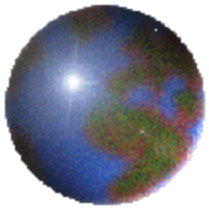
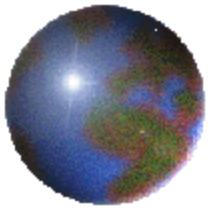
Operating Systems

System Booting Sequence



Biju K Raveendran, Dept. CS/IS,

BITS Pilani K K BIRLA Goa Campus



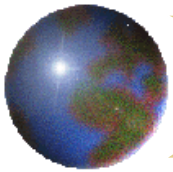
System Booting Sequence

The content of this presentation is from

www.dewassoc.com/kbase/hard_drives/master_boot_record.htm

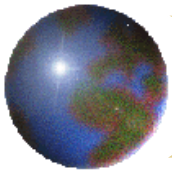
What happens when you power on

- The system BIOS starts the computer running when you turn it on
- Internal power supply turns on and initializes (takes half second approx)
  - “PowerGood” signal.
  - Until this signal is sent, the motherboard will refuse to start up the computer.



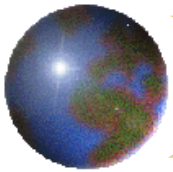
- Processor starts up and looks at preprogrammed place (BIOS ROM, FFFF0h) for the BIOS boot program.
- Why is it in FFFF0h?
  - Size of the ROM can be changed without creating compatibility problem.
  - Next 16bytes (end of conventional memory) contain a jump telling the processor where to go and execute BIOS startup program.
- BIOS performs Power On Self Test (POST).
  - If any problem Boot process stops
  - Beep pattern can be used for diagnosis

- 1 Beep tone - DRAM refresh failure
- 2 Beep tones - DRAM Parity failure
- 3 Beep tones - Base 64K RAM failure
- 4 Beep tones - System timer error
- 5 Beep tones - CPU failure
- 6 Beep tones - Keyboard controller error
- 7 Beep tones - Virtual mode error
- 8 Beep tones - Display memory read/write error
- 9 Beep tones - ROM BIOS checksum error
- 10 Beep tones - CMOS register read/write error
- 11 Beep tones - Cache memory error
- Continuous Beep tone - Memory Failure, Video Memory or Video Card Failure



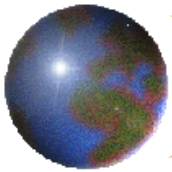
BIOS functionality …

- If POST is successful BIOS calls INT 19 & proceed to look for devices attached to the mother board.
- BIOS looks for video card's built in BIOS program (C000h)and run it.
  - This initializes the video card.
- BIOS then looks for other devices' ROM to see if any of them have BIOSes.
  - Floppy drive 0000:7C00.
  - IDE/ATA hard disk BIOS will be found at C8000h and executed



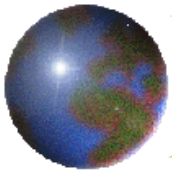
BIOS functionality continues…

- BIOS displays its startup screen
  - The BIOS Manufacturer and Version Number
  - BIOS Date
  - Setup Program Key (Del or F2)
  - System Logo (company logo)
  - "Energy Star" Logo
  - BIOS Serial Number (located bottom of the screen).
- BIOS does more tests like Memory count up test, system inventory test, memory timing (based on what memory it finds) and displays messages (error) in the screen.
- If the BIOS supports the Plug and Play standard
  - Will detect and configure Plug and Play devices at this time



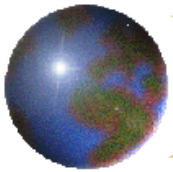
BIOS functionality continues…

- BIOS will display a summary screen about system configuration
  - Processor (CPU) Type,
  - Coprocessor(modern processors have built in, value can be Installed or Integrated),
  - Clock Speed,
  - Floppy Drive A, Floppy Drive B,
  - IDE/ATA Drives,
  - IDE system (IDE (ATAPI) CD-ROMs),
  - Base Memory Size (almost always will be 640K, it is called Conventional memory),
  - Extended Memory Size (The BIOS usually will not report the upper memory area that is reserved for the BIOS ROM and other hardware adapters)



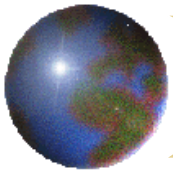
BIOS functionality continues…

- Cache Size
- Memory Type and Configuration ("EDO DRAM at Bank 1")
- Display Type ("VGA/EGA")
- Serial Port(s)
- Parallel Port(s)
- Plug and Play Devices (Plug and Play extension cards).



