

## Re: BIOS startup ??

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Fedora/2007-08/msg03396.html>

---

- *From:* Marko Vojinovic <[yvmarko@xxxxxxxxxxx](mailto:yvmarko@xxxxxxxxxxx)>
  - *Date:* Mon, 20 Aug 2007 18:39:43 +0000
- 

On Sunday 19 August 2007 17:42, William Case wrote:

What happens in the micro-seconds between hitting the power button until BIOS is loaded from CMOS into RAM memory?

Ok, Les has given an extensive description on the hardware level. I have managed to find a similar description on the software level. It is not a reference I was talking about before (I am yet to find it), but it is quite similar, and I hope useful, though some information may be a bit outdated.

Btw, sorry for the long post...

Best, :-)  
Marko

<quote>

The boot process

Computer initialization is a process – not an event. From the moment power is applied until the system sits idle at the command-line prompt or graphical desktop, the PC boot process is a sequence of predictable steps that verify the system and prepare it for operation. By understanding each step in system initialization, you can develop a real appreciation for the way that hardware and software relate to one another – you also stand a much better chance of identifying and resolving problems when a system fails to boot properly. This part of the chapter provides a step-by-step review of a typical PC boot process.

Applying power

PC initialization starts when you turn the system on. When all output voltages from the power supply are valid, the supply generates a Power Good (PG) logic signal. It can take between 100ms and 500ms for the supply to generate a PG signal. When the motherboard timer IC receives the PG signal, the timer stops forcing a Reset signal to the CPU. At this point, the CPU starts

## Re: BIOS startup ??

processing.

### The bootstrap

The very first operation performed by a CPU is to fetch an instruction from address FFFF:0000h. Since this address is almost at the end of available ROM space, the instruction is almost always a jump command (JMP) followed by the actual BIOS ROM starting address. By making all CPUs start at the same point, the BIOS ROM can then send program control anywhere in the particular ROM (and each ROM is usually different). This initial search of address FFFF:0000h and the subsequent re-direction of the CPU is traditionally referred to as the bootstrap in which the PC "pulls itself up by its bootstraps" – or gets itself going. Today, we have shortened the term to boot, and have broadened its meaning to include the entire initialization process.

### Core tests

The core tests are part of the overall Power-On Self-Test (POST) sequence which is the most important use of a system BIOS during initialization. As you might expect, allowing the system to initialize and run with flaws in the motherboard, memory, or drive systems can have catastrophic consequences for files in memory or on disk. To insure system integrity, a set of hardware-specific self-test routines checks the major motherboard components, and identifies the presence of any other specialized BIOS ICs in the system (i.e. drive controller BIOS, video BIOS, SCSI BIOS, and so on).

BIOS starts with a test of the motherboard hardware such as the CPU, math co-processor, timer ICs, direct memory access (DMA) controllers, and interrupt (IRQ) controllers. If an error is detected in this early phase of testing, a series of beeps (or beep codes) are produced. By knowing the BIOS manufacturer and the beep code, you can determine the nature of the problem. Chapter 19 deals with beep and error codes in more detail. Beep codes are used because the video system has not been initialized.

Next, BIOS looks for the presence of a video ROM between memory locations C000:0000h through C780:000h. In just about all systems, the search will reveal a video BIOS ROM on a video adapter board plugged into an available expansion slot. If a video BIOS is found, its contents are evaluated with a checksum test. If the test is successful, control is transferred to the video BIOS which loads and initializes the video adapter. When initialization is complete, you will see a cursor on the screen, and control returns to the system BIOS. When no external video adapter BIOS is located, the system BIOS will provide an initialization routine for the motherboard's video adapter, and a cursor will also appear. Once the video system initializes, you are likely to see a

## Re: BIOS startup ??

bit of text on the display identifying the system or video BIOS ROM maker and revision level. If the checksum test fails, you will see an error message such as; C000 ROM Error or Video ROM Error. Initialization will usually halt right there.

Now that the video system is ready, system BIOS will scan memory from C800:0000h through DF80:0000h in 2KB increments to search for any other ROMs that might be on other adapter cards in the system. If other ROMs are found, their contents are tested and run. As each supplemental ROM is executed, they will show manufacturer and revision ID information. In some cases, a supplemental (or "adapter") ROM may alter an existing BIOS ROM routine. For example, an Ultra DMA/33 drive controller board with its own on-board ROM will replace the motherboards older drive routines. When a ROM fails the checksum test, you will see an error message such as; XXXX ROM Error. The XXXX indicates the segment address where the faulty ROM was detected. When a faulty ROM is detected, system initialization will usually halt.

### POST

BIOS then checks the memory location at 0000:0472h. This address contains a flag that determines whether the initialization is a cold start (power first applied) or a warm start (reset button or <Ctrl>+<Alt>+<Del> key combination). A value of 1234h at this address indicates a warm start – in which case the (POST) routine is skipped. If any other value is found at that location, a cold start is assumed, and the full POST routine will be executed.

