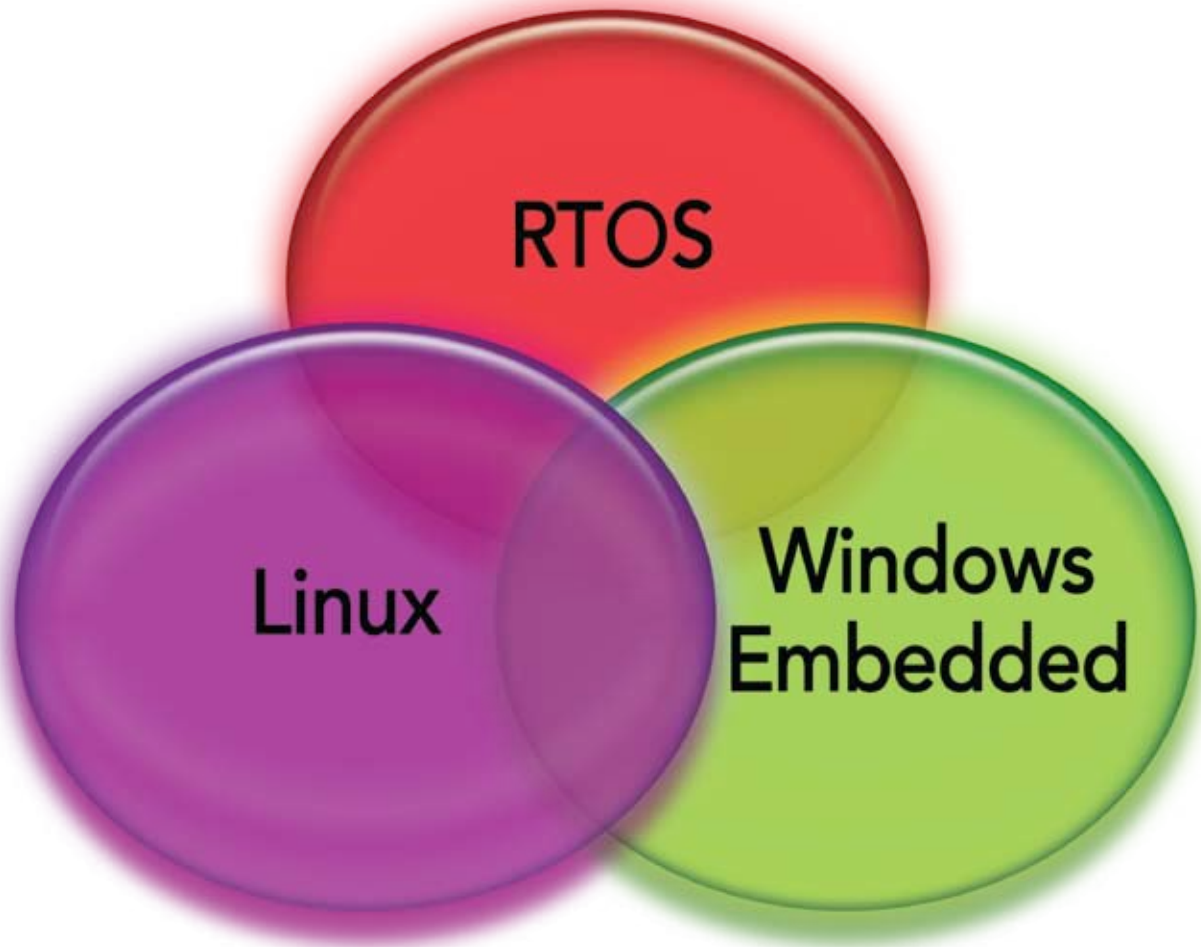


Operating Systems: What's really going on?

By Jerry Gipper



The choices in commercial Operating Systems (OSs) with real-time capability keep getting better every year. They are easier to use and typically well supported with Eclipse-based and open source tools. Several suppliers have filled in the middleware gaps with application-specific stacks that speed platform development. This article looks at some of the trends making the most impact on these OSs.

Embedded computing system software developers have three alternatives when deciding which class of OS they are going to use. There is a fourth, roll-your-own, but that's a quickly dying breed. The choices are: a number of different Real-Time Operating Systems (RTOSs), Windows Embedded, or Linux. Within each of these are many options.

RTOS

The RTOS world is highly fragmented with some minor consolidation occurring over the years. New RTOS solutions are not emerging like they were in the '80s and '90s. The market has reached a level of stability and maturity as commercially available products have become acceptable to designers.

