

# **Why Linux can help Make Windows Sales (and Vice Versa)**

**James Bottomley  
SteelEye Technology**

**26 May 2006**

# Introduction

- Driver writing can be a transferrable skill set.
- Being Operating System Agnostic can also be a good thing.
  - Both for a Company
  - and for an individual
- So is there really a great divide between Windows and Linux?

## Windows vs Linux: The Great Battle?

- This is very much a Marketing Myth.
- The reality depends how susceptible you are to marketing
  - Coolaid believed most readily at the top of any company
  - and least at the bottom

## Is this really true?

- Most companies claim to be a “windows shop” or a “unix shop”
- but in reality, they always have some of each
  - Exchange for email, apache for webserver
  - Unix in the lab and the data centre but windows on the desktop
  - and so on
- Companies trust suppliers who respect their choices
  - So the business goes to the supplier who provides for both Linux and Windows

## Philosophical Differences

- From the OS Kernel Perspective there are big differences
  - Driver model
  - API
  - Almost everything functional
- But from a programming philosophy point of view, discipline is fairly similar
  - Security
  - Memory Management
  - kernel coding technique

## Case Study: SteelEye Replication

- Have two replication products
  - Both called LKDR (LifeKeeper Data Replication)
  - One for Linux is fully open sourced (based on `md` and `nbd`).
  - One for windows is a closed source filter driver.
- Problem was that Windows Driver performed much worse than Linux on low Bandwidth, high Latency lines.
  - On T3 Linux gets 90% utilisation.
  - Windows gets  $\leq 20\%$

## Analysing the Problem

- Looks simple: Windows doesn't pack the pipe.
- Windows product is single threaded
  - sends one packet on to the wire
  - waits for the ack to return from the remote before sending another
- Linux product is fully multi-threaded and can have multiple in-flight packets.

## Solving The Problem

- Looks simple, just enhance the windows product to have multiple packets in-flight.
- Do this by moving our Linux Driver expert (and nbd Maintainer) over to the Windows team to work on it.
- So how to do this while avoiding licensing issues, particularly with the GPL?



## Licensing Issues

- The GPL (and all Open Source Licences) are Copyright Licences.
- Forbid the copying of expression (not ideas) from an Open to a Closed Source base
- Have to be careful about cutting and pasting
- But realistically, Linux and Windows Kernel API is so different that direct cut and paste is usually impossible.

# Implementation

- Actually chose a different implementation in windows from `nbd`
- Used dual threads and queues for transmit and receive
- Project took about two months to complete
- At the end, took it into the lab to measure performance and got ...

## The results

25% Utilisation

## Back to the Drawing Board

- More analysis of the problem
- Issue turned out to be that the network transport is a Windows named pipe
- Protocol is essentially synchronous
- Therefore the entire windows IP stack is working against us.
- Should have realised this because “protocol accelerators” are big business in the Windows space.

## The Final Solution

- Need to eliminate the named pipe
- However, very difficult to tap into the windows TCP stack at the right level.
- Finally attached as a protocol provider to TDI.
- Added an extra month to the development
- But in the end, utilisation went up to ...

## The results

90% Utilisation

## Observations

- Even though the problem moved while it was being solved, queueing work was necessary to the final solution (just not sufficient)
- Project plan was drastically shortened by moving Linux Expertise onto the project
- Result was ultimately successful: Now have completely comparable capabilities over low bandwidth high latency lines for both products.