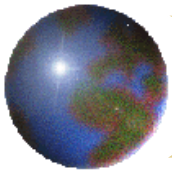
BIOS functionality continues…

- BIOS begins to search for a drive to boot from
- Depending on boot sequence (specified in BIOS settings) system tries to start booting.
- After identifying its target boot drive
  - BIOS looks for boot information to start the operating system boot process.
  - If in hard disk it looks for master boot record at cylinder 0, head 0, sector 1
  - If in floppy disk, it looks at the same address on the floppy disk for a volume boot sector.



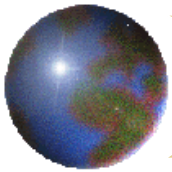
BIOS functionality continues…

- BIOS checks the 16-bit word at absolute address 07DFEh for AA55h.
  - Is the boot signature
  - Ensures the sector contains a valid boot sector.
  - If boot sector is valid, read that sector (512 bytes) from disk into memory at 0000:7C00h and interrupt 19h jumps there to start executing the code.
  - Boot sector code gets control and DL will be loaded with
    - 00h if the boot sector was loaded from drive A,
    - 80h if the boot sector was loaded from drive C



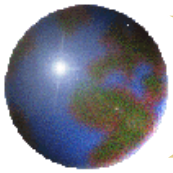
BIOS functionality continues…

- If BIOS finds what it looks for
  - Using the boot sector information it starts the process of booting OS.
- The code in the boot sector takes over from the BIOS.
- If no boot device found
  - System displays error message and then freeze up
  - Message can be "No boot device available" or "NO ROM BASIC - SYSTEM HALTED".
  - This will also happen if you have a bootable hard disk partition but forget to set it active.



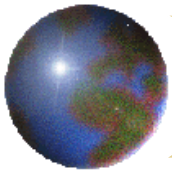
Cold boot and Warm boot

- This process is called a "cold boot"
- "warm boot" (Soft boot)
  - when the machine is rebooted using {Ctrl}+{Alt}+{Delete} or similar.
  - In this case the POST is skipped.



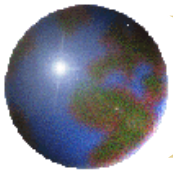
Master Boot Record (MBR)

- Every hard disk must have a consistent "starting point"
  - Stores key information about the disk (how many partitions it has, what sort of partitions they are…)
- The place where this information is stored is called the *master boot record* (*MBR, or master boot sector or boot sector*).
  - Always located at cylinder 0, head 0, and sector 1, the first sector on the disk
  - BIOS looks MBR for instructions and information on how to boot the disk and load the OS.



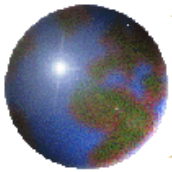
MBR Structure

- Master Partition Table
  - Contain description of partition in hard disk.
  - Can have maximum 4 partition information
  - So a hard disk can have only 4 true (Primary) partitions.
  - One of these 4 should be active partition (computer uses this for booting up)
- Master Boot Code
  - Contains the small initial boot program that the BIOS loads and executes to start the boot process.
  - This program eventually transfers control to the boot program stored on whichever partition is used for booting the PC.



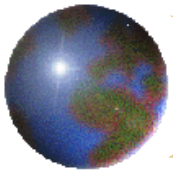
Important Addresses in MBR

- 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.
- The first byte of an active partition table entry is 80.
  - Loaded into the DL register before INT 13 is called to read the boot sector.
  - When INT 13 is called, DL is the BIOS device number.



