

Re: Book on Assembly

Source: <http://coding.derkeiler.com/Archive/Assembler/alt.lang.asm/2006-06/msg00630.html>

- *From:* "santosh" <santosh.k83@xxxxxxxxxx>
 - *Date:* 23 Jun 2006 10:39:57 -0700
-

randyhyde@xxxxxxxxxxxxxx wrote:

santosh wrote:

randyhyde@xxxxxxxxxxxxxx wrote:

Jim Carlock wrote:

<randyhyde@xxxxxxxxxxxxxx> wrote:

[BIG snip]

[small snip]

Would it not be more time efficient to teach them the native conditional branching instructions and tell them to implement the HLL control statements with them as intermediate level assignments?

???

This is exactly what is done (or, at least, what I do). I'm not sure I follow what you're trying to say. Obviously, in order to implement HLL-like control structures with CMP and Jcc, you must first teach them the semantics of the CMP and Jcc instructions. The next step, before you let them write a whole lot of code, is to tell them how to use those CMP and Jcc instructions to simulate an IF statement, or a WHILE loop, or...

This is exactly what I have in mind, but, surprisingly, you do the exact opposite in AoA 32 bits. You start with the traditional HLL control structures, built opaquely into HLA and explanations on using CMP+Jcc start quite late into the book.

Blind use of IF, WHILE, FOR etc., is exactly what the students would have already learnt, (assuming they took a HLL course). Why repeat the same with HLA? Why not *start* off with CMP and Jcc and encourage the students to build semantic equivalents of IF, WHILE, FOR, SWITCH etc.,

Re: Book on Assembly

using these? This could start towards the middle of the course, after the basics have been dealt with.

If you've already answered to this question just point me to the post. I haven't been keeping a close track of this group lately.

Probably what you missed was the original point, which is that students should **not** be taught about simulating IF/WHILE/etc, that they should just use the CMP/Jcc paradigm in whatever way comes natural to them (i.e., arbitrary control structures). I've seen this in action, it is a disaster.

I would partly agree with the above. Assuming the students are already proficient with a HLL, I would think that starting off directly with CMP/Jcc would be better. However, they should **not** be merely left to implement their own versions of HLL control constructs at their own leisure. Instead the instructor can **show** them on how to do it in standard ways, and then let them replicate it themselves. I should think most students would do reasonably well, and any errant use can be caught and corrected by the instructor.

Now I'm not saying that teaching them to use pre-built versions of IF, WHILE etc., is bad; repetitive practise is always good in programming. I just fell that it may be quicker and more efficient to start of with assembly primitives and build up the HLL controls rather than vice versa, as you do in AoA (32 bits).