

Actualización de Slackware a una nueva versión

Actualizar o instalar desde cero

Instalar Slackware desde cero y desde cero es siempre el mejor método si está ejecutando una versión bastante antigua de Slackware y desea omitir algunas. Se habrán producido demasiados cambios intrusivos en la distribución si su Slackware es relativamente antiguo. Hará que el proceso de actualización manual sea doloroso y no garantizará el éxito. En tales casos, es mejor hacer una copia de seguridad de su lista de paquetes

En tales casos, es mejor hacer una copia de seguridad de su lista de paquetes (`ls -lart /var/log/packages`), una copia de seguridad de su `/etc` directorio " " y (usted ya hizo esto, por supuesto) una copia de seguridad de sus datos personales. Formatee su disco e instale desde el medio de arranque Slackware, y una hora más tarde ya podría estar en el negocio.

Si desea actualizar a la próxima versión de Slackware, puede hacerlo manualmente siguiendo las instrucciones en el archivo `UPGRADE.TXT` ue encontrará en la raíz del DVD / CD1 de Slackware. Puede encontrar instrucciones avanzadas y más sugerencias en `CHANGES_AND_HINTS.TXT` en la misma ubicación. También hay un procedimiento semiautomático para este tipo de actualización, usando [slackpkg](#). Te quitará mucho trabajo duro.

Siempre utilizo `slackpkg` (con cuidado) para actualizar mis sistemas Slackware de una versión estable a la siguiente. Llamo a este proceso una *"actualización de sistema"*. Puede usar el mismo procedimiento para migrar a `slackware-current`, mantener actualizado un sistema de `slackware-current` o actualizar de un `slackware` no tan actual a una versión estable recién lanzada.

Consideraciones sobre el Kernel

Solo ejecutar `slackpkg` y esperar lo mejor no va a funcionar. Hay que tener en cuenta algunas consideraciones. Una cosa importante para recordar:

Nunca actualices tu núcleo de trabajo.

¿Porqué es eso? Simple: estará actualizando potencialmente cientos de paquetes y debe estar preparado para el improbable caso de que su computadora no funcione correctamente después de una actualización del sistema. Una cosa por la que no quiere que lo golpeen es un sistema que no arranca en absoluto. Una nueva versión de Slackware puede instalar un kernel que se niega a iniciar su computadora (pocas posibilidades, pero sin embargo ... prepárese). Por esa razón, necesita mantener su núcleo de trabajo "antiguo" instalado y mantener una sección para él en su `/etc/lilo.conf`. De esa manera, si el nuevo kernel no se inicia, puede retroceder al kernel anterior y comenzar a investigar qué fue lo que falló.

Básicamente, estas son las mismas precauciones que debe tomar cuando [compile un nuevo kernel](#)

Consideraciones del controlador de video

Si su computadora está equipada con una tarjeta de video alimentada por un procesador de gráficos [Nvidia](#) o [Ati](#) y ha instalado los controladores de gráficos acelerados de estas compañías (de código cerrado y solo binarios), no debe intentar iniciar una sesión X después de actualizar a la siguiente Lanzamiento de Slackware.

Estos controladores dependen de la versión del kernel, la versión de Mesa y el servidor X.Org. Debe volver a instalar el controlador binario antes de iniciar el modo gráfico. Además, los paquetes mesa y xorg-server de Slackware sobrescriben los archivos esenciales de estos controladores de gráficos acelerados de código cerrado de todos modos.

Si desea saber cómo tratar con estos controladores binarios, tenemos instrucciones más detalladas en el artículo "[Controladores de graficos propietario](#)" en este wiki.

Slackpkg consideraciones

Si actualiza Slackware (consulte a continuación el procedimiento), actualizará [slackpkg](#) as one of the first steps. The `upgradepkg` como uno de los primeros pasos. El `upgradepkg` comando instalará un archivo `/etc/slackpkg/mirrors.new`. Este es el archivo que contiene las URL para los espejos que llevan la nueva versión de Slackware. Tendrá que comparar esto con la versión original y fusionar los contenidos.

Asegúrese de tener exactamente una línea sin comentarios, lo que apunta a un espejo para la versión y arquitectura de Slackware deseadas.

Actualización del sistema utilizando SlackPkg

Los siguientes pasos deberían funcionar para todas las situaciones:

- [Lista Negra](#) los siguientes paquetes de kernel en `"/etc/slackpkg/blacklist"`:

```
kernel-generic
kernel-generic-smp
kernel-huge
kernel-huge-smp
kernel-modules
kernel-modules-smp
```

Esto evitará una actualización accidental de su núcleo de trabajo.

