

# Debian GNU/Linux FAQ

'Autorzy' na stronie [71](#)  
'Tłumacze' na stronie [72](#)

wersja 4.0.3, 26 June 2008

## Streszczenie

Ten dokument odpowiada na często zadawane pytania dotyczące systemu Debian GNU/Linux.

## Prawa autorskie

Copyright © 1996-2003 by Software in the Public Interest

Udziela się zgody na dystrybuowanie wiernych kopii tego dokumentu pod warunkiem opatrzenia wszystkich kopii informacją o prawach autorskich a także niniejszym pozwoleniem.

Udziela się zgody na tworzenie i dystrybucję zmodyfikowanych wersji tego dokumentu na warunkach takich, jak w przypadku wiernych kopii, pod warunkiem, że całe zmodyfikowane dzieło rozprowadzane jest na zasadach identycznych do zawartych w niniejszym pozwoleniu.

Udziela się zgody na kopiowanie i dystrybucję tłumaczeń tego dokumentu w innym języku, zgodnie z powyższymi warunkami dotyczącymi modyfikowanych wersji dokumentu, z tym wyjątkiem, iż tłumaczenie niniejszego pozwolenia musi zostać zatwierdzone przez Free Software Foundation lub pozostać w brzmieniu oryginalnym.

Powyższe tłumaczenie **nie** zostało zatwierdzone przez Free Software Foundation i w związku z tym **nie** jest w żadnym stopniu wiążące. Ma jedynie charakter informacyjny, a w przypadku redystrybucji lub modyfikacji dokumentu należy stosować się do oryginału zamieszczonego poniżej.

Permission is granted to make and distribute verbatim copies of this document provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this document under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this document into another language, under the above conditions for modified versions, except that this permission notice may be included in translations approved by the Free Software Foundation instead of in the original English.

# Spis treści

<b>1</b>	<b>Definicje i przegląd</b>	<b>1</b>
1.1	Co to jest Debian GNU/Linux? . . . . .	1
1.2	OK, teraz już wiem co to jest Debian... a co to jest Linux?! . . . . .	2
1.3	Co to jest ten „Hurd”? . . . . .	3
1.4	Jaka jest różnica pomiędzy systemem Debian GNU/Linux a innymi dystrybucjami Linuksa? Dlaczego powinienem wybrać Debiana zamiast innej dystrybucji? . . . . .	3
1.5	Jak projekt Debian wygląda w porównaniu z projektem GNU Fundacji Wolnego Oprogramowania? . . . . .	4
1.6	Jak się wymawia słowo Debian i co ono oznacza? . . . . .	4
<b>2</b>	<b>Zdobywanie i instalacja systemu Debian GNU/Linux</b>	<b>5</b>
2.1	Jaka jest najnowsza wersja Debiana? . . . . .	5
2.2	Gdzie/jak mogę zdobyć płyty instalacyjne Debiana? . . . . .	5
2.3	Jak mogę zainstalować Debiana z płyt CD-ROM? . . . . .	6
2.4	Mam nagrywarke płyt CD, czy są gdzieś dostępne obrazy płyt CD? . . . . .	6
2.5	Czy mogę zainstalować go z dyskietek? . . . . .	6
2.6	Czy mogę pobrać i zainstalować Debiana bezpośrednio z Internetu? . . . . .	7
<b>3</b>	<b>Zagadnienia związane z kompatybilnością</b>	<b>9</b>
3.1	Na jakim sprzęcie można uruchomić system Debian GNU/Linux? . . . . .	9
3.2	Jak bardzo kompatybilny jest Debian z innymi dystrybucjami Linuksa? . . . . .	10
3.3	Jak bardzo kompatybilny jest kod źródłowy Debiana z kodem źródłowym innych systemów Uniksowych? . . . . .	10
3.4	Czy mogę używać pakietów Debiana (plików “.deb”) w dystrybucji Linuksa, której używam (RedHat/Slackware/...)? Czy mogę używać pakietów RedHata (plików “.rpm”) w systemie Debian GNU/Linux? . . . . .	11

---

3.5	Czy w Debianie mogę uruchamiać moje bardzo stare programy "a.out"?	11
3.6	Czy mogę na Debianie uruchomić moje stare programy libc5?	12
3.7	Czy mogę na Debianie skompilować stare programy libc5?	12
3.8	Jak mogę zainstalować nie-Debianowy program?	12
3.9	Dlaczego otrzymuję błędy "Can't find libX11.so.6" przy uruchamianiu czegoś tam?	13
3.10	Dlaczego nie mogę skompilować programów używających libtermcap?	13
3.11	Dlaczego nie mogę zainstalować AccelX?	13
3.12	Dlaczego moje stare aplikacje Motif XFree 2.1 nie działają?	14
<b>4</b>	<b>Oprogramowanie dostępne w Debianie</b>	<b>15</b>
4.1	Jakiego typu aplikacje i programy narzędziowe oferuje Debian GNU/Linux?	15
4.2	Kto to wszystko napisał?	15
4.3	W jaki sposób mogę pobrać aktualną listę programów, które istnieją w pakietach Debiana?	16
4.4	Czego brakuje?	16
4.5	Dlaczego podczas kompilowania programów dostaję komunikat "ld: cannot find -lfoo"? Dlaczego nie ma libfoo.so w pakiecie bibliotek Debiana?	16
4.6	Czy Debian obsługuje Javę?	17
4.7	Jak mogę sprawdzić, czy używam Debiana i której jego wersji?	17
4.8	Jak Debian obsługuje języki inne niż angielski?	17
4.9	Jak wyglądają regulacje prawne dotyczące amerykańskich przepisów eksportowych?	18
4.10	Gdzie podział się pine?	18
<b>5</b>	<b>Archiwa FTP Debiana</b>	<b>19</b>
5.1	Do czego służą te wszystkie katalogi w archiwach FTP Debiana?	19
5.2	Ile dystrybucji Debiana jest w katalogu dists?	19
5.3	A co z tymi wszystkimi nazwami jak "slink", "potato" itp.?	20
5.3.1	Jakie inne nazwy były wcześniej?	20
5.3.2	W jaki sposób powstają nazwy wydań?	20
5.4	A co z "frozen"?	21
5.5	Dystrybucja "sid"?	21

---

5.5.1	Zapiski historyczne o dystrybucji "sid" . . . . .	21
5.6	Co zawiera katalog stable? . . . . .	22
5.7	Co zawiera katalog testing? . . . . .	22
5.8	Co zawiera katalog unstable? . . . . .	23
5.9	Czym są te wszystkie katalogi wewnątrz <code>dists/stable/main</code> ? . . . . .	23
5.10	Gdzie jest kod źródłowy? . . . . .	23
5.11	Co jest w katalogu <code>pool</code> ? . . . . .	24
5.12	Co jest w "incoming"? . . . . .	24
<b>6</b>	<b>Podstawy zarządzania systemem pakietów Debiana</b>	<b>27</b>
6.1	Czym jest pakiet w systemie Debian? . . . . .	27
6.2	Jaki jest format pakietu binarnego w systemie Debian? . . . . .	28
6.3	Dlaczego nazwy plików pakietów w systemie Debian są tak długie? . . . . .	28
6.4	Czym jest plik kontroli w pakiecie systemu Debian? . . . . .	29
6.5	Czym jest plik <code>conffile</code> w systemie Debian? . . . . .	30
6.6	Czym są w systemie Debian skrypty: <code>preinst</code> , <code>postinst</code> , <code>prerm</code> i <code>postrm</code> ? . . . . .	30
6.7	Co to jest pakiet <i>Wymagany, Ważny, Standardowy, Opcjonalny</i> lub <i>Dodatkowy</i> ? . . . . .	31
6.8	Co to jest wirtualny pakiet? . . . . .	32
6.9	Co oznacza powiedzenie, że pakiet <i>Zależy, Zaleca, Proponuje, Wchodzi w konflikt, Zastępuje</i> lub <i>Dostarcza</i> inny pakiet? . . . . .	32
6.10	Co rozumie się przez <code>Pre-Depends</code> ? . . . . .	33
6.11	Co oznacza się przez <i>nieznany, instalowany, usunięty wyczyszczony i wstrzymany</i> w statusie pakietu? . . . . .	33
6.12	Jak można dokonać zatrzymania pakietu? . . . . .	34
6.13	Jak mogę zainstalować pakiet źródłowy? . . . . .	34
6.14	Jak się buduje binarne pakiety z pakietu źródłowego? . . . . .	35
6.15	Jak mogę stworzyć pakiet systemu Debian samodzielnie? . . . . .	35
<b>7</b>	<b>Narzędzia do zarządzania pakietami w systemie Debian</b>	<b>37</b>
7.1	Jakie programy do zarządzania swoimi pakietami udostępnia? . . . . .	37
7.1.1	<code>dpkg</code> . . . . .	37
7.1.2	<code>dselect</code> . . . . .	38
7.1.3	<code>dpkg-deb</code> . . . . .	40

---

7.1.4	apt-get	41
7.1.5	dpkg-split	41
7.2	Debian twierdzi, że można uaktualniać uruchomiony program; Jak to możliwe?	41
7.3	Jak mogę sprawdzić, które pakiety są już zainstalowane w Debianie?	41
7.4	Jak mam stwierdzić, który pakiet tworzy określony plik?	42
<b>8</b>	<b>Aktualizowanie systemu Debian</b>	<b>43</b>
8.1	Jak mogę uaktualnić mojego Debiana 1.3.1 (lub wcześniejszego), opartego na libc5, do wersji 2.0 (lub późniejszej), opartej na libc6?	43
8.2	Jak mogę utrzymać ciągle aktualną wersję Debiana?	44
8.2.1	APT	44
8.2.2	dpkg-ftp	45
8.2.3	mirror	46
8.2.4	dpkg-mountable	46
8.3	Czy muszę przechodzić w tryb pojedynczego użytkownika, by móc zaktualizować pakiety?	46
8.4	Czy muszę trzymać te wszystkie pliki .deb na swoim dysku?	46
8.5	Jak mogę otrzymać zapiski z czynności dodawania pakietów do systemu?	47
<b>9</b>	<b>Debian i jądro systemu</b>	<b>49</b>
9.1	Czy mogę skompilować i zainstalować jądro systemu bez dostosowywania go do specyfiki Debiana?	49
9.2	Jakich narzędzi dostarcza system Debian w celu budowania własnych wersji jądra systemu?	49
9.3	W jaki sposób mogę wykonać własną wersję dyskietki startowej?	51
9.4	Jakie narzędzia udostępnia Debian do obsługi modułów?	51
9.5	Czy mogę bezpiecznie odinstalować pakiet ze starym jądrem?	51
<b>10</b>	<b>Dostosowanie systemu Debian GNU/Linux do Twoich potrzeb</b>	<b>53</b>
10.1	Skąd mogę mieć pewność, że wszystkie programy używają tego samego rozmiaru papieru?	53
10.2	Jak mogę udostępnić urządzenia peryferyjne bez narażania bezpieczeństwa systemu?	53
10.3	W jaki sposób załadować czcionkę konsoli podczas startu systemu?	54

10.4	W jaki sposób skonfigurować domyślne parametry aplikacji dla środowiska X11?	54
10.5	Wydaje się, że każda dystrybucja posiada inne procedury startu systemu. Jak to wygląda w Debianie?	54
10.6	Wygląda na to że Debian nie używa <code>rc.local</code> aby dostosować proces startu systemu. Jakie narzędzia zostały dostarczone do tego celu?	55
10.7	Jak system zarządzania pakietami radzi sobie z pakietami, zawierającymi pliki konfiguracyjne innych pakietów?	56
10.8	Jak mogę podpisać plik instalowany przez inny pakiet tak żeby używana była moja wersja?	56
10.9	W jaki sposób dodać do listy dostępnych pakietów moje lokalnie zbudowane pakiety tak, aby system zarządzania pakietami o nich wiedział?	57
10.10	Niektórzy użytkownicy lubią <code>mawk</code> , inni <code>gawk</code> ; jedni lubią <code>vim'a</code> , inni lubią <code>elvis'a</code> ; niektórzy lubią <code>trn</code> , inni lubią <code>tin</code> . Jak Debian wspiera taką różnorodność upodobań?	57
<b>11</b>	<b>Pomoc dla systemu Debian GNU/Linux</b>	<b>59</b>
11.1	Jakie inne dokumentacje istnieją dla systemu Debian GNU/Linux?	59
11.2	Czy w Internecie dostępne są jakieś możliwości do dyskusji na temat Debiana?	60
11.2.1	Listy dyskusyjne.	60
11.2.2	Opiekunowie pakietów.	61
11.2.3	Grupy dyskusyjne.	61
11.3	Czy istnieje szybka metoda wyszukiwania materiałów związanych z systemem Debian GNU/Linux?	62
11.4	Czy istnieją logi znanych błędów?	62
11.5	Jak mogę zgłosić błąd dotyczący Debiana?	62
<b>12</b>	<b>Wspomaganie projektu Debian</b>	<b>65</b>
12.1	W jaki sposób mogę zostać deweloperem Debiana?	65
12.2	Jak mogę wnieść zasoby do projektu Debiana?	65
12.3	Jak mogę finansowo wspomóc projekt Debian?	66
12.3.1	Oprogramowanie w Publicznym Interesie (SPI)	66
12.3.2	Fundacja Wolnego Oprogramowania (FSF)	66
<b>13</b>	<b>Redystrybuowanie Debiana GNU/Linux jako produktu komercyjnego</b>	<b>67</b>
13.1	Czy mogę sprzedawać Debiana na przygotowanych przeze mnie płytach CD?	67

---

13.2	Czy Debian może być sprzedawany razem z płatnym oprogramowaniem? . . . .	67
13.3	Robię specjalną dystrybucję Linuksa dla specyficznego rynku. Czy mogę użyć Debiana GNU/Linux i dodać moje własne oprogramowanie? . . . . .	68
13.4	Czy mogę dołożyć do Debiana mój komercyjny program, który łatwo się instaluje?	68
<b>14</b>	<b>Spodziewane zmiany w następnej wersji Debiana</b>	<b>69</b>
14.1	Zwiększone bezpieczeństwo . . . . .	69
14.2	Wsparcie dla użytkowników nie znających języka angielskiego. . . . .	69
14.3	Więcej architektur . . . . .	70
14.4	Więcej jąder systemu . . . . .	70
<b>15</b>	<b>Ogólne informacje dotyczące Często Zadawanych Pytań (FAQ)</b>	<b>71</b>
15.1	Autorzy . . . . .	71
15.2	Tłumacze . . . . .	72
15.3	Informacje zwrotne . . . . .	73
15.4	Dostępność . . . . .	73
15.5	Format dokumentu . . . . .	73

# Rozdział 1

## Definicje i przegląd

### 1.1 Co to jest Debian GNU/Linux?

Debian GNU/Linux jest szczególną *dystrybucją* systemu operacyjnego Linux i licznych pakietów, które można na nim uruchomić.

W zasadzie, użytkownicy mogą pobrać z Internetu lub innych źródeł jądro Linux, a następnie je samodzielnie skompilować. Mogą dalej pobrać kody źródłowe wielu aplikacji w ten sam sposób, skompilować je, a potem zainstalować na swoim systemie. W przypadku skomplikowanych programów, ten proces może być nie tylko czasochłonny, ale też może powodować błędy. Aby tego uniknąć, użytkownicy często wolą otrzymać system operacyjny i pakiety z oprogramowaniem od jednego z dystrybutorów Linuksa. To co wyróżnia od siebie różne dystrybucje Linuksa to wybór oprogramowania, protokołów, sposób tworzenia, instalacji i zarządzania pakietami wraz z narzędziami za to odpowiedzialnymi. Do tego dochodzi jeszcze dokumentacja i inne usługi.

Debian GNU/Linux jest wynikiem prac i poświęcenia wielu ochotników, którym przyświecał cel stworzenia wysokiej jakości wolnego systemu operacyjnego zawierającego pełną gamę aplikacji. Idea wolnego systemu uniksopodobnego zrodziła się wraz z projektem GNU, a wiele aplikacji, które czynią system Debian GNU/Linux tak użytecznym, zostało opracowanych w ramach właśnie tego projektu.

Dla Debiana wolność (z angielskiego „free”, co znaczy zarówno „wolność” jak i „darmowość”) ma znaczenie zgodne z rozumieniem projektu GNU (zobacz Wytyczne Debiana dotyczące Oprogramowania Wolnodostępowego ([http://www.debian.org/social\\_contract#guidelines/](http://www.debian.org/social_contract#guidelines/))). Kiedy mówimy o wolnym oprogramowaniu, odnosimy się do wolności, nie do jego szybkości działania. Wolne oprogramowanie oznacza, że masz wolną rękę do dystrybuowania kopii, że otrzymałeś kody źródłowe, że możesz zmieniać te programy, lub używać ich części w innych wolnych programach.

Projekt Debian został stworzony przez Iana Murdocka w 1993 roku i na początku był sponsorowany przez projekt GNU Fundacji Wolnego Oprogramowania (FSF). Dziś deweloperzy Debiana myślą o nim jak o bezpośrednim potomku projektu GNU.

Debian GNU/Linux jest:

- **szeroko rozbudowany:** aktualna liczba pakietów w Debianie wynosi 18200. Użytkownicy mogą wybrać, które pakiety chcą zainstalować. Debian dostarcza narzędzie do tego celu. Listę i opisy aktualnie dostępnych pakietów dla Debiana możesz znaleźć na każdym z serwerów lustrzanych (<http://www.debian.org/distrib/ftplist/>) Debiana.
- **wolny w użytkowaniu i redystrybucji:** Nie ma żadnego konsorcjum członkowskiego czy wymagań płatności do brania udziału w jego rozprowadzaniu i rozwijaniu. Wszystkie pakiety, które formalnie są częścią systemu Debian GNU/Linux są wolne w redystrybucji, zazwyczaj na warunkach licencji GNU GPL.

Archiwum FTP Debiana zawiera również około 560 pakietów (w archiwum `non-free` i `contrib`), które są rozprowadzane pod specyficznymi warunkami (warunki rozprowadzania zawarte są w każdym pakiecie).

- **dynamiczny:** Dzięki udziałowi 1060 ochotników stale wnoszących nowy i ulepszony kod, Debian rozwija się gwałtownie. Co kilka miesięcy planowane są nowe wydania, a archiwum FTP aktualizowane jest codziennie.

Mimo, że Debian GNU/Linux jest Wolnym Oprogramowaniem, jest on podstawą na której mogą powstawać kolejne dystrybucje Linuksa. Przez dostarczanie niezawodnego, szeroko rozbudowanego systemu podstawowego, Debian dostarcza system Linux użytkownikom ze zwiększoną kompatybilnością i pozwala twórcom dystrybucji Linuksa skupić uwagę na rzeczach, które powodują, że ich dystrybucja jest szczególna. Zobacz 'Robię specjalną dystrybucję Linuksa dla specyficznego rynku. Czy mogę użyć Debiana GNU/Linux i dodać moje własne oprogramowanie?' na stronie 68 w celu uzyskania większej ilości informacji.

## 1.2 OK, teraz już wiem co to jest Debian... a co to jest Linux?!

Mówiąc krótko, Linux jest jądrem uniksopodobnego systemu operacyjnego. Pierwotnie był on przeznaczony dla maszyn 386 (i lepszych), a teraz jest on w trakcie przenoszenia na inne architektury, włącznie z wieloprocesorowymi. Linuksa napisał Linus Torvalds i wielu informatyków z całego świata.

Poza samym jądrem, system „Linux” zazwyczaj posiada:

- system plików zgodny ze Standardem Hierarchii Linuksowego Systemu Plików <http://www.pathname.com/fhs/>.
- szeroki wybór narzędzi Uniksowych, z których wiele było stworzonych przez projekt GNU oraz Fundację Wolnego Oprogramowania (FSF).

Kombinacja jądra Linuksa, jego systemu plików, oprogramowania pochodzącego z projektu GNU i FSF, oraz innego oprogramowania, jest tak zaprojektowana, aby osiągnąć kompatybilność ze standardem POSIX (IEEE 1003.1); zobacz 'Jak bardzo kompatybilny jest kod źródłowy Debiana z kodem źródłowym innych systemów Uniksowych?' na stronie 10.

W celu uzyskania większej ilości informacji na temat Linuksa, zobacz stronę Michaela K. Johnsona Linux Information Sheet (<ftp://ibiblio.org/pub/Linux/docs/HOWTO/INFO-SHEET/>) oraz Meta-FAQ (<ftp://ibiblio.org/pub/Linux/docs/HOWTO/META-FAQ/>).

### 1.3 Co to jest ten „Hurd”?

Hurd jest zespołem serwerów uruchamianych na mikrojądrze GNU Mach. Razem tworzą podstawę dla systemu operacyjnego GNU.

