

Slackware ARM no Raspberry Pi 3

O Raspberry Pi 3 tem um SoC Broadcom BCM2837 que incorpora um CPU ARMv8 Cortex-A53 quad-core [64 bit] a 1.2GHz e um GPU VideoCore IV a 400MHz, e vem com 1GB de SDRAM LPDDR2 a 900MHz. Este computador de placa única ARM revisado e atualizado tem êxito no [Raspberry Pi \(2\)](#), e é consideravelmente mais rápido e muito mais poderoso. Wi-Fi e Bluetooth agora estão incluídos. Embora ainda não tenha RTC. Pense no RPi3 como uma renovação e não como uma inovação. O Slackware ARM, como seria de esperar, funciona sem falhas neste dispositivo, com um aumento significativo de velocidade. Os tempos de compilação são muito mais curtos em comparação ao RPi2, por exemplo.

O Raspberry Pi 3 é suportado fora da árvore oficial do Slackware ARM pela comunidade do Slackware.

Slackware lança 14.2, -current

O Slackware ARM -current ou o Slackware ARM 14.2 podem ser instalados no Raspberry Pi 3 .

Siga o(s) link(s) na tabela abaixo. Estes são mantidos por um autor separado como parte da comunidade Slackware-on-Raspberry Pi.

Site	Versão Slackware	Usando pacotes oficiais do Slackware	Métodos de Instalação	Notas
Projeto SARPi	14.2,-current	Sim	Instalador Slackware	Um tutorial HOW-TO de ponta a ponta, conduzindo você pelo processo de instalação e configuração.

AArch64 ARM64 [Experimental], Slackware ARM -current

Experimental, desenvolvimento, e protótipo, Slackware AArch64 ARM64 link(s).

Site	Versão Slackware	Usando pacotes oficiais do Slackware	Métodos de Instalação	Notas
Projeto SARPi64	-current	Sim	Instalador Slackware	Um projeto de desenvolvimento para o Slackware ARM executando o kernel e módulos AArch64 [ARMv8]. Experimental na natureza e finalidade.

Método de Instalação Manual

Este método é para instalar o Slackware ARM 14.2 em um Raspberry Pi 3 Model B. No entanto, deve funcionar para outras versões do Slackware ARM e Raspberry Pi.

1. Particionar e formatar cartão memória 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
```

Observações:

- Use um cartão SD de 32 GB
- Selecione 32 MB para o tamanho da primeira partição
- Deixe o espaço vazio para a segunda partição

2. Coloque o firmware Raspberry Pi no cartão 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. Colocar o mini sistema de arquivos raiz do Slackware ARM no cartão SD

```
$ wget -c
ftp://ftp.arm.slackware.com/slackwarearm/slackwarearm-devtools/minirootfs/roots/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|\(root:\).*\(:16983:0:0:0:\)|\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
```

Observações:

- Definir password como password para o usuário root
- Utilizar DHCP na interface de rede eth1
- Permitir que o usuário root conecte-se através do SSH

4. Insira o cartão SD no Raspberry Pi

Seu cartão SD está pronto para que você possa inseri-lo no Raspberry Pi e iniciar.

Você pode se conectar remotamente ao seu Raspberry Pi como root através do SSH.

```
$ ssh root@raspberrypi
```

Assim que você estiver logado, você pode querer instalar pacotes adicionais do 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
```

Observações:

- Considere que o nome de host do Raspberry Pi é raspberrypi
- É recomendado adicionar um usuário normal usando o comando adduser e usar este usuário em vez de root
- É recomendado mudar a senha do usuário root usando o comando passwd
- É recomendado não permitir que o usuário root se conecte através do SSH editando o arquivo /etc/ssh/sshd_config
- É recomendado aos pacotes [build seu próprio kernel do Linux](#) porque o kernel que você está executando não combina com os pacotes do Slackware ARM instalados
- É recomendado mudar o nome do host slackware.localdomain editando o arquivo /etc/HOSTNAME

5. Truques e Dicas

5.1. Bluetooth

O Raspberry Pi tem um chip Broadcom fornecendo Bluetooth. No entanto, o firmware necessário não

está instalado no Slackware ARM. Isso significa que você precisa fazer o download e instalá-lo:

```
$ 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
```

Then build your own Slackware ARM bluez - firmware - brcm package and install it.

In order to enable Bluetooth you need to add the Bluetooth module, run the Bluetooth daemon, attach the device then open and initialize the device. Add the following lines to the end of the `/etc/rc.d/rc.local` file:

```
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
```

Observação: Às vezes, há uma falha com o comando `hciattach`, e é por isso que você precisa de um `while`.

