Agregar capacidad multilib a Slackware en la arquitectura x86_64

Este artículo contiene instrucciones sobre cómo crear un *true multilib* Slackware64. Un sistema Linux multilib de 64 bits es capaz de ejecutar software de 64 bits y de 32 bits. El estándar de jerarquía del sistema de archivos documenta el método óptimo para lograr una separación limpia entre el software de 64 bits y 32 bits en un solo sistema. Al comenzar con el desarrollo de "Slackware64" (el puerto oficial de la arquitectura x86_64), optamos por adoptar este estándar. Por lo tanto, Slackware64 se ha configurado para buscar bibliotecas de 64 bits en los directorios `/lib64` y `/usr/lib64`. Esta es la razón por la que llamo a Slackware64 "multilib-ready", aunque las bibliotecas de 32 bits se buscarán en "/lib" y "/usr/lib", Slackware64 no se entrega con ningún software de 32 bits. Hay un paso más que se debe tomar (usted, el usuario) antes de que Slackware64 pueda llamarse "multilib-enabled".

Esto se consigue de la siguiente manera:

1. Primero necesitamos cambiar a versiones multilib de
   - *glibc* (es decir, un glibc que admite *ejecutar* binarios de 32 y 64 bits), y
   - *gcc* (es decir, capaz de *compilar* binarios de 32 bits, así como binarios de 64 bits).
2. Luego, las bibliotecas del sistema se toman de Slackware de 32bit y se instalan en el sistema Slackware de 64bit junto con sus versiones de 64bit, lo que completa el proceso de creación de una capa de software de compatibilidad de 32bit.

Cuando se lanzó Slackware64, tenía una ventaja sobre las "bifurcaciones" de 64 bits que existían entonces. Estas bifurcaciones agregaban la capa de compatibilidad de 32 bits al compilar muchos de sus paquetes como binarios de 32 bits. Slackware, por otro lado, es una distribución que consiste en una versión de 32 bits y 64 bits, ambas de las cuales se están desarrollando en paralelo. Esto significa que no tiene que compilar paquetes de 32 bits desde cero para agregar capacidad multilib al sistema de 64 bits. ¡Simplemente, tómelos del árbol de paquetes de Slackware de 32 bits! Esta fue una de las razones para no agregar multilib completo a Slackware64: creamos las condiciones previas correctas pero requerimos que el usuario actúe si necesita multilib. En una sección más abajo, explicaré cómo puede tomar un paquete Slackware de 32 bits (por ejemplo, el paquete "mesa") y volver a empaquetar su contenido en un paquete "mesa-compat32" que puede instalar directamente en Slackware64 .

Slackware para la arquitectura x86_64 (o "*Slackware64*" para abreviar) es un sistema operativo puro de 64 bits, pero fácilmente actualizable a multilib. *Recién estrenado, Slackware64 solo es capaz de compilar y ejecutar binarios de 64 bits.*

Ventaja de un sistema multilib

Daré algunos ejemplos de programas que requieren soporte multilib en un Slackware de 64 bits porque no se iniciarán ni compilarán en Slackware64 sin la capa de compatibilidad de 32 bits:

- Wine
  La mayoría de los programas de Windows siguen siendo de 32 bits, y para ejecutarlos en Linux con Wine, necesitas una versión de 32 bits de Wine.
- VirtualBox
  El popular software de máquina virtual. Aunque este es (en parte) de código abierto, todavía necesita bibliotecas de compatibilidad de 32 bits en Slackware de 64 bits.

- Steam
  La popular plataforma de juegos aún necesita un Cliente de 32bit. La mayoría de los juegos disponibles son de 32 bits también.
- Skype, Cliente Citrix, ...
  Estos programas son propietarios y de código cerrado. Tenemos que depender del desarrollador para que los binarios de 64 bits estén disponibles. Hasta ahora, eso no ha sucedido con estos programas de ejemplo.

Afortunadamente, el soporte de 64 bits es cada vez más común. Adobe fue un punto delicado durante mucho tiempo, pero finalmente lanzaron su complemento de navegador Flash en una versión de 64 bits. Sun (ahora absorbido por Oracle) reveló una versión de 64 bits de su complemento de navegador Java. Estos dos eventos fueron los principales desencadenantes para comenzar a trabajar en Slackware64.

Obtención de paquetes multilib

Puede descargar un conjunto de paquetes y scripts habilitados para multilib desde mi sitio web: http://slackware.com/~alien/multilib/.

