# Boot + Root + Raid + Lilo : Software Raid HOWTO

# Table of Contents

# Boot + Root + Raid + Lilo : Software Raid HOWTO


## Michael Robinton, Michael@BizSystems.com



v1.04, July 20, 2000


---


*This document provides a cookbook for setting up root raid using the 0.90 raidtools for bootable raid mounted on root using standard LILO. Also covered is the conversion of a conventional disk to a raid1 or raid5 mirror set without the loss of data on the original disk.*


---

# 5.Appendix A. − example raidtab

# 6.Appendix B. − SCSI reference implementation RAID5

# 7.Appendix C. − ide RAID10 with initrd

# 8.Appendix D. − ide RAID1−10 with initrd

---

# 1.Introduction

## 1.1 Acknowledgements

The essence of the information I've put together here was originally provided by Harald Nordgård−Hansen <hnh@bukharin.hiof.no> and posted to the raid mail list in a lilo.conf file with commentary by Martin Bene <mb@sime.com>. Many thanks for your contribution. I've tried to put this information and the helpful work of many others who contribute to the raid mail list and linux raid project into a **COOKBOOK** form, including many examples from real systems so that bootable root raid is easy to set up and understand. One section is devoted to the conversion of a standard single drive system to RAID. The key to the conversion, in my humble opinion, is the understanding of bootable root raid.

## 1.2 Bugs

Yes, I'm sure there are some. If you'd be good enough to report them, I will correct the document. ;−)

## 1.3 Copyright Notice

# 2.What you need BEFORE YOU START

The packages you need and the documentation that answers the most common questions about setting up and running raid are listed below. Please review them throughly.

## 2.1 Required Packages

You need to obtain the most recent versions of these packages.

- a linux kernel that supports raid, initrd

    I used linux−2.2.14 from kernel.org

- ftp://ftp.kernel.org/pub/linux/daemons/raid/alpha/ the most recent tools and patch that adds support for modern raid1/4/5

    I used http://people.redhat.com/mingo/raid−patches/

## 2.2 Where to get Up−to−date copies of this document.

Click here to browse the author's latest version of this document. Corrections and suggestions welcome!

Boot Root Raid + LILO HOWTO

Available in LaTeX (for DVI and PostScript), plain text, and HTML.

    http://www.linuxdoc.org/HOWTO/Boot+Root+Raid+LILO.html

Available in SGML and HTML.

    ftp.bizsystems.net/pub/raid/

## 2.3 Documentation −− Recommended Reading

**If you plan on using raid1/5 over raid0, please read:**

**/usr/src/linux/Documentation/initrd.txt**

as well as the documentation and man pages that accompany the raidtools set.

and..... Software−RAID−HOWTO.html

## 2.4 RAID resources

Mailing lists can be joined at:

- This one seems quiet: majordomo@nuclecu.unam.mx *send a message to***subscribe raiddev**

  send mail to: raiddev@nuclecu.unam.mx
- Raid development: majordomo@vger.rutgers.edu *send a message to***subscribe linux−raid**

  send mail to: linux−raid@vger.rutgers.edu*(this seems to be the most active list)*

---

# 3.Bootable Raid

I'm not going to cover the fundamentals of setting up raid0/1/5 on Linux, that is covered in detail elsewhere. The problem I will address is setting up raid on root and making it bootable with **standard** LILO. The documentation that comes with the LILO sources (not the man pages) and with the raidtools−0.90, covers the details of booting and boot parameters as well as general raid setup − respectively.

There are two scenarios which are covered here. Set up of bootable root raid and the conversion of an existing non−raid system to bootable root raid without data loss.

## 3.1 Booting RAID 1 with standard LILO

To make the boot information redundant and easy to maintain, set up a small RAID1 and mount it on the **/boot** directory of your system disk. LILO does not know about device 0x9?? and can not find the information at boot time because the raid sub system is not active then. As a simple work around, you can pass LILO the geometry information of the drive(s) and from that, LILO can determine the position of the information needed to load the kernel even though it is on the RAID1 partition. This is because the RAID1 partition is the same as a standard partition but with a raid super−block written at the end. The boot raid set should fall with the first 1024 mbytes of the disk drive. In theory the start of the raid partition could fall anywhere in the 1024 megs, but in practice I was unable to get it to work unless the boot−raid started at the first block of the set. This is probably because of something dumb that I did, but it was not worth following up at the time. Since then I've simply set up all my systems with the boot−raid set as the first partition. I have root raid system configurations with bootable RAID1 mounted on **/boot** with root raid sets as follows:

RAID1, RAID5, RAID10 & RAID1−10 ( 1 mirror + 1 raid0 set). The last has a very peculiar lilo file pair since none of the disk geometries are the same, however, the principals are the same for the initial boot process. The RAID10 and RAID1−10 root mounts require the use of *initrd* to mount root after the boot process has taken place. See the appendices for the configuration files for all of these example systems.