Você pode verificar se o Bluetooth está funcionando digitando:

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

Agora, o Bluetooth está configurado corretamente.

5.2. Teclado

O mapa de teclado padrão no Slackware ARM é o do Reino Unido. Se você quiser carregar um outro mapa de teclado, edite o arquivo `/etc/rc.d/rc.keymap`.

Se você quiser mudar o layout do teclado para o X11, você precisa copiar o arquivo de configuração

do X11 e editá-lo:

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

Agora, o teclado está configurado corretamente.

5.3. Memória

Infelizmente, o Raspberry Pi tem apenas 1 GB de memória. Isso significa que você eventualmente não poderá executar alguns aplicativos que exigem mais memória. No entanto, você pode estender sua memória criando uma partição swap temporária de 4 GB:

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

Você pode verificar a memória atual digitando:

```
$ free
```

Agora, a memória está configurada corretamente.

5.4. Processador

O processador Raspberry Pi pode atingir 1,2 GHz. No entanto, por padrão, ele vem definido a 600 MHz, mesmo se for usado em 100%. Você pode verificar a frequência atual do processador, digitando:

```
$ cpufreq-info
```

Para atingir 1,2GHz quando o processador é usado a 100% (ou seja, use a escala de frequência), você precisa alterar os valores padrão. Adicione a seguinte linha ao final do arquivo `/etc/rc.d/rc.local`:

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

Agora, o processador está configurado corretamente.

5.5. Fuso Horário

Infelizmente, o Raspberry Pi não fornece um relógio de tempo real (RTC). É por isso que não há bateria incluída na placa. Isso significa que cada vez que você desligar o Raspberry Pi, o tempo é reiniciado! No entanto, se você tiver acesso à Internet, poderá atualizar o tempo durante a inicialização do Slackware ARM. Adicione a seguinte linha ao final do arquivo `/etc/rc.d/rc.local`:

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

Agora, a hora está correta.

5.6. Video

Infelizmente, o Raspberry Pi não é compatível com o OpenGL (ele é compatível com o OpenGL ES, que é um subconjunto do OpenGL). Isso significa que, por padrão, cada aplicativo que requer o OpenGL será lento. No entanto, você pode alcançar 60 FPS com aplicativos OpenGL no Raspberry Pi usando o driver correto.

Em primeiro lugar, você precisa construir o Mesa ($\geq 17.1.10$) com o driver 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
```

Em seguida, construa seu próprio pacote do Slackware ARM mesa e instale-o (você pode atualizar com segurança o fornecido pelo Slackware ARM).

Em segundo lugar, adicione a seguinte linha ao final do arquivo `/boot/config.txt`:

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

Depois reinicie o Raspberry Pi.

Você pode verificar se é capaz de obter 60 FPS com aplicativos OpenGL no Raspberry Pi digitando o seguinte comando em um terminal X11:

```
$ glxgears
```

Outro problema com o vídeo é que a resolução padrão é 1824×984. Isso significa que você pode ver bordas pretas na tela de 1920×1080. Para corrigir isso, adicione a seguinte linha ao final do arquivo `/boot/config.txt`:

```
disable_overscan=1
```

Depois reinicie o Raspberry Pi.

Você pode verificar se está usando uma resolução de 1920×1080 no Raspberry Pi digitando o

seguinte comando em um terminal X11:

```
$ xrandr
```

Agora, o vídeo está definido corretamente.

5.7. UART

No arquivo `/etc/inittab`, `s0` refere-se ao mini [UART](#) que está desativado por padrão no Raspberry Pi. Consequentemente, a cada 5 minutos, esta linha será gravada em `/dev/tty1` pelo processo `init`:

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

Uma solução fácil é editar o arquivo `/etc/inittab` e substituir a seguinte linha:

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

por esta:

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

Em seguida, reinicie o Raspberry Pi. Agora, o UART está definido corretamente.

Fontes

- Originally written by [Exaga](#)
- Contributions by [yugiohjcj](#)
- Translation PT-BR by [MacgyverPT \(Miguel Rosa\)](#)

[howtos](#), [hardware](#), [arm](#), [author exaga](#), [macgyverpt translated pt](#), [macgyverpt](#)

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

Permanent link:
<https://docs.slackware.com/pt-br:howtos:hardware:arm:raspberrypi3>

Last update: **2020/11/19 10:04 (UTC)**

