



The Business Case for Commercial Embedded Linux

WHEN IT MATTERS, IT RUNS ON WIND RIVER

INTRODUCTION

Linux has been a phenomenon in embedded device development. It is now the predominant platform for devices, and for good reason—it offers performance, features, and a thriving ecosystem to support advanced embedded devices. An obvious part of the appeal is the perception that Linux is free. It certainly is free to download and use, and doesn't have any royalty-bearing fees per se. But there are costs—plus time and risk—associated with Linux development. There is a solid business case for considering a commercially supported Linux during the build-versus-buy decision process for any embedded product.

TABLE OF CONTENTS

Introduction 2

Lower Total Cost of Ownership (TCO) 3

 Your Own Distribution (Whether You Like It or Not) 3

 Yocto Project to the Rescue 3

 The True Cost of “Free” Linux 3

 The Return on Investment from Commercial Embedded Linux 4

Commercial Intellectual Property Review and Disclosure 5

 Embedding Linux Is Redistribution 5

 Reducing Open Source Risk with License Management 5

High Quality Development and Build Tools 5

Full System Simulation 5

Global Support, Services, and Training 6

 The Community Has Short-Term Memory 6

Maintenance and Security Vulnerability Patches 6

An Ecosystem of Hardware and Software Partners 6

Embedded Development Transformation 7

The Return on Your Investment 7

Conclusion 7



LOWER TOTAL COST OF OWNERSHIP (TCO)

When considering Linux for a device, it's important to take the long-term view. Embedded devices are often in production and in service for many years, sometimes decades. In the long run, commercial offerings that provide a proven embedded Linux with support and maintenance are cheaper than maintaining a roll-your-own (RYO) Linux solution in-house.

Your Own Distribution (Whether You Like It or Not)

Newcomers to embedded Linux often don't understand that every RYO Linux platform done in-house is a unique and custom Linux distribution. Embedded Linux requires compiling and linking on a host system for download to a target host. The creation of all of the right puzzle pieces to make the OS run on your embedded target is effectively your own in-house distribution of Linux. For this reason, embedded Linux is different from desktop and server Linux, where one typically uses a well-known distribution already packaged and ready to go. Companies adopting Linux must understand that they are taking on the maintenance and support of a custom Linux distribution, which requires expertise and staff.

Yocto Project to the Rescue

The very nature of embedded Linux created the need for tools to build and assemble all of the components of a platform. In fact, Wind River® did so for its Linux distribution as well. But with no standardization, each vendor and user made its platform its own way. The Yocto Project (www.yoctoproject.org) was created to standardize the platform for embedded Linux, thereby greatly increasing interoperability and creating an ecosystem of hardware and software vendors around a common set of standards and tools. The ecosystem also provides support and expertise toward emerging standards such as those being created today for the Internet of Things. The Yocto Project is freely available and provides a standardized embedded Linux build system for production use.

Wind River is a founding member and has been a very active leading contributor to the Yocto Project, and continues to be a strong supporter of the initiative. Encouraging and building on standards makes embedded Linux stronger in the long run. Wind River Linux is based on the Yocto Project and has the fastest release cadence in the market.

Although the Yocto Project solves one of the key issues with RYO embedded Linux, it doesn't diminish the value of a commercially supported and maintained solution. It's important to consider the real costs of "free" Linux.

The True Cost of "Free" Linux

There are many hidden and unknown costs when adopting embedded Linux, in particular for first-time projects. Newcomers are daunted by the sheer size of the Linux code base and associated compiler, libraries, and platform build systems. Table 1 illustrates the code base and estimated development effort (figures taken from www.openhub.net).

Table 1. Code Base Size and Estimated Development Effort for Linux and Associated Projects

Component	Code Base (Lines of Code)	Estimated Effort (Person-Years)
Yocto Project	~283,500	~72
OpenEmbedded Core	~2,150,000	~625
YearBitBake	~72,000	~17
GNU Compiler Collection	~7,000,000	~2,100
Linux kernel	~17,400,000	~5,600
GNU C library	~1,210,000	~340
Totals	~28,000,000	~8,000

The fact that Linux leverages over 8,000 person-years of development is part of the appeal for embedded development. However, Linux is large, complicated, and mostly unknown to your development team. It takes time to learn how to use, build, and run Linux on an embedded platform. It also takes time and effort to continue to maintain Linux—and you must provide support, patches, and security vulnerability management for Linux and your application code. As stated above, each embedded Linux is a unique distribution and comes with the associated costs and risks.