Además de varios archivos README (este artículo de Wiki es básicamente una versión mejorada de uno de estos README), encontrará un subdirectorio por cada versión de Slackware de 64 bits debajo del directorio de nivel superior "*multilib*". Hay otro directorio llamado "source". El directorio "*source*" contiene las fuentes de los paquetes y los scripts de SlackBuild. Las cosas que realmente le interesan, los paquetes binarios, están disponibles en el directorio *<slackware_release_number>* debajo del directorio de nivel superior. Cada directorio también contiene un subdirectorio "*slackware64-compat32*" donde encontrará un conjunto esencial de paquetes Slackware convertidos de 32 bits, listos para instalar en su Slackware de 64 bits.

Mantener actualizado su multilib

Para mantenerse al día, le aconsejo que vigile el Registro de cambios (RSS feed) que mantengo para mis paquetes multilib. Por lo general, tendré *paquetes actualizados de glibc y gcc* disponibles dentro del día después de que Slackware tenga actualizaciones de gcc y glibc.

Automatización:

1. Eche un vistazo a compat32pkg por Sèbastien Ballet que automatiza este proceso, similar a slackpkg.
2. Si prefiere slackpkg para la administración de paquetes, entonces vale la pena revisar slackpkg+, una extensión de slackpkg que administra los paquetes que instaló desde repositorios de terceros. incluyendo multilib. Cuando se configura correctamente, mantener su multilib es tan fácil como ejecutar:

```
# slackpkg update
# slackpkg upgrade multilib
# slackpkg install multilib
```

El último comando le mostrará si se agregaron paquetes nuevos a la colección de paquetes

“compat32”, como llvm-compat32 y orc-compat32 recientemente.

- Así es como se vería una configuración típica: para una computadora que ejecuta Slackware-current y usa el repositorio de pruebas KDE de Alien BOB. *PKGS_PRIORITY* garantiza que los paquetes multilib de gcc y glibc tengan prioridad sobre los originales de Slackware. La palabra clave “multilib” que define el nombre del repositorio debe ser la misma palabra clave utilizada en los comandos “slackpkg” anteriores. La elección de la palabra “multilib” es arbitraria, bien podría haber sido “compat32”, siempre y cuando se use de manera consistente.
  El contenido de un archivo de ejemplo “/etc/slackpkg/slackpkgplus.conf” sería el siguiente:

```
SLACKPKGPLUS=on
VERBOSE=1
ALLOW32BIT=off
USEBL=1
WGETOPTS="--timeout=5 --tries=1"
GREYLIST=on
PKGS_PRIORITY=( multilib restricted alienbob ktown )
REPOPLUS=( slackpkgplus multilib restricted alienbob ktown )
MIRRORPLUS['multilib']=http://bear.alienbase.nl/mirrors/people/alien/multili
b/current/
MIRRORPLUS['alienbob']=http://bear.alienbase.nl/mirrors/people/alien/sbrepos
/current/x86_64/
MIRRORPLUS['restricted']=http://bear.alienbase.nl/mirrors/people/alien/restr
icted_sbrepos/current/x86_64/
MIRRORPLUS['ktown']=http://bear.alienbase.nl/mirrors/alien-kde/current/lates
t/x86_64/
MIRRORPLUS['slackpkgplus']=http://slakfinder.org/slackpkg+/
```

Habilitar el soporte multilib en Slackware64

Las instrucciones rápido y fácil

Esta sección contiene las instrucciones esenciales para agregar capacidad multilib completa a su sistema Slackware64. Si desea comprender el proceso con más detalle, o necesita información sobre cómo compilar el software de 32 bits en Slackware64, también debe leer las secciones que siguen. Tenga en cuenta que el “#” en frente de los comandos muestra un *intérprete root*.

- Descargue los paquetes de mi sitio web (le di la URL en la sección anterior, pero este ejemplo está usando una URL espejo). Supongamos que está ejecutando Slackware 14.2. Usted ejecute

```
# SLACKVER=14.2
# mkdir multilib
# cd multilib
# lftp -c "open http://bear.alienbase.nl/mirrors/people/alien/multilib/ ;
mirror -c -e ${SLACKVER}"
# cd ${SLACKVER}
```

- Actualice sus paquetes de 64bit Slackware “*gcc*” y “*glibc*” a mis versiones multilib. \\Ejecute el comando

```
# upgradepkg --reinstall --install-new *.t?z
```

después de cambiar al directorio donde descargó estos paquetes.\\Este comando además instalará un paquete adicional llamado "*compat32-tools*".

- Si también descargó un directorio llamado *slackware64-compat32* (mi comando de ejemplo "`lftp`" lo habrá hecho), entonces tiene suerte, ¡porque ya hice la conversión del paquete de 32 bits! Todo lo que hace es ejecutar el comando:

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

que instalará todos los paquetes Slackware de 32 bits convertidos (o los actualizará si ya ha instalado paquetes multilib más antiguos, por ejemplo, cuando está actualizando a un Slackware más nuevo). ¡Eso es todo!

