). PPP używa się głównie do łączenia przez modem z Operatorem Internetu.

**Proces (Process)**

Uruchomiony program.

**Proces zawieszony (Suspended process)**

Proces, który został unieruchomiony do momentu zabicia go lub wznowienia.

**Program obudowujący (Wrapper program)**

Zwany też programem kopertującym. Jest to program, którego jedynym celem jest uruchamianie innych programów, ale zmieniając przy tym w jakiś sposób ich zachowanie, np. poprzez zmianę ich środowiska lub filtrowanie ich wejścia.

**Przeźródło wymiany (Swap space)**

Przeźródło na dysku używane przez jądro jako „wirtualna” pamięć RAM. Jest wolniejsza niż RAM, ale jako że przeźródło na dysku jest tańsze, przestrzeni wymiany może być zwykle dużo więcej. Jądro korzysta z niej, aby przechowywać rzadziej używane dane i jako zapas w razie zapelnienia fizycznej pamięci RAM.

**Punkt montowania (Mount point)**

Pusty katalog w systemie plików, w którym ma być „zamontowany”, albo „doczepiony” inny system plików.

**Sekcja podręcznika (Man section)**

Strony standardowego uniksowego podręcznika („man” = manual) online są pogrupowane w sekcje, aby było łatwiej odnaleźć potrzebną. Wszystkie strony dotyczące programowania w C są w sekcji 3, strony dotyczące administracji systemem – w sekcji 5 itp.

**Serwer nazw (Nameserver)**

Serwer informacji DNS. Serwery nazw tłumaczą nazwy DNS na numeryczne adresy IP.

**Serwer X**

Program w systemie X Window, który współdziała ze sprzętem graficznym i obsługuje faktyczne działanie programów X.

**Standardowe wejście (Standard input, stdin)**

Standardowy uniksowy strumień wejścia (wprowadzania) danych. Dane mogą być przekierowane lub potokowane (przepuszczane) do stdin programu z różnych źródeł.

**Standardowe wyjście (Standard output, stdout)**

Standardowy uniksowy strumień wyjścia. Zwykle wyjście tekstowe programu jest wypisywane na stdout, które jest oddzielone od wiadomości o błędach na standardowym wyjściu błędów (stderr), i które może być przekierowane lub potokowane do standardowych wejść innych programów lub do pliku.

**Standardowe wyjście błędów (Standard error, stderr)**

Standardowy uniksowy strumień wyjścia dla błędów. Programy wypisują komunikaty o błędach na stderr, tak aby łatwo było oddzielić je od normalnego wyjścia.

**Sterownik urządzenia (Device driver)**

Fragment kodu w jądrze, który kontroluje pracę danego komponentu sprzętowego.

**Superblok (Superblock)**

W Linuksie o partycjach mówi się w kategoriach bloków. Blok to 512 bajtów. Superblok to pierwsze 512 bajtów partycji.

**Sygnal (Signal)**

Uniksowe programy mogą porozumiewać się między sobą za pomocą prostych „sygnałów”, które mają swoje numery i z reguły specyficzne znaczenia. Polecenie `kill -l` wyświetli listę dostępnych sygnałów.

**System plików (Filesystem)**

Reprezentacja zapisanych danych, w której „pliki” danych są zorganizowane i przechowywane w „Katalogach”. System plików jest prawie uniwersalną reprezentacją danych zapisanych na dyskach (stałych i wymiennych).

**System X Window (X Window System)**

Zorientowany na pracę w sieci graficzny interfejs użytkownika używany w większości

unikspodobnych systemów operacyjnych, w tym w Linuksie.

### **Środowisko graficzne komputera (Desktop environment)**

Graficzny interfejs użytkownika (graphical user interface – GUI), który działa „na czubku” systemu X Window i udostępnia różne funkcje, takie jak zintegrowane aplikacje, spójny wygląd i zachowanie między programami a componentami, możliwości zarządzania plikami i oknami, itp. Więcej niż prosty menedżer okien.

### **Tablica routingu (Routing table)**

Informacje w jądrze, używane do „rutowania” (wysyłania odpowiednią drogą) danych w sieci. Zawiera takie rzeczy, jak domyślna brama, który interfejs sieciowy jest podłączony do której sieci itp.

