

Microsoft Corporation

Linux OS Competitive Analysis

The Next Java VM?

Vinod Valloppillil (VinodV)

Josh Cohen (JoshCo)

Aug 11, 1998 – v1.00

Microsoft Confidential

**HIGHLY
CONFIDENTIAL**

**MSB 0078120
CONFIDENTIAL**

Plaintiff's Exhibit
6464
Comes V. Microsoft

MSB 00000078120

Table of Contents

TABLE OF CONTENTS	1
EXECUTIVE SUMMARY	3
LINUX HISTORY	3
WHAT IS IT?	3
HISTORY	4
ORGANIZATION	6
LINUX TECHNICAL ANALYSIS & OS STRUCTURE	6
ANATOMY OF A DISTRIBUTION	6
KERNEL - GPL	7
SYSTEM LIBRARIES & APPS - GNU GPL	8
DEVELOPMENT TOOLS (GPL)	9
GUI / UI	9
COMMERCIAL LINUX OS	10
BINARY COMPATIBILITY	10
REDHAT	10
CALDERA	11
OTHERS	12
COMMERCIAL LINUX ISV'S	12
MARKET SHARE	14
INSTALLED BASE	14
SERVER	15
CLIENT	15
DISTRIBUTOR MARKET SHARE	16
LINUX QUALITATIVE ASSESSMENT	16
INSTALLATION	17
UI	17
NETWORKING	18
APPS	19
PERCEIVED PERFORMANCE	19
CONCLUSIONS	20
LINUX COMPETITIVE ISSUES	20
CONSUMERS LOVE IT	20
LINUX VS. NT	21
LINUX VS. JAVA	22
LINUX VS. SUNOS/SOLARIS	23
LINUX ON THE SERVER	23
NETWORK INFRASTRUCTURE	23
ISP ADOPTION	24
THIN SERVERS	24
CASE STUDY: CISCO SYSTEMS, INC.	25

HIGHLY
CONFIDENTIAL

MSB 0078121
CONFIDENTIAL

LINUX ON THE CLIENT 25

 APP / GUI CHAOS..... 25

 UNIX DEVELOPERS..... 26

 NON-PC DEVICES 26

LINUX FORECASTS & FUTURES 26

 CURRENT INITIATIVES / LINUX FUTURES..... 26

 "PARITY GROWTH" 27

 STRENGTHS..... 27

 WEAKNESSES..... 28

 WORST CASE SCENARIOS..... 28

NEXT STEPS & MICROSOFT RESPONSE 30

 BEATING LINUX..... 30

 PROCESS VULNERABILITIES..... 32

REVISION HISTORY 32

MICROSOFT CONFIDENTIAL - PAGE 2

**HIGHLY
CONFIDENTIAL**

**MSB 0078122
CONFIDENTIAL**

Linux Operating System

The Next Java VM?

Executive Summary

The Linux OS is the highest visibility product of the Open Source Software (OSS) process. Linux represents a best-of-breed UNIX, that is trusted in mission critical applications, and - due to its open source code - has a long term credibility which exceeds many other competitive OS's.

Linux poses a significant near-term revenue threat to Windows NT Server in the commodity file, print and network services businesses. Linux's emphasis on serving the hacker and UNIX community alleviates the near-medium term potential for damage to the Windows client desktop.

In the worst case, Linux provides a mechanism for server OEMs to provide integrated, task-specific products and completely bypassing Microsoft revenues in this space.

[This document assumes that the reader has read the "Open Source Software" doc first. Many of the ideas / assertions here are derived from the previous doc and many other applicable Open Source arguments are not repeated here for brevity.]

Linux History

What is it?

Linux (pronounced "LYNN-uks") is the #1 market share Open Source OS on the Internet. Linux derives strongly from the 25+ years of lessons from the UNIX operating system.

Top-Level Features:

- Multi-user / Multi-threaded (kernel & user)
- Multi-platform (x86, Alpha, MIPS, PowerPC, SPARC, etc.), source compatibility
- Protected 32-bit memory space for apps; Virtual Memory support
- 64-bit support (platform dependent)
- SMP (Intel & Sun CPU's)
- Supports multiple file systems (FAT16, FAT32, NTFS, various UNIX)
- High performance networking
 - NFS/SMB/MPX/Appletalk networking
 - Fastest stack in Unix vs. Unix performance tests
- Disk Management
 - Striping, mirroring, RAID 0,1,5
- Xfree86 GUI

MICROSOFT CONFIDENTIAL - PAGE 3

HIGHLY
CONFIDENTIAL

MSB 0078123
CONFIDENTIAL

MSB 000000078123

History

An excellent piece on the history of the Linux Operaton system is provided by Wired Magazine at <http://www.wired.com/wired/5.08/linux.html>. I've paraphrased some of the key points below.

Linux was original the brainchild of Linus Torvalds, an undergraduate student at the University of Helsinki. In addition to a 80386-based kernel, Linus wrote keyboard and screen drivers to attach to PC hardware and provided this code under GNU's Public License on an FTP site in the summer of 1991.

After hosting his work on the FTP site, he announced it's availability on a Minix USENET discussion group in late summer 1991. By January of 1992, over 100 users / hackers had downloaded Linux and - more importantly - were regularly contributing / updating the source code with new fixes, device drivers, etc.

In contrast to the FSF/GNU work, which provided developers an open source abstraction above the underlying, commercial UNIX OS kernel, Linux's team was creating a completely open source kernel. In time, more and more of the GNU user/shell work was ported to Linux to round out the platform for hackers.

Forbes magazine's story on Linux has some excellent data on Linux's development history <http://www.forbes.com/forbes/98/0810/6209094s1.htm>:

Date	Users	Version	Size (LOC)
1991	1	0.01	10k
1992	1000	0.96	40k
1993	20,000	0.99	100k
1994	100,000	1.0	170k
1995	500,000	1.2	250k
1996	1.5M	2.0	400k
1997	3.5M	2.1	800k
1998	7.5M	2.1.110	1.5M

The LOC count appears to be inclusive of all Linux ports including x86, PPC, SPARC, etc.

Linux 1.0 - March 1994

Linux 1.0 was the first major release and led to the creation of "distributions." Prior to 1.0, linux existed as a piecemeal kernel with no centralized place to get a full working OS.

Major Features:

- Virtual Memory Management / memory Mapping / Buffer cache
- Job Control
- Device support for popular Network Cards, Hard Drives, CDROMs, etc
- Named Pipes, IPC
- Original EXTFS support instead of Minixfs
- Preemptive multitasking

MICROSOFT CONFIDENTIAL - PAGE 4

HIGHLY
CONFIDENTIAL

MSB 0078124
CONFIDENTIAL

Management Structure

After the release of version 1.0. The Linux developer community adopted a management structure to control what is added to the kernel with even numbered releases as stable, production release branches and odd numbered versions were "developer" branches.

While major areas of the kernel have "owners" which maintain their areas, Linus remains the final say on what does and does not go into the kernel. In large part, this structure remains in place.

It is important to distinguish that this management structure only controls the actual kernel and does not include supporting areas like the GUI, system utilities and servers, and system libraries.

Since 1.0, the following 1.x branches existed:

1.1 3/95
1.2 8/95
1.3 6/96

Version 1.3 evolved to become version 2.0

Linux 2.0 -- June 1996

Linux v2.0 was the first major release could effectively compete as a UNIX distribution. The kernel, system libraries, the GNU Unix tool, X11, various open source server applications such as BIND and sendmail, etc. were frozen and declared part of Linux 2.0.

