Embedded.com

Operating systems on the rise

By Jim Turley, Courtesy of Embedded Systems Design

Jun 21 2006 (9:00 AM)
URL: http://www.embedded.com/showArticle.jhtml?articleID=187203732

Each year Embedded Systems Design *conducts a large-scale survey of embedded systems developers. We poll thousands of engineers, programmers, and managers to see what they're doing and how they're doing it. Starting this month we'll be publishing the results of our brand new 2006 research. This first month's installment reveals how developers choose and use operating systems in their embedded projects. In future issues we'll reveal some surprising data about microprocessors, development tools, favorite suppliers, programming languages, schedules, budgets, and more. This survey embodies the work of many people; it was complicated and expensive to conduct but we hope you enjoy the results.*

More on methodology used in survey

Operating systems. Nothing cuts to the heart of a development project like the choice of OS. Whether it's a tiny scheduler or kernel, an open-source distribution, a tightly wound real-time operating system, a fully featured commercial RTOS, or no OS at all, it drives all downstream software decisions and many hardware decisions as well.

We asked users about more than 25 different commercial OSes—plus Linux, in-house, and proprietary alternatives—and got as many different answers. There isn't much that all embedded systems have in common but some clear trends were revealed nonetheless. First off, more than a quarter of embedded systems now in development won't have an OS at all. None. As the pie chart in Figure 1 shows, a little over 28% of all our survey-takers said the system they're designing now won't have so much as a tiny scheduler or task switcher. That's the average across all the age groups, industries, company sizes, and experience levels we polled.

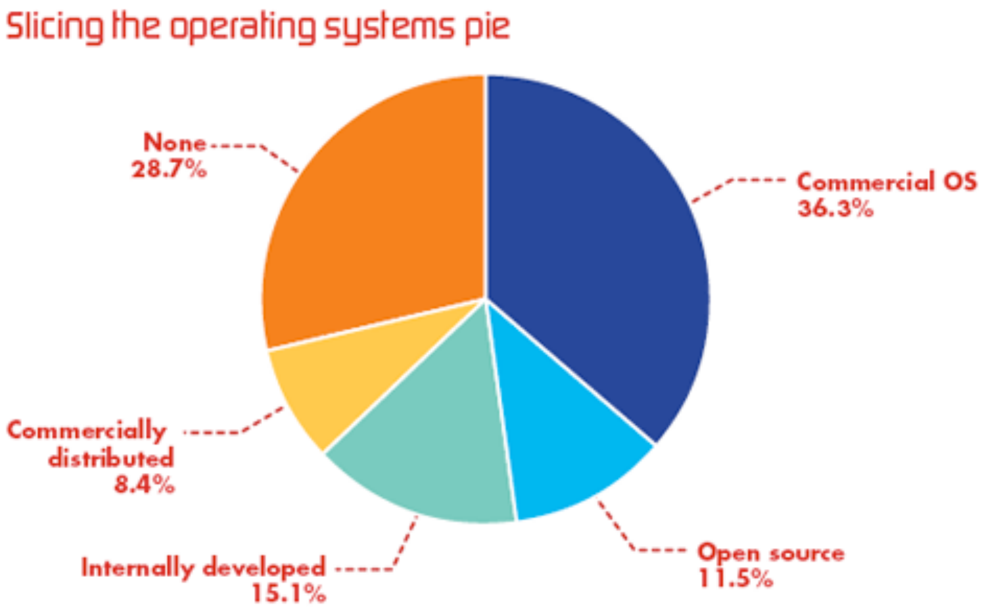

Figure 1

The no-OS percentage was highest among those developing consumer electronics, at just over 40%, considerably higher than the average. Developers in the automotive sector and industrial segment also eschewed OSes, casting a 'no' vote 36% and 35% of the time, respectively. At the opposite end of the spectrum, OSes were most likely to appear in computer peripherals, with 84% polling in the affirmative (16% using no OS).

