



C06-pkg - COURS LINUX - MTN

Guillaume ASTIER

26/02/16



Table des matières

L'ajout et la suppression de logiciels	1
Les différentes méthodes	1
Installation à partir des sources	2
Installer un logiciel à partir de ses sources	2
Compilation d'un logiciel depuis les sources	2
Les gestionnaires de paquets	3
Installer un programme à partir d'un paquet (Debian)	4
Installer un programme à partir d'un dépôt (Debian)	5
Les paquets debian	5
Le format de paquet Deb	6
Arborescence	6
Les scripts d'installation suppression	7
Compilation et installation	8
Gestion des processus	8
Caractéristiques d'un processus	8
Visualiser les processus	9
Commandes de gestion des processus	9
Gestion de tâches dans une session interactive	9

L'ajout et la suppression de logiciels

Les différentes méthodes

Deux méthodes pour installer un logiciel :

- à partir des sources (*.tar.gz, *.tgz, *.tar.bz2, etc.)
- à partir d'un paquet (*.deb, *.rpm, etc.)

Installation à partir des sources

Installer un logiciel à partir de ses sources

Les sources d'un logiciel sont généralement accompagnées des fichiers 'configure' et 'Makefile.in'. Ces fichiers permettent de faciliter la compilation du logiciel. Les étapes à suivre sont alors les suivantes :

-
- Décompresser l'archive avec les outils **tar** et **gzip/bzip2** (par exemple : 'tar xzf .tgz').
 - Vérifier les dépendances et adapter le programme à l'environnement local (commande : './configure').
 - Compiler les sources (commande : 'make').
 - Copier les fichiers binaires dans leurs emplacements définitifs (commande : 'make install').
-

Compilation d'un logiciel depuis les sources

Dans certains cas il est possible que vous vous retrouviez avec une source donc les outils Makefile et/ou configure ne soit pas présent.

Comme vue dans les précédent TP et cours nous allons nous appuyer sur une source en .c car c'est le langage le plus proche de linux et unix.

```
1 /* commentaire : bonjour isen source */
2
3 #include<stdio.h>
4 main()
5 {
6     printf("Bonjour l'isen");
7 }
```

Pour compiler ce dernier nous avons besoin d'un compilateur pour transformer le code source en fichier binaire compréhensible par la machine

- Le compilateur sous linux pour des sources en .c est : gcc
- Le compilateur sous linux pour des sources en .cpp est : g++

```
1 gcc -Wall -o [NomDuFichierBinaire] -c [NomFichierSource.c]
```

L'option Wall permet d'afficher les message de type préventif

Avantages

- Permet d'installer un logiciel qui n'est pas dans la distribution
- Permet d'avoir un binaire optimisé pour votre architecture

Inconvénients

- La compilation peut être très longue
- Les dépendances difficiles à résoudre

Ressources associé et documentation MakeFile [TP associé][2]

Les gestionnaires de paquets

Un paquet (*package*) est une archive générée avec 'ar' qui contient au minimum :

- les fichiers qui seront installés sur le disque (binaire(s) précompilé(s) et fichier(s) de configuration).
- des scripts exécuté à l'installation ou la désinstallation du paquet.
- des informations de contrôle (intégrité, description, dépendances, etc.).

L'extension pour les paquets Debian est *.deb.

Chaque distribution a son système de "paquet" (par exemple RedHat et Suse utilisent RPM).

