Slackware ARM on the Raspberry Pi 1

Since there are so many ARM devices coming on to the market, it is not possible to provide support for them all in the main tree.

The Raspberry Pi is supported outside of the official Slackware ARM tree by the Slackware community.

Slackware releases 13.37, 14.0, 14.2

Since the release of Slackware ARM 14.0, there have been a number of community efforts to bring Slackware to the device:

Slackware ARM 14.2 is the only available version of Slackware that is officially supported/maintained, that runs on the Raspberry Pi 1. Releases of Slackware ARM greater than version 14.2 are not backwards compatible, since they moved to a hard floating point ABI and has a minimum CPU requirement of ARMv7. The Raspberry Pi 1 only has ARMv6 architecture.

You should follow one of the links in the table below. Each is maintained by a separate author as part of the Slackware-on-Raspberry Pi community.

Site	Slackware versions	Using official Slackware packages	Installation methods	Notes
SARPi Project	14.2	Yes	Slackware installer	An end-to-end HOW TO tutorial taking you through the installation and setup process.
Stanley Garvey	14.0	Yes	Slackware installer & pre-made images	Pre-made installed OS images ready to copy to an SD card
Dave's Collective	13.37	Yes	Slackware installer	An excellent set of instructions in order to have Slackware ARM running on your Raspberry Pi.

Manual installation method

Although the community does its best to keep up with the hardware changes there may be times when the above notes and images are unusable. If this happens you may work around the problem by using a miniroot image and a functional boot partition from some other source (like borrowing them from rasbian). If the kernel is the only issue you can compile your own kernel from sources (see here for a guide on doing that http://elinux.org/RPi_Kernel_Compilation).

Here are the steps involved in setting up a minimal Slackware ARM system from a miniroot image:

Download the current stable raspbian image from http://www.raspberrypi.org/downloads Unzip it and mount the partitions therein via loopback and then put all that is needed in a tarball for later use:

```
root@darkstar:/tmp# fdisk -l 2016-05-10-raspbian-jessie-lite.img

Disk 2016-05-10-raspbian-jessie-lite.img: 1.3 GiB, 1386217472 bytes, 2707456
```

```
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
Disklabel type: dos
Disk identifier: 0x84f9d19f

Device                                Boot  Start     End Sectors  Size Id
Type
2016-05-10-raspbian-jessie-lite.img1        8192  137215  129024   63M  c
W95 FAT
2016-05-10-raspbian-jessie-lite.img2      137216 2707455 2570240  1.2G 83
Linux

root@darkstar:/tmp# losetup -o $((8192 * 512)) /dev/loop0 2016-05-10-
raspbian-jessie-lite.img
root@darkstar:/tmp# losetup -o $((137216 * 512)) /dev/loop1 2016-05-10-
raspbian-jessie-lite.img
root@darkstar:/tmp# mount -o ro /dev/loop1 /mnt/floppy/
root@darkstar:/tmp# mount -o ro /dev/loop0 /mnt/floppy/boot
root@darkstar:/tmp# cd /mnt/floppy/
root@darkstar:/mnt/hd# tar vcpzf /tmp/raspbian_boot_stuff.tgz boot
lib/modules/ lib/firmware opt/vc
```

Please note the sectors of the beginning of the partitions: 8192 and 137216. We need to multiply these by 512 to get the byte offset for the loop device setup. This is done by $((8192 * 512)) and $((137216 * 512)). You will need to change these if the image partitioning scheme changes.
Now partition and format an SD like this: (NB the “fdisk -l” is just to show how I partitioned my SD)

```
root@darkstar:~# fdisk  -l -u  /dev/sde

Disk /dev/sde: 4093 MB, 4093640704 bytes
126 heads, 62 sectors/track, 1023 cylinders, total 7995392 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: 0xd0b5414a

   Device Boot      Start         End      Blocks   Id  System
/dev/sde1            2048      133119       65536    c  W95 FAT32 (LBA)
/dev/sde2          133120     7995391     3931136   83  Linux
root@darkstar:~# mkdosfs -F 16 /dev/sde1
root@darkstar:~# mke2fs -t ext4 -b 4096 -i 16384 -m 0  -L root /dev/sde2
root@darkstar:~# mount -o noatime /dev/sde2 /mnt/hd/
root@darkstar:~# mkdir /mnt/hd/boot
root@darkstar:~# mount -o noatime /dev/sde1 /mnt/hd/boot/
```

It's not a typo I got a bad headache figuring out why it did not work: the boot partition is to me made with id “c” but such small partitions have issues when you try to make a fat32 filesystem on them, you will get an error lamenting some issue with insufficient clusters but some sort of filesystem is made and if you ignore that and proceed you end up with something that does not boot. What you

need to do is actually tell mkdosfs to make a fat16 filesystem and then things start to work right.

Now you can extract the Slackware ARM miniroot and then the raspbian_boot_stuff.tgz in /mnt/hd.
Edit the /mnt/hd/boot/cmdline.txt and add at the end “ro” and check that the root parameter matches the partitioning of the SD.
Edit the fstab to match your formatting (if that was like I suggested it will look like this:)

```
root@darkstar:/mnt/hd/etc# cat fstab
proc            /proc           proc    defaults          0       0
/dev/mmcblk0p1  /boot           vfat    errors=remount-ro          0       2
/dev/mmcblk0p2  /               ext4    errors=remount-ro,noatime  0       1
root@darkstar:/mnt/hd/etc#
```

You can now umount the SD, insert it into the RasbberyPI and boot into your Slackware ARM miniroot to add whatever else you need.
I generally add whatever else I need by simply using wget to pull down slackpkg, installing manually the downloaded slackpkg, editing the mirrors file and then install the rest that's needed with slackpkg itself (internet connection is required for this).
You might want to edit or comment the serial console in inittab to suppress the “s0” respawning to fast message.

Incidentally if you download a recent version of raspbian this procedure will create bootable images for the RPi, RPi 2, RPi 3, and RPi Zero.

Sources

- Originally written by Stuart Winter
- Contributions by louigi600 , yugiohjcj , streamthreader

howtos, hardware, arm, author mozes