There wasn't any correlation between developers' ages and their taste for OSes, but there was a connection between OS usage and company size. Perhaps predictably, larger organizations tend to use operating systems in their products more often than smaller companies do. If we ignore industry segment and other factors and look just at company size, small firms (under $50 million in annual revenue) deploy an OS about 62% of the time versus 82% for big (over $1 billion) firms.

Interestingly, on-the-job experience seems to play a role in operating-system usage. The more experienced the developer, the more likely he or she was to employ an OS in the current project.

Why no OS?

With hundreds of our respondents (and, by extension, many thousands of development teams) avoiding embedded operating systems the obvious question is, why? The answer was a straightforward, "we don't need one." More than 85% of the time, a simple lack of need was cited as the sole reason for skipping the operating-system layer. It's easy to think of many products, such as thermostats or microwave ovens, that might not need an OS so this isn't overly surprising. It's frustrating for operating-system vendors, though, who'll have a tough time converting that large portion of the market into potential candidates for their wares.

For the remaining non-OS believers, the reasons were varied. About 30% said an OS would put too much strain on their system's processor and/or memory, with about half blaming the CPU and the other half the RAM. A much smaller percentage (about 10%) said an OS was too expensive. The remaining 7% complained that operating systems are too complicated to use. Note that this final group didn't say that an OS was too complicated for their system (in other words, technical overkill). They said it was too complicated for them, personally. Whether it's true or not, this last response suggests that some OS-friendly marketing campaigns may be in order.

What type of OS?

Even with 28.7% of developers not using an OS, that still leaves over 71% who do. Of that group, 51% (or 36.3% of the total) use a commercial, off-the-shelf OS. This was by far the strongest response among OS users. The next-strongest response got just 21% of the vote (15.1% of total) from those using an internally developed, proprietary, or in-house OS. After that, 16% (11.5% of total) are using an open-source OS, and 11.8% (8.4% of total) use a commercial distribution of an open-source OS. In other words: Linux.

There's a lot we can make of this. First off, commercial OSes are more popular than ever and they're gaining ground. In our 2005 survey, only 44% of respondents were using a commercial OS, so commercial OS usage has grown significantly. Moreover, that growth is coming at the expense of in-house code. As products (and their owners) mature they're more likely to replace the in-house OS with a commercial alternative. We've seen this trend for many years, and it shows no signs of abating. Commercial OSes, as we'll see later, provide more than just technical advantages.

The one weak spot for commercial OSes was in automotive electronics. That industry gave the green light to commercial OSes just 33% of the time, versus the broader 50% approval rating across all product categories. Conversely, avionics was the strongest commercial-OS proponent, with a 64% usage rate.

Open-source enthusiasm waning?

If we mentally combine the open-source and commercial-distribution responses we get 28% of OS users, or about 20% of developers overall, actively using Linux or some other open-source OS. That's a healthy percentage of developers but it's actually lower than last year's result. In 2005, 37% of survey takers were working on projects based around open-source OSes. Figure 2 shows how usage of both commercial and noncommercial distributions of open-source OSes (mostly Linux) has declined sharply from last year.

