

Re: Disobeying jet engines – why?

Re: Disobeying jet engines – why?

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2008-01/msg03933.html>

- *From:* Martin Brown <|||newspam|||@nezumi.demon.co.uk>
 - *Date:* Fri, 25 Jan 2008 15:38:15 +0000
-

In message <prtjp3tq40kvqtka0um88tmap2o6qptd0g@xxxxxxx>, John Larkin <jjlarkin@xx> writes

On Fri, 25 Jan 2008 13:53:26 GMT, Glen Walpert <gwalpert@xxxxxxxxxx> wrote:

On Thu, 24 Jan 2008 11:44:52 -0800, John Larkin <jjlarkin@xx> wrote:

On Thu, 24 Jan 2008 13:18:35 -0600, Damon Hill <damon1SIX1@xxxxxxxxxxxxxxxx> wrote:

Didi <diditgi@xxxxxxxxxx> wrote in
<news:f121757f-1671-4ebe-892d-625ea1c236b6@h11g2000prf.googlegroups.com>:

<http://news.bbc.co.uk/1/hi/england/london/7206596.stm>

Windoze is NOT qualified for critical applications like this. It's something pretty specialized and focused for the application. I suspect the investigation will either find a hardware fault or operator error (misconfigured autopilot).

I fear it will be found to be software or firmware at fault.

I know some of the guys who do the engine control computer firmware for the Pratt&Whitney engines. They use our gear to simulate engine

Re: Disobeying jet engines – why?

sensor signals to the control computer, and run weeks/months of scripts to verify the firmware in all sorts of situations.

Their ECC's use no OS at all, just basic bare-metal state machines.

You mean that they roll their own RTOS rather than buying one known to work. You can make a state machine controller with FPGAs and no OS, but a uP needs some sort of RTOS in order to function even if they don't call it that and it is integral with the rest of the code.

I don't consider what they do to be an "OS". There is no distinct separation between operating system and application code; no tasks are suspended partially-done; no task contexts are saved. I'm not absolutely sure, but I think there's no memory management and only one stack. I'll try to find out. The guys who do this have told me, emphatically, that there is no RTOS.

There may still be a kernel to arbitrate between interrupts and something like a watchdog timer to provide a means for restart and recovery in the even that the CPU itself latches up. I have know CPUs latch up (usually they were provoked to do so by application of large voltage spikes).

And you can get hit by lightning whilst trying to land. Although the airframe behaves as a pretty good Faraday cage.

I do embedded realtime apps that have no OS. I've written a few RTOS's, but haven't needed to use one in a long time.

If the code is...

```
START: do thing1
do thing2
do thing3
goto START
```

where is the OS?

The OS is the bit that should step in when one of thing1, thing2 or thing3 fails to complete. Most non-trivial embedded applications have some kind of RT kernel underpinning the allocation of resources.

I have even known one processor (specific batch) that would very very occasionally fail to obey the RTI instruction. Manufacturer denied it until faced with a logic analyser trace showing the event captured.

Re: Disobeying jet engines – why?

Some of the engines run fuel through the ecc before it's burned, to protect the ecc from temperature extremes.

Fuel can get brutally cold on these long flights. One of the met office guys was talking about -70C or even lower exterior temperatures in the final stages of a transcontinental flight from China/Japan. Hence the concern that waxing in the remaining fuel might have taken out the engines simultaneously.

Losing one engine is unusual but losing both at the same time is exceedingly rare (unless you fly into a volcanic ash cloud, or a flock of birds). We have to wait and see what the investigators find.

Regards,

--

Martin Brown

--

Posted via a free Usenet account from <http://www.teranews.com>

.