



## OSDev.org

The Place to Start for Operating System Developers

Login Register

**The OSDev.org Wiki - Got a question? Search this first!** FAQ Search

It is currently Wed Mar 04, 2015 12:50 pm

[View unanswered posts](#) | [View active topics](#)

[Board index](#) » [Operating System Development](#) » [OS Development](#)

All times are UTC - 6 hours

### virtual floppy on linux

**Moderators:** [mystran](#), [JAAMAN](#), [Owen](#), [Combuster](#), [quok](#), [os64dev](#), [Candy](#), [chase](#), [pcmattman](#), [sortie](#), [AJ](#), [Brendan](#), [01000101](#), [carbonBased](#), [Antti](#), [thepowersgang](#), [JamesM](#), [kmcguire](#)



Page 1 of 1 [ 9 posts ]

[Print view](#)

[Previous topic](#) | [Next topic](#)

#### Author

#### Message

**Megan65**

**Post subject:** virtual floppy on linux

**Posted:** Fri Feb 19, 2010 10:15 am



hi

Member



how can i create an virtual floppy disk in linux and use it like as a normal disk (like vfd in windows)?

**Joined:** Fri Oct 24, 2008

12:34 pm

**Posts:** 48

[Top](#)



**mutex**

**Post subject:** Re: virtual floppy on linux

**Posted:** Fri Feb 19, 2010 10:28 am



Create a image file with requested size using dd:

Member

dd if=/dev/zero of=/tmp/floppy.img bs=1k count=1440



**Joined:** Sat Jul 07, 2007 7:49 pm  
**Posts:** 127

Mount it using loop device;

```
losetup /dev/loop5 /tmp/floppy.img
```

Setup partition info using fdisk;

```
fdisk /dev/loop5
```

Format drive;

```
mkfs -t <fstype> /dev/loop5
```

Mount it to local filesystem;

```
mount /dev/loop5 /tmp/myfloppymount
```

If you are using existing image you can skip fdisk and mkfs...

Good luck!

-

Thomas

[Top](#)



**Megan65**

**Post subject:** Re: virtual floppy on linux

**Posted:** Fri Feb 19, 2010 10:42 am



Member



**Joined:** Fri Oct 24, 2008 12:34 pm  
**Posts:** 48

hi

fdisk /dev/loop5 , dosent work and return this message:

**Quote:**

Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel  
Building a new DOS disklabel with disk identifier 0x8251a9fd.

Changes will remain in memory only, until you decide to write them.  
After that, of course, the previous content won't be recoverable.

You must set cylinders.

You can do this from the extra functions menu.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

[Top](#)



**XanClic**

**Post subject:** Re: virtual floppy on linux

**Posted:** Fri Feb 19, 2010 10:47 am

offline

Member



**Joined:** Wed Feb 13, 2008  
9:38 am  
**Posts:** 138



[Top](#)

**StephanvanSchaik**

**Post subject:** Re: virtual floppy on linux

**Posted:** Fri Feb 19, 2010 11:23 am

offline

Member



**Joined:** Sat Sep 29, 2007  
5:43 pm  
**Posts:** 114  
**Location:** Sittard, The  
Netherlands

Hi,

The wiki already has information about this (not even including the pages about GRUB):

[http://wiki.osdev.org/Disk\\_Images](http://wiki.osdev.org/Disk_Images)

[http://wiki.osdev.org/Loopback\\_Device](http://wiki.osdev.org/Loopback_Device)

To get a virtual floppy drive formatted as FAT12 you can simply do something like:

**Code:**

```
dd if=/dev/zero of=floppy.img bs=512 count=2880
losetup /dev/loop0 floppy.img
mkdosfs /dev/loop0
mount -t vfat /dev/loop0 /media/floppy
```

Unmounting it can be done using:

**Code:**

```
umount /dev/loop0
losetup -d /dev/loop0
```

Regards,  
Stephan J.R. van Schaik.

[Top](#)**SpawnHappyJake****Post subject:** Simply Mounting vs. Emulation**Posted:** Tue May 22, 2012 12:33 am**offline**

**Joined:** Mon May 21, 2012  
10:06 pm  
**Posts:** 3

Be mindful that such mounting of floppy images does not emulate a floppy drive. VFD in Windows does emulate a floppy drive.

I know this thread is a little old, but I've been doing some "Internet clean-up" recently concerning this very subject. Been meaning to comment here on this for a while.

