

# *Dystrybucje Linuksa*



# Wstęp

- **Co to jest dystrybucja?**
  - Jądro wraz z oprogramowaniem potrzebnym do użytkowania systemu.
- **Po co nam dystrybucja?**
  - Dla wygody. Nie musimy sami kompletować oprogramowania potrzebnego do działania systemu. Dystrybucja daje nam pełen zestaw gotowy do zainstalowania.
- **Skąd wziąć?**
  - Internet
  - CD
- **Przeróżne**
  - Większość instalowana na dysk. Najbardziej znane: Red Hat Linux, Debian, Mandriva Linux i Slackware
  - Uruchamiane bezpośrednio z CD np. Knoppix
  - Uruchamiane z dyskietki np. Pocket Linux

# *Z czego zwykle składa się dystrybucja?*

- Jądro systemu
- Biblioteki GNU
- Dodatkowe oprogramowanie
- Dokumentacja
- Multimedia



# *Jakie były początki?*

- Na samym początku Linux składał się z basha, kompilatora gcc oraz jądra realizującego operacje wejścia/wyjścia
- W lutym 1992 powstała MMC Interim Linux – na jej podstawie budowano kolejne wersje.
- W sierpniu 1992 pojawił się SLS – Softlanding Linux System – dosyć obszerny zbiór skompilowanych pakietów. Po zainstalowaniu zajmował 27Mb! Wśród jego aplikacji znalazł się
  - Arkusz kalkulacyjny
  - Gry: chess, tetris, othello, spider
- Powstawały kolejne. Najczęściej nazywane od nazwisk autorów.
- W lipcu 1993 na podstawie SLS powstał Slackware. Najstarsza do dziś używana dystrybucja.
- Również na Podstawie SLS w sierpniu ogłoszono powstanie Debiana



- W 1994 powstał Red Hat
  - od początku było to przedsięwzięcie komercyjne
  - Specjalnie powstała firma aby tworzyć tą dystrybucję Po roku weszła na giełdę i stała się jedną z największych firm tworzących oprogramowanie.
- Powstaje mnóstwo rozbudowanych systemów które są coraz bardziej przyjazne użytkownikowi i na domowych komputerach zaczynają konkurować z Windowsem. Np. Ubuntu.
- Do dziś powstają nowe dystrybucje, i jest ich coraz coraz więcej Spróbujmy zrobić swoją :)



# *Linux From Scratch*



# *Czym jest LFS?*

Linux From Scratch to projekt, który dostarcza szczegółowe instrukcje umożliwiające zbudowanie własnej “dystrybucji” Linuksa.

Witryna: [www.linuxfromscratch.org](http://www.linuxfromscratch.org)

Twórca: Gerard Beekmans

Polskie tłumaczenie: [republika.pl/loz/lfs/](http://republika.pl/loz/lfs/)

Jerzy Iwaniuk

(trochę nieaktualne)

---

---

# Co LFS dostarcza?

- Książkę on-line ze wyszczególnionymi kolejnymi krokami,
  - Linki do stron, z których będziemy mogli ściągnąć potrzebne elementy systemu,
  - Live-CD z Linuksem, którego będziemy mogli używać podczas pracy nad naszą własną dystrybucją.
- 
-



# *Kolejne kroki*

1. Przygotowania,
  2. Konstrukcja tymczasowego systemu,
  3. Instalacja podstawowych programów systemu,
  4. Ustawianie skryptów ładowania systemu,
  5. Przygotowanie systemu LFS do uruchamiania się,
  6. Reboot!
- 
-

# Przygotowania

- Przygotowywanie nowej partycji

*Na system minimalny potrzeba partycji rozmiaru około 1.2 GB. To wystarczy na zapisanie wszystkich źródeł (the source tarballs) i skompilowanie wszystkich pakietów. Jeśli jednak system LFS ma być używany jako system główny, to partycja powinna być większa (ok 2-3 GB). Partycję tworzymy za pomocą programu cfdisk lub fdisk.*
- Przygotowanie nowego systemu plików

*Teraz, gdy mamy już pustą partycję, możemy założyć na niej system plików (ext2, aczkolwiek można znaleźć informacje o tym, jak w systemie LSF zamontować ext3).*
- Ściągnięcie potrzebnych pakietów i łąt (kilkadziesiąt!!!)

*Ściągnięte pakiety i łąty musimy składować i rozpakować w odpowiednio wybranym katalogu roboczym.*
- Przygotowanie środowiska pracy

*Konfigurujemy odpowiednio powłokę bash oraz tworzymy przydatne zmienne systemowe.*

# *Tymczasowy system*

Kompilujemy i instalujemy minimalny system Linux. Będzie on zawierać wystarczające narzędzia do konstrukcji ostatecznego systemu LFS.

Tworzenie tego minimalnego systemu przebiega w dwóch etapach: najpierw budujemy nowiuteńkie i niezależne od komputera narzędzia (kompilator, assembler, linker i biblioteki), a potem używamy ich do pozostałych podstawowych narzędzi.

