



UNIVERSIDAD PANAMERICANA
CAMPUS GUADALAJARA

MARÍA DEL ROSARIO AGRAZ CÁRDENAS

IBM and Linux (A)

Examen General de Conocimientos para optar por el título de
Licenciado en Contaduría con reconocimiento
de Validez Oficial de Estudios de la
SECRETARÍA DE EDUCACIÓN PÚBLICA,
según acuerdo número 085366 con fecha 19-Feb.-1985.

CLASIF: TE
CON 1985 AGR
ADQUIS: 5444A ej 1
FECHA: 13/08/04
DONATIVO DE _____
\$ _____
~~_____~~ 62 p.

1. PROGRAMAS DATA COMPUTACION



UNIVERSIDAD PANAMERICANA

CAMPUS GUADALAJARA

VN
129
3-6
SERV
and

MARÍA DEL ROSARIO AGRAZ CÁRDENAS

IBM and Linux (A)



UNIVERSIDAD PANAMERICANA
SEDE GUADALAJARA
BIBLIOTECA

Examen General de Conocimientos para optar por el título de
Licenciado en Contaduría con reconocimiento
de Validez Oficial de Estudios de la
SECRETARÍA DE EDUCACIÓN PÚBLICA,
según acuerdo número 085366 con fecha 19-Feb.-1985.



CARLISS BALDWIN
SIOBHAN O'MAHONY
JAMES QUINN

IBM and Linux (A)

On November 3, 1998, Daniel Frye, a member of IBM's emerging technologies and business team, sat in his office in Somers, New York. He was preparing to write a proposal to senior management detailing the strategic risks and opportunities of a potential alliance with the Linux Development Community (LDC). To some, Frye was an improbable candidate to draft the proposal: With a Ph.D. in Theoretical Physics from Johns Hopkins, and admittedly "not an operating systems guy," his role of in-house Linux advocate could not have been predicted. But Frye had been the first to raise the subject of Linux at an emerging technologies strategy meeting in August 1998. After casually mentioning its growing popularity at the meeting, he had been charged with "figuring out" what to do about Linux and the development community that supported it.

Before beginning his draft of the proposal, Frye took a moment to reflect on IBM management's growing interest in "open source" software and Linux.

Linux was an "open source" operating system that ran on Intel processors, the chips that were used in most personal computers (PCs) and many servers. Unlike Microsoft's Windows™ operating system or Sun Microsystem's Solaris™, the Linux operating system was "open source" software that was not owned by any company.

In 1996, IBM had gained its first exposure to an open source software program when it selected Apache as the Web server program¹ for the Web site of the Atlanta Summer Olympic Games. Apache had performed well, and in the process had exposed IBM executives to the virtues of open source software.

Most proprietary software² was conveyed to the user in the form of "binary code"—that is, the code was already compiled for machines in strings of ones and zeros. Such code could be read by machines but was not readable by human beings. In contrast, *open source software* was conveyed to the user as "source code"—that is, as instructions in a programming language that a human programmer could read and interpret. Because the source code was accessible and because open source licenses permitted it, open source software was easier to customize and change than binary code.

¹ A Web server is a program that processes and sends data, software, or services via the Web to one or more computers.

² Proprietary software is software that is owned and controlled by a company or individual. Users may obtain the right to use such software, generally by paying a fee to the proprietor.

Professors Carliss Baldwin and Siobhan O'Mahony and Research Associate James Quinn prepared this case. This case was developed from published and unpublished sources. HBS cases are developed solely as the basis for class discussion. Cases are not intended to serve as endorsements, sources of primary data, or illustrations of effective or ineffective management.

Copyright © 2003 President and Fellows of Harvard College. To order copies or request permission to reproduce materials, call 1-800-545-7685, write Harvard Business School Publishing, Boston, MA 02163, or go to <http://www.hbsp.harvard.edu>. No part of this publication may be reproduced, stored in a retrieval system, used in a spreadsheet, or transmitted in any form or by any means—electronic, mechanical, photocopying, recording, or otherwise—without the permission of Harvard Business School.

There were no licensing fees associated with using open source code, but, at the same time, there was also no single vendor to stand behind the code and guarantee its features and performance. Linus Torvalds had created the so-called “kernel” of Linux in 1991 while he was a student at the University of Helsinki, Finland. As of 1998, the Linux operating system was developed and maintained by the Linux development community, an amorphous group of volunteer programmers from all over the world.

Obtaining technical support for open source software was a matter of chance and the availability of volunteers. Red Hat, Inc. had started to offer commercial support contracts in December of 1997, but no legal entity assumed primary responsibility for the quality of the Linux kernel.³

IBM’s experience with Apache provided evidence of the business benefits that could be derived from using open source software in IBM installations. That positive experience, in turn, opened the door for IBM to start thinking harder about Linux.

In the fall of 1998, open source software and Linux, in particular, became a focus of media attention and the subject of heated debate within the computer industry. (See **Exhibit 1**.) At the same time, Linux and open source software raised questions for IBM’s technology strategists. Specifically, should IBM resist the spread of Linux and criticize open source software, as Sun Microsystems and Microsoft were doing? Should it remain neutral with respect to Linux, like Dell Computer? Or, was some form of alliance with the Linux development community in IBM’s interests? What in fact constituted the Linux development community? How might such a relationship work?

IBM in the 1990s

So when you pull back from the trees and look at IBM’s forest, I think you see a company that has made solid progress in repositioning our large established businesses while investing aggressively in new growth areas. . . . Our strength is the breadth and depth of all our businesses. Our competitive advantage is our ability to bring all of our global resources to bear on our customers’ opportunities.

—IBM President and CEO, Louis Gerstner⁴

When Louis Gerstner took over as CEO in 1993, IBM was at a crossroads. The company that had established itself as a leader in mainframe computers in the 1970s, and had invented the PC in 1981, reported losses in 1991 and 1992 totaling \$7.8 billion. Some in the industry referred to IBM as a “dinosaur” unable to adapt to the times. Others felt IBM had grown “arrogant” in its lack of customer focus. Gerstner—the first CEO to be hired from outside IBM’s ranks—acknowledged these deficiencies and added his own, calling IBM “too bureaucratic, too preoccupied with our own view of the world . . . and way too slow getting things to market.”⁵

Gerstner, former CEO of RJR Nabisco and a onetime McKinsey consultant, established four initial goals upon arriving at IBM: to be profitable; to become more competitive; to increase shareholder value; and to grow. He immediately embarked on a restructuring program, reducing the workforce by 45,000 (or 15%) and cutting overall expenses by \$2.8 billion. He also placed employees’ focus on

³ The kernel is the part of the operating system that is responsible for juggling various tasks (i.e., tasks such as sending information to a printer, displaying information on screen, getting data from a Web site, etc.).

⁴ Speech at the 1997 IBM Annual Shareholders’ Meeting, Dallas, Texas, April 29, 2002, <<http://www.ibm.com/lvg/annual97.phtml>>, accessed November 11, 2002.

⁵ IBM, 1993 Annual Report, p. 3.

the customer, declaring, "I want everyone at IBM to be obsessed with satisfying the customer."⁶ To Gerstner, IBM's customer base was one of its main competitive advantages, along with its R&D capabilities and its employee talent base.⁷

Over the next five years, IBM did return to profitability, albeit at a modest rate of growth. IBM fought hard to penetrate the critical client/server market while also managing its growing portfolio of products and services, which included servers, middleware,⁸ professional services, networking, microelectronics, and data storage. IBM also undertook key initiatives that helped stabilize the company within the industry, including the acquisition of Lotus, which gave IBM a lead position in the collaborative software industry. (See **Exhibit 2** for selected financial data and **Exhibit 3** for a breakdown of IBM's revenues by industry segment.)

In 1998, Gerstner declared that "The PC era is over"⁹ and placed IBM's prospects for growth squarely in what he called "network-centric computing." In the new network-centric environment, Gerstner believed that speed-to-market and flexibility would deliver the highest value to IBM customers.

In this new strategic context, an alliance with the Apache development team was a first, tentative step in the direction of working with an open source software development community. There were many open source communities, however, and the Linux development community had its own norms and conventions. For one thing, the Linux community used a software license with much tighter restrictions than were applied to Apache. In addition, Linus Torvalds was a more powerful figure, and exercised more direct control over Linux than any member of the Apache community exercised within that group. Finally, the Linux community was not as clearly defined as the supporters of Apache. Thus, while the Apache experience had been overwhelmingly positive for IBM, working with Linux was in many ways an entirely new ball game.

History of Open Source Software

The roots of the Linux development community stemmed from the "free software movement," which was founded by Richard Stallman in the mid 1980s. Indeed, "free software" was a precursor of "open source software," and many people saw them as one and the same.

In 1971, as a freshman at Harvard, Stallman became a "hacker" at MIT's Artificial Intelligence (AI) Laboratory.¹⁰ At that time, he was initiated into the "playful cleverness" of hacker culture, often sleeping on the floor of the lab when "exhaustion won out over inspiration" during all-night programming sessions. He also embraced the ethic of sharing software code, which was normal

⁶ IBM, 1993 Annual Report, p. 5.

⁷ IBM, 1993 Annual Report, p. 3.

⁸ Middleware embraces all elements of infrastructural software that rides on top of an operating system such as database, Webserving, systems management, collaborative, and networking software.

⁹ IBM, 1998 Annual Report, p. 7.

¹⁰ According to *The New Hacker's Dictionary*, a hacker is 1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary. 2. One who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorizing about programming. Jargon File Version 4.0.0, Jul 25 1996, <http://www.jargon.8hz.com/jargon_23.html#SEC30>, accessed November 11, 2002.

practice among hackers. According to Stallman, "Whenever people from another university or a company wanted to port and use a program, we gladly let them."¹¹

In this environment of community collaboration, Stallman thrived: In 1975, only one year after graduating *magna cum laude* in Physics from Harvard, he led the development of Emacs, the most powerful text editor of the day, "as central to the way people worked then as the Web browser is for the Internet age."¹²

But in the late 1970s and early 1980s, the software-sharing community began to lose ground to corporations seeking to own and profit from proprietary software. Many of the original hackers with whom Stallman worked were hired into industry. At the same time, software became a product, and software vendors began placing restrictions on the sharing of their code.

Stallman rejected the evolving proprietary system. He believed it would allow vendors to exercise undue power over users. He described the new norms based on intellectual property as follows: "If you share with your neighbor, you are a pirate. If you want any changes, beg [the owner] to make them."¹³

In an effort to limit the power of proprietary software developers and to re-establish a free software community, Stallman set out to create a new operating system that would be freely available to all. He believed that a free operating system was the best place to focus his attention because of its fundamental role in the management of a computer. Stallman based his design on Unix, a popular operating system that had the flexibility to run on a variety of computers. He named his system GNU, a recursive acronym for "GNU's not Unix." In 1984, he posted a call to others to join his project, and out of this effort, the free software movement was born.

Stallman initially chose the phrase "free software" to describe both GNU and his movement. He kept this name even though the term "free" created misunderstanding in the minds of many outside the movement. "Free software" as Stallman conceived it did not preclude the sale of software, as many observers assumed. Rather, the term was meant to invoke "four freedoms" designed to protect what Stallman called the "natural rights" of users. These were: 1) the freedom to run a program for any purpose; 2) the freedom to modify a program to suit one's needs; 3) the freedom to redistribute copies either *gratis* or for a fee; 4) and the freedom to distribute modified versions of programs—so that all users could benefit from the improvements.

General Public License (GPL)

Stallman feared that without some form of protection, the GNU source code would be appropriated by some for-profit entity and become part of a proprietary code base:

I wanted to give the community a feeling that it was not a doormat, a feeling that it was not prey to any parasite who would wander along. . . . [Potentially,] anybody can come along and say: "Ah, I want to make a non-free version of this. I'll just take it." And, then, of course, they probably make some improvements, those non-free versions might appeal to users, and replace the free versions. And then, what have you accomplished? You've only made a donation to some proprietary software project. And when people see that that's happening,

¹¹ Chris DiBona, Sam Ockman, and Mark Stone, eds., *Open Sources* (Sebastopol, CA: O'Reilly & Associates, Inc., 1999), p. 53.

¹² Glyn Moody, *Rebel Code: Inside Linux and the Open Source Revolution* (Cambridge, MA: Perseus Publishing, 2001), p. 16.

¹³ DiBona et al., p. 54.

when people see other people take what I do, and they don't ever give back, it can be demoralizing.¹⁴

To prevent this from happening, Stallman drafted the GNU General Public License or GPL. The GPL allowed source code to be modified by the user and distributed to others without restriction. However, Section (2)(b) of the license required that all modifications or derived works released to the public must also be redistributed under the same terms. (See **Exhibit 4** for the full text of Section 2 of the license.) "You must cause any work that you distribute or publish, that in whole or part contains or is derived from the Program or any other part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License."¹⁵

Stallman coined the phrase "copyleft" to characterize his counterintuitive legal standard. As he put it: "Copyleft used copyright law, but flipped it over to serve the opposite of its initial purpose: instead of a means of privatizing software, it became a means of keeping software free."¹⁶ As opposed to claiming "copyright, all rights reserved," Stallman encouraged programmers to license their software "copyleft, all rights reversed."¹⁷

Linux

Origins

Between 1984 and 1990, Stallman made substantial progress in building the GNU operating system. He and other volunteers worked thousands of hours on the code and created a large suite of non-proprietary, Unix-compatible tools. But, as of 1990, no one in the GNU Project had managed to develop a crucial element of the software—the kernel. The kernel is the part of the operating system that is responsible for juggling various tasks (i.e., tasks such as sending information to a printer, displaying information on screen, getting data from a Web site, etc.). For Stallman, the delay in supplying the kernel was not serious enough to undermine his project; it was more important to him that he had successfully created most of the elements, and "that the missing piece of the puzzle was on its way."¹⁸

But Linus Torvalds, a second-year computer science student at the University of Helsinki and a self-taught hacker, did not want to wait for the GNU Project to supply a kernel that would allow him to run Unix on his Intel-based PC. Thus, in April 1991, he set out to create an operating system complete with a kernel for his own use. He decided to model his program after Minix, a "micro-kernel" created by Andy Tanenbaum, who was a well-respected researcher and professor of computer science at Vrije University in the Netherlands. Torvalds believed Minix (which was used primarily as educational tool) could be greatly improved. Thus on Sunday, August 25, 1991, he posted this note to the *comp.os.minix* newsgroup on the Internet:

¹⁴ Siobhan O'Mahony, "The Emergence of a New Commercial Actor: Community Managed Software Projects" (Ph.D. diss., Stanford University, 2002).

¹⁵ Free Software Foundation, Section (2)(b), GNU General Public License, Version 2, June 1991.

¹⁶ DiBona et al., p. 59.

¹⁷ Open Resources Web site, <<http://www.openresources.com/documents/gnu-project/node9.html>>.

¹⁸ Moody, p. 31.

Hello everybody out there using minix—

I'm doing a free operating system (just a hobby, won't be big and professional like gnu) for 386 (486) AT clones.¹⁹ This has been brewing since April, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat . . . I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :)

The response to this public introduction of Linux was remarkable. Programmers and would-be users contacted Torvalds in great numbers over the Internet to offer ideas and suggestions for the new operating system. During these early stages of development, Torvalds and Tanenbaum engaged in a very public debate about the relative merits of Minix and Linux. The professor posted a well-known barb at Torvalds' work in January of 1992: "I still maintain the point that designing a monolithic kernel in 1991 is a fundamental error. Be thankful you are not my student. You would not get a high grade for such a design :-)." ²⁰

Monolithic or not, the kernel that Torvalds developed was auspiciously compatible with Stallman's GNU components. Indeed Torvalds integrated his kernel with components from the GNU Project in late 1992 and licensed it, as was the protocol, under the GPL.

Growth

Combining the Linux kernel with the GNU system resulted in a complete and free operating system that ran on a personal computer. This development re-energized the free software movement. Between 1991 and 1998, Linux grew from 10,000 lines of code used by one person to 1.5 million lines used by 7.5 million people worldwide. It was estimated that over 10,000 programmers—mostly volunteers—had a hand in revising the code. Technically, Torvalds had written less than 5% of the program that bore his name.

By the late 1990s, Linux was a full-featured and highly functional operating system. Users had a couple of different ways to obtain the code. They could simply download it from the Internet for free, or, beginning in the mid 1990s, they could buy Linux from a software distributor such as Red Hat, Inc. or Caldera International for a fee of about \$50.00. The number of CDs shipped to customers was believed to have grown from 450,000 in 1996 to 750,000 in 1997.²¹

Exhibit 5 shows server license share data for Linux and other operating systems as of the end of 1997, the most recently available data. Data on 1998 shipments and downloads were not yet available, but many observers believed that Linux's share of server licenses was growing even more rapidly than it had in 1997. (The exhibit shows the breakdown of server *units* using Linux and other operating systems, not the breakdown of *revenues* to each operating system provider.) Estimates of revenue share were likely to underestimate the growth and penetration of Linux for three reasons: First, users who downloaded the software for free would not show up in the revenue figures. Second, Linux was often used on redeployed hardware, which did not create revenue for any vendor, hence

¹⁹ "386 (486) AT clones" referred to personal computers based on the Intel 80386 and 80486 microprocessors that were compatible with IBM's PC/AT specifications.

²⁰ DiBona et al., p. 225.

²¹ As reported in the Halloween document: Open source Web site, <<http://www.opensource.org/halloween/halloween2.php#quote5>>, accessed November 11, 2002.

did not show up in revenue share figures. Third, even when Linux was deployed on new hardware, it typically was used in systems that were smaller and less expensive than average.

Some analysts believed that Linux's strong performance and low cost were the key drivers of its growth and market penetration. Others pointed to Linux's ability to be narrowly tailored to a user's needs, through adjustments in the code.

For their part, entrepreneurs from startup firms like Red Hat, Caldera, and Cygnus Solutions, which were trying to sell Linux products and services, stressed the advantages of open source code over binary code. Bob Young, the founder of Red Hat, compared buying binary code to buying a car with the hood closed: "Would you buy a car that you could not inspect, fiddle with, and repair the inside? That is what buying binary code is like. You are buying a car with the hood sealed shut: you have no ability to change and modify it."²²

Hype

Customers could see the advantages of using Linux but worried about the availability of applications that could run on it and the ability of other vendors and service providers to support it. Netscape's release of their Internet browser's source code in March of 1998 surprised some analysts, but was viewed as an important sign of the growing acceptance of Linux and the open source development model.²³ It was clear that firms were starting to rethink how their business models might benefit from Linux.

