

Re: incredible

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2004-09/3434.html>

From: Frithiof Andreas Jensen (*frithiof.jensen_at_die_spammer_die.ericsson.com*)

Date: 09/16/04

Date: Thu, 16 Sep 2004 10:21:12 +0200

"John Larkin" <jjlarkin@highlandSNIPtechTHISnologyPLEASE.com> wrote in message news:4fshk09nb1psnf6jf03g607g81r6dr17vb@4ax.com...

> *It wouldn't be if the operating system enforced I/D space separation,
> as was an established practice 30 years ago.*

True – the best part is that this MMU feature has been sitting idle ever since the 486!!

> *I wonder if OOP makes things worse by mixing code and data so
> intimately.*

Not at all – the "intimacy" is merely a "presentation service" to you, the developer;

It **appears like** you put data and code operating on said data into the same "sealed bag" but the linker will work behind the scenes and move everything into their designated memory segments.

The segments are registered with the Memory Management Unit – who enforces the per-segment access control – by the Board Support Package; a piece of code that configures the hardware and boot the OS. (On the PC, probably the OS would do it – since the OS is the "main show", replacing the BIOS at the first opportunity).

The traps are that you can configure the build system so that code and data is really intermingled (maybe it's like that out of the box), the target architecture may not enforce the separation even if you **do** specify it properly, the target architecture may not have the separation features in the MMU **switched on** and of course there is a **performance penalty** in calling across memory boundaries because pointers cannot be passed directly, instead parameters/returns must be copied and functions indexed from tables etc.

The last part is probably **why** the memory separation features are not commonly used – the performance cost of memory protection will make Games suck!

sci.electronics.design: Re: incredible

Embedded systems are big on memory protection and one observes that embedded's do not crash much ;-)