Obecnie Debian jest dostępny tylko dla Linuksa, ale razem z Debian GNU/Hurd zaczęliśmy również oferować Hurd jako platformę programistyczną, serwer oraz stację roboczą. Jednak Debian GNU/Hurd nie został jeszcze oficjalnie wydany i nie nastąpi to jeszcze przez pewien czas.

Proszę zobaczyć <http://www.gnu.org/software/hurd/> w celu uzyskania większej ilości informacji o GNU/Hurd, oraz <http://www.debian.org/ports/hurd/> aby uzyskać więcej informacji o Debian GNU/Hurd.

### 1.4 Jaka jest różnica pomiędzy systemem Debian GNU/Linux a innymi dystrybucjami Linuksa? Dlaczego powinienem wybrać Debiana zamiast innej dystrybucji?

Oto kluczowe cechy, które odróżniają Debiana od innych dystrybucji Linuksa:

**System zarządzania pakietów Debiana:** Cały system, lub jakakolwiek jego część, może być uaktualniany bez ponownego formatowania, bez gubienia istniejącej już konfiguracji i (w większości przypadków) bez ponownego uruchamiania komputera. Większość obecnie dostępnych dystrybucji Linuksa ma swego rodzaju system zarządzania pakietów; system zarządzania pakietów Debiana jest unikalny oraz szczególnie mocny. (zobacz ‘Podstawy zarządzania systemem pakietów Debiana’ na stronie [27](#))

**Otwarty rozwój:** Podczas gdy inne dystrybucje Linuksa są rozwijane przez małe, indywidualne, zamknięte grupy, lub komercyjnych sprzedawców, Debian jest jedyną dystrybucją Linuksa, która jest rozwijana przez wiele osób za pośrednictwem Internetu w tym samym duchu, co Linux oraz inne Wolne Oprogramowanie.

Ponad 1060 ochotników pracuje nad pakietami w liczbie ponad 18200 i polepszaniem systemu Debian GNU/Linux. Deweloperzy Debiana zasilają projekt nie przez pisanie nowych programów (w większości przypadków), ale przez tworzenie pakietów, według standardów projektu, zawierających istniejące oprogramowanie, poprzez informowanie zewnętrznych deweloperów o nowych błędach, oraz poprzez zapewnianie wsparcia użytkownikom. Więcej informacji dotyczącej tego jak zostać współpracownikiem znajduje się w ‘W jaki sposób mogę zostać deweloperem Debiana?’ na stronie [65](#).

**System śledzenia błędów (BTS):** Fakt tak dużego geograficznego rozszania deweloperów Debiana po całym świecie wymaga do użycia specjalnych narzędzi i szybkiego informowania o błędach i poprawkach, bez których rozwój systemu nie byłby tak dynamiczny. Użytkownicy są zachęceni do informowania o zauważonych przez nich błędach w stylu formalnym, po czym informacje o nich stają się dostępne w archiwach WWW, lub poprzez pocztę elektroniczną. Więcej o zarządzaniu logów dotyczących błędów przeczytasz w tym FAQ w ‘Czy istnieją logi znanych błędów?’ na stronie 62.

**Polityka Debiana:** Debian ma rozbudowaną specyfikację własnych standardów jakości, politykę Debiana. Ten dokument definiuje jakość i standardy, które każdy pakiet musi spełniać.

Więcej informacji o tym znajdziesz na naszej stronie o powodach dla których warto wybrać Debiana ([http://www.debian.org/intro/why\\_debian/](http://www.debian.org/intro/why_debian/)).

## 1.5 Jak projekt Debian wygląda w porównaniu z projektem GNU Fundacji Wolnego Oprogramowania?

System Debian jest tworzony w myśl idei Wolnego Oprogramowania, które było najpierw bronię przez Fundację Wolnego Oprogramowania (FSF) (<http://www.gnu.org/>) a w szczególności przez Richarda Stallmana (<http://www.stallman.org/>). Potężne narzędzia programistyczne, aplikacje i innego rodzaju oprogramowanie FSF, są również kluczową częścią Debiana.

Projekt Debian jest oddzielną jednostką Fundacji Wolnego Oprogramowania (FSF), jednakże komunikujemy się regularnie i razem współpracujemy nad różnymi projektami. Na wyraźną prośbę Fundacji Wolnego Oprogramowania nazywamy nasz system „Debian GNU/Linux” i jesteśmy szczęśliwi, że spełniamy tę prośbę.

Stałym celem Fundacji Wolnego Oprogramowania (FSF) jest rozwijanie nowego systemu operacyjnego zwanego GNU, bazującego na Hurdzie (<http://www.gnu.org/software/hurd/>). Debian pracuje z FSF nad tym systemem zwanym Debian GNU/Hurd (<http://www.debian.org/ports/hurd/>).

## 1.6 Jak się wymawia słowo Debian i co ono oznacza?

Słowo Debian wymawiamy: Deb'-i-en, z krótkim /e/ w części Deb i akcentem na pierwszą sylabę. Słowo Debian jest połączeniem imion Debry oraz Iana Murdocka, który założył projekt. (Słowniki dwuznacznie podają sposób wymowy słowa Ian (!), ale Ian preferuje ee'-en.)

## Rozdział 2

# Zdobywanie i instalacja systemu Debian GNU/Linux

### 2.1 Jaka jest najnowsza wersja Debiana?

Obecnie istnieją trzy wersje systemu Debian GNU/Linux:

*wersja 5.0, znana także jako dystrybucja 'stabilna'* Jest to stabilne i dobrze przetestowane oprogramowanie, zmieniające się, jeśli poprawiono jego bezpieczeństwo lub użyteczność.

*dystrybucja 'testowa'* Tutaj są umieszczone pakiety, które znajdują się w następnym wydaniu 'stabilnym'; zostały przetestowane w dystrybucji niestabilnej, ale jeszcze nie nadają się do wydania. Ta dystrybucja jest uaktualniana częściej niż 'stabilna', ale rzadziej niż 'niestabilna'.

*dystrybucja 'niestabilna'* Ta wersja jest aktualnie w fazie rozwoju, jest stale uaktualniana. Możesz pobrać pakiety z archiwum wersji 'niestabilnej' na dowolnej witrynie FTP Debiana i użyć ich do uaktualnienia swojego systemu w każdej chwili, ale nie powinieneś oczekiwać, że system będzie nadawał się do użytku w takim samym stopniu jak poprzednio lub będzie tak samo stabilny - oto dlaczego jest ona nazywana '**niestabilną!**'

Chcąc uzyskać więcej informacji zobacz 'Ile dystrybucji Debiana jest w katalogu dists?' na stronie [19](#).

### 2.2 Gdzie/jak mogę zdobyć płyty instalacyjne Debiana?

Obrazy płyt instalacyjnych możesz zdobyć pobierając odpowiednie pliki z jednego z serwerów lustrzanych Debiana (<http://www.debian.org/mirror/list>).

Pliki instalacyjne systemu są wydzielone w podkatalogach katalogu `dists/stable/main`, a nazwy tych podkatalogów odnoszą się do Twojej architektury, w następujący sposób: `disks-arch` (*arch* to "i386", "sparc", itd, sprawdź witrynę, aby uzyskać dokładną listę). W każdym z podkatalogów dla danej architektury może znajdować się kilka katalogów, każdy dla wersji instalacyjnej systemu, a aktualnie używana znajduje się w katalogu 'current' (który jest dowiązaniem symbolicznym).

Zobacz plik `README.txt` w tym katalogu w celu uzyskania dalszych instrukcji.

## 2.3 Jak mogę zainstalować Debiana z płyt CD-ROM?

Linux obsługuje system plików ISO 9660 (CD-ROM) z rozszerzeniami Rock Ridge (dawniej znany jako "High Sierra"). Kilku sprzedawców (<http://www.debian.org/CD/vendors/>) dostarcza system Debian GNU/Linux w tym formacie.

Ostrzeżenie: Kiedy instalujesz system z płyt CD-ROM, zazwyczaj nie jest dobrym pomysłem wybór metody dostępu do `cdrom` przy pomocy narzędzia `dselect`. Metoda ta jest zazwyczaj bardzo wolna. Na przykład metody `mountable` i `apt` są znacznie lepsze przy instalacji z CD-ROMów (zobacz 'dpkg-mountable' na stronie 46 i 'APT' na stronie 44).

## 2.4 Mam nagrywarke płyt CD, czy są gdzieś dostępne obrazy płyt CD?

Tak. Aby ułatwić sprzedawcom płyt CD dostarczanie dysków wysokiej jakości, udostępniamy oficjalne obrazy płyt CD (<http://cdimage.debian.org/>).

## 2.5 Czy mogę zainstalować go z dyskietek?

Najpierw ostrzeżenie: cały Debian GNU/Linux jest zdecydowanie zbyt duży, aby instalować go ze standardowych dyskietek 1.44 MB - instalacji takiej raczej nie nazwałbyś bardzo przyjemnym doświadczeniem.

Skopiuj pakiety Debiana na sformatowane dyskietki. Obsługiwane formaty to DOS, rodzimy format Linuksa "ext2" lub "minix"; nie zapomnij użyć polecenia `mount` właściwego dla użytego formatu dyskietki.

Użycie dyskietek powoduje następujące komplikacje:

- Krótkie nazwy plików MS-DOS. Jeśli próbujesz umieścić pakiety Debiana na dyskietkach formatu MS-DOS, stwierdzisz, że ich nazwy zazwyczaj są za długie i nie odpowiadają ograniczeniom długości nazw plików MS-DOS do 8.3. Aby rozwiązać ten problem, powinieneś użyć dysków z systemem plików VFAT, ponieważ VFAT obsługuje długie nazwy plików.

- Duże rozmiary plików: niektóre pakiety posiadają rozmiar większy niż 1,44 MB i nie zmieszczą się na jednej dyskietce. Do rozwiązania tej niedogodności użyj narzędzia `dpkg-split` (zobacz 'dpkg-split' na stronie 41), dostępnego w katalogu `tools`, na serwerach lustrzanych Debiana (<http://www.debian.org/mirror/list>).

Chcąc odczytywać i zapisywać dane z dyskietki, musisz posiadać obsługę dyskietek wkompiowaną w jądro; większość jąder wspiera obsługę napędów dyskietek.

Chcąc zamontować dyskietkę w punkcie montowania `/floppy` (katalog powinien zostać utworzony w czasie instalacji) wpisz

- `mount -t msdos /dev/fd0 /floppy/`  
jeżeli dyskietka znajduje się w napędzie A: i posiada system plików MS-DOS,
- `mount -t msdos /dev/fd1 /floppy/`  
jeżeli dyskietka znajduje się w napędzie B: i posiada system plików MS-DOS,
- `mount -t ext2 /dev/fd0 /floppy/`  
jeżeli dyskietka znajduje się w napędzie A: i posiada system plików ext2 (tj. podstawowy system plików Linuksa).

## 2.6 Czy mogę pobrać i zainstalować Debiana bezpośrednio z Internetu?

Tak. Możesz zainicjować system instalacyjny Debiana z zestawu plików, który możesz pobrać z naszego serwera FTP i jego serwerów lustrzanych.

Możesz pobrać mały plik z obrazem płyty CD, utworzyć z niego płytę startową, zainstalować z niej system podstawowy, a resztę poprzez sieć. Aby uzyskać więcej informacji zobacz <http://www.debian.org/CD/netinst/>.

Możesz także pobrać pliki z obrazami dyskietek, utworzyć z nich dyskietki startowe, uruchomić procedurę instalacyjną, a resztę Debiana pobrać poprzez sieć. Aby uzyskać więcej informacji zobacz <http://www.debian.org/distrib/floppyinst>.



---

## Rozdział 3

# Zagadnienia związane z kompatybilnością

### 3.1 Na jakim sprzęcie można uruchomić system Debian GNU/Linux?

Debian GNU/Linux zawiera kompletny kod źródłowy dla wszystkich zawartych w nim aplikacji, więc powinien działać na wszystkich platformach, które są wspierane przez jądro Linux; szczegóły można znaleźć w Linux FAQ (<http://en.tldp.org/FAQ/Linux-FAQ/intro.html#DOES-LINUX-RUN-ON-MY-COMPUTER>)

Obecne wydanie systemu Debian GNU/Linux 5.0, zawiera kompletną binarną dystrybucję dla następujących architektur:

*i386*: obejmuje komputery pracujące z procesorami Intelu lub kompatybilnymi z Intelem, czyli Intelowskie 386, 486, Pentium, Pentium Pro, Pentium II (zarówno Klamath jak i Celeron), Pentium III, i większość kompatybilnych procesorów AMD, Cyrix i innych.

*m68k*: obejmuje komputery Amiga i Atari posiadające procesory Motorola 680x0 dla  $x \geq 2$ ; z MMU.

*alpha*: systemy Alpha produkowane przez Compaq/Digital

*sparc*: obejmuje systemy Sun'a: SPARC i większość UltraSPARC.

*powerpc*: obejmuje maszyny PowerPC IBM/Motorola, wliczając w to CHRP, PowerMac i maszyny PReP.

*arm*: maszyny ARM i StrongARM.

*mips*: systemy big-endian MIPS SGI: Indy i Indiago2; *mipsel*: maszyny little-endian MIPS, Digital DECstations.

*hppa*: maszyny Hawlett-Packard'a PA-RISC (712, C3000, L2000, A500).

*ia64*: komputery Intel IA-64 ("Itanium").

s390: systemy IBM S/390 mainframe.

Binarna dystrybucja Debiana dla architektury Sparc64 (natywny UltraSPARC) jest obecnie w trakcie rozwoju.

Więcej informacji dotyczących uruchamiania systemu, partycjonowania Twoich dysków, obsługi urządzeń PCMCIA (PC Card) i podobnych zagadnień można znaleźć w Installation Manual, który jest dostępny na Naszej stronie WWW <http://www.debian.org/releases/stable/installmanual>.

## 3.2 Jak bardzo kompatybilny jest Debian z innymi dystrybucjami Linuksa?

Deweloperzy Debiana komunikują się z twórcami innych dystrybucji Linuksa w celu zachowania binarnej kompatybilności pomiędzy dystrybucjami Linuksa. Większość linuksowych, komercyjnych produktów działa równie dobrze pod kontrolą Debiana jak i pod kontrolą systemów dla których zostały stworzone.

Debian GNU/Linux przestrzega Linux Filesystem Hierarchy Standard (<http://www.pathname.com/fhs/>). Jednak standard dopuszcza własną interpretację niektórych zasad w nim opisanych, więc mogą istnieć nieznaczne różnice pomiędzy Debianem a pozostałymi systemami Linuksowymi.

## 3.3 Jak bardzo kompatybilny jest kod źródłowy Debiana z kodem źródłowym innych systemów Uniksowych?

Dla większości aplikacji kod źródłowy Linuksa jest kompatybilny z innymi systemami Uniksowymi. Linux wspiera prawie wszystko co jest dostępne w Uniksach System V i darmowych oraz komercyjnych pochodnych BSD. Jednak w obszarze biznesu związanego z Uniksem takie twierdzenie nie ma prawie żadnego znaczenia, ponieważ nie ma sposobu aby je udowodnić. W przemyśle związanym z tworzeniem oprogramowania wymagana jest całkowita kompatybilność, a nie kompatybilność "w prawie" wszystkich przypadkach. Z tego powodu przez lata narastała potrzeba stworzenia standardu, i dzisiaj takim głównym standardem kompatybilności kodu źródłowego w Uniksowych systemach operacyjnych jest POSIX.1 (IEEE Standard 1003.1-1990).

Linux w zamierzeniach ma przestrzegać POSIX.1, ale certyfikat POSIX.1 (i FIPS 151-2) jest dosyć drogi; stąd praca na zasadach zgodnych z POSIX jest dla deweloperów Linuksa trudniejsza. Koszty certyfikatu sprawiają, że zdobycie przez Debiana certyfikatu, nawet w przypadku przejścia procesu sprawdzania, jest mało prawdopodobne. (Proces ten jest obecnie przeprowadzany za darmo, więc oczekuje się, że więcej ludzi będzie pracować zgodnie z wytycznymi POSIX.1)

Unifix GmbH (Braunschweig, Niemcy) stworzył system Linuksowy, który dostał certyfikat jako system zgodny z FIPS 151-2 (rozszerzenie POSIX.1). Ta technologia jest dostępna we

własnej dystrybucji Unifix nazwanej Unifix Linux 2.0 i w Linux-FT Lasermoon'a.

### 3.4 Czy mogę używać pakietów Debiana (plików ".deb") w dystrybucji Linuksa, której używam (RedHat/Slackware/...)? Czy mogę używać pakietów RedHata (plików "rpm") w systemie Debian GNU/Linux?

Różne dystrybucje Linuksa używają różnych formatów pakietów i różnych programów do zarządzania tymi pakietami.

**Prawdopodobnie możesz:** Jest dostępny program, który rozpakuje pakiet Debiana na komputerze, gdzie zainstalowana jest inna dystrybucja Linuksa, i generalnie będzie działał, w sensie takim, że pliki będą rozpakowane. Odwrotna sytuacja prawdopodobnie też zachodzi, to znaczy program rozpakowujący pakiety RedHata albo Slackware'a na komputerze z dystrybucją opartą o system Debian GNU/Linux prawdopodobnie rozpakuje pakiet i umieści większość plików w zamierzonych katalogach. W większości jest to konsekwencja istnienia (i ogólnego przestrzegania zasad) Linux Filesystem Hierarchy Standard. Pakiet Alien (<http://packages.debian.org/alien>) jest używany do konwersji pomiędzy różnymi typami pakietów.

**Prawdopodobnie nie chcesz:** Większość menadżerów pakietów zapisuje swoje pliki administracyjne kiedy są używane do rozpakowywania archiwum. Te pliki administracyjne zwykle nie są ustandaryzowane. Dlatego rozpakowanie pakietu Debiana na 'obcym' komputerze będzie miało nieprzewidywalne efekty (a zwłaszcza nie użyteczne) dla działania menadżera pakietów używanego na tym komputerze. Analogicznie, programy do zarządzania pakietami z innych dystrybucji mogą skutecznie rozpakować swoje archiwa na komputerze z zainstalowanym Debianem, ale prawdopodobnie spowoduje to, że menadżer pakietów Debiana nie będzie poprawnie działać, gdy trzeba będzie zaktualizować lub usunąć pakiety, lub nawet podczas tworzenia listy pakietów zainstalowanych w systemie.

**Lepsze rozwiązanie:** Linux File System Standard (a więc i Debian GNU/Linux) wymaga, aby podkatalogi w `/usr/local/`, były w pełni dostępne dla użytkowników. Dlatego użytkownicy mogą rozpakować 'obce' pakiety w tym katalogu, zarządzać ich konfiguracją, aktualizować i usuwać zależnie od potrzeby.

### 3.5 Czy w Debianie mogę uruchamiać moje bardzo stare programy "a.out"?

Czy na pewno ciągle masz takie programy? :-)

Aby *uruchomić* program, który ma format `a.out` (np., QMAGIC lub ZMAGIC),

- Bądź pewny, że Twoje jądro wspiera uruchamianie programów wykonywalnych `a.out`, albo poprzez wkompilowanie tego bezpośrednio do jądra (`CONFIG_BINFMT_AOUT=y`) albo jako moduł (`CONFIG_BINFMT_AOUT=m`). (Pakiet z obrazem jądra Debiana zawiera moduł `binfmt_aout`.)

Jeżeli jądro Twojego Debiana obsługuje uruchamianie plików wykonywalnych `a.out` za pośrednictwem modułu, wtedy musisz być pewny, że moduł `binfmt_aout` jest załadowany. Możesz to zrobić w czasie procesu startu sytemu poprzez dodanie linii `binfmt_aout` do pliku `/etc/modules`. Możesz to także zrobić z linii poleceń, poprzez wpisanie polecenia `insmod DIRNAME/binfmt_aout.o` gdzie `DIRNAME` jest nazwą katalogu, gdzie jest przechowywany moduł zbudowany dla tej wersji jądra, która w danej chwili jest używana. W systemie z jądrem w wersji 2.2.17 `DIRNAME` będzie `/lib/modules/2.2.17/fs/`.

- zainstaluj pakiet `libc4`, który możesz znaleźć w jednej z dystrybucji wydanej przed 2.0 (ponieważ wtedy usunięto ten pakiet). Możesz spróbować jakiegoś starego CD-ROMu z Debianem (Debian 1.3.1 nadal zawiera ten pakiet) lub zobacz [ftp://archive.debian.org/debian-archive/](http://ftp://archive.debian.org/debian-archive/).
- Jeśli program, który chcesz uruchomić jest klientem X-ów `a.out`, zainstaluj pakiet `xcompat` (zobacz powyżej w celu sprawdzenia dostępności).

Jeśli posiadasz komercyjny program w formacie `a.out`, to dobry czas na poproszenie autorów o aktualizację w formacie `ELF`.

