
Subject: PDFORTH INFO

Posted by [Anonymous](#) on Tue, 18 Mar 1986 00:31:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Originally posted by: franco

<pre>

Article-I.D.: iuvax.54600039

Posted: Mon Mar 17 19:31:00 1986

Date-Received: Wed, 19-Mar-86 05:17:06 EST

Lines: 61

Nf-ID: #N:iuvax:54600039:000:2916

Nf-From: iuvax!franco Mar 17 19:31:00 1986

I have recently been in touch with the author of the "PDFORTH" that I posted to the net. I hope that at least most of you got that OK (I am frustrated by the fact that so much of what gets sent on the net winds up as garbage on the other end - at least I know the "PDFORTH" left here ok since I pulled it off the net and got it to work). Anyway, I thought you might be interested to know that what I sent out was a preliminary version of a FORTH that will eventually be marketed by a (one-man?) operation called Bradley Forthware. I have given the address below for anyone interested.

Also, I thought I would post part of our conversation which presents some of Mr. Bradley's thoughts on his FORTH. Please note that I am posting the information below only because some of you are porbably interested in the info provided. I am in no way endorsing this product. For all I know the info could be totally wrong. I have never met Mr. Bradley nor do I know anything about him except that he claims to have authored the "PDFORTH" (which, I suppose, should be relabeled BFORTH). Mr. Bradley does not know that I am posting the text of part of our conversation. The sentences preceeded by > are mine, the rest are his.

franco@indiana.CSNET

The address is:

Bradley Forthware
295 Hans Ave.
Mountain View CA 94040

> By the way - it works with TOS in ROM (at least on my machine).

I got around to trying out the version you posted after I sent you the message. I couldn't get it to work at all, but I found out enough to figure out why it works with the ROMs. The version you have copies

itself to a fixed location in memory, which is about 0x50000 as I recall. This works okay but it fragments memory and doesn't give you the full benefit of the memory that is available.

(note: The above applies only to the version I posted)

> By the way, from some crude benchmarks I
> ran it seems that your FORTH is at least as fast as DRAGON FORTH.

My analysis shows that my Forth is faster than Dragon Forth without the accelerators, and slower than Dragon Forth with the accelerators. The difference is due to my use of direct threaded code versus the the indirect threaded code used by Dragon Forth. Direct threaded code eliminates a memory fetch from the NEXT routine, so that NEXT becomes 2 instructions assembled in-line -- specifically

```
MOVE.L (A5)+,A0
JMP (A0)
```

I believe that this is the best that can be done with the 68000, short of compiling directly to machine code. On the 68000, Direct Threaded Code is even faster then subroutine threaded code, which many people erroneously believe is the fastest.

The Dragon Forth accelerators work by compiling sections of machine code in-line, eliminating NEXT overhead entirely. I think that Dragon Forth is a very good system. The multitasking especially is very complete.

</pre>
