Soporte Qemu en Slackware ARM

Este documento describe el proceso de instalación de **Slackware ARM** dentro de QEMU.

- QEMU es un emulador de sistema completo que puede emular una amplia gama de arquitecturas de hardware reales. En este caso usaremos QEMU para emular a ARM Ltd. Tarjeta de desarrollo “Versatile Express”.
- QEMU proporciona una plataforma que permite a un sistema operativo creer que se está ejecutando en hardware real.
- QEMU se ejecuta sobre tu pc o servidor Slackware. En muchos casos esto puede ser una máquina de escritorio. Aunque, es posible ejecutar QEMU en un servidor sin monitor y usar VNC para gráficos.

¿A quién va dirigido Slackware ARM en QEMU?

El objetivo de instalar Slackware ARM dentro de QEMU es permitir a las personas que no tienen hardware ARM probar Slackware ARM.

Mientras que QEMU es un emulador excelente, es muy lento comparado con el hardware real de ARM. QEMU se utilizó durante aproximadamente un año para desarrollar la mayor parte de Slackware ARM versión 12.2. Debido a la lenta velocidad de QEMU, distcc puede ser usado en varias máquinas x86 para acelerar las compilaciones. Es posible utilizar Slackware ARM de forma efectiva en QEMU, sin embargo, tenga en cuenta que no presentará la mejor experiencia de usuario debido a su falta de velocidad. Lo ideal es que Slackware ARM se ejecute en hardware ARM real.

Este documento es un trabajo en curso y su objetivo es la publicación de ARM, Slackwarearm-current.

Documentación para Slackware ARM 14.0, 14.1, y 14.2, puede ser encontrado aquí:

- Slackwarearm-14.0
- Slackwarearm-14.1
- Slackwarearm-14.2

Suposiciones del entorno de instalación

Se hacen varias suposiciones para escribir este documento.

- La máquina anfitriona está ejecutando una instalación completa de Slackware x86 o x86_64.
- Usted tiene acceso de root en la máquina anfitriona y puede compilar software.
- La máquina anfitriona Slackware y el emulador QEMU están sobre una red segura local (LAN).
- Puede exportar un directorio compartido (NFS) en la máquina anfitriona que comparte el árbol de Slackware ARM con el huésped de QEMU.
- El directorio raíz del directorio exportado mediante NFS en la máquina anfitriona se encuentra en /export.

Estas directrices se pueden cumplir con una máquina anfitriona Slackware x86/x86_64. Esta única máquina anfitriona puede servir el contenido de NFS, así como alojar la instalación de Slackware ARM dentro de QEMU.

Prerequisitos para una máquina anfitriona Slackware x86

1. Descarga el árbol de paquetes de distribución **Slackwarearm** con rsync.
2. Complete una ubicación con los archivos QEMU de Slackware ARM.
3. Configurar el Sistema de Archivos de Red (NFS) en el Host x86 de Slackware.
4. Instalar QEMU y el software del compilador de árbol de dispositivos en el host x86 de Slackware.
5. Configurar los permisos de QEMU en el host x86 de Slackware.
6. Crear una imagen del disco QEMU usando los scripts de ayuda.

Descargar Slackware ARM

Descargar Slackware ARM a tu anfitrión Slackware x86. En este tutorial es utilizado rsync para un espejo(mirror) Slackware ARM. Antes de ejecutar rsync, haga que su usuario haya leído, escrito y ejecutado permisos dentro del directorio /export.

```
mkdir -p /export/slackwarearm
cd /export/slackwarearm
rsync --exclude '*/source/*' --delete -Pavv
ftp.arm.slackware.com::slackwarearm/$SLACKREL .
```

Si tu quieres puede usar una navegador web, wget, o lftp para descargar Slackware ARM, el espejo (mirror) esta aquí.

Las variables $SLACKREL se refieren a el árbol Slackware ARM que descargaste. Si elegiste Slackwarearm-14.2, la url para rsync sera ftp.arm.slackware.com::slackwarearm/slackwarearm-14.2

Complete los archivos y directorios de Slackware ARM

Para arrancar el instalador de Slackware ARM necesita crear un directorio que almacene el Kernel y el disco RAM inicial. También necesita algunos scripts de ayuda para ejecutar QEMU. En este tutorial, todos estos archivos se almacenarán en /export/armhost.