### 3.6 Czy mogę na Debianie uruchomić moje stare programy `libc5`?

Tak. Zainstaluj po prostu wymagane biblioteki `libc5` z części `oldlibs` (zawierającej stare pakiety właśnie dla zgodności ze starymi aplikacjami).

### 3.7 Czy mogę na Debianie skompilować stare programy `libc5`?

Tak. Zainstaluj `libc5-altdev` i `altgcc` (z części `oldlibs`). Możesz znaleźć odpowiednie, skompilowane z `libc5` `gcc` i `g++` w katalogu `/usr/i486-linuxlibc1/bin`. Umieść je w zmiennej `$PATH` tak by `make` i inne programy uruchamiały je jako pierwsze.

Jeśli musisz skompilować klienta X-ów `libc5`, zainstaluj pakiety `xlib6` i `xlib6-altdev`.

Uważaj, bo środowisko `libc5` nie jest już w pełni obsługiwane przez nasze pakiety.

### 3.8 Jak mogę zainstalować nie-Debianowy program?

Pliki znajdujące się w katalogu `/usr/local/` nie są pod kontrolą systemu zarządzania pakietami Debiana. Dlatego dobrą praktyką jest umieszczanie źródeł swoich programów w

`/usr/local/src/`. Dla przykładu możesz rozpakować pliki z archiwum “`cośtam.tar`” do katalogu `/usr/local/src/foo`. Po kompilacji binaria przenieś do `/usr/local/bin/`, biblioteki do `/usr/local/lib/`, a wszelkie pliki konfiguracyjne do `/usr/local/etc/`.

Jeśli Twoje programy i/lub pliki naprawdę muszą być umieszczone w jakichś innych lokalizacji to nadal możesz trzymać je w `/usr/local/` i utworzyć odpowiednie dowiązania symboliczne z wymaganych lokalizacji do `/usr/local/`. Np. możesz zrobić dowiązanie

```
ln -s /usr/local/bin/cośtam /usr/bin/cośtam
```

W każdym przypadku, jeśli licencja programu pozwala na redystrybucję powinieneś zastanowić się nad stworzeniem pakietu i przesłaniu go do Debiana. Przewodniki jak stać się opiekunem pakietu są załączone w podręczniku Polityki Debiana (zobacz ‘Jakie inne dokumentacje istnieją dla systemu Debian GNU/Linux?’ na stronie [59](#)).

### 3.9 Dlaczego otrzymuję błędy “Can’t find libX11.so.6” przy uruchamianiu czegośtam?

Taki błąd oznaczać może, że program jest złączony z bibliotekami X11 w wersji `libc5`. W takim przypadku musisz zainstalować pakiet `xlib6` z części `oldlibs`.

Możesz otrzymywać podobne komunikaty o błędach odnoszące się do pliku `libXpm.so.4`. W takim przypadku należy zainstalować bibliotekę XPM w wersji `libc5` z pakietu `xpm4`. 7 z części `oldlibs`.

### 3.10 Dlaczego nie mogę skompilować programów używających libtermcap?

Zamiast bazy oraz biblioteki `termcap` Debian używa bazy `terminfo` oraz biblioteki `ncurses` opisującej interfejsy terminali. Użytkownicy kompilujący programy, które wymagają pewnej wiedzy o interfejsie terminali powinni podmienić odwołania do biblioteki `libtermcap` na `libncurses`.

By obsługiwać binaria, które zostały już połączone z biblioteką `termcap` i do których nie posiadasz źródeł, Debian udostępnia pakiet `termcap-compat`. W nim zawarte są `libtermcap.so.2` i `/etc/termcap`. Zainstaluj ten pakiet jeśli program nie uruchamia się wypisując komunikat “can’t load library ‘libtermcap.so.2’” lub wspomina o brakującym pliku `/etc/termcap`.

### 3.11 Dlaczego nie mogę zainstalować AccelX?

AccelX używa do instalacji biblioteki `termcap`. Zobacz ‘Dlaczego nie mogę skompilować programów używających libtermcap?’ na tej samej stronie.

### 3.12 Dlaczego moje stare aplikacje Motif XFree 2.1 nie działają?

Musisz zainstalować pakiet `motifnls`, który dostarcza pliki konfiguracyjne dla XFree-2.1 pozwalające aplikacjom Motifa skompilowanym pod XFree-2.1 uruchamiać się pod XFree-3.1.

Bez tych plików niektóre aplikacje Motifa skompilowane na innych komputerach (takich jak Netscape) mogą się zawieszać przy próbie kopiowania i wklejania do pól tekstowych. Mogą występować również inne problemy.

## Rozdział 4

# Oprogramowanie dostępne w Debianie

### 4.1 Jakiego typu aplikacje i programy narzędziowe oferuje Debian GNU/Linux?

Tak jak większość dystrybucji Linuksa, Debian GNU/Linux oferuje:

- główne publikacje GNU służące do programowania, obsługi plików i obróbki tekstu, takie jak gcc, g++, make, texinfo, Emacs, powłoka Bash i wiele ulepszonych narzędzi systemu Unix,
- Perl, Python, Tcl/Tk i różne powiązane z nimi programy, moduły i biblioteki,
- TeX (LaTeX) i Lyx, dvips, Ghostscript,
- X Window System, który udostępnia sieciowy graficzny interfejs użytkownika wraz z niezliczoną ilością aplikacji napisanych dla środowiska X takich jak GNOME,
- pełen wachlarz aplikacji sieciowych: serwery protokołów internetowych takich jak HTTP (WWW), FTP, NNTP (news), SMTP i POP (poczta), serwery nazw; dostępne są przeglądarki internetowe i narzędzia do tworzenia stron WWW.

Debian obejmuje ponad 18040 pakietów, począwszy od serwerów wiadomości i przeglądarek poprzez programy obsługujące dźwięk, programy faksujące, bazy danych i arkusze kalkulacyjne, programy do obróbki obrazu, komunikacji, sieci, narzędzia pocztowe, serwery sieciowe, a nawet programy obsługujące ham-radio.

Część programów jest dostępna jako pakiety Debiana, choć - z powodu restrykcji licencyjnych - nie są formalną częścią dystrybucji. Ich aktualna ilość to 560.

### 4.2 Kto to wszystko napisał?

*Autorzy* każdego z programów w pakiecie są wymienieni w pliku `/usr/doc/PAKIET/copyright`, gdzie PAKIET należy zastąpić nazwą pakietu.

Koordynatorzy, którzy z tego oprogramowania utworzyli pakiet dla systemu Debian GNU/Linux są wymienieni w pliku kontrolnym (zobacz ‘Czym jest plik kontroli w pakiecie systemu Debian?’ na stronie 29), znajdującym się w każdym pakiecie.

### 4.3 W jaki sposób mogę pobrać aktualną listę programów, które istnieją w pakietach Debiana?

Kompletna lista jest dostępna w dwóch częściach:

**lista pakietów, które mogą być wszędzie dystrybuowane** z dowolnego serwera lustrzanego Debiana (<http://www.debian.org/distrib/ftplist/>), w pliku `indices/Maintainers`. Ten plik zawiera nazwy pakietów oraz nazwiska i adresy ich koordynatorów.

**lista pakietów, które nie mogą być dystrybuowane poza Stany Zjednoczone** z dowolnego serwera lustrzanego Debiana non-US (<http://www.debian.org/misc/README.non-US>), w pliku `indices-non-US/Maintainers`. Ten plik zawiera nazwy pakietów oraz nazwiska i adresy ich koordynatorów.

Strona Dostęp do pakietów Debiana przez WWW (<http://packages.debian.org/>) w wygodny sposób przedstawia informacje o pakietach podzielonych na około dwadzieścia “sekcji” archiwum Debiana.

### 4.4 Czego brakuje?

Lista programów, które wciąż czekają na umieszczenie w pakietach Debiana to Work-Needing and Prospective Packages list (<http://www.debian.org/devel/wnpp/>).

Więcej informacji o dodawaniu brakujących rzeczy znajdziesz ‘W jaki sposób mogę zostać deweloperem Debiana?’ na stronie 65.

### 4.5 Dlaczego podczas kompilowania programów dostaję komunikat “ld: cannot find -lfoo”? Dlaczego nie ma libfoo.so w pakiecie bibliotek Debiana?

Polityka Debiana wymaga, by taki link symboliczny (np. do `libfoo.so.x.y.z`) był umieszczony w oddzielnym pakiecie dla deweloperów. Takie pakiety zwykle nazywają się `libfoo-dev` lub `libfooX-dev` (zakładając, że pakiet z biblioteką nazywa się `libfooX`, gdzie X jest numerem).

## 4.6 Czy Debian obsługuje Javę?

Oficjalny Java Development kit z Sun Microsystems nie jest oprogramowaniem wolnodostępnym, i nie może znajdować się w dystrybucji Debiana. Jednakże zarówno JDK jak i kilka *wolnych* implementacji technologii Java są dostępne w pakietach Debiana. Możesz pisać, poprawiać i uruchamiać w Debianie programy napisane w Javie.

Uruchomienie apletu Javy wymaga przeglądarki internetowej z możliwością obsługi technologii Java. Kilka przeglądarek dostępnych w Debianie, takie jak Mozilla czy Konqueror obsługuje wtyczkę Java, co pozwala na uruchamianie w nich apletów Java. Netscape Navigator, choć nie jest wolnym oprogramowaniem, również jest dostępny jako pakiet Debiana i potrafi uruchamiać aplety Javy.

Sięgnij do Debian Java FAQ (<http://www.debian.org/doc/manuals/debian-java-faq/>) po więcej informacji.

## 4.7 Jak mogę sprawdzić, czy używam Debiana i której jego wersji?

W celu upewnienia się, że twój system został zainstalowany z prawdziwych dysków dystrybucyjnych Debiana sprawdź, czy istnieje plik `/etc/debian_version`, który zawiera tylko jedną linię z numerem wersji takim, jaki wynika z pakietu `base-files`.

Obecność programu `dpkg` świadczy o tym, że możesz instalować pakiety Debiana w swoim systemie, ale program ten został zaadoptowany przez wiele innych systemów operacyjnych i architektur, więc przestał to być wiarygodny sposób upewniania się, że twój system to Debian GNU/Linux.

Musisz pamiętać, że Debian składa się z wielu części, z których niemal każda może być uaktualniana niezależnie. Każde wydanie Debiana zawiera starannie zdefiniowaną i niezmienną zawartość. Uaktualnienia są dostępne oddzielnie. Jednowierszowy opis statusu instalacji pakietu `foo` jest dostępny po wydaniu polecenia `dpkg --list foo`. By zobaczyć wersje wszystkich aktualnie zainstalowanych pakietów, uruchom:

```
dpkg -l
```

Bardziej szczegółowa informacja dostępna jest przez:

```
dpkg --status foo
```

## 4.8 Jak Debian obsługuje języki inne niż angielski?

- Debian GNU/Linux jest rozpowszechniany z mapami znaków dla blisko dwóch tuzinów klawiatur wraz z narzędziami (w pakiecie `kbd`) do instalowania, przeglądania i modyfikowania map.

Podczas instalacji systemu użytkownik jest proszony o określenie klawiatury

- Znaczna większość programów w pakietach obsługuje wprowadzanie znaków innych niż US-ASCII używanych w innych językach (np. ISO-8859-1 czy ISO-8859-2); niektóre programy obsługują wielobajtowe języki, takie jak japoński czy chiński.
- Obecnie dostępna jest dokumentacja w języku niemieckim, hiszpańskim, węgierskim, fińskim, włoskim, japońskim, koreańskim i polskim poprzez pakiet `manpages-JĘZYK` (gdzie JĘZYK jest dwuliterowym kodem kraju). Aby korzystać z dokumentacji w innym języku niż angielski, należy ustawić odpowiednią wartość zmiennej `LC_MESSAGES`.

Na przykład dla polskich stron dokumentacji (tzw. manuals), zmienna `LC_MESSAGES` musi mieć wartość `'pl_PL'`. Spowoduje to, że program `man` będzie starał się znaleźć w `/usr/share/man/pl/` polską dokumentację.

## 4.9 Jak wyglądają regulacje prawne dotyczące amerykańskich przepisów eksportowych?

Prawo amerykańskie zabrania eksportu zaawansowanych technologii obronnych, takich jak np. programy szyfrujące. Dotyczy to m.in. PGP czy SSH.

Aby uchronić przed ryzykiem postępowania wbrew prawu, niektóre pakiety Debiana dostępne są tylko z serwera non-US <ftp://non-US.debian.org/debian-non-US/>. Jest także kilka serwerów lustrzanych poza Stanami Zjednoczonymi. Pełna lista znajduje się tutaj (<ftp://non-US.debian.org/debian-non-US/README.non-US>).

## 4.10 Gdzie podział się pine?

Z powodu obostrzeń w licencji, znajduje się on w obszarze non-free. Ponadto licencja nie zezwala nawet na rozpowszechnianie zmodyfikowanych plików binarnych, będziesz więc musiał samodzielnie go skompilować z plików źródłowych i nakładek Debiana.

Nazwa pakietu ze źródłami to `pine`. Możesz użyć pakietu `pine-tracker` by uzyskać informację o koniecznościach uaktualnień.

Zauważ, że jest wiele darmowych zamienników zarówno `pine` jak i `pico`, takich jak `mutt` i `nano`, które dostępne są w głównej części pakietów.

---

## Rozdział 5

# Archiwa FTP Debiana

### 5.1 Do czego służą te wszystkie katalogi w archiwach FTP Debiana?

Programy przystosowane dla Debiana dostępne są w jednym z kilku katalogów na każdym serwerze lustrzanym Debiana.

Katalog `dists` jest skrótem od “distributions” (z ang. dystrybucje) i jest miejscem, gdzie umieszczane są aktualnie dostępne wydania (i zapowiedzi wydań).

Katalog `pool` zawiera aktualne pakiety, zobacz ‘Co jest w katalogu `pool`?’ na stronie [24](#).

Dodatkowo udostępnione są następujące katalogi:

*/tools/*: Narzędzia DOS’owe do tworzenia dyskietek startowych, dzielenia dysku na partycje, kompresji i dekompresji plików oraz startowania Linuksa.

*/doc/*: Podstawowa dokumentacja Debiana, czyli FAQ (najczęściej zadawane pytania), informacje o systemie zgłaszania błędów itp.

*/indices/*: Pliki Maintainers i override.[The Maintainers file and the override files.]

*/project/*: głównie materiały opiekunów takie jak:

*project/experimental/*: Ten katalog zawiera pakiety i narzędzia, nad którymi ciągle trwają prace i są aktualnie na etapie testów alfa. Użytkownicy nie powinni tych pakietów używać, ponieważ może to być niebezpieczne, nawet dla najbardziej doświadczonych.

*project/orphaned/*: Pakiety osierocone, czyli porzucone przez swoich opiekunów i wycofane z dystrybucji.

### 5.2 Ile dystrybucji Debiana jest w katalogu `dists`?

Zazwyczaj są tam trzy dystrybucje, “stable” (stabilna), “testing” (testowana), “unstable” (niestabilna, rozwijana). Czasami jest także dystrybucja “frozen” (zamrożona) (zob. ‘A co z

“frozen”?’ na sąsiedniej stronie).

### 5.3 A co z tymi wszystkimi nazwami jak “slink”, “potato” itp.?

Są to tylko swego rodzaju “kryptonimy”. Kiedy dystrybucja Debiana jest w fazie przygotowywania, nie ma numeru wersji, tylko kryptonim. Zostały one nadane, żeby ułatwić pracę serwerom lustrzanym i uniknąć zbędnego kopiowania, kiedy dystrybucja np. zmienia stan z niestabilnego na stabilny.

Aktualnie, `stable` jest dowiązaniem symbolicznym do `woody` (i.e. Debian GNU/Linux 5.0) a `testing` do `sarge`. Oznacza to, że `woody` jest aktualnie dystrybucją stabilną, a `sarge` jest w fazie testów.

`unstable` jest stałym dowiązaniem symbolicznym do `sid`, jako że `sid` jest zawsze wersją niestabilną (see ‘Dystrybucja “sid”?’ na następnej stronie).

#### 5.3.1 Jakie inne nazwy były wcześniej?

Wcześniej używano nazw: `buzz` dla wydania 1.1, `rex` dla wydania 1.2, `bo` dla wydania 1.3.x, `hamm` dla wydania 2.0, `slink` dla wydania 2.1 i `potato` dla wydania 2.2.

#### 5.3.2 W jaki sposób powstają nazwy wydań?

Jak na razie były zapożyczane z filmu “Toy Story” wytwórni Pixar.

- `buzz` - (Buzz Lightyear) astronauta,
- `rex` - tyranozaur,
- `bo` - (Bo Peep) dziewczynka, która zaopiekowała się owieczką,
- `hamm` - świnka-skarbonka,
- `slink` - (Slinky Dog) “przegubowy” pies,
- `potato` - oczywiście Pan Ziemiak,
- `woody` - drewniany kowboj,
- `sarge` - sierżant w armii plastikowych żołnierzyków.

## 5.4 A co z “frozen”?

Kiedy dystrybucja “testing” jest wystarczająco dojrzała, kierownik wydania zaczyna ją “zamrażać”. Zmniejszane jest tempo aktualizacji pakietów, aby ograniczyć ilość błędów, które pojawiają się w wersji niestabilnej, przenoszonych do testowej.

Po pewnym czasie, dystrybucja “testing” jest całkowicie zamrożona, co oznacza wstrzymanie aktualizacji pakietów, o ile nie zawierają krytycznych dla wydania błędów. Dystrybucja zamrożona przechodzi przez kilkumiesięczny okres próbny składający się na przemian z aktualizacji i z okresów głębokiego zamrożenia nazywanych “cyklami testowymi”.

Prowadzimy listę błędów w dystrybucji “testing”, które mogą spowodować wstrzymanie konkretnego pakietu, lub całego wydania. W momencie, kiedy liczba takich błędów spadnie poniżej maksymalnej akceptowalnej wartości, dystrybucja uznawana jest za stabilną i wydawana z kolejnym numerem wersji.

Wraz z nowym wydaniem, zdezaktualizowana poprzednia dystrybucja “stable” przenoszona jest do archiwum. Więcej informacji znajdziesz pod adresem Debian - archiwum (<http://www.debian.org/distrib/archive>).

## 5.5 Dystrybucja “sid”?

*sid* czyli *unstable* jest miejscem, do którego trafia większość nowych pakietów. Dystrybucja ta nigdy nie doczeka się bezpośredniego wydania, ponieważ pakiety przeznaczone do wydania muszą najpierw wejść do dystrybucji *testing*, aby mogły być później wydane w dystrybucji *stable*. *sid* zawiera pakiety zarówno dla architektur przeznaczonych do wydania jak i tych nie wydawanych.

Nazwa “sid” także pochodzi z filmu “Toy Story”: Sid był chłopcem z sąsiedztwa, który psuł zabawki :-)

### 5.5.1 Zapiski historyczne o dystrybucji “sid”

Kiedy dzisiejszy Sid jeszcze nie istniał, organizacja sieciowych archiwów Debiana miała jedną dużą wadę: kiedy dokładano nową architekturę do bieżącej dystrybucji *unstable*, pakiety zrobione dla niej mogły być wydane dopiero wtedy, gdy ta dystrybucja stawała się nową dystrybucją *stable*. Dla wielu architektur nie dochodziło do tego i trzeba było przenosić odpowiadające im katalogi, gdy dochodziło do wydania dystrybucji. Było to niepraktyczne, ponieważ przenoszenie katalogów silnie obciążało łącza.

Administratorzy archiwów sieciowych przez kilka lat obchodzili ten problem, umieszczając binaria dla architektur jeszcze nie wydanych w specjalnym katalogu o nazwie *sid*. Dla architektur jeszcze nie wydanych, tworzono w chwili wydania dowiązanie z aktualnego katalogu *stable* do *sid* i od tej pory tworzono je w drzewie *unstable*, jak zwykle. Takie rozwiązanie było trochę mylące dla użytkowników.

Z nadejściem katalogu „pool” (zobacz ‘Co jest w katalogu pool?’ na stronie 24) zaczęto zapisywać pakiety binarne w lokalizacji kanonicznej w tymże katalogu, niezależnie od dystrybucji, więc wydanie dystrybucji przestało być związane z poddawaniem serwerów lustrzanych dużym obciążeniom (natomiast mamy do czynienia z dość sporymi, rozłożonymi w czasie obciążeniami w trakcie całego procesu rozwijania dystrybucji).

## 5.6 Co zawiera katalog stable?

- `stable/main/`: Ten katalog zawiera pakiety oficjalnie uznawane za najbardziej aktualne wydanie systemu Debian GNU/Linux.

Wszystkie z tych pakietów są zgodne z Wytycznymi Debiana dotyczącymi Oprogramowania Wolnodostępnego ([http://www.debian.org/social\\_contract#guidelines](http://www.debian.org/social_contract#guidelines)) i można je swobodnie używać, a także rozpowszechniać.