RTOS vendors have always been challenged by the complex matrix of processors and host development platforms that needed support. They are constantly bombarded with new processor families and variations. Cross this with a healthy list of host development platforms, and support for all these options quickly

becomes a considerable task. Until the past few years, this was a necessary evil of the business – that is, until Eclipse came along. Offering an open development platform comprised of extensible frameworks solved the host development platform problem. Develop with Eclipse, and the matrix is reduced by one dimension.

“The Eclipse Framework is a big win for embedded OEMs, enabling them to access best-of-breed tools from multiple vendors without locking themselves into a particular supplier’s proprietary framework,” says Mike Christofferson, director of product marketing at Enea. Christofferson points out that Eclipse is also a big win for RTOS companies that want to offer their customers a complementary integrated development environment and other point tools without having to provide the entire framework and plunge headlong into the tools business. Today, every major RTOS supplier has Eclipse-enabled tools.

The dominance of multicore processors has hit certain RTOS suppliers harder than others. Some, like QNX, have had a multiprocessor strategy for years, making latest-generation processor support somewhat easier. Others that lacked a solid multiprocessor or multicore strategy resolved the issue by throwing multiple kernels at the problem. Today, that isn’t always an acceptable solution, and it is forcing suppliers to rearchitect their product lines.

Knowing that Linux can no longer be ignored, nearly all RTOS suppliers have developed a Linux strategy. Wind River even appears to be transforming from an RTOS supplier into a Linux supplier. The company recently restructured into four product groups with Linux as one of the four. “The new organization allows us to focus on new technology and market opportunities, be nimble and agile with customers and partners, and drive and measure return on investment,” comments Wind River CEO Ken Klein.

Several of the RTOS suppliers are undergoing major management changes. Enea, Wind River, and LynuxWorks have all made significant management adjustments as they try to adapt to the new business models being driven by the open source movement.

The RTOS community has never done a great job of rallying around any particular set of standards. With the exception of the work done through Eclipse, nothing else has ever brought the community together. Today, it is still a fragmented segment with dozens of choices for the embedded software developer. Each RTOS supplier maintains its own ecosystem and is a member of practically every processor ecosystem in existence.

Windows Embedded

I attended COMDEX in Las Vegas during the mid-’90s when the PowerPC processor was all the rage. The Apple, IBM, and Motorola (AIM) alliance was everywhere. I met with Microsoft representatives as they sought to understand embedded computing. At that time, Windows NT was the only choice, and it just didn’t cut it for embedded applications. Motorola and Microsoft were trying to work out a strategy for Windows in the embedded markets.

Microsoft has made great progress since it first started becoming familiar with the term *embedded*. The company now offers a line of Windows Embedded products, all the way from the low end with the .NET Micro Framework to the high end with the XPE OS, as well as a license-restricted Vista OS. The term *Windows Embedded* refers to the Microsoft family of embedded OSs for smart, connected, and small- or reduced-footprint devices.

Microsoft’s embedded OSs

- › .NET Micro Framework
- › Windows Embedded CE
- › Windows XP Embedded
- › Windows Embedded for Point Of Service
- › Windows Vista for Embedded Systems

For Microsoft, the emphasis is on connected experiences, so the company hones in on what the software must do to power the next generation of thin clients, ATM machines, or Point-Of-Service (POS) terminals. “We’re not just thinking about the OS that needs to sit in that device,” states Kevin Dallas, general manager of Microsoft’s Windows Embedded business unit. “We’re thinking about what the experience needs to look like in terms of manageability on a thin client solution, and then what we, as a company, need to provide in terms of the platform to enable our customers to deliver on that proposition.” Managing connectivity is a key advantage of using one of Microsoft’s embedded OSs. The consumer markets are definitely where most of Microsoft’s focus is.

Dallas thinks that Windows Embedded’s success is due in large part to Microsoft’s partner ecosystem. For instance, he says Microsoft works closely with silicon vendors to understand what the company needs to do in its OSs, tools, and services so that it can adapt to new multicore/many-core processors. Microsoft’s community of roughly 1,500 partners consists of silicon vendors, system integrators, and software and hardware vendors.

Dallas also admits that Microsoft needs to be more transparent with its product roadmap, providing one that expands 24 months out so that customers and partners can really plan ahead.

Linux

Open source is a dominant part of any software discussion today and certainly follows suit when talking about OSs. Linux, the infamous open source OS, gets mentioned by everyone, whether they are System-on-Chip (SoC), board, or OS suppliers. 2008 may be the year that Linux gains the recognition it has been seeking in the world of embedded computing.

