

# RE: RAID gotchas!

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Fedora/2007-06/msg06398.html>

---

- *From:* "Ferguson, Michael" <[ferguson@xxxxxxxxxxxx](mailto:ferguson@xxxxxxxxxxxx)>
  - *Date:* Sat, 30 Jun 2007 14:09:29 -0400
- 

Jeff, Thanks

Michael E. Ferguson

-----Original Message-----

From: [fedora-list-bounces@xxxxxxxxxxxx](mailto:fedora-list-bounces@xxxxxxxxxxxx) [<mailto:fedora-list-bounces@xxxxxxxxxxxx>]

On Behalf Of Jeffrey Ross

Sent: Saturday, June 30, 2007 2:03 PM

To: For users of Fedora

Subject: RAID gotchas!

Well I converted my single disk system last night into a dual disk RAID-1 setup and preserved all my data. I Thought I'd share an overview of the process and what I had problems with, maybe somebody here can use my experiences to help them.

First some background, I'm running Fedora 7 with all the current fedora patches, I am not using any third part repositories (yet) like Livna or Freshrpms.

The system is an Intel Pentium D processor with an Intel DG965RY motherboard utilizing 2- 400GB Seagate ST3400620AS SATA drives. NOTE: don't forget to remove the tiny jumper (and promptly loose it in carpet) on the drive to allow it to run at 3.0GB/s if your system allows it, the jumper comes installed by default limiting speed to 1.5Gb/s.

My system was running fine on /dev/sda, I added the new disk as /dev/sdb and partitioned it as follows:

```
/dev/sdb1 1 32 257008+ fd Linux raid
autodetect
/dev/sdb2 33 1307 10241437+ fd Linux raid
autodetect
/dev/sdb3 1308 1829 4192965 fd Linux raid
autodetect
/dev/sdb4 1830 48641 376017390 5 Extended
/dev/sdb5 1830 2199 2971993+ fd Linux raid
autodetect
```

## RE: RAID gotchas!

```
/dev/sdb6 2200 2568 2963961 fd Linux raid
autodetect
/dev/sdb7 2569 48641 370081341 fd Linux raid
autodetect
```

My partitions were laid out as follows:

```
/dev/sdb1 = /dev/md1 = /boot
/dev/sdb2 = /dev/md2 = /usr
/dev/sdb3 = /dev/md3 = swap
/dev/sdb4 = extended partition
/dev/sdb5 = /dev/md5 = /var
/dev/sdb6 = /dev/md6 = /
/dev/sdb7 = /dev/md7 = /home
```

I kept the partition number the same as the raid partition number just because it made my life easier to keep track of everything but there is no reason it needs to match.

next I needed to create the partitions, this was pretty simple, I used the following command:

```
mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sdb1 missing mdadm
--create /dev/md2 --level=1 --raid-devices=2 /dev/sdb2 missing mdadm
--create /dev/md3 --level=1 --raid-devices=2 /dev/sdb3 missing mdadm
--create /dev/md5 --level=1 --raid-devices=2 /dev/sdb5 missing mdadm
--create /dev/md6 --level=1 --raid-devices=2 /dev/sdb6 missing mdadm
--create /dev/md7 --level=1 --raid-devices=2 /dev/sdb7 missing
```

Note the "missing" at the end of the command, this will allow the system to create the raid volumes since the other disk isn't available yet.

I created the file /etc/mdadm.conf and put the following in it. NOTE: if boot off a raid drive or load the raid module before mounting the partitions I don't think you need the "ARRAY" entries, but I used them anyway.

```
/etc/mdadm.conf:
# Who should get alerts?
MAILADDR root
ARRAY /dev/md1 devices=/dev/sda1,/dev/sdb1 ARRAY /dev/md2
devices=/dev/sda2,/dev/sdb2 ARRAY /dev/md3 devices=/dev/sda3,/dev/sdb3 ARRAY
/dev/md5 devices=/dev/sda5,/dev/sdb5 ARRAY /dev/md6
devices=/dev/sda6,/dev/sdb6 ARRAY /dev/md7 devices=/dev/sda7,/dev/sdb7
```

NOTE: until the second disk was added to the array I only had one entry following the devices entry (eg devices=/dev/sdb1)

now you need to create the file systems, I kept everything as ext3 mkfs -V -t ext3 /dev/mdX

RE: RAID gotchas!