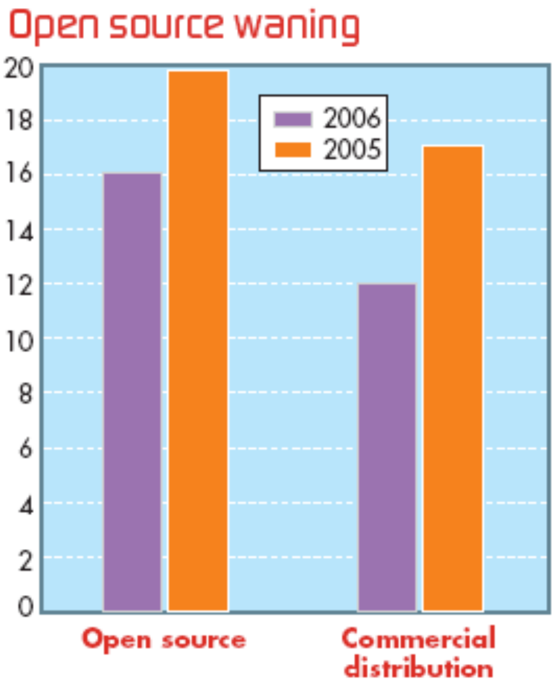

Figure 2

Apparently some disillusionment has set in. Figure 3 charts responses to one of our forward-looking questions. Instead of asking what they're using now, we asked developers what they're planning to use in their next project. Future plans to use open-source Linux have dropped by one-third, from 29% to just 19%, although plans to use commercial distributions of Linux have grown slightly to 17%, up from 14% in 2005. Last year's respondents seemed considerably more optimistic about Linux; this year's open-source plans are much more subdued.

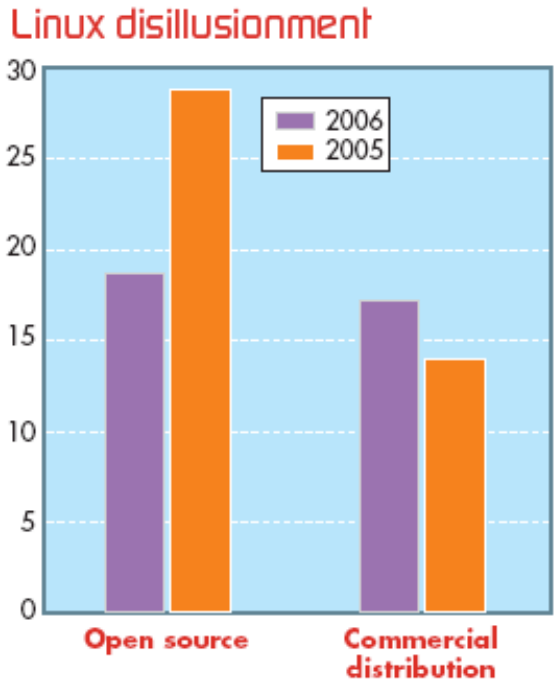

Figure 3

Open-source OSes exerted their strongest pull among engineers and programmers working in the manufacturing sector; they gave a thumbs-up to Linux more than 38% of the time. At the other end, automotive and avionics makers hated it, presenting a closed door to open source 88% of the time.

With so much interest in Linux and all things open-source, we asked developers what the appeal was. Among those expressing an interest in Linux (although not necessarily using it currently), we asked what characteristics made Linux attractive. More than 70% identified its low cost and 61% cited its adaptability and extensibility (the answers are not mutually exclusive). "Personal control" of features got a 45% vote while performance ranked fifth with 39% of respondents choosing that criterion.

Surprisingly, the fourth most-popular reason for considering Linux was that it "avoids the commercial alternatives." That hardly sounds like a ringing endorsement of Linux; more an indictment of the commercial RTOS vendors. Let's be blunt. Hundreds of our survey takers said to the RTOS vendors, in essence, "It's not because we like Linux, it's because we don't like you."

Among the anti-Linux league (those not considering it for their current or upcoming projects) the most commonly offered excuse was that Linux was incompatible with their existing software. Fair enough; almost any new OS would force a rewrite of existing drivers and applications and most developers (60%) weren't eager to make that change.

After that, the secondary responses fell off pretty rapidly. Poor performance and/or real-time capability were listed second most frequently (31%), followed by support concerns (28%), memory usage (23%), legal ambiguity (20%), the state of development tools (19%), and cost (11%). Interestingly, cost was cited by both camps, pro and con, as justification for their decision. Clearly, the cost of Linux is in the eye of the beholder.

Commercial OS landscape

Going back to our initial breakdown, the single largest group of developers is using a commercially available OS. What influenced their choice of OS and what product are they using? Here's what we found out.

The choice of OS, like the choice of processor, is often a group decision—but who gets to vote? As Figure 4 shows, there might be any number of people involved. The software manager (a single person) exerted "a great influence" on the choice of OS for 52% of our respondents, "some influence" for 35% of them, and no influence at all for the remaining 12% of developers. As far as a single person's influence on the decision, the software manager was it.

