

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	18	144553+	83	Linux
/dev/sda2		19	80	498015	82	Linux swap / Solaris
/dev/sda3		81	652	4594590	83	Linux

Disk /dev/sdb: 5368 MB, 5368709120 bytes
 255 heads, 63 sectors/track, 652 cylinders
 Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/sdb doesn't contain a valid partition table
 server1:~#

2 Installing mdadm

The most important tool for setting up RAID is mdadm. Let's install it like this:

```
apt-get install initramfs-tools mdadm
```

You will be asked the following question:

MD arrays needed for the root filesystem: <-- all

Afterwards, we load a few kernel modules (to avoid a reboot):

```
modprobe md
modprobe linear
modprobe multipath
modprobe raid0
modprobe raid1
modprobe raid5
modprobe raid6
modprobe raid10
```

Now run

```
cat /proc/mdstat
```

The output should look as follows:

```
server1:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
unused devices: <none>
server1:~#
```

3 Preparing /dev/sdb

To create a RAID1 array on our already running system, we must prepare the /dev/sdb hard drive for RAID1, then copy the contents of our /dev/sda hard drive to it, and finally add /dev/sda to the RAID1 array.

First, we copy the partition table from /dev/sda to /dev/sdb so that both disks have exactly the same layout:

```
sfdisk -d /dev/sda | sfdisk /dev/sdb
```

The output should be as follows:

```
server1:~# sfdisk -d /dev/sda | sfdisk /dev/sdb
Checking that no-one is using this disk right now ...
OK
```

Disk /dev/sdb: 652 cylinders, 255 heads, 63 sectors/track

```
sfdisk: ERROR: sector 0 does not have an msdos signature
/dev/sdb: unrecognized partition table type
```

Old situation:

No partitions found

New situation:

Units = sectors of 512 bytes, counting from 0

```
Device Boot Start End #sectors Id System
/dev/sdb1 * 63 289169 289107 83 Linux
/dev/sdb2 289170 1285199 996030 82 Linux swap / Solaris
/dev/sdb3 1285200 10474379 9189180 83 Linux
/dev/sdb4 0 - 0 0 Empty
Successfully wrote the new partition table
```

Re-reading the partition table ...

If you created or changed a DOS partition, /dev/foo7, say, then use dd(1) to zero the first 512 bytes: dd if=/dev/zero of=/dev/foo7 bs=512 count=1 (See fdisk(8).)

```
server1:~#
```

The command

```
fdisk -l
```

should now show that both HDDs have the same layout:

```
server1:~# fdisk -l
```

Disk /dev/sda: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

```
Device Boot Start End Blocks Id System
/dev/sda1 * 1 18 144553+ 83 Linux
/dev/sda2 19 80 498015 82 Linux swap / Solaris
/dev/sda3 81 652 4594590 83 Linux
```

Disk /dev/sdb: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

```
Device Boot Start End Blocks Id System
/dev/sdb1 * 1 18 144553+ 83 Linux
/dev/sdb2 19 80 498015 82 Linux swap / Solaris
/dev/sdb3 81 652 4594590 83 Linux
```

```
server1:~#
```

Next we must change the partition type of our three partitions on /dev/sdb to Linux raid autodetect:

fdisk /dev/sdb

server1:~# fdisk /dev/sdb

Command (m for help): <-- m

Command action

- a toggle a bootable flag
- b edit bsd disklabel
- c toggle the dos compatibility flag
- d delete a partition
- l list known partition types
- m print this menu
- n add a new partition
- o create a new empty DOS partition table
- p print the partition table
- q quit without saving changes
- s create a new empty Sun disklabel
- t change a partition's system id
- u change display/entry units
- v verify the partition table
- w write table to disk and exit
- x extra functionality (experts only)

