The Embedded Muse 13

Editor: Jack Ganssle (jack@ganssle.com) January 6, 1997

EC++

Long time readers of The Muse know that I'm a great fan of small systems - those 8 and 16 bit embedded systems that surround us in life.

Yet if you regularly read the trade press you'll be hard pressed to find much written about these small systems. Believe what you read and you'll soon come to the conclusion that even the smallest application requires a 32 bit CPU with 2 Mb of ROM and RAM.

One theme we've been inundated with is C++. You can hardly open a magazine without getting the feeling that C++ is the most common language around. Few articles indicate, though, that C++ is best used only on the largest systems. Forget about it for 8 bits; be very wary of using it on 16 bitters.

Though C++ does indeed bring great benefits to the art of programming, its huge footprint and performance problems limit its usefulness in a lot of embedded systems. Happily, some of the original C++ proponents are working on a subset targeted specifically at the embedded market. EC++ is a reduced version of C++ that brings many of the OOP benefits while eliminating those features that create excessive burdens for real time, embedded work.

If you're contemplating the use of C++ for your next system, check out http://www.caravan.net/ec2plus/, which is the home page of the group developing the new standard.

The goal of developing EC++ is to create a subset of C++ that is easy to use, and that is targeted to the unique needs of embedded systems. The group feels EC++ will avoid excessive memory consumption, support the creation of ROMable code, and be entirely predictable in its response.

Noble ideals indeed.

Several vendors either support the draft EC++ spec already, or plan to soon. These include Greenhills, Metrowerks, and Hiware.

Before redesigning your system for EC++, though, be aware that the committee designing the EC++ spec has little to no interest in 8/16 bit support. EC++, like it's bigger brother, will be targeted primarily at 32 bit applications.

For more information on this new language, check out P. J. Plauger's article on the subject in the December issue of Embedded Systems Programming. He gives a great overview of the differences between EC++ and C++, as well as a history and rationale behind its development.

Thought for the Week

Steve Litt, Webmaster of Troubleshooters.Com (http://www.troubleshooters.com), sent along this gem in response to the Valgol language description in the last issue of The Embedded Muse:

It was gratifying to see your thorough and accurate coverage of the VALGOL language in the Embedded Muse 12. Did you know that there's now an OOP version? In much the same way that C evolved into C++, VALGOL has evolved into VALGOL TO THE MAX. It's been endorsed by the Association of Ventura Blvd Clothing Merchants, so I must use it on a daily basis (I live in Reseda). I'll show you a snippet from one of my VALGOL TO THE MAX programs (a POS system). Although this language is readable to the point of self-documentation, I'm including comments for those not familiar with VALGOL.

VALGOL TO THE MAX'S power comes from its terse readability, its powerful collection of error handling, and its self referential application pointer, called deeeude (similar to "this" in C++, but for the application only).

```
//*****************************************************
//This partial file copy program copies certain records
//between files
//*****************************************************

like y*know (i mean) start.
FILE Tiffany = make a file.
If Tiffany isn't totally stoked
    bag your face, deeeude.      //abend
FILE Sean = make a file.
If Sean isn't totally stoked
    bag your face, deeeude.
Sean sells at the mall          //open file Sean for input
If Sean isn't just totally awesome,
    later days deeeude.
Tiffany shops at the mall.   //open file Tiffany for output
If Tiffany can't hang,
   like, barf me out, deeeude.

//Transfer only records whose key field equals "Nirvana".
Tiffany buys all Sean's Nirvana records - oops, sorry
deeeude, I mean CD's.
If any CD is bogus,
   deeeude -- gag me with a spoon.  //extra-fatal abend
Sean and Tiffany go home.           //close the files
Reseda rules, deeeude.              //exit the subroutine
```

About The Embedded Muse

The Embedded Muse is an occasional newsletter sent via email by Jack Ganssle. Send complaints, comments, and contributions to him at jack@ganssle.com.

To subscribe, send a message to majordomo@ganssle.com, with the words “subscribe embedded *your-email-address*” in the body. To unsubscribe, change the message to “unsubscribe embedded *your-email-address*”.

The Embedded Muse is supported by The Ganssle Group, whose mission is to help embedded folks get better products to market faster. We offer seminars at your site offering hard-hitting ideas - and action - you can take now to ***improve firmware quality and decrease development time***. Contact us at info@ganssle.com for more information.