Sample Linux consortia

- › Consumer Electronics Linux Forum (CELF)
- › Linux Foundation, formerly Open Source Development Labs (OSDL)
- › Linux Phone Standards (LiPS) Forum
- › Open Source Automation Development Lab (OSADL)
- › LiMo Foundation
- › Kernel.org

Developers within the Linux community love to demonstrate that no application is beyond Linux; you can probably find at least a lab demo in almost anything that needs an OS. It's particularly popular in smart handsets and high-end digital television, and is the OS of choice with carrier grade AdvancedTCA platforms. Linux has definitely moved out of the lab and into production in many products destined for these application segments. Linux is being used primarily on 32- and 64-bit processors with memory management but is probably catching as much as 50 percent of new design wins, remarks Bill Weinberg, LinuxPundit.com and LiPS Forum general manager for business development.

The Linux community is more aligned than the RTOS world. Several consortia dedicate themselves to advancing Linux in various market segments that touch on embedded computing.

Consortia efforts are at different stages for each market segment. The consumer space is probably the most mature, thanks to CELF. Jim Ready, CTO of MontaVista Software Inc., asserts, "CELF may be the granddaddy of Linux organizations. The fruits of its efforts get used. CELF is not contentious, solving problems as they are identified and then moving along." Ready notes that the carrier grade standardization worked quite well under the guidance of OSDL.

Standardization efforts aimed at mobile devices are at the early stages. The handset supplier market is more fragmented; thus, it's a bigger endeavor than the Carrier Grade Linux effort. Ready

thinks there is minimal pressure to comply with standards in many market segments, except for the carrier grade segment, which has a huge amount of middleware and requires a uniform environment.

Keeping up with Linux

The Linux release process was originally set up under a number scheme that assigned even numbers to stable production releases (2.4, 2.6) and odd numbers to unstable releases (2.3, 2.5). All of that changed with release 2.6. Production releases occur about once per quarter with no numbering convention to indicate if they are production releases. Weinberg explains these releases usually end up being baselines for commercial releases, but the commercial suppliers don't always keep pace. "OEMs don't want to deal with Linux in the wild," he says.

Stabilization and long-term ownership pose a challenge. Often there is no choice but to go to the latest release because that release may be the only one supporting a hardware choice an OEM previously made.

Ready says release management consumes a great deal of time and energy and changes the release rhythm. OEMs face stabilization challenges, higher rate of change, and a less stable code base. He claims, "Linux is the software engineer's dream. You always get to work on new stuff, never have to fix anything, and are 90 percent complete forever." His point reinforces the fact that the distribution maker is even more critical to the Linux release's stability under the new release structure.

