

APT HOWTO (Obsolete Documentation)

Gustavo Noronha Silva <kov@debian.org>

Přeložil Aleš Nosek <noseka1@centrum.cz>

1.8.11 - srpen 2005

Souhrn

Tento dokument uživatele podrobně seznamuje s činností nástroje pro správu balíčků distribuce Debian, APT. Jeho cílem je usnadnit život novým uživatelům Debianu a pomoci těm, kteří si chtějí prohloubit znalosti administrace tohoto systému. Dokument byl napsán pro projekt Debian pro zlepšení podpory dostupné uživatelům této distribuce.

Autorská práva

Copyright © 2001, 2002, 2003, 2004 Gustavo Noronha Silva

Tento manuál je svobodným software; můžete jej redistribuovat a/nebo upravovat, pokud dodržíte podmínky uvedené v GNU General Public Licence vydané Free Software Foundation; buď ve verzi 2 nebo (dle vašeho výběru) jakékoliv pozdější verzi.

Manuál je šířen v naději, že bude užitečný, ovšem na jeho použití se neposkytují žádné záruky; dokonce ani předpokládané záruky obchodovatelnosti nebo vhodnosti pro konkrétní účel. Viz GNU General Public License.

Kopie GNU General Public License je dostupná jako `/usr/share/common-licenses/GPL` v distribuci Debian GNU/Linuxu nebo na [www](http://www.gnu.org/licenses/) na GNU General Public Licence. Můžete si o ni rovněž napsat na adresu Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Obsah

| | |
|---|-----------|
| 1 Úvod | 1 |
| 2 Základní nastavení | 3 |
| 2.1 Soubor <code>/etc/apt/sources.list</code> | 3 |
| 2.2 Lokální použití APT | 4 |
| 2.3 Výběr nejlepších zrcadel do souboru <code>sources.list</code> : <code>netselect</code> , <code>netselect-apt</code> | 5 |
| 2.4 Přidání CD-ROM do souboru <code>sources.list</code> | 6 |
| 3 Správa balíčků | 7 |
| 3.1 Aktualizace seznamu dostupných balíčků | 7 |
| 3.2 Instalace balíčků | 7 |
| 3.3 Odstraňování balíčků | 9 |
| 3.4 Upgrade balíčků | 10 |
| 3.5 Upgrade na nové vydání Debianu | 11 |
| 3.6 Odstránění nepoužívaných souborů balíčků: <code>apt-get clean and autoclean</code> | 13 |
| 3.7 Použití APT spolu s <code>dselect</code> | 14 |
| 3.8 Jak udržovat smíšený systém | 15 |
| 3.9 Jak upgradovat balíčky z konkrétních verzí Debianu | 16 |
| 3.10 Jak udržet nainstalované určité verze balíčků | 16 |
| 4 Velmi užiteční pomocníci | 19 |
| 4.1 Jak instalovat lokálně kompilované balíčky: <code>equivs</code> | 19 |
| 4.2 Odstránění nepoužívaných lokalizačních souborů: <code>localepurge</code> | 21 |
| 4.3 Jak poznat, které balíčky lze upgradovat | 21 |

| | | |
|-----------|--|-----------|
| 5 | Získávání informací o balíčcích | 23 |
| 5.1 | Zjišťování názvů balíčků | 23 |
| 5.2 | Použití dpkg k nalezení názvů balíčků | 25 |
| 5.3 | Jak instalovat balíčky “na vyžádání” | 26 |
| 5.4 | Jak zjistit, ke kterému balíčku soubor patří | 27 |
| 5.5 | Jak zjistit změny v balíčcích | 27 |
| 6 | Práce se zdrojovými balíčky | 29 |
| 6.1 | Stažení zdrojových balíčků | 29 |
| 6.2 | Balíčky nutné pro kompilaci zdrojového balíčku | 30 |
| 7 | Jak se vypořádat s chybami | 31 |
| 7.1 | Běžné chyby | 31 |
| 7.2 | Kde najdu pomoc? | 32 |
| 8 | Které distribuce podporují APT? | 33 |
| 9 | Credits | 35 |
| 10 | Nové verze tohoto návodu | 37 |

Kapitola 1

Úvod

Na začátku byl `.tar.gz`. Uživatelé museli překládat každý program, který chtěli na svém GNU/Linux systému používat. Když byl vytvořen Debian, pokládalo se za nezbytné, aby systém obsahoval metodu pro správu instalovaných balíčků. Systém pro správu balíčků byl nazván `dpkg`. Takto slavný 'package' (balíček) prvně vstoupil do GNU/Linuxu, dlouho před tím, než se Red Hat rozhodl vytvořit svůj vlastní 'rpm' systém.

Tvůrci GNU/Linuxu museli brzy řešit nové dilema. Potřebovali rychlou, praktickou a výkonnou metodu instalace balíčků, která by automaticky řešila závislosti a při upgradu se postarala o konfigurační soubory. V tomto opět Debian ukázal cestu, když vytvořil APT - Advanced Packaging Tool, který byl potom pro použití s rpm portován na Conectiva a další distribuce.

Tento manuál neobsahuje popis `apt-rpm`, tj. portu APT na Conectiva Linux. Případné doplnění je však vítáno.

Tento manuál je založen na příští verzi Debianu, *Sarge*.

Kapitola 2

Základní nastavení

2.1 Soubor `/etc/apt/sources.list`

Pro svou činnost používá APT soubor, který je seznamem 'zdrojů', ze kterých se balíčky získávají. Tímto souborem je `/etc/apt/sources.list`.

Položky v tomto souboru mají obvykle tento formát:

```
deb http://host/debian distribution section1 section2 section3
deb-src http://host/debian distribution section1 section2 section3
```

Tyto položky jsou samozřejmě pouze ukázkové a v praxi je nelze použít. První slovo na řádce, `deb` nebo `deb-src`, označuje typ archivu: zda obsahuje binární balíčky (`deb`), tj. předkompilované balíčky které normálně používáme, nebo zdrojové balíčky (`deb-src`), což jsou původní zdrojové kódy programu spolu s debianím řídicím souborem `.dsc` a souborem `diff.gz`, který popisuje změny, které je potřeba provést k 'debianizaci' programu.

Implicitní `sources.list` distribuce Debian vypadá obvykle takto:

```
# See sources.list(5) for more information, especially
# Remember that you can only use http, ftp or file URIs
# CDROMs are managed through the apt-cdrom tool.
deb http://http.us.debian.org/debian stable main contrib non-free
deb http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free
deb http://security.debian.org stable/updates main contrib non-free

# Uncomment if you want the apt-get source function to work
#deb-src http://http.us.debian.org/debian stable main contrib non-free
#deb-src http://non-us.debian.org/debian-non-US stable/non-US main contrib no
```

Tyto řádky potřebuje základní instalace Debianu. První `deb` řádek ukazuje do oficiálního archivu, druhý do non-US archivu a třetí do archivu bezpečnostních aktualizací Debianu.

Poslední dva řádky jsou zakomentovány (přidáním '#' na začátek) a apt-get je tedy bude ignorovat. Jsou to deb-src řádky, tj. ukazují na zdrojové balíčky Debianu. Odkomentujte je, pokud chcete stahovat zdrojové kódy programů pro testování nebo rekompilaci.

Soubor /etc/apt/sources.list může obsahovat několik různých typů řádků. APT ví, jak má pracovat s archivy typu http, ftp, file (lokální archiv, např. adresář s připojeným souborovým systémem ISO9660) a ssh.

Po provedení změn v souboru /etc/apt/sources.list nezapomeňte spustit apt-get update. APT tak z uvedených zdrojů získá seznamy balíčků.

2.2 Lokální použití APT

Můžete mít spoustu balíčků .deb, které byste rádi nainstalovali pomocí APT, protože automaticky vyřeší závislosti.

