

Programming Environments

Chairman: Rieks Joosten

Secretary: Paul Snow

July 9, 1986

We, the Forth community, have neglected research into improving the Forth programming environment. Lisp machines, SmallTalk, and UNIX owe a great deal of their popularity and productivity to their attention to the programming environment and the development of libraries of routines and tools. These are issues that Forth can no longer afford to ignore.

Our goal for this meeting was to decide upon a list of environmental topics which need attention. The "standardization" issue was ignored, with the belief that any standardization of the language would give us portability of code between different environments. (Portability of programmers is a different issue all together, but was also ignored.)

We came up with a general list of issues we would like to see addressed by Forth vendors and researchers:

- Support for different languages and operating systems. Forth should have some way of linking in routines written in FORTRAN, C, Pascal, etc. Furthermore, operating systems like UNIX have many tools one should have easy access to from Forth.
- Windows, tools, and integrated environments. Lisp machines have shown how powerful an integrated environment can be. These environments are possible using Forth, and would require much less hardware.
- Program development by many programmers, and other software engineering issues. The issue of large programs is not a simple issue. These projects do exist, and are very demanding of the language and programming environment. Forth will continue to be left out of major government and business contracts until this issue is addressed.
- Efforts to minimize Forth shock. Forth is a major computing language that does not compute with most programmers. How can we make Forth more intuitive? Can Forth be more easily learned and used than C? We believe that the programming environment is the key to these questions.

We spent the largest amount of our time voicing our opinions and asking questions. The following is a summary of what was said, asked, and some possible answers to questions raised.

The issue of editors was raised, of course. Some of us defended line editors, and some demanded "regular" full screen general text editors. And there were many shades of gray in between, and a fair number of bored looks exchanged. What we finally agreed upon was simple: the programming environment should allow alternatives. Quick and dirty fixes to code can sometimes be done more easily without the overhead of a full screen editor, and more demanding alterations might require the full screen, general purpose editor. Or maybe we just don't want to learn another editor. Why can't we have several choices?

Total integration with existing operating systems and languages was a major point with number of the participants. If such capability was common, Forth would often be used by non-Forth programmers for programming jobs requiring an interactive environment, meta compiling capabilities, or fast compilation. Often Forth is not used because it means giving up or redoing large amounts of past work. Total integration would solve this problem.

Tight coupling between code and text has been demonstrated by the Fifth and by RTL. This simple concept can lead to a very productive Forth programming environments that have the look and feel of the lisp environments. More research is needed in this area.

Forth has great capabilities in areas of testing code, debugging code, accessing hardware, and generating new programming systems (through meta compilation). And the Forth engines will make Forth fast. But Forth has ignored software engineering issues such as the control and manipulation of large programs, monitoring changes to code, and measuring programmer productivity. Being fast won't be enough; the success of ideas like the new Forth engines may be tied to what kind of programming environment we put on them. The general feeling of this meeting was pretty optimistic that the needed research will be done.

We seemed to agree that the Forth programming environment must be addressed by venders and researchers if we are going to bring Forth into the next century.