



major.io

Words of wisdom from a  
systems engineer

Who am I?  
icanhazip FAQ  
Résumé  
Keybase  
RSS

Search this website ...



# Mounting a raw partition file made with dd or dd\_rescue in Linux

December 14, 2010 By Major Hayden — 43 Comments

This situation might not affect everyone, but it struck me today and left me scratching my head. Consider a situation where you need to clone one drive to another with `dd` or when a hard drive is failing badly and you use `dd_rescue` to salvage whatever data you can.

Let's say you cloned data from a drive using something like this:

```
# dd if=/dev/sda of=/mnt/nfs/backup/harddrive.img
```

Once that's finished, you should end up with your partition table as well as the grub data from the MBR in your image file. If you run `file` against the image file you made, you should see something like this:

```
# file harddrive.img
harddrive.img: x86 boot sector; GRand Unified Bootloader, stage1 version 0x3, stage2
address 0x2000, stage2 segment 0x200, GRUB version 0.97; partition 1: ID=0x83,
active, starthead 1, startsector 63, 33640047 sectors, code offset 0x48
```

What if you want to pull some files from this image without writing it out to another disk? Mounting it like a loop file isn't going to work:

```
# mount harddrive /mnt/temp
mount: you must specify the filesystem type
```

The key is to mount the file with an offset specified. In the output from `file`, there is a particular portion of the output that will help you:

```
... startsector 63 ...
```

This means that the filesystem itself starts on sector 63. You can also view this with `fdisk -l`:

```
# fdisk -l harddrive.img
```

	Device	Boot	Start	End	Blocks	Id	System
harddrive.img		*	63	33640109	16820023+	83	Linux

Since we need to scoot 63 sectors ahead, and each sector is 512 bytes long, we need to use an offset of 32,256 bytes. Fire up the mount command and you'll be on your way:

```
# mount -o ro,loop,offset=32256 harddrive.img /mnt/loop
# mount | grep harddrive.img
/root/harddrive.img on /mnt/loop type ext3 (ro,loop=/dev/loop1,offset=32256)
```

If you made this image under duress (due to a failing drive or other emergency), you might have to check and repair the filesystem first. Doing that is easy if you make a loop device:

```
# losetup --offset 32256 /dev/loop2 harddrive.img
# fsck /dev/loop2
```

Once that's complete, you can save some time and mount the loop device directly:

```
# mount /dev/loop2 /mnt/loop
```

Share this post:



Tagged With: command line, emergency, sysadmin

43 Comments major.io

Login

Recommend 2 Share

Sort by Best



Join the discussion...



**David Rios** • a year ago

recent versions of losetup have a parameter -P which automatically create a loop device with all partitions. so for the example in the article you could do:

```
# losetup -P /dev/loop2 harddrive.img
```

and then you could access the first partition on /dev/loop2p1, the second on /dev/loop2p2 and so on.

2 ^ | v • Reply • Share



**Jamal Kumar** → David Rios • 10 months ago

Thanks so much, this offset thing wasn't working despite verifying that it was the same blocksize and everything

^ | v • Reply • Share



**fredwillcutt** → Jamal Kumar • 8 months ago

weird. had the same problem (gspilz)

^ | v • Reply • Share



**Joey** • 6 months ago

There's a couple of experimental fuse plugins on github for mounting partitions.

I've put one how here which will also mention the alternatives in the readme...

<https://github.com/joeyhub/fus...>

Sorry for the self promotion :).

1 ^ | v • Reply • Share



**Stephan** • 6 years ago

For partitioned LV's I always use kpartx which does the weird calculations. I assume this also works for flat files (never tried it, but it's nearly the same from systems point of view).

Say you have /dev/someVG/PartitionedLV

```
kpartx -a /dev/someVG/PartitionedLV
```

```
mount /dev/mapper/someVG--PartitionedLV1 /mnt/firstPartition
```

```
mount /dev/mapper/someVG--PartitionedLV2 /mnt/secondPartition
```

After unmounting the Partitions, kpartx -d /dev/someVG/PartitionedLV removes the entries from DM.

