

Re: Serial related oops

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-02/msg06778.html>

- *From:* Russell King <rmk+lkml@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 19 Feb 2007 16:42:00 +0000
-

On Mon, Feb 19, 2007 at 04:29:39PM +0000, Jose Goncalves wrote:

Russell King wrote:

On Tue, Feb 20, 2007 at 02:48:14PM +0000, Frederik Deweerdt wrote:

(trimmed tie-fei.zang from the CC, added by mistake)

On Mon, Feb 19, 2007 at 02:35:20PM +0000, Russell King wrote:

Neither did I, but introducing printk's through the function, we narrowed the problem to this part of the code. And removing it makes the problem go away. We inserted 37 printk's in the function body, and Jose bisected those until the problem went away.

Well, there's still little clue about why this is causing a NULL pointer dereference. The only thing I can think is that somehow performing this test is causing a power glitch to your CPU, causing its registers to get corrupted, and which results in it doing a NULL pointer deref.

That may be the case, indeed.

Re: Serial related oops

But if the problem was a power glitch I should get Oops with or without `printk()` inserted, shouldn't I?

That depends if the `printk()` changes the timing such that it doesn't occur. Don't know, I'm only grasping at straws due to the lack of any concrete information.

If you see other tests to be performed...

Maybe adding some delays in that bit of code? I'm sure you've already thought of that though. Since no one has a proper understanding of the problem, the only suggestions possible are mere shots in the dark.

I'm no kernel expert, but it's not possible to trace what is the instruction that is causing the NULL pointer dereference?

The reported dump shows that the kernel tried to access virtual address 0, and the instruction pointer seems to be the cause of that – it has a value of zero in that dump.

The call trace indicates that the last function was called from around "uart_startup+0x63/0xf4" which is probably the indirect function call to `serial8250_startup()`. That's unconfirmed – the only way to get it confirmed is if you could dump the entire `uart_startup()` function.

```
$ grep uart_startup System.map
(address) T uart_startup
$ objdump -r -d vmlinux --start-addr=0x<address> --stop-addr=0x<address+256>
```

The `grep` should get you the address of `uart_startup`. Replace `<address>` with that value and `<address+256>` with the value plus 256 (0x100) and mail the result.

I have no clue on what is causing this problem but, what I know, is that I can always reproduce it, and it always happens in the same code section of `serial8250_startup()`.

We're both at the same level of clue about the problem then.

—
Russell King
Linux kernel 2.6 ARM Linux – <http://www.arm.linux.org.uk/>
maintainer of:
—

Re: Serial related oops

Re: Serial related oops

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxx
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>