It's important to realise that a plain vanilla Linux kernel does not have the ability to emulate hardware. A plain vanilla Linux kernel does have the ability to take an image of a filesystem and mount it.

You can image any filesystem with dd and mount it in Linux. All this is doing is having the files in that filesystem image displayed in the directory where you mount the image to. It does not emulate hardware. So you cannot

"create an virtual floppy disk in linux and use it like as a normal disk (like vfd in windows)". There would be nothing "floppy" about it. It would just be a filesystem mounted at a directory. It is not a "virtual floppy disk".

Pull up a Linux terminal and do the following: "sudo lshw | more". Or even better, "sudo lshw > \$HOME/Documents/my\_hardware.txt", then view my\_hardware.txt.

Run lshw before and after mounting a floppy image with the mount command. The mount command will have absolutely no effect on the hardware list. You are not emulating a floppy drive, or else it would show up in the hardware list.

Now get an iso image of a CD or DVD. Check lshw before and after mounting it with the mount command. Again, no emulation of hardware.

Now install CDEmu and mount the iso with CDEmu. Now check lshw. Ah. Now it's in the hardware list. Now you're emulating hardware.

You can do something parallel to this in Windows. Instead of lshw, you can open device manager. Try mounting an iso with PowerIso (viewing device manager before and after) and also try mounting an iso with anything else (viewing device manager before and after). PowerIso does not truly emulate an optical drive (doesn't produce anything in device manager). This is the only iso mounting software for Windows that I know of that doesn't produce an optical drive in device manager. BTW, here are more reasons why PowerIso is really weird: mounting an iso with PowerIso makes iTunes think a blank disc has been inserted, and mounting a floppy image with PowerIso shows up in the File Browser with an optical disc icon, if I remember right.

VFD in Windows does add a floppy drive in device manager.

I think the aforementioned wikis concerning loopback devices and image mounting need a little tweaking, partially because this lack of emulation needs to be brought to light.

The loopback device entry starts out like this:

**Quote:**

A Loopback Device is a mechanism used to interpret files as real devices. The main advantage of this method is that all tools used on real disks can be used with a loopback device.

"Real devices" is a little misleading here. If it just said "device" and not "real device", then I could see the wiki meaning "device" in the \*nix sense of the term (like loopback "device", block "device", character "device", etc. You know, abstraction files.). But "real device" definitely would make the average person think "hardware". Again, a loopback device does not emulate hardware.

**Quote:**

The main advantage of this method is that all tools used on real disks can be used with a loopback device.

Not true. A loopback device is only a mounted filesystem. Thus one loopback device cannot include more than one partition. Anything outside of the one mounted filesystem is not in the loopback device. This means no partitioning scheme, no MBR, no GPT, no no-man's land between the MBR and the first partition, none of that. Things like fdisk, gdisk, and gParted aren't going to help you here. If you want to make an MBR or GPT schemed raw image with GRUB2 installed to the MBR+, you might want to use something like qemu-nbd instead of a loopback device. But I have found qemu-nbd to be unstable, and a hypervisor seems to be the next choice since I don't know of any hard drive emulators for Linux, or even anything other than qemu-nbd that creates an abstraction file for a "whole hard drive image", without the actual emulation of hardware in Linux. And of course, low level formatting tools and any tools that have to do with hard drive firmware or the settings thereof is out of the question with a loopback device.

As you OS Devs know, only the kernel has the privileges necessary to see hardware, so all non-kernel programs only know anything about hardware from the information the kernel feeds them. Thus hardware can be emulated simply by the kernel lying and saying there is this (non-existent) hardware. That is what CDEmu does via a Linux kernel module.

Of course there is more to emulating hardware than just showing up in the hardware list, and CDEmu does quite a few things, such as responding to certain protocols, such as the SCSI command set and whatever the Raw DAO 96 protocol is.

I imagine that to emulate a floppy drive, there would be at least a few protocols that need to be supported (which certainly aren't supported by a loopback device).

I have searched and not found a single floppy drive emulator for Linux. However, mounting a floppy image with the mount command is often sufficient for what people are trying to do.

Cheers,  
Jake

[Top](#)**Kevin****Post subject:** Re: Simply Mounting vs. Emulation**Posted:** Tue May 22, 2012 3:20 am**offline**

Member

**Joined:** Sun Feb 01, 2009

6:11 am

**Posts:** 700**Location:** Germany**SpawnHappyJake wrote:**

A loopback device is only a mounted filesystem.

Sorry, but this is totally wrong. A loopback device is not a file system, but a block device; more precisely, a file mapped to a block device.

