Embedded Operating Environments

How Does Linux Compare?

An IDC White Paper

Analysts: Al Gillen and Dan Kusnetzky

Introduction

One of the technology industry's greatest — and most invisible — success stories is the embedded operating environment. Computers performing traditional client and server roles seem to garner the majority of the attention from consumers, business users, and the trade press. However, a vast army of devices exists that, like a traditional computer, run an operating system on top of a microprocessor and, in turn, support application packages on top of the operating systems.

Often, these environments are tightly integrated into a device that has a name other than a "computer," resulting in many casual observers not recognizing these units as a compute-centric device. These devices can manifest themselves in an infinite variety of form factors. Examples of embedded devices unknowingly used by most automobile drivers are the numerous automobile subsystems controlling everything from fuel injection and 4-wheel-drive transfer case units to climate control systems and the modern antilock brake and traction control systems.

However, it would be a mistake to believe that embedded operating systems are used in only automotive applications. Other popular uses include consumer electronics, electronic games, and home appliances. Embedded and handheld operating environments (EHOEs) also have emerged as a powerful tool for miniaturizing sophisticated electronic systems, such as the now ubiquitous handheld global positioning system receivers.

Embedded operating systems also are used in large numbers as building blocks to create a computing system infrastructure that supports network connectivity, routing, communications, printing, and even file serving. EHOEs also see wide deployment in aerospace subsystems, industrial controls, and in robotic devices. The list goes on and on.

Ironically, many individuals marvel at the size of the personal computer industry, into which over 100 million operating systems are shipped every year. By comparison, processors used for embedded applications constitute a market that is at least six times as large. Curiously, most of

IDC
Analyze the Future

5 Speen Street • Framingham, MA 01701 • Phone (508)872-8200 • Fax (508)935-4015

the key vendors in this market are anything but household names — unlike the flashy software and hardware vendors that have commanded the PC market.

Ironically, it is the emergence of sophisticated subPC devices, which today offer some of the basic functionality previously reserved for PCs and notebook systems, that has focused the spotlight on embedded operating systems. A byproduct of the explosion of these so-called smart handheld devices is the misconception that the EHOE market is a brand new sector, invented with the sole mission of powering these high-function, small form-factor gadgets.

All of this has happened to the pleasant surprise of vendors that have been selling EHOEs for the past 10 or 20 years. Over the past few years, this market sector also has become attractive to investors and venture capitalists. The challenge for existing players has been to extend their operating system services to address these new form factor requirements without disturbing other customer sets currently building dramatically different types of devices. At the same time, this shift in use requirements has opened the door for numerous upstarts new to the EHOE space.

Some Background on EHOEs

Definitions

IDC has standard definitions for products built using EHOEs that are listed in the appendix.

A Historical Perspective

Although a number of EHOE vendors can trace their roots back 20 years, much of the market opportunity has existed on devices that were developed using so-called “roll-your-own” operating environments. Since the 1970s, single-board microprocessor systems have been available along with programming tools having varying degrees of sophistication. Many device and industrial control manufacturers used these off-the-shelf, single-board systems or single-board designs of their own along with a rudimentary operating system that their developers created on an application-specific basis.

In some cases, the “operating system” was undistinguishable from the application code, while in other cases a clear distinction existed between the two layers. As the capabilities of the processors increase and as the sophistication of the final product grows more complex, it has become increasingly expensive and complicated for companies to continue to roll their own operating systems. As a result, the market has been making a gradual migration from roll-your-own operating systems to commercial operating environments.

An entire industry sprang up to offer these roll-your-own developers turnkey solutions that include an operating system, development tools, and, often, hardware reference platforms. These components promise shorter time to market for products being developed and provide support and ongoing enhancements to the operating systems stack. This allows developers to focus all their energy on the product and application code development. Furthermore, these operating system vendors take care of porting the operating system layer to each new hardware platform and processor architecture, further insulating the product developers from a never-ending development and maintenance cycle.

Today, several heavy hitters exist in this market, including Wind River Systems and Microsoft, and another dozen smaller companies that offer turn-key EHOEs. Collectively, these companies generated $666 million per year in new license and upgrade revenue during 2000, with the market expected to grow to $2.8 billion by 2005.

Acceleration of EHOE Requirements

The worldwide EHOE market has undergone a significant functional expansion over the past few years due to a boom in the use of EHOEs in portable devices, particularly smart handheld devices. Additionally, the creation of increasing numbers of landline-connected smart devices ranging from Internet screen phones to ATMs and gasoline dispenser machines have broadened the market in many directions.

This functional expansion also suggests that many of the new devices being invented today are transitional devices that will bridge between the fully functional desktop PC of today and the fully functional (and highly integrated) personal device of tomorrow.

This functional expansion has caused a couple of other effects, including:

- The technology shifts and brand-new device types have opened up new markets that, in some cases, have been served by new EHOE entrants.
- Traditional EHOE products have expanded the range of their capabilities by adding TCP/IP stacks, embedded browsers, HTTP serving capabilities, and more.

Simply stated, connectivity and Internet protocols are making their presence felt in the EHOE market.

This expansion in functional requirements also has, once again, upped the ante for roll-your-own developers and has made off-the-shelf solutions more appealing. While the cost of using an off-the-shelf operating system has been an impediment for some developers, the advent of low- or no-cost embedded operating environments — such as Linux — has the potential to erase that objection.

Even if the increasing frequency of product and EHOE revisions and the demand for shorter times to market successfully drives more developers to off-the-shelf EHOEs, the market will likely continue to be served by a variety of vendors. What also is clear is that bigness and huge financial resources don't automatically translate to market dominance in every market sector.

Take, for example, Palm OS. This technology managed to carve out a formidable niche in the industry in a functional role that Microsoft's Windows CE could have owned. But early versions of Windows CE were difficult to use, required more powerful and more expensive hardware, and carried relatively stiff licensing fees. A window of opportunity was left open for Palm — which the company exploited. As a result, subPC client devices powered by competitive EHOEs threaten the company's long-time formula for success, which calls for owning both the client and the server in a given network, then selling applications that bridge both environments.

Another important dynamic that is helping reshape the EHOE market is a fast-growing interest in Linux as an alternative platform to proprietary EHOEs. Through CY00, IDC tracked very few device shipments with embedded Linux inside them; however, IDC's expectation is that shipments of devices built using embedded Linux will ramp up significantly during the next couple of years. This expectation is based on observations of the Linux EHOE vendor community moving at an increasingly fast rate to introduce a wide variety of Linux distributions specifically intended for embedded use and a dramatic increase in interest by developers in embedded Linux. Likely, Linux use will increase dramatically through IDC's forecast period, which extends out to 2005, but as with vendors that sell Linux as a COE and SOE, turning large shipment volumes into revenue will remain a challenge.

