Niki Kovacs (kikinovak)

I'm a 45-year old Austrian living in South France since 1991. I've started hacking away on computers as a kid, first on an 8080 monoprocessor board with 512 bytes (sic) of RAM, then on a Commodore VC-20 with 3,5 kilobytes RAM, which I programmed in Basic and in Assembler, both of which I seem to have completely forgotten.

Around the late eighties, after three frustrating semesters in Computer Science at the Technische Universität Wien, my passion for computers fell off me like a skin, and I discovered another passion, for books. In the early nineties, I moved to France, where I studied literature at the University Paul Valéry in Montpellier.

And then, I gradually rediscovered computers, first as a simple tool - no more than glorified typewriters. And slowly and gradually, my old “hacker self” came back. I discovered Slackware Linux in 2001, and since that date, I'm one hundred percent GNU/Linux and FOSS.

I started to work as a tech writer for some printed magazines in France:

- *Linux Pratique* (2003 - 2008)
- *Planète Linux* (2010 -)

In 2006 I was hired by my local town hall for a two-year job, which consisted in installing and networking eleven public libraries around Sommières, using only Linux and free software.

In 2009 I created Microlinux, a company specialized in Linux-based solutions for professionals. Among my clients, I have local town halls, public libraries, schools, small and bigger companies.

That same year I published two books (in French) about Linux. Linux aux petits oignons, a 530-page cookbook-style bible based on CentOS, and explaining all the basic *NIX principles to Joe User. And Ubuntu efficace, a 300-page book about the innards of Ubuntu.

Over the years I've been using quite many different distributions:

- Debian;
- Mandrake (before it became Mandriva);
- Libranet (remember?);
- CentOS;
- Ubuntu;
- Arch.

Over 2011 and 2012, I did some distro-hopping to find out which distribution would be “best” (for me) to use in production environments. So I installed some networks using a mix of Debian, CentOS, Ubuntu LTS and Slackware. In the end, I decided to adopt Slackware. Though the use of Slackware requires more research in the beginning, I like it for its flexibility and its rock-solid stability.

If you're curious, check out my daily work here:

```
$ svn co svn://svn.tuxfamily.org/svnroot/microlinux/slackware
```

My contributions so far:

- Synchronize your network with NTP
- Synchronisez les machines de votre réseau avec NTP (french translation)
- Roaming profiles with NFS and NIS

Work in progress

Stuff I'm currently working on, and which will eventually end up as a page in the Slackware Documentation Project.

Gérer les paquets tiers confortablement avec sbopkg

L'outil `sbopkg` est une application tierce qui facilite énormément la construction et l'installation de paquets tiers à partir des scripts de SlackBuilds.org.

Installer sbopkg

Aller sur le site http://sbopkg.org et télécharger le paquet :

```
# links http://sbopkg.org
```

Suivre le lien `Downloads` et télécharger le `Package`. Puis :

```
# installpkg sbopkg-0.36.0-noarch-1_cng.tgz
```

Utiliser sbopkg

Lancer `sbopkg` en invoquant son nom en tant que root. Au premier lancement, `sbopkg` propose de créer une série de répertoires nécessaires pour son fonctionnement. Confirmer avec la touche Ctrl :

```
# sbopkg
The following directories do not exist:
Variable                   Assignment
--------                   ----------
REPO_{ROOT,NAME,BRANCH} -> /var/lib/sbopkg/,SBo/,14.0
LOGFILE directory -------> /var/log/sbopkg
QUEUEDIR ----------------> /var/lib/sbopkg/queues
SRCDIR ------------------> /var/cache/sbopkg
You can have sbopkg create them or, if these values are incorrect, you can
abort to edit your config files or pass different flags.
(C)reate or (A)bort?:
```

La première chose à faire, c'est de synchroniser `sbopkg` avec le contenu de SlackBuilds.org avec l'option `Sync with the remote repository`. C'est une bonne idée de lancer une synchronisation avant toute installation. Une fois que la synchronisation est terminée (***SYNC COMPLETE***), confirmez simplement.

Dans l'exemple, nous allons installer l'utilitaire `unrar`.

1. Choisissez l'outil de recherche `Search`.
2. Dans le champ de recherche, tapez `unrar`.
3. Dans la fenêtre des résultat, sélectionnez la ligne `system/unrar`.
4. L'écran subséquent vous permet d'effectuer toute une série d'opérations. Vous pouvez par exemple visualiser le fichier `unrar.info` pour vérifier s'il n'y a pas de dépendances externes (il n'y en a pas).
5. Construisez le paquet avec `Process - Download/build/install unrar`.
6. Confirmez simplement `Install - Download, build and install`.
7. Démarrez l'opération avec `Start`.

