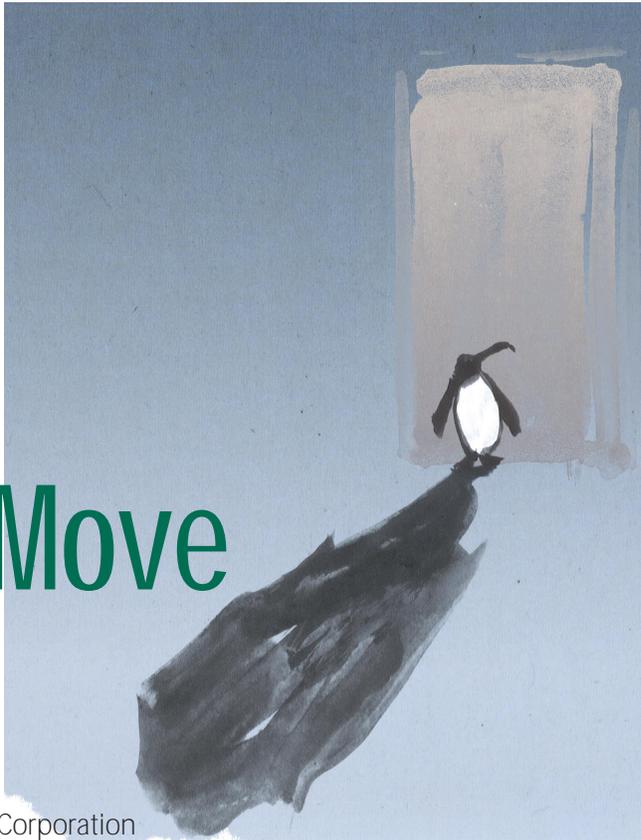




• • •  
What is Linux? And why should you care? This focus section has insights for both newcomers and diehard fans.



## Linux on the Move

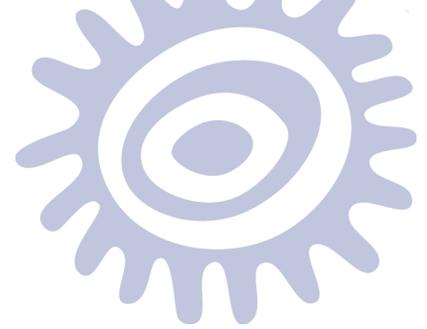


**Terry Bollinger**, The Mitre Corporation  
**Peter Beckman**, Los Alamos National Laboratory



Linux is a free, open-source operating system that looks like Unix, except that it runs on PCs as well as other platforms. Linux was created by Linus Torvalds in 1991. Today, Linux is cooperatively improved by Torvalds and thousands of volunteers from around the world using open-source development methods. At this point in time, "Linux" generally refers to the entire suite of software in a distribution, from the operating system kernel to the Web server and graphical user interface.

When we say that Linux is "free" we mean, well... free. You do not need to pay money to get a copy of it, although it is usually more convenient to buy an inexpensive CD-ROM copy than download an entire distribution over the Internet. Once you get a copy of Linux, you also have the right to make as many copies of it as you want.



## DEFINING TERMS

By “open source” we mean that you also have the right to get copies of all the source code from which Linux and its associated tools were originally compiled. There are no magical, mysterious binary files, although you can of course get the Linux system precompiled if you prefer. Should you at some point encounter a bug, you are perfectly free to fix the source code yourself. In fact, you are encouraged to improve Linux and share your changes with the community. Alternatively, you can report the bug and simply watch someone else fix it for you free, often in a matter of days and sometimes in a matter of hours.

By “Unix” we do not mean some tiny, watered-down subset of Unix. Linux is not a toy operating system by anyone's definition. It has as many or more server and development tools as any traditional Unix OS. It is also a major component of the Internet, where servers use it to provide exceptionally reliable 24-hour-a-day, 7-day-a-week support. Linux has even been used to build powerful parallel-processing supercomputers.