Myths and Realities of GPL

Brief Overview of Open-Source Software and Licensing

Linux is one of a large (and growing) collection of software products offered under one of several licenses that provide access to source code. These licenses are collectively known as open-source software licenses, but they are not created equal.

While the Linux kernel (including general-purpose and embedded versions) is covered by the GNU General Public License (GPL), a full Linux installation will likely include utilities, libraries, shells and shell command tools, a browser, and other related software products — which collectively will be covered by multiple licenses. A developer working in an open-source environment will likely touch, modify, or merely configure software components covered by a variety of these licenses. These licenses may include some or many of the following.

GNU GPL

GNU GPL is perhaps the best known of the open-source licenses used by the Linux kernel, which defines usage terms that are contradictory to most, if not all, of today's proprietary software usage licenses. For example, the GNU GPL permits redistribution (either for sale or for free) of code, and code can be modified to fit a specific usage. However, the license also mandates that code sold or given away must continue to use the GNU GPL and that modified code must also be made available in source form for not more than the cost of reproduction.

An interesting aspect of the licensing is that nowhere does the GNU GPL state that software licensed on this basis must be free. Several key points of the license include:

- Products obtained and distributed under GNU GPL may be sold or given away. However, software originally obtained under GNU GPL that is subsequently resold, redistributed, or otherwise made available must also be licensed under GNU GPL.
- Users may alter the source code or produce derivative versions of it, or they may integrate subsets of the source code into other works. However, any product that includes GPL code within it must in turn be licensed under GNU GPL.
- Products made available that incorporate GNU GPL code must be accompanied by source code, or source code must be made available at cost no higher than what it costs to reproduce the code on standard medium.

Lesser GPL

The GNU Lesser GPL provides an alternate license for use with code that may be linked to proprietary code. Software created under this license can be statically linked to application code that is distributed under a proprietary license, or to code that is covered by the Lesser GPL, or to code covered by other GNU licensing. The Lesser GPL specifically allows inclusion of Lesser GPL-licensed code within a proprietary application, but the Lesser GPL claims no licensing rights over the combined work, provided the proprietary application is provided in object code and/or source code form. However, if any modification has been made to the Lesser GPL code, the altered code must be made available for reproduction cost (or for free) in source form, as with the GNU GPL license.

BSD License

The Berkeley Software Distribution (BSD) License permits both binary and/or source code redistribution, with or without modification, provided the license is presented to the user when not accompanied by source code. This license is less restrictive to developers than the GNU GPL forms, but it does not ensure that users can gain access to source code associated with a particular product built under the BSD License. One issue that is of concern to GNU GPL developers is that the BSD license is effectively incompatible with the GNU GPL. That is, the licenses have contradictory terms with regard to user access rights to source code.

Apache License

The Apache License is similar to the BSD License. It allows users to modify code and to resell code that is or is not modified. There is no requirement to make modified source code available. This license also is considered to be incompatible with the GNU GPL.

Mozilla Public License

The Mozilla Public License (MPL) code allows incorporation into a larger work, but if the MPL code is modified, source code for the modified component(s) must be made available on the same distribution medium (or electronically). Developers can create a larger work that is not subject to MPL by combining MPL code with other code not subject to the MPL; however, code that originally was covered by the MPL must be made available in source code form.

In summary, developers need to have an understanding of what these licenses represent and how their terms dictate how they can legally use software products distributed under any of these licenses. For instance, a product that combines GNU GPL code with proprietary code will be subject to using the GNU GPL License. On the other hand, a developer that combines Lesser GPL with proprietary code — such as would be the case for linking an application to the GNU C Library (glibc) — can retain full rights to the proprietary code if the links to glibc are dynamic. With this combination, a developer can also gain a level of protection for intellectual property by delivering the proprietary code in object form under Lesser GPL terms.

More information on these and other licenses can be found at **www.gnu.org/copyleft/gpl.html**.

How Competitive Vendors See GNU GPL

Microsoft has led the charge in attempting to discredit products developed under GNU GPL, and it has used a series of derogatory terms to describe the self-preserving clauses within the GNU GPL. The company has stated that it will not port any part of its Microsoft.NET Framework to Linux because of the GNU GPL terms. However, Microsoft is porting subsets of its .NET framework to several other nonMicrosoft EHOEs.

Although users need to be aware of the issues raised by Microsoft, it is important to note the company is doing so as part of a larger campaign to defend its territory from Linux rather than as a public service message designed to help the industry.

Likewise, concerns partially related to the terms of the GNU GPL (and the threat that Linux presents) prompted EHOE heavyweight Wind River Systems to purchase the BSD Unix distribution from Berkeley Software Design Inc. and offer BSD as its embedded Unix solution. Because BSD is protected by the BSD License, which allows redistribution without mandating source code be made available, it allows Wind River to make enhancements to the distribution without having to share those enhancements with competitors. Conversely, this license still can't provide the degree of control or flexibility that the GNU GPL provides to device developers that select Linux. Concerns over the GNU GPL terms even have caused some push-back with the EHOE vendor community. One Linux EHOE vendor develops tools to help customers ensure proper insulation between their application code from the kernel and offers an insurance policy to protect its customers from possible damages should they find their code the subject of GPL-related litigation. This is an area where a strong development partner experienced in open-source development, such as Red Hat, can be provide significant value-add to the development process.

What's a Developer to Do?

The terms of GNU GPL have caused varying levels of concern in different sectors of the industry. For example, device builders that view their development activities surrounding their kernel configuration and the development of kernel plug-ins as key proprietary advantages may find the terms of the GNU GPL unacceptable. This is true for some roll-your-own operating system developers and for companies accustomed to working on a proprietary license model.

IDC believes that GNU GPL will continue to be a concern for organizations that feel their operating system code gives their products a true proprietary advantage. However, the benefit of the open-source development model, where code enhancements are produced by a worldwide network of developers — and are then made available for free to users — offers a compelling alternative that should be considered.

IDC offers the following recommendations for device developers that are considering using GNU GPL technologies within their products:

- Read the terms of the Free Software Foundation and other open-source licenses. The license terms are written in plain English and are easily understood. Don't simply accept the opinions found in industry trade magazines and on newsgroups.
- Consider what licenses may affect work that combines open-source software with a proprietary application-specific software package.

