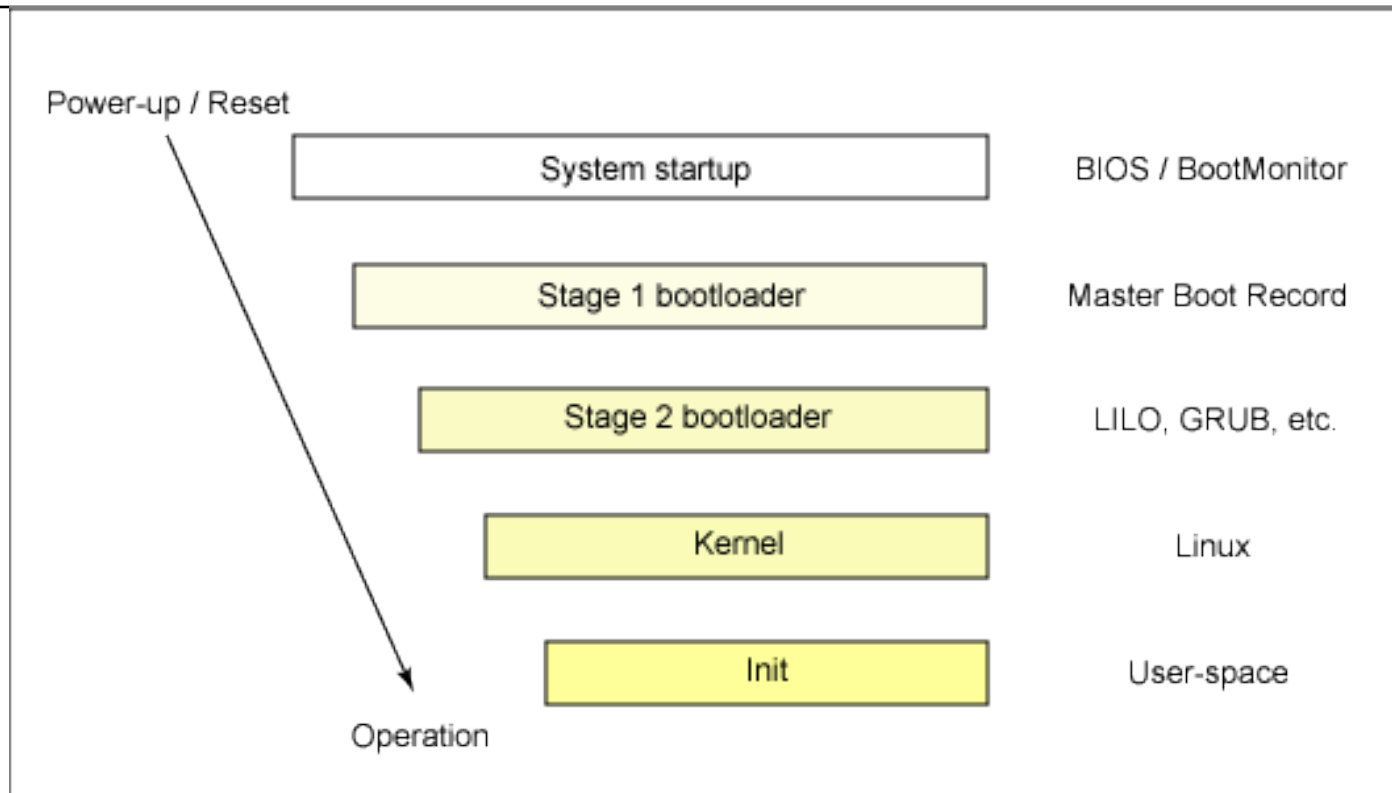




Linux Booting Procedure

How Linux boot?



What is “boot”?

