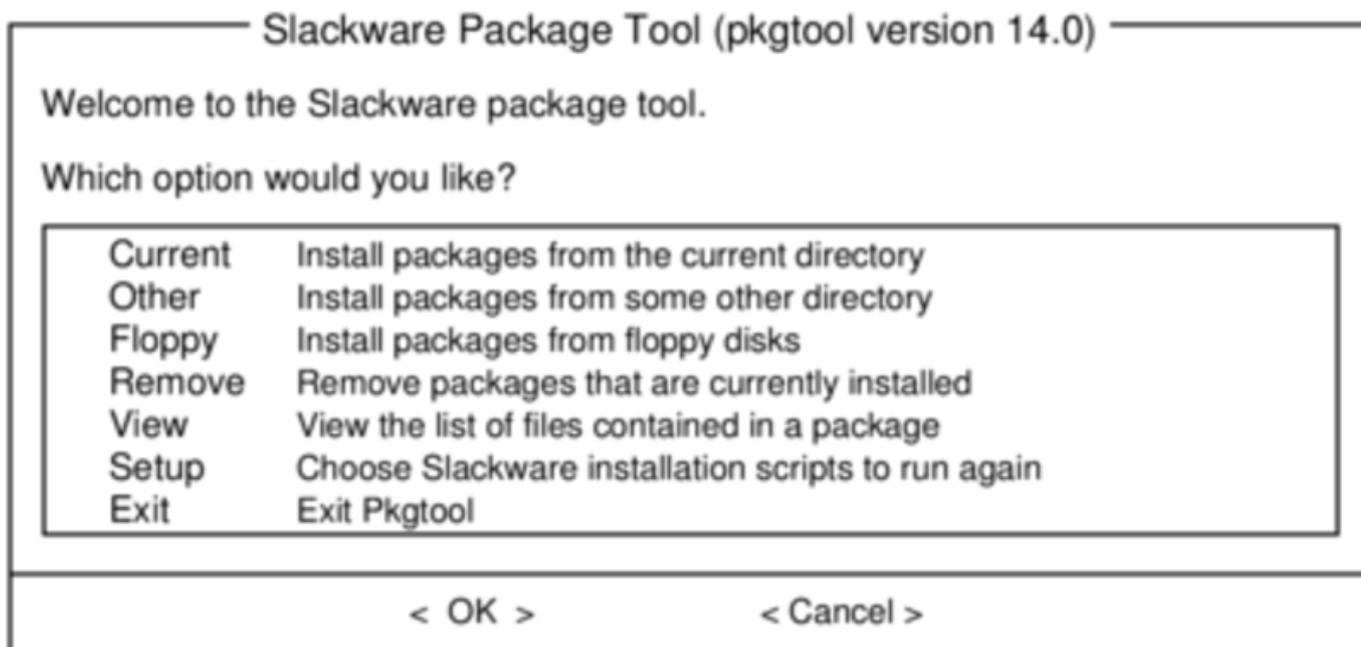


Gestión de paquetes

La gestión de paquetes es una parte esencial de cualquier distribución de Linux. Cada pieza de software incluida por Slackware, junto con muchas herramientas de terceros, se distribuye como código fuente que se puede compilar, pero compilar todas esas miles de aplicaciones y bibliotecas diferentes es tedioso y requiere mucho tiempo. Es por eso que muchas personas prefieren instalar paquetes de software precompilados. De hecho, cuando instaló Slackware, el programa de **configuración** funcionó principalmente ejecutando herramientas de administración de paquetes en una lista de paquetes. Aquí veremos las diversas herramientas utilizadas para manejar los paquetes de Slackware.

pkgtool

La forma más sencilla de realizar tareas de mantenimiento de paquetes es invocar **pkgtool** (8), una interfaz basada en menús para algunas de las otras herramientas. **pkgtool** le permite instalar o eliminar paquetes, así como ver el contenido de esos paquetes y la lista de paquetes actualmente instalados en una interfaz ncurses fácil de usar.



pkgtool es una forma cómoda y fácil de realizar las tareas más básicas, pero para un trabajo más avanzado se necesitan herramientas más flexibles.