### **Terminal**

Interfejs między człowiekiem a komputerem, składający się co najmniej z ekranu (lub ekranu wirtualnego) i jakiejś metody wejścia (prawie zawsze co najmniej z klawiatury).

### **Terminal wirtualny**

Oprogramowanie używane do symulacji wielu terminali, gdy jest dostępny tylko jeden zestaw urządzeń wejścia/wyjścia (klawiatura, monitor, mysz). Specjalne kombinacje klawiszy przełączają terminale wirtualne na pojedynczym terminalu fizycznym.

### **Tło (Background)**

O każdym procesie, który działa, nie przyjmując nic z wejścia terminala ani nie kontrolując go, mówi się, że działa w tle.

### **UID**

Identyfikator użytkownika (User Identifier). Niepowtarzalny numer, który identyfikuje użytkownika w systemie. Większość programów używa UIDów zamiast nazw użytkownika, ponieważ z numerami łatwiej pracować. Nazw użytkownika używa się głównie wtedy, kiedy sam użytkownik musi wiedzieć, co się dzieje.

### **Usługa**

Współdzielenie informacji i/lub danych między programami i komputerami jednego „serwera” (komputera usługującego) z wieloma „klientami” (komputerami korzystającymi z usług). Usługi to np. HTTP, FTP, NFS.

### **VESA**

Stowarzyszenie standardów elektroniki wideo (Video Electronics Standards Association). Terminem „VESA” zwykle określa się standard ustalony przez to stowarzyszenie. Prawie wszystkie nowoczesne karty graficzne są zgodne ze standardem VESA.

### **Wbudowane polecenie powłoki (Shell builtin)**

Polecenie wbudowane w powłokę, w przeciwieństwie do polecenia zewnętrznego programu. Np. `bash` ma wbudowane polecenie `cd`.

### **Węzeł urządzenia (Device node)**

Specjalny typ pliku w katalogu `/dev`, który reprezentuje komponent sprzętowy w systemie operacyjnym.

### **Zapis ósemkowy (Octal)**

8-cyfrowy system liczbowy, z cyframi od 0 do 7.

### **Zestaw narzędzi GUI (Toolkit GUI)**

Zestaw narzędzi GUI to zbiór bibliotek, które udostępniają programiście kod rysujący „widżety” (elementy okien), takie jak paski przewijania, pola wyboru itp. oraz tworzący graficzny interfejs użytkownika. Zestaw narzędzi GUI używany przez program zwykle określa jego wygląd i zachowanie („look and feel”).

### **Zmienna środowiskowa (Environment variable)**

Zmienna ustawiona w powłoce użytkownika, do której może odwoływać się użytkownik lub uruchomione przez niego w tej powłoce programy. Używa się ich głównie do przechowywania preferencji i domyślnych ustawień.

## **Dodatek A. Powszechna Licencja Publiczna GNU**

(Od tłum.: tłumaczenie pochodzi ze strony <http://gnu.org.pl/text/licencja-gnu.html>, oficjalny angielski tekst znajduje się na stronie <http://www.fsf.org/licensing/licenses/gpl.html>)

### **Tłumaczenie [GNU General Public License](#)**

#### **Uwaga!**

To jest nieoficjalne tłumaczenie Powszechnej Licencji Publicznej GNU na język polski. Nie zostało opublikowane przez Free Software Foundation i pod względem prawnym nie stanowi warunków rozpowszechniania oprogramowania stosującego GNU GPL -- ustanawia je wyłącznie [oryginalny angielski tekst licencji GNU GPL](#). Jednak mamy nadzieję, że pomoże ono lepiej zrozumieć Licencję osobom mówiącym po polsku.

#### **Note!**

This is an unofficial translation of the GNU General Public License into Polish. It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU GPL-- only the [original English text of the GNU GPL](#) does that. However, we hope that this translation will help Polish speakers understand the GNU GPL better.

### ***Powszechna Licencja Publiczna GNU***

#### **Wersja 2, czerwiec 1991**

Copyright © 1989, 1991 Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139 USA.