Netscape's release of their source code also led some leaders and activists in the free software to rethink their attitude towards commercial enterprise. Movement leaders decided that they needed a marketing program. They worried that the term and image associated with "free software" would be unattractive to companies who might otherwise decide to use the operating system. One of the leaders of the Open Source Movement explained:

People could not identify with the word "free software"—it was too scary. People needed less confrontation and a less political term. It was dragging us down. A good example is the term "free software" that sounds really good to an idealistic, shaggy-haired hacker in Birkenstocks but scares . . . Jay Random in Techio. We were . . . not adapting our language to our target audience.²⁴

In April 1998, a group that included Torvalds and other long-term members of the free software community met to discuss the issue of public perception, and coined the term "open source." The new name took off in the popular press (see **Exhibit 1**). Reporters and business analysts could now get a sense of why "free" or "open source" code might actually be superior to proprietary or closed code from a user's point of view.

The market penetration of Linux (see **Exhibit 5**), combined with changing perceptions and good press, helped ignite a furious debate within the computer industry. On the one hand, many in the industry believed Linux would become a leading operating system for business computing on a par

²² Michael Kanellos, "Red Hat founder: Microsoft Isn't Net-Ready," *CNET News.com* (June 13, 2000).

²³ Netscape Network Web site, <<http://wp.netscape.com/newref/pr/newsrelease591.html>>, accessed November 11, 2002.

²⁴ O'Mahony.

with Microsoft NT. Ken Jacobs, vice president of the server market division at Oracle, was such a proponent. He asserted: "[Linux] is faster than [Windows] NT, has better support and is cheaper."²⁵

This thinking was echoed by venture capitalists like William Gurley of Benchmark Capital:

At first glance, open source development seems a ludicrous concept: Get a fragmented, distributed group of volunteer programmers to develop and debug software and expect them to work in harmony toward a common goal, in spite of the fact that they may never meet, may never make a dime for their efforts, and must, by the way, freely distribute the source code they create. Yet that concept may be the most powerful movement in software.²⁶

At the same time, Linux had a large number of opponents who believed the operating system would never cross the gap into mainstream business use. In the words of John Rose, group general manager at Compaq's Enterprise Consulting Group: "Support for Linux is growing, but it is not going to displace mission-critical Unix or Windows NT."²⁷

Other critics emphasized the uncertainty of technical support, as expressed in the words of a technical systems specialist from Stillwater, Minnesota: "Linux won't work in the business community until a central site is created for technical support issues. No business in this country is going to wait for a 17-year-old beatnik to [answer its newsgroup post and] fix its problems."²⁸

Still others pointed to Linux's lack of a graphical user interface (GUI), as well as the conservative nature of IT professionals who would be unwilling to bet their jobs on the performance of an open source operating system.

In July of 1998, leading database vendor, Oracle, Computer Associates, and Informix announced that they would support Linux,²⁹ and in October that same year, HP also announced that some of their Internet solutions would support Linux.³⁰ Throughout the fall of 1998, momentum was building behind Linux, open source software, and the open source development model. Every day there seemed to be new Web sites emerging that analyzed and talked about the Linux phenomena: *LinuxToday* and *LinuxJournal* were attracting large numbers in addition to bulletin-board styled Web sites such as Slashdot that were favored by Linux programmers.

A confidential Microsoft document evaluating the competitive risk posed by Linux was unintentionally released and published on the Internet in early November. This report, nicknamed by community members as the 'Halloween document,' caused ripples throughout the Linux community and in the press. Skeptics were not sure whether the report was truly an internal

²⁵ Barbara Darrow and Steven Burke, "Linux to Find Home in Small, Midsize Market," *Computer Reseller News*, November 23, 1998, p. 225.

²⁶ J. William Gurley, "Why free code makes sense," *Fortune*, August 2, 1999, p. 228.

²⁷ Darrow and Burke, p. 225.

²⁸ Paul McNamara and Christine Burns, "Linux Cynics," *Network World*, November 9, 1998, p. 176.

²⁹ Red Hat, Inc. Web site, <http://www.redhat.com/about/presscenter/1998/press_jul2398.html>.

³⁰ *Ibid.*

Microsoft strategy document.³¹ But if it was, one thing was clear: Microsoft considered open source development methods and Linux in particular to be a viable competitive threat.

IBM's Stance toward Linux

Dan Frye was intrigued by the growing momentum that was building behind open source. Given all the hullabaloo in the press and the media, it seemed that IBM should be grappling with the implications of the free software movement, open source development methods, and especially the extraordinary growth and market penetration of Linux. IBM's core strategy was to supply hardware, software, and professional services to its clients, who were primarily the MIS departments of large and medium-size corporations. Such departments were the stewards of mission-critical software applications as well as the day-to-day information flows within their corporations. They were understandably risk-averse, and looked to IBM to supply close to fail-safe systems at a reasonable cost.

However, with the Internet boom, the market had changed. ISPs (Internet Service Providers), ASPs (Application Service Providers), dot-com start-ups, and universities had different priorities than traditional MIS departments. These customers cared about speed, flexibility, and the ability to innovate and scale using the Internet. They were not as risk-averse as many of IBM's longstanding corporate clients. To gain access to this market, IBM needed to grow its software and middleware business and do so on a neutral platform, one that did not depend on products proprietary to Microsoft or other competitors. This was easier said than done.

Like many other hardware vendors and systems integrators, IBM maintained large software development projects aimed at enhancing the operating systems that were shipped on its computers. In late 1998, those operating systems included Microsoft's Windows NTTM and IBM's in-house version of Unix, AIXTM. IBM was one of Microsoft's largest customers, and the two companies cooperated in many ways.

IBM also competed with Microsoft in the application program market—i.e., programs that run “on top of” different operating systems. Indeed, IBM was the second largest software vendor in the world, with 1997 revenues of almost \$13 billion (see **Exhibit 3**). The premier software company, Microsoft, sponsored Windows NT, which was a competitor of Linux. Thus, Microsoft was not likely to port its middleware or application programs to Linux. In fact, the leaked ‘Halloween’ document stated that a strategic option available to Microsoft was to extend or develop new protocols to “deny OSS [open source software] projects entry into the market.”³²

However, the growing trend in the industry at large was to do exactly the opposite—to create and sustain open standards. Open standards were technical standards that supported computing functionality independent of a particular vendor's platform or application, and that were available for all to read and implement without royalties or fees. Many in the industry believed that open standards were essential to innovation on the Internet and to network-centric computing in general.³³ Thus, “driving” open standards was a key element of IBM's network-centric strategy.

³¹ A Microsoft spokesperson later admitted that this document was created internally for competitive research purposes, but that it was “not an official statement by Microsoft on the issue of open source software or the Linux model.” <http://web.archive.org/web...om/ntserver/nts/news/mwarv/linuxresp.asp>.

³² Open Source Web site, <http://www.opensource.org/halloween/halloween1.php>.

³³ Bruce Perens Web site, <http://www.perens.com/OpenStandards/Definition.html>.

In some ways, Linux was the ultimate open standard: it was an operating system that “supported computing functionality independent of a particular vendor’s platform or application” and was also “available to all to read and implement without royalties or fees.” But Linux was controlled by the Linux development community, one of many free software/open source communities that had sprung up on the Internet.

As a corporation, IBM had little experience dealing with open source development communities. It had established a friendly and productive arrangement with the Apache group, but, as indicated, the Apache community was different from the Linux community. Thus, IBM executives worried about whether their company could work with the Linux development community. For one thing, the community consisted only of individuals. Could IBM, a large corporation, enter and play a role in the community? How would it do so? Could it enter in a low-profile way? Could it fit in?

Beyond these practical considerations, Frye and his colleagues worried that IBM’s joining the Linux community would undermine LDC’s identity as a free and open alternative to corporate computing; they worried that once IBM arrived, Linux’s status as a “cool” enterprise might diminish, driving away the very hackers and programming talent that had led to Linux’s success.

Potential Advantages and Disadvantages of Linux for IBM

From IBM’s perspective, Linux offered several potential advantages as an operating system. First, it was a “neutral” platform, not owned by any other corporate entity. Thus, IBM’s software could run on top of Linux without creating a strategic vulnerability to any other corporation. This was also consistent with IBM’s strategy of supporting open standards.

Second, Linux appeared to be exhibiting extraordinarily high growth and penetration in the PC and lower-end server markets. Seemingly, those who used it were very satisfied with its performance. Indeed, some recent formal benchmarking studies had ranked Linux equal to Windows NT in terms of speed and reliability, although, as always, the results were disputed.³⁴

Third, the operating system was “free.” However, “free” was a misnomer when applied to any large software program: the code still had to be supported, and support costs often dwarfed the purchase costs. More intriguing was the fact that the source code was “open”—that is, viewable by anyone and modifiable by the licensee without violating the licensor’s copyright. In theory, such transparency, together with the rights to change the actual code base, might translate into large cost savings and easier upgrade paths for users and systems integrators alike. However, there was as yet no data to support such a theory.

Set against these advantages were a large number of disadvantages. First and foremost was the question of Linux’s long-term sustainability. Although it was successful in the low-end server markets, Linux software development and support appeared very *ad hoc*. Users of Linux could not look to a corporate vendor for support and bug fixes; instead they had to rely on a horde of programmers who worked on Linux for fun and for free. (Of course, since the code was open and modifiable, users could in principle look to their own in-house staff to fix bugs as well.)

Second, IBM would have to address the question of control. The operating system of a computer is a key part of its architecture and a “point of control.” The instructions (called APIs or Application Programming Interfaces) built into the operating system are embedded in essentially all application programs that run on the system. Microsoft’s phenomenal economic success over 20 years proved just how valuable control of an operating system could be.

³⁴ Mindcraft, Inc. Web site, <<http://www.mindcraft.com/whitepapers/fist-nts4rhlinux.html>>.

In the case of Linux, control of the code base was vested in Linus Torvalds and a handful of trusted lieutenants. Unlike the Apache Project, which was governed by a non-profit corporation with a board of directors elected by its members, Linux was run as a virtual dictatorship. Debates over code structure and features might rage on the message boards, but ultimately, Torvalds had the last word on what went in to each Linux release. (After graduating from university, Torvalds had gone on to work as a designer for Transmeta Corporation, a California-based startup. As part of his contract, he spent some of his time managing Linux.)

Thus, although under the GPL anyone could modify the code base for their own use, only people that Linus or one of his key lieutenants trusted had the power to “commit” code to official Linux releases. None of these “committers” had any connection with IBM. Indeed, some members of the free software and open source communities viewed IBM with great antagonism exceeded only by their hostility towards Microsoft Corporation.

At the same time, Frye knew that most employees of IBM were at best, unaware and at worst, distrustful of free software. The development practices of open source were very different from traditional coding practice, and antithetical to the way that many of IBM’s best programmers were accustomed to doing their work. As an example, most proprietary code was seen and reviewed by only a few people with “access privileges” to specific parts of the code base. In contrast, open source projects were based on the principle that *everyone* could see every line of code. Literally anyone could comment on the quality of any snippet of posted code, and the criticisms leveled at new members of the community could be harsh and insulting.

The last disadvantage of Linux was uncertainty surrounding the scope and enforceability of the General Public License. As indicated earlier, the license contained a reciprocal provision: all works derived from the Linux source code had to be openly provided as well. (See **Exhibit 4.**) Would the GPL apply to proprietary code written to run on Linux? If so, IBM would have a hard time justifying the development of proprietary code to run on Linux.

Others claimed that this argument was another example of the sowing of “fear, uncertainty and doubt” (FUD) by enemies of free software and open source. Open source advocates articulated that the GPL covered only modifications to a GPL licensed program and did not “infect” other programs that might surround it. Proprietary and GPL licensed code could be placed on the same CD-ROM or machine without violating the GPL, but they could not be ‘co-mingled’. How and where this line would be drawn in practice was unclear, but what was clear was that the scope of the GPL had never been tested in a court of law.

Alternative Approaches to Linux as of 1998

In late 1998, there were fundamentally four possible ways for IBM to approach Linux. First, IBM could join the opponents of Linux, including Sun and Microsoft. It could decree that Linux would not be shipped on IBM hardware nor supported by IBM. However, this position seemed to run counter to IBM’s commitment to “put the customer first,” because some customers preferred to use Linux and were asking for it.

Second, IBM could adopt a neutral position, similar to that of Dell Computer Corporation. In this case, if a customer asked for Linux, IBM, like Dell, would “load it and ship it.”

Third, beyond “loading and shipping” the operating system, IBM could publicly support the goals of free software and the open source movement, but stay distant from the communities and their development practices. This option would entail little internal change for IBM.

Fourth, IBM could proactively seek some form of alliance with the Linux development community. As indicated, Linux was a neutral operating system. Every computer needs an operating system. However, IBM's revenues and profits were primarily derived from hardware, middleware and services, not operating systems. Thus, both IBM and the Linux community would arguably benefit from the availability of a high-quality, easily maintained, widespread, non-proprietary operating system.

However, even if IBM were to seek an "alliance" with the Linux community, it was not clear how such an arrangement could possibly get off the ground. Leaving aside the possibility of distrust and outright hostility towards IBM in the free software and open source movements, the community of Linux developers was a loosely affiliated group of people living all over the world. They communicated via message boards and newsgroups on the Internet and seldom met face to face.

Nevertheless, there was an organization, of sorts, within the Linux development community. (See Exhibit 6.) Linus Torvalds was still the leader of the project: he managed and announced all releases. Some fondly called him a "benevolent dictator."³⁵ In 1996, his primary lieutenant, Alan Cox (who lived in Wales), had created a role for 'maintainers': experienced contributors who were responsible for particular modules of the kernel and reviewed code provided by a larger body of credited developers.³⁶

Approximately 200 credited or core developers were explicitly acknowledged in the credits file which was listed as part of the program.³⁷ An even larger population of approximately 15,000 casual or occasional contributors volunteered bug fixes, suggestions code, or ideas over the life of the project.

Contributors to Linux liked to call their organization a meritocracy. Over time, developers who contributed more code, and higher quality code, were granted greater respect and greater project responsibilities. Thus, an occasional contributor could become a credited developer and eventually a maintainer if the group deemed his or her skills and contributions worthy enough. However, the structure of the project was still quite loose, and it was hard to tell how decisions actually got made.

In theory, a representative of IBM might try to contact Torvalds (or one of his lieutenants) and propose a potential alliance. One problem with this idea was that there were few private forums to test the waters with such a proposal. The programmers who contributed to Linux worked in public and were known to be strongly opinionated. Second, it was not clear whether Linus, or any other hacker, would value what IBM had to offer. How an overture from one of the largest companies in the computing industry might be received was unknown. While the Linux community seemed to be more welcoming to industry as of late, it made sense to be concerned about the possibility of being rebuffed or outright rejected. How would the idea of such a partnership be accepted? How could IBM communicate the possibility and benefits of an alliance? Was an alliance the right strategy?

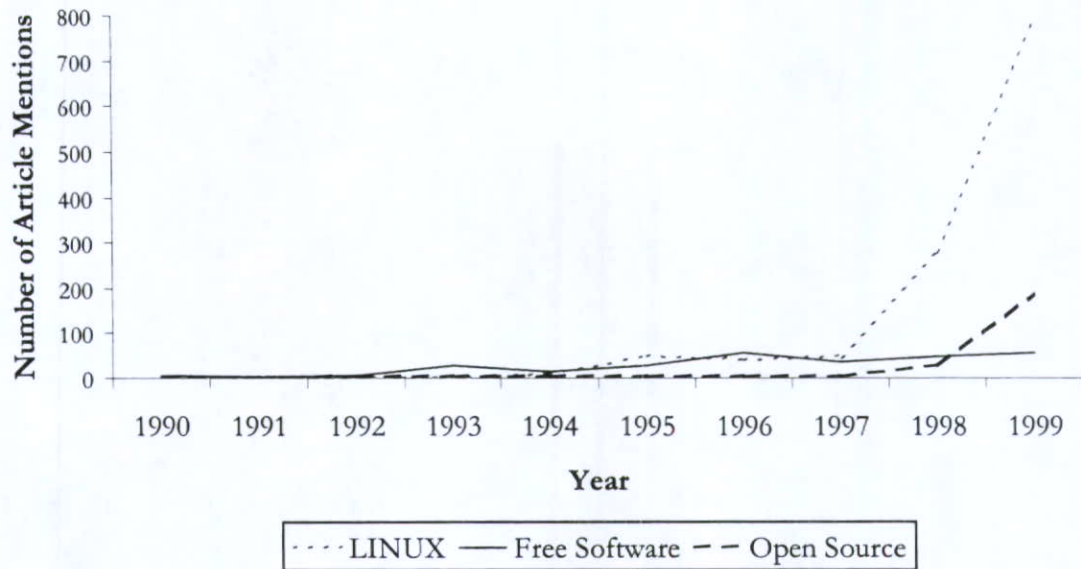
Frye wondered what IBM's next steps should be. And, he thought, if IBM did take a proactive stance, did he personally want to be involved in the process?

³⁵ Eric Raymond, *The Cathedral and the Bazaar* (Sebastopol, CA: O'Reilly Publications, 1999).

³⁶ Jae Yun Moon and Lee Sproull, "The Essence of Distributed Work: The Case of the Linux Kernel," in *Distributed Work*, edited by Pamela J. Hinds and Sara Kiesler (Cambridge, MA: MIT Press, 2002).

³⁷ Moon and Sproull.

Exhibit 1 Mentions of Linux, Free Software, or Open Source in the Press 1990–1998



Source: Adapted from articles cited in Inform/Proquest using the words "Free Software," "Linux," or "Open Source" in the subject, abstract, or keyword fields.

Exhibit 2 IBM Corporation Selected Financial Data FY 1990-1997 (in US\$ millions, except per-share amounts)

FOR THE YEAR	1997	1996	1995	1994	1993	1992	1991	1990
Revenue	\$78,508	\$75,947	\$71,940	\$64,052	\$62,716	\$64,523	\$64,766	\$68,931
Net earnings (loss)	6,093	5,429	4,178	3,021	(8,101)	(4,965)	(2,861)	5,967
Per share of common stock	6.18	5.12	3.61	2.51	(7.11)	(8.70)	(5.01)	10.42
Cash dividends paid on common stock	763	686	572	585	905	2,765	2,771	2,774
Per share of common stock	0.775	0.65	0.5	0.50	0.79	4.84	4.84	4.84
Investment in plant, rental machines and other property	6,793	5,883	4,744	3,078	3,232	4,698	6,502	6,548
Return on stockholders' equity	29.7%	24.8%	18.5%	14.3%	--	--	--	14.8%
AT THE END OF THE YEAR								
Total assets	\$81,499	\$81,132	\$80,292	\$81,091	\$81,113	\$86,705	\$92,473	\$87,568
Net investment in plant, rental machines and other property	18,347	17,407	16,579	16,664	17,521	21,595	27,578	27,241
Working capital	6,911	6,695	9,043	12,112	6,052	2,955	7,018	13,313
Total debt	26,926	22,829	21,629	22,118	27,342	29,320	26,947	19,545
Stockholders' equity	19,816	21,628	22,423	23,413	19,738	27,624	36,679	42,553

Source: Adapted from 1994 and 1997 IBM Annual Reports.