- Lista negra de paquetes de repositorios de terceros agregando líneas apropiadas para sus *etiquetas de repositorio*. Ejemplos para SlackBuilds.org, AlienBOB y multilib:

```
[0-9]+_SBo
[0-9]+alien
```

```
[0-9]+compat32
```

- Si se han agregado nuevos kernel (s) a la versión de Slackware a la que está actualizando, entonces use “installpkg” para instalar esos nuevos paquetes de kernel primero (no use “upgradepkg” porque borrará su kernel existente).

Deberá instalar al menos un kernel (kernel-generic, kernel-generic-smp, kernel-enorme o kernel-huge-smp) y el paquete de módulos del kernel correspondiente (kernel-modules o kernel-modules-smp).

No se puede utilizar slackpkg para este paso.

- Ahora que tenemos los nuevos kernel (s) más módulos en su lugar, podemos comenzar a actualizar el resto de los paquetes. Primero, actualice la slackpkg base de datos del paquete:

```
# slackpkg update
```

- Cuando slackpkg ha actualizado su base de datos interna, lo primero que debe hacer es actualizarse slackpkg la última versión (incluidas las URL para la nueva versión de Slackware y cualquier regla de actualización de paquetes que se aplique a la nueva versión). Si no realiza este paso, slackpkg se actualizará en la mitad de la *actualización del sistema*, y se cancelará inmediatamente después de eso ...

```
# slackpkg upgrade slackpkg  
# slackpkg new-config
```

Ese new-config comando final está allí para que pueda ver la diferencia entre sus slackpkg archivos de configuración antiguos y nuevos , en particular, /etc/slackpkg/mirrors y /etc/slackpkg/blacklist archivos que debe verificar. /etc/slackpkg/slackpkg.conf Generalmente se recomienda sobrescribir .

- A Una nueva versión de Slackware generalmente tiene una versión más nueva de las bibliotecas de GNU C. Los nuevos paquetes están compilados contra esa nueva glibc version. Para evitar un error de actualización, debe actualizar el glibc-solibs paquete manualmente, inmediatamente después de la actualización slackpkg:

```
# slackpkg upgrade glibc-solibs
```

Déjeme dar un ejemplo de tal falla potencial: cuando slackpkg install-new instala libusb-compat paquete, sugpg comando deja de funcionar porque se vincula con libusb.so, que se sobrescribirá con la versión del nuevo libusb-compat paquete. La nueva biblioteca necesita el nuevo glibc paquete gpg deja de funcionar debido al error de enlace de la biblioteca y slackpkg a actualización del sistema porque quiere verificar la firma de gpg de cada paquete antes de actualizarlo. La actualización de glibc-solibs paquete evita que la biblioteca vincule los errores al proporcionar los “GLIBC” correctos.

- Vamos a slackpkg actualizar la computadora a la nueva versión de Slackware:

```
# slackpkg install-new  
# slackpkg upgrade-all
```

slackpkg clean-system

- El primero de esos tres comandos (`slackpkg install-new`) instalará todos los paquetes que están marcados en el archivo `ChangeLog.txt` de Slackware con la cadena "Added." Este comando **no** instalará ningún otro paquete que aún no esté instalado. Por ejemplo, si no tenía KDE instalado anteriormente, el "`slackpkg install-new`" comando " " no agregará paquetes KDE a su computadora de repente.
 - El segundo comando (`slackpkg upgrade-all`) comparará cada paquete oficial de Slackware que tiene instalado actualmente, con la lista de paquetes en su espejo Slackware. Si hay una versión diferente disponible, esa versión se descargará y actualizará. 1)
 - El tercer comando (`slackpkg clean-system`) e mostrará una descripción general de todos los paquetes que tiene instalados actualmente y no forman parte de la versión de Slackware Linux a la que está actualizando. Esto significa que la lista mostrará todos los paquetes que se han eliminado de Slackware. Ejemplos para Slackware 14 son `kdeaccessibility`, `kdebase`, ... Otro ejemplo de tal paquete abandonado en Slackware 14 es `ntfsprogs`. En realidad, este paquete no se *eliminar* pero se *cambio de nombre...* para Slackware que es igual a la *eliminacion*.
¡El comando también le mostrará todos los paquetes de terceros que haya instalado! Use este comando sabiamente: debe de-seleccionar cada paquete que desea conservar (es decir, todos los paquetes de terceros) y luego hacer clic en "Aceptar" para permitir que `slackpkg` elimine todos los paquetes obsoletos.
- Una actualización masiva como esta habrá instalado varios ".new" archivos. Algunos paquetes contienen archivos de configuración que han sido renombrados (dentro del paquete) con una ".new" extensión " " para que un archivo de configuración existente (que contenga sus personalizaciones) no se sobrescriba imprudentemente durante la actualización. `slackpkg` comprobará la existencia de estos ".new" archivos al final de una actualización o instalación y le pedirá que haga algo con ellos.
Se recomienda actualizar a las nuevas versiones de los archivos de configuración cuando sea posible, ya que a menudo traerán mejoras a su configuración de software. `slackpkg` le permite ver las diferencias entre los archivos antiguos y nuevos e incluso, para combinar los dos archivos. Alternativamente, usted puede decidir mantener el ".new" Dejando el archivo antiguo en su lugar, para que pueda investigar las diferencias más adelante si son demasiado intrusivas.
Puede forzar una verificación de los ".new" farchivos " " en cualquier momento, ejecutando el comando