A conventional LILO config file stripped down looks like this:

```
# lilo.conf - assumes drive less than 1024
        boot = /dev/hda
        delay = 40                # extra, but nice
        vga = normal              # not normally needed
        image = /bzImage
        root = /dev/hda1
        read-only
        label = Linux
```

A raid LILO config file pair would look like this:

```
# lilo.conf.hda - primary ide master
        disk=/dev/md0
        bios=0x80
        sectors=63
        heads=16
        cylinders=39770
        partition=/dev/md1
        start=63
        boot=/dev/hda
        map=/boot/map
        install=/boot/boot.b
        image=/boot/bzImage
        root=/dev/md0
        read-only
        label=LinuxRaid

# ---------------------

# lilo.conf.hdc - secondary ide master
        disk=/dev/md0
        bios=0x80                 # see note below
        sectors=63
        heads=16
        cylinders=39770
        partition=/dev/md1
        start=63
        boot=/dev/hdc             # this is the other disk
        map=/boot/map
        install=/boot/boot.b
        image=/boot/bzImage
        root=/dev/md0
        read-only
        label=LinuxRaid
```

# BIOS=line −− if your bios is smart enough (most are not) to detect that that the first disk is missing or failed and will automatically boot from the second disk, then **bios=81** would be the appropriate entry here. This is more common with SCSI bios than IDE bios. I simply plan on relocating the drive so it will replace

the dead drive C: in the event of failure of the primary boot drive.

The geometry information for the drive can be obtained from fdisk with the command:

```
fdisk -ul (little L)
fdisk -ul /dev/hda

Disk /dev/hda: 16 heads, 63 sectors, 39770 cylinders
Units = sectors of 1 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1            63     33263     16600+  fd  Linux raid autodetect
/dev/hda2         33264    443519    205128   82  Linux swap
/dev/hda3        443520  40088159  19822320   fd  Linux raid autodetect

* note the listing of the START of each partition
```

## 3.2 Detailed explaination of lilo.conf for raid boot

The raid lilo.conf file above, commented in detail for each entry.

```
# lilo.conf.hda - primary ide master
#       the location of the /boot directory that will be
#       designated below as containing the kernel, map, etc...
#       note that this is NOT the actual partition containing
#       the boot image and info, but rather the device
#       that logically contains this directory.
#       in this example, /dev/md1 is mounted on /dev/md0/boot
     disk=/dev/md0

#       tell LILO which bios device to use for boot, i.e. C: drive
     bios=0x80

#       tell LILO the geometry of the device
#       this is usually but not always the "logical"
#       geometry. Check the /proc file system or watch
#       the boot messages when the kernel probes for the drive
#
     sectors=63
     heads=16
     cylinders=39770

#       this is a dummy entry to make LILO happy so it
#       will recognize the raid set 0x9?? and then find
#       the START of the boot sector. To really see
#       what this was for, read the documentation
#       that comes with the LILO source distribution.
#       This parameter "must" be different than the
#       disk=  entry above. It can be any other mdx
#       device, used or unused and need not be the one
#       that contains the /boot information
#
     partition=/dev/md1
```

```
#       the first sector of the partition containing /boot information
     start=63

#       the real device that LILO will write the boot information to
     boot=/dev/hda

#       logically where LILO will put the boot information
     map=/boot/map
     install=/boot/boot.b

#       logically where lilo will find the kernel image
     image=/boot/bzImage

#       standard stuff after this
#       root may be a raid1/4/5 device
     root=/dev/md0
     read-only
     label=LinuxRaid
```

---

# 4.Upgrading from non−raid to RAID1/4/5

Upgrading a non−raid system to raid is fairly easy and consists of several discrete steps described below. The description is for a system with a boot partition, root partition and swap partition.

```
OLD disk in the existing system:

    /dev/hda1      boot, may be dos+lodlin or lilo
    /dev/hda2      root
    /dev/hda3      swap
```

We will add an additional disk and convert the entire system to RAID1. You could easily add several disks and make a RAID5 set instead using the same procedure.

