

The Embedded Muse 138

Editor: Jack Ganssle (jack@ganssle.com)

December 11, 2006

You may redistribute this newsletter for noncommercial purposes. For commercial use contact info@ganssle.com.

EDITOR: Jack Ganssle, jack@ganssle.com

CONTENTS:

- Editor's Notes
- Salary Survey
- Letters
- Tool Guardians
- Stack Overflows
- Jobs!
- Joke for the Week
- About The Embedded Muse

Editor's Notes

Did you know it IS possible to create accurate schedules? Or that most projects consume 50% of the development time in debug and test, and that it's not hard to slash that number drastically? Or that we know how to manage the quantitative relationship between complexity and bugs? I can present my Better Firmware Faster class at your facility. See <http://www.ganssle.com/classes.htm>.

Salary Survey

In December of 2004 I conducted a salary survey, whose results are here:
<http://www.ganssle.com/salsurv2004.pdf>.

How much has changed in the last two years? Please take the short, 12 question, 2006 survey. The minute it takes will help all of us in this field. I'll publish the results in the next few weeks.

The survey is here: <http://www.surveymonkey.com/s.asp?u=932393017377>

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

Letters

In the last issue I reviewed the book “Agile Estimating and Planning.” Mason Deaver pointed out that Steve McConnell’s work on the subject, “Software Estimation: Demystifying the Black Art” is an excellent reference. He – and others – reviewed it here: http://www.amazon.com/gp/product/customer-reviews/0735605351/ref=cm_rev_sort/103-3779898-9131851?customer-reviews.sort_by=-HelpfulVotes&x=3&y=12&s=books .

Michael Covington responded to the question “does expensive == high quality?” He thinks cheap and expensive software is the best; the mid-range stuff tends to be inferior. His analysis is here:

<http://www.covingtoninnovations.com/michael/blog/0602/index.html#060212>

Tool Guardians

One difference between hardware and firmware development is the shelf life of the tools. Even when pushing the bleeding edge of technology, a five year old scope or logic analyzer is usually more than adequate. Some dedicated folks still crank out 8 bit systems using old Tektronix 545s - 30 year old vacuum tube beasts that invite a hernia each time they’re moved.

Our software tools have lifetimes of microseconds. Microsoft’s regular Patch Tuesday creates an organically-evolving OS and Office suite. Much of the current embedded hardware technology is all designed to speed firmware upgrades - flash memory lets us reprogram devices on the fly.

Of all the tools we use, though, compilers are the most problematic. Compilers create the code that is our product. A line of C is nothing more than a series of commands asking (pleading?) for a stream of machine code that implements an algorithm. Most other tools, like debuggers, fly in an orbit around the designer, helping him design and implement the product, but never becoming a part of the unit a customer sees.

A compiler problem becomes a product problem. A compiler change changes the product. When your compiler vendor releases a version that generates tighter code, simply recompiling changes the product. Though the tighter code is nice, it raises the specter of new, lurking bugs.

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

Before the vendors start sending unibomber email, I'm not implying that compilers are bug-ridden monstrosities. In my travels I see system after system that works until something changes. It may be a compiler upgrade, or perhaps a different set of parts used in manufacturing. Bugs lurk everywhere, and a compiler change often uncovers problems previously hiding in your code. Sometimes the risks of upgrading outweighs the benefits.

The problem is more severe for older products. If your five year old widget works great, and only needs a very occasional software tweak, do you instantly rebuild the code on each new compiler release? This implies a product change, with perhaps little benefit. Or, do you wait till a crisis requiring a code change hits, and then both fix the bug and use the new version... perhaps changing a ten minute fix to a week of hell?

The right thing to do – technically speaking – would probably be to rebuild all the old stuff. Who can afford that? The testing alone consumes vast amounts of valuable development time.

The risk is high, the benefits are vague. We need compiler guardians who keeps old versions around, with associated antique linkers and other support tools.

Perhaps “compiler guardian” is too narrow a focus. Most tools require some level of attention and management. Someone, or some group of people, have to keep all electronic tools metaphorically sharp. One friend told me this week that the old DOS editor he’s been using for 20 years still runs under Windows XP, but for some reason sucks so much processor power his laptop batteries die very quickly. It’s ironic that some change in Windows means this editor, which ran fine on an 8088 at 4.77 MHz, sucks every compute cycle a 2 GHz Pentium can provide.

Take for instance a PCB layout program, a staple of every engineering department. I was bitten by an upgrade to one of these products, and discovered to my horror that all of the libraries were now obsolete. What a joy it is to maintain both old and new versions of the software.