Zezwala się na kopiowanie i rozpowszechnianie wiernych kopii niniejszego dokumentu licencyjnego, jednak bez prawa wprowadzania zmian.

## Preambuła

Większość licencji na oprogramowanie pomyślana jest po to, aby odebrać użytkownikowi możliwość swobodnego udostępniania innym i zmieniania danego software'u. Natomiast w wypadku Powszechnej Licencji Publicznej GNU (*GNU General Public License*, GPL) celem jest zagwarantowanie użytkownikowi swobody udostępniania i zmieniania tego wolnego oprogramowania, a więc danie pewności, iż oprogramowanie jest wolno dostępne dla wszystkich użytkowników. Niniejsza Powszechna Licencja Publiczna dotyczy większości oprogramowania wydawanego przez Fundację Wolnego Oprogramowania (*Free Software Foundation*) oraz wszelkich innych programów, których autorzy zobowiązują się do jej stosowania. (Niektóre rodzaje oprogramowania wydawanego przez Fundację objęte są Powszechną Licencją Publiczną GNU dla Bibliotek, *GNU Library General Public License*). Użytkownik może stosować ją również do swoich programów.

Mówiąc o wolnym oprogramowaniu mamy na myśli swobodę, a nie cenę. Nasze Powszechne Licencje Publiczne wprowadzono po to, aby zapewnić Państwu swobodę rozpowszechniania kopii tego oprogramowania (i - jeśli ktoś chce - pobierania za tę usługę opłaty), jak również aby udostępnić kod źródłowy oraz umożliwić dokonywanie zmian tego oprogramowania lub wykorzystywania jego fragmentów w nowych, wolnych programach. Nie bez znaczenia jest też sama możliwość dotarcia do Państwa z informacją o wszystkich tych udogodnieniach.

W celu ochrony praw użytkownika jesteśmy zmuszeni wprowadzać ograniczenia zabraniające komukolwiek kwestionowanie jego praw albo sugerowanie rezygnacji z tych praw. Ograniczenia te sprowadzają się do pewnych dla Państwa obowiązków w przypadku rozpowszechniania przez Was kopii naszego oprogramowania bądź dokonywania w nim zmian.

Na przykład, jeśli rozprowadzacie Państwo kopie takiego programu, niezależnie czy gratisowo, czy za opłatą, musicie Państwo odbiorcy udzielić wszelkich tych praw, jakie mieliście sami. Musicie zapewnić mu również otrzymanie kodu źródłowego (lub możliwość otrzymania) oraz przedstawić niniejsze Warunki, aby mógł on poznać swoje prawa.

Ochrona Państwa praw przebiega w dwóch etapach:

1. zastrzegamy prawo własności autorskiej do oprogramowania,
2. oferujemy Państwu niniejszą licencję, która daje Wam sankcjonowane prawem zezwolenie na kopiowanie, rozpowszechnianie i/lub modyfikowanie tego oprogramowania.

Ponadto dla ochrony tak autora, jak i naszej, pragniemy mieć pewność, że każdy zrozumie, iż na niniejsze wolne oprogramowanie nie udziela się gwarancji. W razie dokonania w nim przez kogoś modyfikacji i puszczenia dalej do obrotu, pragniemy, aby dalsi odbiorcy zdawali sobie sprawę z tego, że problemy wprowadzone przez inne osoby nie są wyrazem oryginalnych działań twórców.

I rzecz ostatnia: każdemu wolnemu programowi stale zagrażają patenty na oprogramowanie. Naszym pragnieniem jest unikanie takiego niebezpieczeństwa, kiedy redystrybutorzy wolnego programu indywidualnie uzyskują prawa patentowe, nadając tym samym programowi charakter prawnie zastrzeżony. W celu zapobieżenia takim zjawiskom jednoznacznie wyjaśniliśmy, że każdy patent musi być wydawany albo dla swobodnego użytku przez wszystkich, albo nie wydawany wcale.

Poniżej podajemy dokładne zasady i warunki kopiowania, rozpowszechniania i modyfikowania.

## Zasady i warunki kopiowania, rozpowszechniania i modyfikowania

0.