- `stable/non-free/`: Ten katalog zawiera informacje o pakietach, których rozpowszechnianie zostało ograniczone przez wymagania stawiane dystrybutorowi, które mówią o zwróceniu szczególnej uwagi, na kwestie praw autorskich danego programu.

Na przykład, licencje niektórych pakietów zabraniają komercyjnego rozpowszechniania. Inne znowu mogą być redystrybuowane, ale są typu shareware, a nie wolnym oprogramowaniem. Zanim włączy się którykolwiek z tych pakietów do jakiejś redystrybucji (np. na CD-ROMie), należy przestudiować jego licencję i prawdopodobnie przeprowadzić odpowiednie negocjacje.

- `stable/contrib/`: Ten katalog zawiera informacje o pakietach wolnych w rozumieniu DFSG (Debian Free Software Guidelines) i podlegających swobodnemu rozpowszechnianiu, ale w jakiś sposób zależnych od pakietu, który swobodnemu rozpowszechnianiu nie podlega i z tej przyczyny jest dostępny w sekcji non-free.

## 5.7 Co zawiera katalog testing?

Pakiety są przenoszone do katalogu testing po tym gdy przejdą stosowny okres testowania w unstable. Muszą być zsynchronizowane ze wszystkimi architekturami na których były zbudowane i nie mogą posiadać zależności, które czyniłyby je niemożliwymi do zainstalowania. Muszą również posiadać mniej błędów krytycznych dla wydania od wersji która jest dostępna w testing. W ten sposób mamy nadzieję, że ‘testing’ jest zawsze bliżej by stać się kandydatem do wydania.

Więcej informacji o ogólnym stanie “testing” oraz o poszczególnych pakietach dostępnych jest na <http://www.debian.org/devel/testing>

## 5.8 Co zawiera katalog unstable?

Katalog 'unstable' to aktualny stan prac deweloperów. Zapraszamy użytkowników do używania i testowania tych pakietów, ale ostrzegamy, o ich stanie gotowości. Przewagą używania dystrybucji niestabilnej jest fakt bycia zawsze na czasie z nowinkami systemu GNU/Linux, ale jeśli coś pójdzie nie tak to nie licz na pomoc ;)

W 'unstable' istnieją również podkatalogi main, contrib, non-free o takim samym przeznaczeniu jak w 'stable',

## 5.9 Czym są te wszystkie katalogi wewnątrz dists/stable/main?

W każdym z głównych drzew katalogów (`dists/stable/main`, `dists/stable/contrib`, `dists/stable/non-free`, `dists/unstable/main/`, itd.), pakiety binarne przechowywane są w podkatalogach których nazwy wskazują na jaką architekturę zostały skompilowane:

- `binary-all/`, dla pakietów, które są niezależne od architektury. Katalog zawiera na przykład skrypty Perla lub czystą dokumentację.
- `binary-i386/`, dla pakietów uruchamianych na maszynach PC 80x86.
- `binary-m68k/`, dla pakietów uruchamianych na maszynach opartych na procesorach Motorola 680x0. Aktualnie jest to robione głównie dla komputerów Atari i Amiga oraz dla niektórych płyt przemysłowych opartych na standardzie VME.
- `binary-sparc/`, dla pakietów uruchamianych na stacjach Sun SPARC.
- `binary-alpha/`, dla pakietów uruchamianych na maszynach DEC Alpha.
- `binary-powerpc/`, dla pakietów uruchamianych na maszynach PowerPC.
- `binary-arm/`, dla pakietów uruchamianych na maszynach ARM.

Zauważ, że aktualne pakiety binarne dla *woodiego* i kolejnych wydań nie znajdują się już w tych katalogach, ale w głównym katalogu `pool`. Pliki indeksów (`Packages` i `Packages.gz`) zostały zostawione dla zgodności.

Zobacz 'Na jakim sprzęcie można uruchomić system Debian GNU/Linux?' na stronie 9 by dowiedzieć się więcej.

## 5.10 Gdzie jest kod źródłowy?

Udostępniony jest kod źródłowy każdego elementu systemu Debian. Co więcej, ustalenia licencji większości programów w systemie *nakazują* aby kod źródłowy, albo informacja, gdzie jest udostępniony była rozpowszechniana razem z programami.

Zazwyczaj kod źródłowy jest umieszczony w katalogach "source", które są umieszczone w tym samym miejscu, co katalogi z binariami przeznaczonymi dla konkretnych architektur. Aktualnie można go znaleźć w katalogu `pool` (zobacz 'Co jest w katalogu `pool`?' na tej samej stronie). Aby pobrać źródła bez zagłębiania się w strukturę archiwum, spróbuj użyć polecenia `apt-get source nazwapakietu`.

Niektóre pakiety, ze względu na ograniczenia zawarte w ich licencji, publikowane są tylko w postaci źródeł. Przykładem takiego pakietu jest `pine`, zajrzyj do 'Gdzie podział się `pine`?' na stronie 18 aby dowiedzieć się więcej.

Kod źródłowy może, ale nie musi być dostępny dla pakietów w katalogach "contrib" i "non-free", które formalnie nie są częścią systemu Debian.

## 5.11 Co jest w katalogu `pool`?

Dawniej pakiety były składowane w podkatalogach katalogu `dists` odpowiadających dystrybucji, która je zawierała. Wynikały z tego różne trudności, np. serwery lustrzane musiały pobierać duże ilości danych, kiedy wprowadzano większe poprawki.

Pakiety przechowywane są w wielkim "worku" ("pool"), którego struktura oparata jest na nazwach pakietów źródłowych. Aby dało się nim łatwiej zarządzać, podzielono go na podkatalogi według sekcji ("main", "contrib", "non-free") oraz pierwszych liter nazw pakietów. W tym drzewie umieszczone są katalogi dla każdego pakietu, które zawierają binaria dla wszystkich architektur oraz źródła, z których je wygenerowano.

Możesz sprawdzić gdzie jest umieszczony pakiet wydając polecenie `apt-cache showsrc nazwapakietu` i odczytując linię "Directory:". Na przykład pakiety `apache` są przechowywane w `pool/main/a/apache/`. Ponieważ jest bardzo dużo pakietów `lib*`, są one traktowane w nieco odmienny sposób: na przykład pakiety `libpaper` umieszczone są w `pool/main/libp/libpaper/`.

Katalogi `dists` są ciągle używane do przechowywania plików z indeksami przez programy takie jak `apt`. Także, starsze dystrybucje nie zostały przystosowane do używania mechanizmu "pools".

Nie musisz zaprzętać sobie głowy tymi informacjami, `apt` i przypuszczalnie `dpkg-ftp` (zobacz 'Jak mogę utrzymać ciągle aktualną wersję Debiana?' na stronie 44) zajmą się tym za ciebie.

## 5.12 Co jest w "incoming"?

Kiedy opiekun przesyła pakiet do archiwum, umieszcza się go w katalogu "incoming" do czasu potwierdzenia jego pochodzenia i sprawdzenia, czy rzeczywiście ma zostać tam umieszczony.

Zazwyczaj nikt nie powinien instalować pakietów z tego katalogu. Jednakże dla bardzo rzadkich przypadków awaryjnych jest on udostępniony pod adresem <http://incoming>.

---

[debian.org/](http://debian.org/). Możesz pobrać z niego pakiety samodzielnie, sprawdzić podpisy GPG i sumy kontrolne MD5 w plikach `.changes` i `.dsc` a następnie je zainstalować.



## Rozdział 6

# Podstawy zarządzania systemem pakietów Debiana

### 6.1 Czym jest pakiet w systemie Debian?

Pakiety ogólnie zawierają wszystkie pliki potrzebne aby zaimplementować zbiór powiązanych poleceń lub cech. Są dwa typy pakietów w Debianie:

- *Pakiety binarne*, które zawierają pliki wykonywalne, konfiguracyjne, strony man oraz info, informacje o prawach autorskich i inną dokumentację. Pakiety te są rozpowszechniane w specyficznym dla systemu Debian formacie archiwum (zobacz ‘Jaki jest format pakietu binarnego w systemie Debian?’ na następnej stronie); wyróżniane są zazwyczaj przez rozszerzenie pliku `.deb`. Pakiety binarne mogą być rozpakowane narzędziem systemu Debian `dpkg`; szczegóły są opisane w tym dokumencie.
- *Pakiety źródłowe*, które składają się z pliku `.dsc` opisującego źródłowy pakiet (włącznie z nazwami następnymi plików), `.orig.tar.gz` zawierający oryginalne, niezmodyfikowane źródło w skompresowanym gzipem archiwum tar i plik `.diff.gz` zawierający zazwyczaj zmiany związane z systemem Debian wprowadzone w oryginalnym źródle. Narzędzie `dpkg-source` pakuje i rozpakowuje archiwa źródłowe systemu Debian; szczegóły są zawarte w tym dokumencie.

Instalacja oprogramowania przez system pakietów używa “zależności”, które są ostrożnie projektowane przez opiekunów pakietów. Te zależności są udokumentowane w pliku `control` związanym z każdym pakietem. Dla przykładu, pakiet zawierający kompilator GNU C (`gcc` “zależy” od pakietu `binutils`, który zawiera konsolidator i asembler. Jeżeli użytkownik spróbuje zainstalować `gcc` bez wcześniej zainstalowanego `binutils`, system zarządzania pakietami (`dpkg`) wyśle wiadomość z błędem, że potrzeba jeszcze `binutils` i zatrzyma instalację `gcc`. (Jednak, ułatwienie to może być pominięte przez upartego użytkownika, zobacz `dpkg(8)`.) Zobacz więcej w ‘Co oznacza powiedzenie, że pakiet *Zależy, Zaleca, Proponuje, Wchodzi w konflikt, Zastępuje* lub *Dostarcza* inny pakiet?’ na stronie [32](#) poniżej.

Narzędzia pakietowe systemu Debian mogą być użyte do:

- manipulowania i zarządzania pakietami lub częściami pakietów,
- jako pomoc dla użytkownika przy rozłożeniu pakietów, które muszą być przeniesione za pomocą ograniczonego rozmiarem nośnika takiego, jak dyskietka,
- pomocy wykonawcom w tworzeniu archiwum pakietów, i
- pomocy użytkownikom w instalacji pakietów, które znajdują się na zdalnym serwerze FTP.

## 6.2 Jaki jest format pakietu binarnego w systemie Debian?

“Pakiet” systemu Debian lub plik archiwum systemu Debian zawiera pliki wykonywalne, biblioteki i dokumentację, związane z poszczególnymi częściami programu lub zbiorem powiązanych programów. Normalnie, plik archiwum systemu Debian ma nazwę, która kończy się na `.deb`.

Wewnętrzny format pakietu binarnego Debiana jest opisany w instrukcji `deb(5)`. Ten wewnętrzny format jest tematem zmian (pomiędzy głównymi wydaniem systemu Debian GNU/Linux), dlatego zawsze używaj `dpkg-deb(1)` do manipulowania plikami `.deb`.

## 6.3 Dlaczego nazwy plików pakietów w systemie Debian są tak długie?

Nazwy plików pakietów binarnych systemu Debian są podporządkowane następującej konwencji: `<foo>_<VersionNumber>-<DebianRevisionNumber>.deb`

Zauważ, że `foo` jest zobowiązane być nazwą pakietu. Można poznać nazwę pakietu, związanego ze szczególnym plikiem archiwum systemu Debian (`.deb` file), wykorzystując do sprawdzenia jeden z następujących sposobów:

- obejrzyj plik “Packages” w katalogu, gdzie jest on przechowany na stronie FTP archiwum systemu Debian. Ten plik zawiera zwrotkę opisującą każdy pakiet; pierwsze pole w każdej zwrotce jest formalną nazwą pakietu.
- użyj polecenia `dpkg --info foo_VVV-RRR.deb` (gdzie VVV i RRR są odpowiednio wersją i poprawką pakietu w pytaniu). Polecenie to wyświetli, pomiędzy innymi rzeczami, rozwiniętą nazwę pakietu odpowiadającą plikowi archiwum.

Część VVV jest numerem wersji ustanawianym przez głównego programistę. Nie ma standardów w tym miejscu, więc numer wersji może posiadać format tak zróżnicowany jak “19990513” i “1.3.8pre1”.

Część `RRR` jest numerem weryfikacji w systemie Debian, i jest ustalany przez dewelopera Debiana (lub indywidualnego użytkownika jeśli wybierze zbudowanie pakietu samemu). Ten numer zgadza się z poziomem weryfikacji pakietu systemu Debian, dlatego nowy poziom weryfikacji zazwyczaj oznacza zmiany w pliku `Makefile` (`debian/rules`), pliku kontroli (`debian/control`), skryptach instalacyjnych i usuwających (`debian/p*`), lub w plikach konfiguracyjnych użytych w pakiecie.

## 6.4 Czym jest plik kontroli w pakiecie systemu Debian?

Szczegóły dotyczące treści pliku kontroli są zawarte w instrukcji Debian Packaging, rozdział 4, zobacz 'Jakie inne dokumentacje istnieją dla systemu Debian GNU/Linux?' na stronie 59.

W skrócie, przykładowy plik kontroli jest pokazany poniżej dla pakietu systemu Debian `hello`:

```
Package: hello
Priority: optional
Section: devel
Installed-Size: 45
Maintainer: Adam Heath <doogie@debian.org>
Architecture: i386
Version: 1.3-16
Depends: libc6 (>= 2.1)
Description: The classic greeting, and a good example
 The GNU hello program produces a familiar, friendly greeting. It
 allows nonprogrammers to use a classic computer science tool which
 would otherwise be unavailable to them.
.
 Seriously, though: this is an example of how to do a Debian package.
 It is the Debian version of the GNU Project's 'hello world' program
 (which is itself an example for the GNU Project).
```

Pole `Package` zawiera nazwę pakietu. To jest nazwa, za pomocą której pakiet może być przetwarzany przez narzędzia pakietowe i zazwyczaj jest podobna, ale nie koniecznie taka sama, jak pierwsza część nazwy pliku archiwum systemu Debian.

Pole `Version` zawiera numer wersji nadany przez głównego programistę i (w ostatniej części) poziom weryfikacji programu w pakiecie systemu Debian, tak jak jest to wyjaśnione w 'Dlaczego nazwy plików pakietów w systemie Debian są tak długie?' na sąsiedniej stronie.

Pole `Architecture` określa chip, dla którego ten konkretny pakiet binarny został skompilowany.

Pole `Depends` podaje listę pakietów, które muszą być zainstalowane w podanej kolejności, aby z sukcesem zainstalować pakiet.

Pole `Installed-Size` wskazuje ile miejsca na dysku zabierze zainstalowany pakiet. Pole to jest przeznaczone do użytku przez programy instalujące, aby pokazać czy jest dostępna wystarczająca ilość miejsca na dysku by zainstalować program.

Linia Section podaje nazwę “sekcji”, czyli gdzie ten pakiet systemu Debian jest przechowywany na stronach FTP systemu Debian. Jest to nazwa podkatalogu (wewnątrz jednego z głównych katalogów, zobacz ‘Do czego służą te wszystkie katalogi w archiwach FTP Debiana?’ na stronie 19), gdzie pakiet jest przechowywany.

Pole Priority wskazuje jak ważny jest ten pakiet dla instalacji tak, aby pseudo-inteligentne programy jak `dselect` lub `console-apt` mogły sortować pakiety w kategorii np. pakiety opcjonalnie instalowane. Zobacz ‘Co to jest pakiet *Wymagany, Ważny, Standardowy, Opcjonalny* lub *Dodatkowy*?’ na sąsiedniej stronie.

Pole Maintainer zawiera adres e-mail osoby, która jest aktualnie odpowiedzialna za utrzymywanie tego pakietu.

Pole Description zawiera krótkie podsumowanie cech pakietu.

Więcej informacji o wszystkich możliwych polach jakie może mieć pakiet, proszę zobacz Debian Packaging Manual, rozdział 4., “Control files and their fields”.

## 6.5 Czym jest plik `conffile` w systemie Debian?

Conffiles jest listą plików konfiguracyjnych (zazwyczaj umieszczonych w `/etc`), których system zarządzania pakietami nie będzie nadpisywał gdy pakiet będzie aktualizowany. To gwarantuje, że zawartość tych plików będzie zachowana i jest krytycznie ważną cechą, umożliwiającą aktualizację pakietu w działającym systemie.

Aby ustalić dokładnie, które pliki są zachowywane podczas aktualizacji pakietu, uruchom:

```
dpkg --status pakiet
```

I zobacz poniżej “Conffiles”.

## 6.6 Czym są w systemie Debian skrypty: `preinst`, `postinst`, `prepm` i `postrm` ?

Te pliki są wykonywalnymi skryptami, które są automatycznie uruchamiane przed lub po instalacji pakietu. Wraz z plikiem o nazwie `control`, wszystkie z tych plików są częścią sekcji “control” w pliku archiwum systemu Debian.

Indywidualne pliki to:

**preinst** Ten skrypt jest uruchomiony przed rozpakowaniem pakietu z jego pliku archiwum (“.deb”). Wiele skryptów ‘preinst’ zatrzymuje usługi dla pakietów, które będą aktualizowane, dopóki ich instalacja lub aktualizowanie się nie zakończy (następny, po pomyślnym wykonaniu skryptu ‘preinst’ jest skrypt ‘postinst’).

**postinst** Ten skrypt zazwyczaj kończy jakiejkolwiek wymagane konfiguracje pakietu `foo`, który był już rozpakowany z jego pliku archiwum (".deb"). Często skrypt 'postinst' prosi użytkownika o wprowadzenie informacji, i/lub ostrzega użytkownika, że jeśli zaakceptuje domyślne wartości, powinien pamiętać o tym aby przywrócić i zrekonfigurować pakiet jeśli wymaga tego sytuacja. Wiele skryptów 'postinst' wykonuje potem polecenia potrzebne do uruchomienia lub ponownego uruchomienia usługi po tym, jak nowy pakiet został zainstalowany.

**prepm** Ten skrypt zazwyczaj zatrzymuje różne procesy powiązane z pakietem. Jest uruchamiany przed usunięciem plików powiązanych z pakietem.

**postrm** Ten skrypt zazwyczaj modyfikuje powiązania lub inne pliki związane z `foo` i/lub usuwa pliki stworzone przez pakiet. (Zobacz też 'Co to jest wirtualny pakiet?' na następnej stronie.)

Obecnie wszystkie pliki control można znaleźć w katalogu `/var/lib/dpkg/info`. Pliki związane z pakietem `foo` zaczynają się od "foo" i mają rozszerzenie odpowiednio "preinst", "postinst", itd. . Plik `foo.list` w tym katalogu zawiera listę wszystkich plików, które były zainstalowane z pakietem `foo`. (Zauważ, że pliki te zlokalizowane są wewnątrz katalogu związanego z programem `dpkg`; nie powinieneś polegać na nim.)

## 6.7 Co to jest pakiet *Wymagany, Ważny, Standardowy, Opcjonalny* lub *Dodatkowy*?

Każdy pakiet systemu Debian ma przydzielony *priorytet* przez opiekunów dystrybucji, jako pomoc dla systemu zarządzania pakietami. Priorytety to:

- **Wymagany:** pakiety które są konieczne dla samego funkcjonowania systemu.  
Zawiera wszystkie niezbędne narzędzia do naprawy wad w systemie. Nie możesz usuwać tych pakietów, ponieważ Twój system może stać się całkowicie zepsuty i może być nawet niemożliwe użycie programu `dpkg`, aby to odwrócić. Systemy z samymi pakietami wymaganymi są prawdopodobnie nieużyteczne, ale mają wystarczającą funkcjonalność, aby umożliwić administratorowi systemu uruchomienie i instalację programów.
- **Ważny** Te pakiety powinny się znaleźć w każdym systemie rodziny Unix.  
Są to inne pakiety, bez których system nie będzie działał dobrze lub nie będzie użyteczny. Tutaj *NIE SĄ* zawarte wielkie aplikacje jak np. Emacs, X11 lub TeX. Te pakiety stanowią tylko samą infrastrukturę.
- **Standardowy** Te pakiety są standardowe w każdym systemie Linux, włączając dość małe systemy, ale też nie są ograniczone do systemów znakowych.  
Są to pakiety, które będą zainstalowane domyślnie jeśli użytkownik nie wybierze niczego innego. Nie są tu zawarte duże aplikacje, ale znajduje się tutaj Emacs (jest to bardziej część infrastruktury niż aplikacja) i spora część TeX i LaTeX (jeśli to możliwe, aby były dostępne bez systemu X-windows).