One friend addresses these sorts of problems by never upgrading. When the path becomes too rocky he idles, using the old but presumably reliable software for years after the vendor declares it obsolete. I worry that getting stuck in such a tool time warp will backfire. Support becomes non-existent.

However, “good enough” may be all that’s required, as the goal of engineering is to get a product out on time, on budget, that meets specifications. Using the latest and greatest goodies does, I feel, keep us more productive. Does the continual upgrade cost balance the productivity increases? I wish I knew.

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

Stack Overflows

I recently wrote an article about stack management (<http://embedded.com/showArticle.jhtml?articleID=193501793>). Ben Jackson had an interesting take on the subject:

“I just saw your article about catching stack overflow. One trick I used last year is to use GCC's '-finstrument-functions' flag which brackets every compiled function with enter/exit function calls. You can write an enter function which tests the amount of stack space left. It's not ideal, but it can save a lot of time. After I did this for the kernel to track down a problem, one of the application engineers applied the idea to a thread-intensive application and found several potential problems.

“Another trick is to look for functions where people have mistakenly put large structures or arrays on the stack. Use a tool like 'objdump' to disassemble your program and then a one-line awk or perl script to find the first 'sub...esp' in each function and sort them by size. If you find an irq with 'struct huge', it's just waiting to blow your stack.”

Jobs!

Let me know if you're hiring firmware or embedded designers. No recruiters please, and I reserve the right to edit ads to fit the format and intents of this newsletter.

WiQuest (Allen, TX), a fabless semiconductor company that develops complete solutions for the emerging ultrawideband (UWB) market, is seeking a Senior Firmware Engineer. Job responsibilities include developing firmware for WiQuest's ultrawideband communications products, participating in the development and documentation of system software architecture, verifying correct functioning of both the aforementioned software and WiQuest's hardware, and field support.

Experience with developing embedded firmware for high speed communications is a must. Software implementation experience with one or more of the following is desirable: IEEE 802 Local Area Network (LAN), IEEE 802 Personal Area Network (PAN), wireless Medium Access Control (MAC) protocols such as IEEE 802.11 Wireless Ethernet and/or IEEE 802.15.3 or WiMedia Wireless PAN. Experience with the following is a plus: USB, WUSB, embedded operating systems, ARM or MIPS processors, high speed bus master host interfaces, and software field support. A B.S. degree in Computer Science or a related field (or equivalent work experience) and outstanding technical and communication skills are required. In addition, 5 to 10 years of technical experience in software development is required.

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

email: resumes@wquest.com

ITT, which is into remote sensing systems for image information and communication for government and commercial customers, is currently recruiting for experienced professionals. The Space Systems Division offers competitive salary/ benefits and the opportunity to work with highly professional scientists and engineers on challenging programs in State-Of-The-Art remote sensing applications.

Several key positions in embedded systems are linked below.

<http://sh.webhire.com/servlet/av/jd?ai=591&ji=1886961&sn=I&tf=JobDescriptionE.html>

<http://sh.webhire.com/servlet/av/jd?ai=591&ji=1712736&sn=I&tf=JobDescriptionE.html>

<http://sh.webhire.com/servlet/av/jd?ai=591&ji=1831499&sn=I&tf=JobDescriptionE.html>

<http://sh.webhire.com/servlet/av/jd?ai=591&ji=1906948&sn=I&tf=JobDescriptionE.html>

<http://sh.webhire.com/servlet/av/jd?ai=591&ji=1906975&sn=I&tf=JobDescriptionE.html>

Meso Scale Diagnostics, LLC. (Gaithersburg, MD) has an opening for an electrical engineer. Candidate will work in a multi-disciplinary environment to develop hardware and firmware for compact, portable systems for clinical and biodefense applications. Duties will include prototyping of new concepts, all aspects of analog and digital design, and development of test plans and fixtures for transitioning the design to manufacturing. Candidate will also participate in performance, reliability, and verification/validation testing.

Fluency in schematic capture and PCB layout using Altium Designer /Protel DXP and C/C++ programming is required. Familiarity with ARM and 8051 microcontrollers is preferred. Prior experience in the development of in vitro diagnostic products is a plus. Candidate should have strong analytical and communication skills. A BS or MS in electrical engineering or computer science with at least two years relevant experience is required.

