Booting
PC style

Nov 2008

St. Louis UNIX User's Group

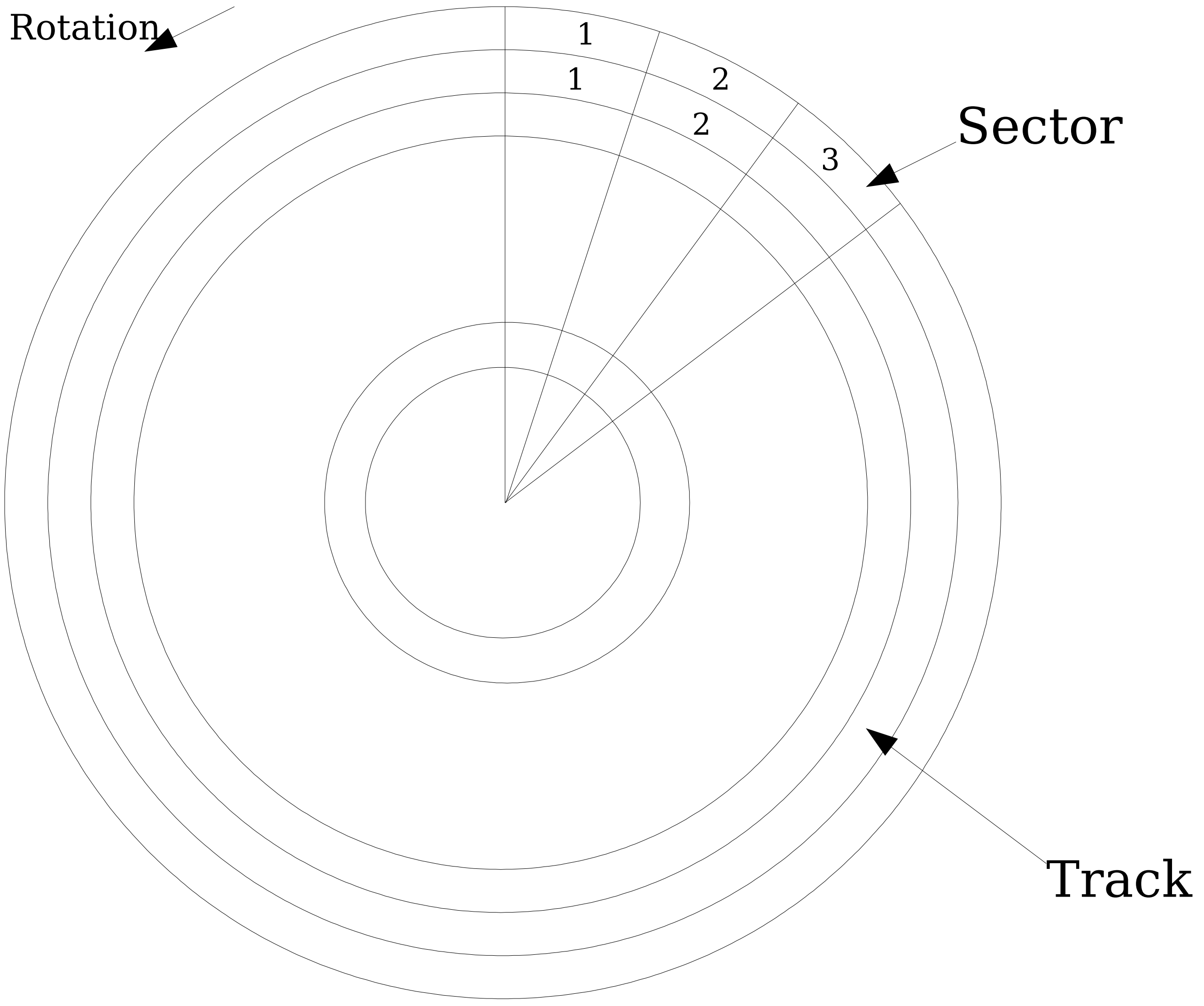
TOC

Constraints
Layout of Disk
Addressing on Disk
Partition Table
MBR
Designation of Disks
Boot Sequence
Assumptions Loaders make about Disk Layout
Stage 1 boots
Stage 2 boots
Loaders
PXE
What else is there?