Niniejsza Licencja dotyczy programów i innych prac, na których umieszczona jest pochodząca od właściciela praw autorskich informacja, że dany program lub praca może być rozpowszechniana na warunkach niniejszej Powszechnej Licencji Publicznej. Używane poniżej słowo "Program" oznacza



właśnie takie programy lub prace, zaś określenie "praca oparta na Programie" dotyczy albo Programu, albo pochodzącej od niego pracy w rozumieniu prawa autorskiego, to jest pracy zawierającej Program lub jego część dosłowną, bądź zmodyfikowaną i/lub przełożoną na inny język. (W dalszym ciągu niniejszego, pojęcie przekładu włącza się bez ograniczeń do terminu "modyfikacja"). Do każdego licencjobiorcy będziemy zwracać się "per Ty".

Niniejsza Licencja nie obejmuje działań innych niż kopiowanie, rozprowadzanie i modyfikowanie - nie mieszczą się one w jej zakresie. Czynność używania Programu nie jest poddana ograniczeniom, a produkty uzyskane z Programu objęte są Licencją tylko wtedy, gdy ich treść stanowi pracę opartą na Programie (niezależnie od stworzenia jej przy użyciu Programu). To, czy fakt taki jest prawdą, zależy od tego, co dany Program wykonuje.

1.

Możesz kopiować i rozprowadzać w dowolnych mediach wierne kopie kodu źródłowego Programu w otrzymanej formie pod warunkiem, że w widoczny sposób i odpowiednio podasz na każdej kopii właściwą informację o prawie autorskim i zrzeczenie się uprawnień z tytułu gwarancji; wszelkie napisy informacyjne na temat Licencji i faktu nieudzielania gwarancji musisz chronić przed uszkodzeniem, zaś wszystkim innym odbiorcom Programu musisz wraz z Programem wręczać egzemplarz niniejszej Licencji.

Możesz pobierać opłatę za fizyczną czynność przekazania kopii i według własnej decyzji możesz za opłatą proponować ochronę gwarancyjną.

2.

Możesz modyfikować swoją kopię czy kopie Programu oraz dowolne jego części, tworząc przez to pracę opartą na Programie, jak również kopiować i rozprowadzać takie modyfikacje i pracę na warunkach podanych w pkt.1 powyżej - pod warunkiem przestrzegania całości poniższych wymogów:

a.

Musisz spowodować umieszczenie na zmodyfikowanych plikach widocznej informacji o tym, że dane pliki zostały przez ciebie zmienione, wraz z datą dokonania zmian.

b.

Musisz doprowadzić do tego, aby każda rozpowszechniana lub publikowana przez ciebie praca, która w całości lub części zawiera Program, albo pochodzi od niego lub jego części, była w całości i bezpłatnie licencjonowana dla wszelkich stron trzecich na warunkach niniejszej Licencji.

c.

Jeżeli zmodyfikowany program podczas korzystania z niego w normalnym trybie odczytuje polecenia interaktywnie, musisz spowodować, aby po uruchomieniu (użyty w interaktywny sposób w najzwyczajszym trybie), wydrukowywał on lub wyświetlał powiadomienie o odnośnym prawie autorskim i braku gwarancji (ewentualnie o zapewnianiu gwarancji przez ciebie), oraz o tym, że użytkownicy mogą redystrybuować ten program na niniejszych warunkach wraz z informacją, jak użytkownik może zapoznać się z treścią niniejszej Licencji. (Wyjątek: jeśli sam Program jest interaktywny, ale normalnie nie drukuje takiego powiadomienia, twoja praca oparta na nim też nie musi wydrukowywać takiego powiadomienia).

Niniejsze wymogi odnoszą się do zmodyfikowanej pracy jako całości. Jeśli dające się ustalić sekcje danej pracy nie pochodzą od Programu i mogą być racjonalnie uważane za samodzielne i odrębne same w sobie, to niniejsza Licencja i jej warunki nie mają zastosowania do takich sekcji przy rozprowadzaniu ich przez ciebie jako odrębne prace. Jeśli jednak rozprowadzasz je jako część całości, będącej pracą opartą na Programie, rozpowszechnianie tej całości musi być dokonywane na warunkach niniejszej Licencji, której zezwolenia dla innych licencjobiorców rozciągają się w całej szerokości na tę całość, a tym samym i na każdą indywidualną jej część, niezależnie od jej autorstwa.

