

# Linux is linux is linux?

Zaprezentuję subiektywnie wybrane rozwiązania stosowane w różnych dystrybucjach linuksa i ideologie przyświecające ich twórcom:

- ✓ sposoby zarządzania oprogramowaniem
- ✓ trudny wybór pomiędzy aktualnością oprogramowania a jego stabilnością
- ✓ problem unifikacji środowisk graficznych

# Problemy z tworzeniem oprogramowania dla linuxa (zwłaszcza komercyjnego)

- windowsów jest kilka, góra kilkanaście: XP Server, XP Prof, XP Home, 2000, 98, NT, 95... ewentualnie razy 3-4 service packi; linuxów tysiące, i to grube – kto policzy, ile jest różnych możliwości kompilacji jądra? w dodatku pomnożonych przez ilość wersji jajka, plus częściej używane patche...
- każda dystrybucja ma swój układ katalogów i swoje własne pliki konfiguracyjne
- w świecie wolnego oprogramowania nowe wersje ukazują się dużo częściej niż w świecie okien
- nie istnieje żaden uniwersalny system instalacji programów w postaci skompilowanej

*Przykład ratowania się przed tą mnogością: oracle 10g.*

Producent tego programu stworzył potężny instalator sprawdzający mnóstwo drobnych ustawień jądra i systemu; co więcej, nie pozwala on na instalację poza RedHatem EL i SuSE EL

# Ręczna instalacja ze źródeł czyli ./configure; make; make install

## Zalety:

- jest uniwersalną metodą instalacji
- pozwala idealnie dostosować program do własnych potrzeb (wkompilować tylko pożądane komponenty)

## Wady:

- często wymaga niestandardowych czynności przed instalacją
- kompilacja źródeł zajmuje dużo czasu
- brak informacji o zależnościach
- brak możliwości automatycznej weryfikacji zainstalowanego oprogramowania

# Pakiety binarne

- zawierają oprogramowanie skompilowane przez producenta dystrybucji
- określają zależności
- definiują dodatkowe czynności przed i po instalacji
- obecnie używane: rpm, deb, tgz (Slackware'owy)

## Wady:

- są trudne do przenoszenia pomiędzy dystrybucjami, nawet jeśli używają tego samego systemu PMS (a czasem nawet w samej dystrybucji – vide moduły jądra)
- programy kompilowane są z narzuconymi przez producenta opcjami

# Opis rpm i deb

rpm:

- menedżer zarządzający: *rpm*
- największe dystrybucje: Fedora / RedHat Enterprise, SuSE, Mandriva
- programy zaspakajające zależności: yum (FC), yast (SuSE), urpmi (Mandriva)
- przeportowany apt

deb:

- menedżer: *dpkg*
- największe dystrybucje: Debian, Ubuntu
- program zaspokajający zależności: apt
- istnieje *alien* zamieniający proste pakiety

# Problemy przy przenoszeniu pakietów

- każda dystrybucja posiada swój własny układ katalogów i zestaw bibliotek
- dystrybucje mogą korzystać z własnych formatów plików konfiguracyjnych
- różne dystrybucje mogą stosować własne nazwy dla pakietów (problemy z zależnościami) i umieszczać w nich różne komponenty
- pakiety z modułami jądra: każde jądro korzysta wyłącznie z modułów skompilowanych dla niego

# Repozytoria, narzędzia rozwiązywania konfliktów między pakietami

**repozytorium** – zbiór pakietów nie stwarzający konfliktów, wraz z dodatkowymi informacjami nt. zależności, umieszczony np. w internecie

- istnieją repozytoria zależne od innych repozytoriów, to znaczy zaspokajające wszystkie zależności wspólnie z pakietami z innego miejsca

narzędzia rozwiązujące konflikty między pakietami:

- pozwalają na instalację, aktualizację i usuwanie oprogramowania w systemie bez znajomości powiązań przez użytkownika
- korzystają z bazy pakietów w systemie (np. rpm, deb) oraz repozytoriów samemu wybierając bezkonfliktowe rozwiązanie