- boot (*n.*)



[1] <http://en.wikipedia.org/wiki/Boot>

Brief etymology^[2]

- Phrase “pull oneself up by one’s bootstraps”
 - Misattributed (at latest in 1901!) to “The Surprising Adventures of Baron Munchausen” (1781, Rudolf Erich Raspe) : The baron pulls himself out of a swamp by his *hair* (pigtail).
 - The use of this phrase is found in 1834 in the U.S.
 - “[S]omeone is attempting or has claimed some ludicrously far-fetched or impossible task”
- In the 20th century, the “*possible task*” meaning has appeared
 - “To begin an enterprise or recover from a setback without any outside help; to succeed only on one's own effort or abilities”



Bootstrapping (in Computer)

- The process of loading the basic software (typically, operating systems) into the main memory from persistent memory (HDD, flash ROM, etc.)
- “Boot” is an abbreviation for “bootstrap(ping)”





System startup

Booting sequence

1. Turn on
2. CPU jump to address of BIOS (0xFFFF0)
3. BIOS runs POST (Power-On Self Test)
4. Find bootable devices
5. Loads and execute boot sector from MBR
6. Load OS

BIOS (Basic Input/Output System)

- BIOS refers to the software code run by a computer when first powered on
- The primary function of BIOS is code program embedded on a chip that recognises and controls various devices that make up the computer.



BIOS on board



BIOS on screen



How computer startup?

- Booting is a bootstrapping process that starts operating systems when the user turns on a computer system
- A boot sequence is the set of operations the computer performs when it is switched on that load an operating system