- Know what license applies to which package, when components can be statically integrated with software that is intended to remain proprietary, and when such components being statically integrated into derivative works are intended to be rereleased under an appropriate open-source license.
- Engage corporate law staff to review the terms of open-source licensed products that you choose to use in a product you will be building.
- Create a corporate development policy defining what licenses are to be used in what capacity. Although this adds a certain administrative overhead for developers, using an open-source foundation also provides software engineers a significant jump on product development — at no additional cost. Even better, the technology available under open-source licensing can be expected to continue to mature, which will further broaden the portfolio of available technologies for a developer building a product based on embedded Linux.
- Select a Linux EHOE vendor that has broad experience with development in the GNU GPL world and can provide guidance and education for the options and alternatives that may affect your licensing choices and the level of intellectual property protection that your software receives. Ideally, the vendor should have experience with general-purpose operating systems development, embedded operating system development, and experience with application packages.

If developers still have concerns about possible licensing issues when developing on top of an open-source software infrastructure, they need only look at some of the applications that have been ported to Linux. Heavy hitters, including IBM and Oracle, have ported some of their most important products — including DB2 and Oracle database software — over to Linux. Those companies (and many others that are currently following IBM's and Oracle's lead) wouldn't be offering Linux versions of their products if they had concerns that their code base would be forced to adopt the GNU GPL license.

Linux and GNU GPL

Other issues related to Linux have varying degrees of significance for developers, including the following.

Code Forking

One well-documented failure of the software industry was the inability of the Unix community to build a single industry standard that could compete effectively with Windows operating environments. Unfortunately, vendors saw benefit to offering proprietary extensions to their respective products, which resulted in incompatibility at the application programming interface (API) level (and, accordingly, at the application binary interface [ABI] level).

The Linux developer community has, so far, been highly successful in not only preventing kernel forking but also communicating the importance of this issue to some vendors most responsible for the failure of Unix to become an industry-standard operating environment.

Today, major players such as IBM and Hewlett-Packard Co. have bought into the concept of open-source software and are actively supporting the efforts of Linux community developers. IBM moved quickly to port its middleware and Lotus groupware products to Linux and, subsequently, added the capability to support Linux virtual environments aboard its zSeries (formerly S/390) and iSeries (formerly AS/400) systems. The company established a Linux software group that consists of 200 developers, whose mission is to work directly with the open-source community and help drive forward the enterprise readiness of Linux. The company was also one of the key sponsors of the Open Source Development Lab.

Hewlett-Packard also has taken an active role in Linux, with the company building one of the best Unix-Linux interoperability stories in the industry, particularly aboard IA64 hardware. HP also hired open-source veteran and former Debian project leader Bruce Perens, whose mission includes the mandate to “challenge HP management.”

Compaq Computer Corp. and Sun Microsystems are supporting Linux as well, although to a lesser degree than IBM and HP. Although Sun has claimed a high degree of support for Linux, its initiatives are largely tied to areas unrelated to helping Linux scale onto larger hardware. The company did release the StarOffice product under the OpenOffice.org initiative as open source, one of the computing industry’s larger singular contributions. Compaq has been slower to proactively support open-source development, although it has sought to build a community around its iPaq product.

It appears that these major players (and second- and third-tier players) understand the game rules under which Linux development is being conducted. They also realize that they need to work with, rather than compete against, the Linux development community.

Finally, should a forked kernel emerge, the terms of the GNU GPL ensure that the proprietary advantage of the new version would be lost upon first shipment. This action offers insurance to prevent code forking from taking place.

Linux as an EHOE

Linux is architected for and widely available as an operating system on commodity hardware based on Intel and Intel-compatible x86 processors. Its original intended use was as a general-purpose server and client operating system. But because of the modular nature of Linux, it is relatively easy to slim down the operating environment by removing utility programs, tools, and other system services that are not needed in an embedded environment.

However, because Linux was not specifically architected as an embedded operating system, many of the design points are focused on server- and client-specific requirements. By comparison, most proprietary EHOEs were conceived for and built with embedded deployments in mind. This means that before Linux can be used as an EHOE, some reconfiguration work must take place, a service that is readily available from major Linux software vendors.

Another issue is that the hardware environment where an EHOE is placed is hardly standardized. Unlike the industry-standard x86 PC or server configuration, each embedded device is configured around a specialized form factor optimized for the intended use. Some devices have more than adequate power, memory, and nonvolatile storage resources; other devices are highly power constrained and have minimal storage capabilities. Some devices have a tablet-style interface, others a full keyboard, and other devices have only a few keys for user interaction. Displays can vary from a full VGA-style display format to a single-line display, or even no display for a deeply embedded application.

Vendors of Linux EHOEs, such as Red Hat, take into consideration these form-factor-specific requirements and generally provide substantial custom configuration. Specialized extensions and kernel add-in modules are required to enable their products to function on an embedded device. This customization of the operating environment, per the terms of the GNU GPL, is by definition GNU GPL-covered software.

Benefits of Linux as an EHOE

Given the broad diversity of hardware form factors, deployment environments, and use requirements in the EHOE market, room clearly exists for a great variety of products. In fact, the very diversity of the industry is the same factor that is making it difficult for the industry to consolidate around a small number of products. Unlike the PC operating system industry, where one major player dominates the market with over 90% share of new client operating environment shipments, the EHOE market is made up of at least 25 vendors (with measurable revenue), of which only two companies individually hold over 20% market share.

Linux has emerged as a disruptive technology in this market, as it has in the client and server operating environment markets. It is disruptive because it breaks many of the rules that have previously existed in the market.

GPL Licensing

The first major difference that Linux introduces is the benefit of GPL licensing. GPL ensures that developers have access to source code for the Linux EHOEs they choose to deploy. Although this is not unique in itself — several commercial vendors have offered source code availability in the EHOE market for some time — what differentiates Linux is how that source can be used.

Source code in the EHOE market is typically used as a tool to assist in the debugging process. When debugging a device built using an EHOE, it is not unusual to debug a failing program branch into an operating system call. Some operating environments, which don't offer source code, are in effect a black box fronted by APIs. If an API call fails, it can be difficult to determine what went wrong without access to source code.

However, Linux takes the concept of source code availability well beyond the simple ability to debug application programs. The terms of GNU GPL empowers developers to alter the kernel to suit their needs. Benefits include royalty-free replication, freedom to configure on a per usage basis, and the ability to port to alternate processor architectures. These are generally extra-cost issues for nonGPL EHOEs.