Si tout s'est bien passé, `sbopkg` construit et installe automatiquement le paquet. Appuyez sur n'importe quelle touche pour revenir dans `sbopkg`.

Installer des paquets tiers précompilés

Certains sites proposent des paquets tiers précompilés pour Slackware. En règle générale, il vaut toujours mieux compiler ses propres paquets, ne serait-ce que pour de simples raisons de sécurité. Après tout, on peut mettre tout et n'importe quoi dans un paquet et lui donner le nom qu'on veut.

Les paquets fournis par Eric Hameleers

Une exception peut être faite pour les paquets fournis par Eric Hameleers, développeur Slackware connu sous le nom de “AlienBob”. Eric fournit de nombreux paquets tiers très populaires parmi la communauté Slackware, notamment les “usines à gaz” comme KDE ou LibreOffice qui nécessitent des heures voire des journées entières de compilation.

- Versions récentes de KDE : http://alien.slackbook.org/ktown/
- Les dernières versions de LibreOffice : http://taper.alienbase.nl/mirrors/people/alien/slackbuilds/libreoffice/
- OpenJDK : http://taper.alienbase.nl/mirrors/people/alien/slackbuilds/openjdk/

- VLC : http://taper.alienbase.nl/mirrors/people/alien/restricted_slackbuilds/vlc/
- Paquets de compatibilité 32-bit pour Slackware64 : http://www.slackware.com/~alien/multilib/

Visitez le blog d'Eric pour être au courant des dernières nouveautés :

- http://alien.slackbook.org/blog/

Documentation

Pages man :

- `installpkg(8)`
- `upgradepkg(8)`
- `removepkg(8)`
- `slackpkg(8)`
- `sbopkg(8)`
- `sbopkg.conf(8)`

Articles en ligne :

- http://docs.slackware.com/slackbook:package_management
- http://www.linux.com/learn/tutorials/261878-intro-to-slackware-package-management
- http://slackbuilds.org/howto/
- http://archive09.linux.com/feature/148826

Livres :

- Linux Cookbook, Carla Schroder, Installing and Managing Software, 14-15
- ibid., Installing Programs from Source Code, 55-57

Gestion des paquets logiciels

Anatomie d'un paquet Slackware

Un paquet Slackware est une simple archive compressée au format TGZ ou TXZ qui contient :

- l'arborescence des répertoires et des fichiers ;
- les scripts post-installation ;
- la description du paquet.

Le nom de chaque paquet fournit une série d'informations :

- le nom de l'application ;
- la version de l'application ;
- l'architecture du paquet ;
- le numéro de build.

Exemples :

- emacs-24.2-i486-1
- mozilla-firefox-15.0.1-i486-1
- vim-7.3.645-x86_64-1

Gérer les paquets Slackware avec les outils traditionnels

Depuis les toutes premières versions, Slackware fournit une collection d'outils simples - les `pkgtools` - qui permettent d'installer, de mettre à jour et de supprimer des paquets logiciels :

- installpkg
- removepkg
- upgradepkg
- explodepkg
- makepkg

Installer un paquet

Installer l'éditeur Emacs [1)] à partir du DVD d'installation :

```
# mount /dev/cdrom /mnt/cdrom
# cd /mnt/cdrom/slackware/e
# installpkg emacs-24.2-i486-1.txz
Verifying package emacs-24.2-i486-1.txz.
Installing package emacs-24.2-i486-1.txz [ADD]:
PACKAGE DESCRIPTION:
# emacs (GNU Emacs)
#
# Emacs is the extensible, customizable, self-documenting real-time
# display editor. If this seems to be a bit of a mouthful, an
# easier explanation is that Emacs is a text editor and more. At
# its core is an interpreter for Emacs Lisp, a dialect of the Lisp
# programming language with extensions to support text editing.
# This version supports X.
#
# http://www.gnu.org/software/emacs/
#
Executing install script for emacs-24.2-i486-1.txz.
Package emacs-24.2-i486-1.txz installed.
```

Si l'on utilise le jeu de CDs, Emacs se trouve sur le 1er CD-Rom.