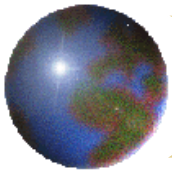
The DOS Boot Process

- Assuming BIOS finds a boot sector
  - The process of loading OS begins.
  - If the operating system is DOS (or Windows versions other than Windows NT or Windows 2000) the load sequence is called *DOS Boot Process*
- Difference between booting from HDD and floppy
  - The floppy disk's structures are slightly different.
  - Floppies cannot be partitioned, So no master boot record or partitions.
  - In the following explanation where the master boot record are searched are skipped.



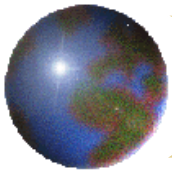
The DOS Boot Process

- The master boot code examines the master partition table.
  - It must determine if there is an extended DOS partition.
    - The extended partition can be logically partitioned to N logical partitions.
  - It must determine if there is a bootable (active) partition specified in the partition table.
- If extended partition found
  - Loads the extended partition table that describes the first logical volume in the extended partition.
  - It is examined to see if it points to another extended partition table.
  - This process is continued until all of the extended partitions have been loaded and recognized by the system.



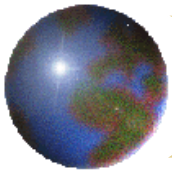
The DOS Boot Process

- After loading the extended partition information (if any)
  - The code attempts to boot the primary partition that is marked active (bootable).
    - If there are no partitions marked active, then the boot process will terminate with an error.
    - If there is a primary partition marked active, the code will boot it.
  - The volume boot sector is loaded into memory and tested, and the boot code that it contains is given control of the remainder of the boot process.

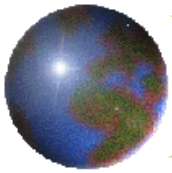


The DOS Boot Process

- The code searches the root directory
  - MS-DOS require "IO.SYS", "MSDOS.SYS" and "COMMAND.COM‟ to run.
  - If any of these files not found the boot program will display an error message like "Non-system disk or disk error - Replace and press any key when ready".
  - If found, the boot program will load them into memory and transfer control to them
  - First, IO.SYS is loaded and its code executed. IO.SYS will then execute MSDOS.SYS
  - Then loads COMMAND.COM and reads and interprets CONFIG.SYS and AUTOEXEC.BAT system control files.



- Depending on Partitions PC can be
  - Single Partition Windows PC
  - Multiple Partition Windows PC
  - Multiple Operating System PC
    - Could use one primary partition for each of up to 4 different file systems
    - Can combine multiple partitions with multiple OS
    - The extended DOS partition system allows you to have up to 24 disk partitions in a single system



- 2 main differences between Primary and Logical partition
  - primary partition can be set as bootable (active)
  - DOS assigns drive letters (C:, D: etc.) differently to primary and logical volumes
- *boot managers* or *boot loaders*
  - insert itself into the very beginning of the boot process
  - It analyzes the primary partitions on the disk and then presents a menu.
  - marks selected partition as active and continue boot process from there.

