



Tuesday, 24 July 2012

## Creating a loopback file-system image with partitions

With building the OpenELEC images for Raspberry Pi - I ran into an odd problem with creating system images for people to download.

The first image I made was DD'd from an 8GB SD card that I had already created and tested. I formatted the SD card, tested in the Pi and then took an image with DD. Then, rewrote it to the card and tested again. All good. Then a few people commented that the image was slightly too large to fit on other 8GB cards which is when I checked and my card it was indeed a few sectors large than some others. So how do you create images which are useful for more people without going out and buying an army of various sized SD cards? The answer is creating loopback devices which are basically file-systems in a single file (which is what you create when you DD a working image from an SD card, for example).

1. Decide what size you would like your image to be. For this example and the OpenELEC image I posted previously, I used 1GB.
2. Create your empty container file using DD. You can replace the expression after seek with 1024000000 if you prefer - this is the size of your image in bytes.

```
dd if=/dev/zero of=/lgb_file_image.img bs=1024 count=0 seek=$((1000*1000))
```

3. Mount this device as a loopback device

```
sudo losetup /dev/loop0 ~/lgb_file_image.img
```

4. This will give you a loopback block device under /dev/loop0. You can now create your partitions using fdisk / parted. I've used the examples from the [OpenELEC wiki](#) here.

```
sudo parted -s /dev/loop0 mklabel msdos
sudo parted -s /dev/loop0 unit cyl mkpart primary fat32 -- 0 16
sudo parted -s /dev/loop0 set 1 boot on
sudo parted -s /dev/loop0 unit cyl mkpart primary ext2 -- 16 -2
sudo mkfs.vfat -n System /dev/loop0p1
sudo mkfs.ext4 -L Storage /dev/loop0p2
```

5. Now you have a filesystem in a file with two partitions. Things can get a little tricky here when you need to mount them individually. You need to umount the current loopback device so you can remount it with some sector offsets and size limits. If you search and go by the scraps of mailing lists that mention this, it can be a little daunting but it's not as bad as it seems. Lets look at an fdisk output of our new filesystem in a file.

```
sudo fdisk -l ~/lgb_file_image
Disk lgb_file_image.img: 1024 MB, 1024000000 bytes
255 heads, 63 sectors/track, 124 cylinders, total 2000000 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: 0xd00251a6

Device            Boot Start      End    Blocks Id System
lgb_file_image.img1 *    2048    264191  131072 c   W95 FAT32 (LBA)
lgb_file_image.img2      264192  1999999  867904 83   Linux
```

What we need to note here are the start and end sectors as we will use these to calculate the offset and sizelimit parameters to pass to losetup when mounting the partitions in the file. Taking the first partition as an example (shown as an actual file with a different number at the end, that's just how fdisk rolls I guess). We have a starting sector of **2048** and an end sector of **264191**. We need to multiply each of these by **512** to get the total amount of bytes instead of sectors (bytes per sector is in the output - line 4 above). So, our starting offset becomes 1048576, and the size being (end - start, multiplied by 512 again) 134217216. We'll add these to our next command.

6. Mount the loopback image file again, specifying the offset and sizelimit for the first partition.

### ROOT9.net

Technical resources, code, troubleshooting and possibly the occasional rant.

### Search This Blog



### Blog Archive

- [2016](#) (2)
- [2014](#) (3)
- [2013](#) (7)
- ▼ [2012](#) (17)
  - [November](#) (3)
  - [October](#) (2)
  - [September](#) (1)
  - [August](#) (1)
  - ▼ [July](#) (3)
    - [Creating a loopback file-system image with parti...](#)
    - [OpenELEC Build R11639 available](#)
    - [Puppet - 'host name was not match with the server'...](#)
- [May](#) (4)
- [February](#) (3)

### Twitter

```
sudo losetup /dev/loop0 ~/1gb_image_file.img -o 1048576 --sizelimit 134217216
```

7. Now, you can mount this loopback device as a block device with the standard mount command

```
sudo mount /dev/loop0 /mnt/my_mount_point
```

This should give you the first FAT partition (if you used the OpenELEC example anyway) mounted under /mnt/my\_mount\_point (make sure it exists first!). Then you can copy files into your loopback device to populate your image. To mount the second partition, follow the same procedure as in steps 5-7 but use a new loopback device (/dev/loop1 is a good starting point) and calculate the new offsets and sizelimits based on the start and end sectors of the second partition. When you're finished, make sure to unmount the loop devices, then detach the loopback devices with

```
sudo losetup -d /dev/loop0
sudo losetup -d /dev/loop1
```

It can be a finicky procedure, but once you've got it nailed it's pretty handy. Let me know if any parts don't work as advertised (especially seeing as it took several attempts to get mine right too!).

