

Gestion de paquets

La gestion de paquets est une part essentielle de toute distribution Linux. Chaque morceau de logiciel inclus dans Slackware, ainsi que de nombreux outils tiers sont distribués sous forme de code source qui peut être compilé, mais compiler ces centaines d'applications et bibliothèques est fastidieux et prend du temps. C'est pourquoi la plupart des gens préfèrent installer des paquets logiciels pré-compilés. En fait, lorsque vous installez Slackware, le programme **setup** fait principalement travailler les outils de gestion de paquets sur une liste de paquets. Nous allons présenter ici les différents outils utilisés pour gérer les paquets Slackware.

pkgtool

La façon la plus simple d'effectuer des opérations de maintenance sur des paquets est de lancer **pkgtool(8)**, une interface sous forme de menus pour d'autres outils. **pkgtool** vous permet d'installer ou supprimer des paquets mais également de voir le contenu de ceux-ci et la liste des paquets installés via une interface *ncurses* agréable.



pkgtool est un moyen facile et pratique pour effectuer la plupart des tâches élémentaires, mais pour des opérations avancées des outils plus souples sont nécessaires.

Installer, désinstaller et mettre à jour des paquets

Alors que **pkgtool** marque des points pour sa facilité, **installpkg(8)** est beaucoup plus adapté pour les tâches exotiques telles qu'installer un unique paquet, installer un ensemble complet de paquets ou scripter une installation. **installpkg** suit une liste de paquets à installer et les installe simplement, sans poser aucune question. Comme tous les outils de gestion de paquets Slackware, il part du principe que vous savez ce que vous faites et ne prétend pas être plus malin que vous. Sous sa forme la plus simple, **installpkg** prend juste une liste de paquets à installer et fait exactement ce à quoi vous vous attendez.

```
darkstar:~# installpkg blackbox-0.70.1-i486-2.txz
Verifying package blackbox-0.70.1-i486-2.txz.
Installing package blackbox-0.70.1-i486-2.txz:
PACKAGE DESCRIPTION:
# blackbox (Blackbox window manager)
#
# Blackbox is that fast, light window manager you have been looking for
# without all those annoying library dependencies.
#
# Also included in this package is the bbkeys utility for controlling
# keyboard shortcut commands from within Blackbox.
#
# The Blackbox home page is http://blackboxwm.sourceforge.net
#
Package blackbox-0.70.1-i486-2.txz installed.
```

Vous pouvez bien sûr installer plusieurs paquets à la fois et même utilisé des caractères joker du shell. Ce qui suit installe tous les paquets de la série "N" depuis un CD-ROM monté :

```
darkstar:~# installpkg /mnt/cdrom/slackware/n/*.txz
```

À n'importe quel moment, vous pouvez voir quels paquets sont installés sur votre système en listant le contenu de /var/log/packages, qui liste non seulement tous les programmes de votre système, mais également leur numéro de version. Disons que vous voulez savoir quels fichiers sont installés depuis un paquet en particulier, lancez **cat** pour afficher le contenu du paquet :

```
darkstar:~#cat /var/log/packages/foo-1.0-x86_64.txz
```

Cela renverra toutes sortes d'informations comme la taille du paquet, la description de ce qu'il fait et le nom et l'emplacement de chacun des fichiers installés comme composant de ce paquet.

Supprimer un paquet est aussi facile que d'en installer un. Comme vous pouvez vous y attendre, la commande pour cela est **removepkg(8)**. Indiquez simplement les paquets à supprimer et **removepkg** vérifiera le contenu de la liste des paquets installés et supprimera tous les fichiers et répertoires pour ce paquet avec une exception. Si un fichier est inclus dans plusieurs paquets. il ne sera pas touché et si un répertoire contient des fichiers plus récents, le répertoire sera laissé intact. De ce fait, supprimer un paquet prend beaucoup plus de temps qu'en installer un.

```
darkstar:~# removepkg blackbox-0.70.1-i486-2.txz
```