- Si no puede encontrar un subdirectorio llamado *slackware64-compat32* entonces no lo descargó o el espejo de descarga no lo proporcionó. En este caso, tiene que hacer usted mismo la conversión del paquete de 32 bits. No es para nada difícil, lleva unos minutos:
- Lo más rápido es si tiene un directorio local con paquetes Slackware originales de 32 bits disponibles (también llamado *local mirror*). Aquellos que compraron un DVD oficial de Slackware pueden simplemente usar ese DVD: es de doble cara y Slackware de 32 bits está en uno de los lados. En el caso de este ejemplo Supondré que tiene un árbol de directorios Slackware de 32 bits disponible en "`/home/ftp/pub/slackware/slackware-14.2/slackware/ `".
  Debe haber subdirectorios llamados 'a', 'ap', 'd', 'l', 'n', 'x' inmediatamente debajo de este directorio. (Si ha montado un DVD de Slackware, su directorio probablemente será "`/media/SlackDVD/slackware /`" pero no lo usaré en los comandos de ejemplo a continuación).
- Crear un nuevo directorio vacío (llamémoslo 'slackware64-compat32') y cambiarlo:

```
# mkdir slackware64-compat32 ; cd slackware64-compat32
```

- Ejecute el siguiente comando para crear un conjunto de paquetes de compatibilidad de 32 bits, usando el directorio para los paquetes oficiales de 32bit Slackware como entrada:

```
# massconvert32.sh -i /home/ftp/pub/slackware/slackware-14.2/slackware/
```

- El paso anterior lleva un tiempo. Cuando termine, proceda a instalar los 90 MB de paquetes Slackware de 32 bits recién convertidos que se crearon en subdirectorios debajo de su *directorio actual*:

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

- ¡Hecho! Ahora puede comenzar a descargar, instalar y ejecutar programas de 32 bits. Esto no fue tan difícil, ¿verdad?

Si usa un administrador de paquetes como *slackpkg* tendrá que agregar todos los nombres de los paquetes `glibc` y `gcc` a su lista negra de paquetes. Si no toma esta precaución, corre el riesgo de que su administrador de paquetes reemplace accidentalmente sus versiones multilib con las versiones puras originales de 64 bits de Slackware.
Si ejecuta Slackware 13.37 o una versión más reciente, slackpkg admite expresiones regulares en el archivo de lista negra. En ese caso, una sola línea en `/etc/slackpkg/blacklist` será suficiente para incluir en la lista negra todos mis paquetes (incluidos los paquetes multilib gcc y glibc y todos los

paquetes `compat32`):

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

Por otro lado, si está utilizando la extensión slackpkg llamada slackpkg+, definitivamente debería **no** incluir estos paquetes en la lista negra, porque eso evita que slackpkg+ los administre!

Si está ejecutando Slackware 13.1 o más reciente y descargó el paquete compat32-tools para esa versión, el script *massconvert32.sh* puede usar un servidor web remoto para descargar los paquetes de Slackware de 32 bits, en lugar de requerir un espejo Slackware local o un DVD. Utilice el parámetro "-u" para especificar la URL remota de esta manera:

```
# massconvert32.sh -u http://someserver.org/path/to/slackware-14.2/slackware
```

Instrucciones detalladas

Actualizando glibc y gcc

Los siguientes paquetes glibc/gcc son reemplazos para - *no adiciones a*- los paquetes estándar de Slackware. Use el programa "`upgradepkg`" para actualizar a mis versiones multilib de gcc y glibc. Los necesitará para ejecutar (glibc) y compilar (gcc), software de 32 bits en su computadora Slackware de 64 bits:

Slackware64 13.0

- La suite del compilador gcc:
  - gcc-4.3.3_multilib-x86_64-4alien.txz
  - gcc-g++-4.3.3_multilib-x86_64-4alien.txz
  - gcc-gfortran-4.3.3_multilib-x86_64-4alien.txz
  - gcc-gnat-4.3.3_multilib-x86_64-4alien.txz
  - gcc-java-4.3.3_multilib-x86_64-4alien.txz
  - gcc-objc-4.3.3_multilib-x86_64-4alien.txz
- Las librerías glibc de GNU:
  - glibc-2.9_multilib-x86_64-3alien.txz
  - glibc-i18n-2.9_multilib-x86_64-3alien.txz
  - glibc-profile-2.9_multilib-x86_64-3alien.txz
  - glibc-solibs-2.9_multilib-x86_64-3alien.txz
  - glibc-zoneinfo-2.9_multilib-noarch-3alien.txz

Slackware64 13.1

- La suite del compilador gcc:
  - gcc-4.4.4_multilib-x86_64-1alien.txz
  - gcc-g++-4.4.4_multilib-x86_64-1alien.txz
  - gcc-gfortran-4.4.4_multilib-x86_64-1alien.txz
  - gcc-gnat-4.4.4_multilib-x86_64-1alien.txz