---

---

# Instalacja potrzebnych programów

a)Przejdźcie do środowiska chroot

b)Zmiana właściciela tymczasowego systemu

*Nie jest to konieczne, ponieważ pliki tymczasowego systemu i tak można na koniec skasować, ale jeśli z jakichś powodów tego nie zrobimy, to pozostanie nam katalog, którego właścicielem jest użytkownik nie istniejący w systemie.*

c)Tworzenie katalogów

*Tworzymy standardowe katalogi: bin, boot, dev, etc, home...*

d)Montowanie systemów plików proc oraz devpts

*W środowisku chroot muszą być dostępne systemy plików proc oraz devpts, żeby pewne programy mogły działać prawidłowo. Możesz montować dowolny system plików ile razy zechcesz i w dowolnych miejscach, więc to żaden problem, że są one już zamontowane w systemie - szczególnie, że są to wirtualne systemy plików.*

# Instalacja potrzebnych programów

## e) Tworzenie niezbędnych linków symbolicznych

*Niektóre programy mają wkompileowane ścieżki, których jeszcze nie ma. Aby je zadowolić tworzymy trochę linków symbolicznych, które zostaną zastąpione prawdziwymi plikami w miarę postępów tego rozdziału, przy instalowaniu całego oprogramowania.*

## f) Tworzenie plików passwd i group

*Żeby użytkownik root mógł się zalogować i być rozpoznany pod nazwą "root", muszą istnieć odpowiednie wpisy w plikach /etc/passwd oraz /etc/group.*

## g) Tworzenie urządzeń

*Robimy to za pomocą gotowego skryptu MAKEDEV*

## h) Instalacja bibliotek i programów

*Jest to najdłuższa część całej pracy: ręcznie instalujemy wszystkie programy i biblioteki z których będzie potem korzystał nasz system operacyjny (a jest ich dużo) .*

## i) Pozbycie się śmieci

---

# Uruchamianie

- Konfiguracja skryptów ładowania systemu
- Kompilacja jądra
- Konfiguracja bootloader'a

*Bootowanie może być skomplikowane. Trzeba naprawdę znać działanie twojego bootloadera i znać pozostałe systemy operacyjne zainstalowane na twoim dysku twardym (dyskach twardych), jeśli chcemy je nadal uruchamiać.*

# REBOOT:)

## Co dalej?

- Beyond Linux From Scratch

*Pozwala rozszerzyć podstawową instalację do systemu bardziej rozbudowanego i wygodniejszego w użyciu.*

- Automatic Linux From Scratch

*Dostarcza narzędzi do zautomatyzowania procesu budowania systemu LFS.*

- Cross Linux From Scratch

*Dostarcza środków do cross-kompilacji systemu LFS (czyli budujemy go na jednym systemie, a kompilujemy na innym).*

- Hardened Linux From Scratch

*Pozwala zbudować system ze zwiększonym bezpieczeństwem.*



# *Standaryzacja Systemu*





# *Po co standaryzować systemy?*

- Rozwarstwienie dystrybucji Linuksa osłabia jego pozycję na rynku na rzecz ujednoliconego Windowsa.
- Wygoda producentów oprogramowania i użytkowników



# *Czym się różnią?*

- Różnym sposobem zarządzania pakietami
- Sposobem instalacji i konfiguracji
- Hierarchią katalogów i rozlokowaniem plików systemowych
- Doborem aplikacji



*Posix*

**P**ortable  
**O**perating  
**S**ystem  
**I**nterface

Uni**X**

Dosłownie: przenośny interfejs systemu  
operacyjnego

---

---

# Posix

- Standaryzuje różne systemy operacyjne
- Zbiór standardów definiujących interfejsy systemu operacyjnego.
- Inne jego nazwy to
  - ISO.IEC 9945:2003 (międzynarodowa norma)
  - IEEE Standard 1003

# *W jakim celu powstał?*

- Aby umożliwić pisanie przenośnego kodu aplikacji.
- Miał ujednolicić systemy oparte na Uniksie aby przywrócić popularność rozwiązaniom opartym o ten system.

# Cechy Posix'a

- Przenośność aplikacji
  - Niekoniecznie przenośne binaria.
  - Nieważny sposób implementacji
  - Jak najmniejszy wpływ na budowę systemu.
  - Jak najwięcej systemów które mogłyby przystosować się do tego standardu.
  - Nie ma wpływu na zagadnienia sprzętowe i administracje systemu
- 
-

# *Czy Linux jest zgodny z POSIX'em ?*

- Linus Torvalds starał się aby tak było. Jednak w początkowej fazie tworzenia Linuksa autor nie miał dostępu do specyfikacji posix'a.
- Mimo, iż nikt nie potrafi udowodnić, że Linux jest w pełni zgodny z POSIX'em to i tak jest uważany za jeden z najbardziej zgodnych systemów z tym standardem.



# *Co z tego wszystkiego mają dystrybucje?*

- Posix dotyczy tylko interfejsu jakim dysponuje programista w danym systemie. Czyli zapewnia to, że możemy instalować oprogramowanie za pomocą kompilacji kodu źródłowego – nie jest to najlepsze rozwiązanie
  - Nie zapobiega rozwarstwianiu się dystrybucji.
  - To dobry standard dla zupełnie różnych systemów operacyjnych, a nie rodziny jaką można nazywać systemy oparte na Linuksie.
  - Nie tak łatwo być zgodnym z tym standardem(koszty certyfikatów zgodności, częste zmiany w dystrybucjach)
- 
-



# *Systemy zgodne z POSIX'em*

- MacOSX
- Minix
- Windows NT

# *Single Unix Specification*

- Dokument opublikowany przez The Open Group.
  - Definiuje jak powinien zachowywać się system klasy Unix.
  - Oparty w dużej mierze o Posix.
  - Różni od Posixa się tym, że to otwarty standard.
  - Mniejsze koszty uzyskania certyfikatu zgodności
  - Od 2002 roku bardziej popularny standard od Posix'a
  - Obecnie aktualna wersja ma numer 3
- 
-

*LSB*

Linux  
Standard  
Base



# LSB

- Ma na celu ujednolicenie dystrybucji Linuksa i zniwelowanie różnic między nimi wyznaczając wspólne standardy.
- Testy których zaliczenie jest wymagane do uzyskania certyfikatu zgodności są udostępniane za darmo
- Standard stworzony przez Free Standards Group – organizacje non-profit

# *Problemy związane z różnicami pomiędzy dystrybucjami Linuksa*

- Kilka formatów pakietów RPM, DEB, TGZ. Korzystanie z tego samego formatu pakietów nie gwarantuje że pakiet zainstaluje się na innej niż macierzysta dystrybucji
- Problem z wypuszczaniem oprogramowania bez jawnego kodu źródłowego – programy w postaci binarnej muszą mieć wiele wersji dostosowanych do poszczególnych dystrybucji