Finalement, mettre à jour est tout aussi facile grâce à (vous vous en doutiez) **upgradepkg(8)** qui commence par installer le nouveau paquet puis supprime tous les fichiers et répertoires en reliquat du précédent. Un point important à retenir est que **upgradepkg** ne vérifie pas si le paquet déjà installé à une version supérieure au "nouveau" paquet, il peut donc aussi être utilisé pour revenir à une ancienne version.

```
darkstar:~# upgradepkg blackbox-0.70.1-i486-2.txz
```

```
+=====
===
```

```
| Upgrading blackbox-0.65.0-x86_64-4 package using
./blackbox-0.70.1-i486-2.txz
+=====
===

Pre-installing package blackbox-0.70.1-i486-2...

Removing package
/var/log/packages/blackbox-0.65.0-x86_64-4-upgraded-2010-02-23,16:50:51...
--> Deleting symlink /usr/share/blackbox/nls/POSIX
--> Deleting symlink /usr/share/blackbox/nls/US_ASCII
--> Deleting symlink /usr/share/blackbox/nls/de
--> Deleting symlink /usr/share/blackbox/nls/en
--> Deleting symlink /usr/share/blackbox/nls/en_GB
...
Package blackbox-0.65.0-x86_64-4 upgraded with new package
./blackbox-0.70.1-i486-2.txz.
```

Tous ces outils disposent d'options pratiques. Par exemple, pour **installpkg**, `-root` installera un paquet dans un répertoire particulier. L'option `-dry-run` indiquera à **upgradepkg** de vous dire ce qu'il essaiera de faire sans effectuer de changement effectif sur le système ¹⁾. Pour plus de détails, vous devriez (comme toujours) vous référer aux pages du manuel.

Formats de compression des paquets

Dans le passé, tous les paquets Slackware étaient compressés avec l'utilitaire **gzip**(1), qui propose un bon compromis entre la vitesse et la taille de compression. Récemment, de nouveaux moyens de compressions ont été ajoutés et les outils de gestion de paquets ont été mis à jour pour les prendre en compte. Aujourd'hui, les paquets officiels Slackware sont compressés avec l'utilitaire **xz** et le terminent avec l'extension `.txz`. Les anciens paquets (et beaucoup de paquets tiers) utilisent toujours l'extension `.tgz`.

Il est important de souligner que les `.tgz` et `.txz` (ou de manière générique, les fichiers `.t?z`) sont des fichiers `.tar` compressés tout à fait classiques. Ceci a de nombreux avantages ; ils sont faciles à créer sur pratiquement tous les systèmes UNIX (beaucoup d'autres formats de paquets nécessitent une chaîne de compilation spéciale) et ils sont tout aussi simples à décortiquer.

Toutefois, il est aussi important de réaliser que bien que les paquets Slackware **soient** des fichiers `.t?z`, tous les fichiers `.t?z` ne sont pas des paquets Slackware. **installpkg** n'installera pas magiquement n'importe quel fichier `.t?z`, uniquement les paquets Slackware.

slackpkg

slackpkg est un outil de gestion automatique pour les paquets Slackware Linux. Il est apparu originellement dans `/extra` pour la version slackware-12.1 et depuis la version slackware-12.2 il est inclus dans la série `ap/` d'une installation de base.

De la même manière que vous utilisez **installpkg** pour installer des paquets Slackware depuis le

répertoire /extra du média d'installation, vous pouvez utiliser **slackpkg** pour récupérer des paquets depuis Internet et les installer. Ceci est particulièrement utile pour les mises à jour de sécurité ou les mises à jour importantes qui sont publiées sur les serveurs Slackware, que vous pouvez vouloir intégrer sur votre propre système.

Sans utiliser **slackpkg**, le processus serait le suivant :

1. Remarquer qu'une mise à jour à été publiée dans le fichier *changelog* de Slackware
2. Consulter votre miroir Slackware local pour trouver un lien pour télécharger le paquet
3. Télécharger le paquet depuis un miroir Slackware vers votre disque dur
4. Utiliser soit **installpkg** ou **pkgtool** pour installer le paquet téléchargé

Avec **slackpkg**, cela se réduit à :

1. Remarquer qu'une mise à jour pour **foo** à été publiée dans le fichier *changelog* de Slackware
2. **slackpkg install foo**

Clairement, cela simplifie grandement une tâche courante.