Around the same time the GNU/FSF agreed, reluctantly, to make the Linux kernel the official kernel of the GNU operating system¹.

Some of the new base libraries and tools:

- Kernel modules 2.0.0 -- Basic kernel module support
- PPP daemon 2.2.0f - Dialup networking
- Dynamic linker (ld.so) 1.7.14 -- Shared libraries
- GNU CC 2.7.2 -- C compiler, tools, and debugger
- Binutils 2.6.0.14 -- Support for various binary executable formats
- Linux C Library Stable: 5.2.18,
- Linux C++ Library 2.7.1.4
- Termcap 2.0.8 -- Console mode terminal drivers
- Procps 1.01 -- ProcFS file system maps kernel objects to the filesystem
- SysVinit 2.64 -- A system V boot system, SYSV compliant named pipes.
- Net-tools 1.32-alpha- Basic Networking tools such as telnet, finger, etc
- Kbd 0.91 -- Console mode keyboard/scrollback/ virtual screens support

Subsequent Versions

The current 2.0.x stable version is 2.0.34, which was released in May 1998. Prior to this, 2.0.33 was released in Dec 1997. The current development branch is 2.1.108 (as of July 14, 1998).

¹ Prior to this, Stallman and Co. were developing the HURD kernel which languished in limbo

Process Slowdown

With the growth of the kernel, Linux's release frequency has slowed measurably. There is growing frustration about when 2.2, the next "stable release" version will ship. The sheer size of the codebase has begun to overrun the resources of Linus. There is a backlog of patches to be merged and often, Linus is becoming the choke point.

The current release tree, 2.0.x has iterated 34 versions in 2 years. The development branch, 2.1.x, which will eventually become 2.2 has been going on since 9/96 spanning 108 versions and no ship date in sight.

Even though the feature freeze is declared, major changes continue to get integrated into the kernel. Most merges seem to be due to fundamental bug fixes and or cross platform issues.

Organization

An analysis / description of the OSS development organization and process is in a second memo titled "Open Source Software." This section describes attributes of OSS that are unique to Linux.

Wired Magazine ran a recent story chronicling the history of Linux "The Greatest OS that (N)ever was" <http://www.wired.com/wired/5.06/linux.html>.

The growth of the development team mirrored the organic, not to say chaotic, development of Linux itself. Linus began choosing and relying on what early Linux hacker Michael K. Johnson calls "a few trusted lieutenants, from whom he will take larger patches and trust those patches. The lieuts more or less own relatively large pieces of the kernel."

As with other OSS projects, the General Public License ("CopyLeft") and it's relatives are considered instrumental towards creating the dynamic behavior around the Linux codebase:

In a sense, GPL provided a written constitution for the new online tribe of Linux hackers. The license said it was OK to build on, or incorporate wholesale, other people's code - just as Linux did - and even to make money doing so (hackers have to eat, after all). But you couldn't transgress the hacker's fundamental law of software: source code must be freely available for further hacking

Linux Technical Analysis & OS Structure

Anatomy of a Distribution

"Linux" is technically just a kernel, not the entire supporting OS. In order to create a usable product, Linux "distributions" are created which bundle the kernel, drivers, apps and many other components necessary for the full UNIX/GUI experience.

MICROSOFT CONFIDENTIAL - PAGE 6

HIGHLY
CONFIDENTIAL

MSB 0078126
CONFIDENTIAL

MSB 00000078126

These subsystems are typically developed in an OSS manner as well and several of them -- e.g. the Xfree86 GUI -- have a codebase size/complexity that exceeds the Linux kernel

These external components come from many sources and are individually hand picked by the distribution vendor for a particular product. A frequent source of controversy stems from distribution vendors bundling non-GPL code with the Linux kernel and mass distributing them.

A partial list of components is in the following table:

Component	Codebase / Name	Provider/Maintainer(s)
Kernel	Basic OS, Networking Stack	Linux (http://www.kernel.org)
File System(s)	Msdos, ext2fs	Linux Kernel
Sys Libs	Glibc, Libc	GNU / FSF
Drivers		Linux, Individual Contributors
User Tools	Gnu user tools	GNU/FSF
System Installation	LISA	Caldera
App Install Management	RedHat Package Manager	RedHat
Development Tools	GNU Development tools GCC	GNU/FSF
Web Server	APACHE	The Apache Group http://www.apache.org/
Mail Server	SendMail	http://www.sendmail.org
DNS Server	BIND	http://www.bind.org
SMB Server	SAMBA	http://www.samba.org
X Server	Xfree86 / MetroX	Xfree86 project / MetroX commercial
Window Manager	FVWM	GPL
Widgets	Motif	X Consortium
Desktop Tools	X Contrib KDE Gnome	X Consortium http://www.kde.org http://www.gnome.org
Management	RPM Package Installed Roll own distribution specific	RedHat (free) Debian / Slackware

Descriptions of some of the larger components are below:

Kernel - GPL

The kernel is the core part of Linux that is expressly managed by Linus and his lieutenants and is protected via the GPL.

Functions contained in the Linux Kernel include:

- Core OS Features (scheduling, memory management, threads, Hardware Abstraction, etc)
- Network Stack
- File system

MICROSOFT CONFIDENTIAL - PAGE 7

HIGHLY
CONFIDENTIAL

MSB 0078127
CONFIDENTIAL

Extensive on-line documentation of the Linux kernel architecture and components can be found on: <http://sunsite.unc.edu/linux/LDP/tk/tk.html>. Note that video drivers exist outside the kernel -- the kernel only has rudimentary text display support to a console.

Drivers -- GPL

An assortment of modules for standard functions and devices are typically part of the kernel distribution. In addition, a selection of non-standard modules is often included. Mostly GPL, however in some cases, NDAs with hardware manufacturers are required to get specs to make a driver, in which case they are not open source.

Linux device drivers are typically developed by users for specific devices on their machines. This incremental, piecemeal process has created a very large pool of device drivers for Linux (as of 7/1/98):

- Video: <http://sunsite.unc.edu/LDP/HOWTO/Hardware-HOWTO-6.html> -- close to 400 drivers available
- Network: <http://sunsite.unc.edu/LDP/HOWTO/Hardware-HOWTO-11.html> -- ~75 network cards supported
- PCMCIA <http://sunsite.unc.edu/LDP/HOWTO/Hardware-HOWTO-26.html> -- ~150 supported cards.

NatBro points out:

An important attribute to note which has led to volume drivers is the ease with which you can write drivers for linux, and the relatively powerful debugging infrastructure that linux has. Finding and installing the DDK, and trying to hook up the kernel debugger and do any sort of interaction with user-mode without tearing the NT system to bits is much more challenging than writing the simple device-drivers for linux. Any idiot could write a driver in 2 days with a book like "Linux Device Drivers" -- there is no such thing as a 2-day device-driver for NT.

Recently, a small number of hardware vendors have begun to provide Linux drivers for their NICs (3Com) and SCSI adapters (Adaptec). These drivers are believed to be protected by the Library-GPL and are consequently not open source (the Library-GPL is described later). It remains to be seen whether this will create the momentum to develop more commercial drivers for Linux.

System Libraries & Apps -- GNU GPL

System libraries provide:

- Basic POSIX api's for system services
- Basic API's to support commandline / shell utilities.