# *Z czego składa się LSB?*

- Ze specyfikacji
- Przykładowej implementacji dystrybucji
- Zestawu testów



# Najważniejsze cechy

- Skupia się na przenośności binariów, a nie tylko kodów źródłowych pomiędzy dystrybucjami
  - Zapewnia, że odpowiednio przygotowane oprogramowanie zainstaluje się na wszystkich zgodnych ze standardem dystrybucjach.
  - Można tworzyć przenośne programy w sensie LSB korzystające z bibliotek nieobjętych standardem (ma to też swoje wady – zwiększa się objętość programów)
  - Sprawdza się w praktyce co powoduje, że coraz więcej firm z branży informatycznej chce wchodzić w skład organizacji Free Standard Group aby mieć wpływ na rozwój tego standardu.
- 
-

# *Co specyfikuje LSB?*

- Interfejs dla programistów udostępniony przez system (oparty o Posix i teoretycznie z nim zgodny)
- Interfejs binarny dla aplikacji
- Hierarchię katalogów w systemie (oparty o Filesystem Hierarchy Standard)
- Położenie niektórych plików systemowych



# Co specyfikuje LSB?

- Podstawowe biblioteki odpowiedzialne za dostęp do procesora i zasobów sprzętowych
- Biblioteki użytkowe
- Komendy i programy użytkowe dostarczane przez system
- Inicjalizację systemu
- Zarządzanie użytkownikami i grupami użytkowników
- Format pakietów RPM
- Najnowsza wersja standardu obejmuje również biblioteki graficzne np. libX11

# *Dystrybucje Linuksa zgodne z LSB*

- Ubuntu
- Debian
- Fedora Core
- Aurox
- SUSE



*Licencje*



# Licencje

Jak wiadomo, każde oprogramowanie jest rozpowszechniane na jakiejś licencji. Jest to ważne, by rozumieć na co dana licencja nam – jako użytkownikowi – pozwala i co jest od nas wymagane. Wśród ogromu najróżniejszych licencji wyróżniamy:

- Licencje oprogramowania
    - Licencje Wolnego Oprogramowania zgodne z GPL
    - Licencje Wolnego Oprogramowania niezgodne z GPL
    - Licencje oprogramowania, które nie jest wolne
  - Licencje dokumentacji
    - Licencje Wolnej Dokumentacji
    - Licencje dokumentacji, która nie jest wolna
  - Licencje dzieł innych niż oprogramowanie i dokumentacja
- 
-

# *Licencje oprogramowania cz.1*

## *Licencje Wolnego Oprogramowania zgodne z GPL*

Cztery rodzaje wolności użytkowników programu:

- wolność uruchamiania programu, w dowolnym celu (wolność 0),
- wolność analizowania, jak program działa, i dostosowywania go do swoich potrzeb (wolność 1). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.
- wolność rozpowszechniania kopii, byście mogli pomóc sąsiadom (wolność 2)
- wolność udoskonalania programu i publicznego rozpowszechniania własnych ulepszeń, dzięki czemu może z nich skorzystać cała społeczność (wolność 3). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.

Oprogramowanie nazywamy **wolnym**, jeśli wszyscy użytkownicy posiadają w pełni wszystkie te prawa.

*Warto zwrócić uwagę na fakt, że ta definicja nie porusza w ogóle kwestii opłat. Oznacza to, że wolne oprogramowanie może być rozpowszechniane za pieniądze, byleby spełniało powyższe warunki. Oznacza to, że możemy zmieniony przez nas program przeznaczyć do sprzedaży.*

# Licencje oprogramowania cz.2

## Licencje Wolnego Oprogramowania zgodne z GPL

Przykładowe Licencje:

- GNU General Public License (GNU GPL)

*Jest to najbardziej popularna licencja wolnego oprogramowania. Jest ona licencją typu copyleft, co oznacza, że wszelkie kopie, lub nawet zmienione wersje oprogramowania z tą licencją, również muszą być z nią rozpowszechniane. Jest to, w pewnym sensie, ograniczeniem wolności, ale w dalszej perspektywie służy to zapewnieniu, że kolejne wersje oprogramowania dalej będą wolne.*

- GNU Lesser General Public License (GNU LGPL)

*Jest licencją wolnego oprogramowania, ale niezbyt silną licencją copyleft, gdyż zezwala na łączenie oprogramowania z modułami, które nie są wolne*

- Licencja X11

*To prosta, liberalna licencja wolnego oprogramowania. Nie jest typu copyleft, jest zgodna z GNU GPL. Na tej licencji były wydane starsze wersje XFree86, korzystają z niej także niektóre z obecnych wariantów XFree86. Późniejsze wersje XFree86 są rozprowadzane na licencji XFree86 1.1 (która jest niezgodna z GPL).*

*Jest czasem nazywana licencją „MIT”, ale to określenie jest mylące, gdyż MIT stosował wiele licencji na oprogramowanie.*

- Zmodyfikowana licencja BSD

*Jest to pierwotna, oryginalna licencja BSD zmieniona przez usunięcie klauzuli dotyczącej reklamowania. Jest to prosta, liberalna licencja wolnego oprogramowania. Nie jest typu copyleft, jest zgodna z GNU GPL.*

# Licencje oprogramowania cz.3

## Licencje Wolnego Oprogramowania niezgodne z GPL

Licencje wymienione w tym podpunkcie nie są zgodne z GPL. Oznacza to najczęściej, że nakładają one na użytkownika dodatkowe ograniczenia lub nawet zobowiązania. Mimo to, są one nadal licencjami wolnego oprogramowania, czyli realizują cztery wymienione wyżej wolności.

