

Software RAID troubleshoot

Once, after upgrading my desktop slackware64 from 14.1 to 14.2 (with kernel upgrade too), i be done with system, who,after lilo menu, write "loading kernel"

and then stops completely - nothing more.

Initial configuration was:

Intel DG965SS motherboard, core 2 duo 2.2 GHz E4500 CPU, 8 Gb RAM,
2 x 1000 Gb Seagate SATA HDD (as sda and sdb)

ST1000DM003-1CH1

dvd-writer on sata4 port

both seagate discs is partitioned as FD type (linux autodetect raid) and 4 partitions (mbr type) -

- 100 Gb root (md1)
- 2 Gb swap (md2)
- 350 Gb /home (md3)
- 550 Gb /Second (md4)

cat /proc/mdstat :

```
Personalities : [linear] [raid0] [raid1] [raid10] [raid6] [raid5] [raid4]
[multipath]
md1 : active raid1 sda1[] sdb1[1]
      104857536 blocks [2/2] [UU]

md2 : active raid1 sda2[] sdb2[1]
      2097088 blocks [2/2] [UU]

md3 : active raid1 sda3[] sdb3[1]
      367001536 blocks [2/2] [UU]

md4 : active raid1 sda4[] sdb4[1]
      502805120 blocks [2/2] [UU]

unused devices: <none>
```

mdadm -Es :

```
ARRAY /dev/md1 UUID=7cc47bea:832f8260:208cdb8d:9e23b04b
ARRAY /dev/md2 UUID=cce81d3a:78965aa5:208cdb8d:9e23b04b
ARRAY /dev/md3 UUID=f0bc71fc:8467ef54:208cdb8d:9e23b04b
ARRAY /dev/md4 UUID=3f4daae2:cbf37a2a:208cdb8d:9e23b04b
```

```
# for p in 1 2 3 4; do mdadm --create /dev/md$p --name=$p --level=1 --raid-  
devices 2 /dev/sda$p /dev/sdb$p --metadata=0.90; done
```

My "fall" and "sucess" story

I have slackware64 14.1 system with raid1 on two discs.

no any mdadm.conf configuration, no any initrd -i use "huge" kernel, and all just works right.

Then i do massive system update via slackpkg update-all, including kernel update too.

check lilo.conf, restart - all looks ok. then i decide to upgrade system to 14.2 via the same slackpkg (looks like live on macos is too boring, too predictable - all work, and so on...:D)

always, i check lilo.conf, check the new kernel is named right, have no old kernel for backup -

only one entry in lilo (who was not good thing at all!), and do reboot.

Then all interesting things start!!! :)

I have LiLo menu, kernel start loading...

it show lots of "..." but then all stop, and nothing more do.

That was indicated some problems with lilo updating, i suppose.

I want to boot, and re-run lilo -v

As so, i booting from Slackware64 Live CD <http://bear.alienbase.nl/mirrors/slackware-live/> from AlienBob, and try mount my root partition for run lilo again.

But - there was a big problem!

there no my /dev/md1, /dev/md2, /dev/md3 and /dev/md4 after i load via slackware live CD!

and my lilo.conf was that:

```
# LILO configuration file  
# generated by 'liloconfig'  
#  
# Start LILO global section  
# Append any additional kernel parameters:  
append=" vt.default_utf8=1"  
boot = /dev/sda  
  
#compact          # faster, but won't work on all systems.  
  
# Boot BMP Image.  
# Bitmap in BMP format: 640x480x8  
  bitmap = /boot/slack.bmp  
# Menu colors (foreground, background, shadow, highlighted  
# foreground, highlighted background, highlighted shadow):  
  bmp-colors = 255,0,255,0,255,0  
# Location of the option table: location x, location y, number of  
# columns, lines per column (max 15), "spill" (this is how many  
# entries must be in the first column before the next begins to  
# be used. We don't specify it here, as there's just one column.  
  bmp-table = 60,6,1,16
```

```
# Timer location x, timer location y, foreground color,  
# background color, shadow color.  
  bmp-timer = 65,27,0,255  
  
# Standard menu.  
# Or, you can comment out the bitmap menu above and  
# use a boot message with the standard menu:  
#message = /boot/boot_message.txt  
  
# Wait until the timeout to boot (if commented out, boot the  
# first entry immediately):  
prompt  
# Timeout before the first entry boots.  
# This is given in tenths of a second, so 600 for every minute:  
timeout = 1200  
# Override dangerous defaults that rewrite the partition table:  
change-rules  
  reset  
# Normal VGA console  
vga = normal  
# Ask for video mode at boot (time out to normal in 30s)  
#vga = ask  
# VESA framebuffer console @ 1024x768x64k  
#vga=791  
# VESA framebuffer console @ 1024x768x32k  
#vga=790  
# VESA framebuffer console @ 1024x768x256  
#vga=773  
# VESA framebuffer console @ 800x600x64k  
#vga=788  
# VESA framebuffer console @ 800x600x32k  
#vga=787  
# VESA framebuffer console @ 800x600x256  
#vga=771  
# VESA framebuffer console @ 640x480x64k  
#vga=785  
# VESA framebuffer console @ 640x480x32k  
#vga=784  
# VESA framebuffer console @ 640x480x256  
#vga=769  
# End LILO global section  
# Linux bootable partition config begins  
image = /boot/vmlinuz  
  root = /dev/md1  
  label = Linux  
  read-only  
# Linux bootable partition config ends
```