    - gcc-java-4.4.4_multilib-x86_64-1alien.txz
    - gcc-objc-4.4.4_multilib-x86_64-1alien.txz
- Las librerías glibc de GNU:
    - glibc-2.11.1_multilib-x86_64-3alien.txz
    - glibc-i18n-2.11.1_multilib-x86_64-3alien.txz
    - glibc-profile-2.11.1_multilib-x86_64-3alien.txz
    - glibc-solibs-2.11.1_multilib-x86_64-3alien.txz
    - glibc-zoneinfo-2.11.1_multilib-noarch-3alien.txz

Slackware64 13.37

- La suite del compilador gcc:
    - gcc-4.5.2_multilib-x86_64-2alien.txz
    - gcc-g++-4.5.2_multilib-x86_64-2alien.txz
    - gcc-gfortran-4.5.2_multilib-x86_64-2alien.txz
    - gcc-gnat-4.5.2_multilib-x86_64-2alien.txz
    - gcc-java-4.5.2_multilib-x86_64-2alien.txz
    - gcc-objc-4.5.2_multilib-x86_64-2alien.txz
- Las librerías glibc de GNU:
    - glibc-2.13_multilib-x86_64-7alien.txz
    - glibc-i18n-2.13_multilib-x86_64-7alien.txz
    - glibc-profile-2.13_multilib-x86_64-7alien.txz
    - glibc-solibs-2.13_multilib-x86_64-7alien.txz

Slackware64 14.0

- La suite del compilador gcc:
    - gcc-g++-4.7.1_multilib-x86_64-1alien.txz
    - gcc-gfortran-4.7.1_multilib-x86_64-1alien.txz
    - gcc-gnat-4.7.1_multilib-x86_64-1alien.txz
    - gcc-go-4.7.1_multilib-x86_64-1alien.txz
    - gcc-java-4.7.1_multilib-x86_64-1alien.txz
    - gcc-objc-4.7.1_multilib-x86_64-1alien.txz
- Las librerías glibc de GNU:
    - glibc-2.15_multilib-x86_64-9alien.txz
    - glibc-i18n-2.15_multilib-x86_64-9alien.txz
    - glibc-profile-2.15_multilib-x86_64-9alien.txz
    - glibc-solibs-2.15_multilib-x86_64-9alien.txz

Slackware64 14.1

- La suite del compilador de GNU:
    - gcc-4.8.2_multilib-x86_64-1alien.txz
    - gcc-g++-4.8.2_multilib-x86_64-1alien.txz
    - gcc-gfortran-4.8.2_multilib-x86_64-1alien.txz
    - gcc-gnat-4.8.2_multilib-x86_64-1alien.txz
    - gcc-go-4.8.2_multilib-x86_64-1alien.txz
    - gcc-java-4.8.2_multilib-x86_64-1alien.txz

    - gcc-objc-4.8.2_multilib-x86_64-1alien.txz
- Las librerías glibc de GNU:
    - glibc-2.17_multilib-x86_64-10alien.txz
    - glibc-i18n-2.17_multilib-x86_64-10alien.txz
    - glibc-profile-2.17_multilib-x86_64-10alien.txz
    - glibc-solibs-2.17_multilib-x86_64-10alien.txz

Slackware64 14.2

- La suite del compilador gcc:
    - gcc-5.3.0_multilib-x86_64-3alien.txz
    - gcc-g++-5.3.0_multilib-x86_64-3alien.txz
    - gcc-gfortran-5.3.0_multilib-x86_64-3alien.txz
    - gcc-gnat-5.3.0_multilib-x86_64-3alien.txz
    - gcc-go-5.3.0_multilib-x86_64-3alien.txz
    - gcc-java-5.3.0_multilib-x86_64-3alien.txz
    - gcc-objc-5.3.0_multilib-x86_64-3alien.txz
- Las librerías glibc de GNU:
    - glibc-2.23_multilib-x86_64-2alien.txz
    - glibc-i18n-2.23_multilib-x86_64-2alien.txz
    - glibc-profile-2.23_multilib-x86_64-2alien.txz
    - glibc-solibs-2.23_multilib-x86_64-2alien.txz

Slackware64 current

* Mientras no vea un directorio separado llamado "*current*" puede usar los archivos en el directorio para la versión estable más reciente.