- **Pierwotna licencja BSD**

*To prosta, liberalna, licencja wolnego oprogramowania, nie oparta na zasadach copyleft. Ma poważną wadę: „uprzykrzoną klauzulę ogłoszeniową BSD”. Nie jest to usterka krytyczna – nie odbiera programom wolności. Powoduje jednak problemy w praktycznym stosowaniu, łącznie z niezgodnością z GNU GPL.*

*Treść klauzuli:*

*3. We wszelkich materiałach reklamowych wzmiankujących możliwości lub zastosowania tego oprogramowania muszą być zamieszczone następujące podziękowania:*

*“Ten produkt zawiera oprogramowanie opracowane przez Uniwersytet Kalifornijski w Berkeley i jego współpracowników.”*

- **Apache Software License, w wersji 2.0**

*Apache Software License jest niezgodna z GPL, ponieważ zawiera specyficzny wymóg, którego nie ma w GPL: pewne przypadki rozwiązania umowy patentowej.*

- **Eclipse Public License Version 1.0**

- **Licencja PHP, w wersji 3.0**

# Licencje oprogramowania cz.3

## Licencje Wolnego Oprogramowania niezgodne z GPL

cdn ...

- **Wolna Licencja Akademicka (Academic Free License) w wersji 2.1.**

*Academic Free License w wersji 1.0 jest licencją wolnego oprogramowania, nie należy do licencji typu copyleft i jest z kilku powodów niezgodna z GNU GPL.*

*AFL jest niezgodna z GPL z dwu powodów. Po pierwsze, zawarte w niej reguły dotyczące stosowania znaków handlowych wyraźnie wykraczają poza to, czego w rzeczywistości wymaga prawo w niektórych krajach – zakazują tego, co byłoby prawnie dozwolonym użyciem znaku handlowego.*

*Druga niezgodność wynika z zawartej w AFL klauzuli o wzajemnym rozwiązaniu umowy w przypadku wniesienia sprawy o naruszenie patentu („Mutual termination for Patent Action”). Odkładając na bok trudne pytanie, czy tego rodzaju klauzula jest dobrym pomysłem czy złym – jest niezgodna z GPL.*

- **Licencja Otwartego Oprogramowania (Open Software License), w wersji 1.0**

*Wszystkie wersje Open Software License zawierają warunek nakazujący dystrybutorom uzyskanie od odbiorców wprost wyrażonej akceptacji licencji. Co oznacza, że oprogramowanie OSL nie może być rozprowadzane przez umieszczenie na serwerach FTP, gdyż w protokole FTP nie ma mechanizmów umożliwiających wyrażenie takiej zgody. Subskrypcja list wysyłkowych, na które trafiają luty programów, musi być możliwa tylko za zgodą na licencję, a strony z archiwami tych list muszą być poprzedzone tekstem do potwierdzenia kliknięciem. Trzeba przepisać na nowo oprogramowanie do kontroli wersji, żeby móc w nim uzyskiwać od użytkowników akceptację warunków licencyjnych. Tak więc, Open Software License niezwykle utrudnia rozwijanie programów przy użyciu zwykłych narzędzi, z których korzysta się w pracy nad wolnym oprogramowaniem.*



# Licencje oprogramowania cz.4

## Licencje oprogramowania, które nie jest wolne

Do tej kategorii przede wszystkim zaliczane są licencje oprogramowania komercyjnego, która wprost ograniczają wymienione powyżej wolności. Istnieją jednakże licencje, których nazwa mogłaby sugerować, że są licencjami wolnego oprogramowania, gdy tymczasem w rzeczywistości nimi nie są.

- (Pierwotna) Licencja Twórcza (Artistic License)

*Nie można stwierdzić, że jest licencją wolnego oprogramowania, ponieważ jest zbyt nieprecyzyjna – pewne ustępy są zbyt niejasne.*

- Sun Community Source License

*Nie jest licencją wolnego oprogramowania – nie zapewnia podstawowych wolności, takich jak możliwość publikacji zmodyfikowanych wersji.*

- Open Public License

*To nie jest licencja wolnego oprogramowania, gdyż wymaga przesyłania każdej publikowanej zmodyfikowanej wersji konkretnemu początkowemu twórcy. Zawiera też inne sformułowania, których znaczenia nie jesteśmy pewni, a które również mogą być problematyczne.*

- Publiczna Licencja AT&T

*Publiczna Licencja AT&T nie jest wolną licencją. Ma kilka poważnych wad m.in:*

- 1.Licencja udzielona na patent obejmujący algorytm traci ważność jeśli zostanie dokonana jakakolwiek modyfikacja odnośnego kodu, nieważne jak mała.*
- 2.Dystrybuując źródła lub łaty musicie żądać od odbiorcy pisemnej umowy, potwierdzającej zaakceptowanie licencji.*
- 3.Rozpowszechnianie łat wymaga powiadamiania AT&T.*

...

# Licencje dokumentacji

- Licencje Wolnej Dokumentacji:
  - Licencja GNU Wolnej Dokumentacji  
*Przeznaczona do stosowania z wolną dokumentacją typu copyleft*
  - Licencja Dokumentacji FreeBSD (FreeBSD Documentation License)  
*To liberalna licencja Wolnej Dokumentacji. Nie jest typu copyleft, jest zgodna z GNU FDL.*
  - Open Publication License, Version 1.0  
*Możliwe jest jej użycie jako licencji wolnej dokumentacji. Jest licencją wolnej dokumentacji typu copyleft pod warunkiem, że posiadacz praw autorskich nie skorzysta z żadnej z „OPCJI LICENCJI” wymienionych w Części VI licencji. Jeżeli jednak zostanie przywołana któraś z tych opcji, licencja przestanie być wolna.*
- Licencje dokumentacji, która nie jest wolna:
  - Open Content License, w wersji 1.0  
*Ta licencja nie zalicza się do wolnych, ponieważ zawiera ograniczenia dotyczące pobierania pieniędzy za kopie.*
  - Open Directory License (znana też jako Licencja dmoz.org)  
*To nie jest licencja wolnej dokumentacji. Główny problem stanowi to, że prawa do rozprowadzania dalej danej wersji nie są trwałe oraz fakt, iż wymaga od użytkowników ciągłego sprawdzania tej witryny, co jest zbytnim ograniczeniem ich wolności.*