Copiar el kernel y el disco RAM:

```
mkdir -p /export/armhost
cp -fa /export/slackwarearm/$SLACKREL/kernels/armv7/{zImage*,initrd*,dtb}
```

```
/export/armhost/
cp -fa /export/slackwarearm/$SLACKREL/isolinux/initrd-armv7.img
/export/armhost/
```

Descargar y copiar los scripts de ayuda de QEMU:

```
# cd /tmp ; mkdir qemu ; cd qemu
# rsync -Prlvv --delete
ftp.arm.slackware.com::slackwarearm/boardsupport/qemu/$SLACKREL .
# cd $SLACKREL
# cp -fav helper-scripts/* /export/armhost/
```

Usted puede buscar los scripts de ayuda aquí si no desea utilizar rsync para descargarlas.

La variable $SLACKREL se refiere al árbol Slackware ARM tree que descargaste. Si elegiste Slackwarearm-14.2, el camino (path) puede ser /export/slackwarearm/slackwarearm-14.2. La URL para rsync para los scripts de ayuda de QEMU helper scripts son:

```
ftp.arm.slackware.com::slackwarearm/boardsupport/qemu/slackwarearm-14.2
```

Configuración de los servicios de sistema de archivos(NFS) de red

El anfitrión x86 de Slackware necesita ser configurado apropiadamente para poder ejecutar el proceso de instalación de Slackware arm dentro de QEMU. La forma más fácil y eficiente de hacerlo es configurando un recurso compartido de Sistema de archivos de red (NFS) en el equipo host. Este recurso compartido NFS será utilizado por el cliente Slackwarearm (que se ejecuta dentro de QEMU) para acceder al árbol de paquetes de distribución de Slackware en el host. Los servicios de NFS no son la única manera de servir el árbol de paquetes de distribución de Slackware al cliente de QEMU. Otros métodos son un poco más lentos cuando se usan con redes emuladas de QEMU. Como resultado, el servicio NFS se utiliza para este tutorial.

Necesita conocer la configuración de red de QEMU para acceder al recurso compartido NFS. Se recomienda que utilice el modo NAT de conexión en red con QEMU. El modo NAT permite el acceso directo al recurso compartido NFS del host x86 de Slackware a través de una red virtual alojada en QEMU. Más adelante en esta guía se discute más información sobre la red virtual y el direccionamiento asignado.

Con un editor de texto, como root, agregue los siguientes host's al archivo /etc/exports:

```
# QEMU guest virtual IP address
/export/slackwarearm
10.0.2.15(insecure,ro,nohide,root_squash,sync,no_subtree_check)
```

El modo NAT no permite el acceso directo a la red de área local física. Tendrá que configurar una interfaz de red puenteada si decide que necesita acceso directo a la red. El modo NAT será suficiente

para la mayoría de los usuarios.

Utilice algo similar a lo siguiente si planea utilizar una interfaz de red puenteada:

```
/export/slackwarearm
xxx.xxx.xxx.x/255.255.255.0(insecure,ro,nohide,root_squash,sync,no_subtree_c
heck)
```

Reemplace xxx.xxx.x.x/255.255.255.0 y equipáralo con la configuración de su red. (Ejemplo: 192.168.1.0/255.255.255.0)

Active el recurso compartido NFS ejecutando lo siguiente como root:

```
# chmod +x /etc/rc.d/{rc.rpc,rc.nfsd}
# /etc/rc.d/rc.nfsd start
# exportfs -va
```

Instalar QEMU y el compilador de árbol de dispositivos

La instalación de SlackBuildes no está fuera del alcance de este artículo. Si necesita ayuda con la instalación de QEMU o del compilador de árboles de dispositivos, por favor, consulte la sección Paginas HOWTO de SlackBuilds.org. Dicho esto, hay algunas recomendaciones:

- Si estas corriendo Slackware-current puedes instalar QEMU y omita la instalación del paquete dispositivo-árbol-compilador. Slackware-current ya incluye el paquete de compilador de árbol de dispositivos en una instalación completa.
- Asegúrese de haber realizado una instalación **completa** de Slackware en su host x86 antes de instalar estos SlackBuilds.
- Los usuarios de Slackware 14.0, 14.1 y 14.2 necesitan instalar paquete de compiladores de dispositivos de árbol de SlackBuilds.org antes de instalar QEMU.
- Puede descargar e instalar QEMU de SlackBuilds.org.
- No tendrá la capacidad en los sistemas 14.0, 14.1 y 14.2 de emular la arquitectura ARM en QEMU si no instala primero el paquete del compilador del árbol de dispositivos.