The system libraries in a Linux distribution are NOT managed by Linus. As such, there has been a small amount of versioning / forking in this area with two dominant libraries -- glibc and libc which introduce minor incompatibilities between different apps.

MICROSOFT CONFIDENTIAL -- PAGE 8

HIGHLY
CONFIDENTIAL

MSB 0078128
CONFIDENTIAL

User Tools (GPL, GNU FSF)

These are basic UNIX command line tools and shell environments. Many shell environments exist although all are supported by the FSF.

Also included in this category are "old standby" apps such as finger, telnet, etc.

Development Tools (GPL)

A hallmark of the UNIX operating system is the free availability of development tools / compilers. The GCC and PERL language compilers are often provided for free with all versions of Linux and are available for other UNIXes as well.

These tools are the "old standbys" of the UNIX development world and are widely used across all Unix platforms. This mass commoditization of development/debug tools is a key contributor to the common skillset efficiencies realized by the Linux process.

By the standards of the novice / intermediate developer accustomed to VB/VS/VC/J, these tools are incredibly primitive.

GUI / UI

X Server

The X Server standard is owned by MIT under contract by the X Consortium. X Consortium's licensing practices are viewed as too restrictive by the OSS crowd so a series of public X initiatives were launched with Xfree86 being the dominant distribution.

Interestingly, the Xfree86 development team licenses their code under the BSD license because they consider GPL too restrictive: <http://www.redhat.com/linux-info/xfree86/developer.html>.

Configuring the XFree86 system on Linux can be a very difficult, time consuming process. Linux has no hardware abstraction layer for video services, and most video card manufacturers do not provide Linux OS video drivers. Thus, XFree86 provides internal support for a wide variety of video cards and chipsets. Correctly configuring XFree86 requires the user to know the manufacturer, model, and chipset for their video card. In many cases, the user must know or calculate the video timings as well.

Widgets & Desktops

There are multiple widget sets which exist in many applications, so all X applications do not look the same or act the same ways like in Windows. Motif is considered the defacto Unix widget set, but since it is not freely distributable, it is contrary to the Linux model.

Consequently, Linux distributions usually choose one of several similar, but not completely compatible Widget sets.

- Motif
- LessTif
- Xaw3d (3d athena widgets that look like motif)
- QT

MICROSOFT CONFIDENTIAL - PAGE 9

HIGHLY
CONFIDENTIAL

MSB 0078129
CONFIDENTIAL

Obviously, this mess has spawned several efforts to unify the "desktop" as well as the widget sets. In typical Linux fashion, there are several competing efforts:

- Gnome/totally new
- KDE
- FreeQT/KDE
- CDE/commercial

Commercial Linux OS

Binary Compatibility

Server

Almost all of the system components necessary to run server applications are part of the core distribution maintained by Linus. Consequently, for a given hardware type, almost all Linux server application binaries will natively run. Across hardware types (e.g. x86 vs. PPC), generally only a recompile of the application is necessary.

There is essentially 100% source code compatibility for system application code.

Solaris / SCO x86 Compatibility

Via compatibility libraries, Linux on x86 is able to natively execute most SCO UNIX and Solaris x86 binaries. Oracle on SCO is widely cited as an example (although Oracle does not "officially" support SCO binaries on top of Linux -- also Oracle has recently announced development of a native Linux version of Oracle 8 to ship in March 1999)

Client

Client distributions, however, are a different story stemming most directly from the current "mess" in X-windows / GUI systems for Linux.

Binary compatibility issues generally stem from differences in non-kernel code that's required to turn the kernel into a full OS.

Binary Incompatibility: Netscape Communicator

One example of this incompatibility is Netscape Communicator for Linux. The released versions of Netscape Communicator for Linux are built based on libc5, instead of the newer glibc which Caldera supports. RedHat, however ships glibc instead of libc5 requiring users install libc5 as well as glibc.

RedHat

<http://www.redhat.com>

RedHat Corporation was founded in 1995 by a pair of Linux developers/enthusiasts with the intent of creating a commercially supported, "cleaned-up" Linux distribution.

MICROSOFT CONFIDENTIAL -- PAGE 10

HIGHLY
CONFIDENTIAL

MSB 0078130
CONFIDENTIAL

MSB 000000078130

The company currently has ~35 employees. Financials and some run-rate information is available in an interview with their CEO in Infoworld ([http://www.infoworld.com/cgi-](http://www.infoworld.com/cgi-bin/displayArchive.pl?98/23/e03-23.102.htm)

[bin/displayArchive.pl?98/23/e03-23.102.htm](http://www.infoworld.com/cgi-bin/displayArchive.pl?98/23/e03-23.102.htm));

Bob Young, president of Red Hat expects the 3-year old company to earn revenues of \$10 million this year and to ship about 400,000 copies of Linux, ranging from \$50 to near \$1,000 for a supported version.

Commercially-Developed Extensions

Perhaps the most interesting aspect of Red Hat's business model is their extremely active and continuing contributions to the Linux community. Several prior initiatives spearheaded by RedHat have been released as OSS for modification. In most cases, these code releases were simple fixes or additional drivers.

Redhat actively employs several key Linux developers and pays them to hack Linux fulltime. Some of the components which have been "donated" back to the Linux effort include:

- **RedHat Package Manager** -- RPM is Linux component which provides application install / maintenance facilities for Linux similar to the Application Manifest being developed by Microsoft.
- **Pluggable-Authentication Manager** -- PAM is similar to the NT SSPI / SAM system and allows for componentized plug-ins to handle the authentication function (RedHat provides an LDAP plugin). PAM was originally available on Sun systems.

One of the larger "grants" however has been the now universal "Redhat Package Manager" or RPM which ships with almost all Linux distributions. RPM creates the concept of an application manifest which simplifies the job of installing & removing applications on top of Linux.

Redhat's current development project is a new GUI for Linux call "Gnome". Gnome is a response to latent concerns with non-GPL versions of the X-windows user interface.

Product Features

Of the Commercial Linux Distributors, Redhat has the largest array of SKU's. At the highest end, Redhat bundles the following with their distributions of Linux:

- Apache Web Server
- Corel WordPerfect
- DBMaker DBMS by Casemaker
- Xfree86 window server

Caldera

Caldera is Ray Noorda's latest company with its eye on the operating system marketplace. Caldera's financials and sales are unpublished but it is widely believed to be the #2 commercial Linux vendor after RedHat.

Caldera bundles several components with their version of Linux including:

- StarOffice 4.0 by Germany's Star Corp.

MICROSOFT CONFIDENTIAL -- PAGE 11

HIGHLY
CONFIDENTIAL

MSB 0078131
CONFIDENTIAL

- Adabas SQL Server by Software AG
- Netware client & Admin
- Netscape fasttrack server + communicator
- Xfree86 and MetroX X-window systems

Others

Other Linux distributions seem to be falling by the wayside of RedHat and Caldera. They include SlackWare, SuSe, and Debian to name a few. A comprehensive list of distributions can be found on <http://www.linux.org>.

Commercial Linux ISV's

There are currently no major ISV's who derive a significant percentage of their sales from the Linux platform. A somewhat complete list of the commercial apps available on Linux can be found on: <http://www.uk.linux.org/LxCommercial.html>.

