Embedded experts: Fix code bugs or cost lives

Rick Merritt
(04/10/2006 10:00 AM EDT)
URL: http://www.eetimes.com/showArticle.jhtml?articleID=184429901

San Jose, Calif. — The Therac 25 was supposed to save lives by zapping tumors with targeted blasts of radiation. Instead, the device delivered massive overdoses that killed three patients and injured several others because of software glitches by a lone programmer whose code was never properly inspected and tested.

The Therac 25 was just one of dozens of examples cited by speakers at last week's Embedded Systems Conference here to drive home a point: People's lives as well as millions of dollars in investments often depend on software engineering, but too many projects fail for lack of good programming discipline and management support.

And the problems may get worse as programmers face the additional challenges of handling multicore devices. Indeed, an annual survey of several thousand embedded engineers polled recently by EE Times and Embedded Systems Design magazine showed that the need for better software debug tools is a major concern, with test and debug taking up more time than any step in a project development.

"This is the only industry left where we can ship products with known defects and not get sued. How long do you think that will last?" asked Jack Ganssle, a consultant and author who presented a class on lessons learned from embedded-software disasters.

"We aren't afraid of software, but we need to be, because one wrong bit out of 100 million can cause people to die," said Ganssle, who said he has worked on more than 100 embedded projects, including the White House security system.

"As embedded systems grow in complexity, the software becomes an ever more important piece. Right now, 50 percent of our DSP spending is on the software side," said Gerald McGuire, general manager of the DSP group at Analog Devices Inc. (Norwood, Mass.), which employs more than 200 software engineers.

As software grows in importance, it is not necessarily becoming more reliable. According to one report, 80 percent of software projects fail because they are over budget, late, missing key features or a combination of factors. Another report suggests that large software systems of more than a million lines of code may have as many as 20,000 errors, 1,800 of them still unresolved after a year.

"We can't get rid of faults," said Lorenzo Fasanelli, a senior embedded-software specialist for Ericsson Labs in Italy. But engineers can speak up about faults, learn from them and rewrite code to proactively find and minimize them, he added.

"We cannot advance the state of the art without studying failure," said Kim Fowler, an author and systems architect who delivered an ESC talk called "Fantastic Failures."

War stories

There are plenty of failures from which to learn. Ganssle cited another radiation system that killed 28 people in a series of tests in Panama in May 2001 before the U.S. Food and Drug Administration shut down the company that made it. Inspections of software after the crash of a U.S. Army Chinook helicopter revealed 500 errors, including 50 critical ones, in just the first 17 percent of code tested.

"Why did they inspect software only after people died?" asked Ganssle, who said a court case on the crash is still in litigation.

Some pacemakers have stimulated hearts to beat at rates of 190 beats a minute, prompting companies to provide software updates delivered to the implanted devices using capacitive coupling. Unfortunately, other pacemaker patients have had their devices inadvertently reprogrammed when walking through metal detectors. In 2003, the pacemaker of a woman in Japan was accidentally reprogrammed by her rice cooker.

A Thai politician had to have police bust the windows on his BMW 745i after a software glitch caused the electric doors and windows to freeze in a locked state, trapping him inside. Ford recalled some models of its 2000 Explorer because lights and wipers would not work in some circumstances. And the 2004 Pontiac Grand Prix faced a software recall for a leap-year fault.

Part of the problem lies in poor engineering discipline, such as a lack of adequate testing, improper error handling and inherently sloppy languages. Management issues, including a demand for ever more features in compressed schedules, and tight budgets are also to blame.

"We need to test everything up front and integrate testing into the design process. Then we need to believe the data we get when we do test," said Ganssle.

When engineers make a change because of a failed test, they often neglect to go back to the beginning of the test suite to make sure the changes haven't introduced new errors, said Dave Stewart, chief technology officer of Embedded Research Solutions Inc. (Annapolis, Md.) in an ESC session on the top problems in real-time software design.

Engineers need to create error-handling modes in their programs, and the modes must exist as just another state for their systems and treat errors as one of many possible inputs, Stewart added.

Fasanelli of Ericsson gave a detailed prescription for how to find, report and minimize faults in embedded software. Programmers must make it a standard practice to classify all inputs and states of a system and note any illegal inputs or edge states, whether or not they affect a program's ability to run, he said.

In addition, programs should routinely track and report their own performance, idle times and memory integrity. Creating such debug features may affect a system's cost, but that will be offset by reduced maintenance, Fasanelli said.

"Exception handling is particularly hard to test because it's hard to generate the exceptions. These tend to be the most poorly tested parts of code," said Ganssle.

Riding a rough C

Ironically, today's most popular programming languages, C and C++, are among the most error prone. That's because C compilers have plenty of latitude to compile and link — without providing any diagnostics — code that can produce serious run-time errors, especially when ported to a new processor.

"There are a lot of little goodies in C that programmers are not fully aware of," said Dan Saks, an author who has documented nearly 40 "gotchas" he presented in a session at ESC. "The lesson is to understand what you can assume and what you can't."

For instance, C doesn't define the number of bits in a byte, though header files can query a processor and adjust the program if the CPU does not support the usual 8-bit byte. Likewise, the common practice of subtracting pointers can result in creating a character of an undefined type, said Saks, president of consulting firm Saks & Associates (Springfield, Ohio).

"The use of C is really criminal," said Ganssle. "C will compile a telephone directory, practically. I guess we use C because we think debugging is fun."

For every 1,000 lines of code, C can generate 500 errors in a worst case, 167 errors on average or 12.5 mistakes for automatically generated code, said Ganssle. That compares with 50 errors worst case, 25 average and 4.8 for auto-generated code using the Ada language, he said. The Spark language emerging from Europe is even better, generating just four errors on average per 1,000 lines of code, he claimed.