## 4.1 Step 1 − prepare a new kernel

Download a clean kernel, raidtools−0.90 (or the most recent version), and the kernel patch to upgrade the kernel to 0.90 raid.

Compile and install the raidtools and READ the documentation.

Compile and install the kernel to support all the flavors (0/1/4/5 ?) of raid that you will be using. Make sure to specify autostart of raid devices in the kernel configuration. Test that the kernel boots properly and examine /proc/mdstat to see that the raid flavors you will use are supported by the new kernel.

## 4.2 Step 2 − set up raidtab for your new raid.

The new disk will be added to an additional IDE controller as the master device, thus becomming /dev/hdc

```
    /dev/hdc1      16megs -- more than enough for several kernel images
    /dev/hdc2      most of the disk
    /dev/hdc3      some more swap space, if needed. otherwise add to hdc2
```

Change the partition types for /dev/hdc1 and /dev/hdc2 to "fd" for raid−autostart.

Using the **failed−disk** parameter, create a raidtab for the desired RAID1 configuration. The failed disk must be the last entry in the table.

```
# example raidtab
# md0 is the root array
raiddev                 /dev/md0
raid-level              1
nr-raid-disks           2
chunk-size              32
# Spare disks for hot reconstruction
nr-spare-disks          0
persistent-superblock   1
device                  /dev/hdc2
raid-disk               0
# this is our old disk, mark as failed for now
device                  /dev/hda2
failed-disk             1

# md1 is the /boot array
raiddev                 /dev/md1
raid-level              1
nr-raid-disks           2
chunk-size              32
# Spare disks for hot reconstruction
nr-spare-disks          0
persistent-superblock   1
device                  /dev/hdc1
raid-disk               0
# boot is marked failed as well
device                  /dev/hda1
failed-disk               1
```

## 4.3 Create, format, and configure RAID

Create the md devices with the commands:

```
    mkraid /dev/md0
    mkraid /dev/md1
```

The raid devices should be created and start. Examination of /proc/mdstat should show the raid personalities in the kernel and the raid devices running.

Format the boot and root devices with:

```
mke2fs /dev/md0
mke2fs /dev/md1
```

Mount the new root device somewhere handy and create the /boot directory and mount the boot partition.

```
mount /dev/md0 /mnt
mkdir /mnt/boot
mount /dev/md1 /mnt/boot
```

## 4.4 Copy the current OS to the new raid device

This is pretty straightforward.

```
cd /
# set up a batch file to do this
cp -a /bin /mnt
cp -a /dev /mnt
cp -a /etc /mnt
cp -a (all directories except /mnt, /proc, and nsf mounts) /mnt
```

This operation can be tricky if you have mounted or linked other disks to your root file system. The example above assumes a very simple system, you may have to modify the procedure somewhat.

## 4.5 Test your new RAID

Make a boot floppy and rdev the kernel.

```
dd if=kernal.image of=/dev/fd0 bs=2k
rdev /dev/fd0 /dev/md0
rdev -r /dev/fd0 0
rdev -R /dev/fd0 1
```

Modify the fstab on the RAID device to reflect the new mount points as follows:

```
/dev/md0        /       ext2    defaults        1 1
/dev/md1        /boot   ext2    defaults        1 1
```

Dismount the raid devices and boot the new file system to see that all works correctly.

```
umount /mnt/boot
umount /mnt
raidstop /dev/md0
raidstop /dev/md1
shutdown -r now
```

Your RAID system should now be up and running in degraded mode with a floppy boot disk. Carefully check that you transferred everything to the new raid system. If you mess up here without a backup, YOU ARE DEAD!

If something did not work, reboot your old system and go back and fix things up until you successfully complete this step.

## 4.6 Integrate old disk into raid array

Success in the previous step means that the raid array is now operational, but without redundancy. We must now re−partition the old drive(s) to fit into the new raid array. Remember that if the geometries are not the same, the the partition size on the old drive must be the same or larger than the raid partitions or they can not be added to the raid set.