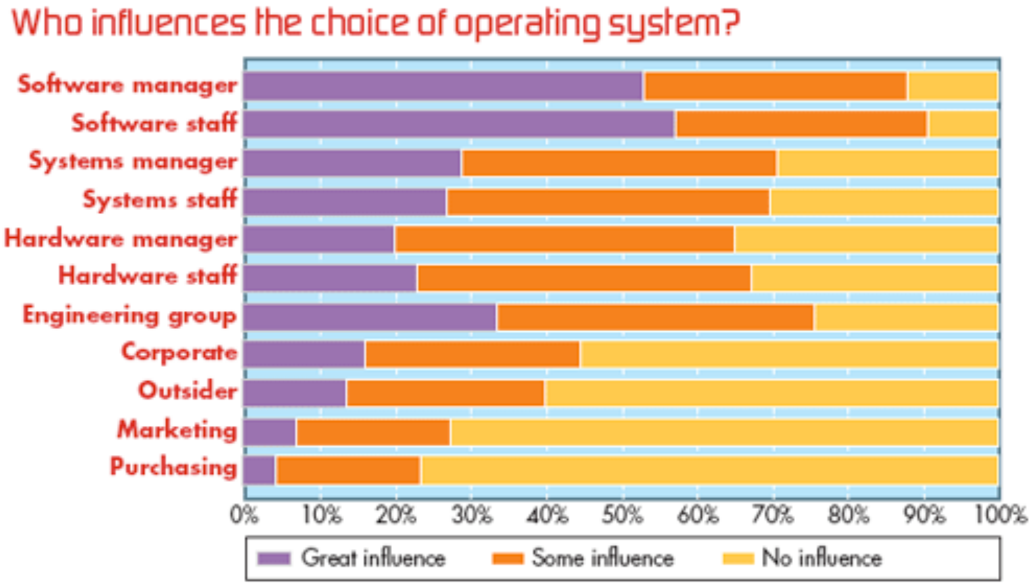

Figure 4

More likely, though, it was a group decision. The software staff had the greatest influence most of the time, as the second bar in the graph shows. Oftentimes the entire engineering group, hardware and software people together, had a big say in what OS they would be using. Corporate management called the shots only about 15% of the time, with corporate outsiders such as a customer or regulatory body having even less influence (13%), followed by the marketing (6%) and purchasing (4%) departments. Even these last two departments affected the decision about 20% of the time, showing that operating systems are not just technical products; they can affect the rest of the company as well.

So what are these people voting on? Figure 5 shows us what's important to those who get a say in the choice of OS. For most, real-time performance is the big issue. Right behind is processor compatibility. In other words, we can't use an OS that hasn't been ported to our preferred microprocessor, microcontroller, or DSP. That makes sense; compatibility is a pass/fail criterion for most developers. It's all the more significant, then, that a slight majority chose performance over compatibility as their hot-button issue.

