

Slackware ARM sobre la Raspberry Pi 3

La Raspberry Pi 3 tiene un chip Broadcom BCM2837 SoC el cual incorpora una CPU de cuatro núcleos ARMv8 Cortex-A53[64 bits] a 1,2 GHz y una GPU VideoCore IV a 400 MHz, y su memoria RAM es LPDDR2 SDRAM de 1 GB a 900 MHz.

Esta versión revisada y actualizada de la placa [Raspberry Pi \(2\)](#), es la sucesora la cual es mucho más rápida y más poderosa. Se han incluido interfaces Wi-Fi y Bluetooth pero no un reloj de tiempo real (RTC). Piense la RPi3 como una renovación y no como una innovación. Slackware ARM, como es de esperar, funciona perfectamente en este dispositivo, con un incremento significativo en la velocidad. Los tiempos de compilación son mucho más cortos en comparación con el RPi2, por ejemplo.

La placa Raspberry Pi 3 es apoyada fuera del árbol oficial de Slackware ARM por la comunidad de Slackware.

Versión Slackware 14.2, -current

Slackware ARM -current o Slackware ARM 14.2 puede ser instalado sobre la Raspberry Pi 3.

Siga los enlaces de la siguiente tabla. Estos son mantenidos por un tercero como parte de la comunidad de Slackware sobre Raspberry Pi.

Sitio web	Versiones de Slackware	Usa paquetes oficiales de Slackware	Método de instalación	Notas
SARPi Project	14.2,-current	Si	Instalador de Slackware	Un tutorial de principio a fin que le guiará a través del proceso de instalación y configuración.

Siempre y cuando utilice la imagen y el firmware raspbian más recientes, la opción [Método del manual de instalación de Raspberry Pi 1](#) también funciona para la Pi 2 y Pi 3.

AArch64 ARM64 [Experimental], Slackware ARM -current

Experimental, desarrollo, y prototipo, Slackware AArch64 ARM64 link(s).

Sitio	Versiones de Slackware	Usa paquetes oficiales de Slackware	Métodos de Instalación	Notas
SARPi64 Project	-current	Si	Instalador de Slackware	Un proycto para Slackware ARM corriendo AArch64 [ARMv8] kernel y módulos. Experimental in nature and purpose.

Método de instalación manual sin una imagen de Raspbian

Este método es para instalar Slackware ARM 14.2 sobre Raspberry Pi 3 modelo B sin una imagen de

Raspbian. Sin embargo, debería funcionar para otras versiones de Slackware ARM y Raspberry Pi.

1. Partición y formato de la tarjeta SD

```
$ sudo fdisk -l /dev/mmcblk0

Disk /dev/mmcblk0: 31.9 GB, 31914983424 bytes
4 heads, 16 sectors/track, 973968 cylinders, total 62333952 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

    Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1            2048        67583        32768    b   W95 FAT32
/dev/mmcblk0p2         67584       62333951       31133184   83   Linux
$ sudo mkfs.vfat /dev/mmcblk0p1
$ sudo mkfs.ext4 /dev/mmcblk0p2
```

Observaciones:

- Yo uso una tarjeta SD de 32GB.
- Yo elijo 32MB para el tamaño de la primera partición.
- Yo dejé el espacio vacío que quedaba para la segunda partición.

2. Ponga el firmware de la Raspberry Pi en la tarjeta SD

```
$ git clone https://github.com/raspberrypi/firmware.git
$ sudo mount /dev/mmcblk0p1 ~/mnt
$ sudo cp -r firmware/boot/* ~/mnt
$ sudo umount ~/mnt
$ sudo mount /dev/mmcblk0p2 ~/mnt
$ sudo mkdir -p ~/mnt/lib/modules
$ sudo cp -r firmware/modules/* ~/mnt/lib/modules
$ sudo umount ~/mnt
```

3. Ponga el sistema de archivos mini root de Slackware ARM en la tarjeta SD

```
$ wget -c
ftp://ftp.arm.slackware.com/slackwarearm/slackwarearm-devtools/minirootfs/ro
ots/slack-14.2-miniroot_01Jul16.tar.xz
$ sudo mount /dev/mmcblk0p2 ~/mnt
$ sudo tar -C ~/mnt -xf slack-14.2-miniroot_01Jul16.tar.xz
$ echo "/dev/mmcblk0p1 /boot vfat defaults 0 0" | sudo tee ~/mnt/etc/fstab
$ echo "/dev/mmcblk0p2 /      ext4 defaults 0 0" | sudo tee -a
~/mnt/etc/fstab
$ echo "proc                /proc proc defaults 0 0" | sudo tee -a
```

```
~/mnt/etc/fstab
$ PASSWD=$(openssl passwd -1 -salt cetkq/enZx6/c2 password)
$ sudo sed -i "s|\(root:\).*\(:16983:0:::::\)|\1${PASSWD}\2|"
~/mnt/etc/shadow
$ sudo sed -i 's|USE_DHCP\[1\]=""|USE_DHCP\[1\]="yes"|'
~/mnt/etc/rc.d/rc.inet1.conf
$ echo "PermitRootLogin yes" | sudo tee -a ~/mnt/etc/ssh/sshd_config
$ sudo umount ~/mnt
```

