

Mount a dd Disk Image with Partition Table inside

After making a backup from a hard disk to a disk image using plain old 'dd', I was just looking into mounting it using the Linux loopback device.

If you 'dd' a single partition into an image file, then this is very straight forward. But if your image file contains multiple partition partitions including the partition table itself, then you need to take additional steps.

So the first thing to know is the exact offset of the partition to be mounted.

You may examine this easily using parted. Just provide it with the path to the image file instead of a block device.

```
root@debian:~# parted full_hd.dd
GNU Parted 2.3
Using /root/full_hd.dd
Welcome to GNU Parted! Type 'help' to view a list of commands.
```

Now switch parted to use 'Byte' units, then print the partition table:

```
(parted) unit B
(parted) print
Model: (file)
Disk /root/full_hd.dd: 8012390400B
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number Start End Size Type File system Flags
 1 1048576B 8012169215B 8011120640B primary ext4 boot
```

Keep a record of the 'Start' offset, you'll need this shortly. Quit parted using the 'quit' command.

Let's look into mounting the partition: pass the partition offset to the mount command using the 'offset=' option like this:

```
root@debian:~# mount -t ext4 -o loop,offset=1048576 full_hd.dd /mnt/test/
```

It may well be possible, that receive this error message as a result to your mount command if you try to mount the disk image read-only:

```
mount: wrong fs type, bad option, bad superblock on /dev/loop0,
missing codepage or helper program, or other error
In some cases useful info is found in syslog - try
dmesg | tail or so
```

In this case, examining the system logs, may reveal something like this:

```
[ 8754.209469] EXT4-fs (loop0): VFS: Can't find ext4 filesystem  
[ 8758.913583] EXT4-fs (loop0): INFO: recovery required on readonly filesystem  
[ 8758.913587] EXT4-fs (loop0): write access unavailable, cannot proceed
```

To get around this, try adding the 'noload' option to skip journal recovery:

```
root@debian:~# mount -t ext4 -o loop,ro,noload,offset=1048576 full_hd.dd /mnt/test/
```