# Przykład: apt wraz nakładką graficzną - synaptikiem

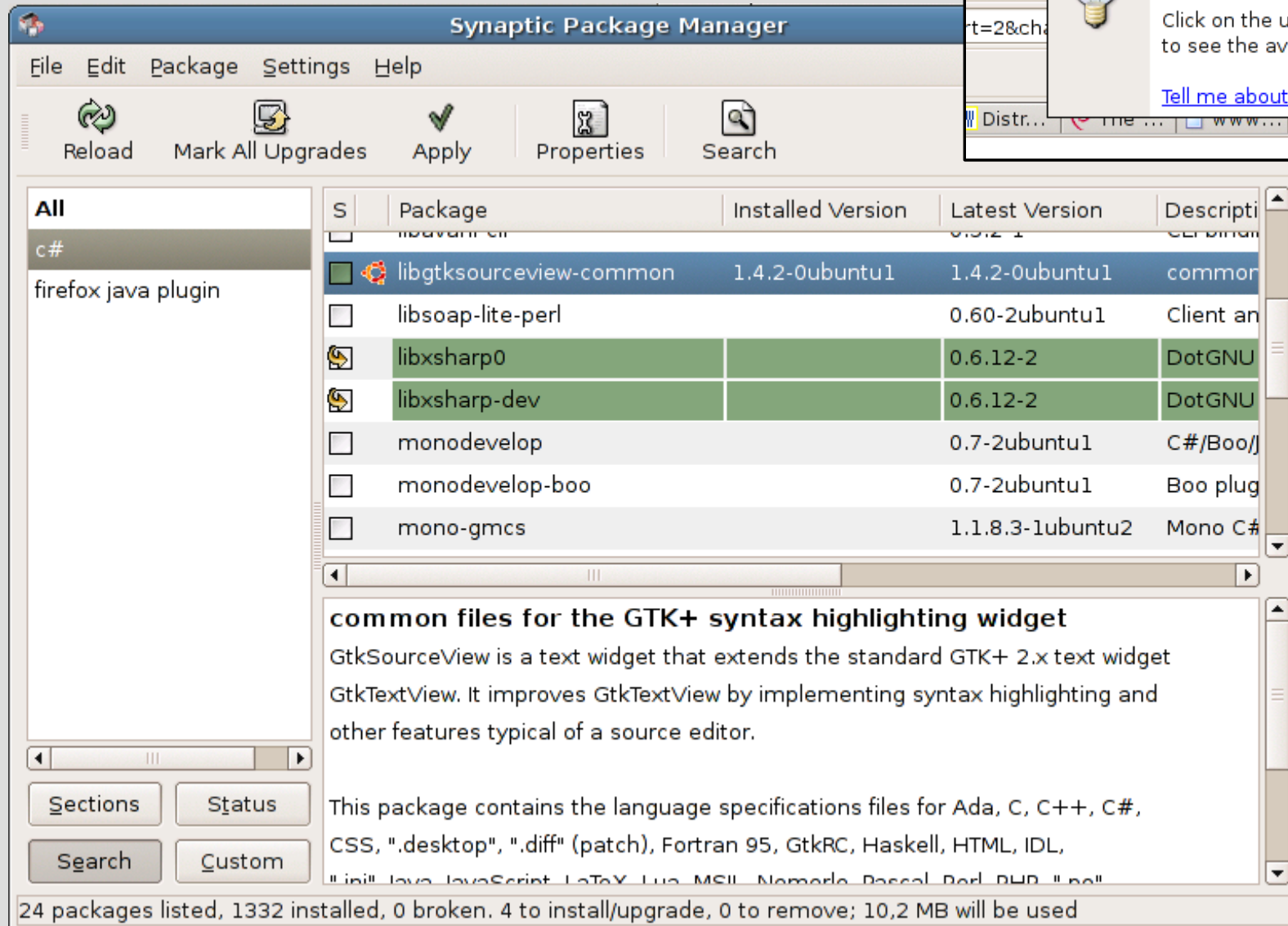
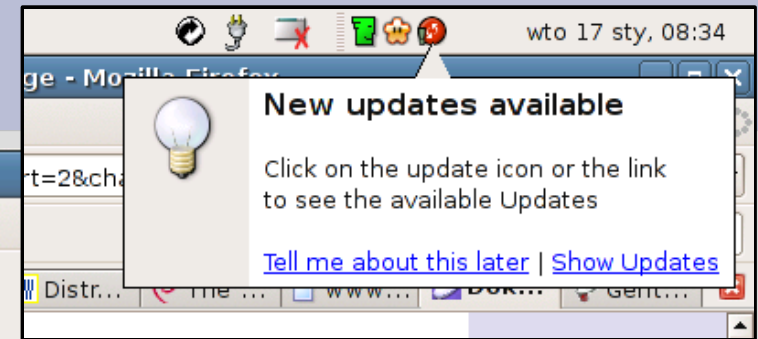
możliwości apta:

- instalacja, uaktualnianie, usuwanie pakietu
- uaktualnienie całego systemu (dist-upgrade, używany jest inny algorytm, niż przy pojedynczych pakietach)
- wyświetlanie informacji o pakietach w repozytorium
- przeszukiwanie repozytorium dla zadanego kryterium
- dostępny dla systemu deb, przeportowany na rpm (z rpmem działa wyraźnie wolniej)

apt informacje o plikach repozytorium przechowuje lokalnie na dysku



# Praca z synaptikiem



S	Package	Installed Version	Latest Version	Description
<input type="checkbox"/>	libavahi-client	0.7-2ubuntu1	0.7-2ubuntu1	Client-side
<input checked="" type="checkbox"/>	libgtksourceview-common	1.4.2-0ubuntu1	1.4.2-0ubuntu1	common
<input type="checkbox"/>	libsoap-lite-perl		0.60-2ubuntu1	Client an
<input checked="" type="checkbox"/>	libxsharp0		0.6.12-2	DotGNU
<input checked="" type="checkbox"/>	libxsharp-dev		0.6.12-2	DotGNU
<input type="checkbox"/>	monodevelop		0.7-2ubuntu1	C#/Boo/J
<input type="checkbox"/>	monodevelop-boo		0.7-2ubuntu1	Boo plug
<input type="checkbox"/>	mono-gmcs		1.1.8.3-1ubuntu2	Mono C#

**common files for the GTK+ syntax highlighting widget**

GtkSourceView is a text widget that extends the standard GTK+ 2.x text widget GtkTextView. It improves GtkTextView by implementing syntax highlighting and other features typical of a source editor.

This package contains the language specifications files for Ada, C, C++, C#, CSS, ".desktop", ".diff" (patch), Fortran 95, GtkRC, Haskell, HTML, IDL, ".ini", Java, JavaScript, LaTeX, Lua, MSIL, Nemerle, Pascal, Perl, PHP, ".po",

24 packages listed, 1332 installed, 0 broken. 4 to install/upgrade, 0 to remove; 10,2 MB will be used

# Gentoo: pliki ebuild

- zupełnie odmienne podejście: zapożyczone z BSD **drzewo Portage** (/usr/portage) zawierające zestaw **plików ebuild** mówiących, w jaki sposób zarządzać oprogramowaniem
- dla każdego programu jeden lub więcej ebuildów
- plik ebuild to zwykłe skrypty napisane w bashu
- w systemie mogą koegzystować różne wersje tego samego programu (technika slotów)
- zależności podobne, jak w przypadku rpmów i debów
- **ebuildy nie instalują prekompilowanych plików binarnych, lecz kompilują programy ze źródeł**
- program do zarządzania – *emerge*

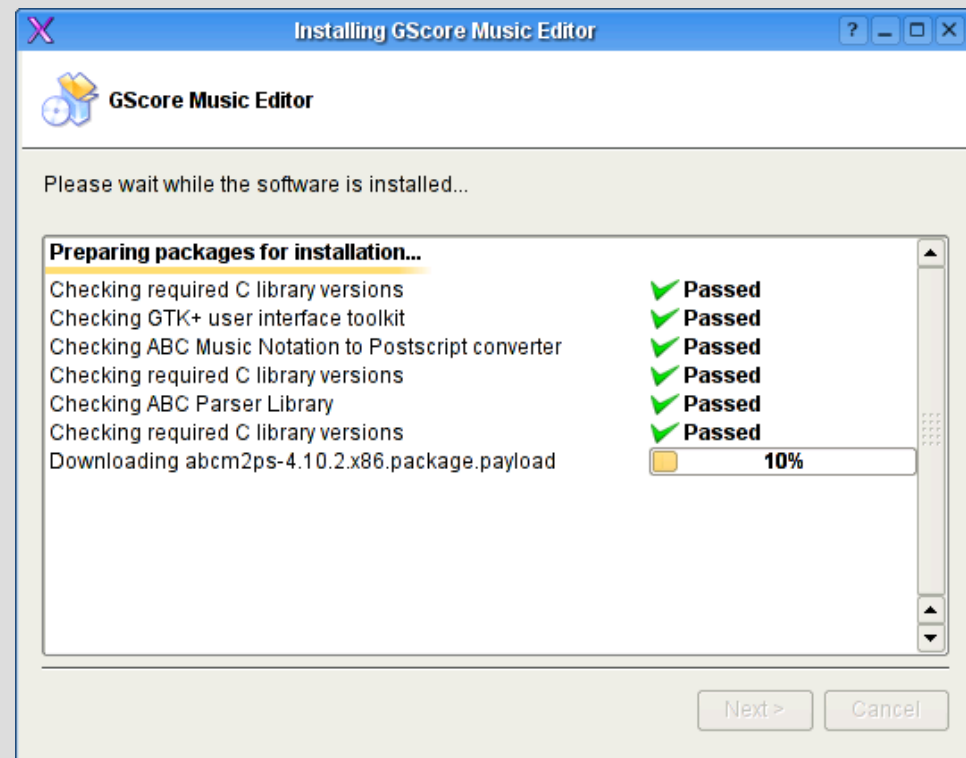
# Pliki binarne czy instalacja ze źródeł?