Free Replication and Royalty-Free Deployments

Most vendors of nonGPL EHOEs sell their products using one of several schemes. These licensing programs include:

- **Development seat license(s) followed by a per-deployment royalty.** Under this scheme, one or more development seats are purchased for building the prototype product and the application package. Upon completion of the design and development phase, production units are built, with a per-unit royalty paid to the EHOE vendor (frequently on an open-ended basis). The royalty fee often will range between a $1 and $20 per unit produced, based on the quantity of devices built and the arrangement negotiated with the EHOE vendor at the onset of the project. This scheme is advantageous for developers that may abort or overhaul development plans before any production units are built.
- **Free development license followed by per-unit royalty costs.** This is very similar to the previous program, except that little or no revenue is associated with the development tools and no charge exists for the EHOE when used in development and prototyping. Upon transitioning to production, the EHOE vendor will receive a per-shipment royalty on an open-ended basis. This model is intended to help vendors build market share against competitors by lowering the cost of getting started on a project. This is particularly useful on high-risk projects that may never make it to production phase or have a short production run.
- **High up-front development costs followed by unlimited run-time deployments for a singular design.** Under this scheme, product developers carefully spec their project out and obtain a one-product-design license for unlimited deployment using a vendor's EHOE. The use license is valid for an unlimited number of devices

without further revision. Any change in functionality of the device requires a renegotiation and, potentially, another one-time fee. This program is most beneficial to developers experienced with a given EHOE and has projects that have a high probability of moving into production and will see lengthy production runs without revision. The up-front development fee typically costs $10,000–$15,000 for a development seat.

Configurable Per-Deployment Requirements

Another side effect of the GNU GPL is that users are able to alter the kernel to suit their specific needs. Even better, embedded developers have complete control of the source code that supports their products. This offers a degree of flexibility not available from any proprietary EHOE. If an embedded product company finds it needs to move to a Linux EHOE provider that better suits its support or consulting needs, the organization can bring its source code along without any backward steps on product development. This benefit is not necessarily about being able to tweak the kernel but rather about the level of control developers have over the software pieces that are included in their product. This is one of the outstanding features that GNU GPL brings to developers.

Linux Portability Is Good

Linux has already demonstrated itself to be among the most portable operating systems ever developed. It is currently available to run on virtually every general-purpose microprocessor used in client and server computers. It is well on its way to being one of the most widely ported EHOEs because of the terms of GNU GPL, which makes it possible to create a port without any license negotiations, royalty payment, or other fanfare. At this time, Linux EHOEs are available to support the most common processors used in the embedded world, including the ARM, Strong Arm, MIPS, Hitachi SH, Motorola/IBM PowerPC, and x86-compatible families. EHOE vendors, such as Red Hat, also offer porting services that will quickly move a Linux EHOE to a new processor architecture.

Drawbacks to Linux as an EHOE

What Is Real Time, and When Is it Needed?

The term real-time operating system (RTOS) refers to an operating system that can perform in a predictable and repeatable manner, regardless of workload. A programmer properly using an RTOS can create application threads that will execute in a predetermined time frame (referred to as deterministic behavior), regardless of other events that the processor is expected to respond to. Furthermore, a capability should exist to prioritize individual application threads on a low-to-high priority scale, which ensures that a high-priority application thread will complete before the resumption of a lower-priority thread.

Deterministic behavior should not be confused with high performance or pure response time because it is possible to have a high-speed response that varies significantly in the time to respond. On the other hand, fast deterministic behavior is superior in most situations to slower deterministic behavior.

A good example of when a RTOS capability would be important is when operating within an antilock brake system.

Consider the effect of a processing delay in a vehicle that is moving at 65mph. This motion translates to about 93 feet per second of motion. If the anticipated response from the embedded control system is 50 milliseconds, the vehicle will travel an additional 5 feet before the antilock brake system engages.

However, if a nonRTOS environment was being used and just before the antilock system was called on a disk page event lasting one-half a second took place, the vehicle would travel an additional 50 feet before the engagement of the antilock brake system.

It would not be acceptable for the CPU managing an antilock brake system to complete a low-priority request before responding to a high-priority request.

In process control, for example, repeatable RTOS performance is also an important metric. The failure of a system to repeatedly respond to a signal to shut off flow of one chemical mixing with another could result in a variation of mixture ratios.

Where Is Linux in RTOS Environments?

Because Linux was not designed as a RTOS and only emerged as a viable EHOE technology in 1999, a number of efforts in the industry to increase the RTOS characteristics of the operating environment have appeared. Although some of the changes are relatively ingenious, none have emerged as an "industry standard" form of making Linux more real-time-capable. IDC believes that industry-standard, real-time extensions to the Linux kernel will emerge as the operating environment matures.

The efforts to make Linux a better RTOS include:

- **Creating a RTOS kernel and running Linux as an unscheduled thread under the RTOS.** High-priority threads that need true RTOS performance would run under the host operating environment at a higher priority than would the Linux thread. The problem with this approach is that applications running in the Linux environment won't function in a RTOS mode.
- **Placing "hooks" in the Linux kernel and using those hooks to insert a replacement scheduler that would provide more RTOS-like behavior in the Linux kernel.** The downside to this approach is that the hooks may not be accepted by the Linux kernel developers, which means that with each successive release of

the Linux kernel, the hooks need to be reinserted and tested for proper functionality. It is also possible that the Linux kernel developers could create a different and incompatible set of features, which largely do the same thing. This would call into question the need for the original “hooks.”

- **Altering the kernel directly to replace or change the scheduler or thread prioritization capabilities.** This type of change is unlikely to gain support by the Linux kernel developers and needs to be reintroduced with each successive release of the Linux kernel. This approach could lead embedded device developers down dead-end paths.

When Is RTOS Not Required?

Although RTOS capabilities are important for certain applications, the high speed of today’s processors often lessens the need for hard RTOS capabilities. Many classes of devices have little or no requirement for deterministic behavior. For instance, most personal digital assistants (PDAs), screenphones, and Web terminals have no more requirement for RTOS behavior than does a laptop or desktop computer.

Further, through intelligent structuring of application programs, it is possible to lessen the requirement for RTOS behavior. Even though a great deal of industry emphasis is on RTOS capabilities and the lack of native RTOS behavior in Linux, IDC believes that many segments of today’s EHOE market have little requirement for this feature.

