



The New Era of Mobile Linux Ubiquity

Bill Weinberg

Analyst/Strategist,
Open Source Development Lab (OSDL)

INTRODUCTION

In the last five years, Linux has progressed from marginal status as a platform for intelligent devices to a position of ubiquity, even dominance, in embedded design. The progress of the open source operating system (OS) spans the gamut of embedded applications, from communications to consumer electronics to automotive applications to industrial control and even to aerospace and defense. All types of devices today benefit from embedded Linux deployment, but those leading the charge are “service delivery vehicles” -- devices that connect to networks to supply voice, data, navigation, and entertainment content.

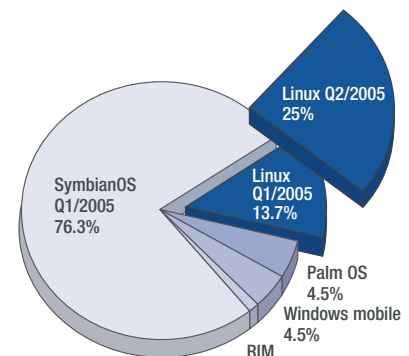
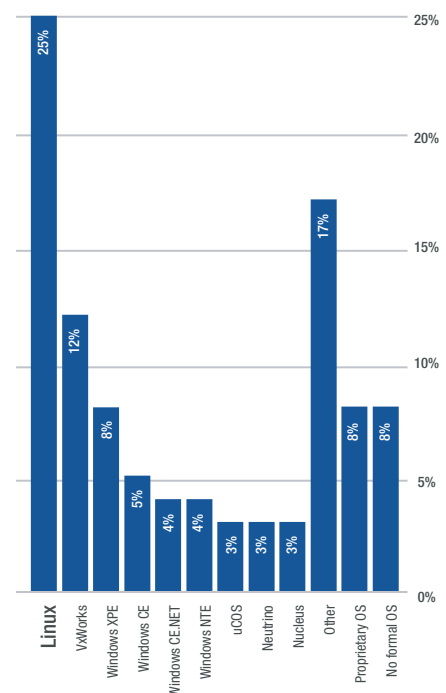
Adoption Trends

Analyst firm Venture Development Corporation (VDC) reports that in 2005, Linux-based OS software garnered 25% of new 32- and 64-bit design wins, with 29% of developers planning to use Linux in their next project. This position places Linux well ahead of traditional real-time / embedded OS platforms and of Microsoft embedded offerings. Within the embedded Linux design space, VDC also cites the two leading design domains to be consumer electronics (35%) and communications (30%).

Mobile wireless represents the intersection between these two leading segments, and Linux is making impressive gains as the platform OS for next-generation mobile telephony, especially high-end “smartphones” and feature-rich mid-tier handsets.

Linux design and deployment growth in the smartphone space has been dramatic. Gartner reports that in the fast growing smartphone segment (85%/year), Linux started out 2005 in good stead at 13.7%, and by Q2 enjoyed an even stronger 25% share – far ahead of Windows Mobile, Palm OS® or RIM (but behind Symbian OS).

Volume-wise, Canalsys reports Linux smartphone shipments growing in excess of 800% year-over-year (as of Q3 2005). Mid-tier Linux phones are also making inroads into Japan’s giant NTT DoCoMo network, with Panasonic and NEC shipping as many as 5 million mid-tier handsets, also known as feature phones. Panasonic’s December 2005 announcement of its intention to focus on high-end phones based on Linux bodes well for the platform in Japan, where Linux phone share could climb as high as 15% by 2007, according to some industry observers.



WHY LINUX FOR MOBILE APPLICATIONS?

When embedded developers first began developing with Linux half a decade ago, their motives were obscure, even questionable. Linux footprint was an order of magnitude larger than legacy RTOSes; its response times were orders of magnitude slower. The platform offered only rudimentary tools, and there were few if any suppliers to support its use. Indeed, Linux often found its way into embedded not *because* of its attributes, but *in spite* of them.