1 ^ | v • Reply • Share



**skyrail02** • a day ago

Appreciate your detailed explanation.

But I went to kpartx method first, and I must admit it's much easier to apply.

Besides iosetup command does not seem to work the same as yours under debian wheezy on armel device.

Thats is a bit strange for me.

^ | v • Reply • Share

**Pavel Coelho** • 15 days ago

hi there, when mouse-over on code snippets the first line of code gets covered by the code container itself

cheers

^ | v • Reply • Share ›

**Myat Nandar Oo** • 2 months ago

Thank u Sir. But i have some trouble.. no error on all steps but, the folder /mnt/

loop is empty... How should i do?

^ | v • Reply • Share ›

**Liembo** • 7 months ago

I just wanted to say thanks for this write up. I used ddrescue (in reverse mode) to recover a 2TB drive that had failed. I was able to fdisk, fsck and mount the filesystem on the disk image that I was able to recover (about 99.5% of it). Getting access to the filesystems was the last step I needed and found your guide here.

^ | v • Reply • Share ›

**kyb** • 10 months ago

I wrote disk image with on-fly arcivation. `dd if=/dev/sdXY | bzip2 -9 > img-file.bz2`. Is there way to mount it without unpacking?

^ | v • Reply • Share ›

**Carmelo** • a year ago

thanks a lot :) it gave me the possibility to mount a dd image of a sd card with several partitions ;)

^ | v • Reply • Share ›

**J3TTBlack88** • a year ago

I have been having trouble with a severely damaged 1TB drive. After running ddrescue for a week, the drive still has 25MB corrupt. I would like to mount the image (since I don't have another drive large enough to write it) but following your instructions I get:

```
file recovery.img > data
```

```
fdisk -l recovery.img >
```

```
Disk ~/recovery.img: 1000.2 GB, 1000204885504 bytes
```

```
255 heads, 63 sectors/track, 121601 cylinders, total 1953525167 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
```

Disk ~/recovery.img doesn't contain a valid partition table

Does anyone have another way to get the start sector in order to mount it as a loop? I obviously need to repair the partition table but testdisk can't see the image without mounting it anyway.

Thanks for the asisstance.

^ | v • Reply • Share ›

**antoniok.spb** • a year ago

To be sure that dealing with 512-byte sectors, use `fdisk -b512 -l` instead of simple `fdisk -l`

^ | v • Reply • Share ›

**Ahmed Shubbar** • a year ago

I cannot find the "startsector" with file command. And fdisk shows 4 partitions, one 80G and another 891G and its only a 8G flash usb memory! The dd\_rescue gave me a file about 8G. Now how can I mount and fix this?

^ | v • Reply • Share ›

**Ciro Santilli** • a year ago

losetup --show -f -P test.img is the new way to go: <http://stackoverflow.com/quest...> || <http://askubuntu.com/questions...> || <http://superuser.com/questions...> || <http://unix.stackexchange.com/...>

^ | v • Reply • Share ›

**Sulman** • a year ago



This is immensely useful. Thanks.

^ | v • Reply • Share ›



**alphasierra** • a year ago

Very useful information. Came in handy. Thanks

^ | v • Reply • Share ›



**Daughenbaugh** • 2 years ago

Thank you for this post. It was quite helpful.

^ | v • Reply • Share ›



**Major Hayden** Mod → **Daughenbaugh** • 2 years ago

Glad to help! :)

^ | v • Reply • Share ›



**Jack** • 2 years ago

This blog has a compact guide to do this using kpart <http://xme.im/mounting-partiti...>

^ | v • Reply • Share ›



**Mike M** • 2 years ago

For anyone trying to make this work with a hard disk that has a GPT , if you try to use fdisk you'll get a warning that "The util fdisk doesn't support GPT. Use GNU Parted." The problem with using parted is that (AFAIK) it isn't easy getting non-abbreviated start and end sectors for the partitions.