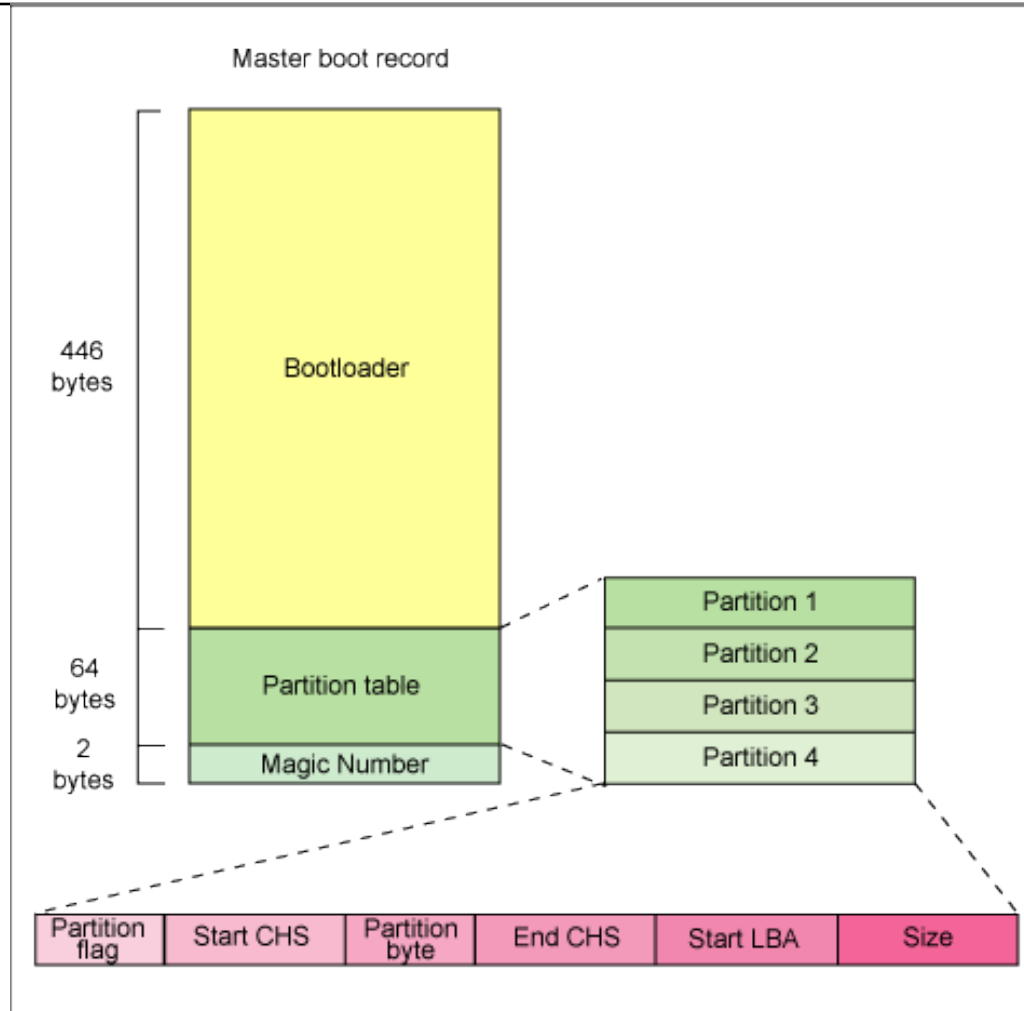


Boot loader

MBR (Master Boot Record)

- OS is booted from a hard disk, where the Master Boot Record (MBR) contains the primary boot loader
- The MBR is a 512-byte sector, located in the first sector on the disk (sector 1 of cylinder 0, head 0)
- After the MBR is loaded into RAM, the BIOS yields control to it.

MBR (Master Boot Record)



MBR (Master Boot Record)

- The first 446 bytes are the primary boot loader, which contains both executable code and error message text
- The next sixty-four bytes are the partition table, which contains a record for each of four partitions
- The MBR ends with two bytes that are defined as the magic number (0xAA55). The magic number serves as a validation check of the MBR

Boot loader

- Boot loader could be more aptly called the kernel loader. The task at this stage is to load the Linux kernel
- Optional, initial RAM disk
- GRUB and LILO are the most popular Linux boot loader.

Other boot loader (Several OS)

- bootman
- GRUB
- LILO
- NTLDR
- XOSL
- BootX
- loadlin
- Gujin
- Boot Camp
- Syslinux
- GAG

Boot loader

- “It is responsible for loading and transferring control to the operating system *kernel* software (such as [the Hurd](#) or Linux).”^[4]
- Boot loader
 - BIOS (PC)
 - UEFI (Universal Extensible Firmware Interface) (PC)
 - “Secure Boot” issue
 - Das U-Boot (Universal bootloader) (for embedded systems)
- Second-stage boot loader
 - LILO (Linux Loader, Ver. 24.0, Released on Jun 7, **2013**)
 - Supports GPT and RAID (!?)
 - GRUB2 (Ver. 2.00, Jun 26, 2012)
 - Supports BIOS and UEFI boot
 - GRUB Legacy (Grand Unified Boot Loader, Ver. 0.97, May 8, 2005)
 - ELILO (EFI Linux Boot Loader, Ver 3.16, Mar 29, 2013)
 - Originally for EFI and Itanium; currently bug fix only
 - SYSLINUX (Ver. 6.02, Oct 13, 2013)
 - NTLDR, BOOTMGR (beginning from Windows Vista)