Jak zwykle nie ma jedynej słusznej odpowiedzi a siła linuxa tkwi w jego różnorodności!

W szczególnych przypadkach w systemie koegzystować mogą programy ręcznie kompilowane oraz zarządzane przez PMS (Package Management System), lecz nie jest to sytuacja pożądana – prowokuje kłopoty przy uaktualnianiu systemu i zachowaniu monitoringu łańcuch bezpieczeństwa

# Próba unifikacji: autopackage

- uzupełnienie istniejącego w systemie menedżera pakietów
- uniwersalne narzędzie do instalacji binarnego oprogramowania w dowolnej dystrybucji
- potrafi dociągnąć z sieci potrzebne pakiety
- obecnie (wersja 1.0) posiada wygodne interfejsy: tekstowy, dla GNOME i KDE
- nowe rozwiązanie: instaluje wszystkie potrzebne biblioteki, choćby miał zainstalować wiele wersji tej samej
- (dopiero) w planach współpraca z głównym menedżerem pakietów
- używa własnego formatu: tar.gz osadzone w bashu



# Nowoczesność czy stabilność?

## Porównanie ważniejszych debiana i ubuntu

	Debian	Ubuntu
ostatnie wydanie stabilne	3.1 (sarge) 2005/06/06	5.10 (breezy) 2005/10/13
poprzednie wydanie stabilne	3.0 (woody) 2002/07/19	5.04 (hoary) 2005/04/08
kernel	2.4.27	2.6.12
apache (httpd)	2.0.54	2.0.54
gcc	3.3.5	4.0.1
iptables	1.2.11	1.3.1
libgnome	2.8.1	2.12.0.1
mysql	4.0.24	4.1.12
openssh	3.8.1p1	4.1p1
okienka	Xfree86 4.3.0	x.org 6.8.2

# A może linux komercyjny?

## Porównanie Fedory i RedHat EL

	Fedora	RedHat EL
osoby rozwijające	RedHat + społeczność	RedHat
model rozwoju	Open Source	Open Source
certyfikaty software	brak	m.in. BEA, CA, IBM, Oracle
certyfikaty hardwr.	brak	>1000: Bell, Dell, HP, IBM
testerzy	społeczność	RedHat, partnerzy, firmy zewnętrzne
częstotliwość nowych wersji	4-6 m-cy	ok. 18 m-cy
czas wspierania wersji (poprawki)	poniżej roku	7 lat
support	społeczność (google:)	wiele opcji, włączając 24x7
cena	brak	od 6 USD / miesiąc

# Freedesktop.org

## czyli projekt zunifikowanych okienek

postanowienia statutowe:

- zgromadzić w jednym miejscu specyfikacje, standardy i dokumenty związane z unifikacją środowiska graficznego
- promowanie takich standardów
- integracja standardów w większe
- praca nad ich implementacją w konkretnych menedżerach

niektóre z projektów:

- Desktop VFS – wirtualny system plików dla aplikacji GUI
- GTK-Qt – wspólny dla KDE i GNOME zestaw widgetów
- Gstreamer – strumieniowy interfejs multimedialny
- HAL – Hardware Abstraction Layer