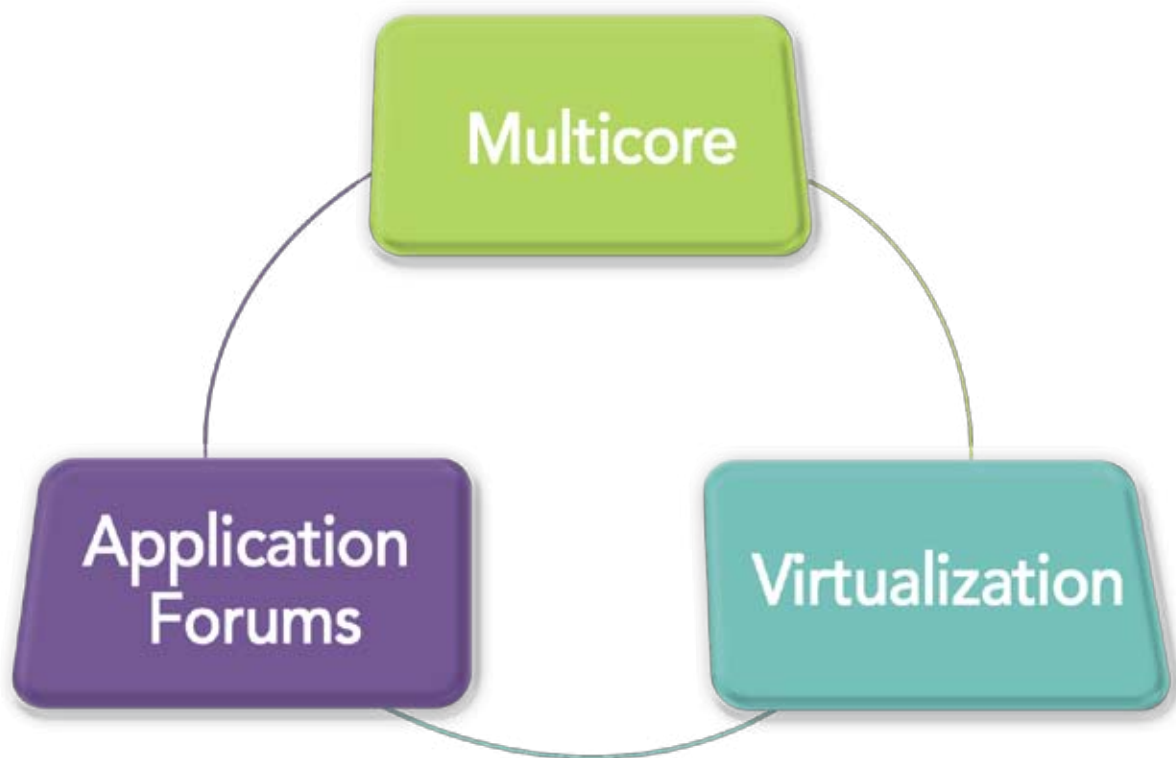


Figure 1

“More frequent production releases from Kernel.org are both good and bad,” adds Weinberg. “In general, everyone keeps up. You don’t have the massive forking potential under the old system.” Linux suppliers are finding that the support model is the better route to revenue than a Linux product model. In many cases, they are becoming the OS team for platform OEMs.

Technology advancement

For the most part, innovation in the OSs themselves has reached a plateau. The basic tenets of kernels’ functions are not changing. On the tools side, the Eclipse model of open source development has taken away the compatibility issues that plagued development tools for years. It’s safe to say that this battle is won, and everyone is working hard to make the software developer’s life easier.

Most OS development efforts are spent in one of three areas: support for multicore processors, developing application-specific platforms in the form of middleware, and implementing virtualization for a variety of reasons (see Figure 1).

Multicore and many-core

The trend of moving from single-core devices with single or multithreaded applications to multicore and many-core devices with synchronization across cores presents a problem. Moving out farther to distributed cores or network devices with massively distributed, networked, multithreaded applications is a difficult challenge for developers today. “If you were to ask OS suppliers, ‘Yes or no. Would you be happier if silicon providers had been able to just crank up the clock frequency?’ you would have gotten a thousand yes’s,” comments Rob McCammon, VP of product management at Open Kernel Labs.

“Developing applications for multicore architectures adds significant challenge and complexity to development and design teams,” says Kuljeet Kalkat, VP of product management at Wind River. “The key question developers ask is how to migrate, debug, and optimize their system for multicore.” Now that many new processor architectures are multicore or many-core, platform developers can no longer ignore this issue. How do you align I/O to the cores? How do you break up the code to run on the proper core? These and many other questions must be answered for each new project.

Multicore processors are not new to the industry. Some OS suppliers have been addressing this issue for years by supporting Symmetrical Multiprocessing (SMP) solutions they developed, while others have simply thrown more kernels at the problem. UNIX-based solutions like Linux have an advantage because of the SMP support built into UNIX OSs.

With today’s focus on multicore processors, it’s time to step up and solve the problems in the right ways. Research like the visual programming language studies underway at the Microsoft Research Center may make a difference. The winners will be those that bring out the best performance in multicore products and have tools that make processor programming manageable.

Application platforms

Most OS suppliers are seeking to differentiate themselves by adding support for selected vertical application segments to their design kits. This support is typically in the form of middleware that builds the software stack for these applications, making a more complete solution, speeding time to market, and reducing risk for the platform developer. Examples include aerospace and defense, automotive, consumer, industrial, mobile handsets, and network equipment.

Weinberg points out that because the value line has moved into stacks and tools, suppliers can’t just sell kernels and tools. Microsoft takes it a step further, “We realize that it’s not enough to build these components into our platform,” Dallas adds. “It really comes down to giving the developers the power through this development environment to actually create these connected experiences.” Expect to see more application platforms such as the mobile device application platform pictured in Figure 2 emerge from suppliers in the future.

Virtualization

Virtualization is a technique for hiding the physical characteristics of computing resources from the way in which other systems, applications, or end users interact with those resources. Virtualization can solve a whole range problems at once, such as:

- Supporting legacy OSs
- Running multiple OSs on a single platform
- Multicore or many-core resource management
- I/O resource partitioning within a platform
- Licensing protection
- Confinement of security compromises or trusted computing

Embedded computing platforms used in mobile handsets and many popular consumer devices are continually becoming more complex and able to handle more functions. New rich applications are combining a variety of functions that must be well integrated and becoming increasingly more connected to each other via the Internet. Of course, security is of high importance as users are sending more personal data through and between devices. Emerging government regulations will mandate how to protect this data. All of this is driving the need to better grasp virtualization.

The mobile handset and consumer markets are certainly ripe for virtualization technology. The size and potential for growth is phenomenal. Smarter connected devices are emerging every day. Our society has enthusiastically embraced electronic devices and expects them to do even more.

Competition heating up

The time to use a commercial OS has never been better. The tools are more robust and user friendly, the breadth of hardware support is immense, and the application support is expanding. As applications continue to grow in complexity, the OS suppliers are racing each other to supply the best solutions. **ECD**