- **Opcjonalny** Zawarte tu są te wszystkie pakiety, które możesz chcieć zainstalować, jeśli nie wiesz czym one są lub nie masz sprecyzowanych wymagań.  
Znajdują się tu X11, pełna dystrybucja TeX i dużo innych aplikacji.
- **Dodatkowy:** Występują tu pakiety, które powodują konflikt z innymi pakietami posiadającymi wyższy priorytet, są zapewne użyteczne jeśli już wiesz czym one są, albo sprecyzowałeś wymagania, które czynią te pakiety nieodpowiednie dla "Opcjonalny".

## 6.8 Co to jest wirtualny pakiet?

Wirtualny pakiet jest ogólną nazwą, która odnosi się do jednej z grupy pakietów, w której każdy z pakietów dostarcza podobnej podstawowej funkcjonalności. Dla przykładu oba programy `tin` i `trn` są czytnikami wiadomości i dlatego powinny spełnić zależności programów, które wymagają czytnika wiadomości w systemie, aby pracować lub być użytecznym. Dlatego też o obu programach można powiedzieć, że dostarczają "wirtualny pakiet" nazwany `news-reader`.

Podobnie `smail` i `sendmail`, oba dostarczają funkcjonalności agenta transportu wiadomości. Oba mogą powiedzieć że dostarczają wirtualny pakiet "agent transportu wiadomości". Jeśli jeden z nich jest zainstalowany, wtedy dowolny program zależny od instalacji `mail-transport-agent` będzie usatysfakcjonowany przez istnienie wirtualnego pakietu.

System Debiana dostarcza mechanizm, który, jeśli jest zainstalowany więcej niż jeden pakiet dostarczający ten sam wirtualny pakiet, pozwala administratorowi systemu ustawić jeden preferowany pakiet. Odpowiednie polecenie to `update-alternatives` i jest opisane dalej w 'Niektórzy użytkownicy lubią mawk, inni gawk; jedni lubią vim'a, inni lubią elvis'a; niektórzy lubią trn, inni lubią tin. Jak Debiana wspiera taką różnorodność upodobań?' na stronie 57.

## 6.9 Co oznacza powiedzenie, że pakiet *Zależy*, *Zaleca*, *Proponuje*, *Wchodzi w konflikt*, *Zastępuje* lub *Dostarcza* inny pakiet?

System pakietów Debiana posiada zasięg "zależności" pakietu, który jest zaprojektowany aby wskazywać (w pojedynczej fladze) poziom, na którym Program A może działać niezależnie od istnienia Programu B w danym systemie:

- Pakiet A *zależy* od Pakietu B, jeśli B absolutnie musi być w kolejności zainstalowany aby uruchomić A. W pewnych przypadkach A zależy nie tylko od B, ale od wersji B. W tym przypadku zależność wersji jest zazwyczaj dolnym limitem, w tym sensie że A zależy od dowolnej wersji B większej niż pewna sprecyzowana wersja.
- Pakiet A *zaleca* Pakiet B, jeśli opiekun pakietu oceni, że większość użytkowników nie będzie chciała A bez posiadania funkcjonalności dostarczanej przez B.
- Pakiet A *proponuje* Pakiet B, jeśli B zawiera pliki, które są powiązane (i zazwyczaj poszerzają) funkcjonalność A.

- Pakiet A *wchodzi w konflikt* z Pakietem B, gdy A nie będzie działać, jeśli B jest zainstalowany w systemie. Najczęściej konflikty są przypadkami gdzie A zawiera pliki, które posiadają ulepszenia w stosunku do plików w B. “wchodzi w konflikt” jest często połączone z “zastępuje”.
- Pakiet A *zastępuje* Pakiet B, kiedy pliki instalowane przez pakiet B są usuwane i (w pewnych przypadkach) nadpisywane przez pliki z A.
- Pakiet A *dostarcza* Pakiet B, kiedy wszystkie z plików i działań pakietu B są włączone w A. Ten mechanizm umożliwia użytkownikom z ograniczonym miejscem na dysku pobranie tylko tych części pakietu A, które są naprawdę potrzebne.

Bardziej szczegółowe informacje o użyciu tych terminów można znaleźć w Packaging manual i Policy manual.

## 6.10 Co rozumie się przez Pre-Depends?

“Pre-Depends” jest specjalną zależnością. W przypadku wielu pakietów, `dpkg` rozpakuje ich pliki archiwum (np. są to pliki `.deb`) niezależnie od tego, czy pliki od których one zależą istnieją w systemie czy nie. Upraszczając, rozpakowanie oznacza, że `dpkg` wydobędzie pliki z pliku archiwum, które zostały przeznaczone do zainstalowania w twoim systemie plików i umieści je we właściwym miejscu. Jeśli te pakiety *depend* istnieją w kilku innych pakietach w Twoim systemie, `dpkg` odmówi dokończenia instalacji (przez wykonanie akcji “configure”), dopóki inne pakiety nie zostaną zainstalowane.

Jakkolwiek dla pewnych pakietów `dpkg` będzie odrzucał nawet ich rozpakowanie, dopóki nie zostaną spełnione pewne zależności. Takie pakiety są nazywane “Pre-depend” i zależą od obecności innych pakietów. System Debian dostarcza ten mechanizm, aby wspierać bezpieczne aktualizowanie wersji systemów z formatu `a.out` do formatu `ELF`, gdzie *porządek* w którym są rozpakowywane pakiety jest krytycznie ważny. Są też inne aktualizacje, gdzie metoda ta jest użyteczna, np. aktualizacja pakietów z priorytetem wymagane i ich zależność od `LibC`.

Tak, jak poprzednio, dokładniejsze informacje o tym można znaleźć w Packaging manual.

## 6.11 Co oznacza się przez *nieznany, instalowany, usunięty wyczyszczony i wstrzymany* w statusie pakietu?

Te flagi mówią, co użytkownik chciał zrobić z pakietem (jako wskazanie albo przez akcję użytkownika w części “Wybór” programu `dselect`, lub bezpośrednio wywołanie programu `dpkg`).

Ich znaczenie to:

- *nieznany* - użytkownik nigdy nie wskazał co chce zrobić z pakietem.

- instalowany - użytkownik chciał zainstalować lub zaktualizować wersję pakietu.
- usunięty - użytkownik chciał usunąć pakiet, ale nie chciał usuwać żadnych istniejących plików konfiguracyjnych.
- wyczyszczony - użytkownik chciał aby pakiet został usunięty całkowicie, włączając jego pliki konfiguracyjne.
- wstrzymany - użytkownik nie chce by ten pakiet był przetwarzany, np. chce zatrzymać obecną wersję z obecnym statusem bez względu na to, jaki on jest.

## 6.12 Jak można dokonać zatrzymania pakietu?

Są dwie drogi zatrzymania pakietów, używając `dpkg` lub `dselect`.

Używając `dpkg` możesz wyeksportować listę wyborów pakietów poleceniem:

```
dpkg --get-selections \* > selections.txt
```

Potem wyedytować wynikowy plik `selections.txt`, zmieniając linię zawierającą pakiet, który chcesz zatrzymać, np. `libc6`, z tego:

```
libc6                                install
```

na to:

```
libc6                                hold
```

Zachować plik i wczytać to do bazy `dpkg` poleceniem:

```
dpkg --set-selections < selections.txt
```

Używając `dselect` wystarczy wybrać ekran `[S]elect`, znaleźć pakiet, który chcesz zatrzymać w obecnym stanie i nacisnąć klawisz '=' (lub 'H'). Zmiany wejdą w życie natychmiast po wyjściu z ekranu `[S]elect`.

## 6.13 Jak mogę zainstalować pakiet źródłowy?

Pakiety źródłowe systemu Debian nie mogą być aktualnie "zainstalowane", są one po prostu rozpakowywane do katalogu, w którym chcesz zbudować pakiety binarne przez nie tworzone.

Pakiety źródłowe są rozpowszechniane w większości na tych samych serwerach lustrzanych, na których możesz uzyskać pakiety binarne. Jeśli ustawisz swoje APT `sources.list(5)` aby zawierało stosowne linie "deb-src", będziesz zdolny łatwo ściągnąć każdy pakiet źródłowy uruchamiając

```
apt-get source foo
```

Aby pomóc w tworzeniu aktualnego pakietu, Pakiet źródłowy systemu Debian dostarcza tak zwanego mechanizmu tworzenia zależności. Oznacza to, że opiekun pakietu źródłowego utrzymuje listę innych pakietów, które są wymagane aby stworzyć pakiet. Aby zobaczyć jak jest to użyteczne uruchom

```
apt-get build-dep foo
```

przed tworzeniem pakietu.

## 6.14 Jak się buduje binarne pakiety z pakietu źródłowego?

Aby skompilować źródła, będziesz potrzebował wszystkie pliki typu `foo_*.dsc`, `foo_*.tar.gz` i `foo_*.diff.gz` (zauważ: czasami nie ma plików `.diff.gz` dla pakietów, które są rodzime dla systemu Debian).

Gdy masz pakiety ('Jak mogę zainstalować pakiet źródłowy?' na poprzedniej stronie) i jeśli masz pakiet `dpkg-dev` zainstalowany, następujące polecenia:

```
dpkg-source -x foo_version-revision.dsc
```

rozpakuje pakiet do katalogu nazwanego `foo-version`.

Jeśli chcesz tylko skompilować pakiet, możesz przejść do katalogu `foo-version` i wydać polecenie

```
dpkg-buildpackage -rfakeroot -b
```

aby zbudować pakiet (zauważ że to także wymaga pakietu `fakeroot`) i potem

```
dpkg -i ../foo_version-revision_arch.deb
```

aby zainstalować nowo-zbudowany pakiet.

## 6.15 Jak mogę stworzyć pakiet systemu Debian samodzielnie?

Bardziej dokładnego opisu o tym szukaj w Instrukcji dla nowych opiekunów, dostępnej w pakiecie `maint-guide` lub na stronie <http://www.debian.org/doc/devel-manuals#maint-guide>.



## Rozdział 7

# Narzędzia do zarządzania pakietami w systemie Debian

### 7.1 Jakie programy do zarządzania swoimi pakietami udostępnia?

#### 7.1.1 dpkg

Jest to główny program służący do zarządzania pakietami. `dpkg` może być wywołany z wieloma różnymi opcjami. Najczęściej stosowane to:

- `dpkg --help` przedstawia dostępne opcje.
- `dpkg --info foo_VVV-RRR.deb` wyświetla informacje o wybranym pakiecie.
- `dpkg --install foo_VVV-RRR.deb` instaluje (rozpakowuje i konfiguruje) pakiet na dysku twardym.
- `dpkg --unpack foo_VVV-RRR.deb` rozpakowuje, lecz pozostawia nieskonfigurowany pakiet. Zauważ, że nieskonfigurowany pakiet może okazać się bezużyteczny. Niektóre pliki mogą wymagać dalszej obróbki i dostosowania do Twoich potrzeb, aby pakiet/program działał poprawnie. Polecenie to pozostawia wszelkie zainstalowane dotychczas wersje programów i wykonuje polecenia przedinstalacyjne (zobacz ‘Czym są w systemie Debian skrypty: `preinst`, `postinst`, `prerm` i `postrm`?’ na stronie 30), zawarte w pakiecie.
- Konfiguracja pakietu, który został wcześniej rozpakowany: `dpkg --configure foo`. To polecenie uruchamia, między innymi, poinstalacyjne skrypty (zobacz: ‘Czym są w systemie Debian skrypty: `preinst`, `postinst`, `prerm` i `postrm`?’ na stronie 30), powiązane z danym pakietem. Uaktualnia również pliki wymienione w pliku konfiguracyjnym danego pakietu. Zauważ, że operacja „configure” wymaga jako argumentu nazwy pakietu (np. `foo`), a nie nazwy pliku z pakietem (np. `foo_VVV-RRR.deb`).

- Rozpakowanie pojedynczego pliku nazwanego „blurf” (lub grupy plików nazwanych „blurf\*”) z pakietu Debiana: `dpkg --fsys-tarfile foo_VVV-RRR.deb | tar -xf - blurf*`
- Usunięcie pakietu (bez jego plików konfiguracyjnych): `dpkg --remove foo`.
- Usunięcie pakietu (razem z jego plikami konfiguracyjnymi): `dpkg --purge foo`.
- Wyświetlenie stanu instalacji pakietów zawierających ciąg znaków (lub wg wyrażenia regularnego) „foo\*”: `dpkg --list 'foo*'`.

### 7.1.2 dselect

Ten program zapewnia Debianowi interfejs dla systemu zarządzania pakietami z opcjami pogrupowanymi w menu. Jest on szczególnie przydatny przy instalowaniu i uaktualnianiu systemu na dużą skalę.

`dselect` potrafi:

- prowadzić/wspomagać użytkownika w trakcie, gdy ten wybiera pakiety do instalacji lub usunięcia, wykluczając jednocześnie możliwość powstania konfliktów pomiędzy zainstalowanymi pakietami i zapewniając właściwą pracę zainstalowanych programów poprzez instalację wszystkich wymaganych przez nie pakietów.
- ostrzegać użytkownika o niespójności lub niekompatybilności pomiędzy pakietami, które wybrał do zainstalowania;
- ustalać w jakiej kolejności pakiety powinny zostać zainstalowane;
- automatycznie przeprowadzać proces instalacji lub usunięcia, oraz
- prowadzić użytkownika przez proces wymaganej dla poszczególnych pakietów konfiguracji.

`dselect` przedstawia użytkownikowi na początek menu złożone z 7 opcji/elementów, z których każdy/a reprezentuje inny aspekt funkcjonalności programu. Użytkownik może wybrać jedną z opcji używając klawiszy strzałek aby podświetlić wybraną opcję i zatwierdzić wybór naciskając klawisz/przycisk `<enter>`.

Dalsze zachowanie programu zależy od dokonanego wcześniej wyboru. Jeżeli nie wybrano `Dostęp/Access` ani `Wybór/Select`, wtedy `dselect` po prostu wykona wybrane polecenie: np. jeżeli użytkownik wybrał polecenie `Usuwanie/Remove` wtedy `dselect` przystąpi do usuwania wszystkich plików wybranych/zaznaczonych ostatnio do usunięcia przy pomocy polecenia `Wybór/Select`.

Zarówno opcja `Dostęp/Access`, jak i opcja `Wybór/Select`, prowadzą do dodatkowego menu wyboru. W obydwu przypadkach, każde menu zostaje wyświetlone osobno, na podzielonym ekranie; w górnej części ekranu znajdują się opcje wyboru, a w dolnej części krótkie opisy (“info”) dla każdej z wybranych opcji.

Dostępna jest także poszerzona/rozszerzona dokumentacja online, użyj klawisza '?' aby uzyskać dostęp do pomocy w dowolnej chwili.

Kolejność, w jakiej czynności są prezentowane w pierwszym menu programu `dselect`, reprezentuje kolejność w jakiej użytkownik normalnie dokonuje instalacji pakietów. Jednakże można wybrać dowolną pozycję z menu, tak często, jak to konieczne.

- Rozpoczynamy od wybrania **sposobu dostępu**. (ang. „Access Method”). Jest to czynność, dzięki której można zaplanować dostęp do źródła pakietów Debiana. Przykładowo, niektórzy mają pakiety na płytach CD, a inni pobierają je z serwera FTP. Wybrany sposób jest zapamiętywany także po wyjściu z programu, więc jeśli nie zmieniamy źródła od poprzedniego razu, nie jest konieczne ponowne wywoływanie tej opcji.
- Teraz **uaktualnij** (Update) listę dostępnych pakietów. Aby tego dokonać, `dselect` odczytuje plik „Packages.gz”, który powinien znajdować się w głównym katalogu zawierającym przeznaczone do instalacji pakiety Debiana. (Jeżeli brak jest takiego pliku, `dselect` utworzy go dla Ciebie).
- **Wybór** (Select) określa pakiety do zainstalowania w tym systemie. Po wybraniu tej pozycji menu, użytkownik jest najpierw zapoznawany z pełnym zestawem pomocy (chyba że zastosowano opcję ‘-expert’ w linii poleceń). Gdy opuszczamy ekran z pomocą, pojawia się ekran umożliwiający wybór pakietów do instalacji lub usunięcia.

Górna część ekranu zawiera przewijaną listę pakietów poukładanych w grupy związane ze sobą. Dolna część zawiera opis pakietu lub grupy pakietów aktualnie podświetlonych. Podświetlając nazwę pakietu lub grupę pakietów, wskazujemy stan, który mamy zamiar zmieniać. Następnie możesz zaznaczyć pakiet:

**do instalacji:** Dokonuje się tego przy pomocy klawisza ‘+’.

**do usunięcia:** Pakiety mogą być usuwane dwoma sposobami:

- usunięcie (remove): usuwa większość plików związanych z tym pakietem. Pozostawia jednak pliki, określone jako konfiguracyjne (zobacz: ‘Czym jest plik `conf`file w systemie Debian?’ na stronie 30) oraz konfigurację pakietu. Dokonuje się tego klawiszem ‘-’.
- oczyszczenie (purge): usuwa *wszystkie* pliki wchodzące w skład tego pakietu. Służy do tego klawisz ‘\_’.

Zauważ, że nie jest możliwe usunięcie grupy „Wszystkie pakiety” (“All Packages”). Gdy tego spróbujesz, system zostanie zredukowany do początkowego stanu po instalacji wersji podstawowej (base system)

**ustawienie „w zawieszeniu” („on hold”)** Dokonuje się tego klawiszem ‘=’ i oznacza to „nie uaktualniaj, nawet jeśli aktualnie zainstalowana wersja nie jest tak świeża, jak wersja ze źródła pakietów, z jakiego korzystasz (jaki zostało skonfigurowane przy pomocy opcji „Sposób dostępu”), pobrana po wyborze opcji **Uaktualnij** (Update)). Tak, jak możesz zawiesić uaktualnianie wersji danych pakietów, tak możesz anulować zawieszenie poprzez naciśnięcie ‘:’. Oznacza to zezwolenie na uaktualnianie wersji pakietów, jeśli nowsze są dostępne. Jest to domyślne ustawienie dla pakietu.

Możesz wybrać różne sposoby prezentacji listy pakietów, używając klawisza „o” do przełączania pomiędzy różnymi metodami sortowania pakietów. Domyślne sortowanie przeprowadzane jest według parametru Priorytet. Wewnątrz tego samego priorytetu, pakiety są poukładane według katalogów (zwanymi też sekcjami) archiwum, w których są przechowywane. W tym trybie, niektóre pakiety z sekcji (powiedzmy) A mogą być umieszczane jako pierwsze, następnie niektóre pakiety z sekcji (powiedzmy) B, zaś po nich znów pakiety z sekcji A, ale posiadające niższy priorytet.

Możesz również poszerzyć znaczenie etykiet na górze ekranu, wykorzystując klawisz „v” (ang. verbose = rozwlekły). Ta metoda przesuwą dużą część widocznych poprzednio informacji poza prawą granicę ekranu. Aby je zobaczyć, naciśnij strzałkę w prawo, aby powrócić na lewo, możesz skorzystać z klawisza strzałki w lewo.

Gdy zaznaczysz pakiet do instalacji lub do usunięcia, przykładowo `foo.deb`, i ten pakiet jest powiązany zależnościami z innym pakietem, przykładowo `blurf.deb`, to `dselect` wyświetli odpowiednie okno, w którym możesz wybrać spośród powiązanych pakietów, akceptując sugerowaną akcję (instalować lub nie instalować, oto jest pytanie) lub odrzucić je całkowicie. Tą drugą czynność przeprowadzamy klawiszami `Shift+D`. Natomiast, aby powrócić do poprzedniego wyboru, naciśnij `Shift+U`. W każdym razie, możesz zapisać swój wybór i powrócić do głównego ekranu kombinacją klawiszy `Shift+Q`.

- Można wybrać menu „Instaluj” („Install”) aby rozpakować i skonfigurować zaznaczone pakiety lub usunąć pobrane pliki poleceniem „Usuń” („Remove”). Można także opuścić program wybierając polecenie „Wyjdź” („Quit”).

### 7.1.3 dpkg-deb

Ten program służy do pracy z plikami-archiwami Debiana (\*.deb). Kilka najczęstszych zastosowań to:

- Pokaż wszystkie opcje: `dpkg-deb --help`.
- Pokaż jakie pliki znajdują się w podanym archiwum: `dpkg-deb --contents foo_VVV-RRR.deb`
- Wypakuj pliki zawarte w podanym archiwum do zadanego katalogu: `dpkg-deb --extract foo_VVV-RRR.deb KATALOG` Polecenie wypakuje pliki z `foo_VVV-RRR.deb` do katalogu `KATALOG/`. To wygodny sposób na zbadanie zawartości paczki bez jej instalacji do głównego systemu plików.

Zauważ, że dowolne pakiety, które były po prostu rozpakowane przy pomocy `dpkg-deb --extract` nie zostały poprawnie zainstalowane. W celu instalacji należy użyć polecenia `dpkg --install`.