Permisos para QEMU

Hay algunos permisos que deben ser establecidos una vez que haya instalado correctamente QEMU en su sistema. El cliente QEMU se iniciará ejecutando el binario /usr/bin/qemu-system-arm. Este binario necesita permisos de root para ser ejecutado. Puede ejecutar este binario con sudo editando /etc/sudoers. Usar sudo es la opción más segura si tiene varios usuarios en su sistema. Si usted es el único usuario de su sistema, es suficiente con configurar el permiso setuid como root. Ajustar estos permisos permitirá a un usuario normal configurar y arrancar el huésped de QEMU sin tener que iniciar sesión como root.

Para establecer el permiso setuid root es necesario que inicie sesión como root. Como root ejecute los siguientes comandos:

```
# chmod +s /usr/bin/qemu-system-arm
```

Si planeas configurar QEMU para que use una red puente, vas a necesitar configurar los permisos setuid para /sbin/ifconfig y /sbin/brctl.

```
# chmod +s /sbin/{ifconfig,brctl}
```

Crear una imagen de disco para QEMU

Antes de arrancar el instalador de Slackware ARM en QEMU, debe crear una imagen de disco que actúe como una emulación de una SD Card. Este disco de imagen es usado para emular el controlador MMC en Slackware ARM. Anteriormente copió los scripts de ayuda de Slackware ARM en /export/armhost/. Dentro de este directorio hay un script, **makeimg** . Este script crea una imagen de disco de 15GB automáticamente en /export/armhost/ cuando se ejecuta. Inicialmente, todo lo que necesita hacer es ejecutar este script. Cambia al directorio donde copiaste los scripts de ayuda y ejecuta **makeimg**:

```
cd /export/armhost
./makeimg
```

Para referencia, este es el script **makeimg**:

```
# Create the QEMU disk image - the emulated SD card.

IMG=sdcard.img
SIZE=15G

rm -f $IMG
qemu-img \
   create \
   $IMG $SIZE
```

Tenga en cuenta que una vez que haya instalado Slackware ARM en esta imagen de disco debe moverlo a un directorio diferente para su almacenamiento, o se arriesga a que se destruya cuando **makeimg** se ejecute en un momento posterior.

Configuración de red para QEMU

Esta sección cubre el proceso de configuración de redes de invitados para QEMU. Dos tipos de procesos son descriptos. *Network Address Translation mode (NAT)* es la primera y la forma recomendada para obtener una red red funcional en los huéspedes de QEMU. La segunda forma es el *modo puente*. El modo NAT no permite el acceso directo a la red física del host x86 Slackware y el modo puente sí. Es mejor utilizar el modo puente si planea realizar operaciones de red más avanzadas que requieran acceso completo al host y a la red física del host. **La mayoría de los usuarios querrán usar el modo NAT**.

Existen muchas formas diferentes para configurar la red de invitados para QEMU. Este documento solo cubrirá las funcionalidades de QEMU para que Slackware ARM arranque. Refiérase a las paginas del manual o a la documentación de QEMU si necesitas una explicación más detallada.

Más tarde, cuando arranque el instalador de Slackware ARM en QEMU, puede que tenga que modificar el **txqueuelen** para sus interfaces de red. Estos es debido a que los paquetes grandes de Slackware caducan (time out) mientras se descargan mientras se descargan del recurso compartido NFS en el host. Esto ocurre porque la emulación de QEMU es muy lenta. El demonio NFS de su máquina anfitriona ocasionalmente apaga el socket de red antes de que los paquetes grandes (rust, kernel-firmware, etc.) terminen de ser copiados a la tarjeta SD. Configurar la txqueuelen a **10000** para todas las interfaces de red debería ser suficiente para prevenir esta anomalía. El siguiente comando parece resolver este problema:

```
ip link set eth0 txqueuelen 10000
```

Ejecute este comando para cada interfaz de red utilizada activamente por QEMU.