Command (m for help): <-- t

Partition number (1-4): <-- 1

Hex code (type L to list codes): <-- L

0	Empty	1e	Hidden W95 FAT1	80	Old Minix	be	Solaris boot
1	FAT12	24	NEC DOS	81	Minix / old Lin	bf	Solaris
2	XENIX root	39	Plan 9	82	Linux swap / So	c1	DRDOS/sec (FAT-
3	XENIX usr	3c	PartitionMagic	83	Linux	c4	DRDOS/sec (FAT-
4	FAT16 <32M	40	Venix 80286	84	OS/2 hidden C:	c6	DRDOS/sec (FAT-
5	Extended	41	PPC PReP Boot	85	Linux extended	c7	Syrinx
6	FAT16	42	SFS	86	NTFS volume set	da	Non-FS data
7	HPFS/NTFS	4d	QNX4.x	87	NTFS volume set	db	CP/M / CTOS / .
8	AIX	4e	QNX4.x 2nd part	88	Linux plaintext	de	Dell Utility
9	AIX bootable	4f	QNX4.x 3rd part	8e	Linux LVM	df	BootIt
a	OS/2 Boot Manag	50	OnTrack DM	93	Amoeba	e1	DOS access
b	W95 FAT32	51	OnTrack DM6 Aux	94	Amoeba BBT	e3	DOS R/O
c	W95 FAT32 (LBA)	52	CP/M	9f	BSD/OS	e4	SpeedStor
e	W95 FAT16 (LBA)	53	OnTrack DM6 Aux	a0	IBM Thinkpad	hi	BeOS fs
f	W95 Ext'd (LBA)	54	OnTrackDM6	a5	FreeBSD	ee	EFI GPT
10	OPUS	55	EZ-Drive	a6	OpenBSD	ef	EFI (FAT-12/16/
11	Hidden FAT12	56	Golden Bow	a7	NeXTSTEP	f0	Linux/PA-RISC b
12	Compaq diagnost	5c	Priam Edisk	a8	Darwin UFS	f1	SpeedStor
14	Hidden FAT16 <3	61	SpeedStor	a9	NetBSD	f4	SpeedStor
16	Hidden FAT16	63	GNU HURD or Sys	ab	Darwin boot	f2	DOS secondary
17	Hidden HPFS/NTF	64	Novell Netware	b7	BSDI fs	fd	Linux raid auto
18	AST SmartSleep	65	Novell Netware	b8	BSDI swap	fe	LANstep
1b	Hidden W95 FAT3	70	DiskSecure Mult	bb	Boot Wizard	hid	ff BBT
1c	Hidden W95 FAT3	75	PC/IX				

Hex code (type L to list codes): <-- fd

Changed system type of partition 1 to fd (Linux raid autodetect)

Command (m for help): <-- t

Partition number (1-4): <-- 2


```
md0 : active raid1 sdb1[1]
      144448 blocks [2/1] [_U]
```

unused devices: <none>

```
server1:~#
```

Next we create filesystems on our RAID arrays (ext3 on /dev/md0 and /dev/md2 and swap on /dev/md1):

```
mkfs.ext3 /dev/md0
mkswap /dev/md1
mkfs.ext3 /dev/md2
```

Next we must adjust /etc/mdadm/mdadm.conf (which doesn't contain any information about our new RAID arrays yet) to the new situation:

```
cp /etc/mdadm/mdadm.conf /etc/mdadm/mdadm.conf_orig
mdadm --examine --scan >> /etc/mdadm/mdadm.conf
```

Display the contents of the file:

```
cat /etc/mdadm/mdadm.conf
```

At the bottom of the file you should now see details about our three (degraded) RAID arrays:

```
# mdadm.conf
#
# Please refer to mdadm.conf(5) for information about this file.
#
# by default, scan all partitions (/proc/partitions) for MD superblocks.
# alternatively, specify devices to scan, using wildcards if desired.
DEVICE partitions

# auto-create devices with Debian standard permissions
CREATE owner=root group=disk mode=0660 auto=yes

# automatically tag new arrays as belonging to the local system
HOMEHOST <system>

# instruct the monitoring daemon where to send mail alerts
MAILADDR root

# This file was auto-generated on Mon, 26 Nov 2007 21:22:04 +0100
# by mkconf $Id: mkconf 261 2006-11-09 13:32:35Z madduck $
ARRAY /dev/md0 level=raid1 num-devices=2
UUID=72d23d35:35d103e3:01b5209e:be9ff10a
ARRAY /dev/md1 level=raid1 num-devices=2
UUID=a50c4299:9e19f9e4:01b5209e:be9ff10a
ARRAY /dev/md2 level=raid1 num-devices=2
UUID=99fee3a5:ae381162:01b5209e:be9ff10a
```

5 Adjusting The System To RAID1

Now let's mount /dev/md0 and /dev/md2 (we don't need to mount the swap array /dev/md1):

```
mkdir /mnt/md0
mkdir /mnt/md2
```