Vytvořte tedy adresář a do něj umístěte všechny soubory .deb, které chcete zapsat do seznamu (indexovat). Například:

```
# mkdir /root/debs
```

Volby nastavené v řídicím souboru balíčku můžete upravit přímo pro váš repozitář. Použijte k tomu soubor override. V tomto souboru můžete definovat několik voleb, které pak překryjí ty, které jsou přednastavené v balíčku. Vypadá to následovně:

```
package priority section
```

package je název balíčku, priorita je low (nízká), medium (střední) nebo high (vysoká) a sekce je ta sekce, do níž balíček patří. Název souboru není podstatný, můžete ho později změnit pomocí parametru programu dpkg-scanpackages. Pokud nechcete soubor override vytvářet, použijte při volání programu dpkg-scanpackages parametru /dev/null.

v adresáři /root udělejte ještě:

```
# dpkg-scanpackages debs soubor | gzip > debs/Packages.gz
```

Na řádku výše je soubor soubor override, příkaz vygeneruje soubor Packages.gz, který obsahuje různé informace o balíčcích. Tyto informace pak využívá APT. K použití balíčků přidejte ještě do souboru /etc/apt/sources.list:

```
deb file:/root debs/
```

Po tomto můžete používat příkazy APT jako obvykle. Můžete také vygenerovat repozitář se zdrojovými balíčky (sources repository). Postup je stejný, avšak pamatujte, že musíte mít v adresáři soubory .orig.tar.gz, .dsc a .diff.gz a musíte Packages.gz nahradit za Sources.gz. Program, který použijete je také jiný: dpkg-scansources. Příkazová řádka bude vypadat takto:


```
# dpkg-scansources debs | gzip > debs/Sources.gz
```

Všimněte si, že `dpkg-scansources` nepotřebuje soubor `override`. Do souboru `/etc/apt/sources.list` přidejte řádek:

```
deb-src file:/root debs/
```

2.3 Výběr nejlepších zrcadel do souboru `sources.list`: `netselect`, `netselect-apt`

Hlavně noví uživatelé často nevědí, jak rozhodnout, “které zrcadlo (mirror) Debianu zahrnout do `sources.list`?”. Existuje mnoho kritérií, podle kterých se rozhodnout. Experti pravděpodobně používají skript, který měří dobu pingu na různá zrcadla. Existuje však program, který nám s tímto pomůže: **netselect**.

Pro nainstalování `netselectu` napište jako obvykle:

```
# apt-get install netselect
```

Pokud spustíte `netselect` bez parametrů, vypíše nápovědu. Pokud ho spustíte s mezerami odděleným seznamem hostitelů (zrcadel), vrátí vám skóre a jednoho z hostitelů. Při výpočtu skóre se bere v úvahu odhadovaná doba pingu a počet skoků - hops (tj. počet hostitelů přes který dotaz prošel, než se dostal k cíli). Čím nižší skóre, tím je odhadovaná rychlost stahování (downloadu) vyšší. `Netselect` vrací hostitele s nejnižším skóre. Pokud chcete vidět bodování všech zadaných hostitelů, přidejte volbu `-vv`. Příklad:

```
# netselect ftp.debian.org http.us.debian.org ftp.at.debian.org download.unes
 365 ftp.debian.org.br
#
```

Ze seznamu zrcadel, který jsme programu ‘`netselect`’ předali jako parametry, byl `ftp.debian.org.br` se skórem 365 vybrán jako nejlepší zrcadlo. (Pozor!! Měření jsem provedl na svém počítači. Topologie sítě se v různých místech podstatně liší, proto tato hodnota není nezbytně správná i na jiných počítačích.

Nyní můžete zrcadlo, které `netselect` vybral jako nejrychlejší, uložit do souboru `/etc/apt/sources.list` (viz ‘Soubor `/etc/apt/sources.list`’ na straně 3) a podívat se na tipy v ‘Správa balíčků’ na straně 7.

Poznámka: seznam zrcadel můžete vždycky najít v souboru http://www.debian.org/mirror/mirrors_full.

Od verze 0.3.ds1 obsahuje balíček `netselect` skript **netselect-apt**, který provádí celý proces výběru zrcadel automaticky. Stačí pouze zadat strom distribuce jako parametr (implicitně `stable`) a v aktuálním adresáři se vytvoří soubor `sources.list` s nejlepšími main a non-US zrcadly. Následující příklad vytvoří `sources.list` `stable` distribuce:

```
# ls sources.list
ls: sources.list: File or directory not found
# netselect-apt stable
(...)
# ls -l sources.list
sources.list
#
```

Poznámka: soubor `sources.list` se vytvoří v aktuálním adresáři. Je ho tedy třeba přesunout do adresáře `/etc/apt`.

Potom se podívejte na tipy ‘Správa balíčků’ na následující straně.

2.4 Přidání CD-ROM do souboru `sources.list`

Pokud chcete, aby APT instaloval a aktualizoval váš systém z CD-ROM, запиšte to do vašeho `sources.list`. Můžete na to použít program `apt-cdrom` takto:

```
# apt-cdrom add
```

s CD Debianu v mechanice. CD-ROM se připojí a pokud v ní je debianí CD, vyhledají se informace o balíčcích, které jsou na něm uloženy. Pokud máte CD-ROM nastavenou nějak neobvykle, použijte následující volby:

```
-h          - nápověda
-d directory - bod připojení CD-ROM
-r          - přejmenování připojeného CD-ROM
-m          - nepřipojuj
-f          - rychlý mód, nekontroluj soubory s balíčky
-a          - Thorough scan mode
```

Například:

```
# apt-cdrom -d /home/kov/mycdrom add
```

CD-ROM můžete také určit bez toho, abyste ji přidávali do vašeho seznamu zdrojů:

```
# apt-cdrom ident
```

Program `apt-cdrom` pracuje jen v případě, když je vaše CD-ROM řádně nakonfigurovaná v `/etc/fstab`.

Kapitola 3

Správa balíčků

3.1 Aktualizace seznamu dostupných balíčků

Ke sledování instalovaných, nenainstalovaných balíčků a balíčků, které jsou k dispozici pro nainstalování, používá balíčkovací systém soukromou databázi. Program `apt-get` využívá tuto databázi, aby zjistil, jak má balíčky, které si uživatel přeje, nainstalovat a aby zjistil, které další balíčky jsou potřeba, aby zvolený balíček správně pracoval.

K aktualizaci této databáze použijte příkaz `apt-get update`. Tento příkaz vyhledá seznamy balíčků v archivech uvedených v souboru `/etc/apt/sources.list`; pro více informací o tomto souboru viz 'Soubor `/etc/apt/sources.list`' na straně 3.

Je dobré spouštět tento příkaz pravidelně. Vy a váš systém tak budete mít stále přehled o možných aktualizacích balíčků, zvláště pak o bezpečnostních aktualizacích.

3.2 Instalace balíčků

Konečně je tu postup, na který jste všichni čekali! Nyní máte připraven `sources.list` a aktualizovanou databázi dostupných balíčků. K nainstalování požadovaného balíčku stačí spustit `apt-get`. Např.:

```
# apt-get install xchat
```

APT najde ve svojí databázi nejnovější verzi tohoto balíčku a získá jej z odpovídajícího archivu, jenž je zapsán v `sources.list`. Pokud instalovaný balíček závisí na jiném, což je náš případ, `apt` zkontroluje závislosti a nainstaluje potřebné balíčky. Zde je příklad:

```
# apt-get install nautilus
Reading Package Lists... Done
Building Dependency Tree... Done
```

```
The following extra packages will be installed:
  bonobo libmedusa0 libnautilus0
The following NEW packages will be installed:
  bonobo libmedusa0 libnautilus0 nautilus
0 packages upgraded, 4 newly installed, 0 to remove and 1 not upgraded.
Need to get 8329kB of archives. After unpacking 17.2MB will be used.
Do you want to continue? [Y/n]
```

Balíček `nautilus` závisí na uvedených sdílených knihovnách, pročež je APT z archivu stáhne také. Pokud byste na příkazové řádce `apt-get` specifikovali názvy těchto knihoven, APT by se neptal, jestli chcete pokračovat; automaticky by předpokládal, že chcete nainstalovat všechny vybrané balíčky.

Tedy APT se ptá pouze tehdy, když potřebuje potvrdit, že může nainstalovat i balíčky, které jste na příkazovém řádku nespecifikovali.