Vérifier si un paquet est installé

L'installation du paquet a créé une entrée dans `/var/log/packages` :

```
# ls /var/log/packages/em*
/var/log/packages/emacs-24.2-i486-1
```

Pour savoir si un paquet est installé, il suffit de vérifier s'il dispose d'une entrée correspondante dans `/var/log/packages`. Exemple :

```
# ls /var/log/packages/*firefox*
/var/log/packages/mozilla-firefox-15.0.1-i486-1
```

Firefox est installé sur le système, dans la version 15.0.1. Autre exemple :

```
# ls /var/log/packages/kdebase*
/bin/ls: impossible d'accéder à /var/log/packages/kdebase*: Aucun fichier ou
dossier de ce type
```

Aucun paquet `kdebase-*` n'est installé.

Supprimer un paquet

Pour supprimer un paquet installé, on utilise `removepkg`. Il suffit de donner le nom du paquet en argument.

Exemple :

```
# removepkg emacs
```

On peut également fournir le nom complet en argument. Dans ce cas, il vaut mieux invoquer la commande en se plaçant dans `/var/log/packages` et en utilisant la complétion automatique :

```
# cd /var/log/packages
# removepkg emacs-24.2-i486-1
```

Mettre à jour un paquet

Slackware fournit des mises à jour de sécurité des paquets pour chaque version. Pour se renseigner sur les dernières actualités autour des mises à jour, visiter le site officiel :

```
# links http://www.slackware.com
```

1. Suivre le lien `ChangeLogs`.
2. Repérer `Slackware-stable ChangeLog`.
3. Lire le fichier `ChangeLog.txt` correspondant à l'architecture du système.

On pourra également utiliser le navigateur Links pour récupérer les mises à jour manuellement. Avant de lancer Links, créer un répertoire `/root/updates` dans lequel on rangera les mises à jour :

```
# cd
# mkdir updates
```

```
# cd updates/
# links mirrors.slackware.com
```

1. Suivre le lien `Slackware File Tree`.
2. Repérer le répertoire correspondant à la version et à l'architecture du système.
3. Aller dans le répertoire `patches/packages`.
4. Télécharger les mises à jour disponibles.

Quitter Links et installer les mises à jour comme ceci :

```
# upgradepkg bind-9.9.1_P4-i486-1_slack14.0.txz

+==============================================================================
===
| Upgrading bind-9.9.1_P3-i486-1 package using ./bind-9.9.1_P4-
i486-1_slack14.0.txz
+==============================================================================
===
Pre-installing package bind-9.9.1_P4-i486-1_slack14.0...
Removing package /var/log/packages/bind-9.9.1_P3-i486-1-
upgraded-2012-11-21,12:14:32...
  --> Deleting /usr/doc/bind-9.9.1-P3/CHANGES
  --> Deleting /usr/doc/bind-9.9.1-P3/COPYRIGHT
  --> Deleting /usr/doc/bind-9.9.1-P3/FAQ
  ...
Verifying package bind-9.9.1_P4-i486-1_slack14.0.txz.
Installing package bind-9.9.1_P4-i486-1_slack14.0.txz:
PACKAGE DESCRIPTION:
bind (DNS server and utilities)
#
# The named daemon and support utilities such as dig, host, and
# nslookup.  Sample configuration files for running a simple caching
# nameserver are included.  Documentation for advanced name server
# setup can be found in /usr/doc/bind-9.x.x/.
#
Executing install script for bind-9.9.1_P4-i486-1_slack14.0.txz.
Package bind-9.9.1_P4-i486-1_slack14.0.txz installed.
Package bind-9.9.1_P3-i486-1 upgraded with new package
./bind-9.9.1_P4-i486-1_slack14.0.txz.
```

Autre exemple :

```
# upgradepkg iptables-1.4.14-i486-2_slack14.0.txz
```

En savoir plus sur le contenu d'un paquet

À chaque paquet installé correspond une entrée dans `/var/log/packages`. Il s'agit de simples fichiers texte qui nous renseignent sur le contenu d'un paquet. Par exemple :

```
# less /var/log/packages/wget-1.14-i486-1
```

```
PACKAGE NAME:     wget-1.14-i486-1
COMPRESSED PACKAGE SIZE:     478.5K
UNCOMPRESSED PACKAGE SIZE:     2.0M
PACKAGE LOCATION: /var/log/mount/slackware/n/wget-1.14-i486-1.txz
PACKAGE DESCRIPTION:
wget: wget (a non-interactive network retriever)
wget:
wget: GNU Wget is a free network utility to retrieve files from the
wget: World Wide Web using HTTP and FTP, the two most widely used Internet
wget: protocols.  It works non-interactively, thus enabling work in the
wget: background after having logged off.
wget:
wget: The author of Wget is Hrvoje Niksic <hniksic@srce.hr>.
wget:
wget:
wget:
FILE LIST:
./
install/
install/slack-desc
install/doinst.sh
usr/
usr/bin/
usr/bin/wget
usr/man/
usr/man/man1/
usr/man/man1/wget.1.gz
usr/info/
usr/info/wget.info.gz
...
```

Gérer les paquets Slackware avec slackpkg

L'utilitaire `slackpkg` a été officiellement inclus dans Slackware depuis la version 13.0. Il permet de gérer les paquets Slackware de manière beaucoup plus confortable.

