

Použití RPM a DEB balíčků v OS Linux

Lumír Jasiok

Centrum výpočetní techniky,

Vysoká škola báňská – Technická univerzita
Ostrava

21. března 2002

Proč vznikly distribuce OS Linux

- Na počátku neexistovala správa balíčků.
- Vše se instalovalo z `.tar.gz` souborů.
- Začaly vznikat tzv. distribuce, které umožňovaly správu instalovaného softwaru.
- Mezi prvními:
 - Debian (`dpkg`)
 - RedHat (`rpm`)

Distribuce Debian

- Linux potřeboval rychlý, praktický a efektivní způsob instalace softwaru s automaticky vyřešenými závislostmi.
- Vývojáři Debianu vyvinuli APT (Advanced Packaging Tool)
- Výrazně usnadňuje správu softwaru.

Konfigurace APT :

Soubor `/etc/apt/sources.list`

```
deb http://ftp.de.debian.org/debian/ stable main non-free contrib
```

```
deb-src http://ftp.de.debian.org/debian/ stable main non-free contrib
```

```
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
```

- `deb` – Označuje binární balík
- `deb-src` – Zdrojové soubory programu + Debianovský kontrolní soubor `.dsc` a soubor s popisem změn nutných pro Debian `.diff`
- APT umí pracovat s archivy typu:
 - `http, ftp, file`

Používání APT lokálně :

V případě, že máme mnoho `.deb` balíčků a chceme si nechat automaticky vyřešit závislosti při instalaci.

1) `mkdir /root/debs`

2) `cd /root; vim soubor`

Vložíme do souboru: `package priority section`

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

Do souboru `/etc/apt/sources.list` vložíme :

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

- Se zdrojovými soubory to lze udělat podobně (`Packages.gz` zaměníme za `Sources.gz` a `deb` za `deb-src`).

Přidání CDROM do /etc/apt/sources.list :

```
apt-cdrom add
```

Update dostupných balíčků :

```
apt-get update
```

Instalace balíčků :

```
apt-get install xchat
```

Pokud soubor nezávisí na ničem co není nainstalováno, pak dojde k instalaci.

- V případě, že na námi inst. balíku závisí jiné, pak je výstup následující:

```
bash-2.05a# 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]
```

Užitečné parametry apt-get :

Dojde pouze ke stažení souboru do `/var/cache/apt/archives`

```
apt-get install -d xchat
```

Simulace instalačního procesu

```
apt-get install -s xchat
```

Pokud upgradujeme pomocí příkazu `apt-get upgrade`, pak se nám zobrazí balíky, které budou nahrazeny

```
apt-get upgrade -u
```

Pokud chceme současně instalovat i odstraňovat programy, můžeme použít následující syntaxi:

```
apt-get install nautilus gnome-panel-
```

Anebo naopak můžeme současně odstraňovat i instalovat:

```
apt-get remove gnome-panel nautilus+
```

Výše uvedený příkaz nesmaže u odstraňovaných balíčků konfigurační soubory.
Smazání konf. souborů spolu s balíčkem můžeme pomocí:

```
apt-get --purge remove gnome-panel
```

Upgrade distribuce:

Nejdříve přidáme nové CD, nebo URL do `/etc/apt/sources.list` a pak zadáme:

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

APT a dselect:

<http://www.debian.org/releases/stable/i386/dselect-beginner.cs.html#contents>

Nabízí tzv. `recommended` (doporučené) a `suggested` (navržené). Kromě toho ještě zobrazujesamozřejmě `depends` (nutné).

Zachování určité verze balíku:

V souboru: `/etc/apt/preferences`

Struktura vypadá takto:

```
Package: <balík>
```

```
Pin: <pin definice>
```

```
Pin-Priority: <priorita>
```

Například, pokud chceme zachovat balík `sylpheed`, pak syntaxe vypadá takto:

```
Package: sylpheed
```

```
Pin: version 0.4.99*
```

Instalovat můžeme jenom balíčky, které mají stejnou verzi (může se změnit jen číslo revize).

`Pin-Priority` je volitelná a standardně má hodnotu 989.

S prioritou < 0 balíček nesmí být instalován.

Priorita od 0 do 100 označuje balíček, který není nainstalován a nemá dostupnou verzi.

Priorita 100 je určena pro nainstalované balíčky.

Nad 100 jsou balíčky, které mohou být instalovány.

Pokud chceme mít možnost donwgradovat zpět, pak musíme zvolit pin $>$ než 1000.

Například pokud chcete vyzkoušet “testing” distribuce, a pak se vrátit zpět, pak zadáme:

```
Package: *
```

```
Pin: release a=stable
```

```
Pin-Priority: 1001
```

Pokud chceme downgradovat, pak stačí spustit příkaz:

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

Hledání názvu balíčku:

```
apt-cache search atari
```

Výstup:

```
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
```

Informace o balíčcích:

```
apt-cache show stella
```

Výstup:

```
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
```

Bude se lišit výstup pro nenainstalovaný a nainstalovaný balíček

Výpis hlavních informací o balíčcích

```
apt-cache showpkg penguin-command
```

Výpis závislostí mezi balíčky:

```
apt-cache depends penguin-command
```

Vyhledání souborů:

```
[root]@[ / ] # 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
```