Re−partition the old drive as required. Example:

```
    /dev/hda1      same or larger than /dev/hdc1
    /dev/hda2      same or larger than /dev/hdc2
    /dev/hda3      anything left over for swap or whatever...
```

Change the **failed−disk** parameter in the raidtab to **raid−disk** and hot add the new (old) disk partitions to the raid array.

```
    raidhotadd /dev/md1 /dev/hda1
    raidhotadd /dev/md0 /dev/hda2
```

Examining /proc/mdstat should show one or more of the raid devices reconstructing the data for the new partitions. After a minute or two... or so, the raid arrays should be fully synchronized (this could take a while for a large partition).

Using the procedure described in the first sections of this document, set up bootable raid on the new raid pair. Hang on to that boot floppy while setting up and testing this last step.

---

# 5.Appendix A. − example raidtab

RAID1 example described in the first sections of this document

```
 df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/md0              19510780   1763188  16756484  10% /
/dev/md1                 15860       984     14051   7% /boot

# -------------------------

 fdisk -ul /dev/hda

Disk /dev/hda: 16 heads, 63 sectors, 39770 cylinders
Units = sectors of 1 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1            63     33263     16600+  fd  Linux raid autodetect
/dev/hda2         33264    443519    205128   83  Linux native
/dev/hda3        443520  40088159  19822320   fd  Linux raid autodetect

# -------------------------

 fdisk -ul /dev/hdc
```

```
Disk /dev/hdc: 16 heads, 63 sectors, 39770 cylinders
Units = sectors of 1 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdc1            63     33263     16600+  fd  Linux raid autodetect
/dev/hdc2         33264    443519    205128   82  Linux swap
/dev/hdc3        443520  40088159  19822320   fd  Linux raid autodetect

# --------------------------

# md0 is the root array, about 20 gigs
raiddev                 /dev/md0
raid-level              1
nr-raid-disks           2
chunk-size              32
# Spare disks for hot reconstruction
nr-spare-disks          0
persistent-superblock   1
device                  /dev/hda3
raid-disk               0
device                  /dev/hdc3
raid-disk               1

# md1 is the /boot array, about 16 megs
raiddev                 /dev/md1
raid-level              1
nr-raid-disks           2
chunk-size              32
# Spare disks for hot reconstruction
nr-spare-disks          0
persistent-superblock   1
device                  /dev/hda1
raid-disk               0
device                  /dev/hdc1
raid-disk               1

# --------------------------

# GLOBAL SECTION
# device containing /boot directory
disk=/dev/md0
# geometry
  bios=0x80
  sectors=63
  heads=16
  cylinders=39770
# dummy
  partition=/dev/md1
# start of device "disk" above
  start=63

boot=/dev/hda
map=/boot/map
install=/boot/boot.b

image=/boot/bzImage
root=/dev/md0
label=LinuxRaid
read-only

# -------------------------
```

```
# GLOBAL SECTION
# device containing /boot directory
disk=/dev/md0
# geometry
  bios=0x80
  sectors=63
  heads=16
  cylinders=39770
# dummy
  partition=/dev/md1
# start of device "disk" above
  start=63

boot=/dev/hdc
map=/boot/map
install=/boot/boot.b

image=/boot/bzImage
root=/dev/md0
label=LinuxRaid
read-only
```