Deux mises en garde s'imposent :

1. Seuls les paquets officiels sont gérés par `slackpkg`.
2. La gestion des dépendances reste toujours à la charge de l'administrateur.

Configuration initiale

Éditer `/etc/slackpkg/mirrors` et décommenter *une seule* source de paquets au choix, par exemple :

```
# /etc/slackpkg/mirrors
...
# FRANCE (FR)
```

```
ftp://mirror.ovh.net/mirrors/ftp.slackware.com/slackware-14.0/
# http://mirror.ovh.net/mirrors/ftp.slackware.com/slackware-14.0/
```

Attention à ne pas se tromper de section et à ne pas utiliser un site miroir de `Slackware-current`, sous peine de se retrouver avec la version de développement de Slackware !

Si l'on préfère gérer les paquets localement et faire fi des mises à jour, on peut également utiliser le DVD d'installation comme source de paquets. Dans ce cas, on modifiera le point de montage défini par défaut :

```
# /etc/slackpkg/mirrors
...
#-------------------------------------------------------------------
# Local CD/DVD drive
#-------------------------------------------------------------------
cdrom://mnt/cdrom/
...
```

Ne pas oublier de monter le DVD avant chaque invocation de `slackpkg` :

```
# mount /dev/cdrom /mnt/cdrom
```

Mettre à jour les informations sur les paquets disponibles :

```
# slackpkg update
```

Notez bien que cette commande n'installe pas de mises à jour de paquets. Elle synchronise seulement les informations sur ce que l'on *peut* installer.

Avant de rechercher, d'installer ou de mettre à jour un paquet, c'est une bonne idée d'invoquer `slackpkg update` pour être sûr d'avoir des infos à jour sur les paquets disponibles.

Installer des paquets

Exemple avec un seul paquet :

```
# slackpkg install mplayerplug-in
```

Il suffit de confirmer l'installation dans l'écran subséquent, et le paquet est directement récupéré et installé.

On peut également fournir plusieurs paquets en argument :

```
# slackpkg install mplayerplug-in bittorrent
```

Les groupes de paquets sont également gérés :

```
# slackpkg install kde
```

Ou encore :

```
# slackpkg install xfce
```

Supprimer des paquets

Exemple avec un seul paquet :

```
# slackpkg remove mplayerplug-in
```

Là aussi, il suffit de confirmer la suppression dans l'écran récapitulatif.

Supprimer plusieurs paquets à la fois :

```
# slackpkg remove mplayerplug-in bittorrent
```

Les groupes de paquets sont également gérés pour la suppression :

```
# slackpkg remove kde
```

Ou encore :

```
# slackpkg remove xfce
```

Mettre à jour des paquets

Lorsqu'une mise à jour est disponible pour un paquet, on peut l'installer comme ceci :

```
# slackpkg upgrade iptables
```

De même pour plusieurs paquets :

```
# slackpkg upgrade mozilla-firefox mozilla-thunderbird
```

Dans la pratique quotidienne, on mettra à jour l'intégralité du système :

```
# slackpkg upgrade-all
```

Rechercher des paquets ou des fichiers individuels

Rechercher un paquet spécifique :

```
# slackpkg search k3b
Looking for k3b in package list. Please wait... DONE
The list below shows all packages with name matching "k3b".
[uninstalled] - k3b-2.0.2_20120226.git-i486-1
```

Si le paquet est déjà installé, on obtiendra le résultat suivant :

```
# slackpkg search Terminal
Looking for Terminal in package list. Please wait... DONE
The list below shows all packages with name matching "Terminal".
[ installed ] - Terminal-0.4.8-i486-1
```

On peut également chercher des fichiers individuels, ce qui affichera le cas échéant le ou les paquets contenant le fichier en question :

```
# slackpkg file-search libncurses.so
Looking for libncurses.so in package list. Please wait... DONE
The list below shows the packages that contains "libncurses\.so" file.
[ installed ] - aaa_elflibs-14.0-i486-4
[ installed ] - ncurses-5.9-i486-1
```

