

Letters to the Editor

Our May 12 editorial on Java generated a flurry of letters from readers. A sampling of these is reprinted here.

I read Jim Turley's recent editorial with considerable interest and amusement. For the Java™ revolution, Turley seems to have appointed himself the industry's curmudgeon, much [as] Nick Tredennick [did] for the RISC revolution some years ago (also in the pages of *Microprocessor Report*). I predict Turley's perspective will enjoy about as much popularity.

... It is true that the notion of a virtual machine was not successful in an era when the Internet was ARPANet, PCs were still novelties, and the most powerful microprocessor was a 4-MHz Z80. Today's computing landscape is vastly different. Old ideas sometimes catch fire in a new time and place. Enough ingenuity applied to the problem of executing bytecodes may substantially reduce their drawbacks. ISAs evolve with time. If free Java programs are unappealing, plenty of companies are willing to charge for them.

... The sheer weight of the interest in and enthusiasm for Java should give pause to anyone who supposes things are satisfactory as they are. Happy campers rarely rise up and attack the cooks. The embedded world is awash in microprocessors, but increasingly, rapid production makes porting a nightmare, while on the desktop, the world increasingly narrows to just one choice. Indeed, the point I considered most evident is that there is a lot of dissatisfaction out there with the status quo in computing. If nothing else, the Java technology "hype" that has Turley so steamed is surely a measure of the breadth and depth of that dissatisfaction.

—Harlan McGhan, *Sun Microelectronics*

While I'm flattered to be compared to an IEEE fellow, I agree that my views on Java are as unpopular as Dr. Tredennick's were at the time. Let us not forget, however, that Nick was right.—J.T.

If there's one thing I love, it's when a publication I really respect confirms everything I've suspected on a subject.... The whole idea of this virtual machine strikes me as 100% Pure Idiocy. It's everything the market has rejected over the last 20 years. People want programs that take advantage of the latest hardware, and they want it to run fast.

Like you say, bytecode cannot possibly run as fast (or, as we really know, anywhere near as fast) as native code. The p-system should have taught us something about that.

I think it's worth noting that once you give up on the idea of bytecode, you also give up on most of the other *causes celebres* of the Java Jihad: portability, the bytecode verifier, the sandbox. Let's consider those causes:

- Portability is substantially a lie. [PC Labs] testing showed that the Win32 virtual machines are faster, more

stable, and more compatible than the rest. This makes perfect sense, as they get the most attention from vendors because they have the biggest customer bases.

- The bytecode verifier. Pardon me, but didn't Turing or someone like that essentially prove that the job of this verifier program is essentially impossible?

- The sandbox. The sandbox metaphor is apt in many ways: Once Java programs grow out of childhood, they must leave the sandbox. You can't do real work in there.

You briefly mentioned Java chips in your story. I've been wondering why this most stupid idea in the entire Java farce has gotten off so easy in *Microprocessor Report*.... It's not just a stupid idea, but a cynical one. Sun was one of the first companies to make a buck in the RISC business, and now they push this most anti-RISC of architectures (complex variable-length instructions, unaligned data, objects in the instruction set), simply because it's getting good press. Even if you brush aside the cost/performance problems, you get back to the portability problem. What happens if they want to add an instruction to the Java VM (oops, I mean the "M")? How would the old chips run the new code?

"100% Pure" doesn't guarantee that it will actually run. [PC Magazine] Java tests show this. JMark was totally pure, yet it ran without problems only on Win32 platforms. Sun knows this to be true. As the saying goes, portable is as portable does.

Another thing. We all know that this is about grasping at anything that can damage Microsoft, not something that will bring benefits to users. Scott McNealy and Larry Ellison will happily tell you this, even on the record. Someone ought to call them on it.

—Name withheld by request

You're writing on a technical subject? Please, give me a break. Comparing bytecode instructions to GWBasic tokens? You said it all right there. Please be aware of your limits—write about something you have some passing familiarity with.

—Daniel Phillips, *independent consultant*

Having recently listened to an otherwise intelligent executive talk with a straight face about smart cards running Java, I *really* appreciated your editorial!

—David Fair, *Philips Semiconductor*

Bravo! Bravo! It is good to see someone address this. When Intel speeds up a processor by 20%, all previous processors are obsolete. It is a breakthrough. But when software saps another 300%, it is a "breakthrough" also. Go figure.

—Scott Duplichan, *Compaq*