What ended up working for me was to use gdisk instead, which works similar to fdisk but is compatible with GPT. It will give you the start and end sector as you would expect. Just be aware that gdisk is still in beta and may have undocumented issues, but for just listing the partition table you should be fine.

^ | v • Reply • Share ›



**Kolja Kube** • 2 years ago

You saved my life with the instructions to repair the filesystem!

My Raspberry Pi had a corrupted SD card, the system was hosed. Your descriptions allowed me to at least backup all data and use it to recreate the system.

Thank you so much!

Kolja

^ | v • Reply • Share ›



**Major Hayden** Mod → **Kolja Kube** • 2 years ago

Awesome!

^ | v • Reply • Share ›



**YouNeed-k** • 2 years ago

I found I needed to use file -k under Ubuntu 14.04.

file -k Your.img

That's -k for "keep going", according to the man page.

Otherwise it stops after displaying the first match, and all I got was "x86 boot sector".

Thanks for posting this. I was faced with the same head-scratcher today.

^ | v • Reply • Share ›



**Saeed** • 2 years ago

hi major

The use of the word in the command loop mount iso files in Linux?

^ | v • Reply • Share ›



**Vinicius Mazzi** • 2 years ago

Thank you so much!!!!

;))))

^ | v • Reply • Share ›



**Will Blair** • 2 years ago



Yeah Mr. Hayden! Yeah Science!

Seriously though, this helped me repair my busted file system on my Raspberry PI when all I had was image generated from dd. Thanks so much!

^ | v • Reply • Share ›



Saumya Kanta Swain • 2 years ago

Thanx a lot for your guide. I had cloned a pendrive which previously had a ubuntu hybrid iso img written to it. As a result, mounting with default options used to mount the ubuntu image instead of the current ntfs filesystem which was at an offset of 2000 sectors.

^ | v • Reply • Share ›



G • 3 years ago

To Lee Jones:

Your Mac OSX 10.6.8 hard drive image likely uses EFI to boot. I'm not on a MAC right now, but have been working on a dd image of a Linux drive installed with EFI bootloader. The dd image is of the whole drive. I needed to find the offset to the second partition to mount it.

First, setup the image to loopback device:

```
losetup /dev/loop0 ./drive.img
```

Next, find where partition sectors start:

```
gparted /dev/loop0
```

This in on Linux, so I don't know if MAC has gparted or some other partitioning tool. . .

I needed to mount the second partition on the drive, gparted showed my partitions in the image, right-clicking on a partition & selecting "information" will give you the first sector of the partition, in my case, 1050624.

Multiply that by 512,  $1050624 * 512 = 537919488$ , the offset number I need.

Now I can mount it:

```
mkdir ./mnt
```

```
mount -o loop,offset=537919488 /dev/loop0 ./mnt
```

The filesystem was now mounted under ./mnt directory & I could access my files.

Worked for me under Linux, should be close to what you need with a MAC.

^ | v • Reply • Share ›



Ribo • 3 years ago

Thanks, cool hack.

^ | v • Reply • Share ›



Lee Jones • 3 years ago

Hi Major.

Ive stumbled upon this page whilst desperately seeking answers to this problem.

Ive used ddrescue to scavenge what data i could from a failing Mac OSX 10.6.8 laptop drive. I've done this a few times and i always get to the same point. I simply can never mount the resulting .dmg disk image file. Ive spent days and days reading and copying and pasting terminal commands, all to no avail.

I must make clear, that i am not a sys admin, i know very little about the Terminal, i just copy what I'm told into there and have always survived.

If someone is familiar with this situation id be so grateful for some help here.

Thanks

Lee

^ | v • Reply • Share ›



Jose Luis Triana • 3 years ago

Thank you very much for your information!

Now I have clear why that particular number (32256) is in the "offset" option to mount loop devices!