Exhibit 3b IBM Segment Information

IBM is in the business of providing customer solutions through the use of advanced technologies. The company operates primarily in the single industry segment that creates value by offering a variety of solutions that include, either singularly or in some combination, services, software, systems, products, financing and technologies.

For purposes of classifying similar information technology products, general-purpose computer systems that operate on a large class of applications are classified as servers when the systems are simultaneously used by multiple users at one time, or as clients when the systems are used by one user at a time. Clients include personal computer and RS/6000 products, general-purpose display-based terminals and monitors, and consumer and financial systems. Servers include the System/390, AS/400, RS/6000 and personal computer server products. Storage consists of externally attached direct access storage devices and tape storage devices. Other peripherals consist of advanced function printers and telecommunication devices. OEM hardware consists primarily of revenue from the sale of HDD storage files and semiconductors.

These hardware classes of products represent groupings that perform similar functions, as opposed to the complete spectrum of products associated with IBM's product divisions. Accordingly, they do not represent the full range of any division's offerings, which could include related peripherals, software and maintenance.

Services represent a full range of solutions in Network Services, which includes managed network operations and services; Professional Services, consisting of systems management or outsourcing, systems integration design and development, education and consulting; and Product Support Services, which consists of availability services for operation support and business recovery systems. Software includes applications and systems software for both the host and distributed systems. Maintenance consists of separately billed charges for maintenance. Financing and other is composed primarily of financing revenue and products and supplies not otherwise classified.

Some products logically fit in more than one class and are assigned to a specific class based on a variety of factors. Over time, products tend to overlap, merge into or split from existing classes as a result of changing technologies, market perceptions and/or customer use. For example, market demand may create requirements for technological enhancements to permit a peripheral product to be functionally integrated with a display, a telecommunication device and a processor to form a workstation. Such interchangeability and technological progress tend to make year-to-year comparisons less valid than they would be in an industry less subject to rapid change.

Source: Excerpted from IBM 1997 10-K.

Exhibit 4 GNU General Public License, Section 2

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

Source: Richard Stallman, GNU General Public License, Version 2, June 1991, from GNU Project Web site, <<http://www.gnu.org/copyleft/gpl.html>>, accessed June 20, 2003. Copyright © 1989, 1991 Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111 USA.

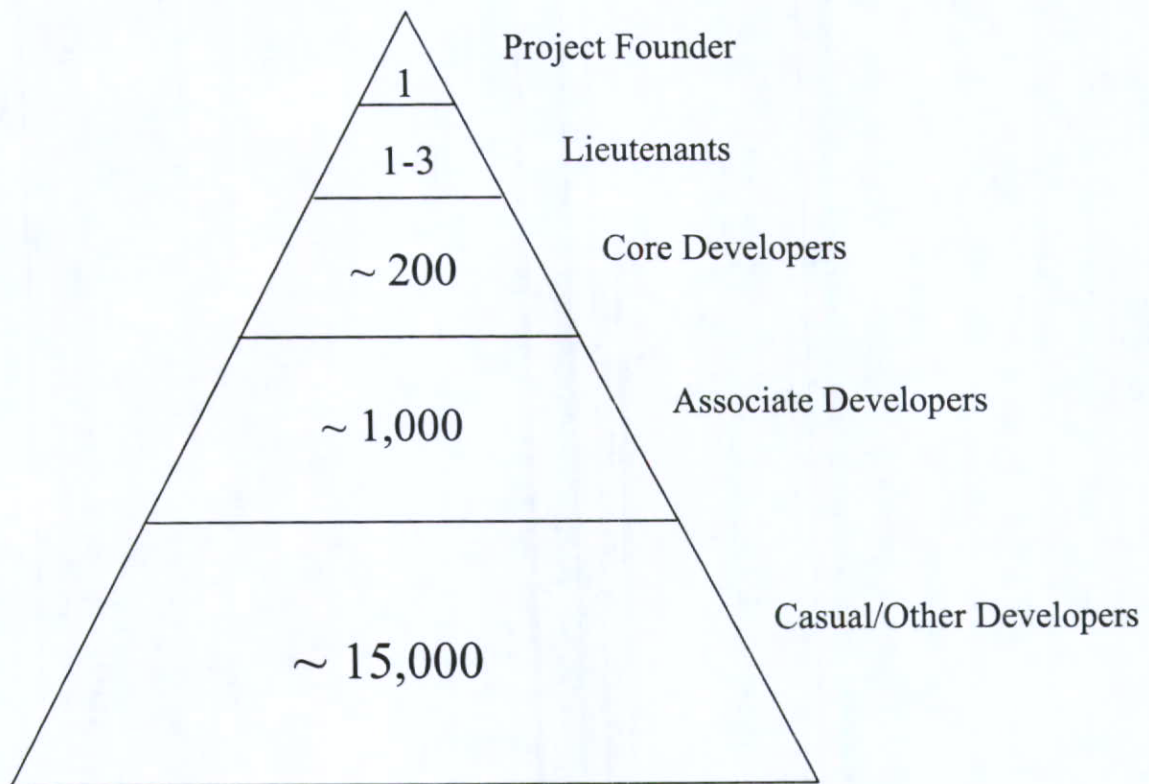
Exhibit 5 Server Market Penetration of Linux Relative to Other Operating Systems

SERVER OPERATING SYSTEM		1996	1997	Growth
NTS	units	805,200	1,505,000	86.91%
	%	23.30%	36.60%	
Netware	units	993,000	900,000	-9.37%
	%	28.73%	21.89%	
Linux	units	200,000	240,000	20.00%
	%	5.79%	5.84%	
Solaris	units	81,000	99,500	22.84%
	%	2.34%	2.42%	
SCO	units	226,000	288,000	27.43%
	%	6.54%	7.00%	
Other	units	1,150,537	1,079,478	-6.18%
	%	33.29%	26.25%	
Totals		3,455,737	4,111,978	18.99%

Key: NTS refers to Microsoft Corporation's Windows NT Server™ operating system; NetWare™ was produced by Novell Corporation; Solaris™ by Sun Microsystems; and SCO™ by the Santa Cruz Operation.

Source: Adapted from Open Source Web site, <www.opensource.org/halloween/halloween2.php#_Toc427383650>, accessed November 1, 2002.

Exhibit 6 The Organization of the Linux Development Community as of 1998



Project Founder—Linus Torvalds.

Lieutenants—Key Linux kernel “Maintainers” who reviewed mailing list submissions (bug reports, bug fixes, new features), built submitted code into larger patches of code, and submitted larger patches back to the mailing list and to Torvalds directly.

Core developers—Many were “credited developers” who had been listed publicly for their substantial contribution to the development of the Linux kernel.

Associate Developers—Broken into two groups: “Development Team” created small patches of code, added features, fixed bugs; “Bug Reporting Team” identified bugs, characterized bugs, and eliminated bugs.

Casual/Other developers—These part-time members of the community, located in 90 countries around the world, contributed comments, documentation, artwork, and small patches of code, often for applications.

Sources: Pamela Hinds and Sara Kiesler, *Distributed Work* (Cambridge, MA: MIT Press, 2002), pp. 393–396. Siobhan O’Mahony, “The Emergence of a New Commercial Actor: Community Managed Software Projects” (Ph.D. diss., Stanford University, 2002)

INDICE

	Página
1. Traducción caso IBM Y LINUX (A)	1
2. Resumen de los hechos	34
3. Síntesis de problemas detectados	
3.1 Análisis FODA	42
3.1.1 Fortalezas	42
3.1.2 Oportunidades	42
3.1.3 Debilidades	43
3.1.4 Amenazas	44
3.2 Jerarquización de los problemas	45
4. Solución 1	47
5. Solución 2	52
6. Resumen ejecutivo	55
7. Conclusiones	56
8. Anexos	58
9. Bibliografía	60

I. TRADUCCIÓN

IBM y Linux (A)

El día 03 de noviembre de 1998, Daniel Frye, miembro del equipo de tecnología emergente y negocios de IBM, sentado en su oficina en Somers, Nueva York. Estaba preparando el escribir una propuesta al directivo de administración detallando los riesgos estratégicos y oportunidades de una alianza potencial con la Comunidad Desarrolladora Linux (LDC). Para algunos, Frye era el candidato menos indicado para desarrollar el borrador de esta propuesta: Con un Doctorado en Física Teórica de Johns Hopkins, y admitiendo que no se trataba de un operador de sistemas, su papel de que alguien de casa soportara la idea Linux no pudo haber sido prevista. Pero Frye había sido el primero en tocar el tema de Linux en la junta de la estrategia en tecnologías emergentes en Agosto de 1998. Casualmente después de mencionar su creciente popularidad en la junta, se había estado encargando de investigar que hacer acerca de la comunidad desarrolladora que soporta a Linux.

Antes de empezar su anteproyecto de la propuesta, Frye tomó un momento para reflexionar sobre el interés de IBM de crecer con un software abierto y Linux.

Linux era un sistema operativo libre que era ejecutable en los procesadores Intel, los chips que eran utilizados en la mayoría de las computadoras personales (PCs) y muchos servidores. Diferente al sistema operativo de Microsoft Windows TM o Sun Microsystem's Solares TM, el sistema operativo Linux era un software libre que no era propiedad de alguna compañía.

En 1996, IBM había ganado su primer lanzamiento de un software abierto cuando había seleccionado a Apache como el programa¹ servidor Web para el sitio Web de

¹ Un servidor Web es un programa que procesa y envía datos, software, o servicios por medio del Web a una o mas computadoras.

los Juegos Olímpicos de Atlanta. Apache había funcionado bien, y en el proceso había expuesto a los ejecutivos de IBM a las virtudes de un software abierto.

La mayoría del software patentado² se había dado a conocer al usuario en forma de código binario –es el código que había sido compilado para máquinas en cadenas de uno y ceros. Tal código pudo ser leído por las máquinas pero no pudo ser leído por seres humanos. En contraste, el software abierto fue dado a conocer al usuario como código fuente –es en forma de instrucciones en un lenguaje de programación que un programador humano puede leer e interpretar. Porque el código fuente era accesible y porque las licencias de la fuente abierta lo permitían, el software libre fue más fácil de diseñar de acuerdo a los deseos y necesidades del cliente que el código binario.

No hubo honorarios de licencia asociados con el uso del código fuente abierto o libre, pero, al mismo tiempo no hubo también vendedor alguno que estuviese respaldando el código y garantizar sus características y funcionamiento. Linux Torvalds había creado el tan llamado “núcleo” de Linux en 1991 mientras era estudiante de la Universidad de Helsinki en Finlandia. Al año 1998, el sistema operativo Linux fue desarrollado y mantenido por la comunidad desarrolladora de Linux, y un grupo desorganizado de programadores voluntarios de todo el mundo.

Obtener soporte técnico para un software libre fue cuestión de oportunidad y disponibilidad de voluntarios. Red Hat, Inc. Había empezado a ofrecer contratos comerciales de soporte en Diciembre de 1997, pero ninguna entidad legal asumió tan fundamental responsabilidad para la calidad del núcleo³ de Linux.

² Software patentado es el software que es poseionado y controlado por una compañía o una persona. Los usuarios deberán adquirir el derecho de usar tal software, generalmente pagando una cuota al propietario.

³ El núcleo es la parte de un sistema operativo que es responsable de ejecutar varias actividades o tareas (ejm., tareas tales como enviar información a la impresora, desplegar información en pantalla, obtener datos de un sitio Web, etc.).

La experiencia de IBM con Apache proporcionó evidencia de los beneficios de negocio que podrían derivar del usar software libre en las instalaciones de IBM. Esa experiencia positiva, a cambio, abrió la puerta para que IBM empezara a pensar arduamente acerca de Linux.

En el otoño de 1998, el software libre y Linux, en particular, fue un punto de atención de los medios publicitarios y la materia de un debate acalorado dentro de la industria computacional. (Ver anexo 1.) Al mismo tiempo, Linux y el software libre dió lugar a cuestionamientos para los estrategas tecnológicos de IBM. Específicamente, ¿IBM iba resistirse a la expansión de Linux y criticar el software libre, como lo estuvo haciendo Sun Microsystems y Microsoft? ¿Iba permanecer neutral respecto a Linux, como Dell Computer? O, ¿estuvo en los intereses de IBM alguna forma de alianza con la comunidad desarrolladora de Linux? ¿Que constituyó en realidad a la comunidad desarrolladora Linux? ¿Cómo funcionaría tal relación?

IBM en los años 1990

Cuando te retiras de los árboles y miras al bosque de IBM, pienso que ves a una compañía que ha realizado un progreso sólido al reposicionar nuestros grandes negocios establecidos mientras se invierte agresivamente en nuevas áreas de crecimiento... Nuestra fuerza es la extensión y profundidad de todos nuestros negocios. Nuestra ventaja competitiva es nuestra habilidad para brindar todos nuestros recursos globales para relacionarnos con las oportunidades de nuestros clientes.

-Presidente y CEO de IBM, Lous Gerstner⁴

⁴ Discurso en la junta anual de accionistas de IBM en 1997, Dallas, Texas, Abril 29, 2002, <http://www.ibm.com/lvg/annual97.phtml>, accesado en Noviembre 11, 2002.

Cuando Luis Gerstner tomó el puesto de CEO en 1993, IBM estaba en una encrucijada. La compañía, que se había posicionado como la líder en fabricación de computadoras en la década de 1970, y había inventado la PC en 1981, reportó pérdidas totales en 1991 y 1992 de \$7.8 billones de dólares. Alguien de la industria se refería a IBM como un dinosaurio incapaz de adaptarse a los tiempos. Otros sintieron que IBM había crecido de una manera arrogante al no preocuparse por enfocarse al cliente. Gerstner —el primer CEO contratado fuera de las personas con mejores posiciones de IBM— reconoció estas deficiencias y agregó su adjetivo, llamando a IBM “demasiado burocrática, demasiado preocupada con nuestra propia manera de ver al mundo... y la manera tan lenta de lanzar productos al mercado.”⁵

Gerstner, un CEO formado en RJR Nabisco y por una temporada consultor de McKinsey, estableció cuatro metas una vez que llegó a IBM; ser rentable; ser mas competitivo; incrementar el valor de la acción, y crecer. Inmediatamente inició un programa de reestructuración, reduciendo la fuerza laboral por 45,000 (o 15%) y recortando gastos por \$2.8 billones de dólares. También enfocó a los empleados sobre los clientes, declarando. “Quiero que todos en IBM se obsesionen con la satisfacción del cliente.”⁶ Para Gerstner, basarse en el cliente de IBM fue de sus principales ventajas competitivas, junto con las capacidades de R&D y la base del talento de sus empleados⁷.

Sobre los siguientes cinco años, IBM regresó a la rentabilidad, aunque a un modesto grado de crecimiento. IBM peleó arduamente para penetrar al crítico mercado del servidor mientras manejaba el crecimiento de su portafolio de productos y servicios, que incluía servidores, middleware⁸, servicios profesionales, redes, microelectrónicos, y almacenamiento de datos. IBM también emprendió iniciativas

⁵ IBM, Reporte anual 1993, p.3.

⁶ IBM, Reporte anual 1993, p.5.

⁷ IBM, Reporte anual 1993, p.3.

⁸ Middleware abarca todos los elementos del software infraestructural que dirige a lo mejor de un sistema operativo como una base de datos, servidor Web, administración de sistemas, compartición y software para red.

clave que ayudaron a estabilizar a la compañía dentro de la industria, incluyendo la adquisición de Lotus, el cual llevó a IBM a una posición cooperativa en la industria del software. (Ver anexo 2 para datos financieros seleccionados y anexo 3 para un desglose de las ventas de IBM por segmento)

En 1998, Grestner declaró que “La era del PC se terminó”⁹ y posicionó a los prospectos de IBM para crecer firmemente en lo que él llamó “computación de redes centrales.” En el nuevo ambiente de redes centrales, Gerstner creyó que la velocidad hacia el mercado y flexibilidad permitirían entregar el valor más alto a los clientes de IBM.

En esta nueva estrategia contextual, una alianza con el equipo desarrollador de Apache fue un escalón tentativo dirigido hacia trabajar con una comunidad desarrolladora de software libre. Había muchas comunidades de fuentes abiertas, de cualquier modo, la comunidad desarrolladora de Linux tuvo sus propias normas y convencionalismos. Por algo, la comunidad Linux utilizó una licencia de software con muchas restricciones estrictas que fueron aplicadas a Apache. Adicionalmente, Linus Torvalds fue una figura mas poderosa, y ejerció un control mas directo sobre Linux que el ejercido por cualquier miembro de la comunidad Apache dentro de ese grupo. Finalmente, la comunidad Linux no estaba definida claramente como el soporte de Apache. De esta manera, mientras la experiencia de Apache había sido inmensamente positiva para IBM, trabajar con Linux era en muchas maneras una pelota de juego totalmente nueva.

Historia del Software libre

Las raíces de la comunidad desarrolladora de Linux resultaron del movimiento del software libre, que se fundó por Richard Stallman a mediados de 1980. En efecto,

⁹ IBM, Reporte anual 1998, p.7.

el software libre fue un precursor del software de fuente abierta, y mucha gente los vió como uno y el mismo.

En 1971, como un estudiante novato de Harvard, Stallman era un "pirata informático" en el Laboratorio Artificial de MIT (AI).¹⁰ En aquel tiempo, había iniciado en la "inteligencia humorística" de la cultura del pirata informático, constantemente durmiendo en el piso del laboratorio cuando "el agotamiento había salido victoriosa sobre la inspiración" durante sesiones de toda la noche de programación. Él también se aprovechó de la ética de repartir el código del software, lo cual era una práctica normal entre piratas informáticos. De acuerdo con Stallman, "Cualquier persona de otra universidad o compañía que quisiera portar y usar un programa con gusto los dejamos."¹¹

En este ambiente de comunidad cooperativa, Stallman prosperó: En 1975, solo una año después de haberse graduado *magna cum laude* en Física de Harvard, encabezó el desarrollo de Emacs, el editor de texto mas poderoso de la época, "tan central para la manera como la gente trabajaba así como el buscador Web es para la era de Internet".¹²

Pero a finales de 1970 e inicios de 1980, la comunidad del software compartido empezó a perder terreno dentro de las corporaciones que buscaban el poseer y lucrar con el software patentado. Muchos de los piratas informáticos originales con quienes Stallman trabajó fueron contratados dentro de la industria. Al mismo tiempo,

¹⁰ De acuerdo con el diccionario *The New Hacker's Dictionary*, un hacker es 1. Una persona que disfruta de explorar los detalles programables de los sistemas y como sacar sus capacidades, opuesto a los demás usuarios, que prefieren aprender solamente lo mínimo necesario. 2. Una persona que programa entusiastamente (siempre obsesivamente) o disfruta más de programar que de teorizar acerca de la programación. Archivo Jargon Versión 4.0.0, Jul 25 1996, <http://www.jargon.8hz.com/jargon_23.html#SEC30>. , accesado en Noviembre 11, 2002.

