

Package management



In dit hoofdstuk gaan we kijken naar package management, het betere archiveren!

Doelstelling

Aan het eind van dit hoofdstuk is de cursist bekend met:

- het "package concept"
- het RPM packaging system
- het Debian packaging system
- shared libraries
- package afhankelijkheden en conflicten
- de bij al bovenstaande acties behorende veel gebruikte opties en commando's

het package concept

Meer dan een verzameling software. Een heel scala aan onderdelen en tools:

- package zelf. Verzameling van (veel) files behorende bij een bepaalde functionaliteit.
- package database. informatie over elk geïnstalleerd file.
- afhankelijkheden. Ook geregistreerd in de database
- verificatie (met checksums) om de integriteit te bewaren
- utilities voor install, upgrade, remove, ...
- utilities om zelf packages te maken (geen examen eis).

RPM

RPM Redhat Package Manager

- Ook populair bij andere distributies (Suse, Centos, Fedora)
- cross-platform, niet alleen voor i386 (PowerPC, SPARC)
- naamgeving: **packagenaam-versie-build.arch.rpm**
apache2-2.2.8-28.8.i586.rpm
 - versie: bepaald door programmeur(s)
 - build: bepaald door de package distributeur (bv. Redhat, Suse, ...)
- hernoemen mag: naamgeving is niet bepalend, de juiste informatie zit in het package.
- mogelijk verschil in de rpm utilities per distributie
- verschil in afhankelijkheden tussen distributies

Veel packages te vinden op rpmfind.net. In het algemeen: gebruik de RPM's gemaakt voor jouw distributie!

RPM commando (1/2)

```
rpm [actie] [optie]... [package-file|package-naam]
```

package-file: volledige naam, bv. apache2-2.2.8-28.8.i586.rpm

package-naam: apache2

acties:

- `-i` install
- `-U` install of upgrade indien reeds aanwezig
- `-e` uninstall
- `-q` query. Geef info over een package
- `--rebuilddb`
- `--checksig package-file`
- `-V -p package-file`

RPM commando (2/2)

```
rpm [actie] [optie]... [package-file|package-naam]
```

package-file: volledige naam, bv. apache2-2.2.8-28.8.i586

package-naam: apache2

opties:

- `-vh` bij `-i`, `-U`: produceer hashmarks, laat de voortgang zien
- `--test` bij `-i`, `-U`: check voor conflicten e.d. zonder install
- `-f file` bij `-q`: geef info over een package wat *file* bevat.
- `-l` bij `-q`: laat de files zien in het package
- `-a` bij `-q`: laat alle geïnstalleerde packages zien
- `-p package-file` bij `-q`: query (uninstalled) *package -file*

YUM

Met rpm: nog wel zelf rpm zoeken en downloaden..

- Schil om rpm: yum
- Gebruikt repositories voor het vinden en downloaden (en vervolgens installeren) van rpm packages
(alleen downloaden van packages: yumdownloader)
- zorgt ook voor de afhankelijkheden
- `yum [actie] [package]...`
 - **acties:** `install`, `update`, `upgrade`, `search`, `info`, `remove`
- config in `/etc/yum.conf` en `/etc/yum.repos.d`
- repository toevoegen:
 - met een yum install van de repository
 - of downloaden (`wget`) van ".repo file" en plaatsen in `/etc/yum.repos.d`

Debian packaging

- Debian heeft eigen package management: de advanced package tool (apt)
- Vergelijkbaar met rpm maar niet compatible
- Ook populair (Debian, Ubuntu, Mint)
- nadruk bij Debian op bug-vrij, veilig en strikt open source (daardoor weinig releases ("lange release-cycle") bij Debian. Ubuntu/Mint zijn meer "bleeding edge")
- minder cross-platform dan rpm

dpkg commando

`dpkg [optie] [actie] [package-file|package-naam]`

package-file: volledige naam, bv. `apache2_2.2.14-5ubuntu8.7_i386.deb`

package-naam: `apache2`

actie:

- `-i` install
- `-l` laat alle geïnstalleerde package zijn
- `-r` remove
- `-p package-naam` laat informatie over een geïnstalleerd package zien
- `--info package-file` geef info over een **niet** geïnstalleerd *package-file*
- `-L` laat de files zien in een geïnstalleerd package
- `--contents package-file` laat de files zien in het **niet** geïnstalleerd *package-file*
- `-S file` Zoek naar package wat *file* bevat

opties:

- `--no-act` bij `-i`, `-r`: simuleer maar doe niks ("dry run")
- `-D` debug mode

apt commando's (1/3)

Met `dpkg`: nog wel zelf `.deb` zoeken en downloaden..