- La suite del compilador gcc:
    - gcc-7.1.0_multilib-x86_64-2alien.txz
    - gcc-brig-7.1.0_multilib-x86_64-2alien.txz
    - gcc-g++-7.1.0_multilib-x86_64-2alien.txz
    - gcc-gfortran-7.1.0_multilib-x86_64-2alien.txz
    - gcc-gnat-7.1.0_multilib-x86_64-2alien.txz
    - gcc-go-7.1.0_multilib-x86_64-2alien.txz
    - gcc-objc-7.1.0_multilib-x86_64-2alien.txz
- Las librerías glibc de GNU:
    - glibc-2.25_multilib-x86_64-3alien.txz
    - glibc-i18n-2.25_multilib-x86_64-3alien.txz
    - glibc-profile-2.25_multilib-x86_64-3alien.txz
    - glibc-solibs-2.25_multilib-x86_64-3alien.txz

Desde la actualización a gcc 7, no hay más paquetes "gcc-java" porque su desarrollo ha cesado.
El paquete `glibc-zoneinfo` no forma parte de multilib, ya que no contiene código. Debe instalar el paquete `glibc-zoneinfo` de Slackware.

Todos los lanzamientos de Slackware

Hay un paquete adicional que necesita instalar usando el programa "installpkg". La versión actual

puede variar para cada versión de Slackware, pero el paquete se puede encontrar en el mismo directorio donde también se encuentran las versiones multilib de `gcc` y `glibc`:

- El “kit de herramientas de 32 bits” (scripts que facilitan la creación de paquetes de 32 bits)
  - compat32-tools-3.7-noarch-1alien.tgz

Añadiendo bibliotecas Slackware de 32 bits

La actualización de glibc y gcc que describí en la sección anterior cambia su sistema de “*multilib-ready*” a “*multilib habilitado*”. Ahora, todo lo que necesita hacer es instalar versiones de 32 bits del software del sistema de Slackware para que los futuros programas de 32 bits que vaya a instalar y/o compilar encuentren todas las bibliotecas de 32 bits que necesitan para funcionar.

Esto no es tan simple como agarrar paquetes de 32bit Slackware e instalarlos en Slackware64:

- En primer lugar, terminará con varios paquetes que llevan el mismo nombre (dos paquetes 'mesa', dos paquetes 'zlib', etc.) lo que le resultará confuso a usted, así como al administrador de paquetes *slackpkg*.
- Y además, si el paquete de 32 bits contiene binarios (algo como `/usr/bin/foo`), sobrescribirá su equivalente de 64 bits cuando instale el paquete de 32 bits sobre él. Si eso sucede, trastornará seriamente su sistema.

Se requiere un poco de cuidado adicional para eliminar archivos innecesarios/indeseables de los paquetes de 32 bits antes de instalarlos. Lo que necesita es un paquete de 32 bits que no entre en conflicto con lo que sea que ya esté presente en Slackware de 64 bits. De ahí el nombre “paquete de compatibilidad de 32 bits”.

Decidí que sería una pérdida de ancho de banda de descarga si yo mismo creara versiones de compatibilidad de 32 bits de los paquetes de Slackware. Después de todo, es probable que haya comprado el Slackware 14.2 DVD, por lo que ya posee las versiones de Slackware de 64 bits y de 32 bits … o bien el árbol de 32 bits de Slackware está disponible para descarga gratuita, por supuesto

En vez de eso, escribí algunos scripts (partes del código del script fueron escritas por Fred Emmott de Slamd64) y lo dispuse en un paquete “*compat32-tools*” . Su propósito es permitirle extraer el contenido de cualquier paquete Slackware de 32 bits y usarlo para crear un nuevo paquete que pueda instalar de manera segura en su Slackware de 64 bits.

Este paquete “*compat32-tools*” necesita una explicación.
Lea el archivo '*README*' detallado en el directorio `/usr/doc/compat32-tools-*/`, que le servirá de guía. Estos son los tres scripts útiles que el paquete instala:

- */etc/profile.d/32dev.sh* \\Este es el mismo script que viene con Slamd64. Reconfigura su entorno de shell para que le resulte más fácil compilar software de 32 bits (prefiriendo los compiladores y bibliotecas de 32 bits sobre sus versiones de 64 bits)

- *convertpkg-compat32* \\Este script toma un paquete Slackware de 32 bits y lo convierte en un paquete '-compat32' que puede instalar de forma segura (usando “installpkg”) en Slackware64, junto con la versión de 64 bits de El mismo paquete de software. Por ejemplo: suponga que necesita las bibliotecas de 32 bits que están en el paquete mesa. Tome el paquete mesa de Slackware de 32 bits (`x/mesa-7.5-i486-1.txz`) y luego ejecute

```
# convertpkg-compat32 -i /path/to/mesa-7.5-i486-1.txz
```

que creará un nuevo paquete llamado “mesa-compat32-7.5-x86_64-1compat32.txz”. Este nuevo paquete (que se crea en su directorio `/tmp` a menos que haya especificado otro destino) es básicamente el paquete antiguo de 32 bits, pero está exento de cosas no esenciales.
El nombre base modificado (*mesa* se convierte en *mesa-compat32*) le permite instalar este nuevo paquete en Slackware64, donde coexistirá con el paquete de 64 bits *mesa*, sin sobrescribir ningún archivo.
La secuencia de comandos deja los archivos temporales en el directorio “`/tmp/package-<prgnam>-compat32`” que puede eliminar de forma segura.