Dlatego też intencją tego fragmentu nie jest roszczenie sobie praw albo podważanie twych praw do pracy napisanej w całości przez ciebie. Chodzi nam raczej o korzystanie z prawa kontrolowania dystrybucji pochodnych i zbiorowych prac opartych na Programie.

I jeszcze jedno: samo tylko połączenie z Programem (lub z pracą opartą na Programie) innej pracy - nie opartej na Programie, w ramach wolumenu nośnika przechowywania lub dystrybucji, nie powoduje objęcia takiej pracy zakresem niniejszej Licencji.

### 3.

Możesz kopiować i rozprowadzać Program (lub opartą na nim pracę - zgodnie z pkt.2 w kodzie wynikowym lub w formie wykonywalnej w myśl postanowień pkt.1 i 2 powyżej, pod warunkiem zrealizowania również poniższych wymogów:

#### a.

Musisz dołączyć do niego odpowiadający mu, kompletny i możliwy do odczytania przez urządzenie cyfrowe kod źródłowy, który musi być rozpowszechniany na warunkach pkt.1 i 2 powyżej i na nośniku zwyczajowo używanym dla wzajemnej wymiany oprogramowania; lub

#### b.

dołączyć do niego pisemną ofertę, ważną co najmniej 3 lata, przyznającą każdej stronie trzeciej - za opłatą nie przekraczającą twego kosztu fizycznego wykonywania dystrybucji źródła - kompletną, odczytywalną przez urządzenia cyfrowe kopię odpowiadającego mu kodu źródłowego, rozprowadzaną na warunkach pkt.1 i 2 powyżej, na nośniku zwyczajowo używanym do wzajemnej wymiany oprogramowania; lub

#### c.

dołączyć do niego informację, jaką otrzymałeś na temat oferty rozprowadzania odpowiedniego kodu źródłowego. (Ta możliwość dozwolona jest tylko dla dystrybucji niehandlowej i jedynie wtedy, gdy otrzymałeś dany program w kodzie wynikowym lub formie wykonywalnej wraz z wymienioną ofertą - zgodnie z podpunktem "b" powyżej).

Określenie kod źródłowy dla pracy oznacza formę pracy preferowaną dla wprowadzania do niej modyfikacji. Dla wykonanej pracy, kompletny kod źródłowy oznacza cały kod źródłowy wszystkich modułów, wszelkie sprzężone z nią pośredniczące pliki opisujące oraz zbiory komend stosowane do sterowania kompilacją i instalowaniem programów. Niemniej jednak, jako wyjątek specjalny, dystrybuowany (w formie źródłowej albo binarnej) kod źródłowy nie musi obejmować niczego, co jest normalnie rozprowadzane przy pomocy głównych komponentów (kompilator, jądro itd.) systemu operacyjnego, na którym pracuje część wykonywalna, o ile sam taki komponent towarzyszy tej części.

Jeśli dystrybucja części wykonywalnej albo kodu wynikowego realizowana jest poprzez oferowanie

dostępu do kopii z wyznaczonego miejsca, to oferowanie równoważnego dostępu dla kopiowania kodu źródłowego z tego samego miejsca liczy się jako rozpowszechnianie kodu źródłowego, nawet gdy strony trzecie nie są zmuszone do kopiowania źródła wraz z kodem wynikowym.

4.

Poza przypadkami jednoznacznie dozwolonymi w niniejszej Licencji, nie możesz kopiować, modyfikować, sublicencjonować ani rozpowszechniać Programu. We wszystkich pozostałych wypadkach, każda próba skopiowania, sublicencjonowania lub rozpowszechnienia Programu jest nieważna i powoduje automatyczne wygaśnięcie twoich praw z tytułu Licencji. Niemniej jednak, stronom, które już otrzymały od ciebie kopie albo prawa w ramach niniejszej Licencji, licencje nie wygasają tak długo, jak długo strony te w pełni stosują się do nich.

5.