# Licencje dzieł innych niż oprogramowanie i dokumentacja

- Powszechna Licencja Publiczna GNU

*GNU GPL można użyć dla ogólnych danych, nie będących oprogramowaniem, pod warunkiem, że możliwe jest rozstrzygnięcie, do czego w danym przypadku odnosi się definicja „kodu źródłowego”. Jak się okazuje, DSL (zob. niżej) również wymaga określenia, czym jest „kod źródłowy” i używa tej samej w przybliżeniu definicji, co GPL.*

- Licencja GNU Wolnej Dokumentacji

*Można jej użyć dla wszelkiego typu tekstów np. podręczników, czy materiałów szkoleniowych.*

- Design Science License

*Jest to wolna licencja typu copyleft przeznaczona dla ogólnych danych. Jest niezgodna z GNU GPL i GNU FDL, więc niezbyt nadaje się do oprogramowania lub dokumentacji, ale jest bra do innych rodzajów danych.*

- Free Art License

*Jest to wolna licencja typu copyleft przeznaczona dla prac artystycznych. Zezwala na komercyjną dystrybucję, ale każde większe dzieło zawierające pracę objętą tą licencją musi być wolne.*

*Pakiety*



# *Pakiety*

- rpm - Red Hat Package Manager
- deb - Debian
- tgz (tar + gzip) - Slackware
- slp – Stampede Linux

# RPM

**RPM (RPM Package Manager, dawniej Red Hat Package Manager)** to program służący do instalacji i zarządzania pakietami zawierającymi oprogramowanie, oraz nazwa tych pakietów (ponieważ mają one rozszerzenie **.rpm**). Pakiety RPM zawierają skompresowane (we wcześniejszych wersjach gzipem, w nowszych bzipem2) archiwum tar zawierające oprogramowanie. Zawierają także (w specjalnym pliku **.spec**) informacje na temat zawartości, m.in. tzw. *zależności* (ang. *dependencies*) - czyli spis programów lub pakietów, które są wymagane do zainstalowania i poprawnej pracy pakietu (działa to także w drugą stronę - pakiety później zainstalowane wymagające danego pakietu, uniemożliwiają jego odinstalowanie).

Program ten powstał na potrzeby dystrybucji Red Hat Linux, aktualnie jest używany również w innych dystrybucjach (np. SuSE, Mandrake, PLD).

# Deb

**Deb** to skrót od "Deborah" (inaczej: "Debra"), imienia żony twórcy Debiana, Iana Murdocka. Jest to format pakietu instalacyjnego używanego przez Debian GNU/Linux - największą dystrybucję systemu operacyjnego Linux. Wszystkie pakiety w tym formacie posiadają rozszerzenie .deb na końcu nazwy pliku.

Pakiet DEB (i jego instalator dpkg) posiada zaawansowaną kontrolę powiązań i zależności pomiędzy poszczególnymi składnikami systemu - programami i używanymi przez nie bibliotekami, dzięki czemu instalując nowy program ma się pewność, że nie będzie on konfliktował z innymi programami i nie odmówi pracy z powodu braku któregoś ze składników wymaganych przez niego do działania.

W systemie Debian istnieje kilka wygodnych programów do zarządzania zainstalowanymi pakietami. Najczęściej używane z nich to: dselect i aptitude. Warto się z nimi zapoznać, gdyż dystrybucja Debian w wersji stabilnej posiada kilka tysięcy pakietów do wyboru.

---

# SLP

Pakiet SLP składa się z dwóch oddzielnych segmentów: skompresowanego archiwum zmiennego rozmiaru i struktury o ustalonym rozmiarze, która zawiera informacje kontrolne.

Skompresowane archiwum reprezentuje początek pakietu SLP. Do generowania archiwum używane są Gnu tar, a następnie bzip2. Standardowe narzędzia 'tar' i 'bzip2' mogą być użyte do wyciągnięcia plików z pakietu; wszystkie informacje znajdujące się w pakiecie po skompresowanym archiwum zostaną pominięte.

Zawartość archiwum pakietu SLP zależy wyłącznie od polityki dystrybucji i generalnie jest poza specyfikacją pakietu SLP. Każda dystrybucja jest odpowiedzialna za utrzymywanie swoich własnych wskazówek dotyczących rozkładu plików w archiwum, ich wydobywania, kompresji dokumentacji i wielu innych informacji zawartych w pakiecie.

Nagłówek SLP jest doklejony do końca skompresowanego archiwum. Nagłówek ma stały rozmiar, stałą strukturę zajmującą 3785 bajtów. Struktura ta zawiera 19 pól o ustalonym rozmiarze.

Możliwe jest stworzenie pakietu, który zawiera sam nagłówek pakietu SLP, a nie zawiera skompresowanego archiwum. Takie pliki noszą nazwę odczepionych plików z nagłówkami SLP. Pliki te mogą być tworzone z prostych powodów albo jako potencjalne szablony, które w razie potrzeby można by dokleić do końca skompresowanego archiwum.