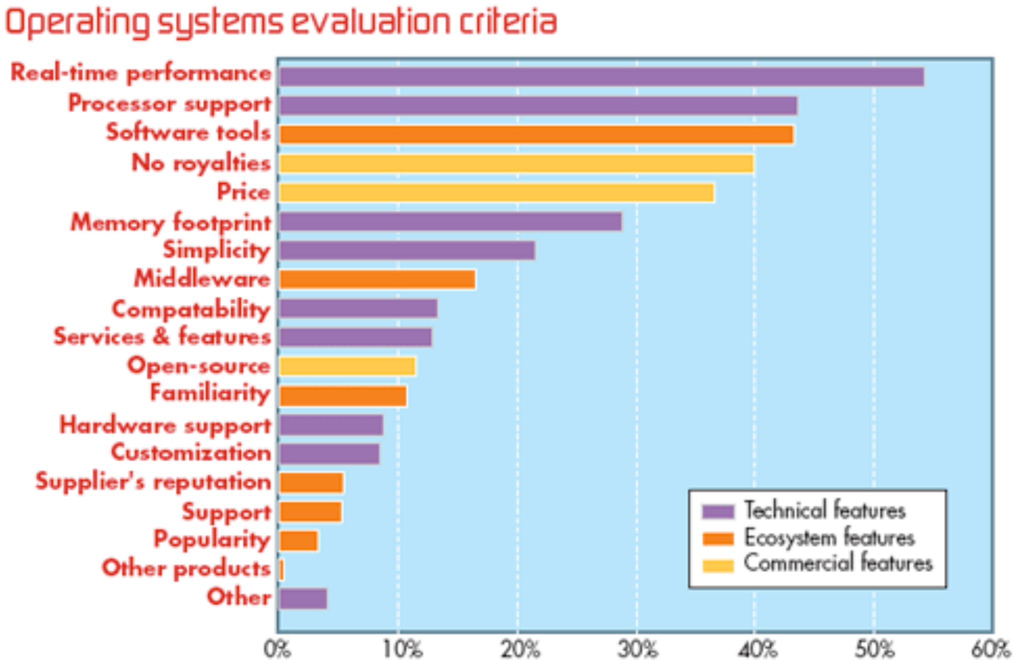

Figure 5

The availability of software tools effectively tied for second place, followed closely by royalty fees and purchase price. What's interesting here is that none of them has anything to do with the OS itself. They're all "ecosystem" issues surrounding the OS and affect the developer's quality of life. As we'll see in future installment of our survey results, non-technical criteria like support, trust, and reputation play an ever-larger role in previously technical decisions. Vendors, take note.

Memory footprint and simplicity both ranked highly, echoing the responses we saw from earlier questions. Memory usage, like performance, determines whether the OS is suitable for the task. Simplicity determines whether the developer should even bother.

The availability of "middleware," which often means networking capability, has moved up the rankings in importance as more developers work on network-enabled or communications-enabled embedded systems. The need for networking has grown just as rapidly as programmers' distaste for developing the necessary protocol stacks. Writing, debugging, and testing network code is dreary and uninteresting work for most programmers. Communications protocols are generally tightly defined and standards-driven so there's little room for creativity—but lots of room for error. Networking protocols are another pass/fail item: your network stack either works or it doesn't, and most managers would rather buy code that works than risk developing code that doesn't.

Commercial RTOS brands

So it all comes down to this, the all-important ranking of commercial OSes. Among commercial embedded operating systems, Wind River's VxWorks was the undisputed leader, as shown in Figure 6. With more than 25% of all the votes, it scored particularly well in avionics applications (where it ranked well above average) but somewhat poorer in automotive, computer-related, and industrial segments (where it scored well below average). Its popularity was insensitive to age but was strongly correlated to company size, with larger development firms using VxWorks much more frequently than small companies.