Lack of Track Record

Linux has already achieved some remarkable milestones. Consider that Linux, as an operating environment, has grown from a pet project of a single programmer into a worldwide phenomenon in barely a decade. This history bodes well for Linux to have a bright future. Despite its relative newcomer status to the embedded space, the product’s history in the server market suggests that it will succeed in the embedded space as well, especially if it proves attractive to the vast number of developers that today build and maintain their own EHOEs.

On the other hand, a potential concern with Linux as an EHOE is the lack of a deep track record for Linux EHOE vendors. No Linux vendor in IDC’s database of EHOE suppliers shows EHOE new license shipment and maintenance revenue during CY00 of more than a couple of million dollars. By contrast, Wind River Systems, a long-term fixture in the EHOE market, generated $147 million in EHOE revenue during CY00.

Customers need to consider a number of other factors when evaluating a Linux EHOE vendor. Because the industry is consolidating around a revenue-free license model, revenue generated will likely originate from engineering services, support services, and custom application and middleware development. This model parallels the trends taking place in the Linux client and server operating environment markets. Therefore, measuring the Linux market solely on the basis of license revenue does not provide a complete picture of what it will take for a company to be viable.

Customers will need to evaluate the ability of a Linux EHOE vendor to offer long-term support services for a no-revenue product, yet retain deep technical talent required to continue product development. These demands suggest that a company with a diverse collection of revenue streams originating from related but unique products will have better potential to succeed than a company with only a single product and no supporting revenue streams. One-product companies may eventually need to merge with partner companies, develop revenue-producing products, or may be forced to cease operations.

So far, only one major public Linux company has achieved a profitability. With Red Hat's announcement of its 1QFY02 results, the company led the way for publicly traded Linux companies to break into the black. Given Red Hat's corporate structure, where Linux expertise can be leveraged across products addressing the server, client, and embedded markets, the company is well positioned to succeed long term in the EHOE services market.

A Standard Linux EHOE?

Finally, a potential problem exists in that embedded Linux is not yet a "standard" EHOE. Given the relative newness of the industry, many vendors offer solutions not 100% compatible with one another.

Addressing that issue head on, an initiative was recently launched by the Embedded Linux Consortium (**www.embedded-linux.org**) to promote use of a unified embedded Linux specification across the industry. This initiative is based on existing specifications, such as POSIX 1003.13 and the Linux Standard Base. Considering that the Embedded Linux Consortium counts among its members heavy hitters such as Red Hat, IBM, Agilent Technologies, 3Com, and Palm and has a total membership of over 100 companies, the organization has the potential to strongly influence the direction of embedded Linux. Although this is a step in the right direction, IDC notes that such standards efforts typically take years to move through the development and formal endorsement process.

Comparison to Other EHOEs

Linux not only brings a new licensing and development paradigm to the industry but also brings forward the real potential that, at least for some market segments, an industry-standard EHOE will emerge.

Numerous EHOEs are on the market, yet most products do not compete directly with the same rivals in multiple market segments. For instance, in the PDA space, Palm OS has emerged as one of the leading competitors. It is very atypical to see a strong RTOS-oriented product, such as Wind River's VxWorks or Accelerated Technology's Nucleus, used aboard a PDA. Conversely, in an environment that requires highly reliable real-time performance, such as General Motors' next-generation, in-vehicle control modules, it is not surprising to find a product from Wind River.

Microsoft's Windows CE bridges such environments, but it is generally not considered to be among the most robust of the RTOS products on the market. On the other hand, CE is popular among developers building nonRTOS devices. This popularity is driven by several factors, including the reuse of existing Win32 development skills and the natural affinity that Windows CE has for interoperability with Windows server or client operating environments.

Where will Linux fit best? The answer is relatively easy to predict because of what embedded Linux brings to the table today and where the technology is going. Consider the following areas of support for Linux:

- **Portability.** Linux already has proven itself to be a highly portable operating environment, with commercial distributions available to support common processors, such as Intel x86, MIPS, ARM, StrongARM, Motorola 68K, Dragon Ball, PowerPC, Hitachi Super-H, SPARC, and others. More important, the EHOE is being ported to other architectures on a continuous basis.

- **Compatibility with Internet protocols.** Embedded Linux directly benefits from the leading-edge support that Linux server and client operating environments provide for networking stacks and Internet protocols. This means that embedded developers using Linux automatically get the most current Internet protocol support by default. By comparison, other EHOEs are forced into a continuous development cycle to add support for these same protocols.

- **Open-source application software.** Because some portion of application software written for Linux is released under either GNU GPL or other open-source licensing, developers benefit from being able to borrow from the work of other parties. Likewise, drivers for new hardware devices almost universally are released under GNU GPL or Lesser GPL, making those drivers available without cost to EHOE developers.

Table 1 details factors that should be considered when evaluating Linux EHOEs versus proprietary EHOEs.

Based on these factors, it is likely that Linux will have the greatest penetration in product types that need Internet connectivity or need to interact with client and server systems. It is also likely that the key competitor that Linux will face directly is Windows CE, which is able to leverage its compatibility with Win32 APIs for application, driver, and networking/Internet development.

Table 1
A Comparison Between Linux and Other EHOEs

Feature	Linux	Other EHOEs
Run-time fees	Generally are none, certain vendors offer a run-time fee alternative to other acquisition costs. Industry is moving toward universally embracing the no run-time fee paradigm.	Typically include either a per-unit royalty payment or a one-time payment of $10,000–$15,000 followed by unlimited run-time deployment rights.
Related software availability	Applications and middleware components are available under open-source licensing, making it faster and easier to borrow and modify existing code for new uses. Borrowing such components usually requires propagation of licensing terms covering those components.	Generally is an added-cost item, particularly for highly proprietary EHOEs. Some open-source applications and middleware products can be ported to proprietary EHOEs.
Development tools cost	GNU C Library (GNU libc) and GNU Compiler Collection (GCC) tools (formerly known as the GNU C Compiler) and the GNU Debugger (GDB) are available as open-source development tools, and they offer support for development environments, including C, C++, Fortran, and Java. Other simulation, emulation, and debugging tools are available at additional cost.	Some products are compatible with GNU GPL development tools; others require proprietary tools from EHOE vendor or third-party vendors. Other simulation, emulation, and debugging tools are available at additional cost.
Portability	Any user can port Linux to other hardware platforms or can contract with a service provider to provide offer porting services. No licensing terms required, but resulting port is subject to GNU GPL terms.	Typically controlled by the EHOE vendor.
Networking stack, file system	Networking stack integrated. NFS support integrated. CIFS compatibility available through other GNU GPL code.	Availability varies by products; sometimes it is an extra cost item. Compatibility with NFS or CIFS not always available.
Portability of expertise	Linux is currently being widely used in university settings to ensure a continuous stream of new developers experienced with Linux and Unix environments. Additionally, users with skills in Unix environments are able to move easily to a Linux environment.	Some environments offer good portability of expertise, such as between Windows CE and Windows client and server operating environments. Other EHOEs are more unique in their deployment and require specialized skills.
Operating system source access	Customer has full access to EHOE source code and can use that access to provide self support or can supply that source to alternate Linux vendors for future support if the original vendor is no longer a preferred partner.	Customer may secure access to source code, but, in most cases, access rights are limited to use as a debugging tool and customers hold no further rights to alter or redistribute the EHOE source.