Constrants

- Only consider x86 PC Hardware
- Only consider PC capable of running DOS or Windows
- Only common BIOS types are considered
- Must be backward compatible

Layout of Disk

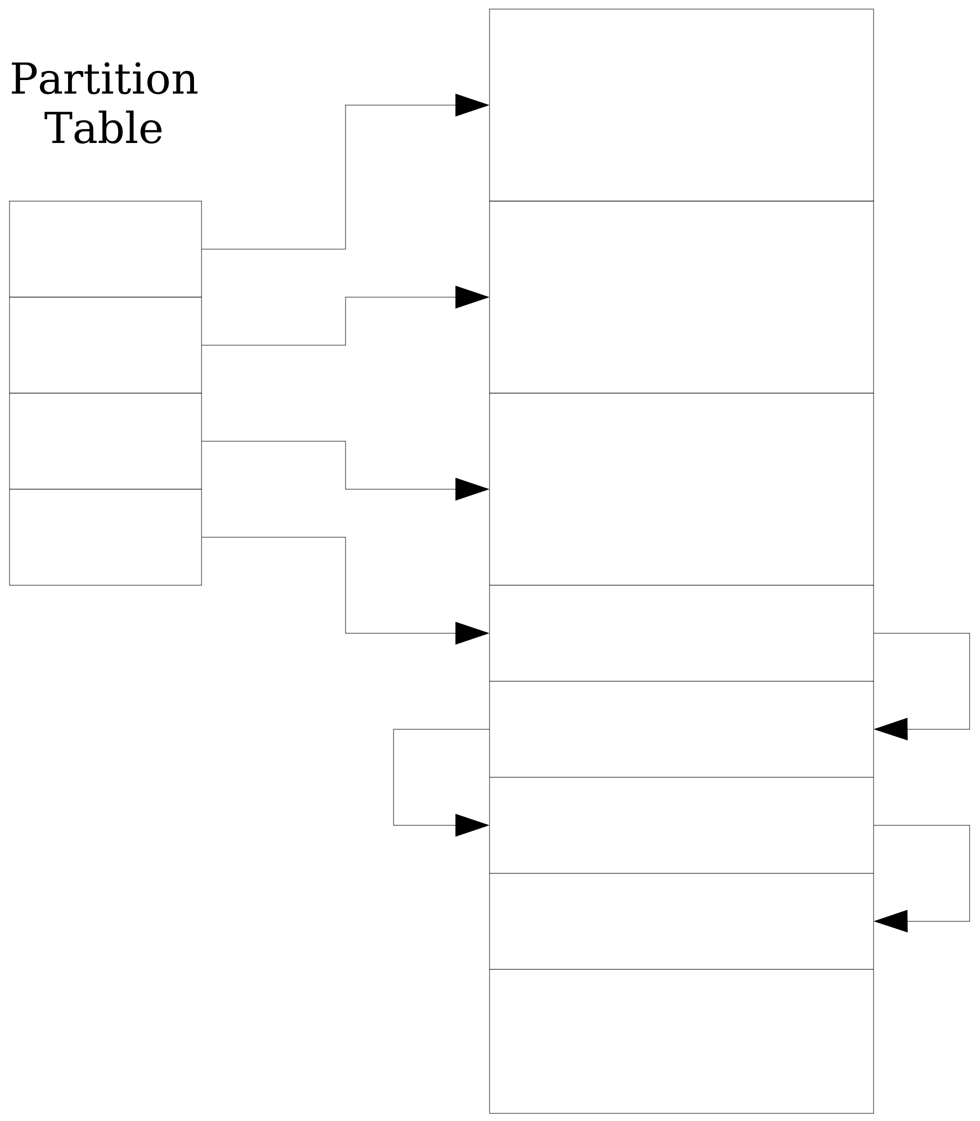


Track 0 is on the outside of a hard disk or floppy, on the inside of a CD or DVD.