¹¹ Chris DiBona, Sam Ockman, y Mark Stone, eds., *Open Sources* (Sebastopol, CA: O'Reilly & Associates, Inc., 1999), p.53.

¹² Glyn Moody, *Rebel Code: Inside Linux and the Open Source Revolution* (Cambridge, MA: Perseus Publishing, 2001), p. 16.

el software se convirtió en producto, y los vendedores de software empezaron a poner restricciones al compartir sus códigos.

Stallman rechazó el evolucionado sistema patentado. Él creyó que eso permitiría al vendedor el ejercer un poder indebido sobre los usuarios. Él describió las nuevas normas basadas en la propiedad intelectual como sigue: “Si tu compartes con tu vecino, eres un pirata. Si quieres cambios, suplica (al propietario) que los hagan.”¹³

En un esfuerzo por limitar el poder de los desarrolladores del software patentado y para restablecer una comunidad de software libre, Stallman empezó a crear un nuevo sistema operativo que estuviera disponible gratuitamente para todos. Él creyó que un sistema operativo gratis era el mejor lugar para enfocar su atención por su papel tan fundamental en el manejo de una computadora. Stallman basó su diseño en Unix, un sistema operativo popular que tuvo la flexibilidad de correr en una variedad de computadoras. Le llamó a su sistema GNU, una sigla recurrente para “GNU no es Unix.” En 1984, hizo un llamado a otros para unirse a su proyecto, y participar de este esfuerzo, el movimiento del software libre había nacido.

Stallman inicialmente escogió la frase “software libre” para describir a ambos GNU y su movimiento. Cuidó este nombre aún cuando la palabra “libre” creó malos entendidos en las mentes de muchos que estaban fuera del movimiento. “Software libre” como Stallman lo concibió no excluye la venta del software, como muchos observadores asumieron. Mejor dicho, la palabra significó el invocar “cuatro libertades” diseñadas para proteger lo que Stallman llamó “derechos naturales” de los usuarios. Estos fueron: 1) la libertad para correr un programa para muchos propósitos; 2) la libertad de modificar un programa para adecuarlo a las necesidades personales; 3) la libertad para redistribuir copias gratis o a cambio de un cuota; 4) y la libertad para distribuir versiones modificadas de programas –para que todos los usuarios pudieran beneficiarse de las mejoras.

¹³ DiBona et al., p.54.

Licencia Pública General

Stallman temía que sin ningún tipo de protección, el código fuente GNU fuera apropiado por alguna entidad lucrativa y hacer propio el código base:

Yo quise dar a la comunidad el sentimiento de que se trataba de una falta de respeto, un sentir de que no se trataba de cazar a cualquier parásito que se dirigiera sin rumbo... (Potencialmente,) nadie puede venir a decir: "Ah, quiero hacer una versión no gratuita de esto. Me lo llevo". Y, entonces, claro, ellos probablemente harán algunas adecuaciones, esas versiones no gratuitas podrían atraer a los usuarios y reemplazarían a las versiones gratuitas. Y entonces, ¿que habrás logrado? Habrás hecho solo una donación para un proyecto de propiedad de software. Y cuando la gente vea que eso está sucediendo, que otras personas estén tomando lo que hago y que nunca lo regresarán, podrá ser desmoralizante.¹⁴

Para evitar que esto sucediera, Stallman hizo un preliminar de Licencia General Pública GNU o GPL. La GPL permitió que el código fuente fuera modificado por el usuario y distribuido por otros sin restricción. Como siempre, Sección (2)(b) de la licencia requiere que todas las modificaciones o trabajos que hayan sido liberados al público deben ser redistribuidos bajo los mismos términos. (Ver Anexo 4 para el texto completo de la Sección 2 de la licencia.) "Se origina cualquier trabajo que se distribuya o publique, que en un todo o en partes contenga o sea derivado del Programa o de cualquier otra parte concerniente, sujeto a ser licenciado como un todo sin cargo a las terceras partes bajo los términos de esta Licencia."¹⁵

¹⁴ Siobhan O'Mahony, "The Emergence of a New Commercial Actor: Community Manager Software Projects" (PhD. Diss., Universidad de Stanford, 2002).

¹⁵ Fundación de Software Libre, Sección (2)(b), GNU Licencia Pública General, Versión 2, Junio 1991.

Stallman inventó la frase “deja copiar” para caracterizar su poco intuitivo estandar legal. Cuando lo creó dijo: “Dejar copiar utiliza la ley de derechos reservados, pero fue lanzado para servir a lo opuesto para lo que fue creado inicialmente; en vez del significado de privatizar software, significa cuidar que el software sea libre.”¹⁶ Como lo opuesto al reclamo “derechos de autor, todos los derechos reservados,” Stallman animó a los programadores a licenciar su software “dejar copiar, todos los derechos revertidos.”¹⁷

Linux

Orígenes

Entre 1984 y 1990, Stallman hizo progresos sustanciales al construir el sistema operativo GNU. El y otros voluntarios trabajaron miles de horas en el código y crearon una larga obra de herramientas compatibles con Unix sin propiedad. Pero al final de 1990, nadie del proyecto GNU había trabajado en desarrollar un elemento crucial del software- el núcleo. El núcleo es la parte del sistema operativo que se responsabiliza de ejecutar varias tareas (ejm., tareas como enviar información a imprimir, desplegar información en pantalla, obtener datos de un sitio Web, etc.). Para Stallman, el retraso en proporcionar el núcleo no fue suficientemente serio como para socavar su proyecto; era mas importante para él crear exitosamente la mayoría de los elementos, y “que la pieza faltante del rompecabezas estaba en camino.”¹⁸

Pero Linus Trovalds, un estudiante de segundo grado de la ciencia computacional en la Universidad de Helsinki y un pirata informático autodidacta, no quiso esperar para proveer al proyecto GNU el núcleo que le permitiera arrancar Unix en su PC basado

¹⁶ DiBona et al., p.59.

¹⁷ Sitio Web Recursos Abiertos, <http://www.openresources.com/documents/gnu-project/node9.html>.

¹⁸ Moody, p. 31.

en Intel. Así, en abril 1991, se propuso crear un sistema operativo completo con un núcleo para su propio uso. Decidió moldear su programa después de Minix, un “micro núcleo” creado por Andy Tanenbaum, quien era un respetado investigador y profesor de la ciencia computacional en la Universidad Vrije en los Países Bajos. Trovalds creyó que Minix (el cual fue utilizado como una herramienta educativa) podía ser enormemente mejorado. Así el domingo 25 de Agosto 1991, envió la siguiente nota al grupo de noticias en Internet de *comp.os.minix*:

Estoy realizando un sistema operativo libre (solo es pasatiempo, no será tan grande y profesional como gnu) para 386 (486) clones AT.¹⁹ Esto ha estado preparándose desde Abril, y empieza a quedar listo. Quisiera una retroalimentación de las cosas que gustan y disgustan a la gente en minix, algo parecido a mi sistema operativo... Tendré listo algo práctico dentro de pocos meses, quisiera conocer que características no les gustan a la gente. Todas las sugerencias son bienvenidas, pero no les prometo que las implementaré:)

La respuesta a esta introducción pública de Linux fue marcada. Programadores y futuros usuarios contactaron a Torvalds en gran cantidad por Internet para ofrecer ideas y sugerencias para el nuevo sistema operativo. Durante estos espacios del desarrollo, Trovalds y Tanenbaum entraron en un debate público acerca de los méritos de Minix y Linux. El profesor le hizo una sabia observación al trabajo de Torvalds en Enero de 1992: “Todavía sostengo el punto de que diseñar un núcleo monolítico en 1991 es un error fundamental. Agradece que no eres mi alumno. No obtendrás una Buena calificación por tal diseño : -).”²⁰

Monolítico o no, el núcleo que Torvalds desarrolló era auspiciosamente compatible con los componentes del GNU de Stallman. De hecho Torvalds integró su núcleo

¹⁹ “Clones 386 (486) AT” se refieren a computadoras personales basadas en microprocesadores Intel 80386 y 80486 que fueron compatibles con la PC de IBM/ especificaciones AT.

²⁰ DiBona et al., p. 225.

con componentes del proyecto GNU en 1992 y licenciado, como había sido el protocolo, bajo GPL.

Crecimiento

De combinar el núcleo Linux con el sistema GNU resultó un completo y libre sistema operativo que corrió en una computadora personal. Este desarrollo reenergizó el movimiento de software libre. Entre 1991 y 1998, Linux creció de 10,000 líneas de código utilizadas por una persona a 1.5 millones de líneas utilizadas por 7.5 millones de personas en todo el mundo. Se estimó que más de 10,000 programadores – muchos de ellos voluntarios – tendrían en sus manos revisar el código. Técnicamente, Torvalds había escrito menos del 5% del programa que llevaba su nombre.

A finales de los años 1990, Linux fue un sistema operativo estelar y altamente funcional. Los usuarios tenían un par de caminos diferentes para obtener el código. Ellos podían simplemente bajarlo del Internet libre de costo, o, al inicio de la mitad de los años 1990, ellos podían comprar Linux a través de un distribuidor como Red Hat, Inc. o Caldera Internacional por un pago de alrededor de \$50.00. El número de CD's embarcados a los clientes se creyó había crecido de 450,000 en 1996 a 750,000 en 1997.²¹

El anexo 5 muestra información comparada para Linux y otros sistemas operativos hasta finales de 1997, la información recientemente disponible. Información de los embarques en 1998 e información bajada no estaban disponibles, pero muchos observadores creían que las licencias de Linux estaban creciendo más rápido de lo que había crecido en 1997.

(El anexo muestra el fracaso de las unidades servidoras que utilizan Linux y otros sistemas operativos, y no el fracaso de las ventas realizadas a cada proveedor de

²¹ Como se reportó en el documento Halloween: Sitio Web Open Source, <http://www.opensource.org/halloween/halloween2.php#quote5>, accesado en Noviembre 11, 2002.

sistema operativo.) La estimación del comparativo de ventas se parecían en que el crecimiento y penetración era hacia la baja por tres razones: primero, los usuarios que bajaban el software gratuitamente no firmaban parte de las ventas. Segundo, Linux fue utilizado constantemente para el aprovechamiento efectivo del hardware, lo cual no representaba ventas para ningún vendedor, por lo tanto los números no muestran incremento en las ventas. Tercero, al igual cuando Linux se ubicó en el hardware nuevo, fue típicamente utilizado en sistemas que eran mas pequeños y menos caros que el promedio.

Algunos analistas creyeron que el fuerte funcionamiento de Linux y costo bajo era la llave que dirigía a su crecimiento y penetración en el mercado. Otros señalaron la habilidad de Linux como muy limitado a la medida de las necesidades del usuario, hablando de los ajustes en el código.

Por su parte, emprendedores de las firmas iniciadoras como Red Hat, Caldera, y Cygnus Soluciones, quienes estaban intentando vender los productos y servicios Linux, enfatizaron las ventajas del código abierto sobre el código binario. Bob Young, el fundador de Red Hat, compararon el comprar el código binario con la compra de un carro con el cofre sellado: “¿Compraría usted un carro que no pudiera revisar, ni quitar sus partes para encontrar las fallas, y repararlo en su interior? Eso es como comprar código binario. Usted está comprando un carro que está encerrado con el cofre sellado: no tendrá la posibilidad de cambiarlo ni modificarlo.”²²

Promoción

Los clientes podían ver las ventajas de utilizar Linux pero temían de la disponibilidad de aplicaciones que pudiesen ejecutar en el y la habilidad de otros vendedores y proveedores de servicio para dar soporte. La puesta en venta del código fuente del buscador del navegador de Internet en Marzo de 1998 sorprendió a algunos

²² Michael Kanellos, “Fundador de Red Hat: Microsoft no está listo”, CNET News.com (Junio 13, 2000).

analistas, pero fue visto como una importante señal de la aceptación del crecimiento de Linux y del desarrollo del modelo “fuente abierta”.²³ Era claro que las firmas estaban empezando a repensar como sus modelos de negocio podrían beneficiarse de Linux.

La puesta en venta del código fuente del navegador también guió a algunos líderes y activistas del software libre a repensar su actitud hacia la iniciativa comercial. El movimiento de los líderes decidió que necesitaban un plan de mercadeo. Ellos temían que el término e imagen asociados con “software libre” podría ser poco atractivo para las compañías que de alguna manera decidirán usar el sistema operativo. Uno de los líderes del movimiento “fuente abierta” explicó:

La gente no se pudo identificar con la palabra “software libre”- asustaba mucho. La gente necesitaba un término menos político y de menos confrontación. Nos estaba perjudicando. Un buen ejemplo es el término “software libre” que suena realmente a un buen idealista, greñudo pirata de software en Birkenstocks pero que asusta... Jay Random en Techio. Nosotros no adaptamos nuestro lenguaje a la audiencia.²⁴

En abril 1998, un grupo que incluía a Torvalds y otros miembros de mucho tiempo de la comunidad del software libre se reunieron para discutir el tema de la percepción del público, y fijaron el término “fuente abierta”. El nuevo nombre salió publicado en la prensa popular (ver anexo 1) . Reporteros y analistas de negocios podían captar el sentido de porque el código “libre” de “fuente abierta” podría actualmente ser superior al código cerrado y de propiedad desde el punto de vista de un usuario.

La penetración en el Mercado de Linux (ver anexo 5), combinado con el cambio de, percepciones y buena prensa, ayudaron a encender un debate furioso dentro de la industria de la computación. De primera mano, muchos en la industria creyeron que

²³ Sitio Web Netscape Network, <http://wp.netscape.com/newref/pr/newsrelease591.html>, accesado en Noviembre 11, 2002.

²⁴ O'Mahony.

Linux sería el sistema operativo líder para la computación de negocios a la par con Microsoft NT. Ken Jacobs, vicepresidente de la división de mercado para servidores en Oracle, era un semejante proponente. El acertó: “(Linux) es mas rápido que (Windows) NT, tiene mejor soporte y es mas barato.”²⁵

Este pensamiento fue producido por capitalistas arriesgados como William Gurley de Benchmark Capital:

A primera vista, el desarrollo “open source” parece un concepto absurdo: Obtener un fragmentado y distribuido grupo de programadores voluntarios para desarrollar y suprimir errores de software y esperar que trabajen en armonía hacia una meta común, a pesar de la realidad de que ellos nunca podrán encontrarse, nunca podrá hacer un diez por sus esfuerzos, y debe, por el camino, libremente distribuir el código libre que ellos crean. Toda vez que ese concepto será el movimiento más poderoso en el software.²⁶

Al mismo tiempo, Linux tuvo un gran número de oponentes que creían que el sistema operativo nunca cruzaría el vacío dentro de la corriente del uso en los negocios. En las palabras de John Rose, administrador general del grupo en Grupo Consultor de Compaq: “El soporte para Linux está creciendo, pero no va a desplazar la crítica misión Unix de Windows NT.”²⁷

Otras críticas enfatizaron sobre la inseguridad del soporte técnico, tal como lo expresó el especialista técnico en sistemas de Stillwater, Minnesota: “Linux no funcionará en la comunidad de los negocios hasta que un sitio central sea creado

²⁵ Barbara Darrow y Steven Burke, “Linux to Find Home in Small, Midsize Market”, *Computer Reseller News*, Noviembre 23, 1998, p. 225.

²⁶ J. William Gurley, “Why free code makes sense”, *Fortune*, Agosto 2, 1999, p. 228.

²⁷ Darroe y Burke, p. 225.

para el soporte técnico. Ningún negocio en este país estará esperando a que un hippie de 17 años (conteste a las noticias del grupo y) resuelva sus problemas.”²⁸

Todavía otros señalan la falta de interfase gráfica del usuario en Linux (GUI), así como la naturaleza conservadora de los profesionales quienes desearán apostar sus trabajos al funcionamiento de un sistema operativo de fuente abierta.

En Julio de 1998, dirigiendo al vendedor de la base de datos, Oracle, Asociados en Computación, e Informix anunciaron que soportarían a Linux,²⁹ y en Octubre del mismo año, HP también anunció que algunas de sus soluciones de Internet estarían soportadas por Linux.³⁰ Durante todo el otoño de 1998, un gran momento se construía detrás de Linux, el software de fuente abierta, y el desarrollo del modelo de fuente abierta. Todos los días parecía que surgían nuevos sitios Web que analizaban y hablaban acerca del fenómeno Linux: *Linux hoy* y *Periódico Linux* atraían grandes números de lectores en adición al boletín que estilizaba sitios Web como Shashdot que fue favorecido por los programadores Linux.

Un documento de Microsoft que evaluaba el riesgo competitivo que representaba Linux fue mostrado y publicado sin intención en Internet cerca de Noviembre. Este reporte apodado por los miembros de la comunidad como el ‘Documento Halloween’, causó carcajadas en toda la comunidad de Linux y en la prensa. Los escépticos no estaban seguros si el reporte era o no verdaderamente un documento³¹ estratégicamente internamente de Microsoft. Pero si así hubiera sido, una cosa si era clara: Microsoft

²⁸ Paul McNamara y Christine Burns, “Linux Cynics,” *Network World*, Noviembre 9, 1998, p.176.

²⁹ Sitio Web de Red Hat, Inc., http://www.redhat.com/about/presscenter/1998/press_jul2398.html.

³⁰ Ibid.

³¹ Más tarde un vocero de Microsoft admitió que este documento había sido creado internamente para propósitos de investigación competitiva, pero que “no era una afirmación oficial hecha por Microsoft sobre el software de fuente abierta o el modelo Linux.”
<http://web.archive.org/web..om/ntserver/ntes/news/mwarv/linuxresp.asp>.

consideró que el método desarrollado de fuentes abiertas y Linux en particular era competitivo.

Postura de IBM hacia Linux

Dan Frye estaba intrigado por el momento de crecimiento que estaba generando la fuente abierta. Con todo el alboroto que esto ocasionó en la prensa y medios publicitarios, pareció que IBM estaría afrontando las implicaciones que traerían los movimientos del software libre, métodos de desarrollo para fuente abierta, y especialmente el extraordinario crecimiento y penetración en el mercado de Linux. La estrategia medular de IBM fue el proveer hardware, software, y servicios profesionales a sus clientes, quienes eran en primer nivel los departamentos de sistemas informáticos de corporaciones de tamaño medio y grande. Tales departamentos eran los administradores de la crítica misión acerca de las aplicaciones del software así como también de los flujos de información dentro de sus corporaciones. Estos departamentos eran incomprensiblemente riesgosos, y acudían a IBM para que les brindara una asistencia cercana el que no fracasaran los sistemas y asegurar un costo razonable.

Así mismo, con el boom del internet, el Mercado había cambiado. Los (proveedores del servicio de Internet), lanzamientos dot-com, y las universidades tenían diferentes prioridades a la de tener departamentos de mantenimiento de sistemas informáticos. Estos clientes cuidaban la velocidad, flexibilidad y la habilidad para innovar y escalar utilizando el internet. Para tener acceso a este mercado, IBM se vió en la necesidad de crecer sus negocios de software y middleware y hacer sobre una plataforma neutral, una que no dependiera de productos licenciados con Microsoft u otros competidores. Esto fue mas fácil decirlo que realizarlo.