C is used in half the development projects done today, according to the results from the 2006 Embedded Market Survey, the 14th such annual poll of engineers working on embedded-design projects. The survey showed that the C++ programming language is gaining in acceptance, however.

ESD editor-in-chief Jim Turley, who presented the annual embedded-market survey results last week, said fully half of the respondents cited C as their primary programming language. Nonetheless, support for C was down from the 2005 survey, albeit only by 3 percent. By contrast, the C++ language gained this year, coming in at 28 percent, and respondents predicted a 4 percent increase in C++ adoption next year.

The survey showed that relatively few engineers — just a few percent — use Java. Matlab, LabView and UML are used about as frequently for embedded projects, although Java garners more attention because of its use in the graphical user interface portion of many systems.

"Almost every language is losing ground to C++," said Turley, who suggested that many design teams have evaluated Java but found it lacking in performance and development tools.

Asked about tool selection, 53 percent of embedded engineers said the quality of the debugger was their most important criterion in choosing a development suite. Only about 13 percent said open-source content is an important selection criterion.

When it comes to operating systems, however, open-source OSes such as Linux are gaining significant support. Fully 20 percent of respondents said they use an open-source OS, with many design teams relying on a commercially distributed form of Linux.

Turley said that one reading of the operating system responses would suggest Linux is gaining support quickly, since "just five years ago the very term 'open source' didn't mean anything." However, other survey questions showed that a declining number of respondents, compared with the 2005 survey, are considering Linux, prompting Turley to conclude that "that the charm of Linux has cooled."

Management must take its share of the blame for the software situation. "Often we are in an overconstrained situation. We have too many features to deliver in too short a time frame," said Fowler at his "Fantastic Failures" session. "The problem is, adding features requires lots of regression testing. The thing to do is ask whether the feature can be saved for the next upgrade — [otherwise] you are just setting yourself up for failure.

"We as engineers need to come up with persuasive ways to warn management" by relating stories of past failures or the implications of long feature lists and tight budgets and schedules, he added.

Tired engineers were a factor in several aerospace disasters in which programmers worked 60- to 80-hour weeks in the months before a launch, Ganssle said.

Skimpy budgeting is another factor in failures, as seen most clearly in civil-engineering disasters. In 1940, officials found a way to build the Tacoma Narrows Bridge for half an initial estimate and did so, but the bridge famously collapsed in high winds after just four months in service. Likewise, the MGM Grand Hotel in Las Vegas saved $200,000 by not using sprinklers but paid out more than $200 million in court and rebuilding costs after a disastrous fire, Ganssle said.

In the confines of a software project, "spending $2,000 on tools might save you $100,000 in programming effort," said Stewart of Embedded Research Solutions.

Multicore effort

Activity on the ESC show floor demonstrated that embedded-software tool vendors are increasingly dealing with issues arising from multicore and multithreading architectures. Both Mentor Graphics Corp. and Green Hills Software Inc. said they have added support for the MIPS32 34K multithreading processor core family, for example. Green Hills, which announced its support for Texas Instruments Inc.'s DaVinci architecture earlier this year, rolled out MIPS32 34K support with its Multi development tools. Green Hills also added support for the single-core MIPS32 24KE family.

QNX Software Systems Ltd. last week announced support for DaVinci, which combines ARM and DSP cores in order to support digital audio and video applications. To help maximize performance, QNX will support an interface layer between the cores based on TI's DSP/BIOS Link technology. This makes it possible to offload media processing to the DSP, freeing up the ARM core for other applications.

The Ottawa company launched a multicore initiative last fall, said Dave Curley, vice president of marketing at QNX. Using the company's Neutrino real-time operating system (RTOS) and the

Momentics IDE, this initiative supports asymmetric, symmetric and bound multiprocessing (BMP). The latter capability, Curley said, is unique, and it lets programmers assign threads to a given processor.

"One of the challenges of multicore is to understand how to work in a multithreaded environment," Curley said. "With BMP, you can tie legacy code to one processor without a rewrite."

QNX's Multi-Core Expedite Program, announced last week, provides 120-day free evaluations of the QNX Neutrino multicore technology development kit and dual-core Intel Pentium processor extreme edition.

Virtutech Inc., a provider of virtual platforms for early software development, claimed last week to have the first simulation model of the Freescale Semiconductor Inc. MPC8641D dual-core processor. Wind River is using this simulation model in its engineering department to develop multicore versions of its own products.

"Multicore is a complete revolution for software people," said Paul McLellan, vice president of marketing at Virtutech (San Jose). "You turn off interrupts for one core, but another core carries on." He noted that Virtutech's Simics environment lets users freeze the entire system when a breakpoint is hit, unlike real hardware, where processors would take some period of time to shut down.

ARM Ltd. said in a press release that its new RealView 3.0 development suite adds a debug engine with "multicore DSP awareness." Bryn Perry, general manager for development systems at ARM, clarified that ARM has the "potential" to support DSP debugging. ARM is working with DSP processor vendors but has not yet announced support for a specific digital signal processor Perry said.

As of today, RealView can connect ARM and DSP debuggers and synchronize them. But some customers want a single debug view of the entire system, and for that ARM needs to partner with DSP vendors, Perry noted.

The MIPS32 34K family is technically not a multicore solution, but it's touted as a multitasking architecture that can provide the benefits of multicore. Mentor Graphics announced last week that its Nucleus RTOS and Eclipse-based Edge tool suite now support that family. In the 34K device, one instance of the Nucleus Plus RTOS runs in each of two virtual processing elements (VPEs). The Nucleus Plus on the first VPE initializes the second and controls all peripheral resources.

— Additional reporting by Richard Goering and David Lammers