In the same file, go to the bottom where you should find some kernel stanzas. Copy the first of them and paste the stanza before the first existing stanza; replace `root=/dev/sda3` with `root=/dev/md2` and `root (hd0,0)` with `root (hd1,0)`:

```
[...]
## ## End Default Options ##

title          Debian GNU/Linux, kernel 2.6.18-4-486 RAID (hd1)
root           (hd1,0)
kernel         /vmlinuz-2.6.18-4-486 root=/dev/md2 ro
initrd         /initrd.img-2.6.18-4-486
savedefault

title          Debian GNU/Linux, kernel 2.6.18-4-486
root           (hd0,0)
kernel         /vmlinuz-2.6.18-4-486 root=/dev/sda3 ro
initrd         /initrd.img-2.6.18-4-486
savedefault

title          Debian GNU/Linux, kernel 2.6.18-4-486 (single-user mode)
root           (hd0,0)
kernel         /vmlinuz-2.6.18-4-486 root=/dev/sda3 ro single
initrd         /initrd.img-2.6.18-4-486
savedefault

### END DEBIAN AUTOMAGIC KERNELS LIST
```

`root (hd1,0)` refers to `/dev/sdb` which is already part of our RAID arrays. We will reboot the system in a few moments; the system will then try to boot from our (still degraded) RAID arrays; if it fails, it will boot from `/dev/sda` (-> fallback 1).

Next we adjust our ramdisk to the new situation:

```
update-initramfs -u
```

Now we copy the contents of `/dev/sda1` and `/dev/sda3` to `/dev/md0` and `/dev/md2` (which are mounted on `/mnt/md0` and `/mnt/md2`):

```
cp -dpRx //mnt/md2
```

```
cd /boot
```

```
cp -dpRx ./mnt/md0
```

6 Preparing GRUB (Part 1)

Afterwards we must install the GRUB bootloader on the second hard drive `/dev/sdb`:

```
grub
```

On the GRUB shell, type in the following commands:

```
root (hd0,0)
```

```
grub> root (hd0,0)
```

```
Filesystem type is ext2fs, partition type 0x83
```

```
grub>
```


setup (hd0)

grub> setup (hd0)

```
Checking if "/boot/grub/stage1" exists... no
Checking if "/grub/stage1" exists... yes
Checking if "/grub/stage2" exists... yes
Checking if "/grub/e2fs_stage1_5" exists... yes
Running "embed /grub/e2fs_stage1_5 (hd0)"... 15 sectors are embedded.
succeeded
Running "install /grub/stage1 (hd0) (hd0)1+15 p (hd0,0)/grub/stage2 /grub/menu.lst"... succeeded
Done.
```

grub>

root (hd1,0)

grub> root (hd1,0)

```
Filesystem type is ext2fs, partition type 0xfd
```

grub>

setup (hd1)

grub> setup (hd1)

```
Checking if "/boot/grub/stage1" exists... no
Checking if "/grub/stage1" exists... yes
Checking if "/grub/stage2" exists... yes
Checking if "/grub/e2fs_stage1_5" exists... yes
Running "embed /grub/e2fs_stage1_5 (hd1)"... 15 sectors are embedded.
succeeded
Running "install /grub/stage1 (hd1) (hd1)1+15 p (hd1,0)/grub/stage2 /grub/menu.lst"... succeeded
Done.
```

grub>

quit

Now, back on the normal shell, we reboot the system and hope that it boots ok from our RAID arrays:

reboot

7 Preparing /dev/sda

If all goes well, you should now find /dev/md0 and /dev/md2 in the output of

df -h

```
server1:~# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/md2	4.4G	730M	3.4G	18%	/
tmpfs	126M	0	126M	0%	/lib/init/rw
udev	10M	68K	10M	1%	/dev
tmpfs	126M	0	126M	0%	/dev/shm
/dev/md0	137M	17M	114M	13%	/boot

```
server1:~#
```

The output of

```
cat /proc/mdstat
```

should be as follows:

```
server1:~# cat /proc/mdstat
Personalities : [raid1]
md2 : active raid1 sdb3[1]
      4594496 blocks [2/1] [_U]

md1 : active raid1 sdb2[1]
      497920 blocks [2/1] [_U]

md0 : active raid1 sdb1[1]
      144448 blocks [2/1] [_U]

unused devices: <none>
server1:~#
```

Now we must change the partition types of our three partitions on /dev/sda to Linux raid autodetect as well:

```
fdisk /dev/sda
```

```
server1:~# fdisk /dev/sda
```

```
Command (m for help): <-- t
Partition number (1-4): <-- 1
Hex code (type L to list codes): <-- fd
Changed system type of partition 1 to fd (Linux raid autodetect)
```

```
Command (m for help): <-- t
Partition number (1-4): <-- 2
Hex code (type L to list codes): <-- fd
Changed system type of partition 2 to fd (Linux raid autodetect)
```

```
Command (m for help): <-- t
Partition number (1-4): <-- 3
Hex code (type L to list codes): <-- fd
Changed system type of partition 3 to fd (Linux raid autodetect)
```

```
Command (m for help): <-- w
The partition table has been altered!
```

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.

The kernel still uses the old table.

The new table will be used at the next reboot.

Syncing disks.