Como muchos otros vendedores de hardware e integradores de sistemas, IBM mantuvo sus largos proyectos de desarrollo de software dirigidos a incrementar los

sistemas operativos que fueron instalados en sus computadoras. Mas tarde en 1998, esos sistemas operativos incluyeron Microsoft Windows NT y versiones IBM caseras de Unix, AIX [™]. IBM era uno de los grandes clientes de Microsoft y las dos empresas cooperaron de muchas maneras.

IBM también compitió con Microsoft en el mercado de aplicación de programas- ejm., programas que corrían por encima de diferentes sistemas operativos. En efecto, IBM fue el segundo vendedor de software mas grande en el mundo, con unas ventas en 1997 por casi \$13 billones (ver anexo 3). La primera compañía de software, Microsoft, patrocinó Windows NT, el cual era competidor de Linux. De esta manera, Microsoft no era la indicada para colocar sus programas de aplicación y artículos medios a Linux. En realidad, el divulgado documento 'Halloween' establecía que una estrategia disponible para Microsoft era extender o desarrollar nuevos protocolos para "negar al software libre la entrada a proyectos dentro del mercado".³²

De cualquier modo, la tendencia de crecimiento en la industria era hacer exactamente lo contrario- crear y sostener estándares abiertos. Estándares abiertos eran estándares técnicos que soportaban de manera independiente la funcionalidad computacional de una plataforma o aplicación partícula de un vendedor, y estaban disponibles para leerse por todos e implementarlos sin regalías u honorarios. Muchos en la industria creyeron que los estándares abiertos eran esenciales para la innovación en el Internet y para centrar trabajos de red computacional en general.³³ Así, dirigir los estándares abiertos era una llave elemental de la estrategia central para trabajos de redes.

De alguna manera, Linux fue el máximo estándar abierto: fue un sistema operativo que "soportó la funcionalidad de la computación independiente de una plataforma o

³² Sitio Web Open Source, <http://www.opensource.org/halloween/halloween1.php>.

³³ Sitio Web Bruce Pernees, <<http://www.perens.com/OpenStandards/Definition.html>>.

aplicación particular de un vendedor” y también estuvo “disponible para todos para leerse e implementarse sin regalías ni honorarios.” Pero Linux fue controlada por la comunidad desarrolladora Linux, una de las muchas comunidades del software libre que se había movido rápido hacia Internet.

Como una corporación, IBM tenía poca experiencia en tratar con las comunidades desarrolladoras del software libre. Había tenido un convenio productivo y amigable con el grupo Apache, pero, como ya se indicó, la comunidad Apache era diferente a la comunidad Linux. De esta manera, los ejecutivos de IBM temían de todos modos de cómo su compañía trabajaría con la comunidad desarrolladora Linux. Por una parte la comunidad estaba conformada de individuos. ¿Podría IBM, una gran corporación, entrar y jugar un papel en la comunidad? ¿Cómo le haría? ¿Podría entrar de manera que no llamase la atención? ¿Podría empatar en la comunidad?

Después de estas consideraciones prácticas, Frye y sus colegas temen al hecho de que IBM al unirse a la comunidad de Linux termine con su identidad de licenciataria como una libre y abierta alternativa para la computación corporativa; temen que una vez que IBM llegue, el status Linux como una empresa poco formal pueda disminuir y terminar con los piratas de software y el talento de la programación que habría llevado a Linux al éxito.

Potenciales ventajas y desventajas de Linux para IBM

Desde la perspectiva de IBM, Linux como sistema operativo ofrecía algunas ventajas potenciales. Primero, era una plataforma neutral, no apropiado por ninguna otra entidad corporativa. De esta manera, el software de IBM podría correr por encima de Linux sin crear una vulnerabilidad estratégica para ninguna corporación. Esto era consistente con la estrategia de IBM de soportar estándares abiertos.

Segundo, Linux parecía estar exhibiendo un extraordinariamente alto crecimiento y penetración en el mercado de PC y de los últimos servidores pequeños. Aparentemente, aquellos quienes lo utilizaron estaban muy satisfechos con su funcionamiento. En efecto, unos estudios formales y recientes de mercadeo habían posicionado a Linux igual que a Windows NT en términos de velocidad, y confiabilidad, aunque, como siempre, los resultados fueron discutidos y cuestionados.³⁴

Tercero, el sistema operativo era “libre”. De cualquier modo, “libre” era un nombre poco apto cuando se aplicaba para un gran programa de software: el código todavía tenía que ser soportado, y el soporte cuesta muchas veces menos que los costos de adquisición. Mas intrigante era la realidad de que el código fuente era “abierto”- esto es, visible por cualquiera y modificable por el licenciador sin violar los derechos de autor del licenciatario. En teoría, tal transparencia, junto con los derechos de cambiar el código base actual, podría llevar a un gran ahorro de costos y a una actualización mas fácil para los usuarios e integradores de sistemas de la misma manera. De cualquier modo, no había aún datos para soportar tal teoría.

Contrario a estas ventajas había un gran número de desventajas. Primero y la de mas importancia era la pregunta de la sustentabilidad a largo plazo de Linux. Aunque era exitoso en los mercados de los servidores de baja finalización, el desarrollo y soporte del software Linux aparecieron muy básicos. Los usuarios de Linux no podrían buscar a un distribuidor corporativo para recibir soporte y arreglar fallas; en vez de que tuvieran que confiar en una multitud de trabajadores quienes trabajan en Linux por diversión y gratis. (Claro, desde que el código fue abierto y modificable, los usuarios en principio tomar su propio staff en casa para arreglar fallas.)

³⁴ Sitio Web Mindcraft, Inc., <http://www.mindcraft.com/whitepapers/fist-nts4rhlinux.html>.

Segundo, IBM tendría que dirigirse a la pregunta del control. El sistema operativo de una computadora en una parte clave de su arquitectura y un punto de control. Las instrucciones (llamadas APIs o Interfaces de Programación de Aplicaciones) construidas en el sistema operativo son en esencia implantados en todos los programas de aplicación que corren en el sistema. El éxito económico fenomenal de Microsoft por mas de 20 años demostró lo valioso que era el control de un sistema operativo.

En el caso de Linux, el control del código base fue concedido por Linus Torvalds y un puño de tenientes confiables. Diferente del Proyecto Apache, el cual había gobernado a través de una corporación no lucrativa con consejo de directores elegidos por sus miembros, Linux arrancó como una dictadura virtual. Debates sobre la estructura del código y características podrían terminar con el mensaje del consejo, pero últimamente, Torvalds tenía la última palabra sobre lo que iba en cada característica de Linux. (Después de graduarse de la universidad, Torvalds había empezado a trabajar como diseñador de Transmeta Corporation, una base puesta en marcha de California. Como parte de su contrato, él gastaba algo de su tiempo manejando Linux.)

Así que aunque bajo la licencia GPL nadie podía modificar el código base para su propio uso, solo las personas que Linus o alguno de sus tenientes en los que confiaba su poder para “encomendar” el código para características oficiales de Linux. Ninguno de estos “encomendados” tenía conexión alguna con IBM. En efecto, algunos miembros del software libre y comunidades de la fuente abierta veían a IBM con gran antagonismo excedido solo por su hostilidad hacia la Corporación Microsoft.

Al mismo tiempo, Frye reconocía que la mayoría de los empleados de IBM eran dentro de lo mejor, ignorantes y los peores desconfiados del software libre. Las prácticas de desarrollo de la fuente abierta eran muy diferentes de lo tradicional en la práctica de la codificación, y antitético a la manera en que muchos de los mejores programadores de IBM estaban acostumbrados a hacer su trabajo. Como ejemplo,

la mayoría del código bajo propiedad era visto y revisado solo por unas cuantas personas con “accesos privilegiados” para partes específicas del código base. En contraste, los proyectos de fuente abierta se basaban en el principio de que todos podían ver todas las líneas del código. Literalmente nadie podía comentar de la calidad de cualquier recorte del código enviado, y las críticas llevadas a los miembros nuevos de la comunidad podrían ser severas e insultantes.

La última desventaja de Linux fue la incertidumbre que rodeaba el alcance y aplicabilidad de la Licencia Pública General. Como se indicó anteriormente, la licencia contenía una provisión recíproca: todos los trabajos derivados del código fuente de Linux deben correctamente proveerse abiertamente. (Ver anexo 4.) ¿Aplicaría GPL al código patentado escrito para correrse en Linux? Si así fuese, IBM tendría un momento difícil justificando el desarrollo de código patentado para correrse en Linux.

Otros reclamaron que este argumento era otro ejemplo del “temor”, inseguridad y duda” (FUD) sembrado por enemigos del software libre y fuente abierta. Fuente abierta defiende articuladamente que la GPL cubría solo modificaciones a un programa licenciado con GPL y no “infectó” otros programas que podrían rodearlo. Código licenciado GPL y patentado pudo ser ubicado en lo mismo del CD-ROM o máquina sin violar el GPL, pero no podían ser ‘mezclados’. Como y donde esta línea sería dibujada en la práctica no estaba claro, pero lo que si estaba claro era que el alcance de GPL nunca había sido puesta a prueba en la corte.

Alternativa cercana a Linux al año 1998

En 1998, había fundamentalmente cuatro posibles caminos para que IBM se acercase a Linux. Primero, IBM podía unirse a los oponentes de Linux, incluyendo Sun y Microsoft. Esto podía decretar que Linux no sería enviado en el hardware de IBM ni soportado por IBM. De cualquier modo, esta posición pareció empezar a

contar para el cometido de IBM de “poner el cliente primero”, porque algunos clientes prefirieron usar Linux, y estaban preguntando por el.

Segundo, IBM pudo adoptar una posición neutral, similar a la de Dell Computer Corporation. En este caso, si un cliente preguntaba por Linux, IBM, Dell, podría “ordenarlo y embarcarlo”.

Tercero, después de “colocar y embarcar” el sistema operativo, IBM podría soportar públicamente las metas del software libre y el movimiento fuente abierta, pero permanecer distante de las comunidades y sus prácticas de desarrollo. Esta opción implicaría un pequeño cambio interno para IBM.

Cuarto, IBM podría proactivamente buscar alguna forma de alianza con la comunidad desarrolladora Linux. Como se mencionó, Linux era un sistema operativo neutral. Todas las computadoras necesitan un sistema operativo. De cualquier forma, las ventas de IBM y utilidades se derivaban principalmente del hardware, middleware y servicios y no de los sistemas operativos. Así que ambos IBM y la comunidad de Linux saldrían discutiblemente beneficiadas por la disponibilidad de la alta calidad, fácil mantenimiento, prevaleciente y no patentado sistema operativo.

De cualquier modo, si IBM buscara una alianza con la comunidad Linux, no estaba claro como tal convenio podría cerrarse firmemente. Dejando de lado la posibilidad de desconfianza y absoluta hostilidad hacia IBM en el software libre y movimientos de fuente abierta, la comunidad de desarrolladores Linux era un grupo afiliado vagamente de gente que vive por todo el mundo. Se comunican vía consejos de mensajes y grupos de noticias en el Internet y rara vez se encuentran personalmente.

No obstante, había una organización, de personas, dentro de la comunidad desarrolladora Linux. (Ver anexo 6.) Linus Torvalds era todavía líder del proyecto: él manejaba y anunciaba todas las liberaciones. Algún aficionado lo llamaba

“dictador benevolente.”³⁵ En 1996 su primer teniente, Alan Cox (quien vivía en Gales), había creado un papel de ‘conservadores’: colaboradores experimentados que eran responsables de módulos particulares del núcleo y revisaban el código proporcionado por un gran cuerpo de desarrolladores acreditados.³⁶

Aproximadamente 200 desarrolladores acreditados fueron explícitamente reconocidos en el archivo de créditos el cual era desplegado como parte del programa.³⁷ Una gran población de aproximadamente 15,000 casuales u ocasionales colaboradores voluntarios reparan fallas, sugerencias de código, o ideas sobre la vida del proyecto.

A los colaboradores de Linux les gustaba llamar a su organización meritocracia. Con el tiempo, desarrolladores que contribuyeron más al código, y a la alta calidad del código, fueron de gran respeto y de grandes proyectos de responsabilidades. Así, un colaborador ocasional podría ser un desarrollador acreditado y eventualmente un conservador si el grupo consideraba sus habilidades y contribuciones altamente suficientes. De cualquier modo, la estructura del proyecto estaba todavía ligera, y era difícil decir como las decisiones tenían que tomarse.

En teoría, un representante de IBM podía intentar el contactar a Torvalds (o a alguno de sus tenientes) y proponer una potencial alianza. Un problema con esta idea era que había algunos foros privados a prueba de agua con tal propuesta. Los programadores que colaboraron a que Linux trabajara en público y fueron conocidos ser fuertemente obstinados. Segundo, no quedaba claro cuando Linus, u otro pirata informático, valoraría lo que IBM tenía que ofrecer. ¿Cómo una oferta de una de las compañías mas grandes en la industria de la computación hasta la fecha, sonaría lógico que estuviese concertada sobre la responsabilidad de ser descartado o

³⁵ Eric Raymond, *The Cathedral and the Bazaar* (Sebastopo, CA: O’Reilly Publications, 1999).

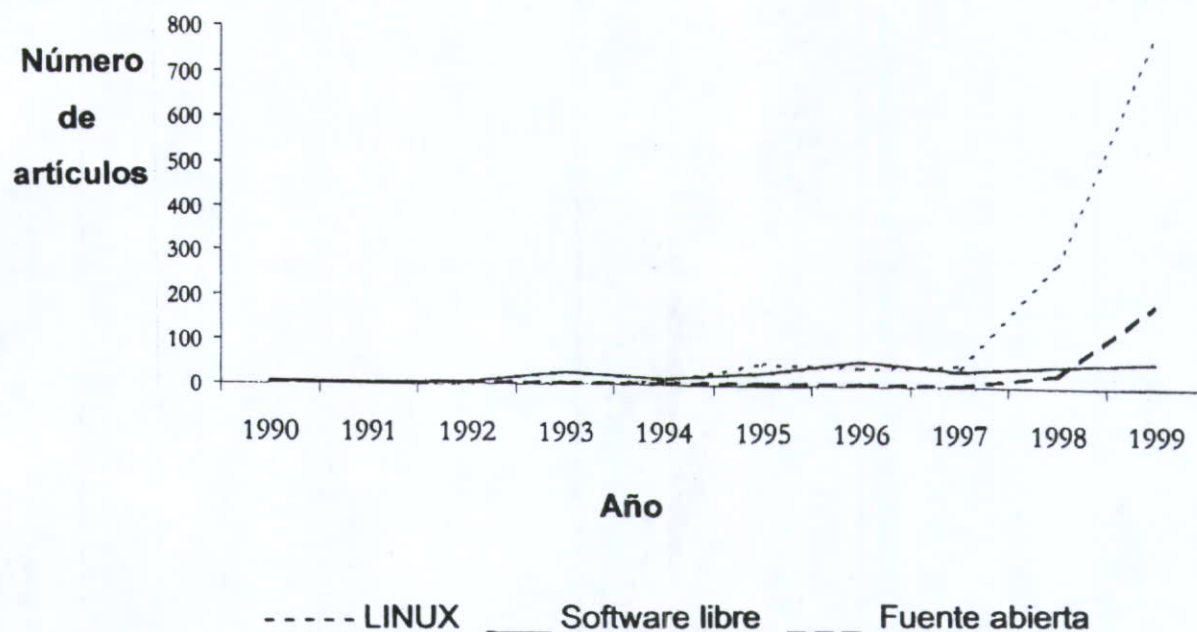
³⁶ Jae Yun Moon y Lee Sproull, “The Essence of Distributed Work: The Case of the Linux Kernel,” en *Distributed Work*, editado por Pamela J. Hinds y Sara Kiesler (Cambridge, MA: MIT Press, 2002).

absolutamente rechazado? ¿Cómo podría IBM comunicar la posibilidades y beneficios de una alianza? ¿Era la alianza la estrategia correcta?

Frye maravillado acerca de cuales serán los siguientes pasos de IBM. Y, él pensó, si IBM tomara una estancia proactiva, ¿quería él estar personalmente involucrado en el proceso?

³⁷ Moon y Sproull.

Anexo 1 Mención de Linux, Software libre, o Fuente abierta en la prensa 1990-1998



Fuente: Adaptado según los artículos citados en Inform/Proquest utilizando las palabras "Software Libre", "Linux", o "Fuente abierta" como tema, abstracto o palabra clave.

Anexo 2 Datos seleccionados de la empresa IBM por los años 1990-1997 (en Millones de dólares estadounidenses, excepto las cantidades por acción)

POR EL EJERCICIO	1997	1996	1995	1994	1993	1992	1991	1990
Ingreso	\$78,508	\$75,947	\$71,940	\$64,052	\$62,716	\$64,523	\$64,766	\$68,931
Utilidad neta (pérdida)	6,093	5,429	4,178	3,021	-8,101	-4,965	-2,861	5,967
Por acción del capital social	6.18	5.12	3.61	2.51	-7.11	-8.7	-5.01	10.42
Dividendos pagados en efectivo sobre el capita	763	686	572	585	905	2,765	2,771	2,774
Por acción del capital social	0.775	0.65	0.5	0.5	0.79	4.84	4.84	4.84
Inversión en planta, renta de maquinaria y otras propiedades	6,793	5,883	4,744	3,078	3,232	4,698	6,502	6,548
Retorno sobre el capital social	29.7%	24.8%	18.5%	14.3%	-	-	-	14.8%
AL FINAL DEL EJERCICIO								
Total activo	\$81,499	\$81,132	\$80,292	\$81,091	\$81,113	\$86,705	\$92,473	\$87,568
Inversión neta en planta, renta de equipo y otras propiedades	18,347	17,407	16,579	16,664	17,521	21,595	27,578	27,241
Capital de trabajo	6,911	6,695	9,043	12,112	6,052	2,955	7,018	13,313
Total deuda	26,926	22,829	21,629	22,118	27,342	29,320	26,947	19,545
Capital contable	19,816	21,628	22,423	23,413	19,738	27,624	36,679	42,553

Fuente: Adaptado de los reportes anuales de IBM de 1994 y 1997.