The full POST checks many of the other higher-level functions on the motherboard, memory, keyboard, video adapter, floppy drive, math co-processor, printer port, serial port, hard drive, and other sub-systems. There are dozens of tests performed by the POST. When an error is encountered, the single-byte POST code is written to I/O port 80h where it may be read by a POST code reader. In other cases, you may see an error message on the display (and system initialization will halt). Keep in mind that POST codes and their meanings will vary slightly between BIOS manufacturers. If the POST completes successfully, the system will respond with a single beep from the speaker. Chapter 19 covers I/O port POST codes.

### Finding the OS

The system now needs to load an operating system (usually DOS or Windows 95). The first step here is to have the BIOS search for a DOS volume boot sector (VBS) on the A: drive. If there is no disk in the drive, you will see the drive light illuminate briefly, and then BIOS will search the next drive in the boot order (usually drive C:). If there is a disk in drive A:, BIOS will load sector 1 (head 0 cylinder 0) from the disk's DOS volume boot sector into

## Re: BIOS startup ??

memory starting at 0000:7C00h. There are a number of potential problems when attempting to load the VBS. Otherwise, the first program in the directory (IO.SYS) will begin to load, followed by MSDOS.SYS.

If the first byte of the DOS VBS is less than 06h (or if the first byte is greater than or equal to 06h, and next nine words of the sector contain the same data pattern), you will see an error message similar to; Diskette boot record error. If the IO.SYS and MSDOS.SYS are not the first two files in the directory (or some other problem is encountered in loading), you'll see an error such as; Non-system disk or disk error. If the boot sector on the diskette is corrupt and cannot be read (DOS 3.3 or earlier), you'll probably get a Disk boot failure message. If the OS cannot be loaded from any floppy drive, the system will search the first fixed drive (hard drive). Hard drives are a bit more involved than floppy disks. BIOS loads sector 1 (head 0 cylinder 0) from the hard drive's master partition boot sector (called the "master boot sector" or MBS) into memory starting at 0000:7C00h, and the last two bytes of the sector are checked. If the final two bytes of the master partition boot sector are NOT 55h and AAh respectively, the boot sector is invalid, and you will see an error message similar to; No boot device available and system initialization will halt. Other systems may depict the error differently, or attempt to load ROM BASIC. If the BIOS attempts to load ROM BASIC, and there is no such feature in the BIOS, you'll see a ROM BASIC error message.

Otherwise, the disk will search for and identify any extended partitions (up to 24 total partitions). Once any extended partitions have been identified, the drive's original boot sector will search for a boot indicator byte marking a partition as active and bootable. If none of the partitions are marked as bootable (or if more than one partition is marked bootable), a disk error message will be displayed such as; Invalid partition table. Some older BIOS versions may attempt to load ROM BASIC, but will generate an error message in most cases anyway.

When an active bootable partition is found in the master partition boot sector, the DOS volume boot sector (VBS) from the bootable partition is loaded into memory and tested. If the DOS volume boot sector cannot be read, you will see an error message similar to; Error loading operating system. When the DOS volume boot sector does load, the last two bytes are tested for a signature of 55h and AAh respectively. If these signature bytes are missing, you will see an error message such as; Missing operating system. Under either error condition, system initialization will halt.

After the signature bytes are identified, the DOS volume boot sector (now in memory) is executed as if it were a program. This "program" checks the root directory to ensure that IO.SYS and MSDOS.SYS (or IBMBIO.COM and IBMDOS.COM) are available. In older

## Re: BIOS startup ??

MS-DOS versions, IO.SYS and MSDOS.SYS have to be the first two directory entries. If the DOS volume boot sector was created with MS-DOS 3.3 or earlier and the two startup files are not the first two files in the directory (or there is an error in loading the files), the system will produce an error code such as; Non-System disk or disk error. If the boot sector is corrupt, you may see a message like; Disk boot failure.

### Loading the OS

If there are no problems detected in the disk's DOS volume boot sector, IO.SYS (or IBMBIO.COM) is loaded and executed. If Windows 95 is on the system, IO.SYS may be renamed WINBOOT.SYS which will be executed instead. IO.SYS contains extensions to BIOS that start low level device drivers for such things as the keyboard, printer, and block devices. Remember that IO.SYS also contains initialization code that is only needed during system startup. A copy of this initialization code is placed at the top of conventional memory which takes over initialization. The next step is to load MSDOS.SYS (or IBMDOS.COM) which is loaded such that it overlaps the part of IO.SYS containing the initialization code. MSDOS.SYS (the MS-DOS kernel) is then executed to initialize base device drivers, detect system status, reset the disk system, initialize devices such as the printer and serial port, and set up system default parameters. The MS-DOS essentials are now loaded, and control returns to the IO.SYS/WINBOOT.SYS initialization code in memory.

NOTE: For Windows 95 systems, IO.SYS (or WINBOOT.SYS) combines the functions of IO.SYS and MSDOS.SYS.

</quote>

Marko Vojinovic  
Institute of Physics  
University of Belgrade

=====  
e-mail: vmarko@xxxxxxxxxxxxx

--  
fedora-list mailing list  
fedora-list@xxxxxxxxxxx  
To unsubscribe: <https://www.redhat.com/mailman/listinfo/fedora-list>