

# Demystifying the **GR**and **U**nified **B**ootloader aka Getting to know **GRUB**

`grub > initrd /initrd.img`

`grub > kernel /vmlinuz`



`grub > chainloader +1`

`grub > geometry (hd0)`

# Topics To Be Covered



- Framework for GRUB: Hard Drive Topology and POST
- GRUB and the Master Boot Record
- GRUB Stages 1.5 and 2.0
- Booting from a GRUB Prompt
  - Linux
    - Non-Linux OSes
- Adding Bling to Your GRUB Menu
- Future of GRUB

# Resources



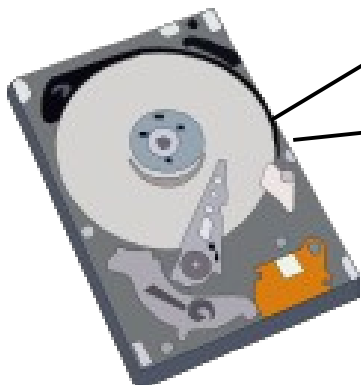
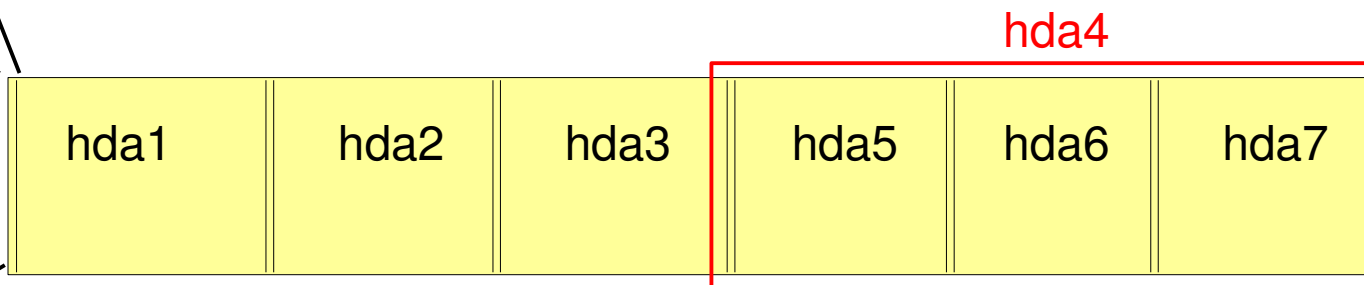
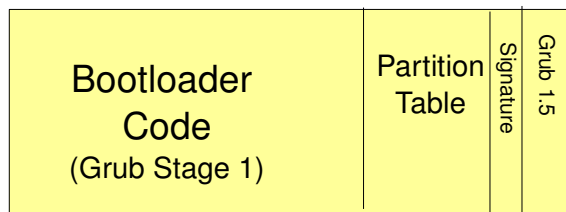
- [Gnu.Org](#)
- [www.linuxselfhelp.com](http://www.linuxselfhelp.com)
- [IBM DeveloperWorks](#)
- <http://lxr.linux.no>
- [LinuxQuestions.org](http://LinuxQuestions.org)
- <http://mirror.href.com>
- [wikipedia](#)
- [linuxbasics.org](http://linuxbasics.org)

# Hard Drive Topology



A typical hard drive with multiple partitions:

1. Can have up to four primary partitions
2. Logical partitions reside in an extended partition
3. Bootloader precedes first partition



# In The Beginning – System BIOS



After you power on the computer, the BIOS performs several tasks:

- Checks the hardware for physical faults (Power On Self Test--POST).
- Identifies and checks peripherals, i.e. keyboard, mouse, etc.
- Searches for bootable devices based on BIOS settings.
- Looks for a Master Boot Record (MBR) at the first sector.
- Loads MBR content into memory, then passes control to it.

# The Master Boot Record (MBR)



Things to know about the Master Boot Record (MBR):

- Holds GRUB Stage 1 code.
- Occupies the first 512 bytes of the booting device:
  - The MBR program code starts at offset 0000.
  - The MBR messages start at offset 008b.
  - The partition table starts at offset 01be.
  - The signature is at offset 01fe.
- Contains instructions on how to load the boot-loader.
- Contains a special signature to indicate its end.
- Activates or loads the boot-loader, which takes over the process (if the boot-loader is installed in the MBR).