Today, technology for building and deploying Linux-based systems is more aligned with the needs of embedded development, but it is also important to understand that embedded systems requirements have changed too. If the prototypical design of the past was stand-alone, resource-constrained and hardware-centric, today's applications deploying Linux are highly-connected, resource-rich, and dominated by software content and the value it adds. Driving this shift are ubiquitous IP networking, falling price per megabyte and soaring capacities of DRAM and flash memory, dominance of 32- and 64-bit CPUs, and the annual doubling of lines of application code in the average device. So, while Linux has evolved to embrace embedded requirements, those same requirements have "grown up" to meet Linux and enterprise OS characteristics at the same time.

Developer Motivations

Individual embedded developers and engineering managers evaluate Linux using the same parameters as they did traditional RTOS platforms, with a few twists. On the technical side, they look at Linux performance (response and throughput), broad hardware support (CPUs, peripherals and board support), and available tools and utilities. Add to these attributes the ability to customize freely and to leverage the vast pool of existing UNIX expertise, as well as resources for Linux literacy. Beyond these legacy values, developers are enticed by the lower costs of acquisition and deployment, and the new experience of community-based self-determination and independence from the vagaries of working with a single supplier.

Manufacturer Rationale

At a corporate level, OEMs certainly recognize the same benefits, but for them Linux and open source represent a "strategic" choice, not the tactical inclusion of an RTOS (more akin to choosing hardware components). OEMs, both globally-branded manufacturers and upstart suppliers in emerging markets, face a common set of product line challenges. They must develop, deploy and support a range of products to meet requirements and niches of the markets they serve. OEMs must offer entry-level and full-featured devices; to differentiate or improve margins, they also need high-end offerings. Diverse geographies, local languages and national standards further complicate matters. Also, OEM product lines can include inherited and acquired technologies. In some cases, each device OEM manufacturer employs unique processors, wireless and multimedia chipsets, and incompatible OSes, middleware and applications. Such heterogeneous product lines require hiring, training and maintaining separate engineering and support teams. In markets faced with slim margins, lack of commonality further strains profitability by denying mobile device OEMs the economies of scale they enjoy elsewhere.

To face these challenges, companies as diverse as IBM, Philips and Sony look to Linux and open source to consolidate heterogeneous hardware, software and engineering, marketing and manufacturing. The broad reach of Linux reduces engineering fragmentation and the downstream costs of divergent product lines and practices. Lowering acquisition costs and Bill-of-Materials (BOM) impact further helps OEM bottom-lines, especially in businesses like mobile/wireless where costs are only recovered over the lifetime of a service subscription.

OVERCOMING BARRIERS TO ADOPTION

Despite the overall upsurge in Linux adoption in intelligent devices, important barriers remain, especially in key vertical markets and applications. In the mobile/wireless space, companies like Motorola, NEC and Panasonic have leveraged their considerable domain expertise and demonstrated the viability of building high-end phones with Linux and other FOSS (Free and Open Source Software) components. However, these manufacturers as well as other companies just entering the space require a more optimized mobile platform and a more streamlined development process for using Linux in mobile handsets. They also want to deploy Linux on mid-tier and low-end devices that lack dedicated GPRS or EVDO interfaces and other expensive support chips.

To that end, industry consortia like OSDL Mobile Linux Initiative (MLI), Consumer Electronics Linux Forum (CELF) and the Linux Phone Standards Forum (LiPS) have formed to identify and bridge gaps in the Linux and FOSS ecosystem to accelerate adoption on mobile phones and other wireless clients. In particular, OSDL MLI, with members like Motorola, PalmSource, (a wholly-owned subsidiary of ACCESS Co., Ltd.), and Siemens, has enumerated the following focus areas:

Technical Challenges

While mobile phone design and manufacturing present myriad technological requirements, the following create the greatest drag upon rapid development and time-to-volume:

- Development Tools
- I/O and Networking
- Memory Management
- Multimedia
- Performance
- Power Management
- Security
- Storage

Economic, Logistical and Regulatory Barriers

Not all barriers are technical; many are operational. Some of the stickiest challenges arise not from how devices operate in consumers' hands, but how they integrate into carrier and mobile operator networks:

- Legal/Regulatory Concerns** Mobile operators present massive requirement sets to their suppliers that include strict conformance to wireless protocols and regulations regarding spectrum usage. With FOSS, there exists the fear that hackers and even end-users could change the radio frequency characteristics beyond regulatory or safety bounds.
- Platform Alignment** While choice is a good thing, too much choice leads to fragmentation. Today, silicon providers, Linux distribution houses, ISVs, and OEMs can offer, deploy and maintain divergent versions of the Linux kernel and stack, complicating integration and application development.
- Capability Mainstreaming** While hardware and software support exists to build and deploy Linux phones, those capabilities are not mainstream, in that the kernel versions, patches, device drivers and other software are not part of the main Linux source tree. Even minor "forks" raise costs and can limit interoperability.

TOWARD UBIQUITY

Given the barriers described above, what steps should the Linux and mobile industries take to make Linux phones easier to build and deploy, to market to carriers and operators, and to achieve ubiquity with end-users?

Platform Pre-Integration

Mobile OEMs and ODMs often face internal challenges for resources and the core competence needed to build next-generation devices. Most handset manufacturers have experience in integrating in-house and prior-generation COTS (Commercial Off-the-Shelf) phone software. However, increasing complexity and need for multi-domain expertise (telephony, TCP/IP and wireless data networking, messaging, multimedia, open source development of middleware and applications, etc.) can easily outstrip available resources and expertise. Outsourcing key components only exacerbates the integration challenge. Outsourcing integration itself often means handing off this critical task to engineers with even less domain expertise than exists in-house.

Pre-integration offers OEMs some obvious benefits:

- Faster and shorter product development cycle, leading to faster time-to-market
- Reduced total development cost
- Consolidation of vendor interface and reduced support burden
- Elimination of non-value-added activities, freeing developers for product differentiation
- Increased delivered product quality and higher customer satisfaction (carriers, operators and end-users)

Defining a Pre-Integrated Mobile Software Stack

Today it is common to speak of a software “stack” – the entire software content of the delivered final product. For OEMs whose main value-add comes from manufacturing or a distribution channel, the availability of components that constitute a complete software stack is key. Most device manufacturers, however, add significant value in terms of original design and functionality, and can end up investing R&D monies in “de-integrating” stacks or solutions that leave no room for that differentiation.

OEMs can feel like Goldilocks* : this stack is *too short* (just a kernel and hardware support). Such minimal, system-level platforms place the middleware, application and integration burden on the OEM. Another stack is *too tall* (it is “complete” but without room or means to differentiate or add value). While a working phone design, out of the box, may be ideal for an ODM whose value-add is only in manufacturing, branded phone OEMs and their customers need to tailor applications and middleware to meet the particulars of branding, service delivery and network interoperability. In the real world, *one size does not fit all*.

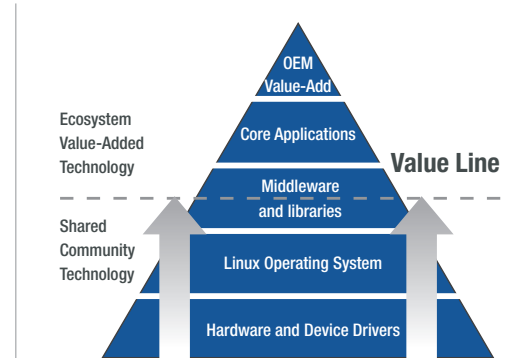
To discover what constitutes the “just right” stack, consider the following continuum:

Stack Description	Completeness	OEM/ODM Value-Add	Challenges
Finished product, off-the-shelf	90-100%	Brand, Manufacturing	Minimal differentiation
Shrink-wrapped “solution” stack	80%	Look & Feel, Management Interface	Branding offset by identical functionality; components commoditized
OS platform, development tools and middleware	60%	Application Stack, Management Interface	Device OEM invests to add value / differentiate
Hardware and OS platform support	40%	Application Stack, Management Interface, Middleware and Drivers	Maximum opportunity to add value --significant engineering required
Bare hardware and ROM monitor only	10%	Whole stack, including OS, Middleware and Applications	Large development and code management

* Cf. *Goldilocks and the Three Bears*, an English-language children’s story in which a small blond-haired girl encounters tough choices in the house of well-provisioned bears.

“Just right” is determined by OEM technology and “buy vs. build” traditions. In a more general sense, a critique of a mobile phone stack or solution will take into account investments by both the stack supplier and the OEM, especially how they add value to base hardware and open source community code.

At OSDL, we often debate the dynamics of the “Value Line” – the level at which technology and value supplied by open source and shared community resources tapers off and opportunities for third parties to add value begin. In the enterprise, the Value Line hovers above the LAMP application platform – Linux + Apache + MySQL + Perl/PHP/Python. For embedded applications, although conventional wisdom places the line at more or less the same level (in enabling middleware), the division is less clear.



Tall Stack

In theory, a complete solution stack would offer OEMs rapid time to market. In today’s market, Microsoft and other suppliers offer platforms positioned as “complete solution stacks” – aggregations of OS, drivers, middleware and domain-specific application code that purport to include 90-95% of the needed software. Often, however, these solutions leave little or no room for product differentiation and branding. A Microsoft Windows Mobile phone is first and foremost a Microsoft device; hardware and other OEM value-add is commoditized, as with a PC. Also, these Tall Stacks are notoriously monolithic and carry heavy royalties, giving mobile OEMs little freedom to optimize choice of components and BOM impact. The ideal Tall Stack, when and if it appears in the market, would provide comprehensive functionality and a high degree of customization capability.

Short Stack

Most Linux suppliers’ products are horizontal platforms with minimal support for vertical applications, especially for mobile phones. Some mobile Linux solutions include kernel-based capabilities for power management, real-time, scalability, etc., that are genuinely needed for handheld telephony, but which present OEMs with a conundrum:

- if those capabilities are mainstream, derived from community source trees, then platform vendor value-add is limited to necessary hardware support, low-level integration and QA;
- if such capabilities provide unique value beyond FOSS, then they are not mainstream, and carry perils of forking and single-vendor dependence.

In either case, system software technologies lie at or below the Value Line, putting OEMs at best in a position of paying a premium for a commodity and at worst paying an open source infrastructure “tax.”

In any case, Short Stacks, even if rich with device drivers and other hardware support, leave the bulk of application-specific engineering to the OEM.

Just Right Stack “Just right” represents a trade-off between completeness and flexibility, between time-to-market and room to add visible value. In the above paradigm, “just right” falls into the 60-80% solution range, can include a mix of open and proprietary, commercial and free components, and should include a solid version of the Linux kernel, a rich set of device drivers, choice of enabling middleware (not just Java), and a range of applications for out-of-the-box and also post-market deployment. It should leave room for OEM branding, customization and differentiation, but in many cases can also carry its own cachet or brand equity as with ACCESS PalmSource or Trolltech.

It is important to note that no stack – Short, Tall or Just Right – stands alone. Completing the mobile solution is accomplished not just by internal OEM resources, but through engaging with the emerging ecosystem of mobile software suppliers and FOSS projects that address mobile telephony.