Source: IDC, 2001

The market already is beginning to bear this prediction out. Although still relatively few in number, the products that have emerged using Linux as an EHOE include appliance servers, PDAs, and Web access devices.

Where Embedded Linux Is Used Today

Linux as an EHOE already is making inroads into devices that, yesterday, would have been candidates for other EHOEs. The following products provide examples of devices that have either been announced for future availability or are available today (as noted).

- Intel AnyPoint DSL Gateway 4200 was announced in June 2001 and is expected to reach end users by September 2001. This product incorporates a version of embedded Linux based on the 2.4 kernel, which was ported by Red Hat to Intel's specs, into a package for asymmetric digital subscriber line (ADSL) connectivity in home settings. The device facilitates configuration of a multiuser home network with multiple ethernet connections. The device is built on a StrongArm processor and includes modifications written by Intel to the Linux ATM drivers (which were returned to the open-source community) and an application package authored by Intel. The company cites GNU GPL as being an advantage because it gave Intel access to drivers and source code that would not have been available using other embedded operating environments.
- Ericsson H610 Cordless Web Screen is a wireless device used to surf the Internet. The device is built using Linux as a base operating system, and it is expected to be available this year.
- Gateway Connected Touch Pad Internet device touts a touch-screen LCD and a wireless keyboard. It works with AOL online services and is offered with both telephone and ethernet connectivity. Priced at $499, the device is available today.
- Compaq iPaq SHD is not shipped with Linux on it. However, multiple efforts are under way, coordinated by Compaq, to extend industry support for running Linux on this device. It is possible to install Linux on iPAQ devices today.
- Sun Cobalt Qube 3 appliance uses embedded Linux to provide all the services required in a small office/home office environment, including Web, DHCP, DNS, NAT, as well as mail services. Connectivity is offered over modem or ethernet connection. The company's philosophy is to abstract the operating system away from the user (little mention exists anywhere in the company's marketing materials of Linux as the base operating environment) and provides a Web-based browser interface for system configuration and management. This product has been available in various configurations for four years.
- Sharp Corp. recently announced its intent to use embedded Linux as the EHOE in its next round of PDAs, currently entering development phase. While deployment is probably a year or more away, the volumes have the potential to be high from a consumer products giant such as Sharp.

In addition to these examples, numerous devices have been demonstrated by a variety of vendors over the past 12–18 months. Some of the devices are being used only as concept devices to see what user reaction might be, while others are legitimate forerunners to production runs still in planning stages.

IDC's Perspective on Embedded Linux Use

The market for Linux as an EHOE has great potential. For companies developing products using an EHOE, Linux presents huge possibilities, but only if the developers can see their way past the current concerns regarding GNU GPL and its effect on application-specific configurations and application code.

Some players in the Linux EHOE vendor community have yet to prove themselves with long-term staying power. IDC forecasts consolidation in the industry as the market chooses which vendors will generate an adequate revenue stream to fund ongoing operations.

Linux EHOE vendors need to provide a broad range of technologies and services in order to be successful. IDC believes the successful companies will offer a Linux operating environment surrounded not only by support and engineering services but also by a healthy collection of middleware and application components.

Strong support from major hardware vendors that are both customers and partners will help the successful companies reach this goal.

At the same time, given the high degree of fragmentation in the Linux EHOE market, IDC expects that a transition to favor Linux as the industry's most commonly used EHOE — if it takes place at all — is several years away. The anticipated transition will be most prominent in certain markets, particularly for devices that have a high degree of Internet connectivity and client/server interaction.

Red Hat has a long history in the EHOE industry through the company's acquisition of Cygnus Solutions (tools, services, and embedded operating environments) and WireSpeed Communications Corp. Inc. (embedded networking and telecommunications software). Combining the skills and technology from those strategic acquisitions with the company's financial strength (derived in part from its Linux client and server operating environment business operations), the company is well positioned to compete in the Linux EHOE market. The company is focusing its energy on two major segments of the EHOE market opportunity: telecommunications and consumer electronics, large markets that have significant potential for further growth.

Appendix: Definitions

For the purpose of consistency, IDC uses the following definitions to describe the operating system infrastructure software and the devices themselves that are built using EHOEs.

- **Embedded and handheld operating environments.** EHOEs and related subsystems include the machine-level instructions and general-purpose functions that control the operation and use of CPU resources in smart and embedded devices. They may offer either some form of a user interface (graphical, text, voice, or other) or include the required services or APIs to allow developers to build their own if so required. These environments generally include defined and documented APIs that can be used by application software, which, in turn, is created by related application development tools. EHOE products may include both source code and executable/runtime code, or just executable/runtime code, and may offer either real-time or nonreal-time operational characteristics. Real time is defined as the ability for the system to respond in a predetermined time to scheduled or interrupt-driven events. Nonreal time is defined as the ability of the system to respond rapidly enough to respond in less than a second to network or user input. EHOEs tracked by IDC must be commercially available. Examples of EHOEs tracked by IDC include EPOC, eCos, Linux, LynxOS, Nucleus, Palm OS, QNX, VxWorks, and Windows CE.

- **Industrial control applications.** Industrial control applications are environments that incorporate an EHOE as a component of a larger intelligent subsystem that either partially or fully manages the motions or functional operations of machinery. These applications can range from the fairly obvious, such as computer-controlled metal-working machine tools, including computer numerically controlled (CNC) milling machines, lathes and electrodischarge machining, to an industrial control unit that is integrated with a much larger manufacturing or process control workload. Examples of larger-scale systems include lights-out manufacturing facilities utilizing industrial control-based smart devices and process manufacturing, including controlling process parameters such as volume, temperature, mixing rates and process transportation of gas, oil, chemical, and water products. Such environments would utilize industrial control systems to manage individual processing or manufacturing operations or may use an industrial control system to coordinate the operations of multiple smart industrial devices.