Nie musisz akceptować niniejszej Licencji, jeżeli jej nie podpisałeś. Niemniej jednak, nic innego nie zapewni ci zezwolenia na modyfikowanie lub rozprowadzanie Programu i pochodzących od niego prac. Działania takie są prawnie zabronione, jeżeli nie przyjmujesz niniejszej Licencji. Dlatego też, poprzez modyfikowanie bądź rozpowszechnianie Programu (lub pracy na nim opartej) dajesz wyraz swojej akceptacji dla Licencji i wszelkich jej postanowień i warunków dotyczących kopiowania, rozprowadzania i modyfikowania Programu lub opartych na nim prac.

6.

W każdym przypadku redystrybucji przez ciebie Programu (albo opartej na nim pracy), odbiorca automatycznie otrzymuje od pierwotnego licencjodawcy licencję na kopiowanie, rozpowszechnianie i modyfikowanie Programu na niniejszych zasadach i warunkach. Na korzystanie przez odbiorcę z udzielonych w niniejszej Licencji praw nie możesz narzucać już dalszych ograniczeń. Nie jesteś stroną odpowiedzialną za kontrolę przestrzegania Licencji przez osoby trzecie.

7.

Jeśli na skutek wyroku sądowego lub zarzutu naruszenia patentu, jak też z każdej innej przyczyny (nie ograniczonej do kwestii patentowych) zostaną narzucone na ciebie (niezależnie czy to mocą wyroku sądowego, umowy, czy w inny sposób) warunki sprzeczne z warunkami niniejszej Licencji, to nie zwalniają one ciebie z warunków Licencji. Jeśli nie możesz prowadzić dystrybucji tak, aby wypełniać jednocześnie swoje obowiązki z tytułu niniejszej Licencji i inne odnośne obowiązki, to w rezultacie nie możesz wcale rozprowadzać Programu. Na przykład, gdyby licencja patentowa nie zezwalała na wolną od opłat licencyjnych redystrybucję Programu przez wszystkie osoby, które otrzymały kopie bezpośrednio lub pośrednio od ciebie, to jedynym sposobem pozwalającym ci na przestrzeganie i licencji patentowej, i Licencji niniejszej, byłoby całkowite powstrzymanie się od jakiegokolwiek dystrybucji Programu.

Jeżeli w jakichś szczególnych okolicznościach któryś fragment niniejszego punktu stałby się nieważny lub niewykonywalny, to intencją jest, aby znajdowała zastosowanie pozostała część punktu, a treść całego punktu była stosowana w pozostałych okolicznościach.

Celem niniejszego punktu nie jest zachęcanie do naruszania patentów czy innych praw własności, albo też do podważania ich ważności; niniejszy punkt za swój jedyny cel ma ochronę integralności systemu rozpowszechniania wolnego oprogramowania, realizowanego za pomocą publicznych licencji. Wielu ludzi bezinteresownie wniosło swój wkład do stworzenia szerokiego zakresu oprogramowania upowszechnianego w tym systemie, mając zaufanie do konsekwentnego jego stosowania; wyłącznie do autora/ofiarodawcy należy decyzja, czy życzy on sobie rozprowadzania

oprogramowania za pośrednictwem innego systemu i licencjobiorca nie może tego prawa wyboru ograniczać.

Intencją niniejszego punktu jest jasne i wyraźne przedstawienie tego, co uważa się za skutki, jakie rodzi pozostała część niniejszej Licencji.

8.

W przypadku, gdy dystrybucja i/lub używanie Programu w niektórych krajach poddane jest ograniczeniom patentowym lub zastrzeżeniom prawami autorskimi, początkowy posiadacz praw autorskich, który poddaje Program pod oddziaływanie niniejszej Licencji, może dodać wyraźnie określone geograficzne ograniczenie rozpowszechniania wyłączające te kraje, dzięki czemu dystrybucja dozwolona będzie wyłącznie w krajach czy wśród krajów nie objętych takim wyłączeniem. W przypadku takim, niniejsza Licencja obejmuje dane ograniczenie tak, jakby było ono wpisane w jej treść.

9.

