The Embedded Muse 18

Editor: Jack Ganssle   (jack@ganssle.com)   March 11, 1998

Editor's Notes

A lot of readers contributed fantastic "dumb mistakes" to this newsletter over the course of quite a few issues. Now I'd like to start something a bit different - Embedded Rules of Thumb. A handful are listed in the next section, and I invite y'all to contribute more.

Most professions have "rules of thumb" that are shortcuts, really - quick ways to understand something, or to make decisions without doing a lot of analysis. Architects, for example, rarely compute floor loading when designing a house. Instead they know that a 10 foot unsupported span needs a certain beam configuration. Doctors know that when your temperature exceeds normal by 2 or three degrees you're most likely sick; when it's 10 degrees high it's time for intensive care.

We embedded folks often skip the analysis of complex problems and shoot from the hip. That's often not a bad approach when such a decision is grounded in experience, and when a precise answer isn't required. To do this effectively we develop rules of thumb. Instead of re-inventing the rules all the time, I'd like to share some of mine and hope that others will contribute.

None of these numbers are meant to be particularly accurate; they're more guidelines to help our intuition. "Sanity checks", if you will, that let us come up with quick answers or check ones resulting from detailed analysis.

Embedded Rules of Thumb

When specifying a system, we know that it's pretty much impossible to guesstimate the size of a function if it'll be more than around 50 lines of code. Therefore, A document that contains one line of "function xxx is 50 lines of code" per line will be around - or at least - 1 page per 2500 lines of estimated code.

A line of code typically costs $10 to $20 done to commercial standards. So, that one page spec document represents something like $25-50k worth of development.

Software emulations of floating point math on an 8 bit processor will take around 1 msec per operation. Trig and other complex math is slower.

The Ganssle Group, www.ganssle.com

RS-232 characters take about 1 msec per character at 9600 baud.

Figure on about 5 bugs per 100 lines of code… if you're a careful coder.

Around 20% of your code will be junk. It might work, but it'll be bug-ridden and a pain to complete.

The speed of light in wire or on a PCB trace is about 2 nsec per foot.

Thought for the Week

Top 20 Replies by Programmers when their programs do not work:

20. "That's weird..."
19. "It's never done that before."
18. "It worked yesterday."
17. "How is that possible?"
16. "It must be a hardware problem."
15. "What did you type in wrong to get it to crash?"
14. "There is something funky in your data."
13. "I haven't touched that module in weeks!"
12. "You must have the wrong version."
11. "It's just some unlucky coincidence."
10. "I can't test everything!"
 9. "THIS can't be the source of THAT."
 8. "It works, but it hasn't been tested."
 7. "Somebody must have changed my code."
 6. "Did you check for a virus on your system?"
 5. "Even though it doesn't work, how does it feel?"
 4. "You can't use that version on your system."
 3. "Why do you want to do it that way?"
 2. "Where were you when the program blew up?"
 1. "I thought I fixed that."

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.

The Ganssle Group, www.ganssle.com