The Embedded Muse 81

Editor: Jack Ganssle (jack@ganssle.com) February 14, 2003

EDITOR: Jack Ganssle, jack@ganssle.com

CONTENTS:

Editor's Notes

As many Muse readers know, I've had a column in Embedded Systems Programming for a dozen years, and write a weekly on-line piece at the magazine's companion website (embedded.com). These are good folks who provide a lot of tremendously useful info to the embedded community. They're now doing an email newsletter. The Embedded.com newsletter points you toward practical solutions for your design challenges and provides top industry news, insights from embedded experts, and product information. The newsletter is aimed at developers at every skill level. If you are interested in subscribing, go to http://www.eetimesnetwork.com/register/. Select "New Users click Here," fill out and submit the form, and check the "Embedded.com" box at the bottom of the list.

I've been getting quite a bit of email from folks wondering when the next Better Firmware Faster seminars will happen. Current plans are for Chicago on April 15 and Baltimore/Washington April 17 – see http://www.ganssle.com/classes.htm for more information.

I often do this seminar on-site, for companies with a dozen or more embedded folks who'd like to learn more efficient ways to build firmware. See http://www.ganssle.com/onsite.htm.

Reader Feedback

In the last Muse I discussed a study that shows how redundancies in code leads to errors. Even totally benign extra code that has no ill effects is a problem, as hard errors tend to cluster around the redundant software.

Bill Gatliff wrote in response:
I always notice that after code reviews, code gets smaller. So the old adage is true: it isn't done when there's nothing more to add, it's done when there's nothing more to take away. Eliminate one line of code, eliminate 0.10 bugs! That's my motto.

How about code that's just plain verbose? Linux is loaded with it! I bet I could reduce LOC by 30% by rewriting Linux. But if I were to do that, I'd go even farther, and redo it in C++ to eliminate entire classes of bugs, like failing to use the right "kind" of pointer in kernel space: physical, virtual, PCI, DMA, etc... C++ would be excellent for handling those conversions automagically, you have to do it explicitly in C.

And oh, I've also noticed that optimizers really like clear and concise code. It's as if they get confused by crud, too.

Jim Powell commented on my comments about the Natural Keyboard:
I read your comment on the Microsoft Natural Keyboard (sorry about your wrist - hope you are healing well...), and wanted to pass along why I switched to one. Several years ago I developed tendonitis in my right wrist, mostly because I sat all day typing on a keyboard and using a mouse. Switching to the split keyboard and a trackball made a tremendous difference for my wrist. I never thought when I was in college that I would one day develop tendonitis working as an EE, but a lot has changed. Anyway, thought I'd pass that along to you to pass along to your readers - if someone is getting pains in their wrists and/or upper arms, try switching to a split keyboard and a different type of pointing device and see if it makes a difference. Tendonitis can be quite nasty.

A GREAT New Book

In Maryland we're required to have our cars tested for emissions every two years. I dutifully took mine in last week. Another fellow in the waiting room went on a loud rant about the experience. His complaint (minus the four letter words) was that, since practically every car passes, it's all a waste of time.

Perhaps. But there are routines we employ to keep us honest. The threat of sanctions means we do the right thing, before we're caught. Most of us don't cheat on our taxes, yet IRS audits are relatively rare. The fear of them (and, I hope, a high ethical standard) keeps us in line. We know, and the car manufacturers know, that our autos will get pollution tested, so we don't rip out the catalytic converter, and the vendors build cars that will be clean for years.

In software, code inspections keep us honest. When companies adopt a new process of inspections some are disappointed; they find fewer problems than anticipated. That's simply because the process of having reviews means people craft better software. No one wants to be caught out in an inspection meeting. And that is a very good thing – people being more careful and checking their work leads to great code.

Karl Wiegers new book "Peer Reviews in Software" (ISBN 0-201-73485-0) is a highly readable, not boring, introduction to the practice of conducting code inspections.

Unlike other books (like "Software Inspection" by Gilb and Graham – a great but less readable tome) Wiegers looks at the entire spectrum of inspections from informal pass-around reviews to the most formal of processes.

The book gives a complete description of the entire review process, from start to finish, in about 80 pages.

Later he writes clearly about how to install a review process in your organization. He left out my favorite trick: in the early stages of an inspections program, when people are afraid of being attacked for their coding mistakes, have the author come to the inspections meeting with a pizza. Then he's the good guy.

Wiegers looks into broken reviews; when things go wrong, why? What can one do? How can we fix problems?

There's about 200 pages of text in the book, in the modern largish-type format. You can breeze through it in a couple of hours and come away with a really good map about how to do inspections.

Highly recommended.

I also have a freebie inspections paper on-line at www.ganssle.com.

Jobs

Phil Hebner wrote that Ball Aerospace in Boulder, Colorado is always looking for software engineers to develop flight software (embedded) and ground support (not embedded) software for various government projects. Most is in C, with some migration to C++. They are especially interested in persons that are of a more junior standing with an active security clearance, especially EBI. Current job openings can be viewed at ball.com/aerospace and resumes can be submitted on-line. SEE NEXT

Lonnie VanZandt says he is not currently in need of staff, but does like to maintain a portfolio of associates that he can partner with or refer work to when he gets overbooked. See http://www.predictableresponse.com/jobs.html.

Let me know if you're looking for embedded people and I'll mention it in the Muse. No headhunters, please.

Thought for the Week

Progress (n.):

The process through which the Internet has evolved from smart people in front of dumb terminals to dumb people in front of smart terminals.

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.