Následující volby programu `apt-get` mohou být užitečné:

```
-h  Tato nápověda.
-d  Pouze stáhne - neinstaluje ani nerozbaluje archívy
-f  Zkusí pokračovat, i když selže kontrola integrity
-s  Pouze simuluje prováděné akce
-y  Na všechny otázky odpovídá Ano
-u  Zobrazí také seznam aktualizovaných balíčků
```

Na jednom řádku lze pro instalaci vybrat více balíčků. Soubory stažené ze sítě k pozdější instalaci se umístí ují do adresáře `/var/cache/apt/archives`

Na stejném řádku lze rovněž specifikovat balíčky, které chcete odstranit. Stačí jen přidat znak `'-'` hned za název balíčku. Např.:

```
# apt-get install nautilus gnome-panel-
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  bonobo libmedusa0 libnautilus0
The following packages will be REMOVED:
  gnome-applets gnome-panel gnome-panel-data gnome-session
The following NEW packages will be installed:
  bonobo libmedusa0 libnautilus0 nautilus
0 packages upgraded, 4 newly installed, 4 to remove and 1 not upgraded.
Need to get 8329kB of archives. After unpacking 2594kB will be used.
Do you want to continue? [Y/n]
```

Více informací o odstraňování balíčků najdete v sekci 'Odstraňování balíčků' na následující straně.

Pokud by se vám stalo, že jste balíček nějak poškodili, nebo pokud jen chcete balíček přeinstalovat novější verzí, která je dostupná, můžete použít volbu `--reinstall`. Např.:

```
# apt-get --reinstall install gdm
Reading Package Lists... Done
Building Dependency Tree... Done
0 packages upgraded, 0 newly installed, 1 reinstalled, 0 to remove and 1 not
Need to get 0B/182kB of archives. After unpacking 0B will be used.
Do you want to continue? [Y/n]
```

3.3 Odstraňování balíčků

Pokud už balíček nechcete používat, můžete jej pomocí APT odebrat ze systému. Stačí jen napsat: `apt-get remove package`. Např.:

```
# apt-get remove gnome-panel
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be REMOVED:
  gnome-applets gnome-panel gnome-panel-data gnome-session
0 packages upgraded, 0 newly installed, 4 to remove and 1 not upgraded.
Need to get 0B of archives. After unpacking 14.6MB will be freed.
Do you want to continue? [Y/n]
```

Jak je vidět, APT se také postará o odebrání balíčků, jež závisí na balíčku, který chcete odstranit. Neexistuje způsob jak odebrat balíček bez toho, aby byly odstraněny i všechny balíčky, které na něm závisí.

Spuštěním `apt-get` uvedeným způsobem odstraní balíčky, ale jejich konfigurační soubory, pokud nějaké mají, zůstanou nedotčeny. Pro úplné odstranění balíčku spusťte:

```
# apt-get --purge remove gnome-panel
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be REMOVED:
  gnome-applets* gnome-panel* gnome-panel-data* gnome-session*
0 packages upgraded, 0 newly installed, 4 to remove and 1 not upgraded.
Need to get 0B of archives. After unpacking 14.6MB will be freed.
Do you want to continue? [Y/n]
```

Všimněte si `'*'` za názvy balíčků. Značí, že spolu s balíčkem budou odstraněny i jeho konfigurační soubory.

Stejně jako při instalaci příkazem `install`, můžete s příkazem `remove` použít značky k obrácení významu pro konkrétní balíček. Pokud v případě odstraňování připojíte hned za název balíčku `'+'`, balíček se bude místo odstranění instalovat.

```
# apt-get --purge remove gnome-panel nautilus+
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  bonobo libmedusa0 libnautilus0 nautilus
The following packages will be REMOVED:
  gnome-applets* gnome-panel* gnome-panel-data* gnome-session*
The following NEW packages will be installed:
  bonobo libmedusa0 libnautilus0 nautilus
0 packages upgraded, 4 newly installed, 4 to remove and 1 not upgraded.
Need to get 8329kB of archives. After unpacking 2594kB will be used.
Do you want to continue? [Y/n]
```

Všimněte si, že `apt-get` vypíše balíčky, které budou instalovány navíc (tj. balíčky, které je potřeba nainstalovat, aby požadovaný balíček správně fungoval), balíčky které budou odstraněny a celkově všechny balíčky, které se budou instalovat (znovu včetně balíčků navíc).

3.4 Upgrade balíčků

Možnost upgradu balíčků je velkým úspěchem APT. Stačí napsat jediný příkaz: `apt-get upgrade`. Tento příkaz můžete použít nejen k upgradu balíčků v rámci stejné distribuce, nýbrž také k upgradu na novou distribuci, ačkoliv v druhém případě se dává přednost příkazu `apt-get dist-upgrade`; více informací najdete v sekci ‘Upgrade na nové vydání Debianu’ na následující straně

Je užitečné spouštět tento příkaz s volbou `-u`. S touto volbou ukáže APT kompletní seznam balíčků, které se budou upgradovat. Jinak byste upgradovali na slepo. APT stáhne poslední verze jednotlivých balíčků a nainstaluje je ve správném pořadí. Před upgradováním je důležité vždy spustit `apt-get update`. Viz sekce ‘Aktualizace seznamu dostupných balíčků’ na straně 7. Zde je příklad:

```
# apt-get -u upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages have been kept back
  cpp gcc lilo
The following packages will be upgraded
  adduser ae apt autoconf debhelper dpkg-dev esound esound-common ftp indent
  ipchains isapnptools libaudiofile-dev libaudiofile0 libesd0 libesd0-dev
  libgtk1.2 libgtk1.2-dev liblockfile1 libnewt0 liborbit-dev liborbit0
  libstdc++2.10-glibc2.2 libtiff3g libtiff3g-dev modconf orbit procs psmisc
29 packages upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
Need to get 5055B/5055kB of archives. After unpacking 1161kB will be used.
Do you want to continue? [Y/n]
```

Postup je velmi jednoduchý. Všimněte si, že na několika prvních řádcích `apt-get` oznamuje, že některé balíčky byly podrženy v aktuálním stavu - `kept back`. To znamená, že jsou k dispozici nové verze balíčků, které se ale z nějakého důvodu nebudou instalovat. Možné důvody jsou nesplněné závislosti (verze balíčku, na kterém náš závisí, není dostupná ke stažení) nebo nové závislosti (naš balíček se od poslední verze stal závislým na dalších balíčcích).

Pro první případ neexistuje "čisté" řešení. V druhém případě stačí pro daný balíček spustit `apt-get install`, což stáhne i balíčky, na kterých ten náš závisí. Ještě čistějším řešením je použít `dist-upgrade`. Viz sekce 'Upgrade na nové vydání Debianu' na této straně.

3.5 Upgrade na nové vydání Debianu

APT umožňuje upgrade celého systému Debian najednou, buď přes Internet nebo z nového CD (koupeného nebo staženého jako ISO image).

Stejný postup se rovněž používá, když se mezi instalovanými balíčky změnila závislosti. S `apt-get upgrade` by tyto balíčky zůstali nedotčeny (`kept back`).

Dejme tomu, že používáte revizi 0 stabilní verze Debianu a koupili jste si CD s revizí 3. K upgradu systému z nového CD můžete použít APT takto: zavolat `apt-cdrom` (viz sekce 'Přidání CD-ROM do souboru `sources.list`' na straně 6), abyste přidali CD do vašeho `/etc/apt/sources.list` a spustit `apt-get dist-upgrade`.

Je důležité poznamenat, že APT vždy hledá nejnovější verze balíčků. Pokud tedy váš `/etc/apt/sources.list` obsahoval archiv, ve kterém jsou novější verze balíčků než na vašem CD, APT stáhne balíčky odtamtud.

