Re: Kernel Calling Conventions

Source: http://coding.derkeiler.com/Archive/Assembler/comp.lang.asm.x86/2006−03/msg00204.html

- *From*: Robert Redelmeier <redelm@xxxxxxxxxxxxxxx>
- *Date*: Mon, 20 Mar 2006 06:00:39 GMT

Kroll <spamtrap@xxxxxxxxxx> wrote in part:

> Greetings all, I was reading the "FreeBSD Assembly Language
> Programming" tutorial (http://www.int80h.org/bsdasm/),
> when I came across something that piqued my curiosity.
>
> The C calling convention is touted as being more convenient, and
> superior to the calling convention used by linux and microsoft
> of passing arguments within registers. I was wondering if this
> is just a bias on the part of the writer or what.

IMHO, "bias" is too mild a word :)

Regs vs stack is a much closer argument for userspace
lib calls. For kernel transitions, registers are _much_
easier: when transitioning from ring3 (userland) to ring0
(kernelspace), a stack switch is done. At the very least,
the old pointer would have to be recalled to fetch the params.
And some work with the LDT may be necessary to recover the
process' memory mapping. I don't recall whether these are
reset on transition.

The main advantage of stack params is that many more
can be passed.

−− Robert

.