Modo de red NAT para QEMU

El modo NAT no requiere una configuración adicional sobre una máquina Slackware x86 o en la máquina invitada QEMU. Aquí hay un ejemplo abreviado de un invitado de QEMU que está siendo lanzado con el modo de red NAT:

```
# cd /export/armhost
# qemu-system-arm -nographic \
   -m 1024 \
   -M vexpress-a9 \
   -k en-us \
   -net nic \
   -net user
   ..snip..
```

Las opciones **-net nic** y **-net user** permiten que QEMU inicie el huésped de Slackware ARM con el modo de red en modo NAT activado. Estos ajustes se documentan más en los scripts de ayuda **installer_launch** y **disk_launch**. Con el modo NAT habilitado, QEMU lanza una red virtual 10.0.2.0/24. Al huésped de QEMU se le asignará la dirección IP 10.0.2.15. El invitado puede acceder al host x86 de Slackware en 10.0.2.2 y el servidor DNS de QEMU se ejecuta en 10.0.2.3. QEMU no tiene acceso directo a la red de área local del host. Esto significa que al huésped de QEMU no se le asigna una dirección IP física por el servicio DHCP de su router. No podrá hacer ping al invitado de QEMU desde la máquina anfitriona, pero el invitado debería poder hacer ping a la máquina anfitriona en 10.0.2.2. El huésped de QEMU debe poder acceder a Internet y comunicarse con la máquina anfitriona.

Modo de red puente para QEMU

La mejor manera de configurar una interfaz de red en puente para QEMU es con el script de ayuda proporcionado, rc.local-additions. Si decide tomar esta ruta, debe desactivar el servicio NetworkManager. NetworkManager viene con Slackware, pero no está desarrollado por Slackware. Como resultado, el proceso de configuración de un puente con NetworkManager no es soportado en esta guía.

Puede encontrar el script de ayuda con comentarios en línea aquí: rc.local-additions

Los siguientes comandos deben ejecutarse como root para desactivar NetworkManager:

```
# /etc/rc.d/rc.networkmanager stop
# chmod -x /etc/rc.d/rc.networkmanager
```

Las siguientes configuraciones de red son asumidas para la **máquina host Slackware x86**. Ajuste estos valores a lo largo del resto de esta guía si utiliza diferentes configuraciones de red.

```
Default Gateway: 192.168.1.1
Static IP address: 192.168.1.2
Network Mask: 255.255.255.0 / 192.168.1.0/24
Name server: 192.168.1.1
```

Necesitas editar el archivo script de ayuda *qemu-network-tun.sh* para QEMU. Se requiere que se muestre la interfaz de red de invitados de QEMU. debería existir en /export/armhost. Cambiar la dirección IP listada en $BRIDGEIP para que coincida con tu configuración de red.

Archivo: /export/armhost/qemu-network-tun.sh

```
#!/bin/sh

# This is the IP of 'tap0' on the Slackware/x86 host:
BRIDGEIP=192.168.1.4

modprobe tun
/sbin/ifconfig $1 $BRIDGEIP netmask 255.255.255.0
/sbin/brctl addif br0 $1
```

Aquí está la parte relevante del script rc.local-addtions que requiere modificación. Este script está pensado para reemplazar en la máquina anfitriona /etc/rc.d/rc.local. Los cambios dependen de la configuración de su red:

```
.. snip ..

# Turn on the bridge.  Note that this is a different IP from
# the one specified in your qemu-network-bridge.sh script
# in your 'armhost' directory on your Slackware x86 box.
# You need a bridge IP, a tunnel (tap0) IP, and then another
# IP which is assigned to the Slackware ARM host (by Linux inside QEMU)
# to its own eth0.
```

```
# I tried bringing this up after eth0 but the bridge didn't work.
# I don't know why that is!
ifconfig br0 192.168.1.3 up

# Put back the original IP for eth0:
ifconfig eth0 192.168.1.2 up

#
route del default
route add default gw 192.168.1.1
```

Estos ajustes de red asumen direcciones IP estáticas. La interfaz br0 es la interfaz de red en modo puente. La interfaz eth0 es la interfaz de red de la máquina anfitriona que permite que la máquina anfitriona conserve la conectividad de red. La puerta de enlace predeterminada, 192.168.1.1, suele apuntar a la puerta de enlace de la red física.