Reasons for this include:

- **First-use Linux apps are free** – most of the primary apps that people require when they move to Linux are already available for free. This includes web servers, POP clients, mail servers, text editors, etc.
- **Linux market is still immature** – the Linux market is still in its infancy and the current state of Linux commercial software may change radically in the coming months
- **Current Linux users are wary of commercial products** – you can scout any of various Linux discussion and mailing lists and quickly run into users admonishing commercial software providers and trying to launch a jihad against category X via open source software (at the time of this writing, Lotus Notes is a popular target)²

Library-GPL

Unlike the GPL (General Public License – described in depth in "Open Source Software") which forces all derivative works to be free, Linux software libraries have the more limited "Library GPL" which allows applications which merely link to Linux to be considered non-derivative.

The Library-GPL removes a key impediment to commercial software vendors developing products on top of Linux.

The Library-GPL is defined at <http://www.fsf.org/copyleft/gpl.html>

Binary Unix Compatibility

Linux adheres to several UNIX standards most notably POSIX 1003.1c. When compiled and running on it's various CPU platforms, Linux is generally binary compatible (more so on the server than on the desktop) with the primary commercial UNIXs including:

² An IBM official in describing IBM's hesitation to support Linux as an app platform stated: "Linux does not run in many corporations that actually buy product." <http://www.techweb.com/se/directlink.cgi?CRN19980727SD127>

- Solaris/SunOS on SPARC
- Solaris on x86
- SCO on x86
- Digital UNIX on Alpha
- SGI IRIX on MIPS

Microsoft

Microsoft's current involvement in Linux is limited to distribution of client code for strategic services such as Netshow as well as helping SAG port DCOM to Linux. IE is currently not officially supported on Linux.

Intel

Intel is directly involved in helping port Linux to Merced. Intel is also involved with the GCC over Merced development efforts.

Netscape

In the press, Netscape is cited as the #1 commercial provider of software for Linux. Marc Andreason has been extensively quoted as saying that "Linux is a tier 1 platform for Netscape".

Until recently, however, the only server product that Netscape explicitly sells for Linux is their Fastrack server with other servers merely being licensed to the respective Linux vendors for their own redistribution. On July 21st, however, Netscape formally announced intentions to port all of their server application products to Linux starting with Mail and Directory services.

All of Netscape's client products are available on the Linux platform.

Oracle

Oracle recently announced (7/18/98) their support for Oracle 8 on top of Linux to be shipped in March 1999.

Sun

Sun's involvement in Linux is inconclusive. Early this year (1998), Sun joined the board of Linux International which is one of many user groups representing Linux.

At one level, Linux competes (quite favorably) against Sun's own Solaris x86 port.

At a secondary level, Sun may view Linux as a strategic ally b/c it generally represents the low-end of the software market and could therefore arguably hurt Microsoft more than it hurts Sun.

SoftwareAG

SoftwareAG has ported it's ADABAS database server to Linux and is currently bundled with Caldera's distribution.

MICROSOFT CONFIDENTIAL - PAGE 13

**HIGHLY
CONFIDENTIAL**

**MSB 0078133
CONFIDENTIAL**

Corel

Corel has ported their WordPerfect Suit to Linux and is currently offering it bundled with several of RedHat's SKU's

Computer Associates

Recently announced intentions to port CA-Ingres DB to Linux:
<http://x10.dejanews.com/getdoc.xp?AN=370037691&CONTEXT=900053229.949289093&hitnum=0>.

Market Share

Linux's exact market share is very difficult to calculate because:

- The majority of Linux installations are downloaded from anonymous FTP sites - NOT purchased. Consequently, there are no published sales figures to track.
- (Some) Commercial Linux purchases can be used to install multiple machines
- Because Linux revs so often, there's a very high likelihood of double-counting actual installations vs. downloads/purchases
- There are no separate client & server distributions. Consequently, it's difficult to compare Linux numbers wholesale to NTS / NTW numbers without accurate usage data from the Linux community.

Below I include data / pointers from some of the more prominent attempts to isolate the number of Linux users.

Installed Base

The most comprehensive Linux market share survey was published by Red Hat in March 1998: <http://www.redhat.com/redhat/linuxmarket.html>

Using available data collected from other distributions, RedHat calculated a retail CD sell rate of:

- 1996: 450,000
- 1997: 750,000

RedHat's estimate of the growth of the Linux installed user base (which includes CD purchases as well as downloads as well as client + server) is:

- 1993: 100k
- 1994: 500k
- 1995: 1.5M
- 1996: 3.5M
- 1997: 7.5M

Other estimates put the Linux installed base from 5 Million (Ziff Davis), to 10 Million (Linux advocates).

MICROSOFT CONFIDENTIAL - PAGE 14

HIGHLY
CONFIDENTIAL

MSB 0078134
CONFIDENTIAL

MSB 000000078134

Server

IDC's most recent "Server Operating Environments" report provides the following breakdown of shipments in the Server OS space.

Using the 240K number shipped in 1997, IDC seems to be estimating ~750K total installed Linux server systems. Compared to other market share studies, IDC's may be underestimating the actual new Linux server installations -- I believe IDC may be counting only top distributions in their survey.

Client

Starting with Dataquest's market share figures published in June '98, I injected the incremental Linux numbers derived from RedHat's market survey (showing 7.5M users at the end of 1997).

Distributor Market Share

IDC provides information on the relative market share of the Linux distributors:

Linux Qualitative Assessment

I purchased and installed a copy of Caldera's OpenLinux v1.2 standard edition. I installed it on an old P5-100 / 32MB RAM machine in my office that used to run NT4. Knowing that device driver support on Linux was well below NT's, I intentionally chose a machine and peripherals that represented the 80% of the *installed base* (e.g. 3c509 NIC, Adaptec SCSI controller, etc.)

Installation

Caldera provided an auto-run CD which launched directly into their setup program -- "LISA". Lisa prompted me for:

- Language selection (an interesting future research project would be to truly understand the depth of localization support provided by Linux.)
- Keyboard selection
- IDE hard drive & CD ROM device detection.

Although the dialogs could use a lot of work (e.g. many questions were phrased as double-negatives -- "Should setup disable plug & play device detection (yes/no)"), up to this point I was asked no questions that a power user couldn't correctly answer.

A second round of device detection impressively auto-discovered my:

- Adaptec SCSI adapter
- Plextor CD-ROM drive

MICROSOFT CONFIDENTIAL -- PAGE 15

HIGHLY
CONFIDENTIAL

MSB 0078135
CONFIDENTIAL

- Seagate Hard Drive
- 3Com 3c509 Ethernet adapter

I selected default device settings for each hardware option, selected "typical" install options and then LISA started copying.

This phase of the install/setup process was finished in 30 minutes (most of that time copying) and with a total of ~15 dialog boxes.

UI

As mentioned earlier, one of the quirks of UNIX / Linux relative to NT is that video drivers run in userspace and are not required for most system functionality. Linux is quite content with just a command prompt.

A second round of installation scripts was necessary to install the GUI. The installer gave me the option of choosing which video subsystem to install / configure and I chose the Xfree86 server because it's an entirely open source system (the other option - MetroX - was provided by Caldera and is believed to be the more stable codebase).

This part of setup definitely required knowledge of video systems even beyond many power users. Not only did I have to know the name / make / model of my video card and chipsets but I was presented with questions about their revision numbers, the scan rates of my monitors, etc. After significant trial and error, I finally got my video system working correctly.

The latest generation Xfree86 + CDE was slick and definitely represented among the best-of-breed in UNIX GUI's. A SUN desktop user would be perfectly at home here. An advanced Win32 GUI user would have a short learning cycle to become productive.