V příkladu, který jsme uvedli v sekci 'Upgrade balíčků' na předchozí straně, jsme viděli, že byly některé balíčky podrženy v aktuálním stavu - `kept back`. Tento problém nyní vyřešíme s `dist-upgrade`:

```
# apt-get -u dist-upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
Calculating Upgrade... Done
The following NEW packages will be installed:
  cpp-2.95 cron exim gcc-2.95 libident libopenldap-runtime libopenldap1
  libpcre2 logrotate mailx
The following packages have been kept back
  lilo
The following packages will be upgraded
  adduser ae apt autoconf cpp debhelper dpkg-dev esound esound-common ftp gcc
  indent ipchains isapnptools libaudiofile-dev libaudiofile0 libesd0
  libesd0-dev libgtk1.2 libgtk1.2-dev liblockfile1 libnewt0 liborbit-dev
  liborbit0 libstdc++2.10-glibc2.2 libtiff3g libtiff3g-dev modconf orbit
  procs psmisc
31 packages upgraded, 10 newly installed, 0 to remove and 1 not upgraded.
```

```
Need to get 0B/7098kB of archives. After unpacking 3118kB will be used.  
Do you want to continue? [Y/n]
```

Všimněte si, že nyní se budou balíčky upgradovat a také se budou instalovat nové (kvůli novým závislostem). Balíček lilo je však stále držen v aktuálním stavu - kept back. Problém bude asi vážnější, než jen nová závislost. Můžeme to zjistit spuštěním:

```
# apt-get -u install lilo  
Reading Package Lists... Done  
Building Dependency Tree... Done  
The following extra packages will be installed:  
  cron debconf exim libident libopenldap-runtime libopenldap1 libpcre2  
  logrotate mailx  
The following packages will be REMOVED:  
  debconf-tiny  
The following NEW packages will be installed:  
  cron debconf exim libident libopenldap-runtime libopenldap1 libpcre2  
  logrotate mailx  
The following packages will be upgraded:  
  lilo  
1 packages upgraded, 9 newly installed, 1 to remove and 31 not upgraded.  
Need to get 225kB/1179kB of archives. After unpacking 2659kB will be used.  
Do you want to continue? [Y/n]
```

Jak je vidět, lilo je v konfliktu s balíčkem debconf-tiny, což znamená, že se nemůže nainstalovat (nebo upgradovat) aniž by byl debconf-tiny odebrán.

Abychom se dozvěděli, co drží nebo odstraňuje balíček napíšeme:

```
# apt-get -o Debug::pkgProblemResolver=yes dist-upgrade  
Reading Package Lists... Done  
Building Dependency Tree... Done  
Calculating Upgrade... Starting  
Starting 2  
Investigating python1.5  
Package python1.5 has broken dep on python1.5-base  
  Considering python1.5-base 0 as a solution to python1.5 0  
  Holding Back python1.5 rather than change python1.5-base  
Investigating python1.5-dev  
Package python1.5-dev has broken dep on python1.5  
  Considering python1.5 0 as a solution to python1.5-dev 0  
  Holding Back python1.5-dev rather than change python1.5  
  Try to Re-Instate python1.5-dev  
Done  
Done
```



```
The following packages have been kept back
  gs python1.5-dev
0 packages upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
```

Tímto způsobem se dá jednoduše zjistit, že balíček python1.5-dev se nemůže nainstalovat kvůli nesplněné závislosti na python1.5.

3.6 Odstránění nepoužívaných souborů balíčků: apt-get clean and autoclean

Když instalujete balíček, APT získá potřebné soubory od hostitelů uvedených v `/etc/apt/sources.list`, uloží je v lokálním repozitáři (`/var/cache/apt/archives/`) a potom pokračuje s instalací, viz 'Instalace balíčků' na straně 7.

Časem se může lokální skladiště zvětšovat a zabírat na disku hodně místa. APT naštěstí poskytuje nástroje pro správu lokálního repozitáře. Medoty `apt-get: clean and autoclean`.

`apt-get clean` odstraní vše kromě souborů zámků z `/var/cache/apt/archives/` a `/var/cache/apt/archives/partial/`. Takže pokud budete chtít balíček přeinstalovat, APT ho bude muset znovu stáhnout.

`apt-get autoclean` odstraní pouze soubory balíčků, které už se nedají stáhnout.

Následující příklad ukazuje, jak `apt-get autoclean` pracuje:

```
# ls /var/cache/apt/archives/logrotate* /var/cache/apt/archives/gpm*
logrotate_3.5.9-7_i386.deb
logrotate_3.5.9-8_i386.deb
gpm_1.19.6-11_i386.deb
```

V `/var/cache/apt/archives` jsou dva soubory balíčku `logrotate` a jeden balíčku `gpm`.

```
# apt-show-versions -p logrotate
logrotate/stable uptodate 3.5.9-8
# apt-show-versions -p gpm
gpm/stable upgradeable from 1.19.6-11 to 1.19.6-12
```

`apt-show-versions` ukazuje, že aktuální verzi balíčku `logrotate` je `logrotate_3.5.9-8_i386.deb`, protože je soubor `logrotate_3.5.9-7_i386.deb` zbytečný. Soubor `gpm_1.19.6-11_i386.deb` je také zbytečný, neboť lze získat novější verzi tohoto balíčku.

```
# apt-get autoclean
Reading Package Lists... Done
Building Dependency Tree... Done
Del gpm 1.19.6-11 [145kB]
Del logrotate 3.5.9-7 [26.5kB]
```

Posléze `apt-get autoclean` odstraní pouze staré soubory. Více informací o `apt-show-versions` naleznete v 'Jak upgradovat balíčky z konkrétních verzí Debianu' na straně 16.

3.7 Použití APT spolu s dselect

`dselect` je program, který uživatelům pomáhá s výběrem debianích balíčků k instalaci. Je považován za poněkud složitý a poněkud nudný, ale s trochou cviku si na jeho konsolové `ncurses` rozhraní zvyknete.

`Dselect` umí využít schopnosti debianích balíčků "doporučovat" a "navrhovat" jiné balíčky k instalaci. Program spusťte jako root příkazem ``dselect``. Jako přístupovou metodu zvolte `'apt'`. Není to nezbytně nutné, pokud však nepoužíváte CD-ROM a chcete balíčky stahovat z Internetu, je `'apt'` nejlepší možnost.

Pokud se chcete s `dselectem` naučit dobře pracovat, přečtěte si dokumentaci uvedenou na stránkách Debianu <http://www.debian.org/doc/ddp>.

Potom, co jste si pomocí `dselectu` vybrali, napište:

```
# apt-get -u dselect-upgrade
```

jako v níže uvedeném příkladě:

```
# apt-get -u dselect-upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be REMOVED:
  lbxproxy
The following NEW packages will be installed:
  bonobo console-tools-libs cpp-3.0 enscript expat fingerd gcc-3.0
  gcc-3.0-base icepref klogd libdigest-md5-perl libfnlib0 libft-perl
  libgc5-dev libgcc300 libhtml-clean-perl libltdl0-dev libsasl-modules
  libstdc++3.0 metamail nethack proftpd-doc psfontmgr python-newt talk tidy
  util-linux-locales vacation xbill xplanet-images
The following packages will be upgraded:
  debian-policy
1 packages upgraded, 30 newly installed, 1 to remove and 0 not upgraded.
Need to get 7140kB of archives. After unpacking 16.3MB will be used.
Do you want to continue? [Y/n]
```

Porovnejme výpis s tím, který se zobrazí spustěním `apt-get dist-upgrade` na stejném systému:

```
# apt-get -u dist-upgrade
Reading Package Lists... Done
```

```
Building Dependency Tree... Done
Calculating Upgrade... Done
The following packages will be upgraded
  debian-policy
1 packages upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 421kB of archives. After unpacking 25.6kB will be freed.
Do you want to continue? [Y/n]
```

Všimněte si, že se spousta výše uvedených balíčků instaluje, protože je jiné balíčky “doporučily” nebo “navrhly”. Ostatní se instalují nebo odstraňují (viz např. lbxproxy) podle výběru, který jsme procházením seznamu balíčků pomocí dselectu vytvořili. Dselect je při použití v kombinaci s APT velice silným nástrojem.

3.8 Jak udržovat smíšený systém

Uživatelé občas používají jednu verzi Debianu jako hlavní distribuci systému a jeden či více balíčků z jiné větve.

