Faire, puis installer des paquets avec sbopkg

Sbopkg est un outil en ligne de commande, avec des dialogues, qui synchronise un ensemble de scripts SlackBuild non-officiels, du dépôt SlackBuilds.org (“SBo”) pour créer des paquets Slackware. Le programme a une interface 'ncurses' qui vous permet de trouver et d'assembler les logiciels que vous voulez compiler et empaqueter à partir de leur source. Vous pouvez aussi l'utiliser de manière non-interactive si vous savez à l'avance quel est votre but - auquel cas il fonctionne en mode “lancez et n'y pensez plus”.

Les informations qui suivent vous aideront à installer sbopkg et à l'utiliser dans Slackware.

Notez que `sbopkg` crée des paquets *Slackware*. Comme pour tout programme de gestion de paquets Slackware, il vous faut être root pour utiliser le programme! Pour cela, appelez

```
$ su -
```

pour obtenir l'invite de commande du super-utilisateur dans son environnement.

Télécharger sbopkg:

Téléchargez sbopkg depuis: http://www.sbopkg.org/downloads.php
Vous pouvez utiliser les sources et construire un paquet vous-même, ou prendre seulement le paquet Slackware déjà préparé. La suite de cet article considère que vous avez téléchargé le paquet tout-prêt.

Installer sbopkg

1. Si, par exemple, le nom de fichier du paquet `sbopkg` que vous avez téléchargé dans votre répertoire courant est *sbopkg-0.36.0-noarch-1_cng.tgz*, la commande `installpkg` qui permettra d'installer sbopkg (ou bien, le mettre à jour s'il était déjà présent dans votre système) est celle-ci:

```
$ su -
# cd /chemin/vers/mes_telechargements/
# upgradepkg --install-new sbopkg-0.36.0-noarch-1_cng.tgz
```

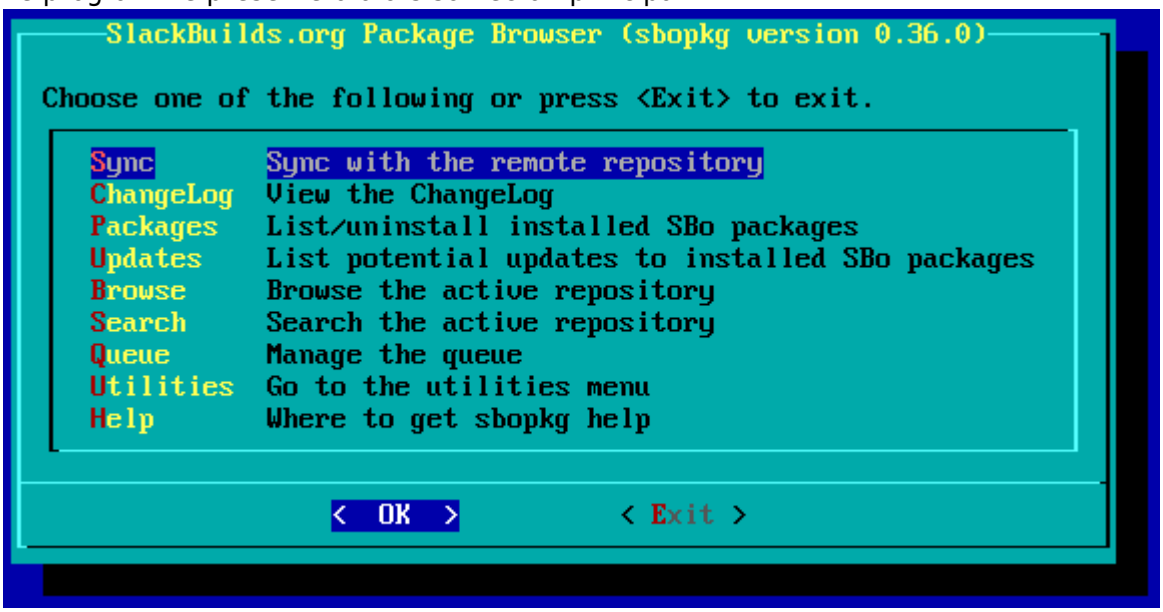
Configurer sbopkg

1. Lisez la documentation de `sbopkg` ! Voyez http://www.sbopkg.org/docs.php , sans oublier qu'il existe une page de man pour `sbopkg`.

2. Au premier lancement de sbopkg, il vous sera demandé si vous êtes d'accord pour que le programme crée les fichiers de configuration nécessaires:

```
# 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
TMP ---------------------> /tmp/SBo

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?:
```

. Choisissez C pour créer ces répertoires nécessaires.
Le programme présentera alors son écran principal:



3. Il vous faut commencer par synchroniser votre ordinateur avec le dépôt de SlackBuilds.org - i.e. ce qui consiste à faire créer par sbopkg une copie locale de tous les SlackBuilds enregistrés sur le serveur distant, pour la version de Slackware que vous utilisez. Ou bien vous le faites au moyen du premier article de menu 'Sync' dans l'écran principal en affichage semi-graphique 'curses', ou bien vous faites appel à la ligne de commande:

```
# sbopkg -r
Syncing with the remote repository into /var/lib/sbopkg/SBo/14.0.
receiving incremental file list
```

```
<long list of filenames...>

sent 451288 bytes  received 36904793 bytes  371702.30 bytes/sec
total size is 35271012  speedup is 0.94

Rsync with the SBo repository for Slackware 14.0 complete.

Importing SBo repository for Slackware 14.0 GPG Key...
Import done.

***SYNC COMPLETE***
```