---

# 6.Appendix B. – SCSI reference implementation RAID5

4 disk SCSI RAID5

```
 df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/md0              11753770   2146076   9000678  19% /
/dev/md1                 15739       885     14042   6% /boot

# --------------------------

 fdisk -ul /dev/sda

Disk /dev/sda: 64 heads, 32 sectors, 4095 cylinders
Units = sectors of 1 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sda1            32     32767     16368   fd  Linux raid autodetect
/dev/sda2         32768    292863    130048    5  Extended
/dev/sda3        292864   8386559   4046848   fd  Linux raid autodetect
/dev/sda5         32800    260095    113648   82  Linux swap
/dev/sda6        260128    292863     16368   83  Linux native - test

# ------------------------

 fdisk -ul /dev/sdb

Disk /dev/sdb: 64 heads, 32 sectors, 4095 cylinders
Units = sectors of 1 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sdb1            32     32767     16368   fd  Linux raid autodetect
/dev/sdb2         32768    292863    130048    5  Extended
/dev/sdb3        292864   8386559   4046848   fd  Linux raid autodetect
/dev/sdb5         32800    260095    113648   82  Linux swap
```

```
/dev/sdb6        260128    292863     16368   83  Linux native - test

# ------------------------

# fdisk -ul /dev/sdc

Disk /dev/sdc: 64 heads, 32 sectors, 4095 cylinders
Units = sectors of 1 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sdc2            32    292863    146416    5  Extended
/dev/sdc3        292864   8386559   4046848   fd  Linux raid autodetect
/dev/sdc5            64    260095    130016   83  Linux native - development
/dev/sdc6        260128    292863     16368   83  Linux native - test

# ------------------------

 fdisk -ul /dev/sdd

Disk /dev/sdd: 64 heads, 32 sectors, 4095 cylinders
Units = sectors of 1 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sdd2            32    292863    146416    5  Extended
/dev/sdd3        292864   8386559   4046848   fd  Linux raid autodetect
/dev/sdd5            64    260095    130016   83  Linux native - development
/dev/sdd6        260128    292863     16368   83  Linux native - test

# --------------------------

# raidtab
#
raiddev /dev/md0
        raid-level      5
        nr-raid-disks   4
        persistent-superblock 1
        chunk-size      32

# Spare disks for hot reconstruction
        nr-spare-disks  0
        device          /dev/sda3
        raid-disk       0
        device          /dev/sdb3
        raid-disk       1
        device          /dev/sdc3
        raid-disk       2
        device          /dev/sdd3
        raid-disk       3

# boot partition
#
raiddev /dev/md1
        raid-level      1
        nr-raid-disks   2
        persistent-superblock 1
        chunk-size      32

# Spare disks for hot reconstruction
        nr-spare-disks  0
        device          /dev/sda1
        raid-disk       0
        device          /dev/sdb1
```

```
        raid-disk       1

# --------------------------

# cat lilo.conf.sda
# GLOBAL SECTION
# device containing /boot directory
disk=/dev/md0
# geometry
  bios=0x80
  sectors=32
  heads=64
  cylinders=4095
# dummy
  partition=/dev/md1
# start of device "disk" above
  start=32

boot=/dev/sda
map=/boot/map
install=/boot/boot.b

image=/boot/bzImage
root=/dev/md0
label=LinuxRaid
read-only

# ------------------------
# cat lilo.conf.sdb
# GLOBAL SECTION
# device containing /boot directory
disk=/dev/md0
# geometry
  bios=0x80
  sectors=32
  heads=64
  cylinders=4095
# dummy
  partition=/dev/md1
# start of device "disk" above
  start=32

boot=/dev/sdb
map=/boot/map
install=/boot/boot.b

image=/boot/bzImage
root=/dev/md0
label=LinuxRaid
read-only
```

