

Slackware ARM sobre una Raspberry Pi 4

La Raspberry Pi 4 fue lanzada el 24 de junio de 2019. Las características del dispositivo Broadcom BCM2711B0 SoC incorpora una CPU Quad-core ARMv8 Cortex-A72 [64 bit] @ 1.5GHz, VideoCore VI GPU @ 500MHz, y viene en tres variaciones diferentes con 1GB, 2GB, o 4GB LPDDR4 SDRAM @ 2400MHz instalada.

Esta computadora ARM de placa única revisada y mejorada sucede a la computadora [Raspberry Pi \(3\)](#), y es aproximadamente 2,5 veces más rápida y sin duda más potente. Wi-Fi y Bluetooth están incluidos sobre la placa. Tiene una verdadera placa de red ethernet Gigabit que tiene un controlador dedicado, el cual ya no está compartido con el bus USB. El dispositivo también cuenta con 2x USB 3.0 y 2x puertos USB 2.0. Aun no tiene reloj de tiempo real (RTC). Slackware ARM funciona perfectamente en este dispositivo, como es de esperar. El escritorio es mucho más rápido y sensible. Los tiempos de compilación se reducen a la mitad en comparación con el RPi3. La Raspberry Pi es apoyada fuera del árbol oficial de Slackware ARM por la comunidad de Slackware.

Slackware versión 14.2, -current

Slackware ARM puede ser instalado sobre una Raspberry Pi 4.

Siga los link (s) en la tabla de abajo. Estos son mantenidos por un autor separado como parte de la comunidad Slackware-sobre-Raspberry Pi.

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

AArch64 ARM64 [Experimental], Slackware ARM -current

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

Site	Versiones de Slackware	Usa paquetes oficiales de Slackware	Métodos de Instalación	Notas
Proyecto SARPi64	-current	Si	Instalador de Slackware	Un proyecto de desarrollo para Slackware ARM ejecutandose sobre AArch64 [ARMv8] kernel y módulos. Experimental en naturaleza y propósito.

Método de instalación manual sin una imagen de Raspbian

Siempre y cuando utilice la versión más reciente de firmware [es decir, después de junio de 2019] y la última imagen de Raspbian Buster [el método manual de instalación de la Raspberry Pi 1](#) también

funciona. Este método es para instalar Slackware ARM 14.2 sobre una Raspberry Pi 4 Modelo B sin una imagen de Raspbian. Aunque, esto podría trabajar para otro Slackware ARM y versiones de 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        6233951    31133184   83   Linux
$ sudo mkfs.vfat /dev/mmcblk0p1
$ sudo mkfs.ext4 /dev/mmcblk0p2
```

Observaciones:

- Utilizar una tarjeta SD de 32GB.
- Seleccione 32 MB para el tamaño de la primera partición.
- Dejar vacío dejado para la segunda partición.

2. Put the Raspberry Pi firmware in the SD Card

```
$ 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. Put the Slackware ARM mini root file system in the SD Card

```
$ 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
```


600MHz incluso si se usa al 100%. Puede verificar la frecuencia actual del procesador escribiendo:

```
$ cpufreq-info
```

Para alcanzar 1.2GHz cuando el procesador se usa al 100% (es decir, usar la escala de frecuencia), debe cambiar los gobernadores predeterminados. Agregue la siguiente línea al final del archivo `/etc/rc.d/rc.local` :

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

Ahora, el procesador está configurado correctamente.

5.2. Time

Desafortunadamente, la Raspberry Pi no proporciona un reloj en tiempo real (RTC). Es por eso que no se incluye batería con la placa. ¡Significa que cada vez que apaga la Raspberry Pi, el tiempo se reinicia! Sin embargo, si tiene acceso a Internet, puede actualizar el tiempo 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 configurado correctamente.

5.3. Video

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

En primer lugar, debe compilar Mesa ($\geq 17.0.4$) con el controlador VC4 DRI:

```
$ CFLAGS="-O2 -march=armv8-a -mtune=cortex-a53 -mfloat-abi=softfp -  
mfpu=neon-vfpv4" \  
  CXXFLAGS="-O2 -march=armv8-a -mtune=cortex-a53 -mfloat-abi=softfp -  
mfpu=neon-vfpv4" \  
  ./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, cree su propio paquete Slackware ARM Mesa e instálelo.

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

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

Luego reinicie la Raspberry Pi.

Puede verificar que puede obtener 60 FPS con aplicaciones OpenGL en la Raspberry Pi escribiendo el siguiente comando en un terminal X:

```
$ glxgears
```

Ahora el video está correctamente configurado.

Fuentes

- Originalmente escrita por [Exaga](#).
- Contribuciones por: [yugiohj](#).
- Traducción — [rramp](#) 2019/07/31 01:48 (UTC).

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

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

Permanent link:
<https://docs.slackware.com/es/howtos:hardware:arm:raspberrypi4>

Last update: **2021/01/02 15:31 (UTC)**