# ***Bezpieczeństwo, autoryzacja i veryfikacja***

	DEB	RPM	TGZ	SLP
Pakiety podpisane przez PGP	Nie	Tak	Nie	Nie
Sumy kontrolne	Tak	Tak	Nie	Nie
Prawa dostępy, właściciele, itp.	Tak	Tak	Tak	Tak

# Funkcjonalność

	DEB	RPM	TGZ	SLP
Rozpoznawany przez file	Tak	Tak	Nie	Nie
Rozpakowanie przez zwykłe narzędzia	Tak	Nie	Tak	Tak*
Metadane dostępne przez zwykłe narzędzia	Tak	Nie	N/A	Nie
Tworzenie przez zwykłe narzędzia	Nie	Nie	Tak	Nie

\* - jeżeli uznamy bunzip2 za standardowe narzędzie

# Metadane

	DEB	RPM	TGZ	SLP
Zależności	Tak	Tak	Nie	Tak
Rekomendacje	Tak	Nie	Nie	Nie
Sugestie	Tak	Nie	Nie	Nie
Konflikty	Tak	Tak	Nie	Tak
Wirtualne pakiety i dostawcy	Tak	Tak	Nie	??
Wersjonowane zależności i konflikty	Tak	Tak	Nie	??
Kompleksowe zależności	Tak	Nie	Nie	Nie
Zależności od plików	Nie	Tak	Nie	Nie
Copyright	Nie	Tak	Nie	Tak
Grupowanie	Tak	Tak	Nie	Tak
Priorytety	Tak	Nie	Nie	Tak

# Specjalne pliki

	DEB	RPM	TGZ	SLP
Pliki konfiguracyjne	Tak	Tak	Nie	Tak
Pliki z dokumentacja	Nie	Tak	Nie	Nie

# Programy pakietowe

	DEB	RPM	TGZ	SLP
Czy dozwolone programy	Tak	Nie	??	Tak
Czy dozwolone programy interaktywne	Tak	Nie	??	??
Programy przedinstalacyjne	Tak	Tak	Nie	Nie
Programy podinstalacyjne	Tak	Tak	Tak	Tak
Programy przed usunięciem	Tak	Tak	Nie	Nie
Programy po usunięciu	Tak	Tak	Tak	Nie
Programy weryfikujące	Nie	Tak	Nie	Nie
Wyzwalacze	Nie	Tak	Nie	Nie

# Skalowalność

	DEB	RPM	TGZ	SLP
Brak silnych ograniczeń	Tak	Tak	Tak	Nie
Nowe metadane	Tak	Tak	N/A	Nie
Nowe sekcje	Tak	Nie	Nie	Nie
Format wersji	Tak	Tak	Nie	Tak

# *Zarządzanie pakietami*



# *Zarządzanie pakietami*

- Pacman (Arch Linux)
  - APT (Debian)
  - Up2date, yum (Fedora)
  - Emerge/Portage (Gentoo)
  - Urpmi (Mandriva Linux)
  - Swaret, Slaptget (Slackware)
  - YaST2 (SUSE Linux)
- 
-



# *apt (Advanced Packaging Tool)*

## Zalety

- Automatyczne ściąganie pakietu
- Korzystanie z repozytorium
- Pozwala na szybki upgrade systemu

## Wady:

- został zaprojektowany do współpracy z plikami .deb jedynie na systemie Debian

Wykorzystywany w: Debian

---

# *yum (Yellow Dog Updater)*

Zalety

- Mniejszy kod źródłowy od apt

Wady:

- Wolne działanie
- Obciążenie sieci (pobieranie informacji o pakietach i ich zależnościach)
- Brak oficjalnego GUI

Wykorzystywany w: Fedora Core, Aurox

---

# *YaST2 (Yet Another Setup Tool)*

## Zalety

- Instalator + narzędzie konfiguracyjne (dość kompletne i właściwe narzędzie do administracji instalacją)
- Interface graficzny i tekstowy

## Wady:

- póki co tylko w jednej dystrybucji (od niedawna na licencji GPL)

Wykorzystywany w: SuSE Linux

---

# *Urpmi*

## Zalety

- Wykorzystuje RPM-y
- Łatwy w obsłudze

## Wady:

- Brak interface-u tekstowego

Wykorzystywany w: Mandriva

---

---

# *Emerge/Portage*

## Zalety

- Drzewo Ebuild-ów
- Lokalne drzewo repozytorium
- Flagi USE

## Wady:

- Brak obsługi prekompilowanych paczek oprogramowania

Wykorzystywany w: Gentoo

---

# *Środowiska Graficzne*



# *Środowiska Graficzne*

- KDE
- GNOME
- Window Maker
- Xfce
- Enlightenment
- ...



# ***KDE (K Desktop Environment)***



- Zawiera bogaty systemem bibliotek do tworzenia aplikacji w środowisku graficznym
- KDE korzysta z biblioteki graficznej Qt
- KDE korzysta z systemu obiektowego Kparts4
- W KDE menedżerem okien jest KWin, jednak aplikacje można uruchamiać także pod innymi menedżerami.
- Aplikacje korzystające z KDE często mają nazwy rozpoczynające się od K np.: Konqueror, Kate, K3b



# ***GNOME (GNU Network Object Model Environment)***



- Śraficznym środowiskiem użytkownika opartym na systemie X Window System oraz widgetach GTK+ i GDK
- Silny nacisk położony jest na prostotę oraz wygodę obsługi
- Środowisko GNOME składa się z wielu mniejszych projektów. Najistotniejsze z nich: ATK, Bonobo, Gconf, GNOME VFS, GNOME Keyring, GNOME Print, GStreamer, GTK+, LibXML, ORBit, Pango, HIG

# Window Maker



## Zalety:

- niewielkie zapotrzebowanie na pamięć operacyjną
- niestandardowy sposób obsługi ( à la NeXTStep)
- prostota obsługi
- estetyczny wygląd
- możliwość przyporządkowywania aplikacjom parametrów okien
- prosta konfiguracja

## Wady:

- niestandardowy sposób obsługi ( à la NeXTStep)
- błędy w ustawieniach (przyporządkowywania aplikacjom parametrów okien)

# Xfce



- Środowisko graficzne dla systemów uniksowych, wzorowane na CDE.
  - Bazuje na bibliotece GTK+
  - Wszystkie pliki konfiguracyjne są napisane w XML-u
  - Xfce jest złożone z wielu modułów, między innymi z menedżera okien Xfwm, które w połączeniu oferują w pełni funkcjonalne środowisko.
  - Xfce może też korzystać z modułów innych środowisk graficznych zamiast swoich własnych.
- 
-

# *Enlightenment*



- może być używany samodzielnie lub wraz ze środowiskami graficznymi takimi jak GNOME czy KDE
- duże możliwości konfiguracji
- atrakcyjna grafika i efekty specjalne.