Nastavení vaší hlavní verze Debianu proveďte v souboru `/etc/apt/apt.conf` (pokud tento neexistuje, vytvořte jej), kam napíšete následující řádku:

```
APT::Default-Release "version";
```

version je verze Debianu, kterou chcete používat jako hlavní distribuci. Vybrat si můžete z možností `stable`, `testing` a `unstable`. K instalování balíčků z jiné verze použijte APT takto:

```
# apt-get -t distribution install package
```

Aby to ale fungovalo, musíte mít pro distribuci, ze které balíček chcete, záznam v souboru `/etc/apt/sources.list`. Balíček se pak musí v daném zdroji vyskytovat.

Můžete také požadovat určitou konkrétní verzi balíčku, pomocí tohoto zápisu:

```
# apt-get install package=version
```

Např. následující řádek nainstaluje verzi `2.2.4-1` balíčku `nautilus`:

```
# apt-get install nautilus=2.2.4-1
```

DŮLEŽITÉ: ‘unstable’ verze Debianu je verze, do které se nejnovější verze balíčků vkládají nejdřív. Distribuce tak zažívá všechny změny, jimiž balíčky procházejí. Změny jsou jak malé tak i velké, které ovlivňují spousty balíčků nebo dokonce celý systém. Z toho důvodu by tuto verzi distribuce *neměli* používat nezkušení uživatelé nebo ti, kteří chtějí mít zajištěnu stabilitu.

‘testing’ distribuce není nutně lepší než ‘unstable’, protože se v ní bezpečnostní aktualizace neprovádějí rychle. Pro servery a jiné produkční systémy by se měla vždy používat verze stabilní - ‘stable’.

3.9 Jak upgradovat balíčky z konkrétních verzí Debianu

`apt-show-versions` poskytuje uživatelům smíšených distribucí bezpečnou cestu, jak upgradovat systém bez toho, aby dostali z méně stabilní distribuce více, než chtěli. Po nainstalování balíčku `apt-show-versions` lze například upgradovat pouze nestabilní balíčky spuštěním:

```
# apt-get install `apt-show-versions -u -b | grep unstable | cut -d ' ' -f 1`
```

3.10 Jak udržet nainstalované určité verze balíčků

Možná, že jste potřebovali změnit něco v balíčku a nemáte čas, nebo nechcete, přenášet tyto změny do nové verze programu. Nebo jste například upgradovali svoji distribuci Debianu na verzi 3.0, ale chcete i nadále používat určité balíčky ve verzi z Debianu 2.2. Můžete označit (přiřadit “pin” - přišpendlit) verzi, kterou jste nainstalovali, takže se nebude upgradovat.

Použití tohoto prostředku je jednoduché, stačí jen editovat soubor `/etc/apt/preferences`.

Formát souboru je prostý:

```
Package: <package>
Pin: <pin definition>
Pin-Priority: <pin's priority>
```

Každá položka musí být od ostatních oddělena odřádkováním. Například pro podržení balíčku `sylpheed`, který jsem upravil aby používal “reply-to-list”, ve verzi 0.4.99, jsem přidal záznam:

```
Package: sylpheed
Pin: version 0.4.99*
```

Všimněte si, že jsem použil `*` (hvězdička). Vznikla tak maska, která říká, že přiřazený “pin” platí pro všechny verze začínající na 0.4.99. Učinil jsem tak proto, že Debian v označení verzí svých balíčků užívá tzv. číslo revize a já se nechtěl vyhnout instalaci možných revizí. Takže například verze 0.4.99-1 a 0.4.99-10 se nainstalují hned, jak budou k dispozici. Uvědomte si, že pokud jste balíček upravili, není tento způsob vhodný.

‘Pin-Priority’ pomáhá určit, jestli bude balíček vyhovující řádkům “Package:” a “Pin:” instalován. Čím vyšší priorita, tím vyšší pravděpodobnost, že se vyhovující balíček nainstaluje. Zevrubný rozbor priorit najdete v `apt_preferences(7)`. Uvedu pár příkladů, které ilustrují základní myšlenku. V následujícím popíšu, jaký efekt by měla změna hodnoty priority v předcházejícím příkladě se `sylpheed`.

- 1001** APT verzi 0.4.99 balíčku `sylpheed` nikdy nenahradí. Pokud je dostupná, APT nainstaluje verzi 0.4.99 dokonce i když tím nahradí již nainstalovaný balíček s vyšší verzí. Pouze balíčky s prioritou vyšší než 1000, můžou downgradeovat tento balíček.
- 1000** Stejný efekt jako priorita 1001, s výjimkou toho, že APT zamítne downgrade nainstalované verze na verzi 0.4.99
- 990** Verze 0.4.99 bude nahrazena pouze vyšší verzí z vydání označeného pomocí proměnné `"APT::Default-Release"` jako preferované. (viz 'Jak udržovat smíšený systém' na straně 15, nahoře).
- 500** Upřednostní jakoukoliv vyšší verzi `sylpheed` než je 0.4.99, dostupnou v jakémkoliv vydání. Verze 0.4.99 však bude mít stále přednost před nižšími verzemi.
- 100** Jakákoliv vyšší verze balíčku `sylpheed` z jakéhokoliv vydání, jakož i jakákoliv nainstalovaná vyšší verze budou mít přednost před verzí 0.4.99. Tj. verze 0.4.99 se bude instalovat, pouze pokud ještě žádná verze nebyla instalována. Tuto prioritu mají nastavenou nainstalované balíčky.
- 1** Dovoleny jsou i záporné priority. Tyto zcela zamezují instalaci verze 0.4.99.

Pin může být specifikován u verze balíčku, `release` balíčku, nebo původce.

Při použití pinu na verzi balíčku, viz výše, lze zadat přesné číslo verze nebo použít masku k označení několika verzí najednou.

Volba `release` závisí na souboru `Release` v repozitáři APT nebo na CD. Pokud repozitář tento soubor neobsahuje, nedá se tato volba použít. Můžete si prohlédnout `Release` soubory, které máte v adresáři `/var/lib/apt/lists/`. Parametry `releasu` jsou: `a` (archiv), `c` (komponenta), `v` (verze), `o` (zdroj) a `l` (label - jmenovka).

Příklad:

```
Package: *
Pin: release v=2.2*, a=stable, c=main, o=Debian, l=Debian
Pin-Priority: 1001
```

V tomto příkladu jsme zvolili verzi Debianu 2.2* (což může být 2.2r2, 2.2r3 – spolu s tečkovanými vydáními, které typicky obsahují bezpečnostní záplaty a další velmi důležité aktualizace), repozitář `stable`, sekci `main` (oproti `contrib` nebo `non-free`) a původce i jmenovku `Debian`. Zdroj (`o=`) udává tvůrce `Release` souboru, jmenovka (`l=`) udává název distribuce: `Debian` pro vlastní Debian a `Progeny` pro `Progeny` například. Zde je ukázka `Release` souboru:

```
$ cat /var/lib/apt/lists/ftp.debian.org.br_debian_dists_potato_main_binary-i386
Archive: stable
Version: 2.2r3
Component: main
Origin: Debian
Label: Debian
Architecture: i386
```


Kapitola 4

Velmi užiteční pomocníci

4.1 Jak instalovat lokálně kompilované balíčky: `equivs`

Někdy chce člověk používat určitou verzi programu, která je však dostupná pouze v podobě zdrojového kódu. Debianí balíček není k dispozici. S balíčkovacím systémem pak můžou být problémy. Představte si, že chcete kompilovat novou verzi Vašeho emailového serveru. Všechno je vpořádku až na to, že spousta balíčků v Debianu závisí na MTA (Mail Transport Agent). Protože jste nainstaloval něco, co jste si zkompiloval sám, balíčkovací systém o tom neví.

To je místo, kde `equivs` přichází na scénu. Abyste ho mohl použít, nainstalujte stejnojmenný balíček. `Equivs` dělá to, že vytvoří prázdný balíček, který slouží ke splnění závislostí. Balíčkovací systém se pak domnívá, že jsou závislosti splněny.