Anexo 3a Análisis de los ingresos por segmento de la empresa IBM por los años 1995-1997 (cifras en millones de dólares estadounidenses)

	MUNDIAL		SOLO EN EE.UU.AA.	
	1997	1996	1997	1996
Cientes	\$13,915	\$13,925	\$5,804	\$5,519
% del total	18%	18%	18%	18%
Servidores	11,868	12,421	4535	4365
% del total	15%	16%	14%	15%
Periféricos:				
Almacenamiento	2,725	2,779	1131	1036
Otros periféricos	2,126	2,304	781	860
% del total	6%	7%	6%	6%
Hardware OEM	5,590	4,550	3,848	3,092
% del total	7%	6%	12%	11%
Total Hardware	36,224	35,979	16,099	14,872
% del total	46%	47%	49%	51%
Servicios	19,302	15,873	7980	6129
% del total	25%	21%	24%	21%
Software	12,844	13,052	4569	4377
% del total	16%	17%	14%	15%
Mantenimiento	6,402	6,981	2461	2525
% del total	8%	9%	8%	9%
Financiamiento y otro	3,736	4,062	1554	1492
% del total	5%	5%	5%	5%
Total ingresos	\$78,508	\$75,647	\$32,663	\$29,395
				\$26,789

Fuente: Adaptado de IBM 1997 10-K. Los porcentajes acumulados pueden variar del 100%, debido a redondeo.

Anexo 3b Información de segmentos de IBM

IBM esta dentro de los negocios como empresa que provee soluciones a sus clientes hacia el uso de tecnologías avanzadas. La compañía opera principalmente en el segmento industrial que crea valor agregado ofreciendo una variedad de soluciones que incluye, tanto en forma individual como en combinación, servicios, software, sistemas, productos, financiamiento y tecnologías.

Para propósitos de clasificar información similar de productos con tecnología, el propósito general de los sistemas de computación que operan en un gran grupo de aplicaciones están clasificados como servidores cuando los sistemas son simultáneamente utilizados por múltiples usuarios al mismo tiempo, o como clientes cuando los sistemas son utilizados por un usuario al mismo tiempo. Los clientes incluyen computadora personal, y productos RS/6000, terminales de uso general basados en despliegues y monitores, consumidor y sistemas de consumo y financieros. Los servidores incluyen el sistema 390, AS/400, RS/6000 y productos de cómputo personal y servidor. El almacenamiento se conforma de dispositivos externos de almacenamiento adjuntos y cintas de almacenamiento. Otros equipos periféricos consisten de impresoras de funciones avanzadas y dispositivos de telecomunicación. El hardware OEM se conforma del ingreso por las ventas de archivos de almacenamiento de HDD y semiconductores.

Estos tipos de producto en hardware representan grupos que desarrollan funciones similares, lo opuesto al espectro de productos asociados con las divisiones de producto de IBM. Por consiguiente, ellos no representan un rango lleno de cualquier división de ofertas, las cuales podrían incluir equipo periférico relacionado, software y mantenimiento.

Los servicios representan un alto grado de soluciones en servicios de redes, los cuales incluyen el manejo de operaciones y servicios de redes; servicios profesionales, consistentes en el manejo de sistemas y subcontratos, diseño y

desarrollo de integración de sistemas, educación y consultoría; servicios de soporte a los productos, consistentes en la disponibilidad de servicios para el soporte de la operación y recuperación de sistemas de negocios. El mantenimiento consiste en facturar por separado los cargos por mantenimiento. Financiamiento y otros se conforman principalmente de financiar ventas, productos no clasificados.

Lógicamente algunos productos encajan en mas de una clase y son asignados a tipos específicos basados en una variedad de factores. Adicionalmente, los productos tienden a imponerse, fusionarse o desintegrarse de clases ya existentes como un resultado de los cambios en tecnologías, percepciones del mercado y / o uso del cliente. Por ejemplo, la demanda del mercado puede crear requerimientos por avances en tecnología que permitan el que un producto periférico sea funcional integrado con un despliegue, un dispositivo de telecomunicación y un procesador que forme parte de una estación de trabajo. Tal intercambio y progreso tecnológico tiende a realizar año con año comparaciones menos válidas de lo que serían en una industria menos sujeta a un cambio rápido.

Anexo 4 Sección 2 de la Licencia Pública General

2. Se podrá modificar copia o copias del Programa o cualquier parte de el, de esta manera se formará un trabajo basado en el programa, copiar y distribuir tales modificaciones o trabajo bajo los términos de la sección I, mencionando que también deberán cumplirse estas condiciones:

a) Deberán originarse los archivos modificados que muestren noticias notables estableciendo que se cambiaron los archivos y la fecha de cualquier cambio.

b) Deberá originarse cualquier tipo de trabajo que se distribuya o publique, que en su todo o en parte contenga o sea derivado del Programa o parte de este, siempre y cuando sea patentado o licenciado como un todo y sin cargo para las terceras partes bajo los términos de esta licencia.

c) Si el programa modificado normalmente lee comandos interactivamente cuando se ejecuta, podrá ser elaborado, cuando empiece a ejecutarse para tal uso interactivo de la manera mas ordinaria, para imprimir o desplegar un anuncio incluyendo un aviso apropiado de derechos reservados y un aviso de que no hay garantía (o algo, diciendo que se proveyó la garantía) y los usuarios podrán redistribuir el programa bajo estas condiciones, y diciéndole al usuario como ver una copia de esta licencia. (Excepción: si el programa en si mismo es interactivo pero normalmente no imprime tal aviso, su trabajo basado en el programa no requerirá imprimir un aviso).

Estos requerimientos aplican al trabajo modificado en su todo. Si hay secciones identificables que no se deriven del programa, y sean razonablemente consideradas como independientes y trabajos individuales en ellos mismos, entonces esta licencia, y sus términos, no aplicarán a esas secciones cuando se distribuyan como trabajos independientes. Pero cuando se distribuyan las mismas secciones como parte de un todo cuyo trabajo fue basado en el programa, la distribución del

todo deberá ser bajo los términos de esta licencia, cuyos permisos para otras licencias se extienden al todo, y así mismo a cada parte y todas las partes referentes a quien las escribió.

Así mismo, no es la intención de esta sección el reclamar derechos o impugnar sus derechos para trabajar por escrito por usted; mejor dicho, el sentido es ejercer el derecho a controlar la distribución de trabajos derivados o colectivos basados en el programa.

Adicionalmente, la incorporación de otro trabajo no basado en el programa con el programa (o con el trabajo basado en el programa) sobre un volumen de almacenaje o distribución media no proporcionará a otro el trabajar bajo el alcance de esta licencia.

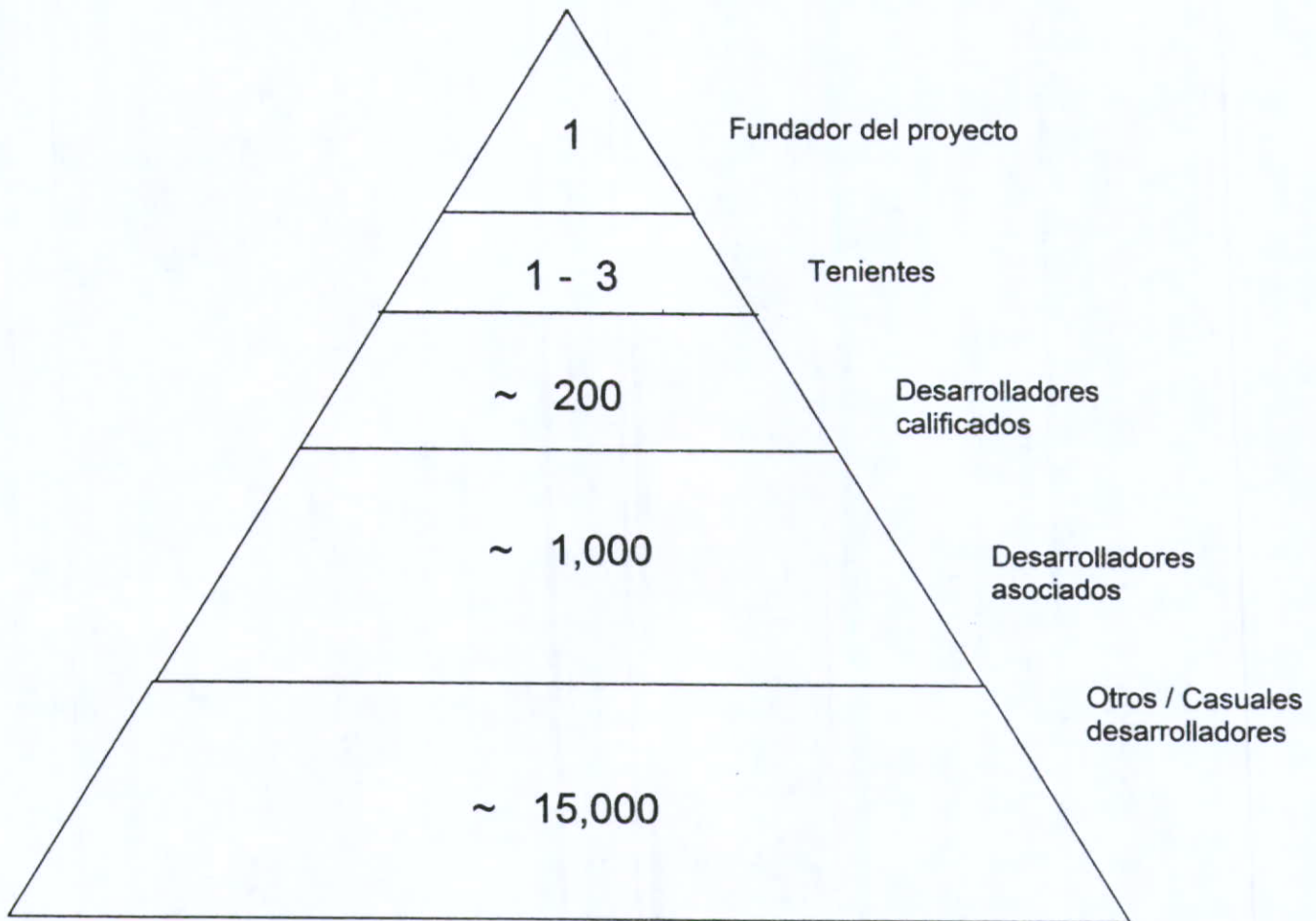
Anexo 5 Penetración de Linux en el mercado de servidores comparado con otros sistemas operativos

SISTEMA OPERATIVO PARA SERVIDOR		1996	1997	Crecimiento
NTS	unidades	805,200	1,505,000	86.91%
	%	23.30%	36.60%	
Netware	unidades	993,000	900,000	-9.37%
	%	28.73%	21.89%	
Linux	unidades	200,000	240,000	20.00%
	%	5.79%	5.84%	
Solaris	unidades	81,000	99,500	22.84%
	%	2.34%	2.42%	
SCO	unidades	226,000	288,000	27.43%
	%	6.54%	7.00%	
Other	unidades	1,150,537	1,079,478	-6.18%
	%	33.29%	26.25%	
Totales		3,455,737	4,111,978	18.99%

Clave: NTS se refiere al sistema operativo para servidor Windows NT de la empresa Microsoft Corporation; Netware producido por Novell Corporation; Solaris por Sun Microsystems; y SCO por la Operación Santa Cruz.

Fuente: Adaptado del sitio Web del Open Source, <www.opensource.org/halloween/halloween2.php#_Toc427383650>, accesado en Noviembre 1, 2002

Anexo 6 Organización de la Comunidad Desarrolladora Linux en 1998



Fundador del proyecto – Linus Torvalds.

Tenientes – Personas que dan mantenimiento clave al núcleo de Linux, (quienes revisan reportes de fallas y reparan fallas) envían códigos que contienen arreglos a la lista de correos y a Torvalds directamente.

Desarrolladores calificados—Muchos fueron desarrolladores acreditados quienes habían estado publicados en la lista para su contribución al desarrollo del núcleo Linux.

Desarrolladores asociados—Divididos en dos grupos: "Equipo de desarrollo" crean pequeños arreglos en el código, agregan mejoras, reparan fallas; "Equipo de reporte de fallas" identifican fallas, catalogan y eliminan fallas.

Casuales/ Otros desarrolladores—Son miembros de medio tiempo en la comunidad, localizados en 90 países en todo el mundo, contribuyen con comentarios, documentación, trabajo de arte, y pequeños arreglos de código, ocasionalmente para aplicaciones.

Fuentes: Pamela Hinds y Sara Kiesler, *Distributed Work* (Cambridge, MA: MIT Press, 2002), pp. 393-396. Soibhan O'Mahony, "The Emergence of a New Commercial Actor: Community Managed Software Projects" (Ph.D. diss., Stanford University, 2002)

2. RESUMEN DE LOS HECHOS

Daniel Frye miembro del equipo de tecnología emergente de IBM el 03 de Noviembre 1998 estaba preparando para el director de IBM la presentación de una propuesta estratégica de riesgos y oportunidades de una alianza entre IBM y la Comunidad Desarrolladora de Linux. Daniel Frye tenía estudios de doctorado en Física Teórica .

Linux era un sistema operativo libre ejecutable en procesadores Intel y servidores que no estaba patentado por alguna empresa computacional.

En 1996 IBM experimentó su primer lanzamiento de software libre con el software Apache el cual fue el programa de servidor Web utilizado en el sitio o pagina Web de los Juegos Olímpicos de Atlanta.

En el software libre el código utilizado es diferente al código binario pudiendo ser leído e interpretado por cualquier persona teniendo el don de adaptabilidad a las necesidades del cliente. Este código no tenía costos de licencia pero no había distribuidores que garantizaran su funcionamiento.

Linus Torvalds fue el creador del núcleo de Linux en 1991 mientras estudiaba en la Universidad de Helsinki en Finlandia.

En 1998 el sistema operativo Linux fue soportado y desarrollado por la comunidad desarrolladora de Linux. Esta comunidad la integraban programadores voluntarios de todo el mundo. La empresa Red Hat empezó a ofrecer contratos para brindar servicios de soporte en Diciembre de 1997. No había entidad que garantizara legalmente la calidad del núcleo de Linux.

En otoño de 1998 se dió un gran punto de atención hacia el software libre y Linux.

IBM líder en fabricación de computadoras en la década de 1970, inventó la PC en 1981, reportó pérdidas totales en 1991 y 1992 de \$7.8 billones de dólares; pasaba por momentos muy difíciles.

En 1993 tomó el puesto de CEO para IBM Luis Gerstner quien se formó como CEO en Nabisco y quien a su vez se refería hacia IBM como una empresa con demasiada burocracia, preocupada por su propia manera de ver el mundo, no se preocupaba por sus clientes y con una manera lenta para lanzar productos al mercado. Estableció cuatro metas para IBM : ser rentable, ser mas competitivo, incrementar el valor de la acción y crecer. Recortó inmediatamente la fuerza laboral por 45,000 es decir un 15% y recortó gastos por \$2.8 billones de dólares. Transmitió al personal de IBM que se obsesionarán por la satisfacción del cliente. En los siguientes cinco años regresó IBM a la rentabilidad y logrando un moderado crecimiento. IBM peleó entrar al mercado del servidor y emprendió varias iniciativas para estabilizarse, entre ellas la adquisición de Lotus. Gerstner posicionó a IBM para crecer en la computación de redes centrales.

ESTRATEGIA
Estrategia

A pesar de que IBM ya había trabajado con Apache, Linux representaba algo nuevo de emprenderse.

Richard Stallman creador del movimiento de software libre a mediados de 1980, siendo estudiante novato de Harvard era pirata de software en el laboratorio artificial de MIT y se había iniciado en la inteligencia humorística haciendo como práctica normal el compartir el código del software. Para 1975 había encabezado el desarrollo del editor de texto más poderoso de la época.

Finales de 1970 e inicios de 1980 el software compartido empezó a perder terreno por las corporaciones que deseaban hacer negocio con el software patentado.

Piratas de software con quienes Stallman trabajó fueron contratados por esta industria del software patentado y los distribuidores empezaron a poner restricciones para compartir sus códigos.

El movimiento del software libre nació cuando Stallman al estar en contra del software patentado, decidió diseñar un sistema operativo basado en Unix y disponible gratuitamente para todos, haciendo un llamado en 1984 a otros para unirse a su proyecto. Este sistema operativo fue nombrado GNU.

Stallman llamó al movimiento "software libre" porque invocaba a cuatro libertades que tenía este sistema para proteger lo que él llamo los derechos naturales de los usuarios siendo estos:

- 1) libertad para correr un programa para muchos propósitos,
- 2) libertad de modificar un programa y adecuarlo a las necesidades personales,
- 3) libertad para redistribuir copias gratis o a cambio de una cuota,
- 4) libertad para distribuir versiones modificadas de programas y que el usuario se viera beneficiado de las mejoras.

El código fuente del sistema operativo GNU no estaba protegido por ninguna licencia que impidiera la propiedad del software, ante esta preocupación, Stallman creó la Licencia Pública General GPL en la que permitió que el código fuente fuera modificado por el usuario y distribuido sin restricción alguna; siempre y cuando la creación fuese derivada del programa GNU.

Linus Torvalds creó el núcleo del sistema operativo GNU, basándose en la creación Minix "micro núcleo" de Andy Tatenbaum. Linus Torvalds el 25 Agosto 1991 envió una nota al grupo de noticias de Internet comp.os.minix y solicitó retroalimentación de las características que le gustaban y no le gustaban a la gente de Minix. Linus recibió respuestas a su llamado y creó el núcleo monolítico de GNU integrándolo en 1992 bajo la licencia de GPL, creándose así Linux.

Entre 1991 y 1998 Linux creció de 10,000 líneas de código utilizadas por una persona a 1.5 millones de líneas utilizadas por 7.5 millones de personas en todo el mundo. Torvalds escribió menos del 5% del programa que llevaba su nombre.

Para finales de 1990 Linux era ya un sistema operativo altamente funcional y se adquiría con el distribuidor Red Hat, Inc. o Caldera Internacional a cambio de un pago de \$50.00. El número de CD's embarcados había crecido de 450,000 en 1996 a 750,000 en 1997.

Analistas creyeron que el fuerte funcionamiento de Linux y costo bajo era la llave que dirigía a su crecimiento y penetración en el mercado, otros puntualizaron la habilidad de Linux como muy limitado a la medida de las necesidades del usuario, hablando de los ajustes en el código.

El nombre de software libre no ayudaba a la promoción de Linux puesto que la gente veía al software libre como algo aterradorante y en abril de 1998 Torvalds y un grupo de personas de la comunidad del software libre decidieron llamarlo "open source". Ken Jacobs vicepresidente de la división de mercadotecnia de Oracle afirmó que Linux era mas rápido que Windows NT, con mejor soporte y más barato.

Para William Gurley de Benchmark Capital "open source" se le hacía algo absurdo el hecho de que a pesar de que se formaba con grupo de programadores voluntarios que nunca trabajaban en equipo personalmente y aún así el concepto era el movimiento más poderoso del software.

John Rose, administrador general de Grupo Consultor de Compaq se oponía diciendo que el soporte para Linux no iba a desplazar la misión Unix de Windows NT. Otros enfatizaron sobre la inseguridad del soporte técnico de Linux como Stilwater, Minnessota, un especialista técnico en sistemas diciendo que Linux no funcionaría para el área de los negocios siempre y cuando no existiera un sitio central de soporte técnico.