Więcej informacji na stronach podręcznika systemowego: `dpkg-deb(1)`.

### 7.1.4 apt-get

`apt-get` umożliwia prostą instalację pakietów z linii poleceń. W przeciwieństwie do `dpkg`, `apt-get` nie obsługuje nazw plików `.deb`, ale pracuje w oparciu o właściwe nazwy pakietów. Korzysta ze źródeł pakietów wymienionych w `/etc/apt/sources.list`.

Jeśli potrzebujesz więcej informacji, zainstaluj pakiet `apt` i przeczytaj strony podręcznika systemowego: `apt-get(8)`, `sources.list(5)` oraz opis `/usr/share/doc/apt/guide.html/index.html`.

### 7.1.5 dpkg-split

Ten program dzieli duże pakiety na małe pliki (na przykład, w celu umieszczenia na dyskietkach – kto teraz używa dyskietek :-)) oraz scala takie podzielone pliki w jeden. Może on być jedynie wykorzystany w systemie opartym o Debiana (to jest takim, który zawiera pakiet `dpkg`), ponieważ uruchamia on inny program, `dpkg-deb`, do rozbioru pliku pakietu Debiana na jego rekordy składowe.

Dla przykładu, chcąc podzielić duży plik `.deb` na `N` kawałków:

- Wykonaj polecenie: `dpkg-split --split foo.deb`. Wyprodukuje ono `N` plików o przybliżonej wielkości 460 KBytes w aktualnym katalogu.
- Skopiuj te `N` plików na dyskietki. I idź do innego Debiana, albo pochodź wokoło stołu i udawaj, że jesteś przy innym komputerze ;-).
- Skopiuj zawartość dyskietek na twardy dysk na innej maszynie
- Połącz kawałki używając polecenia `dpkg-split --join "foo*"`.

## 7.2 Debian twierdzi, że można uaktualniać uruchomiony program; Jak to możliwe?

Jądro (system plików) w Debianie umożliwia podmianę plików, nawet jeśli są używane.

Dostarczamy także program o nazwie `start-stop-daemon`, który jest używany do uruchamiania demonów w czasie startu systemu oraz do ich zatrzymywania, podczas gdy poziom startu (`runlevel`) jest zmieniany. Ten sam program jest używany przez skrypty instalacyjne podczas instalacji nowego pakietu zawierającego demona.

## 7.3 Jak mogę sprawdzić, które pakiety są już zainstalowane w Debianie?

Aby poznać stan wszystkich pakietów zainstalowanych w systemie, wykonaj polecenie:

```
dpkg --list
```

Wypisze ono jednolinijkowe podsumowanie dla każdego pakietu, zawierające dwuznakowe oznaczenie statusu (wyjaśnione w nagłówku), nazwę pakietu, wersję, która jest *zainstalowana*, oraz krótki opis.

Aby poznać stan pakietów, których nazwy zawierają określony ciąg znaków, możesz wykonać polecenie (to polecenie pokazuje pakiety z nazwami rozpoczynającymi się od „foo”):

```
dpkg --list 'foo*'
```

Aby otrzymać obszerniejsze informacje na temat określonego pakietu, wykonaj:

```
dpkg --status nazwa_pakietu
```

## 7.4 Jak mam stwierdzić, który pakiet tworzy określony plik?

Aby zidentyfikować pakiet odpowiedzialny za pojawienie się pliku `nazwa_pliku` wykonaj:

- `dpkg --search nazwa_pliku`

To polecenie spowoduje wyszukanie pliku o nazwie `nazwa_pliku` w zainstalowanych paczkach. (Jest to ekwiwalent poszukiwania wszystkich plików mających rozszerzenie `.list` i znajdujących się w katalogu `/var/lib/dpkg/info/`, a następnie wypisywania nazw wszystkich pakietów zawierających te pliki i odwrotnie).

- `zgrep foo Contents-ARCH.gz`

To polecenie poszukuje plików zawierających tekst `foo` w ich pełnej ścieżce dostępu. Pliki `Contents-ARCH.gz` (gdzie `ARCH` oznacza właściwą architekturę), znajdują się w głównym katalogu z pakietami (`main`, `non-free`, `contrib`), na serwerze FTP Debiana. Pliki `Contents` odnoszą się tylko do pakietów z drzewa pakietów, do którego należą. Z tego powodu użytkownik może będzie musiał przeszukać więcej niż jeden plik `Contents`, aby znaleźć pakiet zawierający plik `foo`.

Ta metoda ma tą przewagę nad `dpkg --search`, że znajduje dowolne pliki w dowolnych pakietach, nie tylko aktualnie zainstalowane w systemie.

## Rozdział 8

# Aktualizowanie systemu Debian

Celem Debiana jest udostępnianie spójnych sposobów uaktualnień oraz bezpiecznego procesu unowocześniania wersji. Robimy wszystko, co w naszej mocy, aby zapewnić łagodny system uaktualniania. Jeśli podczas aktualizacji pakietu użytkownik musi zostać o czymś poinformowany, zostanie to uczynione, a często także zostanie poinstruowany w jaki sposób rozwiązać ewentualne problemy.

Należy czytać notatki do wydań (Release Notes), które opisują szczegóły uaktualnień i które są dostępne na wszystkich płytach CD Debiana oraz pod adresem: <http://www.debian.org/releases/stable/releasenotes>.

### 8.1 Jak mogę uaktualnić mojego Debiana 1.3.1 (lub wcześniejszego), opartego na libc5, do wersji 2.0 (lub późniejszej), opartej na libc6?

Istnieje kilka dróg takiego uaktualnienia:

- Używając prostego skryptu powłoki nazwanego `autoup.sh`, który uaktualnia najważniejsze pakiety. Po tym jak skończy on swoją pracę, możesz użyć programu `dselect` do masowej instalacji wymaganych pakietów. Jest to rekomendowana, aczkolwiek nie jedyna metoda.

Obecnie najnowsze wydanie skryptu `autoup.sh` może być znalezione w następujących miejscach:

- <http://www.debian.org/releases/2.0/autoup/>
- <http://www.taz.net.au/autoup/>
- <http://debian.vicnet.net.au/autoup/>

- Podążając za radami z dokumentu: Debian libc5 to libc6 Mini-HOWTO (<http://debian.vicnet.net.au/autoup/HOWTO/libc5-libc6-Mini-HOWTO.html>)

uaktualniać najważniejsze pakiety ręcznie. Skrypt `autoup.sh` bazuje na tym właśnie Mini-HOWTO, więc ta metoda działa mniej więcej tak, jak użycie `autoup.sh`.

- Używając opartego na `libc5` programu `apt`. APT oznacza A Package Tool i pewnego dnia zastąpi wysłużony program `dselect`. Obecnie `apt` działa z linii poleceń lub jako jedna z metod dostępu `dselecta`. Wersję bazującą na `libc5` znajdziesz w katalogu `dists/slink/main/upgrade-older-i386` w archiwum Debiana.
- Używając programu `dselect` bez uaktualniania wcześniej żadnych pakietów ręcznie. Stanowczo NIE polecamy tej metody jeśli możesz jej uniknąć, ponieważ `dselect` nie instaluje pakietów w optymalnej kolejności. APT działa o wiele lepiej i jest bezpieczniejszy.

## 8.2 Jak mogę utrzymać ciągle aktualną wersję Debiana?

Można po prostu połączyć się anonimowo przez FTP z serwerem zawierającym archiwum Debiana, uważnie przyjrzeć się katalogom, znaleźć pożądane pliki, pobrać je i zainstalować z użyciem `dpkg`. `dpkg` instaluje uaktualnienia na żywo, nawet na pracującym systemie. Czasami poprawione pakiety wymagają instalacji innych, również poprawionych wersji pakietów powiązanych i wtedy instalacja się nie uda, póki owe pakiety nie zostaną zainstalowane.

Wielu użytkowników uzna takie podejście za zbyt czasochłonne, zwłaszcza że Debian rozwija się szybko i typowo około tuzin lub więcej pakietów jest uaktualnianych tygodniowo. Ta ilość jest o wiele większa przed wydaniem kolejnej głównej wersji. Aby poradzić sobie z taką lawiną, użytkownicy zazwyczaj wolą używać bardziej zautomatyzowanych metod.

Oto kilka metod automatyzacji:

### 8.2.1 APT

APT to zaawansowany interfejs do systemu pakietów Debiana. `apt-get` to oparte o linię poleceń narzędzie do obsługi pakietów, a metoda APT `dselecta` to interfejs do APT z programu `dselect`. Obydwa z nich udostępniają prostszy i bezpieczniejszy sposób instalacji oraz uaktualniania pakietów.

APT potrafi ustalać kolejność instalacji, pobierać dane z wielu źródeł jak również wiele innych, unikalnych rzeczy, o których możesz przeczytać w `/usr/share/doc/apt/guide.html/index.html`.

Zainstaluj pakiet `apt` i zmodyfikuj plik `/etc/apt/sources.list`. Jeśli chcesz używać jako źródła pakietów najnowszej, stabilnej wersji Debiana, możesz użyć np.:

```
http://http.us.debian.org/debian stable main contrib non-free
```

Możesz również zastąpić `http.us.debian.org` nazwą szybszego serwera lustrzanego dostępnego z Twojej lokalizacji. Listę serwerów znajdziesz tutaj: <http://www.debian.org/misc/README.mirrors>.

Więcej informacji na ten temat znajdziesz w podręczniku systemowym: `apt-get(8)` oraz `sources.list(8)` a także w przewodniku `/usr/share/doc/apt/guide.html/index.html`.

Uruchom:

```
apt-get update
```

a następnie:

```
apt-get dist-upgrade
```

Odpowiedz na wszystkie ewentualne pytania, zadane przez system i Twój Debian zostanie uaktualniony.

Aby używać systemu APT z programem `dselect`, wybierz metodę dostępu APT w głównym ekranie programu `dselect` (opcja 0) i określ źródła pakietów, jakie mają być używane. Plik konfiguracyjny to `/etc/apt/sources.list`, a jego format opisany jest na stronach podręcznika systemowego: `sources.list(5)`.

Jeśli chcesz wykorzystać płyty CD do instalacji pakietów, możesz użyć programu `apt-cdrom`. Szczegóły znajdziesz w notatkach z wydań (Release Notes) w sekcji „Uaktualnienia z lokalnych serwerów lustrzanych” (“Setting up for an upgrade from a local mirror”).

Pamiętaj, że po pobraniu i zainstalowaniu pakietów są one nadal przechowywane w hierarchii katalogu `/var`. Aby nie dopuścić do zapełnienia partycji, usuwaj zbędne pliki przy użyciu poleceń `apt-get clean` i `apt-get autoclean` lub przenieś je w inne miejsce (np. programem `apt-move`).

## 8.2.2 dpkg-ftp

Jest to starsza metoda wykorzystywana przez program `dselect`. Może zostać wywołana z `dselecta`, przez co pozwala na pobranie plików oraz ich zainstalowanie w jednym kroku. Aby tego dokonać, wybierz metodę `ftp` dostępu do źródła pakietów w programie `dselect` (opcja 0) i określ nazwę odległej maszyny oraz katalog. `dpkg-ftp` automatycznie pobierze pliki, które są wybrane (w tej lub poprzednich sesjach `dselecta`).

Zauważ, że w przeciwieństwie do programu `mirror`, `dpkg-ftp` nie pobiera wszystkiego z serwera, a jedynie tylko te pliki, które zostały zaznaczone (podczas uruchomienia `dpkg-ftp`) i które wymagają uaktualnienia.

`dpkg-ftp` jest nieco przestarzały. Należy raczej używać systemu APT z odnośnikami do źródeł na FTP (`ftp://`) w pliku `sources.list`.

### 8.2.3 mirror

Ten skrypt napisany w Perlu oraz jego (opcjonalny) dodatek o nazwie `mirror-master` mogą być używane do pobrania określonej części drzewa katalogów z wybranego serwera poprzez anonimowy dostęp do FTP.

`mirror` jest szczególnie użyteczny do pobierania dużych zbiorów oprogramowania. Po pierwszym załadowaniu plików, na lokalnym komputerze powstaje plik `.mirrorinfo`. Zmiany w odległym systemie plików są śledzone automatycznie przez `mirror`, który porównuje ten plik z analogicznym po stronie odległego systemu i pobiera tylko zmienione elementy.

Program `mirror` jest zazwyczaj używany do uaktualniania lokalnej kopii umieszczonego na odległym systemie drzewa katalogów. Pobierane pliki nie muszą być plikami Debiana. (Ponieważ `mirror` jest skryptem Perla, może być uruchamiany również na nieuniksowych systemach). Chociaż `mirror` udostępnia mechanizmy pozwalające na wyłączenie plików, których nazwy pasują do określonych wzorców, to jednak program jest najbardziej użyteczny, gdy jego celem jest pobieranie całych fragmentów drzewa katalogów, a nie wybranych pakietów.

### 8.2.4 dpkg-mountable

`dpkg-mountable` dodaje nową metodę dostępu do listy `dselect` - tzw. 'mountable'. Pozwala ona na instalację pakietów z dowolnego systemu plików podanego w `/etc/fstab`. Może to być na przykład zwykła partycja na dysku lub serwerze NFS, która w razie potrzeby zostanie automatycznie zamontowana i odmontowana.

Obsługuje także parę rzeczy niedostępnych w zwykłym `dselect`, takich jak obsługa lokalnych sytemów katalogów (zarówno równoległych do głównego, dystrybucyjnego drzewa, jak i dla zupełnie oddzielnych), ściąganie jedynie tych pakietów, które są wymagane, zamiast czasochłonnego, rekurencyjnego przeszukiwania katalogów, jak również rejestrowanie wszystkich czynności wykonanych przez `dpkg` w fazie instalacji.

## 8.3 Czy muszę przechodzić w tryb pojedynczego użytkownika, by móc zaktualizować pakiety?

Nie. Pakiety mogą być uaktualniane nawet w trakcie normalnej pracy systemu. Debian zawiera program `start-stop-daemon`, który jest wykonywany w celu zatrzymania, a później ponownego uruchomienia działających procesów, jeśli to konieczne, podczas uaktualniania pakietu.

## 8.4 Czy muszę trzymać te wszystkie pliki .deb na swoim dysku?

Nie. Jeśli masz na dysku pobrane pliki (takie, które nie są konieczne - zobacz opis `dpkg-ftp` powyżej), to po zainstalowaniu pakietów, możesz usunąć te pliki z systemu.

## 8.5 Jak mogę otrzymać zapiski z czynności dodawania pakietów do systemu?

`dpkg` notuje jakie pakiety były rozpakowywane, konfigurowane, usuwane, dodawane lub przeczyszczane, jednakże nie przechowuje (obecnie) informacji o działaniach użytkownika na terminalu podczas operacji na pakiecie.

Najlepszym obejściem tego problemu jest uruchomienie `dpkg/dselect/apt-get/cokolwiek` wewnątrz sesji programu `script (1)`.



## Rozdział 9

# Debian i jądro systemu

### 9.1 Czy mogę skompilować i zainstalować jądro systemu bez dostosowywania go do specyfiki Debiana?

Tak.

Z jednym tylko zastrzeżeniem: biblioteki języka C w systemie Debian są budowane z wykorzystaniem najnowszych *stabilnych* wersji plików nagłówkowych **jądra systemu**. Jeśli chcesz skompilować program z wykorzystaniem nowszych plików nagłówkowych jądra niż te ze stabilnej wersji, stoisz przed wyborem pomiędzy uaktualnieniem pakietu zawierającego pliki nagłówkowe (`libc6-dev`), a użyciem nowych plików nagłówkowych, które znajdują się w katalogu z rozpakowanymi plikami źródłowymi jądra. Jeżeli pliki źródłowe jądra znajdują się w katalogu `/usr/src/linux/`, wtedy przy kompilowaniu do polecenia musisz dodać opcję `-I/usr/src/linux/include/`.

### 9.2 Jakich narzędzi dostarcza system Debian w celu budowania własnych wersji jądra systemu?

Użytkownicy którzy pragną zbudować własną wersję jądra systemu mogą pobrać i zainstalować pakiet `kernel-package`. Zawiera on skrypt wykonujący kompilację i umożliwia stworzenie pakietu Debiana z gotowym jądrem przy pomocy polecenia

```
make-kpkg kernel_image
```

, które należy wpisać znajdując się w katalogu z plikami źródłowymi jądra. Więcej informacji można uzyskać przy pomocy polecenia

```
make-kpkg --help
```

lub podręcznika systemowego `make-kpkg(1)`.

Ponieważ opisany pakiet nie zawiera plików źródłowych jądra, użytkownicy muszą pobrać je oddzielnie z wybranego przez siebie serwera lub w postaci pakietu Debiana jeśli pakiet `kernel-source`-wersja jest dostępny (gdzie wersja to numer pożądanej wersji jądra).

Szczegółowe instrukcje dotyczące korzystania z pakietu `kernel-package` można znaleźć w pliku `/usr/share/doc/kernel-package/README.gz`. W skrócie powinno się:

- Rozpakować pliki źródłowe jądra i przenieść się do nowo powstałego katalogu przy pomocy polecenia `cd`.
- Dostosować konfigurację jądra do własnych potrzeb przy pomocy jednego z poniższych poleceń:
  - `make config` (interfejs tekstowy zadający kolejno pytania).
  - `make menuconfig` (interfejs tekstowy wykorzystujący bibliotekę `ncurses` z opcjami pogrupowanymi w menu). Aby móc korzystać z tego polecenia musisz mieć zainstalowany pakiet `libncurses5-dev`.
  - `make xconfig` (interfejs graficzny X11). Korzystanie z tej opcji wymaga zainstalowania istotnych pakietów związanych z X11 i Tcl/Tk.

Wykorzystanie któregokolwiek z powyższych poleceń spowoduje utworzenie nowego pliku `.config` w katalogu z plikami źródłowymi.

- Wprowadzić polecenie: `make-kpkg -rev TwojaNazwa.N kernel_image`, przy czym `N` oznacza nadany przez Ciebie numer. Stworzony w wyniku tak przeprowadzonej kompilacji pakiet z jądrem będzie posiadał unikatowy stworzony przez Ciebie symbol wersji `TwojaNazwa.1`, np.: `kernel-image-2.2.14_TwojaNazwa.1_i386.deb` dla jądra w wersji 2.2.14.
- Zainstalować stworzony w ten sposób pakiet.
  - Uruchom `dpkg --install /usr/src/kernel-image-VVV_Custom.N.deb` aby zainstalować jądro. Skrypt instalacyjny:
    - \* uruchomi program ładujący system, LILO (jeśli ten jest zainstalowany),
    - \* zainstaluje skompilowane przez Ciebie jądro w katalogu `/boot` pod nazwą `vmlinuz_VVV-TwojaNazwa.N` i stworzy odpowiednie dowiązania symboliczne.
    - \* zapyta użytkownika czy wykonać dyskietkę startową. Dyskietka taka zawiera jedynie plik z jądrem. Aby dowiedzieć się więcej zobacz 'W jaki sposób mogę wykonać własną wersję dyskietki startowej?' na następnej stronie.
  - Aby wykorzystać jeden z pozostałych programów ładujących system takich jak `grub` czy `loadlin`, skopiuj ten plik w inne miejsce (np. do katalogu `/boot/grub` lub na partycję z systemem plików MS-DOS).

### 9.3 W jaki sposób mogę wykonać własną wersję dyskietki startowej?

Do tego celu możesz z powodzeniem posłużyć się pakietem `boot-floppies`, który znajduje się w sekcji `admin` zasobów FTP Debiana. Skrypt powłoki znajdujący się w tym pakiecie tworzy dyskietki startowe w formacie `SYSLINUX`. Są to dyskietki z systemem plików `MS-DOS`, których główny rekord startowy został zmodyfikowany tak aby bezpośrednio ładował jądro Linuksa (lub innego systemu w zależności od definicji zawartej w pliku `syslinux.cfg` znajdującym się na dyskietce). Pozostałe skrypty w tym pakiecie służą do tworzenia dyskietek awaryjnych lub instalacyjnych.

Więcej informacji znajdziesz w pliku `/usr/doc/boot-floppies/README` po zainstalowaniu pakietu `boot-floppies`.

### 9.4 Jakie narzędzia udostępnia Debian do obsługi modułów?

Pakiet Debiana `modconf` udostępnia skrypt powłoki (`/usr/sbin/modconf`), który może być wykorzystywany do dostosowywania konfiguracji modułów dla własnych potrzeb. Skrypt ten oferuje interfejs oparty o menu, który pyta użytkownika o szczegóły dotyczące modułów sterowników urządzeń w systemie. Uzyskane odpowiedzi służą do stworzenia odpowiednich wpisów w pliku `/etc/modules.conf` (który zawiera listę aliasów/skrótów i parametry które zostaną użyte przy ładowaniu modułów). Są również konieczne do stworzenia plików znajdujących się w katalogu `/etc/modutils/` oraz zawartości pliku `/etc/modules` (który zawiera listę modułów ładowanych w czasie startu systemu).