Following UNIX philosophy, however, mastery of the GUI was not enough to use the full system. Simple procedures such as reading a file from a floppy disk required jumping into a terminal window, logging in as administrator, and running an arcane "mount" command.

Networking

A very illustrative case of how the Linux user community works was revealed by my experiences with the networking subsystem.

Caldera's OpenLinux installer only provided the client daemon to handle the BootP protocol (as opposed to DHCP) and for some reasons, it didn't install correctly. I looked around on the CD that Caldera provided for a DHCP daemon and couldn't find one.

A small number of web sites and FAQs later, I found an FTP site with a Linux DHCP client. The DHCP client was developed by an engineer employed by Fore Systems (as evidenced by his email address; I believe, however, that it was developed in his own free time). A second set of documentation/manuals was written for the DHCP client by a hacker in Hungary which provided relatively simple instructions on how to install/load the client.

I downloaded & uncompressed the client and typed two simple commands:

Make - compiles the client binaries

MICROSOFT CONFIDENTIAL - PAGE 16

**HIGHLY
CONFIDENTIAL**

**MSB 0078136
CONFIDENTIAL**

Make Install - installed the binaries as a Linux Daemon

Typing "DHCPD" (for DHCP Client Daemon) on the command line triggered the DHCP discovery process and voila, I had IP networking running.

DHCP as an example of Linux process

Since I had just downloaded the DHCP client code, on an impulse I played around a bit. Although the client wasn't as extensible as the DHCP client we are shipping in NT5 (for example, it won't query for arbitrary options & store results), it was obvious how I could write the additional code to implement this functionality. The full client consisted of about 2600 lines of code³.

One example of esoteric, extended functionality that was clearly patched in by a third party was a set of routines to that would pad the DHCP request with host-specific strings required by Cable Modem / ADSL sites.

A few other steps were required to configure the DHCP client to auto-start and auto-configure my Ethernet interface on boot but these were documented in the client code and in the DHCP documentation from the Hungarian developer.

Key takeaways here:

- Contrary to popular belief, even though this was open source, I never had to touch the 'C' code to get the core functionality working.
- The author of the driver and the author of the documentation were two geographically separated individuals.
- The GPL + incremental improvements process had *already* been at work as evidenced by the Cable Modem/ADSL extensions.
- **Most importantly:** a process that NatBro pointed out in the OSS paper - "a modestly skilled UNIX programmer can grow into doing great things with Linux". I'm a poorly skilled UNIX programmer but it was immediately obvious to me how to incrementally extend the DHCP client code (the *feeling* was exhilarating and addictive).

Additionally, due directly to GPL + having the full development environment in front of me, I was in a position where I could write up my changes and email them out within a couple of hours (in contrast to how things like this would get done in NT). Engaging in that process would have prepared me for a larger, more ambitious Linux project in the future.

Apps

Caldera bundled StarOffice from Star Corp in Germany. The Office team is quite familiar with StarOffice as a "second-string" contender in the suite category after Corel (which is bundled with Red Hat) and Lotus.

³ By contrast, the NT5 DHCP client consists of 20,000 LOC. RameshV from the NT5 team pointed out some of the incremental functions in our DHCP client including thread-safety, APIs, auto-net support, suspend/resume PNP support, multiple interface support, additional error checking functionality + messages. Notice that these features aren't DHCP features *per se* but rather, integrative/scenario-driven functionality.

StarOffice was almost entirely an Office 97 clone from a UI perspective. The menus, buttons, placement, etc. were all generally identical. In many cases, large areas of functionality in the menu bar were missing (e.g. Macros). Other stereotypical Office97 features (e.g. red squiggles under misspelled words) were correctly replicated.

As a test, I tried importing a somewhat simple PowerPoint document into StarOffice from a floppy disk. This required jumping into an x-terminal and mounting a new floppy disk into the Linux file system namespace and pointing out to Linux that it was FAT16 formatted. From there, I launched StarOffice's PowerPoint clone and pointed it at the namespace for the floppy and uploaded the file.

Simple slides (such as pure text + bullet points) imported nearly 100% correctly (although fonts and sizing were changed). Complex slides (using PowerPoint's line art, etc.) were almost always totally trashed.

Perceived Performance

Caldera also bundles Netscape's Navigator browser. The browser's UI, of course, perfectly matches Netscape's UI on win32 platforms.

I didn't have the time to run true performance tests, but my anecdotal / perceived performance was impressive. I previously had IE4/NT4 on the same box and by comparison the combination of Linux / Navigator ran at least 30-40% faster when rendering simple HTML + graphics.

Testing end user applications on top of Linux will be an interesting performance test in the future.

On a negative note, after I had instantiated 3 instances of Navigator on the box, performance came to an almost complete standstill, the mouse became unresponsive, none of the keyboard command sequences worked and I had to reboot the box.

Conclusions

Skilled users with modest developer backgrounds are probably delighted to use Linux due to the endless customizability afforded by Open Source. The simplicity and consistency of the process to modify the system presents a very low learning curve towards "joining" the Linux process.

Long term, my simple experiments do indicate that Linux has a chance at the desktop market but only after massive investments in ease of use and configuration. The average desktop user is unfamiliar with "make".

Linux Competitive Issues

Consumers Love It.

A December 1997 survey of Fortune 1000 IT shops by Datapro asked IT managers to rate their server OS's on the basis of: TCO, Interoperability, Price, Manageability, Flexibility, Availability, Java Support, Functionality, and Performance. RedHat provides summary info at: <http://www.redhat.com/redhat/datapro.html>

MICROSOFT CONFIDENTIAL - PAGE 18

**HIGHLY
CONFIDENTIAL**

**MSB 0078138
CONFIDENTIAL**

MSB 000000078138

When overall satisfaction with the OS's was calculated, Linux came out in first place. Linux was rated #1 in 7 of 9 categories in the DataPro study losing only on: functionality breadth, and performance (where it placed #2 after DEC)⁴

Linux vs. NT

Windows NT is target #1 for the Linux community. To characterize their animosity towards NT (or, for that matter, anything Microsoft) as religious would be an understatement⁵. Linux's (real and perceived) virtues over Windows NT include:

- **Customization** - The endless customizability of Linux for specific tasks - ranging from GFLOP clustered workstations to 500K RAM installations to dedicated, in-the-closet 486-based DNS servers - makes Linux a very natural choice for "isolated, single-task" servers such as DNS, File, Mail, Web, etc. Strict application and OS componentization coupled with readily exposed internals make Linux ideal.

The threat here is even more pronounced as over time, the number of servers (and consequently dedication to specific tasks) will increase. Customers enjoy the simpler debugging and fault isolation of individual servers vs a monolithic server running multiple services.

- **Availability/Reliability** - There are hundreds of stories on the web of Linux installations that have been in continuous production for over a year. Stability more than almost any other feature is the #1 goal of the Linux development community (and the #1 cited weakness of Windows)
- **Scaleability/Performance** - Linux is considered faster than NT in networking, and processes. In particular, as a server, Linux's modular architecture allows the administrator to turn off graphics, and other non-related subsystems for extreme performance in a particular service
- **Interoperability** - Every open protocol on the planet (and many of the closed ones) have been ported to Linux. In a Windows environment, work from the SAMBA team enables Linux to look like an NT Domain Controller / File Server.

