

reading ROM BIOS

Source: <http://coding.derkeiler.com/Archive/Assembler/comp.lang.asm.x86/2004-09/0723.html>

From: Allan Adler (*spamtrap_at_crayne.org*)

Date: 09/28/04

Date: Tue, 28 Sep 2004 19:43:15 +0000 (UTC)

I have a Dell Latitude CsX laptop running RedHat 7.1 Linux and I'm trying to read the ROM BIOS. I copied the BIOS to a file and disassembled it using nasm-0.98.38 and now I'm trying to read it. My goal is to understand more about the hardware of the PC by observing the "conversation" that takes place between the CPU and the other hardware on bootup. I've already posted to comp.os.linux.misc and to a laptop group and gotten some helpful suggestions, but this newsgroup may be a more appropriate place to pursue this topic. Following one suggestion, I got van Gilluwe's book, *The Undocumented PC*, 2nd ed. I'm rather handicapped by not really knowing a lot about programming in assembler or machine language on this CPU. However, the nasm documentation has some information and I also downloaded the manuals on the Pentium III from Intel, since the laptop has a Pentium III.

I assumed that the first instruction executed on bootup is at FFFF0. That lead after a couple of jumps to a sequence of about 60 instructions, the last of which is a HLT and the one before which is a write to port 92h which sets a bit that, according to Gilluwe's book, causes the machine to reset. I don't understand why the machine doesn't therefore go into an infinite loop, going back to FFFF0, the couple of jumps and the 60 instructions again and then the write to port 92h and the reset.

Does the machine in fact start at FFFF0?

If so, does the machine return to FFFF0 when the bit is set at port 92h?

If not, where does it go?

If so, why isn't there an infinite loop?

Some respondents on other groups have used terms such as "saving state" and "interrupt vector table". These may be the right things to talk about, but I don't know enough to use the vague suggestion that these things explain everything. If someone has an elementary explanation, it will be greatly appreciated.

Alternatively, if there is a free BIOS, with fully documented source code, for a system involving a Pentium III, I'd be willing to study that instead just to get more experience with BIOSes before reading the BIOS for my laptop.

--

comp.lang.asm.x86: reading ROM BIOS

Ignorantly,

Allan Adler <ara@zurich.csail.mit.edu>

* Disclaimer: I am a guest and *not* a member of the MIT CSAIL. My actions and
* comments do not reflect in any way on MIT. Also, I am nowhere near Boston.