

Our Linux Strategy

VinodV
5/19/99

Microsoft Confidential; © 1999, All Rights Reserved Do Not Forward without Approval from Author.

Introduction

This document discusses both our strategy and our plans for competing with Linux. To understand the strategy it is important to remember the following:

- Linux isn't most importantly a product/feature; it's a philosophy change
- Linux has no new specific features to co-opt
 - Unlike the NC: the NC touted TCO benefits, and thus we introduced ZAK/ZAW
 - Unlike the Internet: the Internet was loaded with technology changes, and thus we invested in browser technologies and reexamined all our existing products

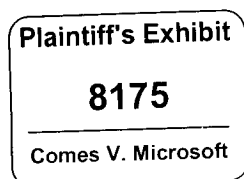
The core strategic thrust of Linux is NOT an attack against some product/feature weakness of Microsoft. It's an attack at the base of the commercial software industry -- *Intellectual Property*.

Previous threats to Microsoft (the NC, Java, etc.) have been about replacing Microsoft's IP with another company's IP that claimed some new benefit (e.g. TCO). What differentiates Linux is that OSS attempts to extricate Intellectual Property all together.

Since many people have proposed how to deal with Linux, we thought it might be helpful as a thought exercise to quickly examine some of the alternative strategies we could consider. For each we include the "fatal flaws" that make them untenable.

1. **Embrace Linux: MS APIs / Linux kernel** -- release an MS version of Linux and/or release key MSFT platform technologies on Linux (e.g. parts of Win32, app server, etc.)
Pros: Ride the wave & try to evangelize Win32
Cons: Dramatically evangelizes Linux & may risk MSFT IP due to GPL license issues
Fatal Flaw:
 - Impossible to make this revenue neutral with Windows biz.
 - Doesn't protect the "crown jewel" IP from being targeted at a later date
2. **Embrace Linux: Linux APIs / MS Kernel** -- try to get Linux APIs on Windows -- get more hardcore about POSIX subsystem on NT to capture Linux app base
Pros: Capture some of the Linux dev mindshare by making it easy to bring Linux apps to NT
Cons: Hurts Win32 evangelization
Fatal Flaw:
 - There are no Linux apps that we covet.
3. **Embrace Open Source: Publish NT Source** -- release NT source code under a license similar to Sun's community source license
Pros: Try to capture Linux's evangelization benefits by publishing NT source
Cons: ISVs getting hooked on undocumented APIs, support costs, etc.
Fatal Flaw:
 - Microsoft is an IP company. Like the rest of the software industry, >90% of our IP valuation stems from Trade Secrecy of the source code. Open Source is mutually exclusive with Trade Secrecy. This plan would instantly make the various Win32 clones (e.g. <http://www.winehq.com>) an order of magnitude more capable.
4. **Lower the price of Windows** -- release older / stripped versions of the OS for at lower price
Pros: Try to capture people who use Linux due to price sensitivity

Page 1



MS-CC-MDL 000000202974
HIGHLY CONFIDENTIAL

Cons: Building new versions of windows. Long-term support headaches Cannibalization of the "real" windows?

Fatal Flaw:

- Assumes that price is the primary motivator for Linux usage. This has not been borne out in reality.

What are the core strategies that we are going to pursue?

1. Fix our Sins

Linux's most immediate contribution is highlighting our sins in some key market segments. There are already (large) investments in the company spun up to deal with our most pressing concerns such as reliability; remote admin; etc so we won't spend further time describing them here. It is critical that we make progress in these areas

Currently, Linux gains horsepower due to VASTLY exaggerated negative claims about our abilities and corresponding VASTLY underreported positive claims about our innovative work. We must reverse the "conventional wisdom" that UNIX is technically superior to NT which is the foundation for Linux marketing. In most ways, NT is superior & the technical message needs to get out.

2. Innovating, Creating New IP

(Re-)recognize that we are an IP company and that in our networked world, functionality delivered via protocols is steadily replacing functionality which was once delivered via APIs Thus, innovation must occur both internal to our products, but also *between* computers.