# MBR & GRUB Stage 1



```
file:///home/mephotep/mbr.bin - KHexEdit
File Edit View Documents Bookmarks Tools Settings Help
Hex Hexadecimal ASCII
0000:0000 eb 48 90 00 00 00 47 52 55 42 16 07 69 7e 38 46 0H...GRUB.i~8F
0000:0010 00 00 00 00 93 45 38 46 ad 72 98 6b 81 00 80 60 .....E8F0r.k...
0000:0020 7d 23 90 08 b8 c0 07 8e d0 bc 00 08 fb 52 53 06 }#...00...0RS.
0000:0030 56 fc 8e d8 31 ed 60 b8 00 12 b3 36 cd 10 03 02 V0.010"0.060...
0000:0040 ff 00 00 20 01 00 00 00 02 fa 90 90 f6 c2 80 .0...00...0.00.
0000:0050 75 02 b2 80 ea 59 7c 00 00 31 c0 8e d8 8e d0 bc u.0.0Y|.10.0.00
0000:0060 00 20 fb a0 40 7c 3c ff 74 02 88 c2 52 be 7f 7d .00|<0t.0R0.}
0000:0070 e8 34 01 f6 c2 80 74 54 b4 41 bb aa 55 cd 13 5a 04.00.tT0A00V0.Z
0000:0080 52 72 49 81 fb 55 aa 75 43 a0 41 7c 84 c0 75 05 RrI.0V0u0C0A|.0u.
0000:0090 83 e1 01 74 37 66 8b 4c 10 be 05 7c c6 44 ff 01 .0.t7f.L.0.|000.
0000:00a0 66 8b 1e 44 7c c7 04 10 00 c7 44 02 01 00 66 89 f..D|0...0D...f.
0000:00b0 5c 08 c7 44 06 00 70 66 31 c0 89 44 04 66 89 44 \.0D..pf10.D.f.D
0000:00c0 0c b4 42 cd 13 72 05 bb 00 70 eb 7d b4 08 cd 13 .0B0.r.0.p0}0.0.
0000:00d0 73 0a f6 c2 80 0f 84 ea 00 e9 8d 00 be 05 7c c6 s.00...0.0.0.|0
0000:00e0 44 ff 00 66 31 c0 88 f0 40 66 89 44 04 31 d2 88 00.f10.0ef.D.10.
0000:00f0 ca c1 e2 02 88 e8 88 f4 40 89 44 08 31 c0 88 d0 000..0@.D.10.0
0000:0100 c0 e8 02 66 89 04 66 a1 44 7c 66 31 d2 66 f7 34 00.f..f00|f10f04
0000:0110 88 54 0a 66 31 d2 66 f7 74 04 88 54 0b 89 44 0c .T.f10f0t...T..D.
0000:0120 3b 44 08 7d 3c 8a 54 0d c0 e2 06 8a 4c 0a fe c1 ;D.>.T.00..L.00
0000:0130 08 d1 8a 6c 0c 5a 8a 74 0b bb 00 70 8e c3 31 db .0.l.Z.t.0.p.010
0000:0140 b8 01 02 cd 13 72 2a 8c c3 8e 06 48 7c 60 1e b9 00..r*.0.H|.0
0000:0150 00 01 8e db 31 f6 31 ff fc f3 a5 1f 61 ff 26 42 ...01010000.a0&B
0000:0160 7c be 85 7d e8 40 00 eb 0e be 8a 7d e8 38 00 eb |0.}0@.0.0.}08.0
0000:0170 06 be 94 7d e8 30 00 be 99 7d e8 2a 00 eb fe 47 .0.}00.0.}0*.00G
0000:0180 52 55 42 20 0d 47 65 6f 6d 00 48 61 72 64 20 44 RUB .Geom.Hard D
0000:0190 69 73 6b 00 52 65 61 64 00 20 45 72 72 6f 72 00 isk.Read. Error.
0000:01a0 bb 01 00 b4 0e cd 10 ac 3c 00 75 f4 c3 00 00 00 00.0.0.0<.u00...
0000:01b0 00 00 00 00 00 00 00 ad 72 98 6b 00 00 80 00 .....0r.k....
0000:01c0 41 29 07 fe ff ff e9 cd 48 00 ac 31 20 02 00 01 A).00000H.01 ...
0000:01d0 01 00 12 fe 7f 28 3f 00 00 00 aa cd 48 00 00 fe ...0.(?...00H..0
0000:01e0 ff ff 05 fe ff ff 56 42 64 03 6b a2 ec 05 00 fe 00.000V8d.k00..0
0000:01f0 ff ff 83 fe ff ff 95 ff 68 02 c1 42 fb 00 55 aa 00.000.0h.0B0.V0
0000:0200
```

The Master Boot Record is 512 bytes in size:

- 1.(000000 - 0001bd): GRUB Stage 1 boot code (446 bytes long)
- 2.(0001be - 0001fd) : Partition table (64 bytes long)
- 3.(0001fe - 0001ff): The boot code signature of 55aa which is 2 bytes long identifies the end of the MBR. If the BIOS can't find this marker, the MBR does not get loaded.