so i do some research, do

```
dmesg | grep md
```

and from array size found out, what number was my root partition (it was 100 gb size). it was md126.
i mount it:

```
mount /dev/md126 /mnt/hd
```

Then i run mc, and check, i was really mount there my root partition.

Then i do in console :

```
chroot /mnt/hd /sbin/lilo -v 3
```

but - lilo command end with error - it was cannot find root partition - /dev/md1.

That was problem, because now /dev/md1 was become as /dev/md126 for whatever reason.
Then, i do some reading about RAID subsystems, forums and so on, made some mistakes and experiments, who resulted on these shortcuts:

I made array assembling strings in mdadm.conf via do these comands in terminal:

```
mdadm -Db /dev/md127 >> /mnt/hd/etc/mdadm.conf  
mdadm -Db /dev/md126 >> /mnt/hd/etc/mdadm.conf  
mdadm -Db /dev/md125 >> /mnt/hd/etc/mdadm.conf  
mdadm -Db /dev/md124 >> /mnt/hd/etc/mdadm.conf
```

In a result i get something like that in end of mdadm.conf:

```
ARRAY /dev/md125 metadata=0.90 UUID=7cc47bea:832f8260:208cdb8d:9e23b04b  
#this one is 100 gb partition: \ (md1)  
ARRAY /dev/md124 metadata=0.90 UUID=f0bc71fc:8467ef54:208cdb8d:9e23b04b #  
375 gb partition - \home - md3  
ARRAY /dev/md126 metadata=0.90 UUID=3f4daae2:cbf37a2a:208cdb8d:9e23b04b #  
514 Gb partition - \Second - md4  
ARRAY /dev/md127 metadata=0.90 UUID=cce81d3a:78965aa5:208cdb8d:9e23b04b # 2  
Gb partition - swap - md2
```

then, based on :

```
dmesg | grep md
```

and /mnt/hd/etc/fstab i found out which md12x must be md1, md2, md3 and md4, and write it there
after # as shown above.

Then i edit it to become in right way:

```
ARRAY /dev/md1 metadata=0.90 UUID=7cc47bea:832f8260:208cdb8d:9e23b04b
ARRAY /dev/md2 metadata=0.90 UUID=cce81d3a:78965aa5:208cdb8d:9e23b04b
ARRAY /dev/md3 metadata=0.90 UUID=f0bc71fc:8467ef54:208cdb8d:9e23b04b
ARRAY /dev/md4 metadata=0.90 UUID=3f4daae2:cbf37a2a:208cdb8d:9e23b04b
```

also i wrote in mdadm.conf that string, just to be sure, hostname do not affect raid naming:

```
HOMEHOST <ignore>
```

then i made mdadm_stop_127.scr script:

```
#!/bin/sh
echo "stopping md127"
mdadm --stop /dev/md127

echo "stopping md126"
mdadm --stop /dev/md126

echo "stopping md125"
mdadm --stop /dev/md125

echo "stopping md124"
mdadm --stop /dev/md124

##mdadm --assemble --scan
#-As
```

I copy it to livesystem root, copy also my edited mdadm.conf from /mnt/hd/etc/mdadm.conf to livesystem /etc, and umount /dev/md125 !

And then i run it - my mdadm_stop_127.scr

I see, all that arrays be stopped, and then i run:

```
mdadm -As
```

then i see

```
cat /proc/mdstat
```

I see, all my RAID become as it must be - /dev/md1, md2, md3, and md4!

then i do mount my root hdd again:

```
mount /dev/md1 /mnt/hd
```

and do:

```
chroot /mnt/hd /sbin/lilo -v 3
```

all looks good. i do restart:

```
shutdown -r now
```

(or press ctrl +alt+ del)

after restart i see, system start loading, and went after previously dead point. i get to login screen, log in as root, and see, there is md1 (root) and md2 (swap), but no md3 and md4 (instead of it i have these /home and /Second arrays as md125 and md124).