[4] <http://www.gnu.org/software/grub/>



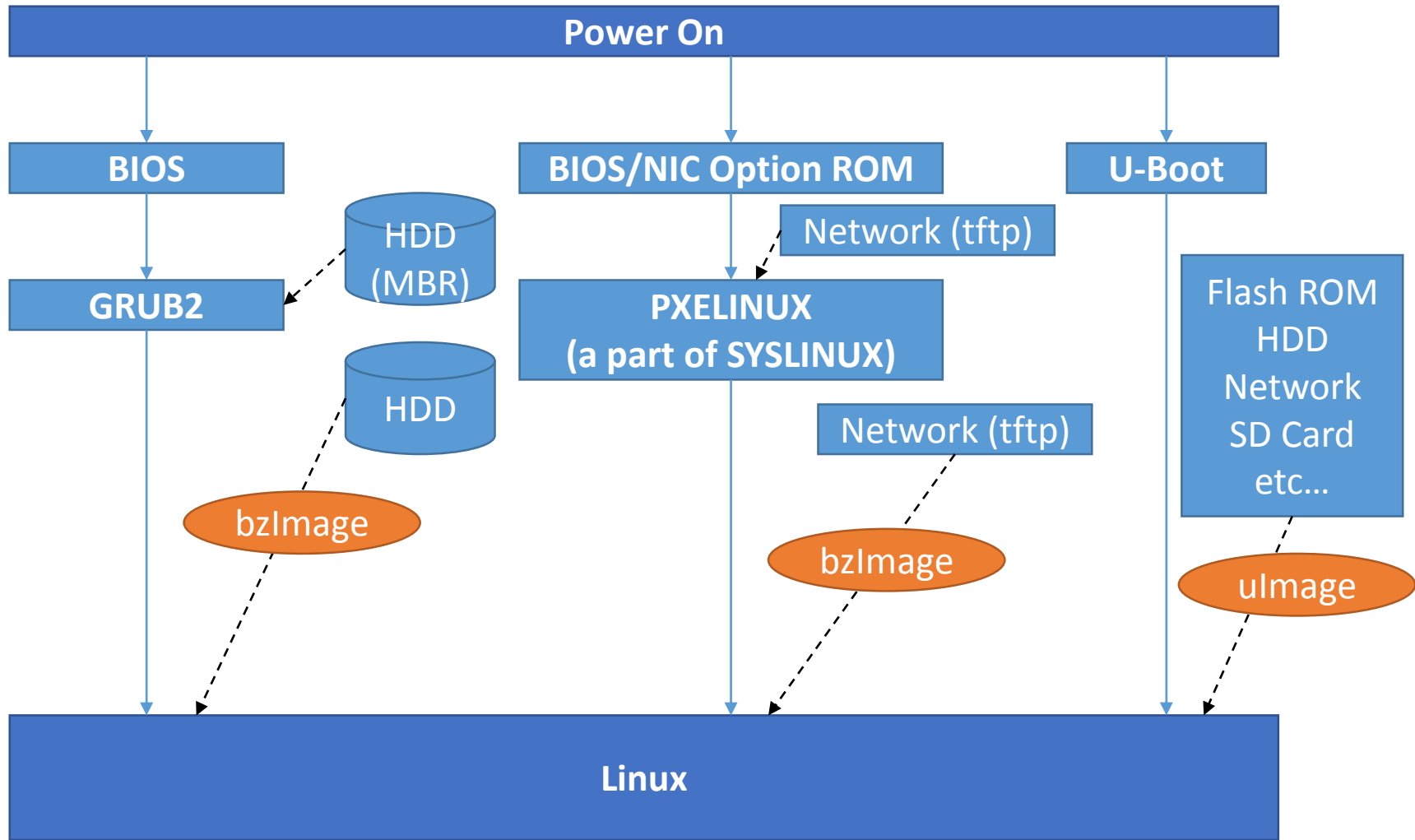
GRUB: GRand Unified Bootloader

- GRUB is an operating system independent boot loader
- A multiboot software packet from GNU
- Flexible command line interface
- File system access
- Support multiple executable format
- Support diskless system
- Download OS from network
- Etc.

GRUB boot process

1. The BIOS finds a bootable device (hard disk) and transfers control to the master boot record
2. The MBR contains GRUB stage 1. Given the small size of the MBR, Stage 1 just load the next stage of GRUB
3. GRUB Stage 1.5 is located in the first 30 kilobytes of hard disk immediately following the MBR. Stage 1.5 loads Stage 2.
4. GRUB Stage 2 receives control, and displays to the user the GRUB boot menu (where the user can manually specify the boot parameters).
5. GRUB loads the user-selected (or default) kernel into memory and passes control on to the kernel.