Entire MBR record in hex and ASCII

```
OFFSET 0 1 2 3  4 5 6 7  8 9 A B  C D E F  *0123456789ABCDEF*
000000 fa33c08e d0bc007c 8bf45007 501ffbfc *.3.....|..P.P...*
000010 bf0006b9 0001f2a5 ea1d0600 00bebe07 *................*
000020 b304803c 80740e80 3c00751c 83c610fe *.....t....u.....*
000030 cb75efcd 188b148b 4c028bee 83c610fe *.u......L.......*
000040 cb741a80 3c0074f4 be8b06ac 3c00740b *.t....t.......t.*
000050 56bb0700 b40ecd10 5eebf0eb febf0500 *V.......^.......*
000060 bb007cb8 010257cd 135f730c 33c0cd13 *..|...W.._s.3...*
000070 4f75edbe a306ebd3 bec206bf fe7d813d *Ou..........}.=*
000080 55aa75c7 8bf5ea00 7c000049 6e76616c *U.u.....|..Inval*
000090 69642070 61727469 74696f6e 20746162 *id partition tab*
0000a0 6c650045 72726f72 206c6f61 64696e67 *le.Error loading*
0000b0 206f7065 72617469 6e672073 79737465 * operating syste*
0000c0 6d004d69 7373696e 67206f70 65726174 *m.Missing operat*
0000d0 696e6720 73797374 656d0000 00000000 *ing system......*
0000e0 00000000 00000000 00000000 00000000 *................*
0000f0 TO 0001af SAME AS ABOVE
0001b0 00000000 00000000 00000000 00008001 *................*
0001c0 0100060d fef83e00 00000678 0d000000 *..........x....*
0001d0 00000000 00000000 00000000 00000000 *................*
0001e0 00000000 00000000 00000000 00000000 *................*
0001f0 00000000 00000000 00000000 000055aa *..............U.*
```

```
                BEGIN:                          NOW AT 0000:7C00, RELOCATE

0000:7C00 FA            CLI                       disable int's
0000:7C01 33C0          XOR     AX,AX             set stack seg to 0000
0000:7C03 8ED0          MOV     SS,AX
0000:7C05 BC007C        MOV     SP,7C00           set stack ptr to 7c00
0000:7C08 8BF4          MOV     SI,SP             SI now 7c00
0000:7C0A 50            PUSH    AX
0000:7C0B 07            POP     ES                ES now 0000:7c00
0000:7C0C 50            PUSH    AX
0000:7C0D 1F            POP     DS                DS now 0000:7c00
0000:7C0E FB            STI                       allow int's
0000:7C0F FC            CLD                       clear direction
0000:7C10 BF0006        MOV     DI,0600           DI now 0600
0000:7C13 B90001        MOV     CX,0100           move 256 words (512
bytes)
0000:7C16 F2            REPNZ                     move MBR from 0000:7c00
0000:7C17 A5            MOVSW                        to 0000:0600
0000:7C18 EA1D060000    JMP     0000:061D         jmp to NEW_LOCATION

        NEW_LOCATION:                           NOW AT 0000:0600
```

```
0000:061D BEBE07      MOV     SI,07BE           point to first table
entry
0000:0620 B304        MOV     BL,04             there are 4 table
entries

        SEARCH_LOOP1:                         SEARCH FOR AN ACTIVE ENTRY

0000:0622 803C80      CMP     BYTE PTR [SI],80  is this the active
entry?
0000:0625 740E        JZ      FOUND_ACTIVE      yes
0000:0627 803C00      CMP     BYTE PTR [SI],00  is this an inactive
entry?
0000:062A 751C        JNZ     NOT_ACTIVE        no
0000:062C 83C610      ADD     SI,+10            incr table ptr by 16
0000:062F FECB        DEC     BL                decr count
0000:0631 75EF        JNZ     SEARCH_LOOP1      jmp if not end of table
0000:0633 CD18        INT     18                GO TO ROM BASIC
```

```
        FOUND_ACTIVE:                           FOUND THE ACTIVE ENTRY

0000:0635 8B14          MOV     DX,[SI]             set DH/DL for INT 13
call
0000:0637 8B4C02        MOV     CX,[SI+02]          set CH/CL for INT 13
call
0000:063A 8BEE          MOV     BP,SI               save table ptr

        SEARCH_LOOP2:                           MAKE SURE ONLY ONE ACTIVE
ENTRY

0000:063C 83C610        ADD     SI,+10              incr table ptr by 16
0000:063F FECB          DEC     BL                  decr count
0000:0641 741A          JZ      READ_BOOT           jmp if end of table
0000:0643 803C00        CMP     BYTE PTR [SI],00    is this an inactive
entry?
0000:0646 74F4          JZ      SEARCH_LOOP2        yes
```

```
          NOT_ACTIVE:                          MORE THAN ONE ACTIVE ENTRY
FOUND

0000:0648 BE8B06      MOV     SI,068B            display "Invld prttn
tbl"

         DISPLAY_MSG:                          DISPLAY MESSAGE LOOP

0000:064B AC          LODSB                    get char of message
0000:064C 3C00        CMP     AL,00            end of message