It's difficult to predict the cost of adopting Linux for development teams, but based on our experience, the investment is considerable. The kind of investment required for a Linux-based operating system that would be suitable for a device used in telecommunications, for instance, is shown in Table 2:

Table 2. Estimated Development and Maintenance Costs of RYO Linux over Time

	Costs
Initial release (porting, testing, hardening, quality assurance)	\$600K
Yearly maintenance	\$200K
Yearly security vulnerabilities assessment and resolution	\$250K
Yearly export classification and open source licensing compliance	\$50K
Five (5) year total costs for RYO Linux	~\$3M

The following screenshot shows a telecommunications device example derived from the Wind River Linux TCO Calculator website. [Try the calculator today](#) to see how Wind River Linux saves customers up to 57% upon switching from rolling their own Linux.

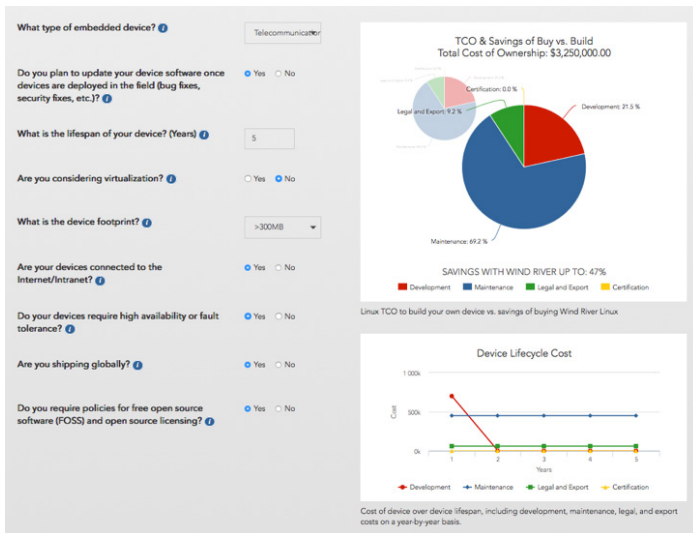


Figure 1. Sample calculation from the Wind River Linux TCO Calculator

This data illustrates that providing the equivalent of commercial quality and support for RYO Linux is very expensive. Note that support over time could actually become more expensive as community support wanes.

Linux may be free to download, but it is not free in the medium to long term—and embedded systems are long-term products; consideration of the costs in terms of product lifespans is important. Commercial Linux is a high-quality product that has gone through rigorous testing and intellectual property inspection and is maintained and supported for the long term.

The Return on Investment from Commercial Embedded Linux

The greatest benefits of commercial Linux come from the medium and long term. Depending on the development team’s level of expertise, the advantage of a commercially supported product may come sooner rather than later. However, let’s assume that the initial investment in time and money for Linux is zero or “free.” Looking at a theoretical investment curve (investment in dollars per three-month period) in Figure 2, we can see that RYO Linux quickly becomes more expensive and continues to grow over time.

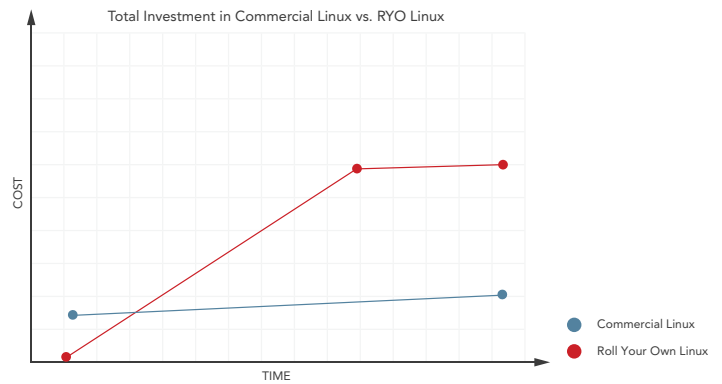


Figure 2. A representative graph of relative investment in commercial Linux versus RYO Linux

This graph is representative and does not use absolute figures. The commercial Linux investment is usually an upfront fee with an ongoing maintenance and support fee over time. RYO Linux gives the perception of no or very low costs upfront, but those costs quickly grow over time. The labor costs for RYO Linux accumulate significantly faster in the long term, even when factoring in diminishing resources committed to the platform.