En julio de 1998 Oracle, Asociados en Computación e Informix anunciaban que soportarían a Linux y en octubre del mismo año HP informaba que algunas de sus soluciones de Internet estarían soportadas por Linux.

En noviembre Microsoft en su documento apodado por miembros de la comunidad del software libre como "Documento Halloween" reconocía que el método desarrollado de fuentes abiertas y Linux en particular era competitivo.

La estrategia de IBM era proveer hardware, software, servicios profesionales a los departamentos de sistemas de las empresas medianas y grandes. Estos departamentos de las empresas corrían con altos riesgos y acudían a IBM para que se les brindara asistencia personalizada para que no fracasaran sus sistemas, a un costo razonable.

Con el surgimiento del Internet, el mercado había cambiado y los proveedores del servicio Internet, lanzamientos dot-com, y las universidades tenían como prioridad la velocidad, flexibilidad, habilidad para innovar y escalar usando internet. Ante esto, IBM se vió en la necesidad de crecer sus negocios de software, middleware y hacer sobre una plataforma neutral una que no dependiera de productos licenciados con Microsoft u otros competidores.

IBM era uno de los clientes de Microsoft puesto que en 1998 IBM tuvo proyectos dirigidos al desarrollo de software para incrementar los sistemas operativos incluyendo versiones IBM caseras de Unix y Microsoft Windows NT.

IBM alcanzó en 1997 unas ventas por casi \$13 billones de dólares convirtiéndose en el vendedor número dos de software en el mundo; compitiendo así con Microsoft en el mercado de aplicación de programas. Microsoft patrocinó a Windows NT, competidor de Linux. Dirigir los estándares abiertos era la clave estratégica para trabajos de redes.

Linux era el máximo estándar abierto pues era un sistema operativo que soportó la funcionalidad de la computación independiente de una plataforma o aplicación particular de un distribuidor estando además libre para leerse por todos e implementarse sin pago de regalías.

Linux es una plataforma neutral, no apropiada por ninguna entidad corporativa. Pudiendo así correr el software de IBM por encima de Linux sin crear vulnerabilidad estratégica para ninguna de las dos empresas. Lo cual era consistente con el hecho de que IBM soportaría estándares abiertos.

Linux representaba un alto crecimiento y penetración en el mercado de PC y servidores al igual que Windows NT por su velocidad y confiabilidad.

El soporte técnico cuesta menos que los costos de adquisición. Todo esto porque el código es abierto y se pueden hacer modificaciones sin violar derechos de autor.

Los usuarios de Linux no podían buscar a un distribuidor para recibir soporte y arreglar fallas.

IBM no tenía el control del sistema operativo. El éxito económico de Microsoft por más de 20 años demostró lo valioso que era tener el control de un sistema operativo. Sólo Linus y sus encomendados tenían control sobre las características de Linux y estos no tenían conexión alguna con IBM. Los miembros de la comunidad del software libre veían a IBM como un gran rival y todo por la envidia excesiva hacia Microsoft.

Frye también reconocía que los empleados de IBM eran ignorantes y desconfiados del software libre.

La práctica de la codificación y desarrollo del código abierto eran contrarias a la manera como los programadores de IBM estaban acostumbrados a hacer su trabajo.

Se daba una gran incertidumbre alrededor por el alcance y aplicabilidad de la Licencia Pública General.

En 1998 había cuatro caminos para que IBM se acercase a Linux.

- 1) IBM podía unirse a los oponentes de Linux, incluyendo Sun y Microsoft. Lo cual indicaba que Linux no sería enviado en el hardware de IBM ni soportado por IBM. Pero esta posición contaba mucho para el cometido de IBM de poner al cliente primero porque algunos clientes prefirieron usar Linux, y estaban preguntando por el.
- 2) Después de colocar y embarcar el sistema operativo, IBM podría soportar públicamente las metas del software libre y el movimiento de fuente abierta pero permanecer distante a las comunidades y sus prácticas de desarrollo.
- 3) IBM podría buscar de manera proactiva alguna forma de alianza con la comunidad desarrolladora Linux. Linux era un sistema operativo neutral. Todas las computadoras necesitan un sistema operativo. De cualquier forma, las ventas de IBM y utilidades se derivaban principalmente del hardware, middleware y servicios y no de los sistemas operativos. Con esto IBM y Linux saldrían beneficiadas por la disponibilidad de la alta calidad, fácil mantenimiento, prevaleciente y no patentado sistema operativo.
- 4) Si IBM buscara una alianza con la comunidad Linux, no estaba claro como tal convenio podría cerrarse firmemente. Puesto que la comunidad de desarrolladores Linux era un grupo afiliado vagamente de gente por todo el mundo sin encontrarse personalmente en un punto y comunicándose vía mensajes.

Linus Torvalds era líder del proyecto, manejando y anunciando todas las liberaciones. En 1996 su primer teniente, Alan Cox (vivía en Gales), creó un papel de conservadores quienes eran colaboradores experimentados que eran

responsables de módulos particulares del núcleo y revisaban el código proporcionado por un gran cuerpo de desarrolladores acreditados.

Aproximadamente 200 desarrolladores acreditados fueron reconocidos y una población de aproximadamente 15,000 casuales colaboradores voluntarios reparan fallas, sugerencias de código o ideas sobre la vida del proyecto. Así un colaborador ocasional podría ser un desarrollador acreditado y eventualmente un conservador si el grupo consideraba sus habilidades y contribuciones altamente suficientes. La estructura del proyecto era todavía ligera y era difícil decir como las decisiones tenían que tomarse.

- 5) No quedaba claro cuando Linus u otro pirata informático valoraría lo que IBM tenía que ofrecer.

3. SÍNTESIS DE PROBLEMAS DETECTADOS

3.1 Análisis FODA (Fortalezas, Oportunidades, Debilidades, Amenazas)

3.1.1 Fortalezas

IBM es una empresa que ha sido capaz de enfrentar los cambios en la forma de hacer negocios y ha sabido manejar su apertura para innovar pues ha pasado por momentos difíciles en que ha perdido posición en el mercado y a pesar de esto sabe reposicionarse al mostrar disposición para el cambio.

El que IBM ya hubiese experimentado el trabajar con software libre como lo fue Apache, nos dice que el terreno no es tan desconocido para IBM.

El incremento en el valor de la acción de IBM la posiciona en mejores condiciones para entrar a satisfacer el mercado de software libre. Esto nos dice que IBM es capaz de generar utilidades y con esto llevar por buen camino su inversión del proyecto de software libre.

3.1.2 Oportunidades

El que IBM sea la mejor en servidores hace que Linux pueda tener un mayor acceso y aceptación dentro de las grandes corporaciones y empresas de una estructura mediana-alta pues son las que trabajan con servidores.

Linux es un proyecto de origen universitario y con voluntarios para realizar las mejoras lo cual permite que las empresas logren sus metas de reducción de costos en cuanto a sistemas informáticos se refiere.

El poder vencer a la competencia dando un mejor servicio al cliente por medio del soporte personalizado. Tiene IBM a su favor que el segmento de ingresos a nivel mundial que ocupaba el segundo lugar hasta 1997 eran los servicios por lo tanto si

ya venía aceptándose en el mercado como una buena opción de servicios computacionales, deberá seguir aprovechando este punto a su favor.

Linux es el sistema operativo que se adapta a las necesidades del cliente, haciendo mejoras a su medida.

93/12

El mercado cada vez mas se involucra con el e-business por medio de Internet, lo cual favorece a las ventas de IBM pues Linux tiene la bondad de ser veloz, factor de suma importancia en el e-business y uso de Internet.

Mejorar las estrategias de mercadotecnia para que Linux tenga cada vez más aceptación y que el mercado tenga mas confianza en el software libre.

IBM tiene experiencia en servicios profesionales brindados a los departamentos de sistemas en empresas medianas y grandes, lo cual lo hace fuerte para en un futuro brindar servicios de soporte Linux. Además de que fortalecería a uno de sus segmentos de ventas más importantes.

3.1.3 Debilidades

El hecho de que Linus Torvalds sea una persona que le interese más la libertad que el hecho de hacer negocio, lo cual hace muy difícil que IBM pueda convencer a la comunidad Linux de que se cierre una alianza duradera.

El que Linux no tenga una forma personalizada de dar soporte a sus usuarios.

Al ser Linux un sistema adecuado a las necesidades del cliente lo hace ser o muy abierto a las posibilidades de expansión en las empresas o muy limitado a lo que el cliente necesita.

Estructura tan débil o poco fuerte que tiene Linux pues dependen de Linus.

3.1.4 Amenazas

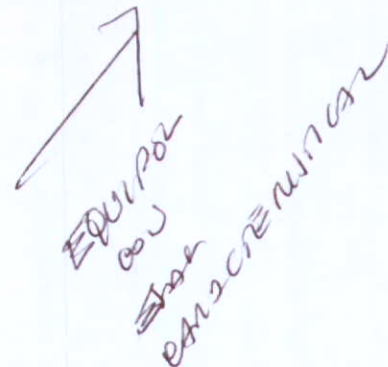
La sustentabilidad de Linux a largo plazo por la poca formalidad de los desarrolladores al no comprometerse personalmente con el cliente y el soporte. Pues el cliente lo toma como una falta de atención a sus necesidades y puede ser que con el paso del tiempo surja la competencia de Linux que sepa dar ese valor agregado en el servicio.

La competencia de Linux como lo es Microsoft puede impulsar un cambio en las leyes que regulen el licenciamiento de software libre a manera de que no permita que GPL funcione como actualmente lo hace.

La no conexión entre los desarrolladores Linux y programadores de IBM.

La no valoración de IBM por parte de Linus en términos de ver a IBM como de los mejores aliados para llegar exitosamente al mercado mundial.

El no poder hacer negocio con la patente de licencias; hará que IBM no tenga grandes ingresos por venta de licencias.



EQUIPO

COJ

Star

EAM2CREMINAR

3.2 Jerarquización de los problemas

1) Pérdida en el mercado de IBM en el segmento de ventas de software, pues a nivel mundial desde 1995 a 1997 ha bajado dos puntos porcentuales; contrario a la misión de IBM de permanecer en el mercado satisfaciendo lo que el cliente demande.

2) El que Linux sea un software libre que puede obtenerse por internet hace que se convierta en una adquisición bajo responsabilidad del mismo usuario lo cual es contraria a la misión de IBM en donde primero esta la satisfacción del cliente, objetivo que no se alcanzaría porque si Linux llega a fallar no hay indemnización al cliente por fallas y costos absorbidos por fallas importantes que afecten a las utilidades del usuario en sus negocios.

3) El no soporte personalizado lo cual pone en riesgo la satisfacción del cliente.

4) La forma de hacer la mercadotecnia con Linux no ha sido llevada por el mejor camino por parte de IBM, puesto que existe una no aceptación de Linux en las mentes de los usuarios de Microsoft al plantearse a Linux como libre en la programación del software.

5) Linux no es un sistema amigable para los usuarios pues no maneja gráficos que permitan al usuario tener más confianza en dicho sistema operativo. Es decir, si se tiene una falla, se solicita soporte por Internet y la solución no es como con Windows que se puede ir el usuario a un menú de opciones y elegir la que le resolverá el problema sino que deberá entrar a editar archivos escribiendo modificaciones en lenguaje máquina.

De parte de los usuarios u operadores del sistema exige que tengan un nivel de capacidad y entendimiento del lenguaje máquina pues es este lenguaje el que permite que Linux sea veloz y eficiente.

6) Cultura libre de los desarrolladores Linux y no dependencia de la comercialización de licencias. Esto hace que la visión de negocio de Linux sea informal. A Linus Torvalds y su comunidad no les gustará depender de IBM, puesto que actualmente tienen total independencia para crear e innovar.

7) Dificultad para hacerle ver al mercado que Linux es mejor y todo por el simple hecho de que el mercado ya está muy acostumbrado a lo cómodo que resulta manejar Windows y mas aún para aquellos usuarios poco involucrados en las innovaciones y avances en el uso de la computadora; que solo utilizan la computadora como una herramienta para sacar adelante sus trabajos.

8) La aceptación de los programadores de IBM de que deberán programar de otra manera muy distinta a la que estaban acostumbrados a venir programando y mas aún que sus creaciones en Linux pueden ser distribuidas sin poder cobrar regalías por su invención.

9) La no empatía entre el personal de IBM y los integrantes de la comunidad Linux. Se ven como rivales y no como complemento de sus negocios. La falta de disponibilidad de trabajo en equipo entre desarrolladores Linux y programadores IBM.

10) El monopolio que existe actualmente con Microsoft y que impide una libre competencia entre Linux y Microsoft.

4. SOLUCIÓN 1

ALIANZA IBM & LINUX

IBM muestra una capacidad financiera saludable para enfrentar una alianza con Linux pues IBM ha venido demostrando desde 1990 hasta 1997 que su inversión dedicada a los segmentos de venta como hardware, servidores, servicios, software y mantenimiento a sido fructuosa pues el retorno sobre el capital social ha alcanzado un nivel del 29.7% en el año 1997. Además de que en alianza con Linux, le permite abrirse las puertas a IBM para satisfacer el mercado e-business.¹

Para IBM es buena opción el entrar al mercado con Linux pues este sistema operativo tiene la bondad de dar velocidad a los servidores de IBM y por lo tanto un mayor aprovechamiento del hardware; productos que representan los mejores segmentos de venta para IBM.

5444

Un dato muy importante que sirve para soportar una alianza entre IBM y Linux es que "hoy, el 31% de las aplicaciones Linux que figuran en el directorio están destinadas al sector de los servicios financieros, el 28% al espacio minorista y el 23% a los organismos gubernamentales"².

La parte donde tiene IBM que invertir más fuerte sería en la capacitación de todos sus programadores para estar en la misma sintonía con la programación abierta o libre. En este tema, sí hay programadores muy conservadores que no quisieran adaptarse a este cambio, la mejor solución es que altos directivos de IBM convencan a sus programadores de los beneficios que tiene el trabajar con Linux y los programadores que no quieran entrar en este cambio, IBM deberá tomar la decisión de liquidar al personal no dispuesto a enfrentar los esfuerzos que implica este cambio en programación.

¹ Página Web de IBM Noticias "Los desarrolladores aceleran la adopción de Linux a una velocidad sin precedentes" 33,000 desarrolladores actualmente construyen aplicaciones basadas en Linux para Software IBM. México, D.F. 9 de Enero 2003 <<http://www.ibm.com/new/mx/2003/01/14-01-03b.html>>

² Idem anterior

IBM puede, como opción final después de descartar a sus programadores no dispuestos para el cambio a Linux, contratar a programadores integrantes de la comunidad desarrolladora Linux, pues siempre habrá personas que necesiten percibir un mejor sueldo.

Las estrategias de mercadotecnia sin duda deben de cambiarse por IBM y deberán ser dirigidas a:

a) Vender el software como la opción más segura contra ataques de virus y con mas ahorro en costos ³ (por adquisición de licencias) dándole a Linux mas aceptación en el mercado pues en los últimos tiempos se han propagado tipos de virus muy dañinos para el manejo y seguridad de la información.⁴

Un ejemplo de que Linux ofrece seguridad es que la fuerza armada y marina de EUA utilizan este sistema operativo. ⁵

b) Trabajar en el desarrollo gráfico de Linux, para que sea mas amigable para los usuarios y así tenga Linux mayor aceptación en el mercado.

Una negociación ganar-ganar entre IBM-Linux y ser capaces ambos de aceptar sus fortalezas que cada uno tiene y trabajar juntos para ofrecer seguridad en el soporte a

³Página Web de Red Hat Business and Industry Solutions "Open Source-Enabled Government" <<http://www.redhat.com/solutions/industries/government/>> Accesado el 31 Enero 2004.

⁴ Entrevista realizada a L.I.A. Jose Juan Reyes Mejía Area de Ventas de Grupo Aullox en Guadalajara, Jalisco. Realizada el 14 de Febrero 2004.

Página Web Enterprise Linux Today "Asia Loves Linux and Microsoft Scrambles" Bruce Einhorn, 6 Enero 2004. http://linuxtoday.com/it_management/2004010601826OSBZDP accesado el 31 Enero 2004.

⁵ Página Web de Titan National Security Solutions "Titan Awarded \$17 Million United Status Courts Information Technology Contract" San Diego, CA Diciembre 18, 2003 Accesado el 13 Febrero 2004. http://www.titan.com/investor/press-releases/press_releases_display.html?id=65&select=6

los usuarios, permitiría llegar mas a fondo del mercado, hasta la última necesidad del cliente. Con el tiempo la inteligencia y aceptación del cliente los llevará a adquirir Linux puesto que actualmente ya es una realidad y un hecho la aceptación que va teniendo dentro del mercado. Con esta solución las metas de IBM no se desviarían porque se llevaría el valor de la acción a un valor más elevado que en 1997.

La mejor manera de la alianza IBM-Linux para conquistar cada vez mas al mercado además de el costo, será el invertir en un esquema legal que permita garantizar a los usuarios el buen funcionamiento de Linux pues cuando compramos un producto lo menos que esperamos de nuestro proveedor es que nos responda por algún daño que nos haya ocasionado su producto. Esto puede realizarse, delimitando en un breve contrato de adquisición del sistema que casos pueden respaldarse a manera de garantía. En el mercado actual y tan globalizado no debemos de ser informales pues eso lleva a perder clientes.⁶

El ofrecer una garantía a los usuarios a cambio de un precio accesible y no habría pérdida para IBM puesto que a mayor volumen de garantías vendidas el precio sería accesible y así IBM crearía un nuevo segmento de ventas contribuyendo así a la utilidad de IBM y a su valor de la acción.

La alianza es la mejor manera en que IBM y Linux podrán enfrentar a la competencia mundial Microsoft o la mas aterradora y difícil de vencer, que no está de mas el considerar que pronto podría hacer su aparición como lo ha venido haciendo en todas las demás industrias, la competencia de Asia (China, Japón y Corea del Sur) (Ver anexo 1).

⁶ Página Web *Forbes.Com* "IBM Refuses to Indemnify Linux Users" Daniel Lyons, 05 Ago 2003
http://www.forbes.com/2003/08/05/cz_dl_0805ibmlinux.html

4.1 VENTAJAS

La alianza IBM-Linux es conveniente para IBM como para Linux ambas empresas incrementarían sus ventas.

Con el soporte a usuarios, el nombre de IBM en esta alianza, haría que el mercado tuviera mas confianza en la adquisición del software libre y atraería mayor número de ventas.

Gran intercambio de tecnología e innovación en el ramo de la programación para IBM y Linux.

Participación mas proactiva de IBM y Linux para ofrecer mejoras a la sociedad en tan valioso instrumento de eficacia como lo es la computadora.

La libertad que actualmente tiene Linux y a través de su creador Linus; permitirá que constantemente haya mejoras en Linux y con esto tener mas satisfecho al mercado.

Los usuarios recibirían un soporte más inmediato y cercano pues la misión de IBM es tener satisfecho al cliente.