Thats look very strange and unlogical, as all raid arrays var create about same time, and was similar - but half of it get right numbers, and half - not. now i try different thing - disable raid autodetect on kernel, before root fs is mounted, and mdadm.conf is available for md module:

i restart machine, press tab on LiLo prompt, and use kernel parameters:

```
Linux raid=noautodetect md=1,/dev/sda1,/dev/sdb1
```

that says kernel not to autodetect raid arrays, but assemble md1 raid (md=1) from /dev/sda1 and /dev/sdb1 partition, because without root kernel system cant start.

as i load system in that way, all looks right - there was /dev/md1, md2, md3 and md4.

then i just do system restart, without any kernel parameters, and all again going to be right - md1 till md4.

looks like, system writes something in RAID arrays superblock, or metadata, or something like that, about previously gived md name, because, if not, after restart i must get again situation as previously - with md1, md2, md125 and ,d124...

that was, in a most, all of story. yet, there is some another workarounds of that situation.

Workarounds for incorrect raid devices naming

1. Using UUID in lilo (i do not check this), and in fstab for mounting partitions.

do a

```
ls /dev/disk/by-uuid/
```

or better, go in that location with midnight commander, and youl see, there is a "files" named as

numbers - that was the raid array disk uuid - and symlink to /dev/mdx.

for automate info feed into fstab you can use this way:

```
cd /dev/disk/by-uuid

ls -d -l $PWD/* >> /etc/fstab
```

after that you must immediately edit /etc/fstab and make in a right way, otherwise you may have problems with mounting in next boot...

```
/dev/md2          swap          swap          defaults      0    0
/dev/md1          /             ext4          defaults      1    1
##/dev/md3        /home        ext4          defaults      1    2
/dev/disk/by-uuid/ef92814a-2db1-4d47-8d70-4c5a8d56e287 /home
ext4             defaults      1    2
/dev/md4          /Second      ext4          defaults      1    2
#/dev/cdrom       /mnt/cdrom   auto          noauto,owner,ro,comment=x-
gvfs-show 0      0
/dev/fd0          /mnt/floppy  auto          noauto,owner  0    0
devpts           /dev/pts     devpts        gid=5,mode=620 0    0
proc             /proc        proc          defaults      0    0
tmpfs            /dev/shm     tmpfs         defaults      0    0
```

take a note! disk UUID by

/dev/disk/by-uuid/

and that one, who you get via

```
mdadm -D
mdadm -Db
mdadm -Es
```

differ, not the same!!! in fstab (lilo too?) you must use UID from /dev/disk/by-uuid/ !

1. 2. Using initramd.

```
#
# mkinitrd_command_generator.sh revision 1.45
#
# This script will now make a recommendation about the command to use
# in case you require an initrd image to boot a kernel that does not
# have support for your storage or root filesystem built in
# (such as the Slackware 'generic' kernels').
# A suitable 'mkinitrd' command will be:

#/usr/share/mkinitrd/

mkinitrd -c -k 3.2.29 -f ext4 -r /dev/md1 -m mbcache:jbd2:ext4 -R -u -o
```

```
/boot/initrd.gz
```

rightly edited mdadm.conf then must be copied in /boot/tree??? before you run this mkinitrd conf.

after you run that mkinitrd, you must update lilo.

Useful commands in this case

show raid array info:

```
mdadm -Es
```

Assemble RAID array based on mdadm.conf

```
mdadm -As
```

show array info:

```
mdadm -D /dev/md127
```

show defined array another info:

```
mdadm -Db /dev/md127
```

show scsi devices info:

```
lsscsi
```

show assembled raid arrays status:

```
cat /proc/mdstat
```

show UUID info about discs (or RAID arrays) in system:

```
ls /dev/disk/by-uuid/
```

```
dmesg |grep md
```

re-run lilo, when booted from another source.

```
chroot /mnt/hd /sbin/lilo -v 3
```

stop named RAID array:


```
mdadm --stop /dev/md127
```

kernel options:

```
$kernelname raid=noautodetect md=1, /dev/sda1,/dev/sdb1
```

turn on not to autodetect RAID arrays, and define raid array /dev/md1, from two partitions (members?)

Useful Links:

- <http://www.linuxquestions.org/questions/slackware-14/repair-lilo-on-software-raid1-4175593663/>
- https://bugzilla.redhat.com/show_bug.cgi?id=606481
- <https://www.linux.org.ru/forum/admin/13033496?lastmod=1479927790872> (in russian)
- https://raid.wiki.kernel.org/index.php/Linux_Raid
- <https://www.kernel.org/doc/Documentation/md.txt>

Sources

Originally written by — [John Ciemgals](#) 2016/11/28 04:50

Rewritten with used materials from “Links” and LinuxQuestions.org Slackware forum, especially user bassmadrigal and bormant from linux.org.ru help — [John Ciemgals](#) 2016/11/28 09:15

[software](#), [raid](#), [raid1](#), [/dev/md127](#), [enumeration](#), [broken](#), [linux](#), [slackware](#), [author](#) [wisedraco](#)

From:

<http://docs.slackware.com/> - **SlackDocs**

Permanent link:

http://docs.slackware.com/howtos:misc:software_raid_troubleshoot_howto

Last update: **2017/06/26 16:08 (BST)**

