

Re: Purpose of defmethod

Source: <http://coding.derkeiler.com/Archive/Lisp/comp.lang.lisp/2004-07/0527.html>

From: Pascal Costanza (costanza_at_web.de)

Date: 07/13/04

Date: Tue, 13 Jul 2004 15:12:41 +0200

Frode Vatvedt Fjeld wrote:

> Gareth McCaughan <gareth.mccaughan@pobox.com> writes:

>

>>No. The implementation could remember the locations of all calls to
>>each function, and patch them if the function gets redefined.

>

> Is there any implementation that actually does this? If not, what are
> the decisive reasons for this? (Not a rhetorical question.)

I am no expert on this topic, but from what I have heard the devil lies in the details. For example, if you re-inline a redefined function, the code size of the call site changes, jump targets need to be recalculated, and so on. Effectively, you need to recompile the call site. Now imagine that that call site is currently executed, so you also have to recalculate the program counter. This can happen for both other threads and the current thread in the general case. Furthermore, if you recompile the call site, and that function was again inlined somewhere else, you have to do all of these things in turn for its call sites, and so on.

Since this can become very costly, you have to measure at runtime whether the cost/benefit ratio is good enough for inlining a function, so you need to count function calls and apply appropriate heuristics. These things depend heavily on the actual CPU, the operating system used, etc. I'll stop here. ;)

Pascal

--

Pascal Costanza

<mailto:costanza@web.de>

<http://www.pascalcostanza.de>

University of Bonn

Institute of Computer Science III

Römerstr. 164, D-53117 Bonn (Germany)