0000:064E 740B        JZ      HANG             yes
0000:0650 56          PUSH    SI               save SI
0000:0651 BB0700      MOV     BX,0007          screen attributes
0000:0654 B40E        MOV     AH,0E            output 1 char of
message
0000:0656 CD10        INT     10                  to the display
0000:0658 5E          POP     SI               restore SI
0000:0659 EBF0        JMP     DISPLAY_MSG      do it again

                HANG:                         HANG THE SYSTEM LOOP
```

```
0000:065B EBFE          JMP     HANG                    sit and stay!

          READ_BOOT:                                  READ ACTIVE PARITION BOOT
RECORD

0000:065D BF0500        MOV     DI,0005                 INT 13 retry count

          INT13RTRY:                                  INT 13 RETRY LOOP

0000:0660 BB007C        MOV     BX,7C00
0000:0663 B80102        MOV     AX,0201                 read 1 sector
0000:0666 57            PUSH    DI                      save DI
0000:0667 CD13          INT     13                      read sector into
0000:7c00
0000:0669 5F            POP     DI                      restore DI
0000:066A 730C          JNB     INT13OK                 jmp if no INT 13
0000:066C 33C0          XOR     AX,AX                   call INT 13 and
0000:066E CD13          INT     13                         do disk reset
0000:0670 4F            DEC     DI                      decr DI
0000:0671 75ED          JNZ     INT13RTRY               if not zero, try again
0000:0673 BEA306        MOV     SI,06A3                 display "Errr ldng
systm"
0000:0676 EBD3          JMP     DISPLAY_MSG             jmp to display loop
```

```
            INT13OK:                                INT 13 ERROR

0000:0678 BEC206      MOV     SI,06C2               "missing op sys"
0000:067B BFFE7D      MOV     DI,7DFE               point to signature
0000:067E 813D55AA    CMP     WORD PTR [DI],AA55    is signature
correct?
0000:0682 75C7        JNZ     DISPLAY_MSG           no
0000:0684 8BF5        MOV     SI,BP                 set SI
0000:0686 EA007C0000  JMP     0000:7C00             JUMP TO THE BOOT
SECTOR
                                                       WITH SI POINTING
TO
                                                       PART TABLE ENTRY

Messages here.

0000:0680 ........ ........ ......49 6e76616c *           Inval*
0000:0690 69642070 61727469 74696f6e 20746162 *id partition tab*
0000:06a0 6c650045 72726f72 206c6f61 64696e67 *le.Error loading*
0000:06b0 206f7065 72617469 6e672073 79737465 * operating syste*
0000:06c0 6d004d69 7373696e 67206f70 65726174 *m.Missing operat*
0000:06d0 696e6720 73797374 656d00.. ........ *ing system.     *

Data not used.

0000:06d0 ........ ........ ......00 00000000 *            .....*
0000:06e0 00000000 00000000 00000000 00000000 *................*
0000:06f0 00000000 00000000 00000000 00000000 *................*
0000:0700 00000000 00000000 00000000 00000000 *................*
0000:0710 00000000 00000000 00000000 00000000 *................*
0000:0720 00000000 00000000 00000000 00000000 *................*
0000:0730 00000000 00000000 00000000 00000000 *................*
```

```
0000:0740 00000000 00000000 00000000 00000000 *................*
0000:0750 00000000 00000000 00000000 00000000 *................*
0000:0760 00000000 00000000 00000000 00000000 *................*
0000:0770 00000000 00000000 00000000 00000000 *................*
0000:0780 00000000 00000000 00000000 00000000 *................*
0000:0790 00000000 00000000 00000000 00000000 *................*
0000:07a0 00000000 00000000 00000000 00000000 *................*
0000:07b0 00000000 00000000 00000000 0000.... *............    *
```

The partition table starts at 0000:07be. Each partition table entry is 16 bytes. This table defines a single primary partition which is also an active (bootable) partition.

```
0000:07b0 ........ ........ ........ ....8001 *              ....*
0000:07c0 0100060d fef83e00 00000678 0d000000 *..........x....*
0000:07d0 00000000 00000000 00000000 00000000 *................*
0000:07e0 00000000 00000000 00000000 00000000 *................*
0000:07f0 00000000 00000000 00000000 0000.... *............    *
```

The last two bytes contain a 55AAH signature.

```
0000:07f0 ........ ........ ........ ....55aa *..............U.*
```