By “PCs as well as other platforms,” we mean that Linux currently has the most impressive cross-platform resume of any operating system—and the list keeps growing. It has been run on the Intel X86, Compaq Alpha, Sun Sparc, Apple Macintosh, SGI MIPS, and even the Palm Pilot.

Linux also possesses an unusually wide range of drivers for cards and peripheral devices. In cases where a newly released board or device is not supported by Linux, it is not uncommon to see the necessary driver show up on the Internet within weeks or even days after the new device first hits the shelves. Usually, the largest impediment standing in the way of developing drivers for new hardware is finding freely available specifications.

By “open-source development” we mean what must be one of the most hair-raising violations of accepted software development practices ever to show up on the software radar screen. Open-source development does not simply ignore accepted precepts of requirements analysis, process maturity, and software configuration control; it laughs at them and tosses them out. The timeline on pp. 32-33 presents a brief history of the movement. The sidebars (pp. 33, 35) provide some additional flavor of what this community is all about.

## GETTING RESULTS

The only traditional software practice that open-source software developers do follow is peer review, and they do that with a vengeance. Each piece of source code is placed on display in front of a global

**The global community of open-source developers is noted for being brutally honest and focused on bottom-line results.**

community of developers who are noted for being brutally honest and very focused on bottom-line results.

The results of this approach include Internet infrastructure components that are arguably among the most reliable pieces of complex software in existence, since even a modest rate of failure in critical components would cause the Internet to degrade to the point of being unusable. Despite the indifference of open-source methods to most software development rules, they do achieve the kind of many-nines reliability to which traditional software development methods still aspire.

## A SPECIAL ISSUE ON LINUX

The Linux (and other) articles in this issue are described in the Article Summaries on page 4. To explore further, see the “Open Source and Linux Resources” box on p. 34.

*IEEE Software* has chosen to focus on Linux not only because it is an interesting and timely topic, but to provide a bit of a wake-up call to traditional software developers who may be a little complacent about Linux and open-source software development. The fact that Linux development does violate commonly held precepts of software development makes it all too easy to dismiss it as “just another software fad” that will soon pass.

The risk of such thinking is that it will fail to recognize and learn from the open-source movement's successes, such as its ability to produce phenomenally reliable software even in the absence of traditional configuration and management controls. A general principle of science is that when blatant contradictions exist between theory and practice, it may

*Continued on page 35*

# A BRIEF HISTORY OF FREE SOFTWARE AND OPEN SOURCE

Jesus M. Gonzalez Barahona and Pedro de las Heras Quirós, Universidad Carlos III de Madrid

Terry Bollinger, The Mitre Corporation

## 1950s and 1960s

Software is distributed with source code and without restrictions in forums such as the “Algorithms” section of *Communications of the ACM*, the IBM Share User Group, and the DEC Decus user group.

## 1969

**April:** RFC 1 describing the first software for the Internet (then ARPANET) is published. Free accessibility to RFCs, and especially to the protocol specifications, was key to Internet development.

## 1971

Richard Stallman begins his career at MIT in a group that uses only free software (nonproprietary software which includes publicly accessible and redistributable source code).

## 1978

Donald Knuth of Stanford University begins working on the TeX system, and distributes it as free software.

## 1980

Early era of nonproprietary software for academic use is largely over. Most software has become proprietary; that is, it is privately owned and its source code is not publicly available.

## 1983

Richard Stallman writes the GNU (“GNU’s Not Unix”) Manifesto, in which he calls for a return to publicly sharable software and source code.

GNU Project begins. Developers begin creating a wide range of generally Unix-like tools and software such as compilers. The Unix kernel is not covered by these early efforts.

## 1985

MIT’s X Window System is distributed as free software under a minimally restrictive license.

## 1989

Cygnus, the first company to identify business opportunities in free software, is founded.

## 1990

Stallman’s organization, the Free Software Foundation, announces its intent to build a powerful Unix-like kernel called GNU Hurd. Their goal is to fill in the last major hole in the GNU suite of software and create a fully nonproprietary development system. However, the scope of Hurd is so large that many fear it will never be completed. (As of early 1999, Hurd is still incomplete.)