Another way to look at this is to consider the return on investment: For every dollar spent on commercial Linux, how much does it pay back? Figure 3 shows that the initial investment is quickly recovered and that overall, there is a positive and significant return on investment for using commercial Linux in the medium term.

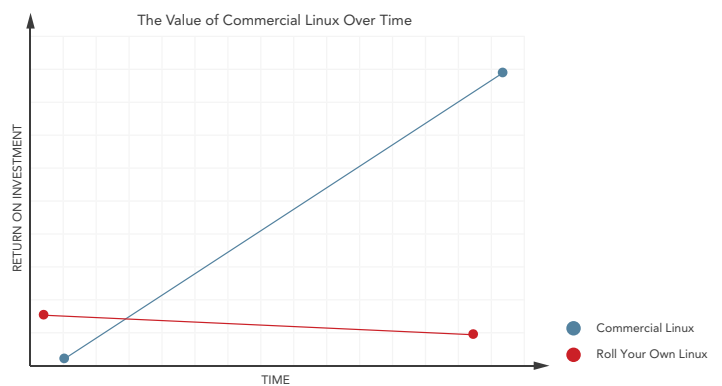


Figure 3. The value of commercial Linux versus RYO Linux (using the scenario from Figure 2, the value is the difference in cost of each approach over time; commercial Linux continues to provide value while the value of a RYO solution diminishes)



There are more than labor costs to consider with RYO Linux; it's also important to understand the implications of using Linux in embedded systems and the risks associated with open source licensing, IP and export compliance, security compliance, and so forth.

"The ability to shorten our development period by 15% was a remarkable result of using Wind River Linux. This led to a considerable reduction in development costs."

—Takahide Inoue, Manager,
New Business Product Development
Department, Personal Solutions Business
Development Group, Sharp Corporation

COMMERCIAL INTELLECTUAL PROPERTY REVIEW AND DISCLOSURE

Linux and its associated tools and libraries comprise a huge code base. Embedded device manufacturers must realize that although Linux is free to use, it is not without licensing implications. Not all open source licenses are the same, but many have redistribution clauses that place requirements on companies building devices. Unlike using Linux on a desktop or server, shipping a product with Linux is redistribution, which opens you up to more licensing requirements.

Embedding Linux Is Redistribution

Creating your device with a Linux run-time system as part of its software is equivalent to distribution under many of the open source licenses used in Linux, including the GNU Public License (GPL). There are on the order of 20 million lines of code for Linux and associated open source tools—a large code base with a multitude of licenses.

With (re)distribution comes the responsibility to make sure your company is complying with the license requirements. For example, providing free access to the source code for the open source portions of your product, including any tools that might ship with the product. Often there are clauses in the licenses about derivative works that can include kernel modules, modification of existing code in Linux, libraries or tools, statically linking to open

source libraries, and other stipulations. It's critical that embedded device manufacturers catalog the licenses of the software they are using, understand the level of risk associated with the license, and are prepared to fulfill the obligations associated with each of the licenses. Unfortunately, many companies don't treat this aspect of Linux seriously and open themselves up to needless liability.

Reducing Open Source Risk with License Management

Managing the licenses of all of the software you use, whether third-party commercial, open source, or even your own source code, is a critical function. Linux uses a huge code base, and it is a large task to manage and document the licenses plus any and all changes needed for your device. In addition, you often need to maintain up-to-date source repositories for download due to licensing terms. Commercial solutions make it possible to reduce risk, time, and cost. Commercially supported Linux delivers full licensing compliance and reporting. Commercial vendors are experienced with Linux licensing and can aid in the adoption of open source into your code base mix.