Podobnie jak (nowe) pliki `Configure.help`, które teraz pomagają przy tworzeniu własnych wersji jądra, pakiet `modconf` zawiera pliki pomocy (znajdujące się w katalogu `/usr/lib/modules_help/`), które dostarczają szczegółowych informacji na temat parametrów właściwych dla poszczególnych modułów.

### 9.5 Czy mogę bezpiecznie odinstalować pakiet ze starym jądrem?

Tak. Skrypt `kernel-image-NNN.prerm` sprawdza, czy jądro które chcesz usunąć nie jest tym, którego właśnie używasz. W każdym przypadku możesz usunąć pakiety z niechcianym jądrem używając polecenia

```
dpkg --purge --force-remove-essential kernel-image-VVV
```

(zastępując oczywiście "VVV" numerem wersji jądra)



## Rozdział 10

# Dostosowanie systemu Debian GNU/Linux do Twoich potrzeb

### 10.1 Skąd mogę mieć pewność, że wszystkie programy używają tego samego rozmiaru papieru?

Zainstaluj pakiet `libpaperg`. Podczas instalacji zostaniesz zapytany o domyślny rozmiar papieru, który ma być używany w całym systemie. Ustawienie to będzie przechowywane w pliku `/etc/papersize`.

Użytkownicy mogą ustalić własny rozmiar papieru używając zmiennej systemowej `PAPERSIZE`. Więcej informacji uzyskasz na stronie podręcznika systemowego `papersize(5)`.

### 10.2 Jak mogę udostępnić urządzenia peryferyjne bez narażania bezpieczeństwa systemu?

Wiele plików urządzeń w katalogu `/dev/` należy do pewnych, wcześniej zdefiniowanych, grup. Na przykład `/dev/fd0` należy do grupy `floppy`, a `/dev/dsp` należy do grupy `audio`.

Jeśli chcesz żeby dany użytkownik miał dostęp do jednego z tych urządzeń to po prostu dodaj go do grupy, do której należy to urządzenie. Możesz to zrobić na przykład tak:

```
adduser użytkownik grupa
```

W ten sposób nie będziesz musiał zmieniać praw dostępu urządzenia.

### 10.3 W jaki sposób załadować czcionkę konsoli podczas startu systemu?

Pakiety `kbd` oraz `console-tools` pozwalają to osiągnąć. Dostosuj odpowiednio do swoich potrzeb plik `/etc/kbd/config` lub `/etc/console-tools/config`.

### 10.4 W jaki sposób skonfigurować domyślne parametry aplikacji dla środowiska X11?

Programy Debiana pod X'y instalują swoje zasoby do katalogu `/etc/X11/app-defaults/`. Jeśli chcesz dostosować aplikacje X'ów globalnie to dokonaj zmiany w tych plikach. Oznaczone są one jako konfiguracyjne więc ich zawartość pozostanie niezmieniona podczas procesu aktualizacji.

### 10.5 Wydaje się, że każda dystrybucja posiada inne procedury startu systemu. Jak to wygląda w Debianie?

Jak wszystkie Uniksy, Debian startuje uruchamiając program `init`. Plik konfiguracyjny dla `init` (tj. `/etc/inittab`) określa, że pierwszym wykonywanym skrypcem powinien być `/etc/init.d/rcS`. Skrypt ten uruchamia wszystkie skrypty z katalogu `/etc/rcS.d/` poprzez bezpośrednie interpretowanie lub uruchamianie podprocesów interpretujących (zależnie od rozszerzeń plików) w celu dokonania inicjalizacji, takiej jak: sprawdzenie oraz zamontowanie systemów plików, załadowanie modułów, uruchomienie usług sieciowych, ustawienie zegara i innych. W dalszej kolejności, w celu utrzymania zgodności ze starszymi wersjami, uruchamia również skrypty (oprócz tych ze znakiem `'.'` w nazwach) z katalogu `/etc/rc.boot/`. Wszystkie skrypty w tym katalogu są zazwyczaj zarezerwowane do użytku administratora systemu. Używanie ich jest przestarzałą praktyką.

Po zakończeniu procedury startu, `init` wykonuje wszystkie skrypty startowe z katalogu zależnego od domyślnego poziomu startu (domyślny poziom startu określany jest przez wpis `id` w pliku `/etc/inittab`). Jak większość Uniksów zgodnych z System V, Linux ma 7 poziomów startu (runlevels):

- 0 (zatrzymanie systemu),
- 1 (tryb pojedynczego użytkownika),
- 2 do 5 (różne tryby wielodostępowe) i
- 6 (ponowne uruchomienie systemu).

Systemy oparte na Debianie posiadają domyślny wpis `id=2` co oznacza, że w momencie wejścia w stan pracy wielodostępowej, domyślnym poziomem startu będzie `'2'` oraz, że zostaną uruchomione skrypty z katalogu `/etc/rc2.d`.

Tak naprawdę skrypty, w każdym z katalogów `/etc/rcN.d/`, są tylko symbolicznymi dowiązaniem do skryptów z `/etc/init.d`. Jednak *nazwy* plików, z katalogów `/etc/rcN.d/`, są tak nadane aby pokazać *sposób* w jaki skrypty z `/etc/init.d/` będą uruchamiane. Przed wejściem do dowolnego poziomu startu wszystkie skrypty zaczynające się literą 'K' zostaną uruchomione. Te skrypty usuwają usługi. Następnie wykonane zostaną wszystkie skrypty zaczynające się literą 'S'. Te skrypty uruchamiają usługi. Dwucyfrowa liczba, która występuje po 'K' lub 'S' oznacza kolejność, w jakiej uruchamiają się skrypty. Mniejsza liczba oznacza, że skrypt uruchomi się wcześniej.

To podejście działa, ponieważ wszystkie skrypty z `/etc/init.d/` pobierają jeden z argumentów: 'start', 'stop', 'reload', 'restart' lub 'force-reload' i wykonają zadanie określone przez ten argument. Skrypty te mogą być używane nawet po zakończeniu rozruchu systemu w celu kontroli różnych procesów.

Na przykład, z argumentem 'reload', polecenie:

```
/etc/init.d/sendmail reload
```

wyśle sygnał demonowi sendmail'a aby ponownie przeczytał i zinterpretował swój plik konfiguracyjny.

## 10.6 Wygląda na to że Debian nie używa `rc.local` aby dostosować proces startu systemu. Jakie narzędzia zostały dostarczone do tego celu?

Przypuśćmy, że system ma uruchomić skrypt `foo` podczas startu systemu lub podczas przejścia w dany poziom startu (runlevel). W takiej sytuacji administrator systemu powinien:

- Umieścić skrypt `foo` w katalogu `/etc/init.d/`.
- Uruchomić polecenie Debiana `update-rc.d` z odpowiednimi parametrami w celu ustanowienia dowiązań pomiędzy (określonymi w linii poleceń) katalogami `rc?.d`, a `/etc/init.d/foo`. W tym przypadku '?' jest cyfrą od 0 do 6 i oznacza odpowiedni poziom startu Systemu V.
- Ponownie uruchomić system.

Polecenie `update-rc.d` ustanowi dowiązania w katalogach `rc?.d` ze skryptem w `/etc/init.d`. Każde dowiązanie składać się będzie, w kolejności: z litery 'S' lub 'K', dwucyfrowej liczby oraz nazwy skryptu. Skrypty zaczynające się literą 'S' w `/etc/rcN.d/` zostaną wykonane przy przejściu do poziomu startu N. Skrypty zaczynające się literą 'K' zostaną wykonane przy wyjściu z poziomu startu N.

Można, na przykład, sprawić by skrypt `foo` wykonał się podczas startu systemu poprzez umieszczenie go w katalogu `/etc/init.d/` oraz utworzenie dowiązań przy pomocy polecenia `update-rc.d foo defaults 19`. Parametr 'defaults' oznacza domyślne poziomy

startu tzn. od 2 do 5. Parametr '19' daje pewność, że `foo` zostanie uruchomiony wcześniej niż skrypty o numerach 20 i wyższych.

## 10.7 Jak system zarządzania pakietami radzi sobie z pakietami, zawierającymi pliki konfiguracyjne innych pakietów?

Niektórzy użytkownicy chcieliby, na przykład, stworzyć nowy serwer instalując grupę pakietów Debiana i lokalnie wygenerowane pakiety zawierające pliki konfiguracyjne. Zazwyczaj nie jest to dobry pomysł ponieważ program `dpkg` nie będzie wiedział o istnieniu plików konfiguracyjnych jeśli znajdują się one w innych pakietach. Może to doprowadzić do nadpisania konfliktowych plików gdy jeden z pakietów oryginalnej „grupy” zostanie uaktualniony.

Zamiast tego utwórz lokalny pakiet, który modyfikuje pliki konfiguracyjne „grupy” pakietów Debiana, które Cię interesują. Wtedy `dpkg` i reszta systemu zarządzania pakietami będzie wiedział, że pliki zostały zmodyfikowane przez lokalnego administratora i nie będzie próbował ich nadpisać w czasie aktualizacji.

## 10.8 Jak mogę nadpisać plik instalowany przez inny pakiet tak żeby używana była moja wersja?

Powiedzmy, że administrator lub lokalny użytkownik woli używać programu „`login-local`” niż „`login`”, który dostarczany jest przez pakiet Debiana o nazwie `login`.

Nie należy:

- Nadpisywać pliku `/bin/login` plikiem `login-local`.

System zarządzania pakietami nie będzie wiedział o tej zmianie i po prostu nadpisze Twój plik `/bin/login` jeśli `login` (lub każdy inny pakiet dostarczający plik `/bin/login`) zostanie zainstalowany lub uaktualniony.

Zrób raczej tak:

- Wykonaj:

```
dpkg-divert --divert /bin/login.debian /bin/login
```

aby każde przyszłe instalacje pakietu `login` zamiast zapisywać plik `/bin/login` zapisywały go jako `/bin/login.debian`

- Następnie wykonaj:

```
cp login-local /bin/login
```

aby przenieść Twój lokalnie zbudowany program na właściwe miejsce.

Więcej informacji znajdziesz w `dpkg-divert(8)`.

## 10.9 W jaki sposób dodać do listy dostępnych pakietów moje lokalnie zbudowane pakiety tak, aby system zarządzania pakietami o nich wiedział?

Wykonaj polecenie:

```
dpkg-scanpackages BIN_KAT PLIK_NADP [PRZEDR_SCIEZKI] > moje_Pakiety
```

gdzie:

- BIN\_KAT jest katalogiem gdzie przechowywane są pliki archiwów Debiana (zwykle mają rozszerzenia „.deb”).
- PLIK\_NADP (ang. override file) jest plikiem, który modyfikowany jest przez opiekunów wydania i, dla pakietów z dystrybucji main, zwykle znajduje się w archiwum FTP Debiana jako `indices/override.main.gz`. Możesz zignorować ten plik dla lokalnych pakietów.
- PRZEDR\_SCIEZKI jest *opcjonalnym* parametrem, którego wartość może zostać dołączona do nazwy tworzonego pliku `moje_Pakiety`.

Kiedy już plik `moje_Pakiety` zostanie utworzony powiadom o tym system zarządzania pakietami wykonując polecenie:

```
dpkg --merge-avail moje_Pakiety
```

Jeśli używasz APT to możesz również dodać lokalne repozytorium do swojego pliku `sources.list(5)`.

## 10.10 Niektórzy użytkownicy lubią mawk, inni gawk; jedni lubią vim'a, inni lubią elvis'a; niektórzy lubią trn, inni lubią tin. Jak Debian wspiera taką różnorodność upodobań?

Istnieje wiele przypadków kiedy dwa pakiety dostarczają dwie różne wersje programu o takiej samej funkcjonalności. Użytkownicy mogą preferować jeden z nich bardziej od drugiego z przyzwyczajenia, lub z powodu interfejsu użytkownika, który dla danego pakietu jest, w jakiś sposób, bardziej przyjazny niż interfejs drugiego. Różni użytkownicy w tym samym systemie mogą dokonać różnych wyborów.

Debian używa systemu pakietów „wirtualnych” aby pozwolić administratorom systemu na wybór (lub pozwolić wybrać użytkownikom) ulubione narzędzia, gdy istnieją dwie lub więcej

wersji z taką samą podstawową funkcjonalnością, która zaspokoi wymagania zależności bez podawania nazwy konkretnego pakietu.

Na przykład: w systemie zainstalowane są dwie różne wersje czytników grup dyskusyjnych. Serwer grup dyskusyjnych może 'zalecać' aby w systemie był zainstalowany *jakiś* czytnik grup dyskusyjnych ale wybór `tin'a` lub `trn'a` pozostawiony zostaje użytkownikowi. Działa to w ten sposób, że oba pakiety `tin` oraz `trn` dostarczają wirtualny pakiet `news-reader`. Który program zostanie wywołany zależy od dowiązania pliku z nazwą wirtualnego pakietu `/etc/alternatives/news-reader` do wybranego pliku czytnika np. `/usr/bin/trn`.

Pojedyncze dowiązanie nie wystarcza aby wspierać pełne użycie alternatywnych programów. Strony pomocy i prawdopodobnie inne, powiązane z programem, pliki muszą także zostać wybrane. Skrypt Perl'a `update-alternatives` dostarcza sposobu, który zapewnia, że wszystkie pliki powiązane z danym pakietem zostaną wybrane jako domyślne dla systemu.

Na przykład, aby sprawdzić jakie pliki wykonywalne dostarcza 'x-window-manager' uruchom:

```
update-alternatives --display x-window-manager
```

Jeśli chcesz to zmienić uruchom:

```
update-alternatives --config x-window-manager
```

i wykonaj instrukcje, które pojawią się na ekranie (po prostu naciśnij klawisz z cyfrą, która znajduje się przy programie, który bardziej lubisz).

Jeśli z jakiegoś powodu pakiet nie zarejestruje się jako menedżer okien (wyślij informacje o błędzie jeśli uznasz to za usterkę) lub jeśli używasz menedżera okien z katalogu `/usr/local` (taki wybór nie pojawi się na ekranie), możesz uaktualnić dowiązania poprzez parametry wywołania tak jak na przykładzie poniżej:

```
update-alternatives --install /usr/bin/x-window-manager \  
x-window-manager /usr/local/bin/wmaker-cvs 50
```

Pierwszy parametr za '-install' jest dowiązaniem symbolicznym, które wskazuje na `/etc/alternatives/NAZWA`, gdzie `NAZWA` jest drugim parametrem. Trzeci parametr to program, do którego `/etc/alternatives/NAZWA` powinien zostać dowiązany, a czwarty jest priorytetem (większe wartości wskazują, że ta alternatywa, przy działaniu automatycznym, będzie wybrana z większym prawdopodobieństwem).

Aby usunąć alternatywny wpis, który dodałeś uruchom po prostu:

```
update-alternatives --remove x-window-manager /usr/local/bin/wmaker-cvs
```

## Rozdział 11

# Pomoc dla systemu Debian GNU/Linux

### 11.1 Jakie inne dokumentacje istnieją dla systemu Debian GNU/Linux?

- Instrukcja instalacji dla obecnego wydania - zobacz: <http://www.debian.org/releases/stable/installmanual>.
- Podręcznik zasad zawiera normy dotyczące dystrybucji, takie jak: struktura i zawartość archiwum Debiana, zagadnienia dotyczące projektowania systemu operacyjnego itd. Zawiera on również wymagania techniczne, jakie musi spełnić każdy pakiet aby zostać włączonym do dystrybucji, oraz dokumentację podstawowych aspektów technicznych pakietów binarnych i źródłowych Debiana.

Pobierz go z pakietu: `debian-policy` lub z <http://www.debian.org/doc/devel-manuals#policy>.

- Dokumentacja dotycząca zainstalowanych pakietów Debiana: Większość pakietów posiada pliki pomocy, które zapisywane są do `/usr/doc/PAKIET`.
- Dokumentacja dla projektu Linux: Pakiety Debiana `doc-linux` zawierają wszystkie najnowsze wersje HOWTO i mini-HOWTO z Projektu Dokumentacji Linuksa (<http://www.tldp.org/>).
- Strony podręcznika systemowego 'man': Większość poleceń posiada strony podręcznika systemowego napisane w postaci plików 'man'. Są one opisane przez części indeksu 'man' w którym się znajdują: n.p. `foo(3)` odnosi się do strony podręcznika, która znajduje się w `/usr/share/man/man3/` i może zostać wywołana przez wydanie polecenia: `man 3 foo`, lub po prostu `man foo` jeśli trzecia część jest pierwszą zawartą w podręczniku systemowym dotyczącym `foo`.

Można dowiedzieć się, który katalog z `/usr/share/man/` zawiera konkretną stronę poprzez polecenie `man -w foo`.

Nowi użytkownicy Debiana powinni wiedzieć, że strony podręcznika systemowego dla wielu poleceń nie są osiągalne, aż do czasu gdy zainstalują pakiety:

- `man-db`, który zawiera program `man` oraz inne programy do kontroli i sterowania stronami podręcznika systemowego.
- `manpages`, który zawiera strony podręcznika systemowego. (zobacz 'Jak Debian obsługuje języki inne niż angielski?' na stronie 17).
- Strony 'info': Dokumentacja użytkowników dotycząca wielu poleceń, zwłaszcza narzędzi GNU, jest dostępna w postaci plików 'info', a nie stron podręcznika systemowego. Pliki takie mogą być przeczytane przez narzędzie GNU `info`, uruchamiane `M-x info` w GNU Emacs lub przez inne przeglądarki.

Największą przewagą względem oryginalnych stron podręcznika systemowego 'man' jest hipertekstowy system stron `info`. *Nie* potrzebują one jednak przeglądarki WWW; `info` może zostać uruchomione w zwykłej konsoli tekstowej. Standard `info` został stworzony przez Richarda Stallmana i poprzedzał WWW.

Zapamiętaj, iż możesz mieć dostęp do dużej ilości dokumentacji w Twoim systemie używając przeglądarki WWW, bezpośrednio, przez polecenia 'dwww' lub 'dhelp', znajdujące się w poszczególnych pakietach.

## 11.2 Czy w Internecie dostępne są jakieś możliwości do dyskusji na temat Debiana?

Tak. Poczta elektroniczna jest główną metodą wspierania użytkowników Debiana.

### 11.2.1 Listy dyskusyjne.

Jest ich wiele Listy dyskusyjne związane z Debianem (<http://www.debian.org/MailingLists/>).

Kompletny spis list dyskusyjnych znajduje się `/usr/share/doc/debian/mailling-lists.txt` pod warunkiem, że w systemie zainstalowany jest pakiet `doc-debian`.

Listy dyskusyjne Debiana nazywane są według następującego wzoru `debian-temat_listy`. Przykładem są `debian-announce`, `debian-user`, `debian-news`. Aby zapisać się na listę dyskusyjną `debian-temat_listy`, wyślij list pocztą elektroniczną do `debian-temat_listy-request@lists.debian.org` wraz ze słowem "subscribe" w temacie wiadomości. Pamiętaj aby dodać `-request` do adresu listu, gdy używasz tej metody zapisywania się lub rezygnacji z listy dyskusyjnej. W przeciwnym wypadku Twój list elektroniczny zostanie wysłany na listę dyskusyjną.

Jeśli posiadasz przeglądarkę WWW obsługującą formularze możesz zapisać się na listę dyskusyjną używając Formularza WWW (<http://www.debian.org/MailingLists/subscribe>). Możesz również zrezygnować z listy używając Formularza WWW (<http://www.debian.org/MailingLists/unsubscribe>).

W razie jakichś problemów adres administratora list dyskusyjnych to: <listmaster@lists.debian.org>

Archiwa list dyskusyjnych Debiana dostępne są pod adresem WWW: <http://lists.debian.org/>.

### Jakie zasady panują na listach dyskusyjnych?

Gdy korzystasz z list dyskusyjnych związanych z Debianem postępuj zgodnie z następującymi regułami:

- Nie wysyłaj spamu. Zobacz Polityka rozpowszechniania list dyskusyjnych związanych z Debianem (<http://www.debian.org/MailingLists/#ads>).
- Nie przeklinaj; to niekulturalne. Ludzie rozwijający Debiana to ochotnicy, którzy poświęcają swój czas, energię i pieniądze próbując scalić projekt Debian.
- Nie używaj przekleństw; wielu ludzi otrzymuje listę za pomocą pakietu radiowego, gdzie przeklinanie jest nielegalne.
- Upewnij się czy używasz odpowiedniej listy. *Nie* wysyłaj swojej prośby o zapisanie lub rezygnację do listy dyskusyjnej<sup>1</sup>
- Zobacz część 'Jak mogę zgłosić błąd dotyczący Debiana?' na następnej stronie aby dowiedzieć się więcej o zgłaszaniu błędów.