## 1991

William and Lynne Jolitz write a series in *Dr. Dobbs Journal* on how to port BSD (Berkeley Software Distribution) Unix to 386-based PCs. This is the start of the BSD family of free OSs (FreeBSD, NetBSD, OpenBSD). The BSD-oriented free-software movement keeps comparatively tight control of the technical content of its freely distributed source code.

**August:** Finnish graduate student Linus Torvalds announces that he has been working on a free Unix-like kernel since April 1991, using GNU tools such as GCC and bash. Initially, Torvalds uses Minix-386 as his development kernel. (Minix is an academically oriented proprietary Unix-like operating system created by Andrew Tanenbaum.)

**October:** Torvalds publicly releases the source code for his Unix-like free-source kernel and names it Linux.

**December:** Torvalds releases the first self-supporting release of Linux, version 0.11. Developers can now work on Linux without using any proprietary tools or operating systems.

## 1992

The US Air Force awards New York University a contract to build a free compiler for what is now called Ada 95. The NYU team chooses GNU GCC for code generation and calls their compiler GNAT, for GNU NYU Ada 95 Translator.

**January:** Tanenbaum publicly criticizes Linux as technically obsolete and overly architecture-specific. In the ensuing Usenet dialog Torvalds adopts an adamantly open-source stance that helps attract new Linux developers.

**July:** 386BSD 0.1 is released by William and Lynne Jolitz, but it is unstable. A legal battle begins to determine if there is any proprietary code in Berkeley Net Release/2, on which 386BSD is based. Some months later, a settlement is achieved with the release of Berkeley Net Release/3, on which all modern free BSD operating systems are based.

## 1993

**August:** Ian Murdock creates a new Linux distribution called Debian Linux, developed by a distributed group of volunteers. Because Debian software can be folded back into other Linux distributions, it does not create any significant long-term split in the basic Linux distribution.

**December:** FreeBSD 1.0, one of the first stable descendents of Jolitz’s earlier 386BSD beta, is released on the Web.

## 1994

Free Ada compiler software receives a commercial boost with the incorporation of Ada Core Technologies, or ACT, by

*Continued on the next page*

the original NYU creators of GNAT. ACT decides to make money by evolving GNAT and selling support services, rather than by selling GNAT itself. Over time and with the help of ACT, GNAT becomes the dominant Ada 95 compiler for most commercial applications.

Marc Ewing begins the Red Hat Linux distribution. Like the Debian distribution, it is intended to improve on the then-dominant Softlanding Linux System (SLS) distribution.

**January:** Debian Linux (version 0.91), developed by 12 volunteers, is released.

**March:** First issue of *Linux Journal* is published.

**October:** NetBSD 1.0 released.

### 1995

**January:** FreeBSD 2.0 is released.

**April:** The first official public release (0.6.2) of Apache is distributed.

### 1996

First Conference on Freely Redistributable Software is held in Cambridge, Massachusetts, USA.

### 1997

**June:** Eric Raymond presents his paper "The Cathedral and the Bazaar" on why the Linux software development works.

### 1998

**January:** Partially persuaded by Raymond's paper, Netscape declares its intent to release the source code for its Navigator browser.

**February:** Chris Peterson and others coin the phrase "open source" to help differentiate the more business-compatible approach to free software found, for example, in the Red Hat Linux and Netscape releases. (The actual Linux distributions remain dependent on GNU and free software, however, and are perhaps better understood as a different way of bundling and promoting multiple strands of both free and open software.)