Posted by **Stuart** at 10:47



+3 Recommend this on Google

Labels: [card](#), [dd](#), [device](#), [filesystem image](#), [image](#), [linux](#), [loopback](#), [sd](#)

## 9 comments:

**blog** 1 September 2012 at 12:40

Following your guidance here I got a modified version of the create\_sdcard script working with the loop0 device. Having moved it to another machine I now find that although fdisk shows /dev/loop0p1 and /dev/loop0p2 after partitioning they're not appearing in /dev (even after doing partprobe /dev/loop0). Do you have any ideas why this might be happening?

I think I can fix things by using offsets in losetup and then formatting the whole device (rather than just a partition), but that seems messy.

[Reply](#)

[Replies](#)

**blog** 1 September 2012 at 14:54

I'm starting to suspect that this is down to the device handling on the VM service provider I'm using, and I've worked around this by creating an empty partitioned and formatted image on a machine that does work properly then using that for the later stages of the image creation process.



**Stuart** 2 September 2012 at 15:34

Hey - I had a few problems with partitions not appearing too and I think it was mostly down to certain tools not recognising the partitions on a loopback device (ie they only read the first 512bytes). I gave in and used the offsets which is a bit of a faff but seems to be more robust.



**Agócs** 26 August 2013 at 17:23

Hello folks,

I'm facing the same problem, missing the partition devices :-/ So I can't make an ext4 to my loop partiitions, because mkfs can't find the dev. Can you describe me how you managed this issue using the offsets? Thanks in advance!



**Agócs** 26 August 2013 at 21:10

Ok, I found the solution: 'kpartx' does the mapping for the partitions to /dev/mapped/loop0pX devices. I learned something new today :)

## Tweets by @sg\_84



**Stuart Green**

@sg\_84

So today's first training module is the easy-going topic of Orthogonal frequency-division multiplexing. #notenoughcoffee

12 Sep



**Stuart Green**

@sg\_84

thanks for the XSS vulnerability #Tweetdeck, and special thanks to @BBCBreaking for retweeting it to millions of people...

11 Jun

Stuart Green Retweeted



**Ars Technica**

@arstechnica

"TrueCrypt is not secure" official SourceForge page abruptly warns ars.to/1IRHly5 by @dangoodin001

"TrueCrypt is not secure," ...  
Support for decade-old crypto...  
arstechnica.com

[Embed](#)

[View on Twitter](#)

**Stuart** 31 August 2013 at 19:49

Sounds interesting, will have to check that out. Thanks for sharing!

[Reply](#)**Jim Moyle** 27 June 2013 at 23:24

Unfortunately when using `sudo parted -s /dev/loop0 unit cyl mkpart primary fat32 -- 0 16` I get the following error:

Error: Error informing the kernel about modifications to partition /dev/loop0 -- Invalid argument. This means Linux won't know about any changes you made to /dev/loop0 until you reboot -- so you shouldn't mount it or use it in any way before rebooting.

I get a similar result with `fdisk`, is this what you mean by partitions not appearing?

[Reply](#)**Stuart** 28 June 2013 at 19:28

It sounds familiar yes, have you tried running `'sudo partprobe /dev/loop0'` to update manually?

[Reply](#)**amias** 3 November 2015 at 11:44

if you use the `-p` option to `losetup` it will make your loopback device partitionable without the reboot warning.

From `losetup --help`

`-P, --partscan` create a partitioned loop device

[Reply](#)

Enter your comment...

Comment as: Unknown (Goog ↕)

[Sign out](#)

[Publish](#)

[Preview](#)

☐ Notify me

[Newer Post](#)[Home](#)[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)