

The Embedded Chasm

Unlike many of my colleagues at the OSDL, and also unlike much of the Open Source community, I came to Linux and Open Source not through the enterprise, but from embedded systems. From this perspective, I find it interesting, even perplexing, that vendors, analysts, community voices and other pundits repeatedly and enthusiastically proffer embedded applications as the next great frontier for Linux, and indeed the next great business opportunity for Open Source. At the same time, software suppliers, analysts and other vocal interests in embedded loudly hem and haw and wring their hands about whether Linux will EVER be appropriate for embedded design and deployment. Note to both camps: Stop. Take a deep breath. Time for a reality check and alignment of preconceived notions and ambitions.

Enterprise Linux adoption is proceeding apace. According to IDC, server deployment is running at 20 per cent and is slated to grow to at least 28 per cent by 2008. Packaged Linux software is growing fastest of all Linux-related revenue at 44 per cent CAGR and is expected to top US\$14 billion by 2008. Desktop adoption, while much less impressive, in 2004 surpassed the Macintosh and is growing annually at double-digit rates. While total these revenues for enterprise Linux-related software are respectable, and while annual commercial deployment volumes are now topping 2 million servers and over 7

“Yes, but my product addresses the unique needs of embedded developers”.

Truth be told, today’s embedded applications are not comparable to those of even five years ago. Today’s intelligent devices are highly connected, with dominant software content, running stacks that compare to enterprise and desktop systems in their complexity. Classic mono-function “toaster” designs are on the wane, being replaced by little servers and tiny desktops whose compute power and resources mirror the profile of application development hosts.

Before you all run off to invest in new ventures selling software or services to OEMs to build next-generation widgets with Linux, take note of the following metrics. While Linux is now a dominant force in embedded designs, realize that the entire embedded software market is sized between US\$700 and US\$800



Bill Weinberg

The apparent short-term potential for embedded deployment of Linux dwarfs PC/server volumes by orders of magnitude

million desktops, the apparent short-term potential for embedded deployment of Linux dwarfs PC/server volumes by orders of magnitude..

The most tantalizing examples come from consumer electronics. In 2005, Motorola, NEC, Matsushita, Samsung, and others will develop and market over a dozen Linux-based digital handset designs (smart phones) and together with cellular network operators could deploy five or ten million units or more. Companies like D-Link and the Linksys division of Cisco and others will ship millions of SOHO Linux-based wireless routers, attached storage devices and wireless media streamers. Philips, Sharp, Sony, Matsushita, Samsung, and of course TIVO will supply cable operators and consumers with millions of set-top boxes, DVRs and hi-resolution television built on Linux. Automakers, automotive systems OEMs and in-car media equipment suppliers like BMW, Delco, Pioneer, Siemens, Volvo, will deploy further millions of intelligent dashboard displays, navigation/telematics, in-car networking, vehicle maintenance, and audio/video applications in car, trucks, and public transport vehicles over the next decade.

Now expand this paradigm to include telecommunication and networking equipment, transportation, aerospace and defense, and myriad other ubiquitous applications. Pretty exciting, huh?

The embedded marketplace is divided into two camps: those that acknowledge, even embrace Linux, and those who continue to rally around their diminishing proprietary base. Even pessimistic analysts project Linux to garner 25-30 per cent of new design wins for 32 and 64 bit systems; optimists aver even higher shares, reaching well beyond 40 per cent by the end of 2005. Only embedded Windows today rivals Linux for new applications.

The legacy RTOS (real-time operating system) crowd offers a litany of “yes, but...” responses to these trends. “Yes, but Linux is too big for embedded applications”. “Yes, but Linux lacks the responsiveness of a real-time kernel”.

million for 2004. Moreover, embedded Linux OS and tools revenue accounts for about one tenth of that market. That means that commercial embedded Linux revenues amount to two orders of magnitude less than the enterprise Linux analogue.

So where is the money in embedded Linux? It’s where many think it should be - in the pockets of device OEMs and their shareholders, who save by side-stepping traditional suppliers with per-unit business models. And their savings on bills-of-materials (BOMs) means that more and more powerful devices are being delivered to market for less and less Dollars, Yen, Pounds and Euros.

What attracts device OEMs to Linux are (1) lower deployment costs (2) vendor independence and (3) faster time to market and to volume from readily-available Open Source software platforms and application components (among other factors that include performance, robustness, security, etc.). Both existing embedded Linux suppliers and device software wanna-bees were best to remember these factors before seeking revenues from run-time unit-scaled fees (a.k.a. royalties). Device OEMs spent two decades fighting high BOM costs and vendor lock-in, as well as making huge investments in zero or low value-added platform software. With Open Source and Linux in their toolboxes and in their products, why would they tolerate proprietary-style “business-as-usual”?

The simplest truth escapes both legacy embedded software suppliers and many new entrants. OEMs buy legacy RTOS software the same way they buy resistors, capacitors, displays and memories - as commodities. Device-makers commit to Linux and Open Source, instead, as strategic platform choices. Through the OSDL, I meet CTOs and officers of Linux and Open Source strategy all the time, but in my entire career I have never met a single Officer of RTOS strategy.