- **Nonconnected smart devices.** IDC defines nonconnected smart devices as devices having a microprocessor, an EHOE, and deployment-specific application software that allows the microprocessor to control the operations of the device. These operations typically provide some intelligent control or operational characteristics of the device itself or of equipment surrounding the device. Examples include antilock braking systems in automobiles, climate control systems both in automobiles and in buildings, traffic control systems, smart vending machines, marine and aviation electronics, and nonconnected personal electronics devices (such as handheld GPS systems).

- **Information appliances.** IDC defines information appliances as a set of new digital electronics products that are consumer focused, low cost (usually less than $500), easy to use, and primarily designed to deliver the interactive benefits of the Internet or an Internet-like service (such as Web browsing or email). The devices offer direct Internet connectivity that enables the user to work interactively with the Internet. In the industry, information appliances are sometimes also referred to as Internet appliances, consumer network computers (NCs), or several other names. Key types of information appliances are NetTVs, screenphones, Internet gaming consoles, Internet smart handheld devices, Web terminals, email terminals, and other devices.
- **Internet smart handheld devices.** Internet smart handheld devices (SHDs) are handheld devices that have Internet access capabilities using an add-on or integrated modem (wired or wireless). In order to be considered an Internet SHD, the unit must allow the user to interact directly and in real time with the Internet. These devices do not use the PC as the gateway to the Internet. Internet smart handheld devices include:
  - **Handheld companions.** These devices are most often categorized as PDAs, PC companions, or personal companions. Characteristics of this product category include the use of these devices as supplements to desktop or portable PCs.
  - **Smart phones.** This category includes the emerging enhanced, superportable cellular phones that enable both voice and data communications. Aside from devices consisting of cellular voice communications, these phones have the ability to access the Internet and store light calendaring and Rolodex data, such as names, addresses, and phone numbers.
  - **Vertical application devices (VADs).** Primarily pen- or keypad-based, these devices are used with specific vertical applications in a variety of industries.
- **NetTVs.** IDC uses the term NetTV to describe an array of television-centric information appliances. NetTVs share as their defining characteristic the use of a television as their primary display. NetTVs can be standalone products that are set on top of the television (set-top), or they can actually be televisions with the NetTV connectivity built in at the time of manufacture.
- **Internet gaming devices.** IDC defines Internet gaming devices as a subset of videogame consoles, which, either directly or via an add-on cartridge, provide access to the Internet, email, or Internet-like

information services. Videogame consoles are machines primarily marketed for delivering electronic games-based entertainment for household consumption.

- **Screenphones.** Screenphones are a type of information appliance that is built around a desktop, telephone-like form factor. A screenphone includes a base module, a voice communications module (a corded or cordless handset and/or speakerphone), a keypad, and a screen display. To be considered a screenphone, a product must provide some benefits of the Internet or an Internet-like service.

- **Email terminals.** Email terminals are standalone, consumer-oriented devices that are specifically marketed and sold as easy-to-use devices for accessing email services. These devices do not include Web browsers and do not allow the user to install any additional applications. An email terminal generally consists of only a keyboard, a small LCD screen, and some soft-function keys. Email terminals are connected to services by a wire (as opposed to cellular or wireless technologies).

- **Web terminals.** These standalone consumer-oriented devices are specifically marketed and sold as easy-to-use devices for browsing the Web or Web-like services (email functionality is usually included). Web terminals are typically based on embedded operating systems that do not allow the user to install off-the-shelf applications that run outside a browser. Web terminals often have an all-in-one form factor that includes a monitor or can connect to a dedicated monitor, but they do not use a television for a monitor.

- **Residential gateway.** The control point through which narrowband connectivity, broadband connectivity, or both enter the home, the residential gateway serves as a secure bridge between the WAN and the LAN. At the very least, the residential gateway must enable and facilitate communications between PC devices; communication with nonPC devices is optimal and expected. The residential gateway combines the functionality of several networking devices, including that of a router, a hub, and digital subscriber line (DSL) customer premise equipment. Most often, it operates with an embedded operating system.

- **Local, wide area, and Internet infrastructure.** Network and Internet infrastructures include active components used to build a LAN, campus, or metropolitan area network, or a WAN. Typically these devices provide a capability to interconnect similar or disparate types of network infrastructures for the purpose of connecting multiple end nodes. These devices will frequently process IP packets in an ethernet environment, but in some device types, they will process protocols in addition to or instead of IP over ethernet or other physical layers. Devices in this category include routers, switches, intelligent hubs, and network gateway devices.

- **Computer infrastructure.** Computer infrastructure includes intelligent subsystems containing EHOE that are incorporated within computer systems. Computer infrastructure includes intelligent I/O subsystems, coprocessing devices, and dedicated devices ("appliance servers") that process computer input and output data streams, such as printers and network-attached storage. Appliance servers are network-enabled devices explicitly designed to provide a single dedicated service, such as Web caching, email, file/print, Internet access, or a predefined suite of services. Appliance servers are nonprogrammable, preconfigured, sealed systems that run on a variety of functionally optimized and/or streamlined operating systems and chip architectures. These devices provide services such as WAP gateways, A/V streaming, VOIP services, and print/document management.

- **Telephony infrastructure.** This infrastructure includes devices providing services and hardware to create private branch exchanges (PBX) as well as merged voice and data networks.

NORTH AMERICA

Corporate Headquarters
5 Speen Street
Framingham, MA 01701
508-872-8200

IDC Canada
36 Toronto Street, Suite 950
Toronto, Ontario
Canada M5C2C5
416-369-0033

IDC Irvine
18831 Von Karman Ave, Ste 200
Irvine, CA 92612
949-250-1960

IDC Mountain View
2131 Landings Drive
Mountain View, CA 94043
650-691-0500

IDC New Jersey
120 Wood Ave South, Suite 509
Iselin, NJ 08830
732-632-9222

IDC New York
2 Park Avenue
Suite 1505
New York, NY 10016
212-726-0900

IDC Texas
100 Congress Ave, Suite 2000
Austin, TX 78701
512-469-6333

IDC Washington
8304 Professional Hill Drive
Fairfax, VA 22031
703-280-5161

ASIA/PACIFIC

IDC Asia/Pacific (Hong Kong)
12/Floor, St. John's Building, 33 Garden Road
Central, Hong Kong
852-2530-3831

IDC Asia/Pacific (Singapore)
71 Bencoolen Street, #02-01
Singapore 189643
65-226-0330