Windows clients must always be able to communicate with Linux servers (and vice-versa). However, there MUST be additional value created when a Windows machine is touching another Windows machine. NOT doing this is akin to giving away the Win32 APIs. Every group defining protocols needs to remember this. Some core initiatives that are excellent demonstrations of this are:

Management -- Deep, rich WMI instrumentation is an area where Windows and Win32 apps must excel. In addition to IP boundaries, Linux's development methodology makes this difficult for Linux to provide leadership in breadth & uniformity of coverage/implementation.

Storage -- Rich, structured, remotable, querable storage dramatically raises the bar versus today's basic file system functionality. The benefits to client application vendors & server vendors are numerous and well detailed in other presentations.

These areas demonstrate functionality that IT managers -- once they've tasted it -- will (hopefully) find compelling enough to mandate across as many systems within their computing universes as possible. Letting our protocols become commoditized is a recipe for failure. We must innovate and keep our great advancements to ourselves. The fine balance between protecting/financing our innovations and interoperability will get more difficult overtime But, it is relatively easy today.

Outside of protocols we need advancements throughout the system. Advances in file formats (e.g., the disk structure), technology such as security, etc. are areas that are critical for us to innovate. We need to accelerate patenting every invention

3. Form Factor Proliferation

This is a well-discussed area. Obviously PC's will not be the exclusive center of computing in the near future and this addition to the OS requirements list provides the opening for low innovation competitors such as Linux in. We need to spread our technology everywhere And where we don't have our OS present, we need to ensure the protocols are not IP latent and in fact open for us to use.

4. **Compete with Linux Head-On**

BED marketing is currently making the transition towards engaging Linux as a tier-1 competitor in the server & client markets. There are still some decisions to be made here (and headcounts to fill) to ensure that on a tactical basis, NT out markets Linux. Some of the core deliverables include white papers, benchmarks, etc. More peripheral questions / issues include reclaiming retail shelf-space from Linux, etc. We need engagement throughout the company (e.g., retail) on this. Finally, getting the word out on NT's architectural advantages over Linux is an imperative.

5. **Getting Credit for the Openness and Availability of our Sources**

One of the key lessons learned from the Linux OS is the power of the Open Source model with respect to creating passionate, technically savvy development communities around a body of code. Reclaiming the hobbyist developer / "scratch an itch" developer communities is paramount for us (they were the original "long hairs" who introduced the PC to corporate America). While we may never be able to fully detract from Linux's energy in this space, it is very important for us to focus our TREMENDOUS developer relations assets into this new "channel." JimAll presented a plan at the 3yr review that involved a 2-pronged attack on this channel:

- a. **Depth Licensing** -- Ramping up full, formal source code licenses to ISVs/IHVs/Corps etc. by at least a factor of 10 vs. today's efforts.
- b. **Breadth Licensing** -- Reorganizing & creating new widely licensed, derivable, redistributable source code bases hosted on web sites targeting specific Win32 developer niche's (e.g., ResKit level functionality).

6. **Securing our Current & Future IP**

Once again, the core of the Linux phenomena -- and the #1 reason it tries to claim the "glow of inevitability" -- is its aversion to Intellectual Property (IP). Obviously, in terms of economic effects, IP is on par with motherhood & apple pie in its role in the world economy.

The belief that the "Open Source" pie will eventually gobble up ingredients from all the other pies is more dangerous to us & the software industry than the current Linux product

Open Source development is the greatest cloning machine of all time. Consequently, we must recognize that "Trade Secrecy" of source code will provide increasingly minimal protection over time and that aggressive patent procurement is our only investment defense. Additionally, strong patent procurement is a key enabler which allows us to publish more of our source code to leverage evangelization benefits (the patent application process is, in a manner of speaking, a form of source publication)

Initiatives (NOT discussed further in this paper) are underway to understand the options in this space.

Immediate Next Steps:

The following are all underway:

1. Ramp-up / staff Linux competitive marketing efforts.
2. Ramp-up source licensing initiatives. DRG/MSDN is the owner for the umbrella but all component teams must begin evaluating what codebases would benefit the platform if they were evangelized via less restrictive licensing.
3. More proactively & aggressive secure patent rights to MSFT innovations that will be significant to the OSS fight. Development teams must shift mindsets from source code secrecy towards patents as the primary means of securing our key innovations.
4. [on-going] Create new IP in base scenarios – file sharing, management, etc.

Please direct any questions / discussion to VinodV