Mayor fuerza en conjunto IBM-Linux para hacer frente a la competencia mundial del software patentado.

4.2 DESVENTAJAS

La participación en las utilidades por el segmento Linux para ambas empresas sería menor a que siguieran como ahorita, cada uno independiente en sus ventas.

El choque cultural entre programadores de IBM (conservadores en programación) y Linux (libres en programación), lo cual puede crear enfrentamientos y por consiguiente un mal desarrollo del software libre. Y ante esta situación estarían enfrentando pérdidas cuantiosas tanto Linux como IBM.

El costo de administrar al personal al enfrentar una alianza resulta elevado porque los esquemas de evaluación de personal e incentivos se cambiarían en gran parte; pues se estaría trabajando bajo un esquema más innovador.

Al cambiar el esquema de incentivos, premiando mas a las personas innovadoras; daría origen al fenómeno de que empleados de IBM ya no quisieran seguir trabajando para IBM y se irían con la competencia del software patentado.

El costo que de inicio representará el crear una nueva estrategia de mercado sería elevado pues el mercado no acepta del todo al software libre.

El número de integrantes que conforman la comunidad desarrolladora Linux es inmenso para los que IBM no tendría tantas plantas laborales a ofrecer.

El buen funcionamiento de una alianza depende de la integridad que se dé dentro de la organización entre los departamentos y aquí es donde IBM deberá hacer una gran inversión para que el proyecto no fracase y tenga consecuencias generando pérdidas financieras enormes por una mala comunicación e integridad inter-departamental.

5. SOLUCIÓN 2

CREACIÓN DEL SEGMENTO DE VENTAS IBM "SOPORTE LINUX"

La segunda solución viable es que IBM dentro de sus segmentos de venta agregue el segmento de vender soporte Linux pues es una opción de mercado bastante atractiva. Forbes señala que Microsoft hizo un estudio donde encontró que los sistemas Windows son mas baratos que Linux después de un período de 5 años en adelante; todo esto por lo caro que es obtener un soporte Linux actualmente, pues pocas empresas lo están ofreciendo, y como la oferta es muy baja respecto a la demanda de este tipo de servicio; esto lo hace muy caro.

Este estudio mencionado mostró que cinco compañías concluían que Windows es menos caro en cuatro o cinco veces. Todo esto se da por que Linux no tiene muchas aplicaciones disponibles para pequeños y grandes negocios; además de que los pequeños negocios no pueden invertir demasiado dinero en mejoras en los desarrollos.⁷

Es por esto que resulta atractivo para IBM enfocarse al soporte Linux como una opción de negocio mas y mejor posicionamiento financiero.

El número de usuarios que tiene Linux actualmente si amerita que IBM se enfoque al soporte de este; como ejemplo tenemos una estimación de que el 15% de todas las computadoras vendidas en oeste de Europa utilizan Linux.⁸

⁷ Página Web *Forbes.com* "The Limitations of Linux" Lisa DiCarlo, Junio 16, 2003 http://www.forbes.com/2003/06/16/cx_id_0616linux.html accesado el 31 Enero 2004

⁸ Página Web *Forbes.com* "German finance ministry division chooses Linux" Febrero 09, 2004 <http://www.forbes.com/newswire/2004/02/10tr1253037.html> accesado el 10 Febrero 2004.

5.2 VENTAJAS

En esta opción IBM no enfrentará los costos en que incurre una empresa a raíz de una alianza como lo serían: gastos legales, de operación, de capacitación y reestructura interna de IBM en cada uno de sus departamentos.

Las empresas de pequeña y gran talla podrían confiar mas en la adquisición del software Linux puesto que sabrían que IBM los apoyaría con soporte mas accesible en costo.

Al dedicarse IBM solo al soporte de Linux e independientemente de una alianza con Linux, IBM se estaría evitando arriesgar parte de su retorno de inversión y estaría establemente generando un ingreso extra por el segmento de "soporte Linux".

El que IBM dedique un segmento de ventas a ofrecer soporte Linux, le permitiría asegurarse de una manera muy cautelosa del buen funcionamiento de Linux y así poder en un futuro lanzar al mercado un software libre mas amigable de usarse por los usuarios y bajo bases diferentes a las de Linus Torvalds; pues no hay que perder de vista que IBM también tiene la gran fortaleza de innovar. Y con este probable lanzamiento IBM podría crear una licencia diferente a GPL que le permitiría permanecer también en el mercado de software patentado.

Con esta solución propuesta, IBM estaría aumentando su nivel de ventas dentro del rubro de servicios y 100% de la utilidad sería solo para IBM.

No representaría un cambio radical en la organización de la empresa sino que solo agregaría personal capacitado para programar en Linux y poder así dar una respuesta en tiempo y forma a los clientes que requieran de dicho soporte.

5.3 DESVENTAJAS

El costo mas representativo para IBM al dar soporte de Linux sería la capacitación de sus programadores en el manejo del código abierto que utiliza Linux.

Esta opción tiene la desventaja de ser poco abierta a las negociaciones mundiales que actualmente se están presenciando entre compañías del mismo sector; para hacer frente a la gran competencia que cada vez más acaba con las empresas.

Con esta solución propuesta en donde el papel de IBM sería muy independiente; estarían perdiendo tanto Linux como IBM; ya que en la actualidad el mercado es de las grandes alianzas comerciales.

Al mismo tiempo la competencia de Linux e IBM al ver que el negocio es dar soporte a los usuarios de Linux; saldrán al escenario empresas como Red Hat y SCO a enfrentar la batalla dando también soporte competitivo sobre Linux. Al suceder esto, el precio de soporte al mercado, estaría bajando y con esto el segmento de ventas de IBM por soporte Linux se verían disminuidas.

El crecimiento en las ventas de IBM por soporte Linux tendrían un crecimiento lento y no tan grande año con año pues tomará tiempo que las empresas y población mundial se convenzan y emigren a Linux; este cambio no sería de inmediato.

6. RESUMEN EJECUTIVO

Al principio

El problema raíz que enfrenta IBM y Linux es que ambas empresas tienen maneras muy diferentes de ver los negocios en el mundo actual. Mientras que IBM tiene muy claro que la fortaleza de negocios la hace el unirse a la competencia y adaptarse al cambio que se va dando en el mercado para no quedar desplazado o destinado a desaparecer como empresa; Linux goza de una libertad extrema que no le importa el factor económico a dimensiones muy elevadas; pues para Linus lo que es de vital importancia es apoyar la innovación.

La estrategia global que propongo es que cada empresa deberá aprovechar lo mejor que tiene para ofrecérselo al mercado y sobre todo destinarse a la conquista por medio de un trabajo en equipo donde se evite perder tiempo y dinero al verse exclusivamente como rivales. Se debe buscar la estratégica negociación ganar-ganar pues todos los humanos necesitamos del trabajo que demanda el mercado para poder subsistir.

*PAO
PUNTO*

La inversión que esto implica sería el intercambio de conocimientos, la adaptación al cambio por parte de IBM y Linux así como el dedicar tiempo al soporte del cliente siendo esta el arma que todo proveedor debe tener enfrente de su cliente pues es lo que hace mantener relaciones comerciales de largo plazo. La capacitación constante tanto en la nueva manera de hacer negocios como en la programación; es la mayor inversión que IBM y Linux deberán realizar para permanecer en el mercado. Adicionalmente la inversión en tiempo que deberá hacerse para cabildear el mejoramiento de la licencia GPL.

7. CONCLUSIONES

Es de gran aportación a la sociedad mundial el poner a su alcance este avance Linux pues nos lleva a mejorar cada vez mas el aprovechamiento de los sistemas computacionales que son una gran herramienta de trabajo y una manera de ser eficaz en el uso del tiempo para poder hacer mas en las empresas y destinar mas tiempo a la innovación en los negocios. Es muy desgastante el trabajar para una empresa donde el avance del trabajo depende de la ineficiencia de los sistemas computacionales o el no poder confiar 100% en ellos por las deficiencias de estructuras y no modificables por no tener acceso al código que los estructuró.

Actualmente enfrentamos una economía de mercado libre a nivel mundial, exigiendo por parte de las empresas competir en el mercado vía precio y respuesta oportuna a la demanda del mercado. No debemos perder de vista que para acceder al mercado vía precio se necesita reducir costos y sobre todo los costos fijos que es donde se ubica el costo de mantenimiento en software y equipo computacional.

Las empresas pueden competir ya sea reduciendo costos o disminuyendo su porcentaje de utilidad, pero el afectar esta última, no representa atractivo de negocio para los accionistas; por ello constantemente nos veremos en la constante batalla de reducir costos; reducir costos significa innovación en todos los departamentos de una organización.

Linux exige que la gente piense; siendo esto algo difícil de convencer en la gente de este mundo moderno y tecnológico. → CAMBIO CULTURA A NO

La introducción de Linux a una empresa exigirá por parte de sus empleados que sean multidisciplinarios y con esto quiero decir que deberán de involucrarse en el aprendizaje para manejar u operar Linux puesto que las respuestas a las fallas son consultadas vía internet con la comunidad desarrolladora del software libre.

TAÑO

El mundo está dejándose llevar por lo cómodo y lo que no exija al ser humano hacer el esfuerzo de pensar. El hombre quiere que todo se automatice lo cual hace que se pierda el valor de la humanidad y anteponiendo el consumismo desmedido a los valores humanos. El reconocer el pensamiento humano es aceptar que las mejores aportaciones de la humanidad serán las que se tomen en cuenta en la búsqueda de una mayor calidad de vida.

IBM y Linux son grandes fortalezas en su ramo de la tecnología de software que juntos harán una mejor aportación a la libertad de aportaciones para la humanidad y sus formas de relacionarse. IBM es una empresa altamente conservadora y permitirá a Linux regular las desviaciones de libertad que podrían ocasionar los desarrolladores (como la inseguridad mundial o terrorismo) que a través del código abierto podrían ocasionar un gran desastre mundial.

Interpretemos que IBM es el controlador de la libertad desenfrenada de Linux y que sería bueno para la humanidad que formen una alianza.

Linux será siempre la mejor opción (reducción de costos) en el desarrollo de sistemas por haberse originado como un proyecto de universidad donde los desarrolladores no tenían como prioridad el hacer negocio sino dar la libertad de innovación; contrario a Microsoft que siempre verá por hacer negocio con software patentado.

Al ahorrar costos, todos nos beneficiamos porque ese dinero ahorrado se puede destinar a otros proyectos prioritarios y que por falta de recurso dejan de emprenderse. Tal es el caso en la administración pública de todos los países que día a día están en constante austeridad para ofrecer mejores servicios a las poblaciones.

8. ANEXOS

Anexo 1

Accesando la Red

Por Eduardo L. Nisivoccia

Periódico El Mural

Lunes 2 de Febrero 2004

En el evento Linuxworld 2004, celebrado en Nueva York, no solamente hubo novedades, también estuvo presente la amenaza de SCO sobre esta comunidad que apoya el sistema Linux, de código abierto, que va ganando posiciones en el sector empresarial de todo el mundo.

En esta ya prolongada guerra, la empresa SCO busca convencer al público y principalmente a los jueces de que ellos son los únicos poseedores del código de Unix y que parte de ese código ha sido incluido indebidamente en Linux.

SCO ya ha demandado a IBM y a Novell (que acaba de adquirir SUSE Linux); al primero por violaciones de propiedad intelectual, y al segundo por afirmar que no había perdido sus propios derechos cuando vendió el código fuente de Unix a SCO.

Novell, Sun Microsystems, Hewlett-Packard y Red Hat ofrecen a sus usuarios una protección contra demandas o problemas derivados de los reclamos de derechos de propiedad violados hechos en el presente o futuro por SCO. IBM, que tiene un demanda por 3 mil millones de dólares por parte de SCO, informó que no aplicaría un programa de protección a sus clientes, ya que el pleito planteado "no tiene bases". Ampliando su oferta de productos Linux, IBM presentó un programa para enseñar y ayudar a las empresas para que puedan migrar más fácilmente de Windows NT hacia Linux.

Por su parte, las autoridades de Novell anunciaron que "2004 será el año que Linux capture el mercado de servidores en las empresas, y poco después los usuarios de PCs para negocios también comenzarán a migrar hacia Linux".

Todos los proveedores hicieron declaraciones similarmente optimistas sobre el futuro de Linux como alternativa viable a los sistemas operativos de Windows.

Pero tampoco SCO se quedó quieto, sino que hizo coincidir el evento con la liberación del contenido de una carta enviada al Congreso de los Estados Unidos, pidiendo que declare a Linux "un peligro a la seguridad y economía del país".

Los argumentos de SCO, que está aparentemente decidida a destruir Linux y desaparecerlo del mercado por las vías que sean posibles, establecen que "un experto en computadoras en Corea del Norte, que tiene un cierto número de PCs y una conexión a internet, puede bajar e instalar la última versión de Linux completo con capacidades de multiproceso y construir una súper computadora virtual burlando los controles de EU".

El creador de Linux, Linus Torvalds, declaró que estas afirmaciones carecían de sentido, ya que Estados Unidos establece controles de exportación al hardware, pero no al software.

Por otro lado, los países asiáticos no quieren quedarse fuera del alcance de Linux y están desarrollando una nueva versión de este sistema operativo que se llamará Asianux. El desarrollo está a cargo de Red Flag (un vendedor de Linux) y Miracle Linux (Oracle posee el 58.55 por ciento de esta empresa).

Los gobiernos de Japón, China y Corea del Sur están interesados en la promoción de Linux como una alternativa a los sistemas operativos de Microsoft, y en septiembre lanzaron un proyecto conjunto para el desarrollo de software de código abierto.

Para complicar más las cosas, salió el virus (gusano) MyDoom, con el aparente objetivo de realizar un ataque que paralice el sitio de SCO desde el 1 de febrero hasta el 12 de febrero SCO, considerándolo un ataque de los partidarios de Linux, ofreció una recompensa de 250 mil dólares a quien dé información que permita capturar al autor del virus.

9. BIBLIOGRAFIA

<http://www.uv.mx/iiesca/revista4/foda.htm>

“El FODA: Una técnica para el análisis de problemas en el contexto de la planeación en las organizaciones”

García López, Teresa; Cano Flores, Milagros

Accesado el 27 Ene 2004; 13pp

http://www.ibm.com/mx/press/news/2004/01/news_sw_07012004.phtml

“IBM lanza dos centros Linux en Latinoamérica”

IBM Press room – Software México; 07 Enero 2004

Accesado el 04 Feb 2004; 2pp.

<http://www.ibm.com/news/mx/2003/01/14-01-03b.html>

“Los desarrolladores aceleran la adopción de Linux a una velocidad sin precedentes”

IBM Noticias-Mexico; 14 Enero 2003

Accesado el 04 Feb 2004; 4pp.

<http://www.forbes.com/newswire/2004/02/10/rtr1253037.html>

“German finance ministry division coses Linux”

Forbes.com; 09 Feb 2004

Accesado el 10 Feb 2004; 4pp.

<http://www.redhat.com/solutions/industries/government/>

“Open source-Enabled Government”

Redhat.com|Government

Accesado el 31 Ene 2004; 2 pp.

[http://www.titan.com/investor/press-](http://www.titan.com/investor/press-releases/press_releases_display.html?id=65&select=6)

[releases/press_releases_display.html?id=65&select=6](http://www.titan.com/investor/press-releases/press_releases_display.html?id=65&select=6)

“Titan awarded \$17 Million United States Courts Information Technology Contract”

Titan National Security Solutions ; 13 Diciembre 2003
Accesado el 13 Feb 2004 ; 1pp.

http://linuxtoday.com/it_management/2004010601826OSBZDP
"Asia Loves Linux—And Microsoft Scrambles"

Enterprise Linux Today; Bruce Einhorn; 06 Ene 2004
Accesado el 31 Ene 2004; 5 pp.

<http://www.fcw.com/fcw/articles/2003/1208/web-nws-12-09-03.asp>
"Weather Service looks for Linux benefits"

Fcw.com; Rutrell Yasin; 09 Dic 2003
Accesado el 31 Ene 2004; 2 pp.

http://www.forbes.com/2003/06/20/cz_eb_0620linux.html?partner=yahoo&referrer
"The Feds Love Linux"

Forbes.com; Erika Brown; 20 Junio 2003
Accesado el 31 Enero 2004

http://www.forbes.com/2003/08/05/cz_dl_0805ibmlinux.html
"IBM refuses to indemnify Linux users"

Forbes.com; Daniel Lyons; 05 Ago 2003
Accesado el 31 Ene 2004; 4pp.

http://www.forbes.com/2003/06/17/cx_ld_0617linux.html
"Why you won't be getting a Linux PC"

Forbes.com; Lisa DiCarlo ; 17 Junio 2003
Accesado el 31 Ene 2004; 4pp.

http://www.forbes.com/2003/06/16/cx_ld_0616linux.html
"The limitations of Linux"

Forbes.com; Lisa DiCarlo ; 16 Junio 2003

Accesado el 31 Ene 2004; 4pp.

Entrevista realizada a L.I.A. José Juan Reyes Mejía
Area de Ventas en Grupo Aullox de Guadalajara, Jalisco.
Realizada el 14 de Febrero 2004

Llano Cifuentes, Carlos

"La enseñanza de la dirección"

IPADE; México; 1ª.ed; 1998; 129 pp.

L.Nisivoccia, Eduardo

"Linuxworld"

Periódico El Mural; pág. 2ª ; Secc. Interfase
02 de Febrero 2004.

Rodriguez Valencia, Joaquín.

"Cómo aplicar la planeación estratégica a la pequeña y mediana empresa"

ECAFSA; México; 4ª.ed; 2001; 319 pp.

Ansoff H., Igor.

"La Dirección Estratégica en la Práctica Empresarial"

Pearson; México; 2ª. ed; 1997; 548 pp.

Revista "Linux Free Magazine"

Año 1 Número 2

KP Ediciones; España; 31- Dic-2003

Agreement of 11/15/2004

For the purpose of this Agreement, the term "Agreement" shall mean the Agreement between the Parties dated 11/15/2004.

1.1.1. This Agreement shall be governed by the laws of the State of California.

1.1.2. This Agreement shall be binding on the Parties.

1.1.3. This Agreement shall be binding on the Parties.

1.1.4. This Agreement shall be binding on the Parties.

1.1.5. This Agreement shall be binding on the Parties.

1.1.6. This Agreement shall be binding on the Parties.

1.1.7. This Agreement shall be binding on the Parties.

1.1.8. This Agreement shall be binding on the Parties.

1.1.9. This Agreement shall be binding on the Parties.

1.1.10. This Agreement shall be binding on the Parties.

1.1.11. This Agreement shall be binding on the Parties.

1.1.12. This Agreement shall be binding on the Parties.

1.1.13. This Agreement shall be binding on the Parties.

1.1.14. This Agreement shall be binding on the Parties.

1.1.15. This Agreement shall be binding on the Parties.