

The Embedded Muse 63

Editor: Jack Ganssle (jack@ganssle.com)

May 7, 2001

Memory Allocation

If you use malloc and free in embedded systems I'm sure you're aware of the perils of heap fragmentation – even perfectly-balanced mallocs/frees may result in a heap with plenty of available memory, but broken into chunks too small to be useable.

Some folks don't use these constructs at all, a very safe alternative. Yet sometimes we don't have much choice. What do you do if you've purchased a package (say, a protocol stack) that uses these?

I figure if we're going to use a potentially-problematic resource we'd better instrument our code to find the problems before they turn into deadly bugs. One free allocation monitor is "mem.c", which replaces the C library's malloc/free with instrumented versions. It will detect most common problems. It is usually available at <http://www.snippets.org>, but this site has lately been down as it's being re-engineered.

An alternative is at <http://www.fourmilab.ch/smartall/>. It's just a few hundred lines of C that very intelligently picks up all sorts of memory allocation problems. While targeted at workstations and MS-DOS, porting the code to an embedded application is pretty trivial.

Thought for the Week

Programmer to Team Leader:

"We can't do this proposed project. ****CAN NOT**** It will involve a major design change and no one in our team knows the design of this legacy system. And above that, nobody in our company knows the language in which this application has been written. So even if somebody wants to work on it, they can't. If you ask my personal opinion, the company should never take these type of projects"

Team Leader to Project Manager:

"This project will involve a design change. Currently, we don't have any staff with experience in this type of work. Also, the language is unfamiliar to us, so we will have to arrange for some training if we take this project. In my personal opinion, we are not ready to take on a project of this nature."

Copyright 2001 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

Project Manager to 1st Level Manager:

"This project involves a design change in the system and we don't have much experience in that area. Also, not many people in our company are appropriately trained for it. In my personal opinion, we might be able to do the project but we would need more time than usual to complete it."

1st Level Manager to Senior Level Manager:

"This project involves design re-engineering. We have some people who have worked in this area and others who know the implementation language. So they can train other people. In my personal opinion we should take this project, but with caution."

Senior Level Manager to CEO:

"This project will demonstrate to the industry our capabilities in remodeling the design of a complete legacy system. We have all the necessary skills and people to execute this project successfully. Some people have already given in-house training in this area to other staff members. In my personal opinion, we should not let this project slip by us under any circumstances."

CEO to Client:

"This is the type of project in which our company specializes. We have executed many projects of the same nature for many large clients. Trust me when I say that we are the most competent firm in the industry for doing this kind of work. It is my personal opinion that we can execute this project successfully and well within the given time frame."

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.

Copyright 2001 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

The Ganssle Group, www.ganssle.com