TURBOL NUX.

Gotchas of Starting Again: Building Linux for the Enterprise Backend

John Terpstra VP, Community Relations April 26, 2000

TURBOL NUX.





Origins of TurboLinux

Pacific Hi-Tech started distributing CD-ROMS

Company name changed to TurboLinux in June 1999

First TurboLinux product based on Red Hat distribution

Met Cliff Miller in September, 1998

Commenced discussing the workstation and server products in September, 1998

Commenced product split into Workstation and Server

Questions, Questions

Why put games on a server product?

Why put a proxy server (squid) on a workstation product?

An opening

MS BackOffice demonstrated a spot in enterprise backend

What functionality would be required?

Workstation product would be targeted at developers

Spirit of co-opetition rather than competition – a key differentiator of Linux

The Birth

TurboLinux Server was born

Would provide a one-to-one functionality match to Windows NT Back Office

File/Print

Web-serving

SMS Enterprise wide

♦SQL

Enterprise Messaging



Infancy

TurboLinux Server 1.6 marketed as TurboLinux Server 4.0 Released November, 1999

Based on kernel 2.2.10

EGCS compiler

♥glibc 2.0.7▶ glibc 2.1 had already been released

Why?

- Questions asked about using glibc 2.0.7
- Outsiders considered it a liability
- It used a character-based installer
- Market perceived us as horribly backwards
- We were not considered up to date
- Definitely not considered a bleeding edge distribution

Here's Why We Used glibc 2.0.7



- Best known
- Applications worked on it
- Thoroughly tested
- Customers ran for over a year with no downtime (still running!)
- Only change was kernel
- •Use of established product shows enterprise stability

The Problem

Dilemma

Application developers and ISVs demanded use of glibc 2.1

Some distributions using glibc 2.1 ended up with broken products

Small Challenge

Marketing group requested recompilation using glibc 2.1

Development team was given 2 weeks to produce

Problems

✤4.0 migration to glibc 2.1 would be called 4.1

Work started on 5.0 concurrently (new product specs)

Server 4.0 couldn't recognize some ATAPI CD-Roms

Change to gcc-2.95 would fix that

Gcc-2.95 was used at the same time as change to glibe 2.1

Results

2 week plan turned into 5 months of total development resources

917 source packages make up Server and Workstation

♦800+ could not be recompiled with glibc 2.1/gcc 2.95

Required major re-engineering

 \mathbf{P} Rather than toss out 4.1, work began to merge 4.1 and 5.0

Results cont...

◆4.1 + 5.0 would result in 6.0

- Opportunity to start again would result in being on the leading edge
- Numeric change would avoid confusion in the marketplace
- Estimated at 8 weeks work
- October 6, 1999 Beginning of 6.0
- Self-hosting (means we could compile most of 6.0 from within itself)

Barnades

glibc / kernel / compiler had overlapping header-files

Differences between headers caused problems compiling applications

October 15 - able to compile approximately 785 packages

We were still a long way from home

Product integrity has a high price

We must be vigilant about development

Time I ssues

Glibc 2.1.2: a 22,000 line patch (from glibc 2.1.1)

Had to rebuild glibc libraries over 70 times

Rebuilt kernel 40 times

Rebuilt compiler 35 times

Each iteration required recompiling EVERYTHING

Time, cont...

Started with 4.0 build product and kept rebuilding

7 complete rebuilds

Had to eliminate backwards references in libraries and executables

Stumbling Blocks

Some Open Source packages do not allow building with file system layout that complies with FHS

Some package could be manipulated into FHS compliant file system layout by moving files around but other packages would not allow this.

Call to Developers

The Call

Please can we all work together?

We need to modify Autoconf use so that:
This would make it easier for all distributions to build applications with FHS compliance

- Every application needs test suite to validate integrity of applications
- We need developers to work with TurboLinux (Call to Arms)