W miarę potrzeby Fundacja Wolnego Oprogramowania może publikować poprawione i/lub nowe wersje Powszechnej Licencji Publicznej. Takie nowe wersje będą napisane w duchu podobnym do obecnej wersji, ale mogą różnić się w szczegółach poruszających nowe problemy czy zagadnienia.

Każdej wersji nadaje się wyróżniający ją numer. Jeżeli Program podaje numer wersji niniejszej Licencji, odnoszący się do tej wersji i "wszelkich wersji następnych", masz do wyboru albo stosować się do postanowień i warunków tej wersji, albo którejkolwiek wersji późniejszej wydanej przez Fundację Wolnego Oprogramowania. O ile Program nie podaje numeru wersji niniejszej Licencji, możesz wybrać dowolną wersję kiedykolwiek opublikowaną przez Fundację.

10.

Jeśli chcesz włączyć części Programu do innych wolnych programów, których warunki rozpowszechniania są inne, zwróć się pisemnie do autora z prośbą o pozwolenie. W przypadku oprogramowania objętego przez Fundację prawem autorskim, napisz do Fundacji; czasami czynimy od tego odstępstwa. W naszej decyzji kierujemy się dwoma celami: utrzymania wolnego statusu wszystkich pochodnych naszego wolnego oprogramowania oraz - generalnie - promowania współdziałania i wielokrotnego stosowania oprogramowania.

## **WYRÓB BEZ GWARANCJI**

11.

**PONIEWAŻ PROGRAM JEST LICENCJONOWANY BEZPŁATNIE, NIE JEST OBJĘTY GWARANCJĄ W ZAKRESIE DOZWOLONYM PRZEZ OBOWIĄZUJĄCE PRZEPISY. O ILE NA PIŚMIE NIE STANOWI SIĘ INACZEJ, POSIADACZE PRAW AUTORSKICH I/LUB INNE STRONY ZAPEWNIAJĄ PROGRAM W STANIE, W JAKIM JEST ("JAK WIDAĆ") BEZ JAKIEJKOLWIEK GWARANCJI, ANI WYRAŻNEJ, ANI DOMYŚLNEJ, W TYM MIĘDZY INNYMI DOMYŚLNYCH GWARANCJI CO DO PRZYDATNOŚCI HANDLOWEJ I PRZYDATNOŚCI DO OKREŚLONYCH ZASTOSOWAŃ. CAŁOŚĆ RYZYKA W ZAKRESIE JAKOŚCI I SKUTECZNOŚCI DZIAŁANIA PROGRAMU PONOSISZ SAM. W RAZIE GDYBY PROGRAM OKAZAŁ SIĘ WADLIWY, PONOSISZ KOSZT CAŁEGO NIEZBĘDNEGO SERWISU, NAPRAWY I KORYGOWANIA.**

12.

**O ILE OBOWIĄZUJĄCE PRAWO NIE STANOWI INACZEJ ALBO CZEGOŚ INNEGO NIE**

UZGODNIONO W FORMIE PISEMNEJ, ŻADEN POSIADACZ PRAW AUTORSKICH ANI INNA STRONA MODYFIKUJĄCA I/LUB REDYSTRYBUJĄCA PROGRAM ZGODNIE Z POWYŻSZYMI ZEZWOLENIAMI, W ŻADNYM WYPADKU NIE JEST ODPOWIEDZIALNA WOBEC CIEBIE ZA SZKODY, W TYM SZKODY OGÓLNE, SPECJALNE, UBOCZNE LUB SKUTKOWE, WYNIKŁE Z UŻYCIA BĄDŹ NIEMOŻLIWOŚCI UŻYCIA PROGRAMU (W TYM, MIĘDZY INNYMI, ZA UTRATĘ DANYCH LUB POWSTANIE DANYCH NIEDOKŁADNYCH, ALBO ZA STRATY PONIESIONE PRZEZ CIEBIE LUB STRONĘ TRZECIE, JAK TEŻ NIEDZIAŁANIE PROGRAMU Z INNYMI PROGRAMAMI), NAWET JEŚLI DANY POSIADACZ BĄDŹ INNA STRONA ZOSTALI POWIADOMIENI O MOŻLIWOŚCI POWSTANIA TAKICH SZKÓD.