Instalación, eliminación y actualización de

paquetes

Si bien **pkgtool** obtiene puntos por conveniencia, **installpkg** (8) es mucho más capaz de manejar tareas impares, como instalar rápidamente un solo paquete, instalar un conjunto completo de paquetes de discos o crear scripts para una instalación. **installpkg** toma una lista de paquetes para instalar, y simplemente los instala sin hacer preguntas. Como todas las herramientas de administración de paquetes de Slackware, asume que sabes lo que estás haciendo y no pretendes ser más inteligente que tú. En su forma más simple, **installpkg** simplemente toma una lista de paquetes para instalar, y hace exactamente lo que usted esperaría.

```
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.
```

Por supuesto, puede instalar varios paquetes a la vez y, de hecho, usar comodines de shell. Lo siguiente instala todos los paquetes de la serie "N" desde un CD-ROM montado:

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

En cualquier momento, puede ver qué paquetes están instalados en su sistema enumerando el contenido de `/var/log/packages`, que enumera no solo todas las aplicaciones de su sistema, sino también el número de versión. Si desea saber qué archivos individuales se instalaron como parte de ese paquete, **cat** le mostrará el contenido del paquete:

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

Esto devolverá todo, desde el tamaño del paquete, una descripción de lo que hace, y el nombre y la ubicación de cada archivo instalado como parte del paquete.

Eliminar un paquete es tan fácil como instalarlo. Como es de esperar, el comando para hacer esto es **removepkg** (8). Simplemente dígame qué paquetes eliminar, y **removepkg** comprobará el contenido de la base de datos de paquetes y eliminará todos los archivos y directorios de ese paquete con una advertencia. Si ese archivo se incluye en varios paquetes instalados, se omitirá y si un directorio tiene nuevos archivos, el directorio se dejará en su lugar. Debido a esto, la eliminación de paquetes lleva más tiempo que instalarlos.

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

Finalmente, la actualización es igual de fácil con (usted lo adivinó), **upgradepkg** (8) que primero instala un nuevo paquete, luego elimina los archivos y directorios que quedan del paquete anterior. Una cosa importante a recordar es que **upgradepkg** no verifica si el paquete instalado anteriormente tiene un número de versión más alto que el paquete “nuevo”, por lo que también se puede usar para cambiar a versiones anteriores.

```
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.
```

Todas estas herramientas tienen argumentos útiles. Por ejemplo, `-root` to **installpkg** instalará paquetes en un directorio arbitrario. El argumento `-dry-run` le **indicará** a **upgradepkg** que simplemente le diga qué intentará sin realizar cambios en el sistema. Para obtener detalles completos, debe (como siempre) consultar las páginas del manual.

Formatos de compresión de paquetes

En el pasado, todos los paquetes de Slackware se comprimían con la utilidad de compresión **gzip** (1), que era una buena combinación entre velocidad de compresión y tamaño. Recientemente, se han agregado nuevos esquemas de compresión y se han actualizado las herramientas de administración de paquetes para manejarlos. Hoy en día, los paquetes oficiales de Slackware están comprimidos con la utilidad **xz** y terminan con extensiones `.txz`. Los paquetes más antiguos (y muchos paquetes de terceros) todavía usan la extensión `.tgz`.

Vale la pena enfatizar que `.tgz` y `.txz` (o, más sucintamente, los archivos `.t?z`) son extensiones muy únicas y no estándar para archivos `.tar` comprimidos. Esto tiene muchas ventajas; son fáciles de construir en casi cualquier sistema UNIX (muchos otros formatos de paquetes requieren cadenas de herramientas especiales), y son tan simples de desarmar.

Sin embargo, también es importante darse cuenta de que solo porque todos los paquetes Slackware **son** archivos .t?z, no todos los archivos .t?z son paquetes de Slackware. **installpkg** no instalará mágicamente cualquier archivo .t?z, solo paquetes de Slackware.

slackpkg

Slackpkg es una herramienta automatizada para la gestión de paquetes de Slackware Linux. Originalmente apareció en /extra para el lanzamiento de slackware-12.1, y desde el lanzamiento de slackware-12.2 se ha incluido en la ap/series de una instalación base.

Al igual que puede usar **installpkg** para instalar paquetes de Slackware desde el directorio /extra incluido en los medios de instalación, puede usar **slackpkg** para extraer paquetes de Internet e instalarlos. Esto es particularmente útil para actualizaciones de seguridad o actualizaciones importantes de aplicaciones que se publican en los servidores Slackware, algunas de las cuales es posible que desee comenzar a usar en su propio sistema.

