

Slackware ARM sur Raspberry Pi 4

Le Raspberry Pi 4 est sorti le 24 Juin 2019. L'appareil a pour SOC un processeur Broadcom BCM2711B0 qui intègre quatre cœurs ARMv8 Cortex-A72 [64 bit] CPU @ 1.5GHz, et un GPU VideoCore VI GPU @ 500MHz ; il est disponible sous trois configurations différentes avec des mémoires de 1 Go, 2 Go ou 4 Go LPDDR4 SDRAM @ 2400MHz. Cette carte révisée et mise à jour succède au [Raspberry Pi \(3\)](#); elle est approximativement 2,5 fois plus rapide et certainement plus puissante. Le Wi-Fi et le Bluetooth sont disponibles sur la carte. Elle possède un véritable port Gigabit Ethernet qui a un contrôleur dédié, qui n'est désormais plus partagé avec le bus USB. L'appareil offre également 2 ports USB3 et 2 ports USB2 mais encore pas de RTC. . Slackware ARM tourne de façon fluide sur cette carte comme vous pouvez l'attendre. Le bureau est beaucoup plus rapide et réactif. Les temps de compilation sont divisés par deux par rapport au RPI3.

Le Raspberry Pi 4 est supporté en dehors de la branche officielle de Slackware ARM par la communauté Slackware .

Slackware version current

Slackware ARM peut être installée sur un Raspberry Pi 4.

Suivez les liens de la table ci-dessous. Ils sont maintenus par un auteur séparé faisant partie de la communauté Slackware sur Raspberry.

Site	Versions de Slackware	Utilisation des paquetages Slackware officiels	Méthodes d'installation	Notes
SARPi Project	current	Oui	Installateur Slackware	Un guide complet pour vous expliquer pas à pas comment installer et configurer votre système.

AArch64 ARM64 [Expérimental], Slackware ARM -current

Expérimental, en développement, et prototype, Slackware AArch64 ARM64 link(s).

Site	Slackware versions	Utilisation des paquets Slackware	Méthode d'Installation	Notes
SARPi64 Project	-current	Oui	Installateur Slackware	Un projet de développement pour Slackware ARM tournant le noyau et les modules AArch64 [ARMv8] . Expérimental par nature et par but .

Installation manuelle sans image Raspian

Tant que vous utilisez la version la plus récente du firmware [i.e. postérieure à Juin 2019] et la dernière image Raspian Buster, la [méthode d'installation manuelle du Raspberry Pi 1](#) marche aussi pour le Pi 4.

1. Partitionnement et formatage de la carte 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
```

Remarques:

- J'utilise une carte SD de 32 Go
- Je choisis 32 Mo comme taille de la première partition
- Je laisse tout l'espace vide pour la seconde partition

2. Mettre le firmware du Raspberry Pi sur la carte 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. Mettre le filesystem mini root de Slackware ARM sur la carte 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|\(root:\).*\(:16983:0:0:0:0\)|\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
```

Remarques:

- Je mets "password" comme mot de passe pour l'utilisateur "root"
- J'active DHCP sur l'interface réseau "eth1"
- J'autorise l'utilisateur "root" à se connecter par SSH

4. Insérer la carte SD dans le Raspberry Pi

Votre carte SD est prête et vous pouvez l'insérer dans le Raspberry Pi et le démarrer.

Vous pouvez vous connecter à distance sur votre Raspberry Pi par SSH en tant que "root".

```
$ ssh root@raspberrypi
```

Dès que vous êtes connecté, vous pouvez installer les paquetages additionnels 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
```

- Je considère que nom d'hôte du Raspberry Pi est "raspberrypi"
- Je recommande de choisir un utilisateur normal plutôt que "root"
- Je recommande de changer le mot de passe de "root"
- Je recommande d'interdire à "root" de se connecter par ssh
- Je recommande de [créer vos propres paquetages du noyau Linux](#) car le noyau que vous utilisez ne coïncide pas avec les paquetages installés de Slackware ARM

5. Trucs et astuces

5.1. Processeur

Le processeur du Raspberry Pi peut atteindre 1.2GHz. Cependant, par défaut, il est bridé à 600MHz même s'il est utilisé à 100 %. Vous pouvez vérifier la fréquence courante du processeur en tapant:

```
$ cpufreq-info
```

Pour atteindre 1.2GHz quand le processeur est utilisé à 100 % (autrement dit pour faire de l'adaptation de fréquence), il faut changer les "gouverneurs" par défaut. Ajoutez cette ligne à la fin

du fichier `/etc/rc.d/rc.local`:

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

Maintenant le processeur est correctement configuré.

5.2. Temps

Malheureusement, le Raspberry Pi ne possède pas d'horloge temps réel (RTC). C'est la raison pour laquelle il n'y a pas de pile sur la carte mère. Cela veut dire que chaque fois que vous éteignez le Raspberry Pi, l'horloge est remise à 0! Cependant, si vous un accès Internet, vous pouvez mettre l'horloge à l'heure durant le démarrage de Slackware ARM. Ajoutez la ligne suivante à la fin du fichier `/etc/rc.d/rc.local`:

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

Maintenant, l'horloge est à la bonne heure.

5.3. Video

Malheureusement, le Raspberry Pi n'est pas compatible OpenGL (il est compatible OpenGL ES qui est un sous-ensemble d'OpenGL). Cela veut dire que, par défaut, toutes les applications qui ont besoin d'OpenGL seront lentes. Cependant, vous pouvez atteindre 60 images par seconde pour des applications OpenGL sur le Raspberry Pi en utilisant le bon pilote.

En premier, il faut construire Mesa ($\geq 17.0.4$) avec le pilote 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
```

Ensuite, fabriquez votre propre paquetage Mesa pour Slackware ARM et installez le.

Deuxièmement, ajoutez la ligne suivante à la fin du fichier `/boot/config.txt`:

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

Puis redémarrez le Raspberry Pi. Vous pouvez vérifier que vous pouvez atteindre les 60 images par seconde pour des applications OpenGL sur le Raspberry Pi en tapant la commande suivante dans un terminal X:

```
$ glxgears
```

Désormais la vidéo est correctement configurée.

Sources

- Version originale [Exaga](#)
- Traduction française [Desiderius](#)

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

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

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

Last update: **2020/02/21 15:09 (UTC)**