- *massconvert32.sh*
  Este script contiene una lista interna de lo que considero el subconjunto esencial de los paquetes de 32bit Slackware. Utiliza el script “*convertpkg-compat32*” anterior para capturar todos los paquetes que se encuentran en esta lista interna y los convierte en paquetes '-compat32'.
  Debe ejecutar este script solo una vez, por ejemplo, como este (el ejemplo asume que usted montó su DVD Slackware de 32 bits en `/mnt/dvd`):

```
# massconvert32.sh -i /mnt/dvd/slackware -d ~/compat32
```

Esta acción dará como resultado aproximadamente 150 MB de paquetes nuevos que encontrará en el directorio recién creado `~/compat32` (el nombre del directorio es arbitrario, por supuesto, lo elegí así para este ejemplo). Estos paquetes comprenden el componente de 32 bits de su sistema multilib Slackware64.
Deben instalarse usando “`installpkg`”, y le proporcionan una capa de compatibilidad de 32 bits bastante completa sobre Slackware64:

```
# installpkg ~/compat32/*/*.t?z
```

Si está actualizando desde una versión anterior de estos paquetes (porque, por ejemplo, actualizó su Slackware de 64 bits a una versión más reciente), entonces no usa “`installpkg`”, por supuesto, sino “`upgradepkg –install-new`” en su lugar:

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

El parámetro “*-install-new*” es necesario para instalar los nuevos paquetes *compat32* que se agregaron entre versiones.

Al instalar los paquetes *compat32* notará que algunos mostrarán errores sobre archivos faltantes en `/etc`. Esto es “por diseño”, y estos errores pueden ser ignorados. Estos mensajes son causados por el hecho de que los archivos en `/etc` se eliminan de un paquete “-compat32” durante la conversión (excepto para *pango* y *gtk+2*). Supongo que los paquetes originales de 64 bits ya habrán instalado los archivos en `/etc`.
Un ejemplo de estos “errores” para el paquete `cups-compat32`:

```
Executing install script for cups-compat32-1.3.11-x86_64-1.txz.
install/doinst.sh: line 5: [: too many arguments
cat: etc/cups/interfaces: Is a directory
cat: etc/cups/ppd: Is a directory
cat: etc/cups/ssl: Is a directory
```

```
cat: etc/cups/*.new: No such file or directory
cat: etc/dbus-1/system.d/cups.conf.new: No such file or directory
chmod: cannot access `etc/rc.d/rc.cups.new': No such file or directory
cat: etc/rc.d/rc.cups.new: No such file or directory
Package cups-compat32-1.3.11-x86_64-1.txz installed.
```

Si estaba considerando utilizar el script `convertpkg-compat32` para convertir un paquete **non-Slackware** en un paquete *-compat32*, debo recomendar encarecidamente que no lo haga. El script está escrito con un solo propósito y es hacer que las versiones de 32 bits de los binarios/bibliotecas oficiales de Slackware64 estén disponibles en una configuración multilib. Como tal, la secuencia de comandos eliminará muchas cosas que están presentes en el paquete original de 32 bits, cosas que se espera que se hayan instalado como parte de la versión de 64 bits del paquete. En casi todos los casos en los que ha descargado un paquete de 32 bits non-Slackware y desea que funcione en Slackware64, la mejor manera es encontrar las fuentes y crear una versión de 64 bits del paquete. Alternativamente, simplemente *instale el paquete original de* 32bit en lugar de intentar "convertirlo" y luego ejecútelo desde la línea de comandos para descubrir las bibliotecas de 32bit que faltan que aún debe extraer de un paquete oficial de Slackware.

Ejecutando programas de 32 bits

Ejecutar un programa de 32 bits precompilado es fácil después de haber completado la preparación del sistema anterior. Solo descargue, instale y ejecute!

En ocasiones, es posible que se encuentre con un programa que requiera una cierta biblioteca Slackware de 32 bits que aún no tiene disponible. En ese caso, averigüe qué paquete de 32bit Slackware contiene esta biblioteca faltante. Use el script "*convertpkg-compat32*" para convertir ese paquete Slackware de 32 bits original e instale el paquete resultante de "*compatibilidad*" de 32 bits en Slackware64.

