Re: Disobeying jet engines − why?

Source: http://sci.tech−archive.net/Archive/sci.electronics.design/2008−01/msg04859.html

- *From*: Didi <diditgi@xxxxxxxxx>
- *Date*: Wed, 30 Jan 2008 10:28:23 −0800 (PST)

Well you do need a debugger above a certain code complexity allright.

> I do not see this, honestly.
> I do the complicated stuff in Linux, not sure if I did hit the 1M lines
> of code, way over 100000 I am in some programs though.
> I mean gdb is just 3 characters away, but in C printf() always helps me out.

Oh in C (or Basic or whatever HLL) using the debugger is not much
use, I agree.

But I was talking programming, not rearranging someone elses
code :−).

I do not take seriously any embedded thing written in C or any HLL for
that.
If they use it (as I am sure thay do) on jets like in this thread,
we can only expect more disobeying engines and anything.

The main thing HLLs do is hide the CPU details from the programmer.
If the CPU is too ugly to program (most are...) with its details in
sight, pick another CPU, there are options. If that does not
help, well, one should consider another job...

Dimiter

−−
Dimiter Popoff Transgalactic Instruments

http://www.tgi−sci.com
−−
http://www.flickr.com/photos/didi_tgi/sets/72157600228621276/

On Jan 30, 8:05 pm, Jan Panteltje <pNaonStpealm...@xxxxxxxxx> wrote:

> On a sunny day (Wed, 30 Jan 2008 09:16:30 −0800 (PST)) it happened Didi
> <didi...@xxxxxxxxx> wrote in
> <601837e4−0e79−44ea−8066−42fa66422...@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>:

> If you know your code, then you need no debugger! HIGH
> LEVEL LANGUAGES
> LIKE HEX or ASM need no debugger (I ain't kidding).

> Well you do need a debugger above a certain code complexity allright.

I do not see this, honestly.
I do the complicated stuff in Linux, not sure if I did hit the 1M lines
of code, way over 100000 I am in some programs though.
I mean gdb is just 3 characters away, but in C printf() always helps me out.

There are some basic rules, and libc.info is my friend, gives me all the help
I need, and prevents me from doing things better not done.
And I try to avoid exotic constructs.
In C: keep your pointers in range, check for null pointers in every function.

.