#### 11.2.2 Opiekunowie pakietów.

Użytkownicy mogą również adresować pytania do opiekunów poszczególnych pakietów. By zadać pytanie opiekunowi pakietu xyz, wyślij list pocztą elektroniczną pod adres [xyz@packages.debian.org](mailto:xyz@packages.debian.org).

#### 11.2.3 Grupy dyskusyjne.

Pytania nie dotyczące jednoznacznie Debiana użytkownicy powinni zadawać na grupach dyskusyjnych związanych z Linuksem. Są to grupy dyskusyjne comp.os.linux.\* lub linux.\*. Wiele adresów grup dyskusyjnych dotyczących Linuksa oraz pokrewnych materiałów znaleźć możemy na stronach WWW : np. Linux Online (<http://www.linux.org/docs/usenet.html>) lub LinuxJournal (<http://www.linuxjournal.com/helpdesk.php>).

---

<sup>1</sup>. Używaj do tego adresu [debian-temat\\_listy-REQUEST@lists.debian.org](mailto:debian-temat_listy-REQUEST@lists.debian.org).

### 11.3 Czy istnieje szybka metoda wyszukiwania materiałów związanych z systemem Debian GNU/Linux?

Jest wiele wyszukiwarek, które pomagają szukać dokumentów związanych z Debianem:

- Debian WWW search site (<http://search.debian.org/>).
- Google Groups (<http://groups.google.com/>): przeszukuje listy dyskusyjne.  
Na przykład: aby dowiedzieć się więcej na temat sterowników do kontrolerów Promise dla Debiana, spróbuj wyszukać wyrażenie `Promise Linux sterownik`. Zostaną znalezione wszystkie wiadomości, które zawierają podany ciąg znaków, n.p. dyskusje, których tematem był ten problem. Jeśli dodasz `Debian` do szukanego wyrażenia, w wyniku otrzymasz wiadomości ściśle powiązane z Debianem.
- Zwykle wyszukiwarki, takie jak: AltaVista (<http://www.altavista.com/>) lub Google (<http://www.google.com/>) jeśli będziesz używać prawidłowych warunków wyszukiwania.  
Na przykład: wyszukanie terminu `"cgi-perl"` dostarcza o wiele dokładniejsze wyjaśnienia, niż krótki opis zawarty w tym pakiecie.

### 11.4 Czy istnieją logi znanych błędów?

Dystrybucja Debian GNU/Linux posiada "System Śledzenia Błędów - Bug Tracking System (BTS)", który przechowuje informacje o błędach zgłoszonych przez użytkowników i opiekunów pakietów. Każdy błąd otrzymuje swój numer i przechowywany jest w pliku tak długo, aż nie zostanie uznany za usunięty.

Kopie tych informacji są dostępne pod adresem <http://www.debian.org/Bugs/>.

Dostęp do bazy danych "Systemu Śledzenia Błędów (BTS)" zapewnia serwer pocztowy. Aby dowiedzieć się więcej, wyślij list drogą elektroniczną zawierający słowo `"help"` na adres: [request@bugs.debian.org](mailto:request@bugs.debian.org).

### 11.5 Jak mogę zgłosić błąd dotyczący Debiana?

Jeśli znaleziono błąd w Debianie, postępuje się zgodnie z instrukcją zgłaszania błędów dotyczących Debiana. Instrukcję tę można otrzymać na kilka sposobów:

- Poprzez anonimowy FTP. Strony serwerów lustrzanych Debiana zawierają instrukcję w pliku `doc/bug-reporting.txt`.
- Poprzez stronę WWW. Kopia instrukcji znajduje się pod adresem <http://www.debian.org/Bugs/Reporting>.

- W każdej dystrybucji Debiana poprzez instalację pakietu `doc-debian`. Instrukcja znajduje się w pliku `/usr/doc/debian/bug-reporting.txt`.

Możesz również użyć pakietu `bug` lub `reportbug`, które poprowadzą Cię przez proces zgłaszania błędów i wyślą wiadomość o nich pod odpowiedni adres wraz z dodatkowymi informacjami, dotyczącymi Twojego systemu.

Jeśli chcesz wysłać raport o błędzie z programu pocztowego, wyślij wiadomość do `<submit@bugs.debian.org>`. Pierwsza linia wiadomości musi zawierać:

```
Package: nazwa pakietu
```

Następna linia powinna zawierać wersję pakietu podaną w podobny sposób:

```
Version: numer wersji
```

Numer wersji dla jakiegokolwiek pakietu zainstalowanego w Twoim systemie można uzyskać wydając polecenie:

```
dpkg -s nazwa pakietu
```

Reszta wiadomości powinna zawierać dokładny opis błędu, dystrybucji Debiana jakiej używasz oraz numery wersji innych stosowanych pakietów. Wersję dystrybucji Debiana, którą obecnie używasz, można wyświetlić poprzez polecenie

```
cat /etc/debian_version
```

.

Po wysłaniu przez Ciebie zgłoszenia błędu zostaniemy automatycznie powiadomieni o otrzymaniu zgłoszenia. Również automatycznie błąd otrzyma numer (zapisany w logu błędów) oraz zostanie wysłany na listę dyskusyjną `debian-bugs-dist`.

Jeśli znaleziony błąd jest powiązany z wieloma pakietami, zamiast wysyłać wiele podobnych zgłoszeń błędu, należy wysłać zgłoszenia odnośnie błędów do `<maintonly@bugs.debian.org>` (zamiast adresu `submit@...`), aby powiadomić tylko opiekunów poszczególnych pakietów, a następnie wysłać podsumowanie zgłoszonych błędów do listy mailingowej `debian-devel` lub `debian-bugs-dist`.

Dodatkowo istnieje tester pakietów Debiana nazywany Lintian (<http://www.debian.org/lintian/>). Został on tak zaprojektowany, aby automatycznie przeszukiwać pakiety w celu znalezienia naruszeń zasad dystrybucji czy błędów wykrytych już wcześniej w innych pakietach. Zatem jeśli znajdziesz błąd, który może również wystąpić w innych pakietach, lepiej będzie jeśli skontaktujesz się z opiekunami pakietu Lintian `<lintian-maint@debian.`

org>. Powstanie nowej metody wyszukiwania błędów zapobiegnie pojawianiu się tego samego błędu w innych pakietach dystrybucji.

Możesz również używać adresu <quiet@bugs.debian.org>, aby zgłaszać błędy tylko do "Systemu Śledzenia Błędów (BTS)", bez jednoczesnego wysyłania ich do listy dyskusyjnej debian-bugs-dist lub do opiekuna pakietu. Adres ten jest wykorzystywany bardzo rzadko np. jeśli chcesz dołączyć drugorzędne dane do swojego zgłoszenia błędu lub gdy chcesz zaznaczyć coś w logu "Systemu Śledzenia Błędów (BTS)", a już wcześniej powiadomiono o tym opiekuna projektu.

## Rozdział 12

# Wspomaganie projektu Debian

Wspomóc projekt Debian możesz na kilka sposobów. Np. przez ofiarowanie swojego czasu (na projektowanie nowych pakietów, opiekowanie się pakietami już istniejącymi, lub na wsparcie dla użytkowników), zasobów (serwery lustrzane dla archiwów FTP i WWW) oraz pieniędzy (opłaty za nowe stoiska do testów jak również sprzęt na kolejne archiwa).

### 12.1 W jaki sposób mogę zostać deweloperem Debiana?

Rozwijanie Debiana jest otwarte dla wszystkich. Poszukiwani są zarówno nowi użytkownicy z odpowiednimi umiejętnościami i/lub chęcią nauki, do opieki nad już istniejącymi pakietami osieroconymi przez ich poprzednich deweloperów, nad rozwojem nowych, jak również do wsparcia technicznego dla użytkowników.

Opis jak zostać deweloperem Debiana możesz znaleźć na stronie New Maintainer's Corner (<http://www.debian.org/devel/join/newmaint>) znajdującej się na głównej stronie projektu Debiana.

### 12.2 Jak mogę wnieść zasoby do projektu Debiana?

Od kiedy sam Debian jak i całe mnóstwo oprogramowania dla niego jest ogólnie i powszechnie dostępne na całym świecie, rośnie zapotrzebowanie na serwery lustrzane. Wskazane jest zatem, aby umieszczać całe archiwum na kolejnych serwerach lustrzanych, ale nie jest to absolutnie niezbędne. Proszę odwiedzić stronę na temat serwerów lustrzanych Debiana (<http://www.debian.org/mirror/size>), aby dowiedzieć się więcej o wymaganej przestrzeni dyskowej dla serwera lustrzanego, na którym mają zostać umieszczone zasoby Debiana.

Obsługa serwerów lustrzanych jest głównie realizowana automatycznie poprzez skrypty, bez jakiegokolwiek interakcji. Jednakże okazjonalnie zdarzają się krótkotrwałe przerwy w pracy serwerów lustrzanych lub zmiany systemowe, wtedy ludzka interwencja jest niezbędna.

Jeśli posiadasz szybkie połączenie z Internetem, odpowiednie zasoby i jesteś chętny poświęcić swój czas (lub znaleźć kogoś) na regularne zarządzanie serwerem lustrzanym zawierającym zasoby Debiana, czy to w całości czy w części, proszę skontaktować się z nami pod adresem `<debian-admin@lists.debian.org>`.

## 12.3 Jak mogę finansowo wspomóc projekt Debian?

Można tego dokonać przez indywidualną darowiznę na rzecz jednej z dwóch kluczowych organizacji dla rozwoju Debiana.

### 12.3.1 Oprogramowanie w Publicznym Interesie (SPI)

Oprogramowanie w Publicznym Interesie (Software in the Public Interest) jest niedochodową organizacją, utworzoną gdy Fundacja Wolnego Oprogramowania (FSF) zrezygnowała ze sponsorowania Debiana. Celem organizacji jest rozwijanie i rozprowadzanie Wolnego Oprogramowania.

Nasze cele są bardzo podobne do celów Fundacji Wolnego Oprogramowania (FSF) i zachęcamy programistów do stosowania licencji GNU GPL w swoich programach. Jednakże różnimy się trochę pod kątem tworzenia i dystrybuowania Linuksa, które odbiega w wielu szczegółach technicznych od systemu GNU zainicjowanego przez FSF. Wciąż jednak komunikujemy się z Fundacją Wolnego Oprogramowania i współpracujemy poprzez przesyłanie zmian do oprogramowania GNU oraz poprzez zachęcanie naszych użytkowników do wspierania finansowego celów FSF oraz projektu GNU.

Więcej informacji o SPI możesz uzyskać na stronie: <http://www.spi-inc.org/>.

### 12.3.2 Fundacja Wolnego Oprogramowania (FSF)

Aktualnie nie ma formalnego związku pomiędzy Debianem a Fundacją Wolnego Oprogramowania. Jednak FSF jest odpowiedzialna za niektóre z najważniejszych komponentów oprogramowania w Debianie, włączając kompilator GNU C, GNU Emacs oraz wiele wykonawczych bibliotek języka C, które są wykorzystywane przez wszystkie programy w systemie. Fundacja Wolnego Oprogramowania jest pionierem tego, czym jest dzisiaj Wolne Oprogramowanie. Z fundacji tej wywodzi się licencja GNU GPL, która jest stosowana w większości oprogramowania Debiana. To dzięki niej istnieje projekt, którego celem jest stworzenie całkowicie darmowego Uniksa. Debiana powinno się uważać za potomka systemu GNU.

O Fundacji Wolnego Oprogramowania (FSF) możesz poczytać na tej stronie: <http://www.fsf.org/>.

## Rozdział 13

# Redystrybuowanie Debiana GNU/Linux jako produktu komercyjnego

### 13.1 Czy mogę sprzedawać Debiana na przygotowanych przeze mnie płytach CD?

Śmiało. Nie potrzebujesz pozwolenia na dystrybuowanie czegoś co *sam przygotowałeś*. Nie musisz nam płacić. Jednakże wszyscy wytwórcy CD muszą honorować licencje programów w Debianie. Dla przykładu, wiele programów jest objętych licencją GPL, która wymaga od Ciebie dystrybuowania przez Ciebie również kodu.

Ponadto będziemy publikować listę wytwórców CD, którzy ofiarowują pieniądze, oprogramowanie i czas dla projektu Debiana. Będziemy też zachęcać użytkowników do kupowania u tych wydawców Debiana, którzy ofiarowują pieniądze.

### 13.2 Czy Debian może być sprzedawany razem z płatnym oprogramowaniem?

Tak. Cały czas główne komponenty Debiana są bezpłatnym oprogramowaniem. Udostępniamy jednak miejsce dla programów, które darmowymi nie są.

Wydawcy Debiana na CD *mogą* dystrybuować programy jakie umieściliśmy w katalogu, niezależnie od warunków licencji lub prywatnych ustaleń autorów poszczególnych programów. Wydawcy Debiana na CD mogą również dołączać do niego płatne programy jakie posiadają z innych źródeł. To nie jest nic nowego: darmowe i komercyjne oprogramowanie jest często wydawane na tych samych CD co Debian. Oczywiście wciąż zachęcamy autorów różnych programów do pisania ich jako oprogramowania bezpłatnego.

### **13.3 Robię specjalną dystrybucję Linuksa dla specyficznego rynku. Czy mogę użyć Debiana GNU/Linux i dodać moje własne oprogramowanie?**

Tak. Dla przykładu ktoś przygotowuje dystrybucję Linuksa dla krótkofalowców, wraz ze specjalistycznym oprogramowaniem. Rozpoczyna od Debiana jako podstawowego systemu i dodaje swoje programy np. sterujące pracą nadajnika, śledzące satelity, itd. Wszystkie te programy są na płytach CD razem z Debianem, więc użytkownicy dystrybucji wypuszczonej przez niego będą w stanie łatwo uaktualnić swoje programy, kiedy on udostępni kolejne płyty CD.

Na rynku jest już kilka podobnych dystrybucji Debiana przygotowanych dla pewnej grupy ludzi np. Corel Linux czy Storm Linux, które różnią się od oryginalnej dystrybucji Debiana GNU/Linux, jednakże nadal zawierają sporo naszego oprogramowania.

Debian dostarcza również mechanizm pozwalający programistom i administratorom instalować lokalne wersje wybranych plików w taki sposób, że nie będą nadpisane, kiedy inne pakiety są uaktualniane. Więcej o tym mowa w pytaniu 'Jak mogę nadpisać plik instalowany przez inny pakiet tak żeby używana była moja wersja?' na stronie [56](#).

### **13.4 Czy mogę dołożyć do Debiana mój komercyjny program, który łatwo się instaluje?**

Śmiało. Narzędzie do zarządzania pakietami jest wolnym programem; pakiety mogą ale nie muszą być wolnym oprogramowaniem, można instalować je wszystkie.

## Rozdział 14

# Spodziewane zmiany w następnej wersji Debiana

### 14.1 Zwiększone bezpieczeństwo

Debian udostępnia ukrywanie haseł od swojej wersji 1.3. Na dodatek linuksowa biblioteka Pluggable Authentication Modules (znana jako libpam (<http://www.kernel.org/pub/linux/libs/pam/>)), która pozwala administratorom systemu na wybranie sposobu autoryzacji specyficznej dla danej aplikacji, pierwotnie ustala uwierzytelnianie za pomocą ukrywania haseł.

Włączając pełną obsługę dla zaawansowanych metod uwierzytelniania takich jak Kerberos, RSBAC i innych.

### 14.2 Wsparcie dla użytkowników nie znających języka angielskiego.

Debian zawsze ma jakieś wsparcie dla użytkowników nie znających języka angielskiego. Zobacz 'Jak Debian obsługuje języki inne niż angielski?' na stronie [17](#).

Mamy nadzieję znaleźć ludzi, którzy udostępnią wsparcie dla większej liczby języków i przekładów. Niektóre programy są już przygotowane do lokalizacji, tak więc potrzebujemy tłumaczy. Wiele programów wciąż jest tłumaczonych na nowe języki.

Projekt Tłumaczeń GNU <ftp://ftp.gnu.org/pub/gnu/ABOUT-NLS> pracuje nad tłumaczeniem programów GNU.

### **14.3 Więcej architektur**

Już wkrótce spodziewany jest kompletny Debian na inne architektury, takie jak SPARC64 czy SuperH.

### **14.4 Więcej jąder systemu**

Poza Debianem GNU/Hurd, Debian jest przenoszony na wiele jąder BSD, czyli na NetBSD, FreeBSD i OpenBSD.

## Rozdział 15

# Ogólne informacje dotyczące Często Zadawanych Pytań (FAQ)

### 15.1 Autorzy

Pierwszym wydaniem tego FAQ zajmowali się J.H.M. Dassen (Ray) i Chuck Stickelman. Autorami odnowionego FAQ systemu Debian GNU/Linux są: Susan G. Kleinmann i Sven Rudolph. Po nich FAQ zajmował się Santiago Vila. Koordynatorem obecnej wersji jest Josip Rodin.

Część informacji zaczerpnięto z:

- The Debian-1.1 release announcement, autorstwa Brucea Perensa (<http://www.perens.com/>).
- The Linux FAQ, autorstwa Iana Jacksona (<http://www.chiark.greenend.org.uk/~ijackson/>).
- Debian Mailing Lists Archives (<http://lists.debian.org/>),

FIXME the dpkg programmers' manual and the Debian Policy manual (see 'Jakie inne dokumentacje istnieją dla systemu Debian GNU/Linux?' na stronie 59)

- od wielu deweloperów, ochotników, beta-testerów oraz
- niespodziewanych wspomnień autorów. :-)

Autorzy pragną podziękować wszystkim tym, którzy pomogli stworzyć ten dokument.

Reklamacji nie uwzględnia się. Znaki handlowe są zastrzeżone.

## 15.2 Tłumacze

Do powstania polskiej wersji tego dokumentu przyczyniły się następujące osoby z projektu PDDP (<http://debian.linux.org.pl>):

- Marcin Andruszkiewicz
- Marcin Betlej - <marbej@debian.linux.org.pl>
- Mariusz Centka - <mariusz.centka@debian.linux.org.pl>
- Karol Czachorowski
- Tomasz Dziejcz
- Paweł Ekert
- Bartosz Feński - <fenio@debian.linux.org.pl> - koordynator tłumaczenia
- Radosław Grzanka - <radekg@debian.linux.org.pl>
- Bartosz 'Xebord' Janowski
- Dominik Juszczyk - <djus@debian.linux.org.pl>
- Marcin Paweł Kobierzycki - <m-kobierzycki@debian.linux.org.pl>
- Maciej Krzywiński
- Marcin Kuras - <kura@debian.linux.org.pl>
- Jacek Lachowicz
- Bartłomiej "MYCHA" Mroczkowski - <mycha@debian.linux.org.pl>
- Tomasz Z. Napierała - <zen@debian.linux.org.pl>
- Marcin Płaneta - <welnian@debian.linux.org.pl>
- Mateusz Prichacz - <mateusz@debian.linux.org.pl>
- Marcin Rogowski
- Sławomir Siejka
- Michał Skuza
- Dominik Stodolny - <phobos@debian.linux.org.pl>
- Przemysław Adam Śmiejek <tristan@debian.linux.org.pl>
- Grzegorz Witkowski
- Krzysztof Witkowski <tjup@debian.linux.org.pl>
- Bartosz Zapałowski <zapal@debian.linux.org.pl>

### 15.3 Informacje zwrotne

Komentarze i uzupełnienia tego dokumentu są zawsze mile widziane. Prosimy o wysłanie listu na adres `<doc-debian@packages.debian.org>`, albo dopisanie do listy życzeń dotyczących pakietu `doc-debian`.

### 15.4 Dostępność

Najnowsza wersja niniejszego FAQ jest do poczytania na stronach WWW Debiana pod adresem <http://www.debian.org/doc/FAQ/>.

FAQ można również pobrać w postaci czystego pliku tekstowego, pliku HTML, w formacie PostScript albo PDF z <http://www.debian.org/doc/user-manuals#faq>. Znajduje się tam w kilku wersjach językowych.

Oryginalne pliki SGML użyte do stworzenia tej dokumentacji są dostępne w pakiecie źródłowym `doc-debian` albo w repozytorium CVS: `:pserver:anonymous@cvs.debian.org:/cvs/debian-doc/ddp/manuals.sgml/faq`

### 15.5 Format dokumentu

Ten dokument został napisany przy pomocy DebianDoc SGML DTD (utworzonego na podstawie LinuxDoc SGML). System DebianDoc umożliwia nam tworzenie plików w wielu formatach na podstawie jednego źródła, tzn. ten dokument może być oglądany jako strona internetowa, czysty tekst, TeX DVI, PostScript, PDF albo GNU Info.

Narzędzia konwertujące dla formatu DebianDoc SGML są dostępne w pakiecie `debiandoc-sgml`.