## KONIEC ZASAD I WARUNKÓW

### Jak stosować niniejsze Warunki do Twoich nowych programów

Jeśli opracowujesz nowy program i chciałbyś, aby stał się on przydatny dla szerokiego ogółu, najlepszą drogą do osiągnięcia tego będzie nadanie twemu programowi charakteru wolnego oprogramowania, które każdy może redystrybuować i zmieniać na niniejszych warunkach.

W tym celu do programu dołącz poniższe informacje. Bezpieczniej jest dołączyć je na początku każdego pliku źródłowego, dzięki czemu najskuteczniej można przekazać fakt nieistnienia gwarancji; każdy plik powinien przy tym nosić uwagę "copyright" i odnośnik, gdzie można znaleźć pełną informację.

*nazwa programu i informacja, do czego on służy.*  
Copyright (C) 19../20.. *nazwisko autora*

Niniejszy program jest wolnym oprogramowaniem; możesz go rozprowadzać dalej i/lub modyfikować na warunkach Powszechnej Licencji Publicznej GNU, wydanej przez Fundację Wolnego Oprogramowania - według wersji 2-giej tej Licencji lub którejsz z późniejszych wersji.

Niniejszy program rozpowszechniany jest z nadzieją, iż będzie on użyteczny - jednak BEZ JAKIEJKOLWIEK GWARANCJI, nawet domyślnej gwarancji PRZYDATNOŚCI HANDLOWEJ albo PRZYDATNOŚCI DO OKREŚLONYCH ZASTOSOWAŃ. W celu uzyskania bliższych informacji - Powszechna Licencja Publiczna GNU.

Z pewnością wraz z niniejszym programem otrzymałeś też egzemplarz Powszechnej Licencji Publicznej GNU (GNU General Public License); jeśli nie - napisz do Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Podaj też informacje o sposobie kontaktowania się z tobą pocztą elektroniczną lub zwykłą.

Jeśli dany program jest interaktywny, spraw, aby w momencie wchodzenia w tryb interaktywny wyświetlał on komunikat jak w poniższym przykładzie:

Gnomovision wersja 69, Copyright C 19.. *nazwisko autora*

Gnomovision wydawany jest ABSOLUTNIE BEZ ŻADNEJ GWARANCJI - w celu uzyskania dalszych szczegółów wpisz "show w".  
To jest wolne oprogramowanie i mile widziane jest dalsze rozpowszechnianie go przez ciebie na określonych warunkach - w celu uzyskania bliższych szczegółów wpisz "show c".

Powyższe hipotetyczne polecenia "show w" i "show c" winny powodować wyświetlenie odpowiednich części Powszechnej Licencji Publicznej. Oczywiście możesz używać innych poleceń niż "show w" i "show c"; mogą to być nawet kliknięcia myszą lub pozycje menu - co tylko sobie uznasz za stosowne.

Powinieneś też poprosić swego pracodawcę (jeśli pracujesz jako programista) czy też swoją szkołę (jeśli jesteś uczniem), o podpisanie, w razie potrzeby, "Rezygnacji z praw autorskich" do programu. Poniżej podajemy przykład (zmień nazwy/nazwiska):

*My, firma Jojodyne Sp. z o.o. niniejszym zrzekamy się i rezygnujemy z wszelkich interesów prawnych w zakresie praw autorskich do programu "Gnomovision" (który realizuje następujące funkcje...), napisanego przez p.Jana Kowalskiego.*

*Podpis: /-/ Gniewosław Wielkoważny*

*Gniewosław Wielkoważny, Prezes...itp*

Powszechna Licencja Publiczna nie zezwala na włączanie twego programu do programów prawnie zastrzeżonych. Jeśli twój program jest biblioteką podprogramów, możesz rozważyć, czy nie będzie korzystniej zezwolić na powiązanie prawnie zastrzeżonych aplikacji z biblioteką. Jeśli chciałbyś właśnie tego dokonać, zamiast niniejszej Licencji zastosuj Powszechną Licencję Publiczną GNU dla Bibliotek.

Tłum. Leszek Pietryka (leszekptr@gazeta.pl)