# *Przegląd najważniejszych dystrybucji Linuksa*



# *Przeznaczenia*

Czyli do czego chcemy tego całego Linuksa używać?

(Czyli jeszcze inaczej – które rzeczy są dla nas najważniejsze przy wybieraniu dystrybucji?)

- Entuzjaści
- Serwery
- Stacje robocze
- Inne, specjalne zastosowania



# *Główne dystrybucje*

- Ubuntu
- Mandriva
- SUSE
- Fedora Core
- Debian
- Gentoo
- Slackware
- Knoppix
- MEPIS
- Xandros
- ....



### Własności:

- Ubuntu jest zbudowany na podstawie Debiana Sid (gałąź niestabilna)
- Dystrybucje zaprojektowano do wykorzystania na desktopach i serwerach z obsługą aktualizacji z Debiana 3.0 (Woody)
- domyślnie wyłączone jest konto "root"; pierwszy zarejestrowany użytkownik po instalacji otrzymuje przywileje superużytkownika poprzez polecenie "sudo". Pomysłowi temu, choć na początku wygląda niewygodnie, trzeba przyklasnąć, ponieważ namawia do dobrych praktyk bezpieczeństwa.

Plusy: Świetna społeczność deweloperów i użytkowników; stały harmonogram wydań i pomocy; udostępniona ścieżka aktualizacji ze stabilnej dystrybucji Debiana.

Minusy: Autorzy Ubuntu nie stworzyli na razie modelu biznesowego dystrybucji.

Obsługa pakietów oprogramowania: APT (DEB)

Darmowe pobieranie: Tak



# Mandriva



## Własności:

- Proces rozwijania Mandrivy jest całkowicie otwarty i przejrzysty.
- Nowe pakiety pojawiają się raz dziennie w tzw. katalogu "cooker". Gdy nowe wydanie wchodzi w fazę beta, to co znajduje się w katalogu cooker jest przyjmowane jako pierwsza beta.
- bardzo aktualna wersja dystrybucji Linuksa.
- więcej błędów i mniejszą stabilność w porównaniu do innych dystrybucji

Plusy: Dystrybucja przyjazna dla użytkownika, graficzne narzędzia konfiguracyjne, doskonałe wsparcie ze strony społeczności, zmiana rozmiaru partycji NTFS.

Minusy: Niektóre wydania są z błędami, wydania są najpierw dostępne tylko dla członków MandrivaClub.

Obsługa pakietów oprogramowania: urpmi (RPM)

Darmowe pobieranie: Instalacja przez FTP dostępna zaraz po wydaniu, ISO po kilku tygodniach.

# *SUSE Linux*



Własności:

- Proces tworzenia dystrybucji SUSE odbywa się za całkiem zamkniętymi drzwiami.
- Nie ma dostępnych do testowania publicznych wersji beta.
- dostęp do wersji bezpłatnej możliwy jest dopiero po upływie 1-2 miesięcy po trafileniu wersji pudełkowej na półki sklepowe.

Plusy: Profesjonalne skupianie się na szczegółach, łatwy w użyciu konfigurator YaST.

Minusy: Dostępna jedynie w niektórych częściach świata od sprzedawców oprogramowania lub instalacja poprzez FTP, zawiera prawnie chronione składniki ograniczające redystrybuowanie.

Obsługa pakietów oprogramowania: YaST (RPM), dostępne inne repozytoria APT (RPM)

Darmowe pobieranie: obrazy pojawiły się 2 miesiące po oficjalnym wydaniu. Edycja Professional dostępna jest do instalacji przez FTP zazwyczaj 1-2 miesiące po oficjalnym wydaniu.



### Własności:

- Fedora jest oficjalnie sponsorowana przez Red Hata, a jej rozwijanie odbywa się przy współpracy społeczności i głównie wykorzystywana jest jako platforma testowa dla Red Hat Enterprise Linux.
- Pakiety nie są zbyt aktualne; gdy ogłaszana jest nowa wersja beta, wersje pakietów są "zamrażane" oprócz aktualizacji dotyczących bezpieczeństwa. Wynikiem tego jest dobrze przetestowana i stabilna dystrybucja
- program beta i zgłaszanie błędów jest publiczne

Plusy: Szeroko stosowany, doskonała pomoc ze strony społeczności, wiele innowacji.

Minusy: Ograniczona długość życia dystrybucji darmowych, słaba obsługa multimediów, sprawa przechodzenia Red Hata na Fedorę

Obsługa pakietów oprogramowania: up2date (RPM), YUM (RPM)

Darmowe pobieranie: Tak

# Debian



Własności:

- Projekt posiada doskonałe zarządzanie
- W każdym momencie procesu tworzenia dystrybucji, w głównym katalogu znajdują się trzy gałęzie - "stabilna", "testowa" i "niestabilna".
- W rezultacie procesu twórczego powstaje jak najbardziej stabilna i niezawodna acz niezbyt aktualna dystrybucja.
- narzędzie "apt-get"

Plusy: Całkowicie darmowy, doskonałe materiały na stronach i wśród społeczności, doskonale przetestowany, bezbolesna instalacja oprogramowania z apt-get.

Minusy: Powolny cykl wydawniczy, przeterminowana wersja stabilna.