Compilación de programas de 32 bits

En caso de que necesite compilar un programa de 32 bits (wine y grub son dos ejemplos de programas de código abierto que solo son de 32 bits), primero configure el entorno de shell ejecutando (como root) el comando:

```
# . /etc/profile.d/32dev.sh
```

Tenga en cuenta el 'punto' al principio de la línea, que en realidad forma parte de la línea de comandos. El uso del punto es equivalente al comando 'fuente'.
La ejecución de este comando cambia o crea varias variables de entorno. El efecto de esto es que las versiones de 32 bits de los binarios se prefieren a las de 64 bits cuando compila el código fuente: estará ejecutando una compilación de 32 bits. El efecto durará hasta que cierre sesión en su shell. En este entorno modificado, podrá usar SlackBuilds estándar para crear paquetes de 32 bits para Slackware64. Hay un par de cosas a tener en cuenta:

1. Debe definir la variable ARCH como 'i486' porque incluso en su computadora 'x86_64' está compilando un programa de 32 bits.
   Esto se relaciona con el *triplet* de "$ARCH-slackware-linux" que normalmente se usa en el

comando “configurar”.
   1. Como excepción, tendrá que compilar el paquete “wine” con 'ARCH = x86_64' porque instalará este paquete directamente en su computadora multilib sin convertirlo en un paquete 'compat32'.
2. Si desea instalar este paquete de 32 bits en Slackware64-multilib, deberá convertirlo en un paquete 'compat32':

```
# convertpkg-compat32 -i /path/to/your/fresh/foo-VERSION-i486-BUILD.tgz
# upgradepkg --install-new /tmp/foo-compat32-VERSION-x86_64-
BUILDcompat32.txz
```

Advertencias

- Después de instalar los paquetes “`-compat32`”, es posible que tenga que volver a instalar los controladores binarios *Nvidia* o *Ati* video X.Org. Estos paquetes de controladores contienen bibliotecas de 64 bits y de 32 bits para ser de máxima utilidad en un sistema operativo multilib de 64 bits. Si instaló los archivos del controlador para ambas arquitecturas, el paquete “`mesa-compat32`” sobrescribirá algunos de los archivos de la biblioteca de 32 bits.

  Por otra parte, si originalmente *sólo* instaló las bibliotecas de controladores 64bit para su tarjeta Nvidia/Ati, se recomienda después de la instalación de los paquetes *multilib*, volver a instalar el paquete del controlador binario. Esta vez, elija instalar también los archivos del controlador de 32 bits.

  Las aplicaciones gráficas de 32 bits que va a ejecutar en su instalación multilib requerirán estas bibliotecas de controladores de 32 bits. Es probable que se produzcan bloqueos si no instala los archivos correctos.

- Si desea compilar su kernel de 64 bits por sí mismo, asegúrese de compilar la capacidad de emulación de 32 bits en él o de lo contrario, multilib fallará misteriosamente. Necesitará esta pieza de configuración del kernel: **CONFIG_IA32_EMULATION**

Paquetes convertidos por massconvert32.sh

Esta es la lista de paquetes que se convierten en versiones “*-compat32*” mediante el script `massconvert32.sh`. Tenga en cuenta que algunos de estos paquetes no son parte de Slackware 13.0 o 13.1, se agregaron en una versión posterior de Slackware de modo que producen un mensaje “**** FAIL: package 'package_name' was not found!” al ejecutar el script en una versión anterior. Al contrario también es cierto: algunos paquetes se han eliminado en versiones posteriores de Slackware y también activarán el mensaje “***** FAIL: package 'package_name' was not found!*” . No se preocupe por eso.

```
# La serie A/:

aaa_elflibs
attr
bzip2
```

```
cups
cxxlibs
dbus
e2fsprogs
eudev
libgudev
openssl-solibs
udev
util-linux
xz

# La serie AP/:

cups
cups-filters
flac
mariadb
mpg123
mysql
sqlite

# La serie D/:

libtool
llvm
opencl-headers

# La serie L/:

SDL2
alsa-lib
alsa-oss
alsa-plugins
atk
audiofile
cairo
dbus-glib
elfutils
esound
expat
ffmpeg
fftw
freetype
fribidi
gamin
gc
gdk-pixbuf2
giflib
glib2
gmp
gnome-keyring
```

```
gtk+2
gst-plugins-base
gst-plugins-base0
gst-plugins-good
gst-plugins-good0
gst-plugins-libav
gstreamer
gstreamer0
hal
harfbuzz
icu4c
jasper
json-c
lame
lcms
lcms2
libaio
libart_lgpl
libasyncns
libclc
libedit
libelf
libexif
libffi
libglade
libgphoto2
libidn
libieee1284
libjpeg
libjpeg-turbo
libmng
libmpc
libnl3
libnotify
libogg
libpcap
libpng
libsamplerate
libsndfile
libtasn1
libtermcap
libtiff
libunistring
libusb
libvorbis
libxml2
libxslt
lzo
ncurses
ocl-icd
openjpeg
```

```
orc
pango
popt
pulseaudio
python-six
qt
readline
sbc
sdl
seamonkey-solibs
speexdsp
startup-notification
svgalib
v4l-utils
zlib

# La serie N/:

curl
cyrus-sasl
gnutls
libgcrypt
libgpg-error
libtirpc
nettle
openldap-client
openssl
p11-kit
samba

# La serie X/:

fontconfig
freeglut
glew
glu
libFS
libICE
libSM
libX11
libXScrnSaver
libXTrap
libXau
libXaw
libXcomposite
libXcursor
libXdamage
libXdmcp
libXevie
libXext
libXfixes
```

```
libXfont
libXfont2
libXfontcache
libXft
libXi
libXinerama<note tip>tip</note>
libXmu
libXp
libXpm
libXprintUtil
libXrandr
libXrender
libXres
libXt
libXtst
libXv
libXvMC
libXxf86dga
libXxf86misc
libXxf86vm
libdmx
libdrm
libepoxy
libfontenc
libinput
libpciaccess
libva
libva-intel-driver
libvdpau
libxcb
libxshmfence
mesa
pixman
vulkan-sdk
xcb-util

# La serie XAP/:

sane
```