Copiar rc.local-addtions a /etc/rc.d/rc.local una vez que termine de editarla. A continuación, márquela como ejecutable.

```
# cp /path/to/rc.local-additions /etc/rc.d/rc.local
# chmod +x /etc/rc.d/rc.local
```

En este punto es recomendable rebootear tu Slackware x86 host para asegurar que las configuraciones en rc.local están siendo usados y el NetworkManager es completamente deshabilitado.

Necesitas editar el archivo /etc/resolv.conf sobre el host. Agregar en la dirección IP de tu DNS primario y secundario ya que no está recibiendo estas direcciones IP por otros medios. El servidor de nombres es 192.168.1.1 (o puerta de enlace por defecto) en este tutorial

Modifique los scripts del ayudante del lanzador una vez que esté seguro de que su host tiene la configuración de red adecuada. Preste mucha atención a la variable **$NETTYPE**. Los detalles sobre cómo utilizar esta variable están documentados en los scripts de ayuda **installer_launch** y **disk_launch**. Edita la variable **$MACADDR** para cada instancia de QEMU si estás ejecutando más de un invitado de Slackware ARM a la vez.

Instalar Slackware ARM

Asumiré que ahora está en X Windows, ejecutándose como su cuenta de usuario normal, y que ha seguido los pasos descritos anteriormente en este documento.Como se ha dicho anteriormente, QEMU se ejecuta extremadamente lento cuando se emula la arquitectura de ARM. Dependiendo de la configuración de su hardware, el instalador de Slackware puede tardar varias horas o más en copiar todos los paquetes al disco emulado. Si el instalador de Slackware parece no responder, compruebe el monitor de procesos de su sistema (top o htop) para ver si el proceso de QEMU está todavía activo.

Una buena señal de que QEMU está todavía activo es que un solo núcleo de CPU está operando al 100 por ciento.

El instalador de Slackware ARM es en su mayoría idéntico al instalador de Slackware x86. No hay una curva de aprendizaje para instalar Slackware ARM si ya ha instalado Slackware anteriormente.

Iniciando el instalador

Para poder arrancar el instalador necesitará configurar y ejecutar el script **installer_launch** dentro de una ventana de terminal.

```
cd /export/armhost
./installer_launch
```

El script **installer_launch** puede ser encontrado aquí.

Verá algunas advertencias de QEMU sobre la imposibilidad de abrir dispositivos de audio y vídeo. Estas advertencias pueden ignorarse de forma segura. A continuación verá los mensajes de arranque del núcleo de Linux y eventualmente el instalador preguntando sobre el mapa de claves que quiere usar. Una vez que seleccione su mapa de claves e inicie sesión en el sistema notará que el instalador obtiene una dirección IP a través de DHCP. La dirección IP asignada por DHCP y la configuración de red resultante depende en gran medida de cómo configure su red en la máquina anfitriona. Si QEMU no asigna una dirección IP al huésped, entonces tendrá que volver y verificar que los ajustes de red estén configurados adecuadamente.

Particionando

La tarjeta SD emulada creada con el comando **makeimg** es una imagen en blanco. Tendrá que particionar esta tarjeta SD con el instalador. En nuestro caso es mejor mantener el esquema de partición simple. Se recomienda que cree una partición de intercambio de 200 MB y que asigne el resto del disco a la partición raíz. Puede utilizar las herramientas **fdisk** o **cfdisk** para crear las particiones.

Ejemplo de un esquema de partición:

```
/dev/mmcblk0p1 - 200MB swap
/dev/mmcblk0p2 - el resto del disco, "Linux" - tipo 83.
```

Instalación y configuración

Correr el comando **setup** en el intérprete de comandos del shell después de salir de la herramienta de particionado. Haga que el instalador reconozca su partición de intercambio y su partición raíz. Se recomienda que seleccione el sistema de ficheros ext4 cuando formatee la partición raíz. A continuación se le pedirá que seleccione el medio de origen. Elija la opción 4, **Instalar desde NFS (Network Filesystem)**. Introduzca la dirección IP de su host Slackware x86. Introduzca la ruta del recurso compartido montado en NFS.