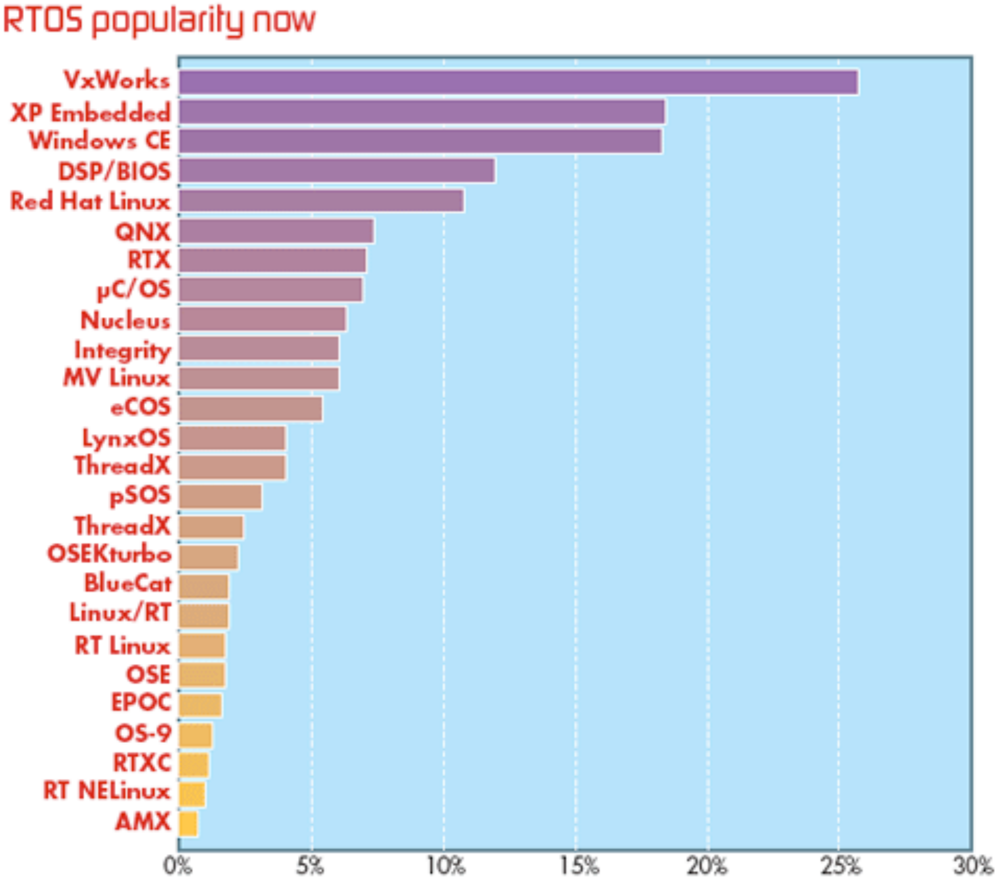

Figure 6

Tied for second place are the Microsoft twins, Windows XP for Embedded and Windows CE (there were separated by one vote). In a predictable reversal, WinXP and WinCE were most popular where VxWorks was weakest: industrial and computer-related applications. XPe was especially strong in manufacturing industries, where VxWorks also made a good showing.

In the number four spot we have Texas Instruments' DSP/BIOS, a product with obvious hardware ties. That's followed by Red Hat's Linux, the first of the many Linux variants to make the list.

After that, the numbers all drop to below 8%. QNX, RTX, C/OS, and Mentor Graphics' Nucleus RTOS head the list of several popular embedded OSes with single-digit market shares. ThreadX is deliberately listed twice: once for the Green Hills distribution and once for Express Logic's product. Oddly, Green Hills appears to sell more copies of ThreadX than the company that developed it.

Looking ahead, we asked those same survey takers what OS they'd consider using in their next project. Would they stay with their current OS or switch to another? We also asked non-OS users the same question to see if they were considering a commercial OS for the first time. The results are summarized in Figure 7.