## RE: RAID gotchas!

where X was 1, 2, 5, 6, & 7

Don't forget swap!  
mkswap /dev/md3

Now mount your new partitions, I mounted them under "/mnt"

```
mkdir /mnt/new-root  
mount /dev/md6 /mnt/new-root
```

create the new mount points (you could restore root first then just mount them) mkdir /mnt/new-root/var mkdir /mnt/new-root/usr mkdir /mnt/new-root/home mkdir /mnt/new-root/boot

```
mount /dev/md1 /mnt/new-root/boot  
mount /dev/md2 /mnt/new-root/usr
```

etc

Now comes the fun part, you need to move your data to the new partition. Although I've read where you can shrink the partition and convert to a raid volume, I decided against that.

I used dump/restore using the command:

```
dump -0 -b 1024 -f - /dev/sdaX | restore -rf -
```

NOTE: I'd recommend single user mode for the copy, better yet unmount the source volume if possible, secondly run this command in the destination directory!

Second NOTE: by using the option `-b 1024` the performance of dump was increased about 10 fold however upon completion you will get "broken pipe" error, I found everything was copied properly and didn't worry about it.

All your data is copied to the RAID volume but a reboot will only load from the old disk. I did the following:

modified /etc/fstab to read:

```
/dev/md6 / ext3 defaults 1 1  
/dev/md7 /home ext3 defaults 1 2  
/dev/md5 /var ext3 defaults 1 2  
/dev/md2 /usr ext3 defaults 1 2  
/dev/md1 /boot ext3 defaults 1 2  
/dev/md3 swap swap  
defaults 0 0
```

Note for clarity I removed tmpfs, devpts etc. Also in hindsight I probably

RE: RAID gotchas!

## RE: RAID gotchas!

could just use the label command "e2label /dev/md1 /boot" etc but I wanted to be positive what would be mounted.

I modified grub.conf on \*both\* the new partition and the old partition to read:

```
kernel /vmlinuz-2.6.21-1.3228.fc7 ro root=/dev/md6
```

and then ran grub to install the boot loaded on the disks.

```
grub-install /dev/sda
```

```
grub-install /dev/sdb
```

At this point I rebooted.... and if your familiar with how raid works you'll know the system wouldn't boot. At this point I booted off the rescue disk and was able to mount all my raid partitions. Of course that didn't help me get the system reloaded so I started searching the internet for clues.

The answer came in mkinitrd mounted my partitions in the recovery mode, now knowing what I know now should have been prior to the first reboot.

rename the existing initrd file to something else ( eg .old ?) then from the new /boot directory run the following, and copy it to the old boot directory as well (unless you can boot from /dev/sdb in your bios)

```
mkinitrd --preload=raid1 initrd-2.6.21.1.3228.fc7.img 2.6.21.1.3228.fc7
```

At this point you should be able to reboot and the system will be running on the (degraded) RAID disks.

If your happy with everything then you can repartition your original drive (fdisk) to match /dev/sdb once that is done you need to add the new partitions to the raid volume. To do this enter the command:

```
mdadm --add /dev/mdX /dev/sdaX (where X is the partition number of the volume)
```

Do this for all your remaining partitions, and then you can cat /proc/mdstat and see the volumes being rebuilt ( hint: "watch cat /proc/mdstat")

Your done!

Hope this was helpful to somebody.

Jeff

RE: RAID gotchas!

—

fedora-list mailing list

fedora-list@xxxxxxxxxx

To unsubscribe: <https://www.redhat.com/mailman/listinfo/fedora-list>

—

fedora-list mailing list

fedora-list@xxxxxxxxxx

To unsubscribe: <https://www.redhat.com/mailman/listinfo/fedora-list>