Abstract for 1986 Rochester Forth Applications Conference

The LMI Forth-83 Metacompiler

submitted by Ray Duncan
Laboratory Microsystems Inc.
P. O. Box 10430
Marina del Rey, CA 90295
(213) 306-7412

Forth Metacompilers, cross-compilers, and target compilers have been used since the invention of Forth to transport the language to new processors, modify the underlying implementations, and compile programs into ROM. They have also found use in commercial applications where there is a need for code security because Forth programs of significant size that are compiled without dictionary headers are nearly impossible to decompile.

Without exception, previous Forth metacompilers have been single-pass, inflexible, extremely dependent on the structure and interpretation mechanisms of the host system, and have incorporated little or no error-handling.

The LMI Forth-83 Metacompiler contains a number of major departures from traditional Forth metacompilers.

* The LMI Metacompiler is written in Forth-83 and has minimal dependence on the word size or structure of the host or target systems.

* The input to the Metacompiler may be either traditional Forth screens or ordinary text files. Source files may "include" other files, up to 4 levels of nesting.

* The LMI Metacompiler makes at least two passes over the source code. This allows straightforward handling of forward references. It also allows compiler words (CREATE...DOES> or IMMEDIATE words) defined in the target's source code to be invoked during metacompilation; the daughter words of CREATE...DOES> defining words may also be meta-interpreted.

* Multiple vocabularies may be declared and used within a target application.

* Symbols declared in the target are stored in a hashed table rather than the traditional linked list. This permits fast lookups coupled with efficient use of memory.

* Local labels are supported within the body of CODE definitions, reducing the load on the symbol table.

* When errors are detected during metacompilation, the LMI Metacompiler attempts a best-case fix-up and proceeds through the source code. Thus multiple errors may be detected and corrected on each run of the Metacompiler. Only a very few errors are considered severe enough to abort compilation altogether.

* The LMI Metacompiler prints a detailed error audit, a sorted symbol table, and a list of unresolved symbols.

* Metacompilation may be continued from intermediate states. Fixed address overlays may also be compiled.

The LMI Forth-83 Metacompiler was first hosted on MS-DOS but has since been ported to CP/M, 68000 UNIX, and 80286 XENIX. Forth-83 Standard, ROMable targets have been developed for the 8080, Z-80, 8086/88/286, 68000, 6502, 8096/97, 8051/31, 1802, and 6303 microprocessors thus far.

The presentation will include discussion of the general problems of Forth metacompilation, target-compilation, and cross-compilation and some details of implementation of the LMI Forth-83 Metacompiler.