

Re: My view on this "Is blah an assembler"

Source: <http://coding.derkeiler.com/Archive/Assembler/alt.lang.asm/2004-08/0860.html>

From: C (blackmarlin_at_asean-mail.com)

Date: 08/16/04

Date: 16 Aug 2004 11:34:52 -0700

"Beth" <BethStone21@hotmail.NOSPICEDHAM.com> wrote in message news:<WOQTc.1198\$Xu.828@newsfe4-gui.ntli.net>...

[snip]

> *You're actually criticising "type casting"...legitimately because it is
> an often horrid thing to have to include in a syntax...HLA merely, like
> every other assembly language, picks up the nastiness of this...it's not at
> all unique to HLA...other assemblers start to look horrid when you start to
> get into "mov [stringstr ptr ebx].memberA, byte ptr 34" style of syntax
> too...*

Hmm, in Luxasm that would be (assuming ebx is typed as 'stringstr') ...

```
mov ebx->memberA:1, 34
```

Shorter, tidier, but still ugly.

[snip]

> *I'm not sure that C (the person, not the language) picked up my suggestion
> of this for LuxAsm syntax or not...but I was thinking that we see lots of
> assemblers use things like the following: "stos operand" versus "stosb",
> "stosw", "stosd"...or "push dword" versus "pushd"...or, of course, somewhat
> univeral is "jmp" (tool to decide size with "jump optimisation"), "jmp
> near", "jmp far" (explicit sizes, used regardless of "jump optimisation"),
> etc...*

Well, I did not pick up on your suggestion, but I have had similar ideas myself — have a look in the /document/ directory on the Luxasm CVS — there are some files noting ideas for a logically consistant syntax, which could be used as an alternate to the default Intel style syntax, possibly by changing namespaces. So you would have something like...

```
add ax, bx -> add.w a, b  
mov ax, bx -> mov.w a, b
```

alt.lang.asm: Re: My view on this "Is blah an assembler"

```
mov cs, dx -> mov.sw c, d
movzx eax, bl -> mov.db a, b
movsb -> mov.ab
paddb mm0, mm1 -> add.xb r0, r1
paddb xmm0, xmm1 -> add.vb r0, r1
```

[snip]

> *I think C worked out a ":2" / ":4" convention or something...nice – perfect*
> *for the "constant"*