* # slackpkg new-config

y utilizando la interfaz de usuario fácil de `slackpkg` para combinar los cambios.

- Probablemente debería decidir en este momento usar un kernel genérico, especialmente si está usando LVM o RAID, o si instala Slackware en un disco encriptado con LUKS. Esto también se recomienda en Slackware README en el DVD / CD. Por otro lado, si la configuración de su sistema es sencilla y su hardware es bastante nuevo, podría decidir quedarse con un kernel "enorme".
*Recuerde que no se **puede** usar un disco ram inicial en combinación con un kernel "huge" kernel, ¡pero **tiene** que crear un nuevo ramdisk inicial si va a usar un kernel genérico!*
Si no está seguro en este momento de cómo se debe lograr esto, entonces debe asegurarse de que está iniciando un núcleo "huge" que no requerirá un disco de ram inicial.

Sin embargo, el “mkinitrd_command_generator.sh” El script puede ayudarte aquí. Ejecute este script con la *nueva versión* del kernel como parámetro y le mostrará un ejemplo del “mkinitrd” comando “ ” que funcionará para su configuración de hardware particular y la configuración del sistema:

```
# /usr/share/mkinitrd/mkinitrd_command_generator.sh -k 3.2.29
```

dará esto como salida (la versión del núcleo 3.2.29 es la de Slackware 14)

```
#
# mkinitrd_command_generator.sh revision 1.45
#
# This script will now make a recommendation about the command to use
# in case you require an initrd image to boot a kernel that does not
# have support for your storage or root filesystem built in
# (such as the Slackware 'generic' kernels').
# A suitable 'mkinitrd' command will be:

mkinitrd -c -k 3.2.29 -f ext4 -r /dev/sdb2 -m usb-
storage:pcmcia_core:pcmcia:mmc_core:ssb:modprobe:usbhid:ehci-hcd:ohci-
hcd:mbcache:jbd2:ext4 -u -o /boot/initrd.gz
```

Puede copiar y pegar esta línea de comando en su consola y dejar que cree un disco RAM inicial para usted.

Si ya estaba ejecutando un kernel genérico y, por lo tanto, ya tiene un initrd, le recomendamos encarecidamente que cree un **nuevo** initrd con un **nuevo** nombre único. Por ejemplo, puede copiar el ejemplo anterior y modificar el nombre del archivo initrd de la siguiente manera:

```
mkinitrd -c -k 3.2.29 -f ext4 -r /dev/sdb2 -m usb-
storage:pcmcia_core:pcmcia:mmc_core:ssb:modprobe:usbhid:ehci-hcd:ohci-
hcd:mbcache:jbd2:ext4 -u -o /boot/initrd_3.2.29.gz
```

- A Una vez que haya decidido qué kernel va a usar y ha creado un disco RAM inicial, debe actualizar su “/etc/lilo.conf” archivo “ ” con una sección para el nuevo kernel (*¡no elimine su kernel en ejecución!*). TLa “mkinitrd_command_generator.sh” secuencia de comandos “ ” puede ayudarlo a encontrar el bloque correcto al que anexar /etc/lilo.conf. Por ejemplo, el comando:

```
# /usr/share/mkinitrd/mkinitrd_command_generator.sh -l /boot/vmlinuz-
generic-3.2.29
```

resultará en la siguiente salida que puede copiar / pegar:

```
# Linux bootable partition config begins
# initrd created with 'mkinitrd -c -k 3.2.29 -f ext4 -r /dev/sdb2 -m
usb-storage:pcmcia_core:pcmcia:mmc_core:ssb:modprobe:usbhid:ehci-
hcd:ohci-hcd:mbcache:jbd2:ext4 -u -o /boot/initrd.gz'
image = /boot/vmlinuz-generic-3.2.29
  initrd = /boot/initrd.gz
```

```
root = /dev/sdb2
label = 3.2.29
read-only
# Linux bootable partition config ends
```

Tenga en cuenta que este comando agrega una línea “initrd” a la sección del kernel. Si permite mkinitrd crear un nombre único para su ramdisk inicial, asegúrese de aplicar ese nombre en la sección anterior.