IDC Australia
Level 4, 76 Berry Street
North Sydney
NSW 2060, Australia
61-2-9922-5300

IDC China
Room 611, Beijing Times Square,
88 West Chang'an Avenue, Beijing,
P.R. China, 100031
86-10-8391-3456

IDC (India) Limited
Cyber House
B-35, Sector 32 - Institutional
Gurgaon - 122002
Haryana, India
91-124-6381673 to 80

IDC Japan
10F The Itoyama Tower
3-7-18, Mita Minato-ku
Tokyo 108-0073, Japan
81-3-5440-3400

IDC Korea Ltd
Suite 704, Korea Trade Center
159-1, Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-729
82-2-55-14380

IDC Malaysia
Suite 13-03, Level 13, Wisma KiaPeng
No. 3, Jalan Kia Peng
50450 Kuala Lumpur, Malaysia
6-03-2163 3715

IDC New Zealand
Level 7, 246 Queen Street
Auckland, New Zealand
64-9-309-8252

IDC Philippines
7F, SEDCCO 1Bldg
Rada Street Corner
Legaspi Street
Legaspi Village
Makati City, Philippines
632-894-4808

IDC Taiwan Ltd.
10F, 31
Jen-Ai Rd, Sec 4,
Taipei 106, Taiwan, R.O.C.
886-2-2731-7288

IDC Thailand
27 Soi Charoen Nakorn 14
Charoen Nakorn Road, Klongtonsai
Klongsan Bangkok 10600, Thailand
66-2-439-4591-2

IDC Vietnam
37 Ton Duc Thang Street
Unit 1606
District-1 Hochiminh City Vietnam
84-8-910-1235

LATIN AMERICA

IDC Miami
Latin America Headquarters
8200 NW 41 Street
Suite 300
Miami, FL 33126
305-267-2616

IDC Argentina
Trends Consulting
Rivadavia 413, 4th Floor, Suite 6
C1002AAC, Buenos Aires, Argentina
54-11-4343-8899

IDC Brasil
Alameda Ribeirão Preto, 130 cj 41
01331-000 São Paulo
SP Brazil
55-11-253-7869

International Data Corp. Chile
Luis Thayer Ojeda 166 Piso 12
Providencia, Santiago 9, Chile
56-2-231-0111

IDC Colombia
Carrera 40 # 103-78
Bogota, Colombia
571-533-2326

IDC Mexico
Select - IDC
Av. Nuevo Leon No. 54 Desp. 501
Col. Hipodromo, Condesa
C.P. 06100 Mexico, D.F.
52-5-256-1426

IDC Venezuela
Calle Guaicapuro
Edif. Torre Seguros Alianza
Piso 6, Ofc. 6-D, El Rosal
Caracas 1060, Venezuela
58-2-951-3270

EUROPE, MIDDLE EAST, AND AFRICA

IDC Austria
c/o Loisel, Spiel, Zach Consulting
Mayerhofgasse 6
A-1040 Vienna, Austria
43-1-50-50-900

IDC Benelux (Belgium)
29 Avenue Louis Gribaumont
B-1150 Brussels, Belgium
32-2-779-46-04

IDC Benelux (The Netherlands)
A. Fokkerweg 1
1059 CM Amsterdam
The Netherlands
31-20-669-2721

IDC Central Europe (ECE)
Male Namesti 13
Praha 1 110 00, Czech Republic
420-2-2142-3140

IDC Central Europe (Germany)
Nibelungenplatz 3, 11th Floor
60318 Frankfurt, Germany
49-69-90502-0

IDC Central Europe (Switzerland)
Niederlassung Züerich
WTC, Leutschenbachstrasse 95
CH - 8050 Züerich
Switzerland
41-1-307-1000

IDC Egypt
39 Iraq Street
Mohandesseen, Cairo, Egypt
20-2-336-7355

IDC France
Immeuble La Fayette
2, Place des Vosges, Cedex 65
92051 Paris la Defense 5, France
33-14-904-8000

IDC Hungary
Nador utca 23, 5th Floor
H-1051 Budapest, Hungary
36-1-473-2370

IDC Israel
4 Gershon Street
Tel Aviv 67017, Israel
972-3-5611660

IDC Italy
Viale Monza, 14
20127 Milano, Italy
390-2-284-571

IDC Nigeria
House 2, 'C' Close, 403 Road, 4th Avenue
New Extension, Festac Town
Lagos, Nigeria
234-1-883585

IDC Nordic (Denmark)
Jagtvej 169B
DK-2100 Copenhagen, Denmark
45-39-162222

IDC Nordic (Finland)
Jarrumiehenkatu 2
FIN-00520
Helsinki, Finland
358-9-8770-466

IDC Nordic (Sweden)
Box 1096 Kistagången 21
S-164 25 Kista, Sweden
46-8-751-0415

IDC Poland/ProMarket
Wrobla 43
02-736 Warsaw, Poland
48-22-754-0518

IDC Portugal
Av. Antonio Serpa, 36 Piso 9
1050-027 Lisbon
Portugal
351-21-796-5487

IDC Russia
c/o PX Post, RDS 186
Ulitsa Zorge 10
Moscow 125525
Russian Federation
7-501-929-9959

IDC South Africa
c/o BMI-TechKnowledge
3rd Floor, 356 Rivonia Blvd.
PO Box 4603, Rivonia, 2128
South Africa
27-11-803-6412

IDC Spain
Ochandiano, 6
Centro Empresarial El Plantio
28023 Madrid
34-91-7080007

IDC Turkey
Tevfik Erdonmez Sok. 2/1 Gul Apt.
Kat 9D; 46 Esentepe
Istanbul, Turkey
90-212-275-0995

IDC U.K.
British Standards House
389 Chiswick High Road
London W4 4AE
United Kingdom
44-20-8987-7100

IDC is the foremost global market intelligence and advisory firm helping clients gain insight into technology and ebusiness trends to develop sound business strategies. Using a combination of rigorous primary research, in-depth analysis, and client interaction, IDC forecasts worldwide markets and trends to deliver dependable service and client advice. More than 700 analysts in 43 countries provide global research with local content. IDC's customers comprise the world's leading IT suppliers, IT organizations, ebusiness companies and the financial community. Additional information can be found at www.idc.com.

IDC is a division of IDG, the world's leading IT media, research and exposition company.

Sponsored by Red Hat
01C3026SOFTWA3026
August 2001

IDC
Analyze the Future

IDC
5 Speen Street • Framingham, MA 01701
(508) 872-8200 • Fax (508) 935-4015 •
www.idc.com