Si l'on veut en savoir plus sur le contenu d'un paquet :

```
# slackpkg info mesa

PACKAGE NAME:  mesa-8.0.4-i486-1.txz
PACKAGE LOCATION:  ./slackware/x
PACKAGE SIZE (compressed):  19208 K
PACKAGE SIZE (uncompressed):  83930 K
PACKAGE DESCRIPTION:
mesa: mesa (a 3-D graphics library)
mesa:
mesa: Mesa is a 3-D graphics library with an API very similar to that of
mesa: another well-known 3-D graphics library.  :-)  The Mesa libraries are
mesa: used by X to provide both software and hardware accelerated graphics.
mesa:
mesa: Mesa was written by Brian Paul.
mesa:
```

Faire le ménage

Supprimer tous les paquets tiers qui ne font pas partie de la distribution officielle :

```
# slackpkg clean-system
```

Dans l'écran récapitulatif, il suffit de désélectionner les paquets que l'on souhaite garder.

On peut également se servir de `slackpkg` pour réparer un paquet endommagé. Admettons que j'aie accidentellement supprimé le fichier `/usr/bin/glxgears`. Dans un premier temps, il me faut rechercher le paquet qui le contient :

```
# slackpkg file-search glxgears
Looking for glxgears in package list. Please wait... DONE
The list below shows the packages that contains "glxgears" file.
[ installed ] - mesa-8.0.4-i486-1
```

À partir de là, il me suffit de réinstaller le paquet en question :

```
# slackpkg reinstall mesa
```

Recompiler des paquets officiels

Slackware fournit le code source de l'ensemble du système dans le répertoire `source`. À chaque paquet du système correspond un répertoire source. Ces répertoires source contiennent généralement :

- le code source de l'application ou de la bibliothèque en question ;
- sa recette de fabrication sous forme de fichier `*.SlackBuild` ;
- le descriptif du paquet, nommé `slack-desc` ;
- parfois, un fichier post-installation nommé `doinst.sh` ;
- d'autres fichiers comme les patches, les entrées de menu, etc.

Fabriquer un paquet à partir du code source

Dans l'exemple ci-dessous, nous allons compiler l'application `Terminal` à partir des sources fournies par Slackware. Au préalable, il faut donc désinstaller le paquet correspondant s'il est installé :

```
# removepkg Terminal
```

Choisir un endroit pour ranger le code source et les scripts, par exemple :

```
# cd
# mkdir -pv source/Terminal
mkdir: création du répertoire « source »
mkdir: création du répertoire « source/Terminal »
# cd source/Terminal/
# links mirrors.slackware.com
```

Récupérer le contenu de `source/xfce/Terminal` sur un miroir de Slackware. Au total, on a :

```
# ls -lh
total 1,4M
-rw-r--r-- 1 root root  821 nov.  24 15:09 slack-desc
```

```
-rw-r--r-- 1 root root 1,4M nov.  24 15:11 Terminal-0.4.8.tar.xz
-rw-r--r-- 1 root root 3,6K nov.  24 15:10 Terminal.SlackBuild
```

Rendre le fichier `Terminal.SlackBuild` exécutable et lancer la construction du paquet :

```
# chmod +x Terminal.SlackBuild
# ./Terminal.SlackBuild
```

Le script lance alors la construction du paquet. L'opération se termine par le message suivant :

```
Slackware package /tmp/Terminal-0.4.8-i486-1.txz created.
```

Il ne reste plus qu'à installer ce paquet :

```
# installpkg /tmp/Terminal-0.4.8-i486-1.txz
```

Modifier un paquet officiel

L'intérêt de recompiler un paquet officiel, c'est de pouvoir le modifier, par exemple en ajoutant ou en retirant certaines fonctionnalités. Dans l'exemple suivant, nous allons recompiler le paquet `audacious-plugins` pour modifier le lecteur audio Audacious. Celui-ci comporte deux interfaces graphiques au choix, et nous allons en supprimer une.