"50% of companies will face challenges due to lack of free open source software (FOSS) policy and management"

—Gartner, FOSS Analysis, November 2011

HIGH QUALITY DEVELOPMENT AND BUILD TOOLS

Commercial Linux is accompanied by value-add development tools that go well beyond what is available from (mainly command line) open source tools. In addition to the familiar coding, testing, and debugging capabilities, modern commercial integrated development environments provide tools for building the Linux and device image, system performance and behavioral analysis, memory profiling, and hardware-level debugging via JTAG. Having the right set of tools increases developer efficiency. Support and training on these tools to accelerate time-to-productivity are appealing benefits of commercial solutions.

FULL SYSTEM SIMULATION

A significant innovation in software tools has been full system simulation—the simulation of not just the embedded target processor, but also the system-on-chip (SoC) features, board-level hardware, and even interconnected targets in a complete system

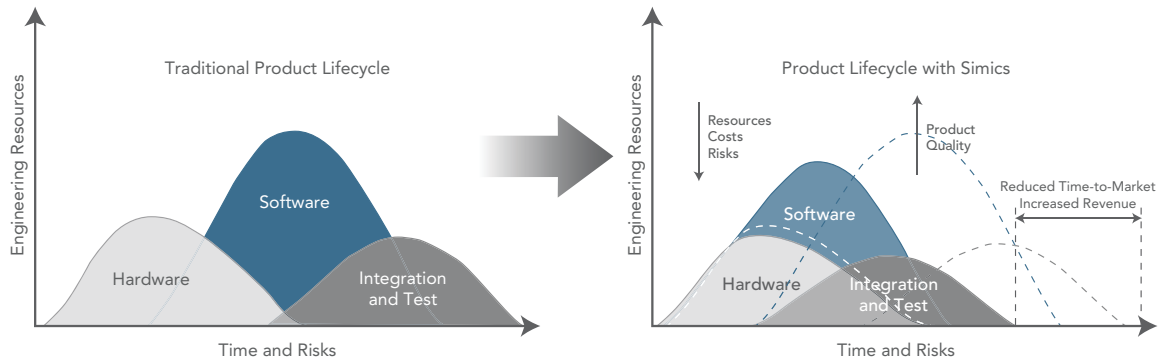


Figure 4. The transformed development cycle when using full system simulation with Wind River Simics

of subsystems. Developers, with their commercial development tools, can deploy and test on a simulated full system. This innovation greatly improves the support for iterative and agile development processes, but also removes the hardware supply bottleneck that plagues most embedded development projects. Wind River Simics® is an example of a full system simulation product.

The ability to acquire and apply innovative and highly productive development tools provides an advantage that RYO Linux development cannot duplicate. In fact, these tools have proven to provide measurable increases in productivity and quality in embedded projects in general. These benefits are additive to the lower total cost of ownership that other advantages of commercial embedded Linux bring to the table.

GLOBAL SUPPORT, SERVICES, AND TRAINING

A key aspect of commercial Linux is the ability of the vendor to support and provide services for their offering. Although open source community support is available and often helpful, if changes and fixes are needed the community is unlikely to do this work for you. A deep understanding of the Linux kernel, for example, may be needed in order to satisfy a performance requirement. Commercial Linux offers quick resolution with engineering services while avoiding entangling your project with GPL licensing issues. A significant portion of Linux patches and updates are security related, and since your embedded device is now its own unique Linux distribution, you are now responsible for maintaining it.

The Community Has Short-Term Memory

Community support for Linux abounds for current releases of the kernel and distribution releases. However, this support quickly wanes as versions become outdated from the most current releases. It's unrealistic to expect open source community support for an older

version of the Linux kernel, for example, that has been built into a product that has been on the market for several years. As the years go by, companies must rely on internal expertise to support their products. As these products mature, this expertise may mobilize to other projects, or the experts may depart the company altogether. Commercial Linux offers support that extends well beyond what an open source community can offer. Also, commercial Linux vendors can offer long-term support that goes well beyond the initial product support period. This assurance is critical in embedded devices due to the long product lifespans involved.