Installer un programme à partir d'un paquet (Debian)

Debian utilise une convention de nommage pour les fichiers de paquet :

```
1  nompaquet_version-révisionDebian_plateforme.deb
2
3  * nompaquet : nom du paquet.
4  * version : version originale du logiciel du paquet.
5  * révisionDebian : numéro de révision du paquet.
6    Ce numéro est donné par le mainteneur du paquet.
7  * plateforme : la plateforme cible du paquet
8    (i386, amd64, alpha, etc).
9  * Si le paquet n'est pas dépendant d'une plateforme,
10   il est noté all.
```

Le programme de base pour manipuler un paquetage est 'dpkg' :

```
1  dpkg [option] [fichier.deb ...]
2
3  -i      : installe les fichiers contenus dans le paquet
4  -r      : désinstalle le paquet sans supprimer les
5            fichiers de conf
6  --purge : désinstalle le paquet et les fichiers de conf
7  -l      : liste de tous les paquets installés
8  -L      : affiche les fichiers contenus dans le paquet
9  -S      : affiche le nom du paquet contenant le fichier
```

Avantages

- Gain de temps, les binaires sont déjà compilés !
- C'est le seul moyen d'installer un paquet qui n'est pas présent dans la distribution.

Inconvénients

- la gestion des dépendances est purement informative, sans résolution automatique.

Installer un programme à partir d'un dépôt (Debian)

Le système APT (Advanced Package Tool) permet une gestion intelligente des paquets (sources, versions, dépendances et conflits).

On définit les sources des paquets dans un fichier de configuration et 'aptitude' s'occupe d'aller chercher les paquets sur un CD ou un DVD, ou sur une source réseau.

```
1 # cat /etc/apt/sources.list
2 deb http://mirror.ovh.net/debian/ wheezy main contrib non-free
3 deb-src http://mirror.ovh.net/debian/ wheezy main contrib non-free
4 [...]
5 deb http://security.debian.org/ wheezy/updates main contrib non-free
```

Pour plus d'informations : 'man sources.list'.

-
- 'aptitude update' : Met à jour la liste des paquets disponibles.
 - 'aptitude install p1 p2' : Installe les paquets p1 et p2 et tous les paquets dont ils dépendent.
 - 'aptitude remove p1' : Désinstalle le paquet paquet1 sans effacer ses fichiers de configuration.
 - 'aptitude purge p2' : Idem mais en effaçant ses fichiers de configuration.
 - 'aptitude clean' : Efface du disque dur les paquets téléchargés.
 - 'aptitude search str' : Recherche un paquet dont le nom ou la description contient la chaîne "str".
 - 'aptitude show p3' : Affiche les caractéristiques et la description du paquet p3.

Les paquets debian

Fichiers .deb (paquets binaires)

Moyen puissant et pratique pour distribuer des logiciels aux utilisateurs

Un des deux formats de paquets les plus courants avec RPM

Universel :

```
1 * 30 000 paquets binaires dans Debian
2
3 * la plupart des logiciels libres sont empaquetés dans Debian !
4
5 * 12 portages (architectures), dont 2 non Linux (Hurd et kFreeBSD)
6
7 * Utilisé aussi par 120 distributions dérivées de Debian
```

Le format de paquet Deb

- Fichier .deb : une archive ar

```
1 $ ar tv wget_1.12-2.1_i386.deb
2 rw-r--r-- 0/0 4 Sep 5 15:43 2010 debian-binary
3 rw-r--r-- 0/0 2403 Sep 5 15:43 2010 control.tar.gz
4 rw-r--r-- 0/0 751613 Sep 5 15:43 2010 data.tar.gz
```

- debian-binary : version du format de fichier .deb, "2.0"
- control.tar.gz : métadonnées sur le paquet control, md5sums, (pre|post)(rm|inst), triggers, shlibs...
- data.tar.gz : fichiers de données du paquet

Arborescence

Un paquet Debian est constitué de deux parties notables : un dossier contenant le nom du paquet, ses dépendances, etc. d'un côté, et les fichiers du programme de l'autre. L'arborescence d'un paquet Debian se présente sous cette forme :

```
1 dossier_du_paquet/
2   DEBIAN/
3     control
4     preinst
5     postinst
6     prerm
7     postrm
8   usr/
9   bin/
10  votre_programme
```

Le fichier `contôle` contient les informations nécessaires à son installation. Il doit comporter les indications ci-dessous :