El camino completo del recurso compartido NFS es requerido:

```
Ingrese la dirección IP: 192.168.1.2 # Host machine IP address
Ingrese el directorio: /export/slackwarearm/$SLACKREL/slackware
```

A continuación, se le pedirá que seleccione un paquete. Slackware ARM tiene todos los paquetes estándar de Slackware excepto los que son x86 solamente. Es altamente recomendable que hagas una instalación **full** para satisfacer todas las dependencias del sistema. Por favor se paciente, este es el momento en el que más tiempo se consume en el proceso de instalación.

Luego de que la instalación allá finalizado, ejecute 'MKFONTDIR y MKFONTSCALE UPDATE' tome un largo tiempo.

En la pantalla de configuración de red es mejor seleccionar la opción DHCP. La opción DHCP es la que mejor complementa al modo NAT de QEMU y a las dos opciones de red en modo puente. La única razón por la que no se debe seleccionar DHCP es si la red física utiliza direcciones IP estáticas.

A continuación, llegará a la selección del Administrador de ventanas para el servidor X Windows. Se recomienda seleccionar un gestor de ventanas ligero, como Fluxbox o WindowMaker. KDE y Xfce no son muy útiles dentro del huésped de QEMU debido a las limitaciones de velocidad.

Post-Instalación

Una vez que haya completado el proceso de instalación debería ir a un intérprete de comandos del shell para configurar el demonio SSH. Por omisión, OpenSSH no permite a root iniciar sesión con una contraseña. Esto es un problema de seguridad. Puede que quiera pensar en esto cuidadosamente si su dispositivo está conectado directamente a una red no confiable. Es mejor crear una cuenta de usuario para conexiones remotas al servicio SSH y escalar los privilegios localmente con “su” o “sudo”. Si desea usar root para conectarse remotamente, siga estos pasos:

1. Optar por ingresar a un 'shell' al salir del instalador
2. en el shell, ingrese:

```
# sed -i 's?^#PermitRootLogin.*?PermitRootLogin yes?g'
/mnt/etc/ssh/sshd_config
# poweroff
```

De esta forma se completa el proceso de instalación de Slackware ARM dentro de QEMU.

Iniciar Slackware ARM con QEMU

¡Felicitaciones por haber llegado hasta aquí! El proximo paso es bootear en tu instalación fresca de Slackware ARM. Localice el script de ayuda **disk_launch** en /export/armhost y modifique este para que se adecué a sus necesidades.

```
# cd /export/armhost
# vi disk_launch
```

Este script tiene algunas variables que podes cambiar.

- **ROOTFSTYPE** - Tipo de sistema de archivos raíz para el sistema, ext4 es seleccionado por defecto.
- **ROOTFSDEV** - location of the root partition within the SD Card image
- **KEYBOARD** - keyboard locale you wish to use, typically the same as what you chose during installation
- **NETTYPE** - configuración de red, modo NAT o modo puente.

The **disk_launch** script for Slackwarearm-current can be found online, here.

El primer inicio del sistema tardará bastante. Esto se debe a que Slackware generará la caché de fuentes por primera vez. Arranca QEMU ejecutando el script disk_launch.

```
# ./disk_launch
```

Asumiendo que todo está bien, puedes empezar a usar Slackware ARM como lo harías con cualquier otra instalación de Slackware.

Slackware ARM Interfaz gráfica de usuario

Trabajo en progreso

Esta sección discutirá los aspectos positivos y negativos de ejecutar X Windows, qué administrador de ventanas o entorno de escritorio usar, y las formas en que puede iniciarlo.

Fuentes

- Escrito originalmente por: Stuart Winter.
- Fuente original: http://ftp.arm.slackware.com/slackwarearm/boardsupport/qemu/.
- Traducido por: — *rramp 2019/07/31 03:10 (UTC)*.

- Modificado y mantenido por: Brenton Earl (mralk3).

howtos, hardware, arm, user mralk3