Addressing on Disk

- CHS – Cylinder-Head-Section addressing
  - Cylinder = All Heads in the same Track
  - Track = 0-1023
  - Head = 0-255
  - Sector = 1-63
- LBA – Logical Block Addressing
  - One number 0-n addressing all sectors on the disk in order
- Disk Geometry
  - The specific number of Cylinders, Heads, and Sectors on a specific disk
  - Used to convert between CHS and LBA

Partition Table



1-4 Primary partitions

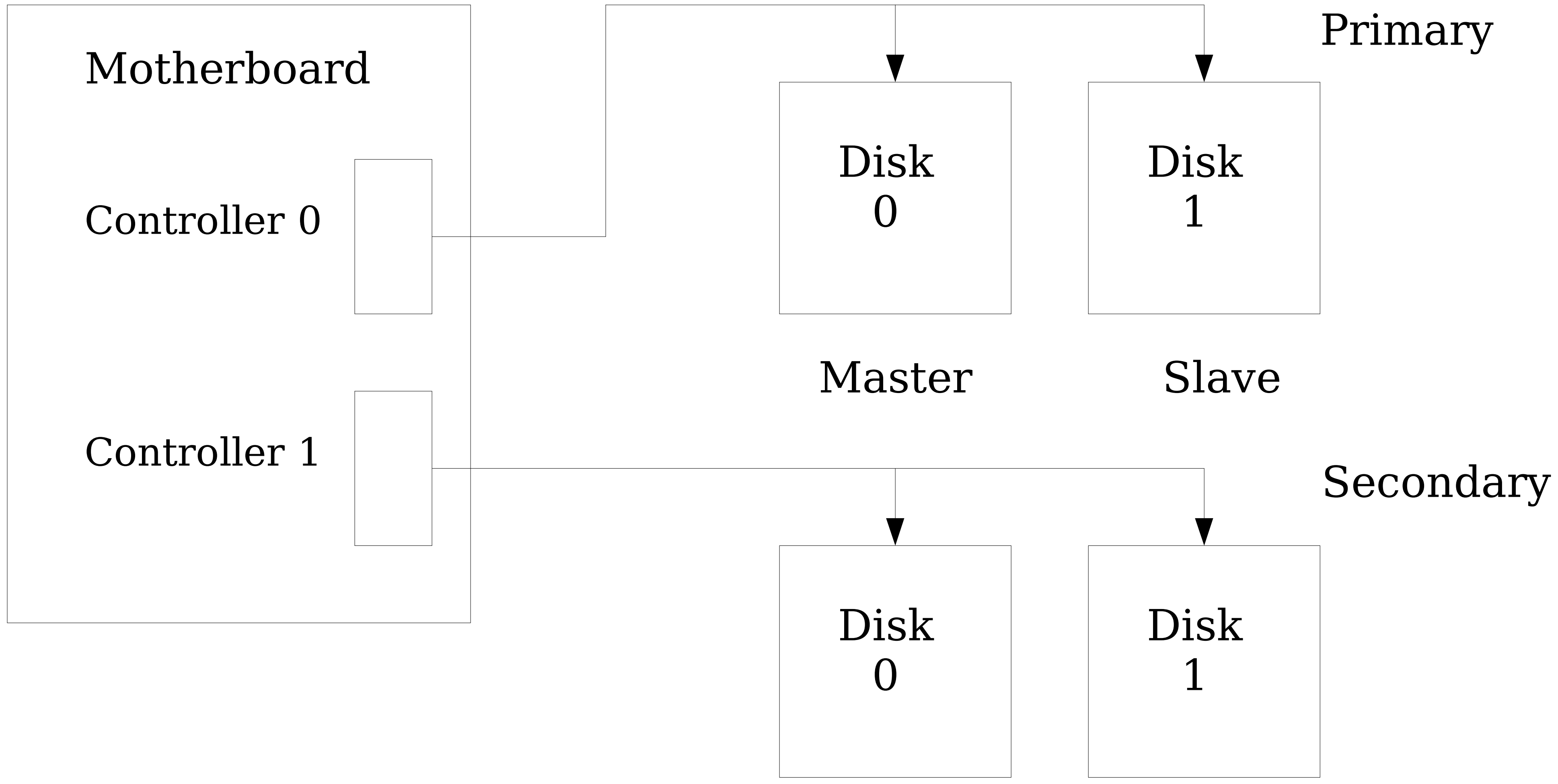
or

1-3 Primary partitions
plus
n Extended partitions
linked together

MBR

- Master Boot Record
  - Cylinder=0, Head=0, Sector=1 (LBA=0)
  - Contains
    - 1st stage boot (446 bytes)
    - Partition table
    - MBR Signature (0xaa55, on disk 55 aa)

Designation of Disks



Boot Sequence

- Normal boot sequence
  - BIOS checks
  - Search for bootable media
  - 1st stage boot from MBR
  - 2nd stage boot from partition (optional)
  - Loader (3rd stage) (optional)
- 1st and 2nd stages use BIOS for I/O and run in “real” mode (max addr = 1MB)

Loader's Assumptions

- Some loaders make assumptions about how the disk partition program lays out the disk
- Windows/Linux (FDISK)
  - Track(Cyl) 0 not used except for MBR
  - Primary Partition starts on Track(Cyl) boundary
- FreeBSD (other)
  - Partitions can start anywhere (LBA 1)

Stage 1 boot

- DOS/Windows
  - Loads 1 sector of the First track/cylinder of the Primary partition marked as active
- FreeBSD
  - Offers the user choice of which Primary Partition of which disk via a function key. Uses the previous choice after 5 seconds.
- VxWorks
  - Loads first sector of next track.

Stage 1 boot (cont)

- LILO, Grub, SystemCommander
  - Loads 2nd stage from second sector of track 0 which then extends itself into a full loader by reading up to 23 sectors of track 0