**April:** Netscape source code is released, and initial fixes and enhancements begin arriving within hours.

**July:** Debian 2.0 is released by more than 300 volunteer package maintainers working on more than 1,500 packages.

**August:** Linux Torvalds and Linux appear on the front cover of *Forbes* magazine.

**October:** IBM decides to go open source with Apache on their AS/400 servers.

**October:** Intel and Netscape invest in Red Hat Linux.

**November:** Internal Microsoft "Halloween" documents analyzing strengths and weaknesses of open-source software and Linux are leaked to the public by Eric Raymond. ❖

Jesus-Mariano Gonzales Barahona and Pedro de las Heras Quirós are assistant professors of computer science at Universidad Carlos III de Madrid, Legano, Spain. Terry Bollinger is a principal information systems engineer at The Mitre Corp. E-mail, jgb@computer.org or terry@mitre.org.

## LINUX: COMPOSABILITY AND RELIABILITY FOR THE CREATIVE TECHNOLOGIST

Dave Hollenbeck, Hewlett Packard

As a microprocessor designer, I must out-think my competitors every day. We are locked in a frantic race to advance computer technology—I need every ounce of creativity I can muster. To apply this creativity, I need tools that are flexible and adaptable.

Prepackaged, one-size-fits-all software seems to be the norm, but my job requires more basic tools. I do not want a painting—I want the paintbrushes, paints, and canvas to create my own work of art. I need to mix and match the new, experimental technologies with the old. If I only had, or could create, a slightly different shade of yellow, I would have the perfect solution, for today. Tomorrow's challenge will perhaps be best met by a unique shade of blue.

Linux offers the programmability and configurability essential for the rigorous task of creating new technology. These qualities are just as necessary in the platform as they are in the applications—I am as likely to find a better way to use my platform as to compute a particular answer. If I cannot get at the functionality within my platform and bend it to my needs, it is that much less useful. In addition, I certainly cannot wait for a long and imperfect software development cycle. Linux offers the most

complete form of flexibility: the source code itself.

Linux is also reliable. Thanks to an extensive community of user/testers, the frustrating machine lockup is a rarity. How can I take seriously a platform which, when I install additional packages, frequently causes existing, working software to fail? My time and efforts need to be focused on real problems, not nuisances of platform or application imperfections. If I do not trust my software, I'll have to do lots of extra work to gain the confidence I need to release a product.

Although other markets may differ, to the creative technologist, the computer is not an appliance. I must have every bit of power at my disposal in my quest to solve technological problems. If I can gain an advantage by using a more adaptable, powerful platform, I will. Moreover, I will not hesitate when I decide. My competitors are at my heels. ❖

Dave Hollenbeck is a technical lead at Hewlett Packard's VLSI Technology Center in Fort Collins, Colorado, where he designs high-performance microprocessors. Readers may contact him at dbh@fc.hp.com.

## OPEN-SOURCE AND LINUX RESOURCES

Linux is gaining momentum as a platform for mainstream computing. Companies such as Computer Associates International, Corel, Informix, and IBM are making versions of their systems and applications available for Linux, often opening the source code to developers. The emergence of these middleware products should enable Linux to gain ground in the commercial marketplace, including mission-critical applications. This list of Web sites and product listings demonstrates the growing acceptance of Linux and open-source software development.

### WEB SITES

The Open Source Page offers general and technical information on open-source development, with case studies and scenarios illustrating its application and usefulness. You can find the "Halloween Documents" here, too. <http://www.opensource.org/>

The Free Software Foundation seeks to eliminate restrictions on copying, redistribution, understanding, and modification of software, and coordinates the GNU operating system development project. <http://www.fsf.org/>

"Release 1.0," Esther Dyson's monthly newsletter, focused on open source in its Nov. 1998 issue. <http://www.edventure.com/release1/1198.html>

O'Reilly & Associates, a publisher, offers news, information, and products for open-source development as well as software development in general. <http://www.oreilly.com/>

The Linux Resources page provides a wide range of information on Linux, including where to get Linux products and support. <http://www.linuxresources.com/>

*Linux Journal*, a monthly magazine from SSC Publishing, is at <http://www.linuxjournal.com/>.