Než začneme, je dobré připomenout, že jsou bezpečnější cesty jak nainstalovat program, který už má v Debianu balíček zkompilovaný s jinými nastaveními, a že by člověk neměl používat `equivs` k nahrazování závislostí, pokud neví, co přesně dělá. Více informací je v sekci 'Práce se zdrojovými balíčky' na straně [29](#).

Pokračujme s MTA příkladem. Právě jste nainstaloval svůj nově zkompilovaný `postfix` a pokračujete s instalací `mutt`. Náhle zjistíte, že balíček `mutt` chce nainstalovat jiný MTA. Ale vy už máte svůj.

Přesuňte se do nějakého adresáře (`/tmp` například) a spusťte:

```
# equivs-control název
```

název nahrad'te jménem řídicího souboru, který chcete vytvořit. Vytvořený soubor bude vypadat následovně:

```
Section: misc
Priority: optional
Standards-Version: 3.0.1
```

```
Package: <vlože jméno balíčku; implicitně je equivs-dummy>
Version: <vlože číslo verze; implicitně je 1.0>
Maintainer: <vaše jméno a emailová adresa; implicitně je uživatelské jméno>
Pre-Depends: <balíčky>
Depends: <balíčky>
Recommends: <balíčky>
Suggests: <balíček>
Provides: <(virtuální)balíček>
Architecture: all
Copyright: <soubor s copyrightem; implicitně je GPL2>
Changelog: <soubor changelog; implicitně je všeobecný changelog>
Readme: <README.Debian soubor; implicitní je všeobecný>
Extra-Files: <dodatečné soubory pro doc adresář, oddělené čárkou>
Description: <krátký popis; implicitně jsou to nějaká moudrá slova>
    dlouhý popis a informace
.
druhý odstavec
```

Toto potřebujeme jenom upravit, aby to dělalo, co chceme. Podívejte se na formát polí a jejich popis, není tu nutné nic vysvětlovat, udělejme, co je potřeba:

```
Section: misc
Priority: optional
Standards-Version: 3.0.1

Package: mta-local
Provides: mail-transport-agent
```

Ano, to je vše. mutt závisí na mail-transport-agent, což je virtuální balíček poskytovaný všemi MTA. Mohl jsem ten balíček jednoduše nazvat mail-transport-agent, ale upřednostnil jsem schéma s virtuálními balíčky použitím Provides.

Nyní pouze potřebujeme balíček vytvořit:

```
# equivs-build name
dh_testdir
touch build-stamp
dh_testdir
dh_testroot
dh_clean -k
# Add here commands to install the package into debian/tmp.
touch install-stamp
dh_testdir
dh_testroot
```



```
dh_installdocs
dh_installchangelogs
dh_compress
dh_fixperms
dh_installdeb
dh_gencontrol
dh_md5sums
dh_builddeb
dpkg-deb: building package `name' in `../name_1.0_all.deb'.
```

The package has been created.

Attention, the package has been created in the current directory,

A nainstalovat výsledný `.deb`.

Jak je vidět, je spousta možností, jak využít `equivs`. Například byste si mohl vytvořit balíček `my-favorites`, který by závisel na programech, které obvykle instalujete. Jen popusťte uzdu fantazie, ale buďte opatrní.

Nutno poznamenat, že příklady řídicích souborů jsou v `/usr/share/doc/equivs/examples`. Vyzkoušejte je!

4.2 Odstranění nepoužívaných lokalizačních souborů: `localepurge`

Spousta uživatelů Debianu používá jen jednu locale. Brazílský uživatel Debianu například obvykle celou dobu používá `pt_BR` locale a o `es` locale se nestará.

`localepurge` je pro tyto uživatele velmi užitečný nástroj. Můžete uvolnit spousta místa tím, že budete mít pouze locales, které opravdu používáte. Udělejte jen `apt-get install localepurge`.

Konfigurace je velmi jednoduchá, `debconf` provede uživatele konfigurací krok za krokem. Přece jenom buďte opatrní při odpovědi na první otázku. Špatná odpověď může způsobit odstranění všech souborů locale, dokonce i těch které používáte. Jediná možnost, jak tyto soubory dostat zpátky, je přeinstalování všech balíčků, které je poskytují.

4.3 Jak poznat, které balíčky lze upgradovat

`apt-show-versions` je program, který ukáže, které balíčky v systému lze aktualizovat (updatovat) a některé další užitečné informace. Volba `-u` zobrazí seznam balíčků, které je možno upgradovat:

```
$ apt-show-versions -u
libeel0/unstable upgradeable from 1.0.2-5 to 1.0.2-7
libeel-data/unstable upgradeable from 1.0.2-5 to 1.0.2-7
```


Kapitola 5

Získávání informací o balíčcích

Pro systém APT existují některé front-endové programy, které značně zjednodušují zjišťování seznamu balíčků dostupných pro instalaci nebo už nainstalovaných, stejně jako zjištění sekce, do které balíček patří, jakou má prioritu, jaký je jeho popis apod.

Ale... naším cílem je naučit se používat čisté APT. Tak tedy, jak můžete zjistit jméno balíčku, který chcete instalovat?

Pro řešení této úlohy máme spoustu prostředků. Začneme s `apt-cache`. Tento program využívá systém APT pro údržbu svojí databáze. Letmo se podíváme na některé možnosti jeho praktického využití.

5.1 Zjišťování názvů balíčků

Předpokládejme například, že si chcete připomenout staré dobré časy Atari 2600. Chcete použít APT k nainstalování emulátoru Atari a potom ke stažení nějakých her. Můžete udělat:

```
# apt-cache search atari
atari-fdisk-cross - Partition editor for Atari (running on non-Atari)
circuslinux - The clowns are trying to pop balloons to score points!
madbomber - A Kaboom! clone
tcs - Character set translator.
atari800 - Atari emulator for svgalib/X/curses
stella - Atari 2600 Emulator for X windows
xmess-x - X binaries for Multi-Emulator Super System
```

Našli jsme několik balíčků, jež se vztahují k tomu, co jsme hledali, spolu s krátkým popisem. K získání více informací o konkrétním balíčku, mohu použít:

```
# apt-cache show stella
Package: stella
```

```
Priority: extra
Section: non-free/otherosfs
Installed-Size: 830
Maintainer: Tom Lear <tom@trap.mtview.ca.us>
Architecture: i386
Version: 1.1-2
Depends: libc6 (>= 2.1), libstdc++2.10, xlib6g (>= 3.3.5-1)
Filename: dists/potato/non-free/binary-i386/otherosfs/stella_1.1-2.deb
Size: 483430
MD5sum: 11b3e86a41a60falc4b334dd96c1d4b5
Description: Atari 2600 Emulator for X windows
Stella is a portable emulator of the old Atari 2600 video-game console
written in C++. You can play most Atari 2600 games with it. The latest
news, code and binaries for Stella can be found at:
http://www4.ncsu.edu/~bwmott/2600
```

Výstupem je spousta detailních informací o balíčku, který chcete (nebo nechcete) instalovat, spolu s plným popisem balíčku. Pokud je už balíček v systému nainstalován a existuje novější verze, uvidíte informace o obou verzích. Například:

```
# apt-cache show lilo
Package: lilo
Priority: important
Section: base
Installed-Size: 271
Maintainer: Russell Coker <russell@coker.com.au>
Architecture: i386
Version: 1:21.7-3
Depends: libc6 (>= 2.2.1-2), debconf (>=0.2.26), logrotate
Suggests: lilo-doc
Conflicts: manpages (<<1.29-3)
Filename: pool/main/l/lilo/lilo_21.7-3_i386.deb
Size: 143052
MD5sum: 63fe29b5317fe34ed8ec3ae955f8270e
Description: LInux LOader - The Classic OS loader can load Linux and others
This Package contains lilo (the installer) and boot-record-images to
install Linux, OS/2, DOS and generic Boot Sectors of other OSes.
.
You can use Lilo to manage your Master Boot Record (with a simple text screen)
or call Lilo from other Boot-Loaders to jump-start the Linux kernel.

Package: lilo
Status: install ok installed
Priority: important
Section: base
```

```

Installed-Size: 190
Maintainer: Vincent Renardias <vincent@debian.org>
Version: 1:21.4.3-2
Depends: libc6 (>= 2.1.2)
Recommends: mbr
Suggests: lilo-doc
Description: LInux LOader - The Classic OS loader can load Linux and others
 This Package contains lilo (the installer) and boot-record-images to
 install Linux, OS/2, DOS and generic Boot Sectors of other OSes.
.
You can use Lilo to manage your Master Boot Record (with a simple text scree
 or call Lilo from other Boot-Loaders to jump-start the Linux kernel.