Sin **slackpkg**, el proceso sería:

1. Observe en el registro de cambios de Slackware que se ha publicado una actualización.
2. Busque en su mirror local de Slackware para encontrar un enlace de descarga del paquete.
3. Descargue el paquete desde un mirror Slackware a su disco duro.
4. Use **installpkg** o **pkgtool** para instalar el paquete descargado.

Con **slackpkg**, esto se reduce a:

1. Observe en el registro de cambios de Slackware que se ha lanzado una actualización para **foo**.
2. **slackpkg** instalar foo.

Claramente, esto simplifica una tarea bastante común.

Para usar **slackpkg**, configure su sistema con un mirror de Slackware editando /etc/slackpkg/mirrors como root. Encuentre el mirror asociado con su versión y arquitectura de Slackware, y descomprímalo. Esta lista de réplicas ofrece acceso ftp y http, pero debe descomentar **solo una** mirror.

Una vez que se haya seleccionado un mirror, actualice la lista de archivos remotos emitiendo el comando inicial slackpkg update. Esto se debe hacer cada vez que note que se ha publicado un nuevo paquete (se recomienda registrarse regularmente con el registro de cambios de Slackware; consulte e [Capítulo 18, Seguimiento de las actualizaciones](#) para obtener más información).

Para buscar un paquete, usar slackpkg search fooe instalar uso slackpkg install foo.

Una vez que se ha instalado un paquete con **slackpkg**, se puede eliminar o actualizar utilizando **pkgtool** y los otros comandos de administración de paquetes tal como se detalla en [Instalar, desinstalar y actualizar paquetes](#).

Para obtener más información, consulte las páginas **man** para slackpkg (8) y slackpkg.conf (5), y consulte su sitio web en <http://www.slackpkg.org/>

rpm2tgz

Uno de los formatos de paquete más omnipresentes para el software de Linux es RPM; no es raro encontrar un desarrollador que ofrezca su aplicación para descargar como código fuente o un archivo RPM, y nada más. En este caso, tendrías tres opciones:

1. Construye tu propio paquete Slackware.
2. Compilar e instalar directamente desde el código fuente.
3. Convertir e instalar desde RPM.

La creación a partir del código fuente o la creación de su propio paquete Slackware no suele ser tan complejo como podría pensar, pero en general no se recomienda instalar directamente desde el código fuente porque no hay una manera fácil de rastrear lo que se ha instalado en su sistema después de emitir el comando `make install`. La creación de sus propios paquetes de Slackware está fuera del alcance de este capítulo. Así que esto nos deja con la útil herramienta **rpm2tgz**.

rpm2tgz convierte los paquetes RPM en un paquete Slackware que luego puede instalarse a través de **pkgtool** o **installpkg**. Esto evita la necesidad de crear su propio paquete Slackware, pero le otorga la ventaja de poder eliminar, actualizar y rastrear lo que ha instalado.



Si bien un paquete Slackware es solo un script de shell y un código fuente, en comparación, un paquete RPM puede ser un laberinto de listas de dependencias e instrucciones especiales. Por lo tanto, **rpm2tgz** no siempre funcionará, especialmente en aplicaciones muy complejas, y nunca resolverá mágicamente las dependencias.

Para probar **rpm2tgz**, descargue un archivo RPM de una fuente confiable y **conviértalo**

```
rpm2tgz foo-xxxx.rpm
```

El resultado es un archivo .tgz, por lo que una vez finalizada la conversión, el RPM original se puede descartar de forma segura. Use **installpkg** para instalar el paquete Slackware que acaba de crear, siempre que haya instalado todo el código de dependencia para que la aplicación funcione.

Navegación de capítulos

Capítulo anterior: [Utilidades básicas de red](#)

Siguiente capítulo: [Mantener un seguimiento de las actualizaciones](#)

Fuentes

- Fuente original: <http://www.slackbook.org/beta>
- Originalmente escrito por Alan Hicks, Chris Lumens, David Cantrell, Logan Johnson

— *M3rsy* 2019/02/02 00:54 (UTC)

From:

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

Permanent link:

https://docs.slackware.com/es:slackbook:package_management

Last update: **2019/03/03 18:21 (UTC)**