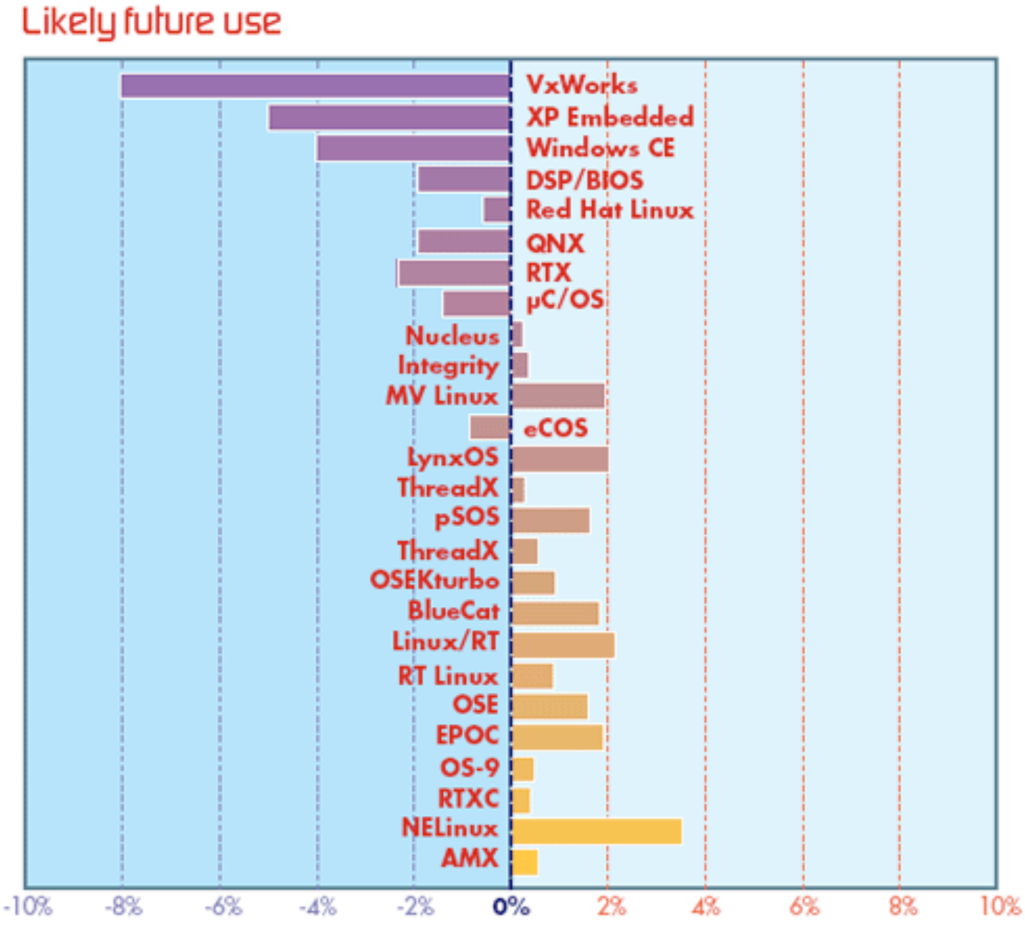

Figure 7

The chart shows the delta between the "would consider" responses and the "using now" responses from the previous graph. In brief, the winners lose and the losers win. How is that possible and what does it all mean?

First, the responses don't necessarily mean that VxWorks users are unhappy with their current RTOS and are ready to jump to (for example) LynxOS or Wind River's Platform NE Linux. We can't correlate the responses from the first question to the second; the person who said he's using VxWorks now may not be the same person who said he'd consider EPOC in the future. What we can say with certainty is that about 8% fewer people said they'd consider VxWorks compared with the number using it now. Whether or not those are the same people is impossible to tell.

Microsoft's two embedded OSes didn't fare well, either. In fact, the eight top choices all lost ground in the game of "guess tomorrow's operating system."

Reflecting a kind of Zen balance, almost all of the middle and low scorers gained share of mind in roughly equal proportion to the share lost by the big players. Whether accidentally or by design, the numbers reflect a zero-sum game. This gives us a hint as to the true nature of the data. Recall that more than 28% of embedded projects use no OS at all, and that this proportion has been shrinking steadily for years. Those developers have to go somewhere; what OS will they choose as their first? Logically they're going to choose a smaller, more inexpensive OS over a large and fully featured one. If you're upgrading your system from no OS at all it makes sense to start small. Thus, we see that the smaller RTOS vendors gain share of mind among potential next-generation projects.

Will the current commercial vendors lose some customers? Sure, just as every supplier occasionally loses customers to a competitor. But they'll probably gain even more new customers. Nothing sinister lurks in the data here; nothing to trouble the sleep of the friendly neighborhood RTOS marketing staff. OS loyalty is fairly low anyway. As the chart in Figure 8 shows, more than one-third (36%) of developers are using a different OS than they did in their previous project. Usually that's just because of a change in hardware (44%) or, more ominously, that the new OS was not the developers' choice and was forced upon them.