```

Všimněte si, že první ve výpisu je dostupný balíček a druhý je balíček, který je již instalován. Pro více obecných informací o balíčku můžete použít:

```

# apt-cache showpkg penguin-command
Package: penguin-command
Versions:
1.4.5-1 (/var/lib/apt/lists/download.sourceforge.net_debian_dists_unstable_ma

Reverse Depends:
Dependencies:
1.4.5-1 - libc6 (2 2.2.1-2) libpng2 (0 (null)) libsdl-mixer1.1 (2 1.1.0) libs
Provides:
1.4.5-1 -
Reverse Provides:

```

A ke zjištění na kterých balíčcích závisí:

```

# apt-cache depends penguin-command
penguin-command
  Depends: libc6
  Depends: libpng2
  Depends: libsdl-mixer1.1
  Depends: libsdl1.1
  Depends: zlib1g

```

Celkem tedy máme široký rozsah zbraní, které můžeme použít ke zjištění názvu balíčku, který chceme.

5.2 Použití dpkg k nalezení názvů balíčků

Jedna z možností, jak najít název balíčku, je znát název některého důležitého souboru nacházejícího se uvnitř balíčku. Například abychom našli název balíčku, který obsahuje konkrétní

“.h” soubor, který potřebujeme pro kompilaci, můžeme spustit:

```
# dpkg -S stdio.h
libc6-dev: /usr/include/stdio.h
libc6-dev: /usr/include/bits/stdio.h
perl: /usr/lib/perl/5.6.0/CORE/nostdio.h
```

nebo:

```
# dpkg -S /usr/include/stdio.h
libc6-dev: /usr/include/stdio.h
```

Pro zjištění názvů balíčků nainstalovaných ve vašem systému, což je užitečné například pokud plánujete pročistit váš disk, můžete spustit:

```
# dpkg -l | grep mozilla
ii  mozilla-browser 0.9.6-7          Mozilla Web Browser
```

Problém tohoto příkazu je, že může “uříznout” konec názvu balíčku. V příkladu výše je plný název balíčku mozilla-browser. K opravení tohoto můžete použít proměnnou prostředí COLUMNS takto:

```
[kov]@[couve] $ COLUMNS=132 dpkg -l | grep mozilla
ii  mozilla-browser          0.9.6-7          Mozilla Web Brows
```

nebo popis nebo jeho část takto:

```
# apt-cache search "Mozilla Web Browser"
mozilla-browser - Mozilla Web Browser
```

5.3 Jak instalovat balíčky “na vyžádání”

Kompilujete program a znenadání bum! Je tu chyba, protože je potřeba .h soubor, který nemáte. Program auto-apt vás může takovýchto situací ušetřit. Zastaví kompilační proces a zeptá se vás na instalaci potřebných balíčků. Jakmile je balíček nainstalován, nechá proces pokračovat.

V podstatě musíte udělat následující:

```
# auto-apt run příkaz
```

Kde 'příkaz' je vykonávaný příkaz, který může potřebovat nějaké nedostupné soubory. Například:

```
# auto-apt run ./configure
```

auto-apt se pak zeptá, jestli chcete nainstalovat potřebné balíčky a automaticky zavolá apt-get. Pokud používáte X, program místo implicitního textového rozhraní nabídne grafické.

Auto-apt používá databáze, které, aby byl efektivní, je potřeba udržovat aktuální. Toho lze dosáhnout voláním příkazů `auto-apt update`, `auto-apt updatedb` a `auto-apt update-local`.

5.4 Jak zjistit, ke kterému balíčku soubor patří

Když chcete instalovat balíček a nemůžete zjistit jeho název pomocí `apt-cache` a přitom znáte název souboru vlastního programu nebo název některého jiného souboru, který k balíčku patří, pak můžete k nalezení názvu balíčku použít `apt-file`. Pracuje se s ním takhle:

```
$ apt-file search filename
```

Program funguje podobně jako `dpkg -S`, ale zobrazí i odinstalované balíčky, které hledaný soubor obsahují. Lze ho také použít ke zjištění, které balíčky obsahují potřebné hlavičkové soubory, které chybí při kompilaci programu. Avšak `auto-apt` je mnohem lepší způsob, jak řešit takovéto problémy, viz 'Jak instalovat balíčky "na vyžádání"' na předchozí straně.

Obsah balíčku si také můžete nechat zobrazit spuštěním:

```
$ apt-file list packagename
```

`apt-file` používá stejně jako `auto-apt` databázi, ve které je uloženo, které soubory jednotlivé balíčky obsahují. Databázi je třeba udržovat aktuální. Aktualizace se provádí spuštěním:

```
# apt-file update
```

Implicitně používá `apt-file` stejnou databázi jako `auto-apt`, viz 'Jak instalovat balíčky "na vyžádání"' na předchozí straně.

5.5 Jak zjistit změny v balíčcích

Každý balíček instaluje do svého adresáře s dokumentací (`/usr/share/doc/packagename`) soubor s názvem `changelog.Debian.gz`, který obsahuje seznam změn provedených v

balíčku od jeho poslední verze. Tyto soubory si můžete prohlédnout např. pomocí `zless`. Po kompletním upgradu systému však už není hledání všech nových souborů se seznamy změn tak snadné.

Je tu způsob, jak tuto úlohu zautomatizovat - nástrojem nazvaným `apt-listchanges`. Nejprve nainstalujte balíček `apt-listchanges`. Během instalace balíčku ho `Debconf` nakonfiguruje. Některé z otázek neuvídíte, záleží na prioritě, kterou jste pro `Debconf` nastavil. Na otázky odpovězte podle uvážení.

První otázka se ptá, jakým způsobem vás má `apt-listchanges` o změnách informovat. Můžete si je nechat posílat na mail, což je výhodné při automatických upgradech, nebo si je můžete nechat ukázat pomocí filtru, jakým je např. `less`, takže si můžete změny prohlédnout ještě před tím, než necháte upgrade pokračovat. Pokud nechcete, aby se `apt-listchanges` při upgradech spouštěl automaticky, odpovězte `none`.

Po nainstalování `apt-listchanges`, jakmile `apt` stáhne balíčky (resp. získá z CD nebo připojeného disku), před tím, než se budou instalovat, vypíše `apt-listchanges` seznam změn v těchto balíčcích.

Kapitola 6

Práce se zdrojovými balíčky

6.1 Stažení zdrojových balíčků

Ve světě svobodného software je běžné studovat zdrojový kód nebo dokonce opravovat chyby v tomto kódu. K tomuto budete potřebovat zdrojový kód programu stáhnout. APT systém poskytuje jednoduchý způsob, jak získat zdrojové kódy k mnoha programům obsaženým v distribuci, včetně všech souborů potřebných k vytvoření balíčku `.deb`.

Jiný obvyklý způsob užití zdrojových balíčků Debianu je upravení novější verze programu např. z `unstable` distribuce pro použití v `stable` distribuci. Kompilováním balíčku oproti `stable` distribuci, vygenerujete `.debs` se závislostmi upravenými tak, že pasují k ostatním balíčkůům této distribuce.

Abyste toho dosáhli, měla by položka `deb-src` ve vašem `/etc/apt/sources.list` ukazovat na `unstable` distribuci. Musí být rovněž povolena (odkomentována). Viz sekci 'Soubor `/etc/apt/sources.list`' na straně 3.

Ke stažení zdrojového balíčku použijte příkaz:

```
$ apt-get source packagename
```

Tím se stáhnou tři soubory: `.orig.tar.gz`, `.dsc` a `.diff.gz`. Pokud byl balíček vytvořen přímo pro Debian, nebudou dva posledně jmenované soubory staženy a první soubor nebude mít v názvu `"orig"`.