Recently, the NT performance team ran their NetBench file/print test against a recent Linux distribution. Results indicate that although NT slightly outperforms Linux, Linux's performance is still quite acceptable and competitive considering the years of tuning that has been applied to the NT SMB stack.

⁴ George Spix points out that this survey in many cases quotes a self-selected audience. They "must have already been lovers to buy it... all hobbyists like their 'rigs'".

⁵ Linus Torvalds himself is not quite as religious as many of the other Linux developers but has some great comments in an interview he gave for Boot magazine:

http://www.bootnet.com/youaskedforit/lp_linux_manifesto.html

MICROSOFT CONFIDENTIAL - PAGE 19

HIGHLY
CONFIDENTIAL

MSB 0078139
CONFIDENTIAL

MSB 000000078139

Linux vs. Java

Linux developers are generally wary of Sun's Java. Most of the skepticism towards Java stems directly from Sun's tight control over the language -- and lack of OSS.

The Linux community has been asking Sun to treat the Linux platform as a tier-1 Java platform almost since the dawn of the language. However, Sun does NOT support the JDK for Linux.

Interestingly, in order to develop the Linux JDK, several Linux developers signed NDA's to develop the port (<http://www.blackdown.org>). These pressures have also spawned several OSS JVM clones including <http://www.kaffe.org>.

Linus comments in (<http://www.linuxresources.com/news/linux-expo.html>)

While Linus would like to see an officially supported Java Development Kit from Sun, he is still not impressed with Java and would prefer to stay out of the Microsoft/Sun clash over Java purity;

Linux vs. SunOS/Solaris

The Linux community has ambivalent feelings towards Sun. On the one hand, as the most vocal critic of Microsoft, Sun is praised. On the other hand, as the most visible yardstick in the UNIX world, beating Solaris / SunOs is a favorite past-time of Linux hackers.

Using the Lmbench OS benchmark, Linux outperforms SunOS not only on x86 but, impressively, *on Sun Hardware as well* in networking, process / context switch times, disk I/O, etc.

Some (not very scientific or comprehensive) OS performance statistics can be found on: <http://www.calp.rutgers.edu/~davem/scoreboard.html>

In generating these performance results, the great number of eyes (and consequently large amount of hand tuning / optimizing of critical code paths) is most frequently cited.

A general architectural comparison citing the performance benefits of Linux over SunOS can be found on: <http://www.nuclecu.unam.mx/~miguel/uselinux/SparcLinux.html>

Sun has recently announced (8/10) the free licensing of Solaris binaries for non-commercial institutions (<http://www.sun.com/edu/solaris/index.html>). Presumably this is due to competitive pressures from Linux.

Linux on the Server

The vast majority of Linux's installed, production base is projected to be in servers.

Reasons why Linux is strong in this market include:

- **Unix heritage** -- the server market, especially at the high-end, is already familiar & comfortable with UNIX, Internet-based freeware, etc.

MICROSOFT CONFIDENTIAL - PAGE 20

HIGHLY
CONFIDENTIAL

MSB 0078140
CONFIDENTIAL

- **Professional users** - high end server administrators are often developers/power users themselves and are therefore comfortable with recompiling apps, etc.
- **"Generic" services** - these are services defined via open, lowest-common-denominator protocols such as DNS, SMTP, etc. Functional differentiation is lower in the server market than it is in the client market. There is a lower bar for experimentation with servers since it disrupts downstream client activity very little.
- **Dedicated Functionality** - because servers are typically tasked with a single function (e.g. mail, file/print, database, etc.), the level of required integration with other services and devices in the organization is much lower.

Network Infrastructure

Linux is often used to provide commodity, low horsepower, high reliability network infrastructure services. For example:

- DNS
- DHCP
- Print Servers
- File Servers

ISP Adoption

One of Linux's core user bases is ISP shops. Some of the reasons for this include:

- **Cost** - ISPs live on horribly tiny margins. Linux's free price + wide hardware support is consequently very attractive.
- **Maintainability** - If something breaks, it needs to be fixed immediately. In larger ISPs, the technical expertise to debug code breaks or at least install quickly available patches is plentiful. Remote manageability in particular is a key attribute.
- **Reliability** - perception that non-Linux OS's aren't reliable or scalable enough (in particular Windows NT)
- **UNIX background** - ISPs are traditionally Unix havens. ISP sys admins are very well versed in arcane UNIX command line admin, remote administration, etc. In a group that's very predisposed towards UNIX's strengths, Linux represents a best-of-breed UNIX.

Thin Servers

Linux is emerging as a key operating system in the nascent thin server market

- **Source code availability** - Freely available source code provides for easy customization of the OS
- **Commodity protocols** - Thin servers speak very simple, non-extensible, commodity protocols to clients such as HTTP, SMTP, and SMB.

MICROSOFT CONFIDENTIAL - PAGE 21

HIGHLY
CONFIDENTIAL

MSB 0078141
CONFIDENTIAL

- **Modularity & Small size** - Because the OS was designed in a very non-integrated, componentized manner from the outset, it's very easy to build boxes that don't have a monitor, keyboard, etc.
- **Cost** - Obviously, margins are very low in embedded devices & a free OS helps
- **Code Maintenance** - Because the Linux source is constantly being upgraded, embedded developers are reassured that new changes / fixes can be snapped back to their systems at any time.
- **Tool Availability** - Unix tools are far more powerful than the current crop of embedded development tools.

One of the most prominent thin-server on the market based on Linux today is the Cobalt Microserver (<http://www.cobaltmicro.com>). Other thin server vendors (most notably Whistle Interjet) are using FreeBSD derived products.

Case Study: Cisco Systems, Inc.

IDC published a study of 3 corporate IS departments which had significantly deployed Linux. Cisco has several hundred Linux servers deployed through their organization serving the following functions:

- NFS/SMB server
- Print Server (LPD & SMB)
- Small office productivity (AppixWare office suite, Netscape Navigator)
- WWW Server & Proxy
- Software development

Linux on the Client

Due to its UNIX heritage and Hacker OS background, Linux is a weak client-desktop OS. Additionally, the OSS paper points out why, in a broad sense, OSS is much more of a server threat than a desktop threat.

There are, however, several initiatives attempting to push Linux as a viable desktop replacement⁹. Each of the various Desktop environments (GNOME, KDE, CDE) come bundled with basic productivity applications and there are 2 full fledged office suite products (from Corel and StarOffice) which provide varying degrees of file format compatibility with Microsoft Office.

⁹ Linus Torvalds states that his intent has always been to target Linux at the desktop (<http://www.heise.de/ct/english/98/16/032/>): "Note that the reason I propagate Linux as a desktop operating system is because I think that's the more difficult market. I was personally never very worried about Linux as a server platform - servers are "easy" compared to desktops, because in the end servers are fairly anonymous - you only see the network behaviour of a server, while with a desktop system it's much more of a "complete immersion" environment, not just the network part"

App / GUI Chaos

Unlike the Kernel -- where Linus Torvalds maintains the core source tree, the Linux GUI has NOT been singularly managed and consequently has a highly forked tree.

Linux does not have a consistent UI look and feel due to the variety of widget sets (a widget is analogous to an OCX or VBX) such as Motif, LessTiff, MIT Athena, Sun OpenLook, etc. Because widgets represent central UI concepts (such as a close button, dropdown menu, dialog box, etc.), users get different look-and-feels and often different usage semantics.