Pour utiliser **slackpkg**, configurez un miroir Slackware pour votre système en éditant `/etc/slackpkg/mirrors` en tant que root. Trouvez un miroir correspondant à votre version et votre architecture et dé-commentez-le. La liste de miroirs propose des accès FTP et HTTP mais vous ne devez en dé-commenter qu'**un seul**.

Une fois le miroir choisi, mettez à jour la liste des fichiers distant en lançant la commande initiale `slackpkg update`. Cela doit être effectué à chaque fois que vous constatez qu'une mise à jour a été publiée (consulter régulièrement le *changelog* de Slackware est recommandé ; lisez le [chapitre 18 "Gérer les mises à jour"](#) pour plus d'informations).

Pour rechercher un paquet, utilisez `slackpkg search foo`, et pour l'installez utilisez `slackpkg install foo`.

Une fois qu'un paquet à été installé avec **slackpkg**, il peut être supprimé ou mis à jour en utilisant **pkgtool** et les autres commandes de gestion de paquets tel que détaillé dans [Installer, désinstaller et mettre à jour des paquets](#)

Plus plus d'informations, consultez les pages de manuel de `slackpkg(8)` et `slackpkg.conf(5)`, et visitez leur site web <http://www.slackpkg.org/>.

rpm2tgz

Un des formats de paquets pour les logiciels Linux le plus répandu est RPM ; il n'est pas rare de trouver des développeurs proposer leur application en téléchargement soit sous forme de code source, soit sous forme de fichier RPM et rien d'autre. Dans ce cas, vous disposez de trois options :

1. Créer votre propre paquet Slackware
2. Compiler et installer directement depuis le code source
3. Convertir et installer le RPM

Compiler depuis le code source ou créer votre propre paquet Slackware n'est généralement pas aussi compliqué que vous pouvez le penser mais l'installer depuis les sources est souvent déconseillé car il n'y a pas de moyen simple de vérifier ce qui a été installé sur votre système après la commande

make install. Créer votre propre paquet Slackware est hors du cadre de ce chapitre. Il ne nous reste donc que l'outil for pratique **rpm2tgz**.

rpm2tgz convertit des paquets RPM en paquets Slackware qui peuvent être installés via **pkgtool** ou **installpkg**. Cela vous évite de créer votre propre paquet Slackware mais vous laisse la possibilité de le supprimer, de le mettre à jour et de vérifier son installation.



Alors qu'un paquet Slackware est un simple script shell et du code source, un paquet RPM peut être un labyrinthe de dépendances et d'instructions particulières en comparaison. De fait, **rpm2tgz** ne fonctionnera pas toujours, notamment avec les applications très complexes, et ne résoudra jamais les problèmes de dépendances par magie.

Pour essayer **rpm2tgz**, téléchargez un fichier RPM depuis une source de confiance et convertissez-le :

```
rpm2tgz foo-x.x.xx.rpm
```

Le résultat est un fichier .tgz. Vous pouvez sereinement supprimer le fichier RPM originel une fois la conversion terminée. Utilisez **installpkg** pour installer le paquet Slackware nouvellement créé, en ayant pris soin d'installer l'ensemble des dépendances nécessaires au fonctionnement du programme.

Navigation

Chapitre précédent : [Utilitaires réseau de base](#)

Chapitre suivant : [Gérer les mises à jour](#)

Sources

- Source originale : <http://www.slackbook.org/beta>
- Publication initiale d'Alan Hicks, Chris Lumens, David Cantrell, Logan Johnson
- Traduction initiale de [escaflown](#)
- Traduction de [Ellendhel](#)

[slackbook](#), [package management](#), [pkgtool](#), [slackpkg](#), [rpm2tgz](#)

¹⁾

Ceci n'est pas valable pour **installpkg**. Et si vous n'entrez pas le nom de l'option correctement, la mise à jour se fera pour de bon - NdT.

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

Permanent link:
https://docs.slackware.com/fr:slackbook:package_management

Last update: **2018/04/05 15:27 (UTC)**