Linux International is an association of companies and groups that work to promote and advance Linux. <http://www.li.org/>

The Linux Documentation Project offers detailed documentation of the Linux OS, general information, development projects, commercial products, and links. <http://metalab.unc.edu/LDP/>

*The Linux Weekly News* is a weekly newsletter with news briefs, editorials, and links on Linux products and the Linux community. <http://www.lwn.net/>

Freshmeat is a software repository that carries a complete set of applications used on Linux systems. The site also hosts Linux news and editorials. <http://freshmeat.net/>

The Linux Kernel Archives is the primary site for the Linux kernel source and has 20 Gbytes of disk space set aside for mirroring the largest Linux-related software archives. <http://www.kernel.org/>

For a list of worldwide open-source and free-software events, see <http://lwn.net/cgi-bin/webcal.pl>.

### PRODUCT INFORMATION

The new Linux 2.2 kernel will be available soon on Intel x86 and 64-bit platforms and on all Motorola chips, such as the

680x0 series and PowerPC family. Updates include improved caching for faster file I/O, improved performance of symmetric multiprocessing, and a more efficient way of dealing with files on network servers. The update does not include USB or DVD features, but subsequent patches are expected. <http://www.kernel.org/>

Ingres II Linux Edition Beta Refresh, Computer Associates International Inc., offers a full replacement of the last Beta Linux Edition with fixes for problems reported during the beta cycle. Ingres II allows Linux users to create applications with GNU development tools (C compilers) and to develop Internet-commerce-enabled applications using the Apache Server. <http://www.cai.com/>

WordPerfect 8 for Linux Personal Edition is available as a free download from Corel. According to the company, there were 82,925 attempts in 12 hours to obtain the free product. Corel has plans to release Linux versions of all its software products, including Paradox, Photopaint, and Draw, before Summer 1999. <http://www.corel.com/>

Informix Dynamic Server, Linux Edition Suite, provides a consistent database architecture and enables the development and deployment of client-server and Web/intranet applications across an organization. The Dynamic Server and the Informix-SE on Linux Developer Bundle are available for download on the Informix Web site until 31 March 1999. <http://www.informix.com/informix/solutions/linux/freelx.html>

IBM's DB2 Universal Developer's Edition Version 5.2 for Linux beta and Net.Data Version 2 for Linux beta code are available for download from the Web. <http://www.software.ibm.com/>

GNUPro 98r2, the open-source compiler and debugger tool-chain for native and embedded software development, is now available from Cygnus Solutions. The product includes ANSI-conforming C/C++ compilers, a macro-assembler, the Cygnus Insight visual debugger, binary utilities and libraries, and support for Windows NT/95 and Unix-hosted toolkits. Cygnus sells GNUPro 98r2 bundled with a subscription service that includes regular upgrades. <http://www.cygnus.com/product/gnupro.html/>

The S.u.S.E. Linux Office Suite 99 available from S.u.S.E. includes Applixware 4.4.1 for Linux. The software package includes a spreadsheet, word processor, presentation graphics, and database and fax program. The suite, compatible with Red Hat, Caldera, and other popular versions of Linux, is designed to enable document exchange with Microsoft Office 97. <http://www.suse.de/e/>

Prolifics 4.0, a 4GL application and Web enterprise development tool, will support the Linux operating system in its January 1999 product release. The product also supports Java and COM and Corba objects and is currently available for Windows NT, Unix, and MVS. <http://www.prolifics.com/>

For additional information on these or other Linux products visit Linux Mall at <http://www.linuxmall.com/>.

# REACTOS: AN OPEN-SOURCE ALTERNATIVE TO NT?

**Jason Filby**, South Africa Free State Health Department

ReactOS is an open-source operating system, currently under development, that will be compatible with Windows NT drivers and applications. We encourage developers around the world to support and contribute to this effort. The home page, [www.sid-dis.com/reactos](http://www.sid-dis.com/reactos), contains project information, mailing lists, message boards, and a download section. Right now, ReactOS is being designed for NT 5 compatibility. The kernel is currently in a pre-alpha state; at the moment we cannot predict when the final release will be available.