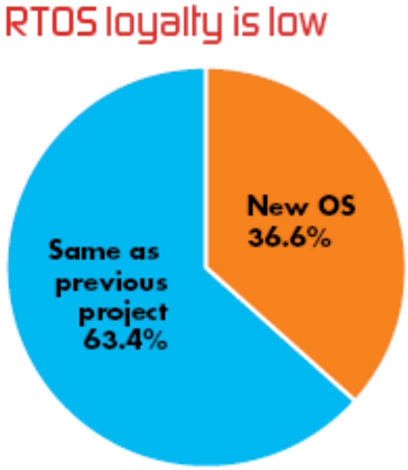

Figure 8

About two-thirds of the OS defections are what we might call voluntary changes. Twenty-seven percent said they'd switched operating systems because the new one "had better features," with 21% opting for "a better roadmap for growth" and 20% jumping ship for "better development tools." Another 15% said they bailed because the new OS was cheaper. Very few (8%) said their old OS was no longer available and even fewer (6%) switched because they were unhappy with their previous OS supplier. Fewer than 5% switched because the old system was too slow.

The overall picture is therefore good for commercial OS suppliers. As the number and type of embedded systems increase, so will the total available market. As in-house and proprietary OSes slowly ride into the sunset, development teams will be filling the gap mostly with commercial alternatives. And although open-source operating systems are well and truly established, with one-fifth of developers using one now, their growth seems to have flattened. Only about 17% of embedded systems developers would consider Linux who aren't already using it. What's certain is that the variety of embedded systems—and the operating systems that serve them—will continue to flourish.

***Jim Turley** is editor in chief of Embedded Systems Design magazine. You can contact him at jturley@cmp.com.*

Survey details

As in previous years, our annual survey looks at a cross-section of developers throughout North America. (European and Asian developers are surveyed later in the year by our sister publications in those regions.) In this case, "developers" means programmers, hardware engineers, managers, program directors, and hardware/software switch-hitters.

Several thousand developers are invited to participate in the survey. Their names are chosen randomly (called an nth-name select) from three sources: subscribers to this magazine, attendees of the Embedded Systems Conferences, and readers of EE Times who are working on embedded projects. In all, about one-third of our survey takers come from each of these three sources. This year over 1,200 people completed the entire survey; we discard partial results.

Because the survey is fielded electronically instead of being printed, we can be smart about what questions we ask. The survey is therefore a bit different for each respondent. For example, if a survey taker says she isn't using an operating system we'll skip most of the downstream OS questions and instead move on to development tools. FPGA users get asked detailed FPGA-related questions, and so on.

We'll cover precise demographics in a future issue but for now it's worth pointing out that our survey respondents are about evenly split among software, hardware, and system-level job functions. Specifically, they're not all programmers. Nor are they from one particular industry. The largest business segments (industrial controls, test and measurement, medical equipment) cumulatively made up only a third of the total base, so the survey is quite wide-ranging. Age, company size, and job experience were also evenly spread out. If you were one of our survey takers this year, thanks from all of us. If you're not, enjoy the results of this year's research.

The complete detailed survey results are available from CMP Media. For information, contact David Blaza at dblaza@cmp.com, or order online at www.embedded.com.

Back to top

Reader Response

Why do you separate out ThreadX by distribution? If needed, ask a follow-on question about the vendor. I would assume that GreenHills sells more copies of ThreadX because they have a good compiler and provide probe support. WindRiver is now re-selling ThreadX. The real-time Linux variants can be separated because under the hood they have different technical features. You should also list other open source RTOS like RTEMS which has been around a while and I'm sure has some "market-share". By the way thanks for the interesting article.

- Mark Taube
Tucson, AZ

Figures 2 and 3 are confusing. 2005 should come before 2006.

- Craig Treece
Fort Worth, TX

We used uC/OS-II in a previous project -- but not for the current & next projects.

I still like the kernel, but Micrium is a different company than it used to be (when it was basically J. LaBrosse) -- now the customer support is weak and the prices are MUCH higher.

FreeRTOS, RTEMS, and eCOS are just a few of *many* kernels with free usage and available source code.

The fact that Micrium now sells (for a good buck) things like a GUI, USB shim, Filesystem, etc... indicate that the market for their kernel is softening, similar to the survey results.

- Dejan Schmidt
California

Accelerate Your Design Process 10X
with Graphical System Design
Design
Prototype
Deploy