A block device (no matter if it's called loop0 or sda) can *contain* a file system. It can also contain a partition table. It can contain anything. Using fdisk on a loop device *is* meaningful and if you want to create a partitioned image, it is what you should do.

Now if you have a partitioned image and you want to actually mount it or create a file system on the image, what you need is block devices for the individual partitions. You can either do that manually by using the --offset option of losetup, or you can use kpartx. I recommend the latter.

qemu-nbd does have its uses, but they are really related to non-raw file formats, say if you want to mount a qcow2 or a VMDK image.

---

Developer of [tyndur](#) - community OS of [Lowlevel](#) (German)

[Top](#)**SpawnHappyJake****Post subject:** Re: virtual floppy on linux**Posted:** Fri May 25, 2012 10:55 pm**offline****Joined:** Mon May 21, 2012

10:06 pm

**Posts:** 3**Quote:**

Sorry, but this is totally wrong. A loopback device is not a file system, but a block device; more precisely, a file mapped to a block device.

Ok, I see. Thanks. Sorry about that. My world has been embettered. After looking at that wiki again, seeing the losetup command opened new horizons for me. I shall play around with that.

Cheers,  
Jake

[Top](#)



**SpawnHappyJake**

**Post subject:** Re: virtual floppy on linux

**Posted:** Sat May 26, 2012 12:49 am

**offline**

**Joined:** Mon May 21, 2012  
10:06 pm  
**Posts:** 3

Thanks again Kevin! Boy I'm a happy camper! I made a 1 GiB image with dd, and used losetup to connect it to /dev/loop0.

Then I used gdisk (like fdisk, but does gpt. It's really cool. I like that it rounds to 2048-sector boundaries for you. (one 2048-sector boundary = 1 MiB)) to partition it. I could have sicked gdisk directly to the image, but for fun I aimed it at /dev/loop0 (since that is what I was playing with), which works just as well. I made 4 partitions: a storage partition, an EFI System partition, a BIOS boot partition, and a second storage partition.

After scheming the image as GPT and making partitions, the kernel did not auto-create additional abstraction files for each partition, not even after detaching and reattaching the image to /dev/loop0.

But I haven't used all my free loopback abstraction files according to "mount" and "ls /dev". So I hooked the first storage partition to /dev/loop1 and the second storage partition to /dev/loop2. I was careful to use the -o (offset) and --sizelimit options just right to only attach the partitions, nothing more, nothing less, to the abstraction files.

I was then able to use mkfs.ext3 on them to format the partitions to ext3.

Then I could mount them like this (after making /media/play1 and /media/play2 with "sudo mkdir"):

```
sudo mount -o users /dev/loop1 /media/play1
sudo mount -o users /dev/loop2 /media/play2
```

The "-o users" allowing all users to mount and unmount them. But I couldn't read or write to them without root privileges. Thanks to Mr. Willimason over here: <http://superuser.com/questions/174776/modify-fstab-entry-so-all-users-can-read-and-write-to-an-ext4-volume>, that was a short-lived problem:



```
sudo chmod 777 /media/play1  
sudo chmod 777 /media/play2
```

Yeah, works great! I'm don't need qemu-ndb for raw images anymore! That's a relief. I think next I'll try installing grub2 to it and seeing if I can boot the ArchBang installer in QEMU. There's a bug in the version of QEMU I have that makes it get hung-up on the ISOLINUX bootloader, so I can't just boot the iso. I tried VirtualBox, but the VM crashes every time I run rankmirrors. Running in QEMU will tell me if that's a VirtualBox bug or a bug somewhere in ArchBang. Tried compiling the latest released tarball of QEMU as well as QEMU out of the git, but got make errors, even with --disable-werror. Oh well. This losetup stuff is a big help.

Thanks again,  
Jake

[Top](#)

Display posts from previous: All posts ▼ Sort by Post time ▼ Ascending ▼ Go



Page 1 of 1 [ 9 posts ]

[Board index](#) » [Operating System Development](#) » [OS Development](#)

All times are UTC - 6 hours

### Who is online

Users browsing this forum: No registered users and 6 guests

You **cannot** post new topics in this forum  
You **cannot** reply to topics in this forum  
You **cannot** edit your posts in this forum  
You **cannot** delete your posts in this forum  
You **cannot** post attachments in this forum

Search for:  Go

Jump to: OS Development ▼ Go

Powered by [phpBB](#) © 2000, 2002, 2005, 2007 phpBB Group