Pour commencer, supprimer le paquet s'il est installé :

```
# removepkg audacious-plugins
```

Ensuite, créer un endroit pour ranger le code source :

```
# cd /root/source
# mkdir audacious-plugins
# cd audacious-plugins
# links mirrors.slackware.com
```

Récupérer le contenu du répertoire `source/xap/audacious-plugins` et rendre le script `audacious-plugins.SlackBuild` exécutable :

```
# chmod +x audacious-plugins.SlackBuild
# ls -lh
total 1,4M
-rw-r--r-- 1 root root 1,4M nov.  24 15:28 audacious-plugins-3.3.1.tar.xz
-rwxr-xr-x 1 root root 4,0K nov.  24 15:28 audacious-plugins.SlackBuild*
-rw-r--r-- 1 root root  892 nov.  24 15:28 slack-desc
```

Maintenant, éditer `audacious-plugins.SlackBuild` et ajouter une option :

```
...
# Configure:
CFLAGS="$SLKCFLAGS" \
```

```
CXXFLAGS="$SLKCFLAGS" \
./configure \
  --prefix=/usr \
  --libdir=/usr/lib${LIBDIRSUFFIX} \
  --sysconfdir=/etc \
  --mandir=/usr/man \
  --enable-amidiplug \
  --disable-gtkui \          -> ajouter cette option
  --program-prefix= \
  --program-suffix= \
  ${ARCHOPTS} \
  --build=$ARCH-slackware-linux
...
```

Il ne reste plus qu'à construire et installer le paquet :

```
# ./audacious-plugins.SlackBuild
...
Slackware package /tmp/audacious-plugins-3.3.1-i486-1.txz created.
# installpkg /tmp/audacious-plugins-3.3.1-i486-1.txz
```

Choisir les options de compilation

Le script de configuration des sources (plus exactement la ligne à rallonge qui commence par `./configure` dans le SlackBuild) affiche souvent un résumé des options activées ou désactivées. Pour interrompre le processus de construction du paquet et afficher ce résumé, on peut temporairement éditer le SlackBuild comme ceci :

```
...
# Configure:
CFLAGS="$SLKCFLAGS" \
CXXFLAGS="$SLKCFLAGS" \
./configure \
  --prefix=/usr \
  --libdir=/usr/lib${LIBDIRSUFFIX} \
  --sysconfdir=/etc \
  --mandir=/usr/man \
  --enable-amidiplug \
  --program-prefix= \
  --program-suffix= \
  ${ARCHOPTS} \
  --build=$ARCH-slackware-linux

exit 1          -> ajouter cette commande pour interrompre le script

# Build and install:
make $NUMJOBS || make || exit 1
make install DESTDIR=$PKG || exit 1
...
```

Lancer le script, qui affichera un résumé de la configuration au bout de quelques secondes :

```
# ./audacious-plugins.SlackBuild
...
Configuration:
...

Interfaces
----------
GTK (gtkui):                            yes
Winamp Classic (skins):                 yes
```

Les options de configuration sont toutes fournies par le code source lui-même :

```
# tar xvf audacious-plugins-3.3.1.tar.xz
# cd audacious-plugins-3.3.1
# ./configure --help | less
...
--disable-speedpitch    disable Speed and Pitch effect plugin
--disable-gtkui         disable GTK interface (gtkui)
--disable-skins         disable Winamp Classic interface (skins)
--disable-lyricwiki     disable LyricWiki plugin (default=enabled)
...
```

Le script se charge déjà de décompresser les sources automatiquement dans `/tmp`. On peut donc très bien invoquer `./configure –help | less` à partir de ce répertoire, sans décompresser les sources dans le répertoire courant.

L'activation de certaines fonctionnalités comme par exemple la gestion de certains formats audio propriétaires dépend de la présence de certaines bibliothèques sur le système.

Une fois qu'on a choisi toutes les options de configuration, il ne reste plus qu'à supprimer la commande `exit 1` du script et lancer la compilation et l'installation :

```
# ./audacious-plugins.SlackBuild
...
Slackware package /tmp/audacious-plugins-3.3.1-i486-1.txz created.
# installpkg /tmp/audacious-plugins-3.3.1-i486-1.txz
```

Compiler des paquets tiers

Comparé à des distributions comme Ubuntu ou Debian, Slackware n'offre qu'un choix de paquets relativement limité. On en arrivera très vite au point de vouloir installer une application ou une bibliothèque qui n'est pas fournie par la distribution. Dans ce cas, que faire ?

Le portail SlackBuilds.org (http://slackbuilds.org) sera sans doute la meilleure adresse pour trouver

des paquets tiers. Attention, SlackBuilds.org n'est *pas* un dépôt de paquets binaires. Il s'agit plutôt d'une collection extrêmement bien fournie de scripts de compilation dûment soignés et testés, qui vous permettent de compiler à peu près n'importe quel paquet Slackware en un tournemain.