Closing the Gaps

While much deployment growth to date has been organic, driven by both OEM and community efforts, the mobile Linux platform demands more investment to achieve ubiquity. Specifically:

- Greater ease of development: tools, software components, libraries, drivers, codecs, etc.
- Improved processes: integration, testing and documentation
- Standardized platform: scalable, secure and standards-conformant for interoperability
- Customizable: room to add value / differentiate, especially in HMI (Human-Machine Interfacing)
- Domain-specificity: support for call stacks on the right platforms for single and multi-core phones

Vendors like ACCESS/PalmSource, Mizi, MontaVista Software, Trolltech, Wind River, and others are investing their own resources to close these gaps, complemented by a range of community resources and initiatives.

In October 2005, OSDL launched its fourth and latest working group, the Mobile Linux Initiative (MLI). MLI includes members from all levels of the mobile telephony ecosystem – chipset manufacturers, Linux distribution and platform suppliers, middleware ISVs, handset manufacturers, integrators, carriers and operators.

OSDL Mobile Linux Initiative Mission

To accelerate Linux adoption in the mobile space:

- Identify and address technical and non-technical industry requirements
- Create and foster implementations in open source
- Advocate/explain industry needs to the kernel/open source community
- Promote mobile Linux (including education of mobile operators about benefits of open source)
- Clarify legal and regulatory issues surrounding mobile phones as they relate to Linux and open source
- Enable and foster pre-platform developer ecosystem

MLI intends to focus on solution creation, not just publish APIs and posit standards that can end up as unfunded mandates. To that end, MLI members are today marshalling resources to create unique implementations to meet handset OEM, carrier and operator needs, to foster the advancement of existing open source projects, and to open existing internal technologies for the benefit of the MLI audience and the community in general.

To learn more about OSDL and MLI, visit www.osdl.org.

Community Resources

A myriad of projects and resources exist in open source to assist OEMs, their customers and their suppliers in building and deploying Linux-based phones:

- The Linux kernel tree - www.kernel.org
- Linux mailing list - www.kml.org
- ARM Limited tree and Support Zone
www.arm.com
- Architecture trees for Intel XScale, TI OMAP, and Freescale ARM CPUs and Code Sorcery
- Embedded Linux books by Karim Yaghmour, Craig Hollabough and Christopher Hallinan
- Flash file systems projects CramFS, JFFS2, YAFFS
- Linux Phone Standard Forum - www.lipsforum.org
- CE Linux Forum - www.celinuxforum.org
- Compilers and other tools available from Red Hat
- Dozens of other projects at www.sourceforge.net
- Linux phone pages on www.linuxdevices.com

CONCLUSION

The goals of this paper have been to illustrate the rapid growth of Linux in mobile telephony, to provide insight into the trends and motives behind that growth, and to provide developers and others interested in mobile Linux with insights and means to meet the challenge of building and deploying Linux-based mobile handsets. This paper also aimed to offer guidance and justification to developers, their management, customers and partners in evaluating Linux platform solutions and other resources.

Phone manufacturers, mobile ISVs, carriers, operators and subscribers take note: Linux is coming to a phone near you, and coming soon. In your product lines, on your networks, or in your pocket, Linux-based mobile phones are changing the mobile-wireless industry and offering new opportunities for value-added. Pick up that phone – Tux is calling you.



ACCESS Systems Americas, Inc.

1188 East Arques Avenue
Sunnyvale, CA 94085
Phone: (408) 400-3000
Fax: (408) 400-1500

Copyright (c) 2006, ACCESS Co., Ltd, and PalmSource, Inc. Access is a trademark or registered trademark of ACCESS Co., Ltd., in Japan and other countries. PalmSource and certain other trade names, trademarks and logos are trademarks which may be registered in the United States, France, Germany, Japan, the United Kingdom, and other countries and are either owned by PalmSource, Inc. or its affiliates, or are licensed by PalmSource, Inc. from Palm Trademark Holding Company, LLC. These marks may not be used in connection with any product or service that does not belong to PalmSource, Inc. (except as expressly permitted by a license with PalmSource, Inc.), in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits PalmSource, Inc., its licensor, its subsidiaries or affiliates. All other brands and trademarks used herein are or may be trademarks of, and are used to identify other products or services of, their respective owners.