## FYI: Extracting the MBR

```
# dd if=/dev/hda of=mbr.bin bs=512 count=1
# od -xa mbr.bin
```

# GRUB – Stage 1 Errors



Four errors that can occur during stage 1:

- **Geometry:** The location of the stage2 or stage1.5 is not in the portion of the disk supported directly by the BIOS read call
- **Hard Disk:** The stage2 or stage1.5 is being read from a hard disk, and the attempt to determine the size and geometry of the hard disk failed.
- **Floppy:** Same as above, but issue is on the boot floppy.
- **Read Error:** A disk-read error happened while trying to read the stage 2 or stage1.5.s.

```
89 44 \.D..pf1.D.f.D
cd 13 .B.r.p}
7c c6 s...|.
d2 88 D.f1.@f.D.1.
88 d0 ...@.D.1.
f7 34 .f.fD|f1f4
44 0c .T.f1ft..T.D.
fe c1 ;D.<.T...L.
31 db .l.Z.t.p.1
1e b9 ...r*.H|^
26 42 ...110000.a&B
00 eb |.}@.}.8.
fe 47 .}.0.}.*..G
20 44 RUB .Geom.Hard D
72 00 isk.Read.Error.
00 00 ...<.u...
80 00 .....rk....
00 01 A).0000H.01 ...
00 fe ....(?...H..
00 fe ..VBd.k...
55 aa ..h.B.V
```



# GRUB – Stage 1.5



## What is Grub 1.5

- Read into memory by the Stage 1 boot loader, if necessary.
- Required by some hardware as an intermediate step to get to the Stage 2 boot loader.
- Located in the space immediately following the MBR.
- Knows about the boot files system; opens the file system on the specified partition.
- Looks for Stage 2 executable.

## What Can Go Wrong (aka Stage 1.5 errors)

- Uses the same error list as stage 2.
- Will appear as 'Error #' with no accompanying error description (unlike stage 2). Example: If GRUB is unable to mount the designated partition:
  - Stage 1.5 error readout: Error 17
  - Stage 2 error readout: Error 17: Cannot mount selected partition

# GRUB – Stage 2

## (What's on the Menu?)



- Opens menu.lst file in /boot/grub directory of the partition to be booted (or grub.conf on redhat).

```
Ubuntu, kernel 2.6.22-10-generic
Ubuntu, kernel 2.6.22-10-generic (recovery mode)
Ubuntu, kernel 2.6.20-16-generic
Ubuntu, kernel 2.6.20-16-generic (recovery mode)
Ubuntu, kernel 2.6.20-15-generic
Ubuntu, kernel 2.6.20-15-generic (recovery mode)
Ubuntu, memtest86+
```

Use the ↑ and ↓ keys to select which entry is highlighted.  
Press enter to boot the selected OS, 'e' to edit the  
commands before booting, or 'c' for a command-line.

- Reads instructions and if no manual intervention, boots specified kernel.
- Instructions contain pieces of information required to boot: root, kernel, and initrd.

# GRUB – What's on the menu



```
title      Ubuntu 8.04.1, kernel 2.6.24-19-generic
root       (hd0,2)
kernel     /boot/vmlinuz-2.6.24-19-generic
           root=UUID=2762ce06-0457-41bb-9ea9-af99fd5eb172 ro quiet splash
initrd     /boot/initrd.img-2.6.24-19-generic
quiet
```

## Other common commands for the GRUB menu

- **Default** – The menu selection that will boot if no human interaction occurs. Numbering starts at 0.
- **Fallback** -- if the default boot entry has any errors, instead of waiting for the user to do something, immediately start over using the num entry (same numbering as the default command (see default)). This obviously won't help if the machine was rebooted by a kernel that GRUB loaded. You can specify multiple fallback entry numbers.
- **Hiddenmenu** – hides the grub menu from view
- **Timeout** – Amount of time GRUB waits until booting the default.
- **Color** – Customizes grub menu colors
- **Splashimage** – Adds background image to GRUB menu display

# Booting from grub >



Generally, GRUB can boot any Multiboot-compliant OS using the following commands:

- Set GRUB's root device to the drive where the OS images are stored.

```
grub> root (hd0,2)
```

- Load the kernel image, set root partition.

```
grub> kernel /vmlinuz root=/dev/sda3
```

- Load the initial ramdisk, if needed.

