

Linux-Kernel: Can we kill f inb_p, outb_p and other random I/O on port 0x80, in 2.6?

Can we kill f inb_p, outb_p and other random I/O on port 0x80, in 2.6?

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2003-09/5700.html>

From: Eric W. Biederman (*ebiederm_at_xmission.com*)

Date: 09/22/03

To: <linux-kernel@vger.kernel.org>

Date: 21 Sep 2003 18:27:44 -0600

inb_p and outb_p issue outb's to port 0x80 to achieve a short delay. In a reasonable system there is nothing listening to port 0x80 or there is a post card, but there are no other devices there.

On a modern system with no post card the outb travels it's way down to the LPC bus, and the outb is terminated by an abort because nothing is listening.

So far so good. Except for the fact that recent high volume ROM chips get confused when they see an abort on the LPC bus. Making it problematic to update the ROM from under Linux.

I don't know if there are other buggy LPC devices or not. But I do know that it is generally bad form do I/O to a random port.

So can we gradually kill inb_p, outb_p in 2.6? An the other miscellaneous users of I/O port 0x80 for I/O delays?

Or possibly rewriting outb_p to look something like:
outb(); udelay(200); or whatever the appropriate delay is?

When debugging this I modified arch/i386/io.h to read:
#define __SLOW_DOWN_IO__ ""
Which totally removed the delay and the system ran fine.

Eric

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@vger.kernel.org

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>