---

# 7.Appendix C. − ide RAID10 with initrd

RAID1 over striped RAID0 pair.... the disks in the RAID0 sets are not quite the same size, but close enough.

```
/dev/md0 is the /boot partition and is autostarted by the kernel
```

```
/dev/md1 and /dev/md3 are the two RAID0 sets autostarted by the kernel
/dev/md2 is the root partition and is started by initrd

df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/md2                118531     76485     35925  68% /
/dev/md0                  1917      1361       457  75% /boot

# ---------------------------

 fdisk -ul /dev/hda

Disk /dev/hda: 4 heads, 46 sectors, 903 cylinders
Units = sectors of 1 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1            46      4231      2093   fd  Linux raid autodetect
/dev/hda2          4232    166151     80960   fd  Linux raid autodetect

# ---------------------------

 fdisk -ul /dev/hdb

Disk /dev/hdb: 5 heads, 17 sectors, 981 cylinders
Units = sectors of 1 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdb1            17     83384     41684   fd  Linux raid autodetect

# ---------------------------

 fdisk -ul /dev/hdc

Disk /dev/hdc: 7 heads, 17 sectors, 1024 cylinders
Units = sectors of 1 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdc1            17     84013     41998+  fd  Linux raid autodetect
/dev/hdc2         84014    121855     18921   82  Linux swap

# ---------------------------

 fdisk -ul /dev/hdd

Disk /dev/hdd: 4 heads, 46 sectors, 903 cylinders
Units = sectors of 1 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdd1            46      4231      2093   fd  Linux raid autodetect
/dev/hdd2          4232    166151     80960   fd  Linux raid autodetect

# ---------------------------

# raidtab
#
raiddev /dev/md0
        raid-level      1
        nr-raid-disks   2
        persistent-superblock   1
        chunk-size      8
        device          /dev/hda1
        raid-disk       0
```

```
        device          /dev/hdd1
        raid-disk       1

raiddev /dev/md1
        raid-level      0
        nr-raid-disks   2
        persistent-superblock   1
        chunk-size      8
        device          /dev/hdd2
        raid-disk       0
        device          /dev/hdb1
        raid-disk       1

raiddev /dev/md2
        raid-level      1
        nr-raid-disks   2
        persistent-superblock   1
        chunk-size      8
        device          /dev/md1
        raid-disk       0
        device          /dev/md3
        raid-disk       1

raiddev /dev/md3
        raid-level      0
        nr-raid-disks   2
        persistent-superblock   1
        chunk-size      8
        device          /dev/hda2
        raid-disk       0
        device          /dev/hdc1
        raid-disk       1

# ----------------------------

contents of linuxrc

 cat linuxrc
#!/bin/sh
# ver 1.02 2-22-00
#
############# really BEGIN 'linuxrc' ###############
#
# mount the proc file system
/bin/mount /proc

# start raid 1 made of raid 0's
/bin/raidstart /dev/md2

# tell the console what's happening
/bin/cat /proc/mdstat

# Everything is fine, let the kernel mount /dev/md2
# tell the kernel to switch to /dev/md2 as the /root device
# The 0x900 value is the device number calculated by:
#  256*major_device_number + minor_device number
echo "/dev/md2 mounted on root"
echo 0x902>/proc/sys/kernel/real-root-dev

# umount /proc to deallocate initrd device ram space
/bin/umount /proc
exit
```

```
# ----------------------------

contents of initrd

./bin/ash
./bin/echo
./bin/raidstart
./bin/mount
./bin/umount
./bin/cat
./bin/sh
./dev/tty1
./dev/md0
./dev/md1
./dev/md2
./dev/md3
./dev/md4
./dev/console
./dev/hda
./dev/hda1
./dev/hda2
./dev/hda3
./dev/hdb
./dev/hdb1
./dev/hdb2
./dev/hdb3
./dev/hdc
./dev/hdc1
./dev/hdc2
./dev/hdc3
./dev/hdd
./dev/hdd1
./dev/hdd2
./dev/hdd3
./dev/initrd
./dev/ram0
./dev/ram1
./dev/ram2
./dev/ram3
./dev/ram4
./dev/ram5
./dev/ram6
./dev/ram7
./etc/raidtab
./etc/fstab
./lib/ld-2.1.2.so
./lib/ld-linux.so.1
./lib/ld-linux.so.1.9.9
./lib/ld-linux.so.2
./lib/ld.so
./lib/libc-2.1.2.so
./lib/libc.so.6
./linuxrc
./proc
```