Mirrors de descarga multilib

Puede descargar los paquetes multilib desde (al menos) estas ubicaciones:

- http://slackware.com/~alien/multilib/
- http://bear.alienbase.nl/mirrors/people/alien/multilib/
- http://slackware.uk/people/alien/multilib/
- http://alien.slackbook.org/slackware/multilib/
- http://slackbuilds.org/mirror/alien/multilib/

Herramientas de soporte de terceros

- Sèbastien Ballet ha escrito una herramienta llamada *compat32pkg*. En su sitio web tiene *compat32pkg* disponible para descargar, así como una extensa documentación sobre cómo usarlo en Slackware64.
  Citaré el sitio:
  “Compat32pkg es una herramienta automatizada que proporciona todo lo necesario para administrar (convertir, instalar, actualizar, eliminar) la parte de 32 bits de la multilib de AlienBob para slackware-64, y todos los paquetes de 32 bits de Slackware-32 para los cuales los usuarios pueden encontrar las necesidades en un entorno de 64 bits, como Firefox, seamonkey, jre, … ”
- También hay slackpkg+, escrito por Matteo Rossini (nicknamed zerouno) con contribuciones de (entre otros) Sèbastien Ballet. Este es un complemento para el propio [[http://slackpkg.org/|slackpkg] de Slackware] que agrega la capacidad de instalar paquetes desde repositorios de Slackware no oficiales externos (de terceros). Tiene un buen soporte para agregar multilib a su Slackware de 64 bits y mantenerlo actualizado.

Traducciones

- Bruno Russo tradujo este artículo al portugués (Brasil):
  http://www.brunorusso.eti.br/slackware/doku.php?id=multilib_para_o_slackware_x86_64
- Mehdi Esmaeelpour tradujo este artículo al persa:
  http://www.slack-world.com/index.php/articles/43-general-system/85-multilib-slackware64
- Patrick FONIO y Sebastien BALLET tradujeron este artículo al francés:
  http://wiki.slackware-fr.org/avance:articles:slackware64-multilib
- Victor Adrián Amelotti tradujo este artículo al español:
  https://docs.slackware.com/es:slackware:multilib

Agradecimientos

- Muchas gracias a Fred Emmott, quien creó Slamd64, el tenedor original no oficial de 64 bits de Slackware. Aunque Slackware64 no se basó en el trabajo de Fred, aún aprendí la mayor parte de lo que sé sobre la configuración de la parte de 32 bits de un Linux multilib de sus escritos que se encuentran en Slamd64.
  Tenga en cuenta que Slamd64 tenía los paquetes multilib Glibc y gcc de 64 bits y 32 bits separados. Sin embargo, creo que es más limpio mantener sin división estos paquetes multilib esenciales. Seguí el concepto ya utilizado en el propio paquete *binutils* de Slackware64, que tiene capacidad multilib de 64 bits y 32 bits agrupados en un paquete.
- Aporte de Linux From Scratch.
  La CLFS Wiki (http://trac.cross-lfs.org/wiki/read#ReadtheCrossLinuxFromScratchBookOnline) es una 'lectura obligatoria' si desea saber cómo adaptar Linux a una nueva arquitectura . Tomé varias ideas, conceptos y parches de ellos al crear Slackware64 desde cero, y nuevamente cuando creé mis paquetes multilib gcc/glibc desde cero (mi README en este multilib-from-scratch está disponible en el directorio ./source).

Have fun!

Eric

Fuentes

- El artículo original, escrito por Eric Hameleers, está en http://alien.slackbook.org/dokuwiki/doku.php?id=slackware:multilib

slackware, multilib, author alienbob

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

Permanent link:
https://docs.slackware.com/es:slackware:multilib

Last update: **2019/03/18 00:15 (UTC)**