Re: Assembler in Lex/Yacc

Source: http://coding.derkeiler.com/Archive/Assembler/comp.lang.asm.x86/2005−12/msg00028.html

- *From*: "toby" <spamtrap@xxxxxxxxxx>
- *Date*: 3 Dec 2005 14:01:28 −0800

Sanky wrote:
> Hi there,
>
> I was working on an assembler for X64 architecture. I was wondering
> what are the tradeoffs in designing an assembler using lex or re2c and
> yacc? Why is that handwritten assemblers are more popular than those
> developed using lex/yacc? Developing a grammar is the toughest part,

Not particularly. As others point out, assembler grammars are not too complex (much less so than an HLL, for instance).

> but once you have a grammar ready, I think the rest of the routines are
> pretty strightforward? Is it that code generated by Lex/Yacc is not as
> efficient?

In terms of lexing/parsing speed, it screams.

> YASM does get close, but again it resorts to hand written code at some
> places.
>
> Is it that Lex/Yacc is an overkill or is that Lex/Yacc are not the
> right tools? I'm new to compiler and assembler design, and would be
> helpful if anyone can guide.

Personally I think they are excellent tools − for systems of low to moderate complexity. Randy's HLA probably lies well outside the sweet spot, for reasons he goes into.

Here are two complete assemblers I implemented with lex/yacc (actually flex and bison):
http://www.telegraphics.com.au/svn/dpa/trunk
Some rationale for choosing lex/yacc:
http://www.telegraphics.com.au/sw/info/dpa.html
One is simple, one is more complex. Error handling was not particularly difficult (contrary to Randy's remark − but this may be because the project is much simpler than his).

−−T

>
> Thanks!

.

- ***References***:
  - ***Assembler in Lex/Yacc***
    - *From:* Sanky

- Prev by Date: ***Re: sin,cos math functions in assembly***
- Next by Date: ***Re: sin,cos math functions in assembly***
- Previous by thread: ***Re: Assembler in Lex/Yacc***
- Next by thread: ***Re: Assembler in Lex/Yacc***
- Index(es):
  - ***Date***
  - ***Thread***