

## Re: How to generate a #SMI?

---

*Source:* <http://coding.derkeiler.com/Archive/Assembler/comp.lang.asm.x86/2006-09/msg00176.html>

---

- *From:* "Mark\_Larson" <[spamtrap@xxxxxxxxxxx](mailto:spamtrap@xxxxxxxxxxx)>
  - *Date:* 19 Sep 2006 10:22:14 -0700
- 

Lighter wrote:

How to generate a #SMI?

As we know, beginning with 80386, IA-32 processors provide three modes of operation: 1, Real-address mode; 2, Protected mode; 3, System management mode (SMM for short). According to the manuals of Intel, there is only one way to enter SMM, which is through #SMI. But, I wonder how the current process generates a SMI, because #SMI is an external interrupt, and it cannot be generated by the instruction `int n`.

To my thinking, the current process can use I/O instructions to do this. Say, the system provides a special I/O port; This port is an interface of an actual device. Upon the device detects signal at the given port, it generates a #SMI.

Am I right?

Many thanks in advance to those who can explain this for me.

Yes you are right, however the device does not generate the SMI. There are two different implementations of using I/O ports. One for emulating devices and one for a BIOS API. They are done slightly differently. You set up different chipset registers depending on which one you want.

For instance for USB emulation you have to trap access to port 60h/64h. This is a register in the chipset you set up to generate a SMI when a access to port 60h/64h occurs. You can even set it up so it only occurs on read or writes ( but not both) for the I/O port.

The second is a different chipset register you setup to make the system generate a SMI when a write to that port occurs. It's similar to a software interrupt. Dell BIOSes have quite a few things they can do. As an example, we have programs that run under Windows that use the SMI API to get hardware information ( fan speed, cpu temperature,

Re: How to generate a #SMI?

etc).

Mark Larson

.