Message-ID:
Date: Fri, 9-Nov-84 14:20:17 EST
Article-I.D.: ima.437
Posted: Fri Nov 9 14:20:17 1984
Date-Received: Sat, 10-Nov-84 06:55:56 EST
Lines: 37
Nf-ID: #N:ima:12700006:000:1655
Nf-From: ima!johnl   Nov 9 14:19:00 1984

Of late I've been trying to write a little 8088 assembler code with
the Microsoft assembler v. 1.25. I gather that it's less buggy than
the IBM version, but it's still hard to see how they could have
tested it for more than about 5 minutes before shipping it. Here are
some bugs that I have tripped over recently.

This program generates bad object code:

```
_code segment
 extrn func:far
 call func
_code ends
 end
```

The relocatable fixup generated for the call says to try to produce
the pointer to "func" relative to the current code segment rather
than relative to whatever segment the symbol is in, as it should.
This means that, generally speaking, far calls don't work at all.

There also seems to be a problem with relocatable values in generated
STRUC structures. For some reason, the assembler generates "logical
iterated data" object types (data with repeat counts, though the
repeat counts are almost always 1) when assembling struc things, and
the interaction between LIDATA chunks and relocation FIXUPP chunks in
the object file is so obscure that I can't figure out whether the
assembler is generating bad code or the linker we use, PSA's Plink,
is interpreting it wrong. We can't use Microsoft's linker to check,
because the program I'm trying to make work has too many symbols for
the small fixed-size table in MS-link.

Does anybody know if the IBM 2.00 assembler is any better? It's
enough to make a fellow yearn for the good old days toggling octal
numbers into the front panel of a PDP-8.

John Levine, Javelin Software, Cambridge MA 617-494-1400
{ decvax!cca | yale | bbncca | allegra | cbosgd | ihnp4 }!ima!johnl
Levine@YALE.ARPA

Subject: Re: MS Assembler woes
Posted by johnl on Tue, 04 Jun 2013 04:01:45 GMT
View Forum Message <> Reply to Message

Well, I did some more experimentation on the MS assembler 1.25, and found
that all of the following generate bad object code:

```
 call far [si]
 call far ptr [si]
 call [si].dword  ; where dword is a doubleword in a struc
 call far [si].dword

 ; try to do it myself
 db 9ah
 dw offset subr, seg subr
```

I finally had to put in the bytes for the "call far [si]" in hex. Ptui.

John Levine, ima!johnl