The idea of an open-source operating system compatible with NT drivers and applications arose in 1996 when Yannik Majoris decided to form the FreeWin95 project. While many expressed interest in the project, there was little code to show for it. Eventually, the talk died down and the project almost ceased to exist.

In January 1998, project leaders decided the project would either have to be scrapped or taken seriously. We chose to make a fresh start and officially reactivated the project in February 1998. We renamed it ReactOS, as it was a reaction to the Microsoft operating system monopoly. It took a few months for people to settle into their teams and to work out philosophical issues. But even during this initial stage, people were writing code and becoming more interested in doing actual work.

Although in development for just a few months, ReactOS has progressed rapidly. We are currently in pre-alpha release 0.0.0.13. The consistent development progress has continued to attract new members, and we are exploring ways to get current members more actively involved. New code development has temporarily slowed down and given way to bug fixes and docu-

mentation. In fact, our next kernel release will contain mostly bug fixes. We seek to document all coded functions and make this available for online browsing and download.

Our first priority is to provide compatibility with NT drivers and applications. This alone will keep us busy for some time to come—every time Microsoft updates the NT specifications, we'll have to focus on maintaining our compatibility. We are currently exploring the possibility of sharing win32 API information with the WINE project ([www.winehq.com](http://www.winehq.com)).

We also, however, seek to do some things differently than NT. Unfortunately, there's a fine line between maintaining compatibility and using better techniques. We hope, however, to provide compatibility where Microsoft doesn't. For example, some people have trouble sharing Linux and NT on the same machine, or using Microsoft products that don't support 100 percent pure Java. Moreover, Microsoft is integrating its own products into the operating system, thereby shutting out competitors. In such cases, ReactOS will eventually provide a solution and still allow compatibility with NT applications and drivers.

ReactOS has had a short but encouraging life. Its final release will no doubt make waves in the industry. If we can provide a viable alternative to Windows NT, we trust we will have much support. ❖

**Jason Filby** is an Oracle applications developer for the Free State Health Department, South Africa. He is project coordinator for the ReactOS project, to which he contributes on a part-time basis. Readers may contact him at [jasonfilby@yahoo.com](mailto:jasonfilby@yahoo.com).

*Continued from page 31*

be time to go back and look at the theory more closely. For example, just as the Internet is altering the dynamics of markets for physical goods, it may also be irrevocably changing the nature of software development. Software development

is, after all, simply a "market of the minds" in which innovation is expressed and exchanged in the form of software. If Linux is indeed a harbinger of such changes, understanding it better may help equip us for the changes it portends. ❖

## About the Authors



**Terry Bollinger** is a principal information systems engineer at The Mitre Corporation, where he focuses on distributed software architectures. His interests in software and technology encompass such issues as reuse, process, and group intelligence. He is also interested in the possibility that more fundamental foundations to software exist than have currently been recognized. Among his other interests

are physics (he has given a university physics lecture on how light may be emitted from ultrasonically created bubbles), writing, science, and especially spending time with his wife and four young children.

Bollinger has master's and bachelor's degrees in computer science from the University of Missouri at Rolla. Readers may contact him at [terry@mitre.org](mailto:terry@mitre.org).



**Peter Beckman** is a senior computer scientist at the Advanced Computing Laboratory at Los Alamos National Laboratory (LANL). He is the Computer Science Institute project leader for the Accelerated Strategic Computing Initiative at LANL and the team leader for the scalable runtime systems team at the ACL. He also organizes the Extreme Linux workshops and activities ([www.extremelinux.org](http://www.extremelinux.org)) and enjoys tinkering on the ACL's Linux Myrinet "Little Blue Penguin" cluster ([www.acl.lanl.gov/cluster](http://www.acl.lanl.gov/cluster)).

Beckman graduated in 1993 with a PhD in computer science from Indiana University. Readers may contact him at [beckman@acl.lanl.gov](mailto:beckman@acl.lanl.gov).