What loads and what is loaded



GRUB

- boot.img (512 byte)
 - Usually located in the first sector (MBR) in HDD
 - Loaded at 0x7c00 by BIOS
 - Real-mode
 - Loads the next sector from HDD
 - The position is embedded by the GRUB installer (in sector, blue part)
 - Typically, at Sector 1 (the next sector)
- core.img
 - Located at the gap sectors between MBR and the first partition
 - The first partition begins at the 63rd sector (traditionally) or at 1MB (recently, as seen in right)
 - The first sector in core.img loads the remaining part of core.img from HDD

Jump!

```
# dd if=/dev/vda count=1 bs=512 2> /dev/null | od -t x1 -A x
000000 eb 63 90 00 00 00 00 00 00 00 00 00 00 00 00
000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
*
000030 00 00 00 00 00 00 00 00 00 00 00 00 00 03 02
000040 ff 00 00 20 01 00 00 00 00 02 fa 90 90 f6 c2 80
000050 75 02 b2 80 ea 59 7c 00 00 31 00 80 01 00 00 00
000060 00 00 00 00 ff fa 90 90 f6 c2 80 74 05 f6 c2 70
000070 74 02 b2 80 ea 79 7c 00 00 31 c0 8e d8 8e d0 bc
000080 00 20 fb a0 64 7c 3c ff 74 02 88 c2 52 be 80 7d
000090 e8 17 01 be 05 7c b4 41 bb aa 55 cd 13 5a 52 72
0000a0 3d 81 fb 55 aa 75 37 83 e1 01 74 32 31 c0 89 44
0000b0 04 40 88 44 ff 89 44 02 c7 04 10 00 66 8b 1e 5c
0000c0 7c 66 89 5c 08 66 8b 1e 60 7c 66 89 5c 0c c7 44
0000d0 06 00 70 b4 42 cd 13 72 05 bb 00 70 eb 76 b4 08
0000e0 cd 13 73 0d f6 c2 80 0f 84 d8 00 be 8b 7d e9 82
0000f0 00 66 0f b6 c6 88 64 ff 40 66 89 44 04 0f b6 d1
000100 c1 e2 02 88 e8 88 f4 40 89 44 08 0f b6 c2 c0 e8
000110 02 66 89 04 66 a1 60 7c 66 09 c0 75 4e 66 a1 5c
000120 7c 66 31 d2 66 f7 34 88 d1 31 d2 66 f7 74 04 3b
000130 44 08 7d 37 fe c1 88 c5 30 c0 c1 e8 02 08 c1 88
000140 d0 5a 88 c6 bb 00 70 8e c3 31 db b8 01 02 cd 13
000150 72 1e 8c c3 60 1e b9 00 01 8e db 31 f6 bf 00 80
000160 8e c6 fc f3 a5 1f 61 ff 26 5a 7c be 86 7d eb 03
000170 be 95 7d e8 34 00 be 9a 7d e8 2e 00 cd 18 eb fe
000180 47 52 55 42 20 00 47 65 6f 6d 00 48 61 72 64 20
000190 44 69 73 6b 00 52 65 61 64 00 20 45 72 72 6f 72
0001a0 0d 0a 00 bb 01 00 b4 0e cd 10 ac 3c 00 75 f4 c3
0001b0 00 00 00 00 00 00 00 00 0e 14 50 70 00 00 00 20
0001c0 21 00 83 35 37 3e 00 08 00 00 00 38 0f 00 00 35
0001d0 38 3e 82 51 60 31 00 40 0f 00 00 98 3b 00 00 51
0001e0 61 31 83 fe ff ff 00 d8 4a 00 00 10 7e 03 00 fe
0001f0 ff ff 05 fe ff ff fe ef c8 03 02 08 37 15 55 aa
000200
```

Boot sector signature



Kernel

Kernel image

- The kernel is the central part in most computer operating systems because of its task, which is the management of the system's resources and the communication between hardware and software components
- Kernel is always store on memory until computer is tern off
- Kernel image is not an executable kernel, but a compress kernel image
- zImage size less than 512 KB
- bzImage size greater than 512 KB



Task of kernel

- Process management
- Memory management
- Device management
- System call

References

https://www.slideshare.net/DhavalKaneria/linux-booting-procedure-32339908?from_action=save

<https://www.slideshare.net/shimosawa/linux-kernel-booting-process-1-for-nlkb>