Observaciones:

- Yo configuré “password” como clave para el usuario “root”.
- Yo configuré DHCP sobre la interfaz de red “eth1”.
- Yo permito que el usuario “root” se pueda conectar a través de SSH.

4. Inserte la tarjeta SD en la Raspberry Pi

Tu tarjeta SD está lista para insertarla en la Raspberry Pi e iniciar.

Puedes conectarte a tu Raspberry Pi de forma remota como “root” a través de SSH.

```
$ ssh root@raspberrypi
```

Tan pronto como inicie sesión, es posible que desee instalar paquetes adicionales de Slackware ARM:

```
$ wget --mirror ftp://ftp.arm.slackware.com/slackwarearm/slackwarearm-14.2
$ upgradepkg --install-new
ftp.arm.slackware.com/slackwarearm/slackwarearm-14.2/slackware/*/*.txz
$ removepkg
ftp.arm.slackware.com/slackwarearm/slackwarearm-14.2/slackware/*/kernel_*.txz
```

Observaciones:

- Yo considero que el nombre(hostname) de la Raspberry Pi es “raspberrypi”.
- Recomiendo añadir un usuario normal y usar este usuario en lugar de “root”.
- Recomiendo cambiar la clave del usuario “root”.
- Recomiendo deshabilitar las conexiones a través de SSH para el usuario “root” editando el archivo /etc/ssh/sshd_config.
- Recomiendo [construir tu propio kernel](#) porque el kernel que se está ejecutando no coincide con los paquetes instalados de Slackware ARM.

5. Sugerencias y trucos

5.1. Bluetooth

La Raspberry Pi tiene un chip Bluetooth de la empresa Broadcom. Aunque, el firmware requerido no está instalado sobre Slackware ARM. Esto significa que necesita descargarlo e instalarlo:

```
$ git clone https://github.com/OpenELEC/misc-firmware.git
$ mkdir -pv /where/you/want/to/install/etc/firmware
$ cp -v misc-firmware/firmware/brcm/BCM43430A1.hcd
/where/you/want/to/install/etc/firmware
```

Entonces construye el paquete para Slackware ARM `bluez-firmware-brcm` e instalalo.

Con el objetivo de habilitar Bluetooth necesitas agregar el módulo Bluetooth, correr el demonio Bluetooth, conecte el dispositivo, luego abra e inicialice el dispositivo. Agregue las siguientes líneas al final de el archivo `/etc/rc.d/rc.local`:

```
if ps axc | grep -q bluetoothd; then
    killall bluetoothd
fi
if ps axc | grep -q hciattach; then
    killall hciattach
fi
if lsmod | grep -q hci_uart; then
    rmmmod hci_uart
fi
if lsmod | grep -q btbcm; then
    rmmmod btbcm
fi
modprobe btbcm
bluetoothd &
while ! hciattach /dev/ttyAMA0 bcm43xx; do
    echo "Unable to attach the device! We try again..."
done
hciconfig hci0 up
```

Observaciones:

Remark: Sometimes there is a failure with the `hciattach` command so that is why you need a `while`.

Para chequear que el Bluetooth está trabajando tipee:

```
$ hcitool dev
$ hcitool scan
$ sudo bluetoothctl
```

Ahora el Bluetooth está configurado correctamente.

5.2. Teclado

El mapa de teclado predeterminado en Slackware ARM es el de Reino Unido. Si tu quieres cargar otro mapa de teclado, edite el archivo `/etc/rc.d/rc.keymap`.

Si desea cambiar la distribución del teclado para X11, debe copiar el archivo de configuración X11 y

luego editarlo:

```
sudo cp /usr/share/X11/xorg.conf.d/90-keyboard-layout.conf
/etc/X11/xorg.conf.d
```

Ahora, el teclado esta correctamente configurado.