  - LILO, assumes the partitions are formatted ext2 and OS kernels are Linux, or will load second stage from any other Primary partition, called chain loading.
  - Grub and System Commander, like LILO except understand more file system and OS kernel types

Stage 2 boot

- DOS
  - Understands Windows file system formats (at least FAT)
  - Loads IO.SYS and MSDOS.SYS
- VxWorks
  - Understands FAT12 and FAT16 only
  - Loads BOOTROM.SYS
  - BOOTROM.SYS is a loader; it loads and initializes a complete VxWorks system
- LILO, Grub, SystemCommander & Windows
  - Loads same as in stage 1

Loaders

- The final target can be an OS image (usually compressed)
  - LILO – only understands Linux images, on pre-defined partition
  - Grub – understands a couple more images, but not FreeBSD
  - SystemCommander – understands even more
  - VxWorks BOOTROM.SYS – only understands VxWorks kernels and provides some kernel initialization (VxWorks kernels are not completely self initializating)
  - NTLDR – Windows loader (similar to LILO)

PXE

- PXE – Pre-eXecution Environment
  - Network booting, called “pixie”
  - Requires special ROM on network cards
  - Uses DHCP (bootp) to get information
  - Uses tftp (bootp) to get loader
  - BIOS is usually not used for actual boot

What else is there?

- EFI – Extensible Firmware Interface
  - Intel originated (Intel Boot Initiative)
  - Unified EFI (UEFI) Forum specification
  - First used with Intel for Itanium
  - Later used by:
    - HP for Itanium 2
    - Microsoft for 64-bit Windows
    - Apple for Intel-based Macintosh

EFI Framework

- OS, Boot Manager, Device Drivers in ROM
  - Controlled by well-known EFI variables
- GUID disk partition table
- \EFI\ disk partition, Fat32
- Applications on disk
  - Loaders
  - Shell
- Implementations are vendor specific