In addition to Widgets, the "desktop" or "shell" has also forked. Primary players in the shell arena include:

- **Common Desktop Environment (CDE)** -- a collaboration between major commercial Unix manufacturers. CDE, however, is not GPL'd and has thus resulted in multiple Linux groups creating CDE replacements. CDE is available on Linux.
- **K Desktop Environment (KDE)** - a "free" CDE clone. KDE replaces all functionality in CDE but does not provide a widget set. (In practice the widget set is actually MORE lines of code than the desktop). Consequently, the KDE developers chose the QT widget set which was most liberally licensed -- but still not GPL'd -- and compatible with most other UNIX systems. This however, launched the final band of GPL zealots who created...
- **Gnome** -- a radical new UI initiative based loosely on X-windows and incorporating CORBA into the desktop. While this is an ambitious task, and may be more revolutionary than CDE is, its long from being complete and is lacking in application support.

The lack of singular, customer-focused management has resulted in the unwillingness to compromise between the different initiatives and is evident of the management costs in the Linux process.

Unix Developers

Linux as a client has found a home with UNIX developers. Many developers prefer to use Linux for their dev machines in order to write code for other UNIX platforms. The ease of debugging on top of a platform where there is open source is often cited.

Non-PC Devices

Corel's NC devices were based on a Linux derived OS. These efforts, however, have since been suspended (with the Corel developed application-level code being returned to the OSS community)

MICROSOFT CONFIDENTIAL -- PAGE 23

HIGHLY
CONFIDENTIAL

MSB 0078143
CONFIDENTIAL

MSB 000000078143

Linux Forecasts & Futures

Current Initiatives / Linux Futures

There are literally hundreds of small research projects attempting to improve various parts of the Linux OS.

Some projects include:

- **Linux 2.2** - High Availability features such as deeper RAID support (RAID 0, 1, 5 supported today), volume management; file system performance improvements; asynchronous I/O & completion ports; ipv6; . An excellent feature summary can be found on: <http://wn.net/980730/a/2.2chFinal.html>.
- **Linux 3.0** - Linus forecasts that the next version of the Kernel will incorporate better SMP scalability and begin to attack the clustering problem. Development is far from starting so details / commitments are extremely sketchy.
- **Beowulf clustering** - Beowulf is a shared-nothing cluster that runs today on Linux. It requires specially developed applications which are able to spawn subprocesses on remote hosts for computing. As such, it is not a real competitor to WolfPack and most of the magic in Beowulf is in the applications rather than system services. However, as a press-magnet, Beowulf clusters with appropriate software have been demonstrated at supercomputer power (a 10GFLOP was recently ranked #315 on the top 500 supercomputers list maintained by the NCSA).
- **DIPC** - Distributed Inter-Process Control Pack - provides standard IPC functions to client apps (semaphores, shared memory, etc.) but is able to remote those functions to network hosts.
- **GNOME** - Next generation UI initiative for Linux loosely based on X-windows +CORBA . More info at <http://www.gnome.org>. Many of the key developers for Gnome work for RedHat.

"Parity Growth"

The biggest future issue for Linux is what to do once they've reached parity with UNIX. JimAll used the phrase "chasing tailights" to captures the core issue: in the fog of the market place, you can move faster by being "number 2 gaining on number 1" than by being number 1.

Linux has now reached parity / incrementally ahead of other Unixes. Consequently, it will be much harder to achieve the big leaps the development team is accustomed to.

From Wired's piece on Linux:

This two-track development process has made Linux probably more advanced and yet more stable than any other version of Unix today. "Linux is now entering an era of pure development instead of just catching up," says Jacques Gélinas.

MICROSOFT CONFIDENTIAL - PAGE 24

HIGHLY
CONFIDENTIAL

MSB 0078144
CONFIDENTIAL

MSB 000000078144

Strengths

A second paper on "Open Source Software" goes into depth on the generic advantages of the Open Source Process.

Unix Heritage & Fast Copying

Linux unabashedly steals the best ideas from the various UNIX flavors. This means free R&D. Recently, Linux has begun to copy NT-ish features such as transmitfile(), a hacked form of IO Completion Ports, etc.

Established / high-visibility bazaar

Linux is the most often cited example of a "credible" open source project. By being the largest OSS project today, it's the most sustainable in the future.

Dominance In Education / Research Markets

New ideas from academia + new computer scientists are being trained wholesale in the Linux OS. In particular, Europe and Asia are very hooked on the Linux OS. Email from BartelB (Marketing Manager EdCU):

For higher education in particular, Linux represents an alternative to the Commercial demons of software, (not a quantitative statement but in talking with many CS students who supply 60% of the labor for higher education IT departments, they express these feelings and its a problem). They feel that once they commit to a windows platform there creativity will be lost. Money is not there driving force, they don't want to be "Borged".

Weaknesses

The paper on "Open Source Software" provides general process weaknesses. Here, we'll try to list only the weaknesses that are unique to Linux.

Unix Heritage

Linux's biggest advantage can also quickly become a disadvantage - particularly in volume markets where ease of use is paramount. Some nascent efforts have been launched to make Linux friendlier but they are generally receiving relative apathy from the dev community (<http://www.seul.org>).

Too Many Managers

In a typical Linux distribution, the majority of the code comes from sources outside of the main Linux tree. This piecemeal approach will make it especially hard to solve architectural problems and launch new, cross-component initiatives.

Worst case scenarios

This section is pure speculation. What are some of the worst case scenarios for Linux to hurt Microsoft?

MICROSOFT CONFIDENTIAL - PAGE 25

HIGHLY
CONFIDENTIAL

MSB 0078145
CONFIDENTIAL

MSB 00000078145

Customer Adoption - It gets good enough

Using today's server requirements, Linux is a credible alternative to commercial developed servers in many, high volume applications. The effect of this on our server revenue model would be immense.

Our client-side revenue model is still strong however for a variety of reasons including switching costs for the entire pool of win32 source code. Linux advocates, however, are working on various emulators and function call impersonators to attack this cost.

This points back to an obvious solution - innovation in the core platform is an ongoing requirement.

Channel Adoption

The "Open Source Software" paper has a section on OSS business models. Summarizing that section, there are 4 primary business models we have identified for Open Source Software.

1. **Secondary Services** - The vendor / developer of OSS makes their money on service contracts, customer integration, etc.
2. **Loss Leader for Market Entry** - The vendor / developer of OSS uses OSS's process advantages (in particular credibility) as a lever against established commercial vendors.
3. **Commoditizing Downstream Suppliers** - The vendor / developer of OSS is also the producer of a product / service further in the value chain and closer to the consumer.
4. **Standards Preemption** - Because OSS process are argued to be winner-take-all, it may suit the vendor / developer to seed the OSS market with their codebase to preempt a competitive codebase from taking hold.

IBM Adopts Linux?

IBM is most capable at capturing revenues from all 4 of the business models associated with Linux.

1. **Secondary Services** - IBM is very strong in consulting, integration, support, etc. This is their fastest growing business today.
2. **Loss Leader** - IBM's client/low-end operating system business is in shambles (remember OS/2?). Additionally, IBM has stumbled on various NC/JavaOS systems as well. By leveraging Linux's credibility (as well as applying IBM's development resources toward improving ease of use?), IBM would hope to upset the status quo in the volume OS space and hope to capture revenue in the ensuing disruption.
3. **Commoditizing Downstream Suppliers** - As a PC/Hardware OEM, IBM's margins increase by commoditizing a key cost item - the OS. In particular, the commoditized & highly customizable qualities of the Linux OS actually provide greater differentiation for hardware vendors.

