The colorForth Magenta Variable

2003 Sep 05
Howerd Oakford www.inventio.co.uk

Abstract

The colorForth magenta variable returns the address of a cell in pre-parsed source. Changing the value of a magenta variable at run time automatically changes the value assigned to that variable in the source code, opening the way for closer integration of program and data.

Simplicity

Chuck Moore's latest Forth development, colorForth, is simpler than other Forths because it uses a design which eliminates many features that were previously thought to be essential :

- The Text Input Buffer, ACCEPT (or EXPECT) and QUIT were used to accumulate and interpret characters typed in at a terminal.
- The linked-list dictionary allowed variable length name, code and parameter fields to be compiled into a single array – the dictionary.
- Words such as .S and DUMP allowed the stack and memory to be displayed.

I find it surprising that Chuck has been able to simplify or remove these standard Forth structures, and yet still provide an environment that can be used to develop complex applications :

- The contents of the stack are displayed at the bottom of the editor screen. The top stack item is also displayed as a Huffman decoded string. "Typing in" the name of a Forth word accumulates the Huffman encoded bit pattern into the top stack item. As each key is pressed, the top of stack changes and its string is displayed.
- Because both the name and the "tokenised code field" (colour) of the word is packed into its token the dictionary can be reduced to two linear arrays, the tokens and the addresses of their parameter fields. Dictionary searches, and hence compilation, are extremely fast.
- The pre-parsed source is held in 1K octet blocks. These are kept in volatile memory, and are copied from disk at power up, and to disk by "save".
- .S is not required, as the stack is displayed in the editor.
- The Text Input Buffer (say one line of 80 characters) has been replaced by a two-dimensional edit screen. This may look like a conventional block-based Forth editor, but is in fact very different.

Old and New

It is interesting to note that there is no simple mapping of conventional Forth constructs to colorForth .
For example, the functionality of QUIT has been replaced by a combination of Huffman encoding into the top stack item, continuous background task updating of the display and a more complex editor.

Similarly the colorForth magenta variable cannot be described in terms of just one conventional Forth construct. The closest analogy is VARIABLE, and indeed executing either type of variable leaves the address of its data on the stack. The difference lies in where the data is stored. A conventional variable returns an address that points to the parameter field in the compiled code. A magenta variable returns the address of the next 32 bit cell in the pre-parsed source.

A Neat Trick

Consider the initialisation of the two types of variable :

```
VARIABLE x 123 x !              \ conventional
```

The value specified in the source code is stored in the variable's parameter field.

```
[magenta] x 0                   \ colorForth
```

When the magenta variable is defined it is given a default value of 0. When a new value is stored into it the source changes.

```
[yellow] 123 x !                \ yellow "interpreted" commands
[magenta] x 123                 \ colorForth
```

Because the value is stored in the source code itself, ***storing a new value changes the source.***

Transient Compilation and Source Sharing

Compilation converts human-readable source code into machine-executable instructions. Since humans and computers are very different, this is a non-trivial operation. Because the magenta variable's data is stored in the source, ***the compiled code can be discarded without losing information***.

This has three effects :

- The possibility of "just in time" or transient compilation. For example, "north" is defined as "46 load", which loads code to display the "North Bridge" PCI registers – debug state is retained, even after an "empty".
- Source code can be distributed with the current values of its variables. For example, the current distribution of cfdos.blk contains a program to display the Mandelbrot set in blocks 64 and 66, together with the x, y, scale and iterations required to produce a particular pattern.
- Programs can be executed remotely by transferring only source code. The current state information is contained in the source.

Summary

The colorForth environment allows a new construct, the magenta variable, whose data is stored in the source code. This makes it possible to compile and discard applications without losing their state information, which in turn allows a different, simpler, style of programming.