Obsługa pakietów oprogramowania: APT (DEB)

Darmowe pobieranie: Tak



Własności:

- Gentoo Linux to dystrybucja oparta o pakiety źródłowe.
- idea Gentoo polega na kompilacji pakietów źródłowych na komputerze użytkownika.
- Podstawową zaletą tego jest to, że całe oprogramowanie jest zoptymalizowane do architektury komputera na którym zostało zbudowane.
- pakiety oprogramowania są trzymane w centralnym repozytorium i zazwyczaj są aktualne i dostępne w przeciągu kilku dni (czasem w przeciągu kilku godzin) od momentu wypuszczenia ich przez zewnętrznych deweloperów.

Plusy: Bezbolesna instalacja pojedynczych pakietów oprogramowania, wysoce aktualna, wspaniała dokumentacja, poczucie "geeka" przy budowaniu dystrybucji dopasowanej do potrzeb użytkownika.

Minusy: Długotrwała i nudna instalacja systemu, sporadyczna niestabilność i ryzyko padu.

Obsługa pakietów oprogramowania: Portage (SRC)

Darmowe pobieranie: Tak

---

---

# Slackware



## Własności:

- najstarszą istniejącą dystrybucja Linuksa
- Nie oferuje wodotrysków, trzymając się tekstowego instalatora i niegraficznych narzędzi konfiguracyjnych
- Jest bardzo stabilny i bezpieczny
- doskonały do zastosowań serwerowych.
- korzysta z wielu pakietów w ich pierwotnym kształcie i bez zbyt wielu własnych rozszerzeń
- Wydania są niezbyt częste (około raz do roku), choć najnowsze pakiety są cały czas dostępne do ściągnięcia po wydaniu dystrybucji.

Plusy: Wysoce stabilna i wolna od błędów, ściśle związana z zasadami UNIXa.

Minusy: Cała konfiguracja odbywa się poprzez edycję plików tekstowych, ograniczone automatyczne rozpoznawanie sprzętu.

Obsługa pakietów oprogramowania: Slackware Package Management (TGZ)

Darmowe pobieranie: Tak



Własności:

- dystrybucja Knoppix oparta o Debiana
- Nowy standard w dziedzinie automatycznego rozpoznawania sprzętu
- "bezdotykowe" uruchamianie
- spory wybór oprogramowania
- technika dekompresji w locie
- zdolność do instalacji na dysku twardym sprawia

Plusy: Doskonałe automatyczne rozpoznawanie sprzętu, działa bezpośrednio z płyty CD bez potrzeby instalacji na dysku twardym, może być używany jako narzędzie do odzyskiwania danych.

Minusy: Wolny i niewydajny jeśli uruchamiany bezpośrednio z płyty CD.

Obsługa pakietów oprogramowania: APT (DEB)

Darmowe pobieranie: Tak

---

---



### Własności:

- połączenie Debiana Sida i Knoppiksa
- może być używana zarówno jako Live CD jak i w pełni funkcjonalna dystrybucja instalowana na dysku twardym
- MEPIS dostarczana wiele niewolnych, acz w pełni użytecznych aplikacji, wszystkie prekonfigurowane i gotowe do użycia
- nie trzeba spędzać godzin nad konfiguracją poinstalacyjną i dostosowywaniem systemu.
- doskonały system automatycznego wykrywania sprzętu

Plusy: Pełna dystrybucja i Live CD w jednym, posiada wbudowane sterowniki do kart graficznych, Flash, Java, kodeki, i inne niewolne oprogramowanie, świetny system autodetekcji sprzętu.

Minusy: Niezbyt dojrzała dystrybucja, deweloperzy powinni zastanowić się nad zatrudnieniem grafika, aby ulepszyć domyślny wygląd dystrybucji.

Obsługa pakietów oprogramowania: APT (DEB)

Darmowe pobieranie: Tak

---



# ***Bibliografia***

- <http://pl.wikipedia.org/wiki/POSIX>
- <http://en.wikipedia.org/wiki/POSIX>
- [http://hedera.linuxnews.pl/\\_news/2003/11/28/\\_long/2339.shtml](http://hedera.linuxnews.pl/_news/2003/11/28/_long/2339.shtml)
- <http://www.unix.org/version3/>
- <http://www.opengroup.org/onlinepubs/009695399/>
- [http://pl.wikipedia.org/wiki/Single\\_UNIX\\_Specification](http://pl.wikipedia.org/wiki/Single_UNIX_Specification)
- <http://www.linuxbase.org/>
- [http://pl.wikipedia.org/wiki/Linux\\_Standard\\_Base](http://pl.wikipedia.org/wiki/Linux_Standard_Base)
- [http://pl.wikipedia.org/wiki/Dystrybucje\\_Linuxa](http://pl.wikipedia.org/wiki/Dystrybucje_Linuxa)
- <http://www.gnu.org/licenses/license-list.pl.html#TOCGPLCompatibleLicenses>
- [http://pl.wikipedia.org/wiki/Wolne\\_oprogramowanie](http://pl.wikipedia.org/wiki/Wolne_oprogramowanie)
- <http://pl.wikipedia.org/wiki/GPL>
- <http://distrowatch.com/dwres.php?resource=major>
- <http://www.kitenet.net/~joey/pkg-comp/>
- <http://pl.wikipedia.org/wiki/KDE>
- <http://pl.wikipedia.org/wiki/Gnome>
- <http://pl.wikipedia.org/wiki/Rpm>
- <http://pl.wikipedia.org/wiki/Deb>
- <http://pl.wikipedia.org/wiki/Xfce>
- [http://pl.wikipedia.org/wiki/Window\\_Maker](http://pl.wikipedia.org/wiki/Window_Maker)
- <http://pl.wikipedia.org/wiki/Enlightenment>
- [http://pl.wikipedia.org/wiki/Linux\\_From\\_Scratch](http://pl.wikipedia.org/wiki/Linux_From_Scratch)
- <http://www.linuxfromscratch.org/>