

Cours GNU/linux

Gestion de package

Présentation Distribution

Unix :

Un boot loader

Un Kernel

Une arborescences

Outils :

- Le processus Init
- Shell
- les commandes posix
- Services systèmes

Distribution :

Outils d'installation du système

Outils de gestions de packages

Packages

- Ensemble de fichiers

Binaires

- Exécutable
- librairies

Scripts

Fichiers de configuration

Fichiers de données

Package .deb

Nomenclature :

apache_1.3.31-6_i386

Package : apache

Version : 1.3.31-6

Arch : i386

structure :

Archive ar (ordonnée)

- Debian-binary : contient la version du packaging
- control.tar.gz : fichiers de contrôle liés à la gestion du package
- data.tar.gz : archive des fichiers à installer/désinstaller du système

Control.tar.gz

Conffile : liste des fichiers de config

Control : info sur le package

Md5sums : checksum du contenu de data.gz

Config : script de configuration

Scripts de pré post install suppression

- preinst
- prerm
- postinst
- postrm

TD : oscultation d'un package

on se crée un rep de travail

```
# mkdir work ; cd work
```

on récupère un package

```
# wget http://ftp.fr.debian.org/debian/pool/main/a/at/at_3.1.13-2+deb7u1_amd64.deb
```

on extrait les fichiers de l'archive ar (du package)

```
# ar -x at_3.1.13-2+deb7u1_amd64.deb
```

on se crée un sous rep extrait les fichiers control

```
# mkdir control ; cd control/
```

```
# tar zvxf ../control.tar.gz
```

on osculte les fichiers

on décompresse l'archive et on osculte

```
# unxz data.tar.xz
```

```
# tar tvf data.tar
```

Le gestionnaire de package : dpkg

Liste des packages installé : “dpkg -l”

Contenu d'un package installé : “dpkg -L \$PKG”

Contenu d'un package : “dpkg -c \$PKG”

Package owner d'un fichier : “dpkg -S \$FILE”

coreutil

util-linux

netbase

Config / statut

/etc/dpkg/dpkg.cfg

/var/log/dpkg.log

/var/lib/dpkg/available : package disponible

/var/lib/dpkg/status : package connu et statut

/var/lib/dpkg/info : métadonnées extraite des packages installés

Utilisation dpkg 1/2

`dpkg -i` ou `--install` : Installation ou mise à jour simple d'un package

Pas de résolution : Simple blocage de l'install

Processus

- Extraction des fichiers de controles
- Si mise à jour, Pre suppression de l'ancien packet
- Pré installation du nouveau
- Sauvegarde des fichiers anciens et extraction des nouveaux
- Si mise à jour, Post suppression de l'ancien packet
- Configuration du packet et post-installation

`dpkg -r` ou `--remove` : désinstallation d'un package

Pas de gestion des dépendances

Processus : Pre suppression, Suppression, Post suppression

Conserve les fichiers de configuration (`--purge` supprime tout)

Utilisation dpkg 2/2

`dpkg --reconfigure` : reconfiguration d'un package installé

Sauvegarde des fichiers de configuration locale, execution de la configuration et la post installation.

`dpkg --repack` : reconstruit le package à partir de fichiers locaux (sauvegarde du produit et de sa conf)

Gestion de configuration :

<https://wiki.debian.org/DpkgConfHandling>

Conversion de package : alien

Alien permet de convertir des package rpm en dpkg

Produit propriétaire fournis en rpm

Attention, par défaut il n'intègre pas les scripts pre/post install/désinstall

Et pour cause : le script peut être adapté à la distribution

Commande :

alien --scripts -d <package.rpm>

Debian APT 1/2

Outil front-end de gestion de package.

Assure la gestion des dépendances

Permet une gestion complète de la distribution

Indépendant de l'outil de package : (cf : apt-rpm)

Installation :

```
apt-get install $PKG
```

Supression :

```
apt-get remove $PKG
```

```
apt-get --purge remove $PKG
```

Debian APT 2/2

Mise à jour “apt-get update”

resynchronisation de l'index des packages :
(/var/lib/dpkg/available)

Mise à jour système “apt-get upgrade”

Mise à jour de tous les package installés

Configuration APT

Configuration :

`/etc/apt/apt.conf`

`cat /etc/apt/apt.conf.d/*`

Dépôts / Repository :

`/etc/apt/sources.list`

`cat /etc/apt/sources.list.d/*`

– Canaux :

Type : deb / deb-src

Url : repository (voir /dist/)

Tag du canal : distribution-canal / type de package main contrib non-free

Depots de package

Canaux Debian :

<http://ftp.fr.debian.org/debian/>

Distributions ou branche

Type de packages :

- Maintenance debian : main
- Contrib : contrib, non-free

Canaux Ubuntu :

<http://fr.archive.ubuntu.com/ubuntu>

Distribution ou branche

Type de packages :

- Maintenance ubuntu : main, restricted
- Contib : universe, multiverse

Répertoire binary-

Arch : fichier

package.gz

Trouvez le package
sur le repository

openssh-server

TD

Que fait script est lancé avant la désinstallation de openssh-server ?

Retrouver le script.

Le lire.

Librairies / bibliothèques partagées

Fonctions API compilés dans un fichier

binaire sans fonction main()

Cmd : `file /lib/ld-2*.so`

- Elf : Executable and Linking Format (cf wikipedia)

Compilation

Statique :

- Import directe des librairies compilée dans le binaire compilé
- Pas de dépendance sur les librairies
- Occupation mémoire du binaire

Dynamique :

- Création d'un lien vers la bibliothèque partagées (commande `ld` , ou `gcc -l<lib>`)

Librairies / bibliothèques partagées

Utilisation par les executables : utilisation de ld-linux.so

Recherche de la bibliothèque dans /lib et /usr/lib et dans
\$LD_LIBRARY_PATH

Chargement en mémoire de la bibliothèque et mise à disposition des
API pour tout les binaires

Cache :

- Cmd : ldconfig (sous root)
 - Mise à jour du cache
 - Mise à jour des liens symbolique vers les bibliothèques
- /etc/ld.so.cache (donnée binaire)
- /etc/ld.so.conf configuration du cache
- /etc/ld.so.conf.d/* configuration du cache déposé par les packet

Cmd ldd : liste les librairies utilisées par le binaire :

- ldd /bin/ls

Compilation de package

Pourquoi ?

Pas de package associée à votre distribution

Besoin d'utiliser un patch

PB de compatibilité sur les packages disponibles

Processus

Récupération du code source

Extraction (/opt/src)

Configuration : `./configure --help`

`./configure --option=value`

→ création du makefile

Compilation : `make <objet>`

Installation : `make install`

Netoyage : `make clean`

Commandes avancées

Cmd : apt-rdepend

Fournies récursivement tout les package dépendants

Cmd : auto-apt \$cmd

Récupère les erreurs de fichiers manquant sur la commande \$cmd et les recherche et les propose à l'installation

Copie de selections de package :

```
dpkg --get-selections > mes_paquets.txt # sauvegarde des packets installé
```

```
dpkg --set-selections < mes_paquets.txt # selectionner les packets
```

```
# Installation des packet sélectionné :
```

```
apt-get update
```

```
apt-get dselect-upgrade
```

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

```
apt-get upgrade
```