I couldn't find any clearer or better explanation of the reason behind that strange numbers!

Thank you for your words of wisdom!

cheers

José Luis.

^ | v • Reply • Share ›



Tiamarchos • 3 years ago

This simple post has saved me. Been trying all sorts fro 2 weeks now!

Thank you!!

^ | v • Reply • Share ›

^ | v • Reply • Share ›



**Juan Bañuls** • 3 years ago

This is exactly the info I was looking for. Thank you very much!

^ | v • Reply • Share ›



**Mark** • 4 years ago

I have a file sda.img made from /dev/sda, using DD. I tried the above, but failed.

My img file is now located on /dev/sdc1 , ext4 file system.

I would like to mount it to /media/myslave on my currant system.

My goal is to shrink the image to fit on a smaller drive.

If there is a easier way to capture the active partitions, mrb, and grub from my original disk and make it boot up, then i would rather do that.

I am running ubuntu 10.04

Thank you for our help

^ | v • Reply • Share ›



**Rod** • 4 years ago

This is a very useful post. Thanks a lot.

^ | v • Reply • Share ›



**Luiz M. Rodrigues** • 6 years ago

This procedure above, would be the same if the image file was Windows FAT?

Could I mount this Windows Fat image made by ddrescue in a Windows 7 system? If yes, Which tool would I use?

^ | v • Reply • Share ›



**Metafaniel** • 6 years ago

I didn't understood the concept of the commands losetup and fsck because I didn't know them, I really should study them.

But following your guide, you've helped me A LOT!

I used ddrescue to clone to an ISO file a not-so-damaged FAT32 based mass memory from a Nokia N96. I managed to do this easily however I couldn't mount the ISO to access the data.

Your perfect guide helped me A LOT but I used a 8192 offset because of the 4k blocksize I used to clone, because it's a FAT32 16GB drive...

THANKS A LOT!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

=D

^ | v • Reply • Share ›



**Major Hayden** • 6 years ago

James -

You could certainly use kpartx in this case. I wasn't aware of kpartx prior to writing this post, so I didn't consider it. ;-)

^ | v • Reply • Share ›



**James Hannah** • 6 years ago

Can you use kpartx for this?

I've used it in the past to mount disk images containing partitions (even with LVM inside) from virtual machines taken off KVM and Xen hardware nodes in the past - it's always been great for that.

^ | v • Reply • Share ›



**Ross** • 6 years ago

Oh, duh. I had completely forgotten hard drive sectors are usually 512 bytes in size. :|

^ | v • Reply • Share ›



**Major Hayden** • 6 years ago

Ross -

On the original hard drive that the image was pulled from, each sector on the disk was 512 bytes long. The offset was 63 sectors from the beginning of the disk, so that's why you multiply 63 by 512 to find the offset. The block size on the filesystem would be an entirely different discussion than what is covered in this post.

^ | v • Reply • Share ›



Ross • 6 years ago

I am not a clever person.

How did you determine a block size of 512 for the .img file? Or was it noted prior to cloning the disk?

^ | v • Reply • Share ›

ALSO ON MAJOR.IO

### Nobody is using your software project. Now what?

1 comment • a year ago•



**Mauro Morales** — Would you care about elaborating on the Technical talks? Most specifically I'd like to know if you have an idea of what makes a good ...

### OpenStack Summit in Austin is almost here!

3 comments • 9 months ago•



**Florian Haas** — Likewise!

### Tinkering with systemd's predictable network names

3 comments • a year ago•



**goldyfruit** — Nice catch about the 99-default.link !

### Bring back two and three finger taps in Fedora 24

6 comments • 7 months ago•



**Jules** — hi mine not working on fedora 24 My xinput output is[ninja@localhost ~]\$ xinput list ↳ Virtual core pointer id=2 [master pointer (3)] | ↳ Virtual core ...

✉ Subscribe   Add Disqus to your site   Add Disqus Add   Privacy