Contact Peter Adams padams@meso-scale.com

Dynon Avionics is seeking an embedded software engineer to contribute to the development of our product line of general aviation instruments. Dynon is a small

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

company (in Seattle area) where each person makes a large impact. The position requires someone who can work with people from various disciplines to develop products from concept to market.

The ideal candidate will have experience in developing embedded C code for an ARM processor platform but a variety of backgrounds will be considered. Desired Skills/Experience:

- * EE/CE/CS bachelor's degree
- * 3-6 years experience
- * Experience with embedded C software development
- * Real Time Embedded operating systems concepts and low level programming
- * Knowledge of embedded digital hardware architectures
- * Knowledge of communication protocols
- * Experience with ARM processor development
- * High energy level and work ethic
- * Aviation background a plus
- * Experience with OpenGL a plus

Send resume to employment@dynonavionics.com

Cisco in San Jose, California has an exciting opportunity to work on their next generation operating system. IOS-XR is a microkernel which can run on a single node processor or in a distributed environment consisting of thousands of nodes. The candidate will be involved in developing IOS-XR management plane software which is critical to Cisco's success in multi billion dollar high end core and next generation edge routing, switching and advanced technologies market.

The right candidate will have the following:

- * 5+ years experience in C programming
- * 5+ years of Real-time systems development experience
- * Strong knowledge of operating systems architecture
- * UNIX development
- * Experience with Clearcase or similar version control system
- * Familiarity with POSIX threads and micro-kernel desirable
- * Typically requires MSEE/CS combined with 5-7 years of related experience, or BSEE/CS combined with 7-10+ yrs related experience

Joke for the Week

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

ENGINEERS TAKE THE FUN OUT OF CHRISTMAS

There are approximately two billion children (persons under 18) in the world. However, since Santa does not visit children of Muslim, Hindu, Jewish or Buddhist (except maybe in Japan) religions, this reduces the workload for Christmas night to 15% of the total, or 378 million (according to the population reference bureau).

Santa has about 31 hours of Christmas to work with, thanks to the different time zones and the rotation of the earth, assuming east to west (which seems logical). This works out to 967.7 visits per second at an average (census) rate of 3.5 children per household, which comes to 108 million homes, presuming there is at least 1 good child in each. This is to say that for each Christian household with a good child, Santa has around 1/1000th of a second to park the sleigh, hop out, jump down the chimney, fill the stocking, distribute the remaining presents under the tree, eat whatever snacks have been left for him, get back up the chimney, jump into the sleigh and get on to the next house.

Assuming that each of these 108 million stops is evenly distributed around the earth (which, of course, we know to be false, but will accept for the purposes of our calculations), we are now talking about 0.78 miles per household; a total trip of 75.5 million miles, not counting bathroom stops or breaks. This means Santa's sleigh is moving at 650 miles per second or 3,000 times the speed of sound. For purposes of comparison, the fastest man made vehicle, the Ulysses space probe, moves at a poky 27.4 miles per second, and a conventional reindeer can run (at best) 15 miles per hour.

The payload of the sleigh adds another interesting element. Assuming that each child gets nothing more than a medium sized LEGO set (two pounds), the sleigh is carrying over 500 thousand tons, not counting Santa himself. On land, a conventional reindeer can pull no more than 300 pounds. Even granting that the "flying" reindeer can pull 10 times the normal amount, the job can't be done with eight or even nine of them, Santa would need 360,000 of them. This increases the payload, not counting the weight of the sleigh, another 54,000 tons, or roughly seven times the weight of the Queen Elizabeth (the ship, not the monarch). A mass of nearly 600,000 tons traveling at 650 miles per second creates enormous air resistance - this would heat up the reindeer in the same fashion as a spacecraft re-entering the earth's atmosphere. The lead pair of reindeer would adsorb 14.3 quintillion joules of energy per second each. In short, they would burst into flames almost instantaneously, exposing the reindeer behind them and creating deafening sonic booms in their wake. The entire reindeer team would be vaporized within 4.26 thousandths of a second, or right about the time Santa reached the fifth house on his trip.

Not that it matters, however, since Santa, as a result of accelerating from a dead stop to 650 miles/second in .001 seconds, would be subjected to acceleration forces of 17,000 g's. A 250 pound Santa (which seems ludicrously slim considering all the high calorie snacks he must have consumed over the years) would be pinned to the back of the sleigh

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

by 4,315,015 pounds of force, instantly crushing his bones and organs and reducing him to a quivering blob of pink goo. Therefore, if Santa did exist, he's dead now.

Merry Christmas

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*”. ". BUT - please use YOUR email address in place of “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.

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.