Soubor `.dsc` používá `dpkg-source` pro rozbalení zdrojového balíčku do adresáře `packagename-version`. Uvnitř každého staženého zdrojového balíčku je adresář `debian/`, který obsahuje soubory nutné k vytvoření `.deb` balíčku.

Pro automatický vytvoření balíčku, který jste si stáhli, přidejte `-b` na příkazový řádek takto:

```
$ apt-get -b source packagename
```

Pokud jste se rozhodli nevytvářet `.deb` balíček hned při stahování, můžete ho vytvořit později spuštěním:

```
$ dpkg-buildpackage -rfakeroot -uc -b
```

z adresáře, který byl pro balíček vytvořen po stažení. Pro nainstalování balíčku vytvořeného výše uvedenými příkazy musíte správce balíčků použít přímo. Takto:

```
# dpkg -i file.deb
```

Je rozdíl mezi `apt-get` metodou `source` a ostatními `apt-get` metodami. Metodu `source` může použít i běžný uživatel bez speciálních rootovských práv. Soubory se stáhnou do adresáře, ze kterého byl příkaz `apt-get source package` volán.

6.2 Balíčky nutné pro kompilaci zdrojového balíčku

Pro kompilaci zdrojového balíčku jsou obvykle nutné určité hlavičkové soubory a sdílené knihovny. Každý zdrojový balíček má ve svém řídicím souboru (control file) uvedeno políčko 'Build-Depends:', která udává, které dodatečné balíčky jsou pro kompilaci tohoto zdrojového balíčku potřeba.

APT nabízí jednoduchý způsob stažení těchto balíčků. Spusťte `apt-get build-dep balíček`, kde 'balíček' je název balíčku, který budete kompilovat. Například:

```
# apt-get build-dep gmc
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  comerr-dev e2fslibs-dev gdk-implib-dev implib-progs libgnome-dev libgnorba-de
  libgpmgl-dev
0 packages upgraded, 7 newly installed, 0 to remove and 1 not upgraded.
Need to get 1069kB of archives. After unpacking 3514kB will be used.
Do you want to continue? [Y/n]
```

Balíčky, které se budou instalovat, jsou nutné, aby se `gmc` řádně zkompiloval. Nutno poznamenat, že příkaz nehledá zdrojový balíček programu, který chcete kompilovat. Musíte jej tedy stáhnout zvlášť spuštěním `apt-get source`.

Pokud byste se chtěl pouze podívat, které balíčky jsou pro kompilaci daného balíčku nutné, je tu varianta příkazu `apt-cache show` (viz 'Získávání informací o balíčcích' na straně 23), která mezi jinými informacemi zobrazí řádek `Build-Depends`, na kterém jsou tyto balíčky uvedeny.

```
# apt-cache showsrc package
```

Kapitola 7

Jak se vypořádat s chybami

7.1 Běžné chyby

Chyby budou vždycky vznikat, spousta z nich nepozorností uživatele. Následuje seznam některých velmi častých chyb s popisem, jak se s nimi vypořádat.

Pokud se snažíte spustit `apt-get install package` a obdržíte chybu, která vypadá podobně jako tahle ...

```
Reading Package Lists... Done
Building Dependency Tree... Done
W: Couldn't stat source package list 'http://people.debian.org unstable/ Pack
W: You may want to run apt-get update to correct these missing files
E: Couldn't find package penguineyes
```

zapomněl jste po editaci souboru `/etc/apt/sources.list` spustit `apt-get update`.

Pokud chyba vypadá jako tahle

```
E: Could not open lock file /var/lib/dpkg/lock - open (13 Permission denied)
E: Unable to lock the administration directory (/var/lib/dpkg/), are you root
```

při pokusu o spuštění kterékoliv metody `apt-get` vyjma `source`, nemáte rootovská práva, tj. jste přihlášen jako běžný uživatel.

Podobná chyba nastane, pokud spouštíte dvě verze programu `apt-get` najednou, nebo pokud se dokonce snažíte spustit `apt-get` zatímco je `dpkg` proces aktivní. Jedinou metodou, kterou lze použít zároveň s jinými, je metoda `source`.

Pokud se instalace uprostřed zastaví a vy zjistíte, že už není možno instalovat nebo odstraňovat balíčky, zkuste použít tyto příkazy:

```
# apt-get -f install
# dpkg --configure -a
```

A potom to zkuste znovu. Může být nezbytné spustit druhý výše jmenovaný příkaz více než jednou. Je to důležitá lekce pro dobrodruhy, kteří používají 'unstable'.

Pokud za běhu `apt-get update` obdržíte chybou hlášku "E: Dynamic MMap ran out of room", přidejte následující řádek do `/etc/apt/apt.conf`:

```
APT::Cache-Limit 10000000;
```

7.2 Kde najdu pomoc?

Pokud máte pocit, že ničemu nerozumíte, podívejte se do obsáhlé dokumentace, která je pro balíčkovací systém Debianu k dispozici. Velmi pomoci vám může `--help` a manuálové stránky, stejně jako dokumentace obsažená v adresářích `/usr/share/doc` nebo `/usr/share/doc/apt`.

Pokud vám ani tato dokumentace nepomůže, zkuste odpověď najít v mailových konferencích Debianu. Více informací o konkrétních uživatelských konferencích můžete najít na stránkách Debianu: <http://www.debian.org>.

Uvědomte si, že tyto konference a zdroje by měli využívat pouze uživatelé Debianu; uživatelé ostatních systémů najdou lepší podporu v komunitních zdrojích své vlastní distribuce.

Kapitola 8

Které distribuce podporují APT?

Zde je seznam několika distribucí, které používají APT:

Debian GNU/Linux (<http://www.debian.org>) - právě pro tuto distribuci byl APT vyvinut

Conectiva (<http://www.conectiva.com.br>) - toto byla první distribuce, která portovala APT pro používání s 'rpm'

Libranet (<http://www.libranet.com>)

Mandrake (<http://www.mandrake.com>)

PLD (<http://www.pld.org.pl>)

Vine (<http://www.vinelinux.org>)

APT4RPM (<http://apt4rpm.sf.net>)

Alt Linux (<http://www.altlinux.ru/>)

Red Hat (<http://www.redhat.com/>)

Sun Solaris (<http://www.sun.com/>)

SuSE (<http://www.suse.de/>)

Yellow Dog Linux (<http://www.yellowdoglinux.com/>)

Kapitola 9

Credits

Velký dík patří mým přátelům z projektu Debian-BR a Debianu samotného, kteří mi jsou stálou oporou a dodávají mi sílu pokračovat v dobročinné práci, stejně jako mi pomáhají v mém cíli zahránit svět. :)

Chtěl bych rovněž poděkovat CIPSGA za obrovskou pomoc našemu projektu a všem svobodným projektům, které vycházejí ze skvělých myšlenek.

Zvláštní poděkování patří:

Yooseong Yang <yooseong@debian.org>

Michael Bramer <grisu@debian.org>

Bryan Stillwell <bryan@bokeoa.com>

Pawel Tecza <pawel.tecza@poczta.fm>

Hugo Mora <h.mora@melix.com.mx>

Luca Monducci <luca.mo@tiscali.it>

Tomohiro KUBOTA <kubota@debian.org>

Pablo Lorenzoni <spectra@debian.org>

Steve Langasek <vorlon@netexpress.net>

Arnaldo Carvalho de Melo <acme@conectiva.com.br>

Erik Rossen <rossen@freesurf.ch>

Ross Boylan <RossBoylan@stanfordalumni.org>

Matt Kraai <kraai@debian.org>

Aaron M. Ucko <ucko@debian.org>

Jon Åslund <d98-jas@nada.kth.se>

Kapitola 10

Nové verze tohoto návodu

Tento návod byl vytvořen v rámci projektu Debian-BR (<http://www.debian-br.org>), jehož cílem je pomoci při každodenním používání Debianu.

Nové verze tohoto dokumentu budou dostupné na stránce Debian Documentation Project na adrese <http://www.debian.org/doc/ddp>.

Komentáře a kritiku posílejte přímo mně na <kov@debian.org>.