5.3. Memoria

Desafortunadamente, la Raspberry Pi tiene solo 1 GB de memoria. Esto significa que eventualmente no podrá ejecutar algunas aplicaciones que requieren más memoria. Sin embargo, puede ampliar su memoria creando una partición de intercambio temporal de 4 GB:

```
$ dd if=/dev/zero of=/tmp/swap bs=1M count=4000
$ mkswap /tmp/swap
$ sudo swapon /tmp/swap
```

Usted puede chequear la memoria actual tipeando:

```
$ free
```

Ahora, la memoria está correctamente configurada.

5.4. Procesador

El procesador de la Raspberry Pi puede alcanzar los 1,2 GHz. Aunque, por defecto, se mantiene a 600MHz incluso si se utiliza al 100%. Podes chequear la frecuencia actual del procesador tipeando:

```
$ cpufreq-info
```

Con el objetivo de alcanzar la frecuencia de 1,2 GHz cuando el procesador es usado al 100% (es decir, usa la escala de frecuencia), es necesario cambiar los gobernadores predeterminados. Agregue la siguiente línea al final de el archivo `/etc/rc.d/rc.local`:

```
echo ondemand | sudo tee
/sys/devices/system/cpu/cpu*/cpufreq/scaling_governor
```

Ahora, el procesador está correctamente configurado.

5.5. Tiempo

Desafortunadamente, la Raspberry Pi no incluye un reloj de tiempo real (RTC). Es por eso que no hay ninguna batería incluida con la placa. ¡Esto significa que cada vez que se apaga la Raspberry Pi, el tiempo se reinicia! Sin embargo, si tiene acceso a Internet, puede actualizar la hora durante el arranque de Slackware ARM. Agregue la siguiente línea al final del archivo `/etc/rc.d/rc.local`:

```
ntpdate pool.ntp.org
```

Ahora, el tiempo está correctamente configurado.

5.6. Video

Desafortunadamente, la Raspberry Pi no es compatible con OpenGL (es compatible con OpenGL ES que es un subconjunto de OpenGL). Esto significa que, por defecto, cada aplicación que requiera OpenGL será lenta. Sin embargo, puede alcanzar los 60 FPS con aplicaciones OpenGL en la Raspberry Pi utilizando el controlador correcto.

Primero, necesitas construir Mesa ($\geq 17.0.4$) con el controlador VC4 DRI:

```
$ CFLAGS="-O2 -march=armv8-a -mtune=cortex-a53 -mfloat-abi=softfp -  
mfpv4" \  
  CXXFLAGS="-O2 -march=armv8-a -mtune=cortex-a53 -mfloat-abi=softfp -  
mfpv4" \  
  ./configure \  
  --prefix=/usr \  
  --sysconfdir=/etc \  
  --with-dri-driverdir=/usr/lib/xorg/modules/dri \  
  --with-egl-platforms=x11,drm \  
  --with-gallium-drivers=vc4  
$ make -j4  
$ make install DESTDIR=/where/you/want/to/install
```

Luego construya su propio paquete Slackware ARM Mesa e instálelo.

En segundo lugar, añada la siguiente línea al final de la sección del archivo `/boot/config.txt`:

```
dtoverlay=vc4-fkms-v3d
```

Entonces reinicie la Raspberry Pi.

Puede comprobar que es capaz de obtener 60 FPS con aplicaciones OpenGL en la Raspberry Pi escribiendo el siguiente comando en una terminal X:

```
$ glxgears
```

Ahora, el video está correctamente configurado.

5.7. UART

En el archivo `/etc/inittab`, `s0` se refiere a la mini [UART](#) que está desactivado por defecto en la Raspberry Pi.

Por consiguiente, cada 5 minutos, esta línea se escribirá en `/dev/tty1` por el proceso `init`:

```
INIT: Id "s0" respawning too fast: disabled for 5 minutes
```

An easy fix is to edit the `/etc/inittab` file and replace the following line:

```
s0:12345:respawn:/sbin/agetty 115200 ttyS0 vt100
```

with this one:

```
#s0:12345:respawn:/sbin/agetty 115200 ttyS0 vt100
```

Entonces reinicia la Raspberry Pi. Ahora, la UART esta correctamente configurada.

Fuentes

- Escrito originalmente por [Exaga](#).
- Contribuciones por [yugiohjcj](#).
- Traducido al español por — [rramp](#) 2019/04/11 16:57 (UTC)

[howtos](#), [hardware](#), [arm](#), [author exaga](#)

From:

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

Permanent link:

<https://docs.slackware.com/es:howtos:hardware:arm:raspberrypi3>

Last update: **2020/06/22 13:58 (UTC)**

