

Re: Realtime Preemption, 2.6.12, Beginners Guide?

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-07/1518.html>

From: Alistair John Strachan (s0348365_at_sms.ed.ac.uk)

Date: 07/07/05

To: Ingo Molnar <mingo@elte.hu>
Date: Thu, 7 Jul 2005 12:21:47 +0100

On Thursday 07 Jul 2005 10:46, Alistair John Strachan wrote:

[snip]

- >
- > *Okay, when I brought my laptop back into work today for audio work, it*
- > *locked up again within two minutes. I realise now what the problem is, but*
- > *I don't have a serial cable here, so I'll have to rely on capturing the*
- > *oops from the console.*
- >
- > *The only difference between work and home is that I connect over an OpenVPN*
- > *connection at work, which is a userspace program that creates a "tun"*
- > *device as a virtual network adaptor.*
- >
- > *I'm convinced this is the problem, because I enabled IPMASQ on the company*
- > *server today and bypassed the VPN, and I'm happily typing this email from*
- > *the same computer on the same network, just with no VPN started.*
- >
- > *It's a bizarre problem, but my guess is that your user test bed don't end*
- > *up using tap/tun very often, which means it's poorly tested.*

Okay, I've managed to figure out a reproducible test for this problem. Bring up a normal network interface, in my case wlan (my ipw2200 wireless). Start an openvpn session to any nearby host.

Without X loaded, running in multi-user text mode, I switched VTs to the second VT and started;

```
sleep 10; wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.0.tar.bz2
```

Then I switched back to the primary VT and waited. About 15 seconds later, I get:

```
=====
=====
=====
=====
=====
=====
=====
```

Linux-Kernel: Re: Realtime Preemption, 2.6.12, Beginners Guide?

```
=====
=====
[...]
```

Filling my screen up, page after page. Then eventually the kernel freezes and I get no more output. Now, I scanned through your realtime-preempt patch and only found a single place where this occurs; a small hunk in linux/arch/i386/kernel/traps.c show_trace().

```
while (1) {
    struct thread_info *context;
    context = (struct thread_info *)
        ((unsigned long)stack & ~(THREAD_SIZE - 1));
    ebp = print_context_stack(context, stack, ebp);
    stack = (unsigned long*)context->previous_esp;
    if (!stack)
        break;
    printk("=====\n");
}
```

Something's been corrupted so badly that "stack" is always true; presumably it never reaches the bottom stack, and printk()s forever. That, or show_trace() is being called from multiple contexts.

Unfortunately, since this is called when the kernel crashes, it's impossible for me to capture any messages prior to this spam, if there even are any.

```
--
Cheers,
Alistair.
personal: alistair()devzero!co!uk
university: s0348365()sms!ed!ac!uk
student: CS/CSim Undergraduate
contact: 1F2 55 South Clerk Street,
         Edinburgh. EH8 9PP.
```

```
-
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/
```