```
server1:~#
```

Now we can add /dev/sda1, /dev/sda2, and /dev/sda3 to the respective RAID arrays:

```
mdadm --add /dev/md0 /dev/sda1
mdadm --add /dev/md1 /dev/sda2
mdadm --add /dev/md2 /dev/sda3
```

Now take a look at

```
cat /proc/mdstat
```

... and you should see that the RAID arrays are being synchronized:

```
server1:~# cat /proc/mdstat
Personalities : [raid1]
md2 : active raid1 sda3[2] sdb3[1]
      4594496 blocks [2/1] [_U]
      [=====>.....] recovery = 29.7% (1367040/4594496) finish=0.6min speed=85440K/sec

md1 : active raid1 sda2[0] sdb2[1]
      497920 blocks [2/2] [UU]

md0 : active raid1 sda1[0] sdb1[1]
      144448 blocks [2/2] [UU]

unused devices: <none>
server1:~#
```

(You can run

```
watch cat /proc/mdstat
```

to get an ongoing output of the process. To leave watch, press CTRL+C.)

Wait until the synchronization has finished (the output should then look like this:

```
server1:~# cat /proc/mdstat
Personalities : [raid1]
md2 : active raid1 sda3[0] sdb3[1]
      4594496 blocks [2/2] [UU]

md1 : active raid1 sda2[0] sdb2[1]
      497920 blocks [2/2] [UU]

md0 : active raid1 sda1[0] sdb1[1]
      144448 blocks [2/2] [UU]

unused devices: <none>
server1:~#
```

).

Then adjust /etc/mdadm/mdadm.conf to the new situation:

```
cp /etc/mdadm/mdadm.conf_orig /etc/mdadm/mdadm.conf
mdadm --examine --scan >> /etc/mdadm/mdadm.conf
```

/etc/mdadm/mdadm.conf should now look something like this:

2 heads, 4 sectors/track, 36112 cylinders
Units = cylinders of 8 * 512 = 4096 bytes

Disk /dev/md0 doesn't contain a valid partition table

Disk /dev/md1: 509 MB, 509870080 bytes
2 heads, 4 sectors/track, 124480 cylinders
Units = cylinders of 8 * 512 = 4096 bytes

Disk /dev/md1 doesn't contain a valid partition table

Disk /dev/md2: 4704 MB, 4704763904 bytes
2 heads, 4 sectors/track, 1148624 cylinders
Units = cylinders of 8 * 512 = 4096 bytes

Disk /dev/md2 doesn't contain a valid partition table
server1:~#

Now we copy the partition table of /dev/sda to /dev/sdb:

```
sfdisk -d /dev/sda | sfdisk /dev/sdb
```

(If you get an error, you can try the --force option:

```
sfdisk -d /dev/sda | sfdisk --force /dev/sdb
```

)

```
server1:~# sfdisk -d /dev/sda | sfdisk /dev/sdb  
Checking that no-one is using this disk right now ...  
OK
```

Disk /dev/sdb: 652 cylinders, 255 heads, 63 sectors/track

```
sfdisk: ERROR: sector 0 does not have an msdos signature  
/dev/sdb: unrecognized partition table type
```

Old situation:

No partitions found

New situation:

Units = sectors of 512 bytes, counting from 0

Device	Boot	Start	End	#sectors	Id	System
/dev/sdb1	*	63	289169	289107	fd	Linux raid autodetect
/dev/sdb2		289170	1285199	996030	fd	Linux raid autodetect
/dev/sdb3		1285200	10474379	9189180	fd	Linux raid autodetect
/dev/sdb4		0	-	0	0	Empty

Successfully wrote the new partition table

Re-reading the partition table ...

If you created or changed a DOS partition, /dev/foo7, say, then use dd(1) to zero the first 512 bytes: dd if=/dev/zero of=/dev/foo7 bs=512 count=1 (See fdisk(8).)

```
server1:~#
```

Afterwards we remove any remains of a previous RAID array from /dev/sdb...