MAINTENANCE AND SECURITY VULNERABILITY PATCHES

Commercial Linux provides regular product updates and maintenance and security patches. Fixing, installing, and deploying these changes is a considerable effort for the embedded Linux vendor—an additional cost that RYO Linux brings to your project. Commercial vendors can stay on top of Linux kernel and security updates that can quickly impair the productivity of a development team. Also, commercial vendors provide more regular patch schedules and convenient delivery methods than RYO alternatives.

AN ECOSYSTEM OF HARDWARE AND SOFTWARE PARTNERS

Embedded OS and tools vendors form a cooperative network of partnerships and alliances. No single vendor can supply all of the needed software for a device, so partnerships are essential. Commercial embedded Linux provides the opportunity to leverage open source and proprietary offerings. The goal is to provide high-quality, off-the-shelf solutions to avoid your development from “reinventing the wheel” with each new product. Examples of partner products include specialized tools such as open source license management solutions, static analysis, and test automation.

Examples of run-time products include graphics engines, security whitelisting tools, enhanced connectivity solutions, and industrial protocol support. Even services are available such as security testing and evaluations and licensing audits. The leverage provided by these complementary products increases the return on investment for commercial embedded Linux by greatly enhancing the run-time platform, tools, and services while limiting the amount of time your project spends on the platform itself. Allow your team to concentrate on the differentiating features that produce revenue rather than the table-stakes capability of the run-time environment.

EMBEDDED DEVELOPMENT TRANSFORMATION

The embedded device marketplace is highly competitive, and manufacturers are under more pressure than ever before. Security, the Internet of Things, machine-to-machine networks, new technology introduction, and market regulations and certifications are causing more strain on development teams. Companies are not only surviving, but thriving in this environment. A transformation of the way companies develop embedded software is underway, not only in processes and technology but also at the business level. Successful companies are looking honestly at their build-versus-buy decisions and realizing the risk reduction and TCO benefits that commercial off-the-shelf solutions offer. See the Wind River whitepaper "[It's Transformation Time for Embedded Development](#)" for more details.

THE RETURN ON YOUR INVESTMENT

With all these points in favor of commercial Linux, what is the net benefit for embedded device development teams? The following list illustrates that the benefits go beyond costs:

- **Lower TCO:** Commercial embedded Linux has significantly lower costs over time than RYO Linux.
- **Risk reduction:** Using a commercial embedded Linux is less risky than a RYO solution. Technical, business, and legal risks are greatly reduced with commercial-level support, maintenance, and IP compliance and review.
- **Time-to-market:** Reduced development and integration time with ready-to-go platforms means shorter development times. When time-to-market is critical, schedules are at risk with RYO solutions.
- **Quality and security:** Commercial embedded Linux's quality assurance, monthly maintenance updates, and critical security vulnerability mitigation improves end product quality and security.

- **Building value:** By concentrating on new, differentiated features rather than building, maintaining, and debugging a RYO Linux platform, users of commercial embedded Linux build value.

"Wind River Linux has made development much easier and faster. We are able to save 20 to 30 percent in development time using Wind River Linux and tools. Along with our internal enhancements, our entire product life-cycle has significantly improved."

—Yossi Bar-Sheshet,
Director, Software, Corrigent

CONCLUSION

Commercial embedded Linux offers a clear return on investment versus RYO, in-house developed and maintained Linux. Not only is the TCO lower, but also the technical, business, and legal risk of commercial embedded Linux is much lower. The ability of the embedded Linux vendor to supply training, services, maintenance, and support greatly increases productivity and also reduces the overhead of maintaining your own, unique, embedded Linux distribution. Success in a competitive market means concentrating on what you do best, building great products—and avoiding the risk and expense of platform development that adds to neither your feature set nor your bottom line.

Choosing a trusted commercial Linux vendor will take care of all these needs. Wind River has been a pioneer in the embedded market for over three decades and delivers the market-leading embedded Linux solution. Try the "[Build vs. Buy Wind River Linux TCO Calculator](#)" today to see how Wind River Linux can save you up to 57% while reducing risk and freeing up the time necessary to focus on your core competencies.

Contact [Wind River](#) today or learn more about [Wind River Linux](#) to get on the fast track to cost-effective, low-risk embedded Linux development.