4. Maintenant vous pouvez choisir les paquets que vous voulez faire construire depuis leurs sources, par `sbopkg`.

Utiliser les listes de fichiers avec sbopkg

Le programme `sbopkg` lui-même procure un bon niveau d'automatisation: l'interface interactive *curses* vous aide à gérer tout le travail de téléchargement des fichiers du code source, en vous permettant de choisir plusieurs programmes puis de les compiler et empaqueter tous en une seule fois.
Mais on peut automatiser ce processus encore davantage, par l'emploi des queuefiles qui contiennent des séquences prédéfinies de fichiers 'build' pour des ensembles de paquets. Ces séquences 'queuefiles' elles-mêmes seront invoquées de manière récursive vous permettant ainsi de créer une chaine complexe de construction de paquets très aisément.

Heureusement Mauro Giachero (un co-développeur de sbopkg) a créé des queuefiles pour la *totalité* du dépôt SlackBuilds.org. Voyez le dépôt git http://gitorious.org/sbopkg-slackware-queues. Chaque programme qui se trouve dans le dépôt SBo a son propre fichier associé queuefile, indiquant les dépendances qu'il faut compiler *avant* de pouvoir compiler le programme que vous voulez installer.

L'URL suivante donne une liste des paquets individuels ainsi qu'un fichier compressé tar.gz de la table complète de queuefiles pour Slackware 14:
http://gitorious.org/sbopkg-slackware-queues/sbopkg-slackware-queues/trees/sbo-14.0

Vous pouvez télécharger une archive de tout le contenu du dépôt pour la Slackware 14.0 à SlackBuilds.org depuis cette URL:
http://gitorious.org/sbopkg-slackware-queues/sbopkg-slackware-queues/archive-tarball/sbo-14.0 . Le fichier d'archive que vous obtiendrez se nommera “`sbopkg-slackware-queues-sbopkg-slackware-queues-sbo-14.0.tar.gz`”.

Pour extraire tous les fichiers *.sqf et les mettre dans le répertoire déterminé par `sbopkg`: “`/var/lib/sbopkg/queues/`” on utilise ces commandes:

```
# cd /var/lib/sbopkg/queues
# tar xvf /chemin/vers/mes_telechargements/sbopkg-slackware-queues-sbopkg-slackware-queues-sbo-14.0.tar.gz
# cd sbopkg-slackware-queues-sbopkg-slackware-queues
# mv * ../
# cd ..
```

```
# rm -r sbopkg-slackware-queues-sbopkg-slackware-queues
```

Maintenant chaque fichier de type 'queue' (*.sqf) peut être choisi pour installer tout paquet que vous avez envisagé (avec ses dépendances).

On peut obtenir les 'queuefiles' d'une autre façon:

```
cd /var/lib/sbopkg
```

Et puis:

```
git clone git://gitorious.org/sbopkg-slackware-queues/sbopkg-slackware-queues.git queues
```

Pour mettre à jour les queuesfiles dans l'avenir:

```
# cd /var/lib/sbopkg/queues
# git pull
```

Exemple d'utilisation de sbopkg

A titre d'exemple, installons Gramps. Le queuefile pour Gramps liste les dépendances suivantes, dans l'ordre d'installation:

```
orbit2
pyorbit
libbonobo
gnome-mime-data
gnome-vfs
libgnome
gnome-python
gramps
```

Prenez le 'queue file' “`gramps.sqf`” grâce à l'interface curses de sbopkg, et Gramps sera construit et installé avec succès dans votre système Slackware 14, avec toutes ses dépendances.

Une autre possibilité est d'employer l'interface moins interactive de la ligne de commande:

```
# sbopkg -i gramps
Both a queuefile and a package were found with the name "gramps".

Use (Q)ueuefile, (P)ackage, or (A)bort?: q

One or more build option files for the avahi
```

```
SlackBuild script were found:

Queuefile options: MONO=yes

Use (N)one, (Q)ueuefile, or (A)bort?: n

##########################################
       New queue process started on:
       Thu Nov 15 23:05:16 CET 2012
##########################################

++++++++++++++++++++++++++++++++++++++++++
PACKAGE BUILDING/INSTALLATION PRE-CHECK LOG
Using the SBo repository for Slackware 14.0
Queue Process:  Build and install

<long list of package checks...>

Pre-check complete.

Do you wish to proceed based on the search results above? Packages not
found will be skipped during the process.

(P)roceed or (Q)uit?: p
```

etcetera.

Références pour sbopkg

- http://www.sbopkg.org
- http://www.sbopkg.org/docs.php
- http://www.sbopkg.org/downloads.php
- http://www.sbopkg.org/queues.php

- http://gitorious.org/sbopkg-slackware-queues/sbopkg-slackware-queues/trees/master
- http://gitorious.org/sbopkg-slackware-queues/sbopkg-slackware-queues/trees/sbo-14.0

Sources

- Originally written by ldkraemer
- Contributions by Eric Hameleers
- Translated by P-M Averseng

howtos, software, sbo, package management, author ldkraemer, translator pierreaverseng

From:
https://docs.slackware.com/ - **SlackDocs**

Permanent link:
https://docs.slackware.com/fr:howtos:slackware_admin:building_packages_with_sbopkg

Last update: **2017/12/12 17:30 (UTC)**