MICROSOFT CONFIDENTIAL - PAGE 26

**HIGHLY
CONFIDENTIAL**

**MSB 0078146
CONFIDENTIAL**

4. **Standards Preemption** - The standard to pre-empt is anything Microsoft - in particular new OS services that we integrate directly into future versions of NT.

IBM, despite their Apache announcements, seems unlikely to advocate this in the short run. I'd imagine that religion within their various OS development efforts alone would provide a significant amount of near term inertia.

Sun Adopts?

Sun's rationale for adopting Linux would be less encompassing than IBM's.

1. **Secondary Services** - Sun is not very strong in consulting / integration revenue. They do, however, make significant revenue in support and maintenance.
2. **Loss Leader** - Sun could market Linux as a low-end OS and try to make money in the UNIX applications space above it. Because Linux could potentially be a far larger market than anything Sun is accustomed to, this would be a net positive for them.
3. **Commoditizing Downstream Suppliers** - Sun is also a hardware vendor (with some excellent systems). Sun would lose their current OS revenue but the ability to sell their hardware into a broader channel could be compelling.

Linux adoption, however, puts Sun at significant risk if their SPARC operations cannot keep up with Intel's innovation pace.

4. **Standards Preemption - Beat Microsoft Standards**

PC OEM's

Other worst case adoption scenarios are subsets of the Sun / IBM case and involve other PC vendors such as Compaq and Dell.

Note, however, that Compaq and Dell merely have to credibly threaten Linux adoption in order to push for lower OEM OS pricing.

Server ISVs

One interesting spin on the "Commoditizing Downstream Suppliers" strategy could be **backward integration** by server ISVs. For example, Oracle could ship a version of Parallel Server for Linux that *includes the Linux OS within the distribution*.

This is basically a play on the thin-server concept. Instead of *integrating* multiple small business functions on a single server, this attempts to *disintegrate* the features of an enterprise OS into the minimal set necessary to run the specific server application. It plays into the business models identified as follows:

1. **Secondary Services** - Companies like Oracle/SAP/Baan/etc. already make a large percentage of their income from on-site consulting agreements
2. **Loss Leader** - treating the OS as a loss leader helps them concentrate revenues for a particular hardware unit into their hands

MICROSOFT CONFIDENTIAL - PAGE 27

HIGHLY
CONFIDENTIAL

MSB 0078147
CONFIDENTIAL

MSB 000000078147

3. **Downstream commoditization** - Oracle has no problem declaring the Server OS as a kernel, memory manager, IP stack, and some disk.
4. **Standards Preemption** - beat Microsoft.

Next Steps & Microsoft Response

A lot more thought and work needs to go into formulating Microsoft's response to Linux. Some initial thoughts on how to compete with Linux in particular are contained below. One "blue sky" avenue that should be investigated is if there is any way to turn Linux into an opportunity for Microsoft.

A more generalized assessment of how to beat the Open Source Software process which begat Linux is contained in the "Open Source Software" document.

Beating Linux

Beat UNIX

The single biggest contributor to Linux's success is the general viability of the UNIX market. Systematically attacking UNIX in general helps attack Linux in particular. Some Linux-targeted initiatives in this space (not a comprehensive list) include:

- **Improve Low-End "IAM"** - Scaleability, Interoperability, Availability, and Management (SIAM) are the most often cited reasons for using UNIX over NT in mission critical, high-end applications.

In today's Linux deployments however, scaleability is not the driver as much as Interop, Reliability, and Headless Management.

- **UNIX services for NT Add-on pack**

Modularize / Embed Windows NT

Relative to other UNIX's Linux is considered more *customizable*. Addressing this functionality involves more than just the embedded Windows NT project. Greater componentatization & general dependency reduction within NT will improve not only it's stability but also the ability of highly skilled users/admins to deploy task-specific NT installations.

This requires:

- Wide availability of the Embedded NT toolkit
- Greater focus on ease-of-use in the toolkit

Beat commodity protocols / services

Linux's homebase is currently commodity network and server infrastructure. By folding extended functionality into today's commodity services and create new protocols, we raise the bar & change the rules of the game.

MICROSOFT CONFIDENTIAL - PAGE 28

HIGHLY
CONFIDENTIAL

MSB 0078148
CONFIDENTIAL

Some of the specifics mentioned in the OSS paper:

- **DNS integration with Directory.** Leveraging the Directory Service to add value to DNS via dynamic updates, security, authentication
- **HTTP-DAV.** DAV is complex and the protocol spec provides an infinite level of implementation complexity for various applications (e.g. the design for Exchange over DAV is good but certainly not the single obvious design). Apache will be hard pressed to pick and choose the correct first areas of DAV to implement.
- **Structured storage.** Changes the rules of the game in the file serving space (a key Linux/Apache application). Create a compelling client-side advantage which can be extended to the server as well (e.g. heterogeneous join of client & server datastores).
- **MSMQ for Distributed Applications.** MSMQ is a great example of a distributed technology where most of the value is in the services and implementation and NOT in the wire protocol.

Leverage ISV's for system improvements

A key long term advantage that Linux will enjoy is the massive pool of developers willing to improve areas of the core platform. Microsoft will never be able to employ a similar headcount.

A key mechanism to combat this is to make it easy (and provide incentives) for ISV's to extend system components in NT for custom, vertical applications. One example here could be Veritas' specialized file system drivers for NT.

"WinTone"

Linux's modularity and customization also implies inconsistencies in services available on an arbitrary Linux installation. Microsoft can provide a bundle of services that are universally available in all OS releases (current initiatives include WBEM-based management) that generate network externalities when combined across many devices in the network.

Put another way, the extreme modularity of Linux devalues what a "Linux-logged" app means. By contrast, Window's monolithic nature gives an app developer more leeway in terms of what API's are callable.

Process Vulnerabilities

Where is Microsoft vulnerable to Linux? As stated earlier, the primary threat resides on the server vs. the client.

Linux will "Cream Skim" the Best NT Server Features

The Linux community is very willing to copy features from other OS's if it will serve their needs. Consequently, there is the very real long term threat that as MS expends the development dollars to create a bevy of new features in NT, Linux will simply cherry pick the best features and incorporate them into their codebase.

The effect of patents and copyright in combatting Linux remains to be investigated.

MICROSOFT CONFIDENTIAL - PAGE 29

**HIGHLY
CONFIDENTIAL**

**MSB 0078149
CONFIDENTIAL**

Linux is recreating the MS "3rd release is a charm" advantage -- FASTER

Microsoft's market power doesn't stem from products as much as it does from our iterative process. The first release of a Microsoft product often fails poorly in the market and primarily of generates fine granularity feedback from consumers. Similarly, Linux has shown that they are capable of iterative cycles -- but at an order of magnitude faster rate. On the flip side, however, our incremental releases are arguably much larger whereas many of Linux's incremental releases are tantamount to pure bug fixing.

Revision History

Date	Revision	Comments
8/03/98	0.95	
8/10/98	0.97	Started revision table Added reference to SUN's non-commercial license Added Linus quote for desktop vs. server issues
8/11/1998	1.00	Added perf charts, published first release.

MICROSOFT CONFIDENTIAL -- PAGE 30

HIGHLY
CONFIDENTIALMSB 0078150
CONFIDENTIAL

MSB 00000078150