Compiler des paquets à partir des scripts de SlackBuilds.org

Dans l'exemple, nous allons compiler et installer le paquet `cowsay` à partir des scripts fournis par SlackBuilds.org.

Aller dans l'environnement de construction de paquets que nous avons défini plus haut :

```
# cd /root/source
```

À partir de là, télécharger :

1. l'archive compressée contenant les scripts pour construire le paquet ;
2. l'archive compressée contenant le code source du paquet.

Concrètement :

```
# links http://slackbuilds.org
```

1. Dans le champ de recherche en haut à gauche, taper `cowsay`, placer le curseur sur `Search` et confirmer par Entrée.
2. Dans la page des résultats de la recherche, suivre le lien `cowsay`.
3. Sur la page de `cowsay`, télécharger le SlackBuild (`cowsay.tar.gz`) et le code source (`cowsay-3.03.tar.gz`) et quitter Links.

Alternativement, vous pouvez utiliser `lynx` au lieu de `links`.

Voici nos deux archives téléchargées :

```
# ls -l cowsay*
-rw-r--r-- 1 root root 15136 nov.  25 08:14 cowsay-3.03.tar.gz
-rw-r--r-- 1 root root  2855 nov.  25 08:14 cowsay.tar.gz
```

Décompresser l'archive contenant les scripts :

```
# tar xvzf cowsay.tar.gz
cowsay/
cowsay/cowsay.SlackBuild.patch
cowsay/README
cowsay/slack-desc
cowsay/cowsay.SlackBuild
cowsay/cowsay.info
```

À la limite, on peut faire un peu de ménage et supprimer l'archive qui ne sert plus à rien :

```
# rm -f cowsay.tar.gz
```

Puis, déplacer le code source dans le répertoire nouvellement créé :

```
# mv -v cowsay-3.03.tar.gz cowsay/
« cowsay-3.03.tar.gz » -> « cowsay/cowsay-3.03.tar.gz »
```

Voici ce que l'on doit avoir :

```
# tree cowsay
cowsay
|-- cowsay-3.03.tar.gz
|-- cowsay.info
|-- cowsay.SlackBuild
|-- cowsay.SlackBuild.patch
|-- README
`-- slack-desc
```

Changer dans le répertoire, vérifier éventuellement si le script `cowsay.SlackBuild` est bien exécutable, puis exécuter ce script pour lancer la construction du paquet :

```
# cd cowsay/
# ls -l cowsay.SlackBuild
-rwxr-xr-x 1 kikinovak users 1475 mai   27  2010 cowsay.SlackBuild*
# ./cowsay.SlackBuild
...
```

Là encore, si tout se passe bien, l'opération produit un paquet dans `/tmp`, et plus exactement dans le répertoire `$OUTPUT` défini par le script :

```
...
Slackware package /tmp/cowsay-3.03-noarch-1_SBo.tgz created.
```

Il ne reste qu'à installer ce paquet avec `installpkg` :

```
# installpkg /tmp/cowsay-3.03-noarch-1_SBo.tgz
# cowsay Et voilà !
 -------------
< Et voilà ! >
 -------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

Gérer les dépendances de paquets

Certains paquets nécessitent la présence d'autres paquets sur le système pour compiler (*build*

dependencies) et/ou fonctionner (*runtime dependencies*) correctement. Dans certains cas, un paquet requis peut lui-même dépendre d'autres paquets, et ainsi de suite.

À titre d'exemple, jetons un oeil sur la page de `libgnomeprint` de SlackBuilds.org. La description du paquet est suivie d'un avertissement :

```
This requires: libgnomecups.
```

Chaque collection de scripts contient par ailleurs un fichier `*.info` qui explicite le ou les paquets requis. Jetons un oeil sur le fichier `libgnomeprint.info`, et nous y trouvons un champ REQUIRES :

```
PRGNAM="libgnomeprint"
VERSION="2.18.8"
HOMEPAGE="http://www.gnome.org"
...
REQUIRES="libgnomecups" ----> dépendances du paquet
...
```

Le champ REQUIRES a été introduit depuis Slackware 14.0.

Cela signifie tout simplement qu'avant de construire le paquet `libgnomeprint`, nous devons impérativement construire et installer le paquet `libgnomecups`.

En dehors des dépendances requises, un paquet peut également présenter des dépendances optionnelles, qui ajoutent certaines fonctionnalités. L'éditeur Leafpad, par exemple, peut être construit avec les dépendances optionnelles `libgnomeprint` et `libgnomeprintui`.

Gérer les paquets tiers confortablement avec sbopkg

L'outil `sbopkg` est une application tierce qui facilite énormément la construction et l'installation de paquets tiers à partir des scripts de SlackBuilds.org.

Installer sbopkg

Aller sur le site http://sbopkg.org et télécharger le paquet :