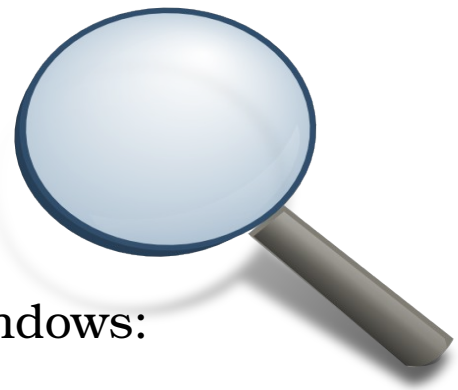
```
grub> initrd /initrd.img
```

- If you need modules, load them via module or modulenounzip.

- Boot the operating system (boot).

```
grub> boot
```

# Booting Other OSes



GRUB can be used to boot other operating systems such Windows:

- Set GRUB's root device to the partition  
**grub> rootnoverify (hd0,0)**
- Set the active flag in the partition  
**grub> makeactive**
- Load the boot loader  
**grub> chainloader +1**
- Boot the operating system  
**grub> boot**
- Booting Windows<sup>®</sup> from second hard disk:  
**grub> root (hd1,0)**  
**grub> map (hd0) (hd1)**  
**grub> map (hd1) (hd0)**  
**grub> makeactive**  
**grub> chainloader +1**  
**grub> savedefault**

# Change GRUB Partition



If you install several distributions, it's quite possible that the pointer to the partition holding GRUB stage 2 could change:

```
grub> find /boot/grub/stage1
(hd0,2)
(hd0,6)
(hd0,7)
(hd0,9)
```

```
grub> root (hd0,2)
```

```
grub> setup (hd0)
```

```
Checking if "/boot/grub/stage1" exists... yes
```

```
Checking if "/boot/grub/stage2" exists... yes
```

```
Checking if "/boot/grub/e2fs_stage1_5" exists... yes
```

```
Running "embed /boot/grub/e2fs_stage1_5 (hd0)"... 17 sectors  
are embedded.
```

```
succeeded
```

```
Running "install /boot/grub/stage1 (hd0) (hd0)1+17 p
```

```
(hd0,2)/boot/grub/stage2 /boot/grub/menu.lst"... succeeded
```

```
Done.
```

# Adding Basic Bling To GRUB



- You can add color to the basic GRUB menu by adding this line :

**color fore-color/back-color fore-color/back-color**

- Syntax of this line consists of three elements: 1) command; 2) Normal color (fore/back); 3) highlighted color (fore/back)
- Available background colors:
  - black
  - green
  - red
  - brown
  - blue
  - cyan
  - magenta
  - light-gray
- These below can be specified only for the foreground.
  - dark-gray
  - light-green
  - light-red
  - yellow
  - light-blue
  - light-cyan
  - light-magenta
  - white
- You can prefix blink- to foreground for a blinking foreground color.

# Adding More Bling to GRUB



- GRUB can be configured to display an image when the menu is displayed.
- 
- Creating a nifty grub splash image requires a few simple steps:
  - Find your image
  - Reformat it to the following parameters:
  - Resize it to 640px X 480px (width and height respectively) using gimp or other image editor
  - Change color format to indexed with a maximum of 14 (gimp: Image > Mode > Indexed Colors. In the Maximum number of colors box, set to 14. Click OK)
  - Save as an '.xpm' file, e.g. grubsplashpic.xpm
- Implementing a nifty grub splash image.
  - Compress it using gzip. (`gzip -c mygrubsplash.xpm > mygrubsplash.xpm.gz`)
  - Move it to the `/boot/grub` directory.
  - Modify `menu.lst` (or `grub.conf`) to point to your splash image: `splashimage /boot/grub/splashimages/FILENAME.xpm.gz`



# Adding Bling to GRUB



Before



# Adding Bling to GRUB



After



# Tools for GRUB Menu



- Editing by hand using a text editor:
  - vi
  - emacs
  - kate
  - kwrite
  - nano
  
- Grub menu also can be edited graphically:
  - QGRUBEditor ([qt-apps.org](http://qt-apps.org))
  - kgrubeditor ([kde-apps.org](http://kde-apps.org))

# The Future of GRUB



Will be replaced by GRUB2. (The original GRUB to be called GRUB Legacy)

- Replacement of Stage 1.5 with a compact core image
- Support for dynamic loading to the core image
- Trend towards making the overall GRUB framework object oriented
- Support for internationalization, such as non-ASCII character sets
- Support for different hardware architectures and different platforms (other than Linux)