- Package : le nom de votre paquet. (utilisé par apt)
 - Version : 1.0 (ex)
 - Section : Quel est le type de paquet (admin,mail,contrib,devel,X11 ...)
 - Priority : l'importance de votre paquet pour le système
 - Architecture : all,x86,amd64,sparc ...
 - Depends : les nom des paquets nécessaire au bon fonctionnement
 - Maintainer : le nom et l'email de celui qui s'occupe de la création du `.deb`.
 - Description : la description de votre paquet.
 - Homepage : l'adresse du site internet
-

Ex :

```
1 Package: mtnisen
2 Version: 1.0_alpha
3 Section: admin
4 Priority: optional
5 Architecture: all
6 Depends: bash, figlet
7 Maintainer: Doc Brown
8 Description: Ne sert à rien a part à faire jolie
9 Homepage: http://isen.gastier.net
```

Les scripts d'installation suppression

Vous pouvez, si c'est nécessaire (c'est facultatif), ajouter des scripts qui seront exécutés avant/après l'installation/la suppression du paquet. Ils auront pour nom :

- `preinst` : exécuté avant l'installation
- `postinst` : exécuté après l'installation

- prerm : exécuté avant la suppression
- postrm : exécuté après la suppression

N'oubliez pas les droits 755 sur les fichiers

Compilation et installation

Une fois l'arborescence terminée il suffit de construire le paquet avec l'outil dpkg-deb :

```
1 sudo dpkg-deb --build nom_du_paquet
```

Une fois la construction terminée vous avez un paquet du nom de votre répertoire et pour l'installer il faut utiliser :

```
1 sudo dpkg -i nom_du_paquet.deb
```

Gestion des processus

Caractéristiques d'un processus

Linux étant un système multitâche, plusieurs programmes peuvent tourner en même temps.

Lorsque un programme est lancé, un processus est créé. Il s'agit d'une entité active qui possède des caractéristiques (priorité, registres, compteur ordinal, mémoire, etc.). Certaines caractéristiques varient dans le temps.

Le système identifie les processus à l'aide d'un identifiant (PID, Process Identification).

La gestion des processus sous Linux est dite hiérarchisée. Un processus peut lui-même créer un autre processus (fork + exec). Il devient donc le processus parent (PPID, Parent Process ID)

Visualiser les processus

```
1 $ ps -ef
2 UID    PID  PPID  C  STIME      TTY      TIME    COMMAND
3 root     1     0    0  Dec 6      ?         1:02    init
4 jean   319  300    0 10:30:30    ?         0:02    vim
5 olivier 321  319    0 10:30:34    ttyt1     0:02    csh
6 olivier 324  321    0 10:32:12    ttyt1     0:00    ps
```

Autres commandes: 'pstree', 'top', etc.

Commandes de gestion des processus

Les commandes nice et renice permettent de positionner ou de modifier la priorité d'un processus. L'intervalle des valeurs possibles va de -20 (priorité la plus favorable) à 19 (la moins favorable).

```
1 # nice -n -20 find / -type f -name "*.sh"
2
3 $ renice 20 7643
```

La commande kill permet d'envoyer un signal à un processus. kill, killall, pgrep/pkill, xkill

```
1 kill 456
2 kill -9 -1
3 pkill firefox
```

Gestion de tâches dans une session interactive

Les processus interactifs sont lancés et gérés à partir du terminal de l'utilisateur. Ils peuvent fonctionner en deux modes :

- Mode foreground (le processus monopolise le terminal jusqu'à sa terminaison)
- Mode background (le processus travaille en parallèle avec le terminal)

Mode foreground:

```
1 $ sleep 10
2 [...]
```

Mode background:

```
1  $ sleep 10 &
2  [1] 3384
3  $
```

La séquence de touches “ctrl-z” et les commandes jobs, bg, fg, permettent de faire passer un processus d’un mode à l’autre.