```
# links http://sbopkg.org
```

Suivre le lien `Downloads` et télécharger le `Package`. Puis :

```
# installpkg sbopkg-0.36.0-noarch-1_cng.tgz
```

Utiliser sbopkg

Lancer `sbopkg` en invoquant son nom en tant que root. Au premier lancement, `sbopkg` propose de

créer une série de répertoires nécessaires pour son fonctionnement. Confirmer avec la touche Ctrl :

```
# sbopkg
The following directories do not exist:
Variable                  Assignment
--------                  ----------
REPO_{ROOT,NAME,BRANCH} -> /var/lib/sbopkg/,SBo/,14.0
LOGFILE directory -------> /var/log/sbopkg
QUEUEDIR ----------------> /var/lib/sbopkg/queues
SRCDIR ------------------> /var/cache/sbopkg
You can have sbopkg create them or, if these values are incorrect, you can
abort to edit your config files or pass different flags.
(C)reate or (A)bort?:
```

La première chose à faire, c'est de synchroniser `sbopkg` avec le contenu de SlackBuilds.org avec l'option `Sync with the remote repository`. C'est une bonne idée de lancer une synchronisation avant toute installation. Une fois que la synchronisation est terminée (***SYNC COMPLETE***), confirmez simplement.

Dans l'exemple, nous allons installer l'utilitaire `unrar`.

1. Choisissez l'outil de recherche `Search`.
2. Dans le champ de recherche, tapez `unrar`.
3. Dans la fenêtre des résultat, sélectionnez la ligne `system/unrar`.
4. L'écran subséquent vous permet d'effectuer toute une série d'opérations. Vous pouvez par exemple visualiser le fichier `unrar.info` pour vérifier s'il n'y a pas de dépendances externes (il n'y en a pas).
5. Construisez le paquet avec `Process - Download/build/install unrar`.
6. Confirmez simplement `Install - Download, build and install`.
7. Démarrez l'opération avec `Start`.

Si tout s'est bien passé, `sbopkg` construit et installe automatiquement le paquet. Appuyez sur n'importe quelle touche pour revenir dans `sbopkg`.

Installer des paquets tiers précompilés

Certains sites proposent des paquets tiers précompilés pour Slackware. En règle générale, il vaut toujours mieux compiler ses propres paquets, ne serait-ce que pour de simples raisons de sécurité. Après tout, on peut mettre tout et n'importe quoi dans un paquet et lui donner le nom qu'on veut.

Les paquets fournis par Eric Hameleers

Une exception peut être faite pour les paquets fournis par Eric Hameleers, développeur Slackware connu sous le nom de “AlienBob”. Eric fournit de nombreux paquets tiers très populaires parmi la communauté Slackware, notamment les “usines à gaz” comme KDE ou LibreOffice qui nécessitent des heures voire des journées entières de compilation.

- Versions récentes de KDE : http://alien.slackbook.org/ktown/
- Les dernières versions de LibreOffice :

http://taper.alienbase.nl/mirrors/people/alien/slackbuilds/libreoffice/
- OpenJDK : http://taper.alienbase.nl/mirrors/people/alien/slackbuilds/openjdk/
- VLC : http://taper.alienbase.nl/mirrors/people/alien/restricted_slackbuilds/vlc/
- Paquets de compatibilité 32-bit pour Slackware64 : http://www.slackware.com/~alien/multilib/

Visitez le blog d'Eric pour être au courant des dernières nouveautés :

- http://alien.slackbook.org/blog/

Documentation

Pages man :

- `installpkg(8)`
- `upgradepkg(8)`
- `removepkg(8)`
- `slackpkg(8)`
- `sbopkg(8)`
- `sbopkg.conf(8)`

Articles en ligne :

- http://docs.slackware.com/slackbook:package_management
- http://www.linux.com/learn/tutorials/261878-intro-to-slackware-package-management
- http://slackbuilds.org/howto/
- http://archive09.linux.com/feature/148826

Livres :

- Linux Cookbook, Carla Schroder, Installing and Managing Software, 14-15
- ibid., Installing Programs from Source Code, 55-57

1)
à condition qu'il ne soit pas déjà installé