- Schil om `dpkg`: de apt commando's (`apt-cache`, `apt-get`, `aptitude`)

`apt-cache actie string`

Zoek in de debian package database (de "package cache")

- veel gebruikt: `apt-cache search ^zoekstring`

Meer info:

- `apt-cache pkgnames apache`
- `apt-cache show apache2`

apt commando's (2/3)

`apt-cache`: zoeken en info

`apt-get`: install, update, remove (vergelijkbaar met yum)

`apt-get [optie] [actie] [package-naam]...`

actie:

- `install` installeer package (gevonden met `apt-cache`)
- `update` resync the de package db met de repositories. De repositories staan in `/etc/apt/sources.list`
- `dist-upgrade` slimme upgrade. Altijd eerst een `apt-get update` doen
- `remove` verwijder package
- `purge` verwijder package en alle configuratie files

opties:

- `-s` simuleer maar doe niks ("dry run")
- `-f --fix-broken` probeert (dependency) problemen op te lossen
- `-m --fix-missing` negeert packages die ontbreken (bv. niet gedownload kunnen worden)

(package namen vind je met `apt-cache`)

apt commando's (3/3)

`dselect`, `aptitude`: ook interactieve (text) modes

`synaptic`: volledig grafisch

Alledrie nogal wat overhead maar handig om naar packages te zoeken

`dpkg-reconfigure`: runt het originele (configuratie script)

- `dpkg-reconfigure postfix`

converteren

Soms vind je bv. alleen maar een rpm package van een specifiek programma terwijl je met een debian systeem werkt.

Met het debian programma `alien` kun je packages converteren:

```
alien --to-deb zenity-2.28.0-1.el6.x86_64.rpm
```

of andersom:

```
alien --to-rpm zenity_3.4.0-0ubuntu4_i386.deb
```

Beide package systemen moeten wel geïnstalleerd zijn.

`alien` kan ook overweg met tar-files

shared libraries

- "vergelijk" met Windows DLL's
- veel voorkomende code groeperen in libraries
- + beschikbaar voor verschillende programma's
- + voorkomt dat ieder programma zelf de code moet hebben
 - + minder memory gebruik (shared memory)
 - + minder diskgebruik
- + minder hercompileren van programma's
- namen eindigen veelal op `.so` (shared objectcode)
- - complexe afhankelijkheden kunnen in de weg zitten
- - om de libraries te vinden is extra configuratie nodig
- - library file niet beschikbaar: veel programma's niet meer werkend

Hele bekende: de GNU C library (glibc)

vinden van shared libraries

- Zoekpaden in `/etc/ld.so.conf`
 - Doet veelal een include van `/etc/ld.so.conf.d/*.conf`
- Run `ldconfig` als pad verandert

`ldconfig` zonder opties update zowel de library cache als de links

 - `-N` update alleen de symlinks, niet de cache
 - `-X` update alleen de cache, niet de symlinks
 - `-v` verbose
- Met de environment variabele `LD_LIBRARY_PATH` zijn paden toe te voegen aan de bestaande zoekpaden:
 - `export LD_LIBRARY_PATH=/usr/local/myprog/lib`
 - veelal voor tijdelijke (test) aanpassingen

oplossen van library problemen

Een soms voorkomende fout is het missen van een library, bv.:

```
vim: error while loading shared libraries: libgpm.so.2:  
cannot open shared object file: No such file or directory
```

Zaken waar u aan kunt denken om dit op te lossen:

- zorg dat uw systeem up-to-date is
- kijk met `ldd` (bv. `ldd /usr/bin/vim`) naar de afhankelijkheden
- kijk of het probleem eventueel met een symlink is op te lossen. Niet echt aan te bevelen maar kan in bepaalde situaties afdoende zijn. Daarna wel `ldconfig` runnen.
- re-install programma (package), in dit geval `vim`
- kopieer de ontbrekende library van een vergelijkbaar systeem

Samenvatting

- twee belangrijke package managers "rpm" en "apt" (er zijn er meer)
- gebruik de utilities `yum` (`apt-cache/apt-get`)
- veel mogelijkheden: ontdek de opties!
- shared libraries spelen een prominente rol

Oefeningen

Tijd voor oefening!