# 8.Appendix D. − ide RAID1−10 with initrd

This is a system made up of an assortment of odds and ends. The root mounted raid device is comprised of a RAID1 made up of one RAID0 array from odd sized disks and a larger regular disk partition. Examination of the lilo.conf files may give you better insight into the reasoning behind the various parameters.

```
/dev/md0 is the /boot partition and is autostarted by the kernel
/dev/md1 is one half of the mirror set for md2, autostarted by kernel
/dev/hda3 is the other half of the mirror set for md2
/dev/md2 is the RAID1 /dev/md1 + /dev/hda3, started by initrd

df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/md2                138381     74421     56815  57% /
/dev/md0                  2011      1360       549  71% /boot

# ----------------------------

 fdisk -ul /dev/hda

Disk /dev/hda: 8 heads, 46 sectors, 903 cylinders
Units = sectors of 1 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1            46      4415      2185   fd  Linux raid autodetect
/dev/hda2          4416     43423     19504   82  Linux swap
/dev/hda3         43424    332303    144440   83  Linux native

# ----------------------------

 fdisk -ul /dev/hdc

Disk /dev/hdc: 8 heads, 39 sectors, 762 cylinders
Units = sectors of 1 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdc1            39      4367      2164+  fd  Linux raid autodetect
/dev/hdc2          4368     70199     32916   82  Linux swap
/dev/hdc3         70200    237743     83772   fd  Linux raid autodetect

# ----------------------------

 fdisk -ul /dev/hdd

Disk /dev/hdd: 4 heads, 39 sectors, 762 cylinders
Units = sectors of 1 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdd1            39    118871     59416+  fd  Linux raid autodetect

# ----------------------------

# raidtab
#
raiddev /dev/md0
        raid-level      1
        nr-raid-disks   2
        persistent-superblock   1
        chunk-size      8
```

```
        device          /dev/hdc1
        raid-disk       1
        device          /dev/hda1
        raid-disk       0

raiddev /dev/md1
        raid-level      0
        nr-raid-disks   2
        persistent-superblock   1
        chunk-size      8
        device          /dev/hdc3
        raid-disk       0
        device          /dev/hdd1
        raid-disk       1

raiddev /dev/md2
        raid-level      1
        nr-raid-disks   2
        persistent-superblock   1
        chunk-size      8
        device          /dev/md1
        raid-disk       1
        device          /dev/hda3
        raid-disk       0

# ----------------------------

 cat linuxrc
#!/bin/sh
# ver 1.02 2-22-00
#
############# really BEGIN 'linuxrc' ###############
#
# mount the proc file system
/bin/mount /proc

# autostart /boot partition and raid0
/bin/raidstart /dev/md2

# tell the console what's happening
/bin/cat /proc/mdstat

# Everything is fine, let the kernel mount /dev/md2
# tell the kernel to switch to /dev/md2 as the /root device
# The 0x900 value is the device number calculated by:
#  256*major_device_number + minor_device number
echo "/dev/md2 mounted on root"
echo 0x902>/proc/sys/kernel/real-root-dev

# umount /proc to deallocate initrd device ram space
/bin/umount /proc
exit

# ----------------------------

contents of initrd.gz

./bin
./bin/ash
./bin/echo
./bin/raidstart
./bin/mount
```

```
./bin/umount
./bin/cat
./bin/sh
./dev/tty1
./dev/md0
./dev/md1
./dev/md2
./dev/md3
./dev/console
./dev/hda
./dev/hda1
./dev/hda2
./dev/hda3
./dev/hdc
./dev/hdc1
./dev/hdc2
./dev/hdc3
./dev/hdd
./dev/hdd1
./dev/hdd2
./dev/hdd3
./dev/initrd
./dev/ram0
./dev/ram1
./dev/ram2
./dev/ram3
./dev/ram4
./dev/ram5
./dev/ram6
./dev/ram7
./etc/raidtab
./etc/fstab
./lib/ld-2.1.2.so
./lib/ld-linux.so.1
./lib/ld-linux.so.1.9.9
./lib/ld-linux.so.2
./lib/ld.so
./lib/libc-2.1.2.so
./lib/libc.so.6
./linuxrc
./proc

# ----------------------------

 cat lilo.conf.hda
# GLOBAL SECTION
# device containing /boot directory
disk=/dev/md2
# geometry
  bios=0x80
  cylinders=903
  heads=8
  sectors=46
# geometry for 2nd disk
# bios will be the same because it will have to be moved to hda
#  cylinders=762
#  heads=8
#  sectors=39

# dummy
  partition=/dev/md0
# start of device "disk" above
```

```
  start=46
# second device
#  start=39

# seem to have some trouble with 2.2.14 recognizing the right IRQ
  append = "ide1=0x170,0x376,12 ether=10,0x300,eth0 ether=5,0x320,eth1"

boot=/dev/hda
map=/boot/map
install=/boot/boot.b

initrd=/boot/initrd.gz

image=/boot/zImage
root=/dev/md2
label=LinuxRaid
read-only

# ----------------------------

 cat lilo.conf.hdc
# GLOBAL SECTION
# device containing /boot directory
disk=/dev/md2
# geometry
  bios=0x80
#  cylinders=903
#  heads=8
#  sectors=46
# geometry for 2nd disk
# bios will be the same because it will have to be moved to hda
  cylinders=762
  heads=8
  sectors=39

# dummy
  partition=/dev/md0
# start of device "disk" above
#  start=46
# second device
  start=39

# seem to have some trouble with 2.2.14 recognizing the right IRQ
  append = "ide1=0x170,0x376,12 ether=10,0x300,eth0 ether=5,0x320,eth1"

boot=/dev/hdc
map=/boot/map
install=/boot/boot.b

initrd=/boot/initrd.gz

image=/boot/zImage
root=/dev/md2
label=LinuxRaid
read-only
```