La “initrd” no es necesaria si va a ejecutar un kernel “huge” .

- Finalmente, ejecute el comando “lilo” para que el cambio sea permanente y agregue el nuevo kernel al menú de arranque de lilo. Simplemente puede ejecutar “eliloconfig” si está utilizando EFI después de actualizar Slackware y eso instalará el kernel más reciente en la partición EFI.
Recuerde, siempre debe poder reiniciar en un kernel anterior en caso de que el nuevo kernel de Slackware le resulte difícil.

Línea inferior: puede confiar en que slackpkg realizará una actualización del sistema de manera segura, pero necesitará su inteligencia y cuidado.

Consideraciones multilib

Si está actualizando una computadora Slackware multilib de 64 bits, hay otras consideraciones que hacer.

Una instalación multilib significa que haya sustituido de 64 bits de Slackware gcc y glibc paquetes con versiones multilib (ie.e. apoyo binarios de 32 bits y 64 bits). Además, ha instalado un conjunto de paquetes Slackware de 32 bits “convertidos” en su Slackware multilib de 64 bits. Todas estas modificaciones son necesarias para permitirle ejecutar y compilar software de 32 bits.

Por supuesto, al actualizar un sistema de este tipo, debe actualizar los paquetes estándar de Slackware, pero por separado debe actualizar los paquetes específicos de multilib con nuevas versiones que puede obtener en <http://slackware.com/~alien/multilib/>

- Primero (si usa slackpkg), debe incluir en la lista negra todos estos paquetes multilib para que no se puedan reemplazar o eliminar accidentalmente durante una actualización del sistema. Si no los pone en la lista negra, se enfrentará a muchas desecciones manuales en “ ”slackpkg clean-system“.
A partir del lanzamiento de [compat32-tools](#) Slackware 14.0, simplemente puede agregar dos líneas al /etc/slackpkg/blacklist archivo:

```
[0-9]+alien
[0-9]+compat32
```

Luego, tendrá que descargar y actualizar manualmente los paquetes multilib. En el siguiente ejemplo, usaré Slackware 14.0 como la versión a la que se actualizará.

- Descargue los paquetes multilib adecuados para su nueva versión de Slackware desde un espejo, como este:

```
# rsync -av
rsync://taper.alienbase.nl/mirrors/people/alien/multilib/14.0/
multilib-14.0/
```

Este comando creará un nuevo subdirectorio " multilib-14.0 " en su directorio actual con todos los paquetes dentro

- Instalar / actualizar los paquetes existentes `gcc` y `glibc` , y `compat32-tools` :

```
# cd multilib-14.0
# upgradepkg --install-new *.t?z
```

- Actualice el conjunto de paquetes Slackware convertidos de 32 bits (a menudo denominados paquetes "compat32"):

```
# upgradepkg --install-new slackware64-compat32/*-compat32/*.t?z
```

Alternativamente, puede ejecutar el script " `massconvert32.sh` " que se habrá instalado como parte del paquete `compat32-tools` . Páselo como un directorio de paquetes de Slackware de 32 bits (o una URL espejo de Slackware de 32 bits) como parámetro y eso creará un conjunto de paquetes convertidos " `compat32` " que luego puede instalar. Solo tendría que hacer esto si sospecha que el contenido del directorio " `slackware64-compat32` " no está actualizado.

Consideraciones de Java

Slackware solía instalar *Java Run-time Engine* antes de la versión 14.0 (los archivos binarios de JRE fueron originalmente de Sun y luego distribuidos por Oracle después de comprar Sun). Pero Oracle cambió la licencia de redistribución para que Slackware (al igual que todas las demás distribuciones) ya no pudiera enviar estos binarios Java como parte de la distribución. Cuando realiza una *actualización del sistema* a Slackware 14.0, una versión anterior del *JRE* se quedará atrás en su sistema. Esta versión "6u25" tiene varias vulnerabilidades críticas y debe eliminarla manualmente de su computadora lo más rápido posible, utilizando el comando

```
removepkg jre
```

Si necesita Java, eche un vistazo al directorio " `/extra/source/java` " de la versión Slackware 14. Encontrará un script allí para crear un paquete Slackware a partir del software Java más reciente de Oracle, que luego puede instalar utilizando el comando " `installpkg` ". Consulte también nuestro artículo Wiki " [Java en Slackware](#) "

Fuentes

- Originalmente escrito por [Eric Hameleers](#)

[howtos](#), [slackpkg](#), [author alienbob](#)

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

Permanent link:
https://docs.slackware.com/es:howtos:slackware_admin:systemupgrade

Last update: **2019/08/21 00:58 (UTC)**

