*BSD is dying

- Anonymous Coward, Slashdot


Jordan Hubbard
Apple Computer, Inc.
Oh really?

Let’s look at some stats...
FreeBSD Users: 2.5 Million

Server Installations (Netcraft)
Mac OS X Users: 12 Million
Applications: 12,000 Mac OS X Native
Since the arrival of Mac OS X, BSD has become the biggest desktop UNIX variant on the planet.

Yes, even bigger than Linux

Take *that*, Anonymous Coward!
Selective overview of Mac OS X

I HAVE UNIX UNDERPinnings!
I HAVE UNIX UNDERPinnings!

BIG DEAL.

UNIX UNDERPANTS??

"CHMOD 700 *
BABY."
OS Foundation

Open Source “Darwin” base
OS Foundation

User
Kernel

BSD Kernel

FileSystem
Network

Process Management
BSD Kernel

- FreeBSD 5.1 based (networking, vfs, filesystems, etc)
- Unified Buffer Cache (different than FreeBSD’s)
- Clustered I/O performance enhancements
- Local File Systems
  - hfs, ufs, iso9660, udf, fat, ntfs
- Network File Systems
  - nfs, afp, smb, webDAV, ftpfs, afs
OS Foundation

User Kernel

Mach Kernel
VM
Scheduling
IPC
Mach Kernel

• Based on Mach 3
• VM, tasks, threads, scheduling and IPC
• Fine grain locking for SMP
• Support for > 4GB Physical memory
• [fairly] Light-weight threading model
• Real-time scheduling
• Event driven application programming model (via Mach ports)
OS Foundation

User
Kernel

IOKit
Driver Families
Drivers
IOKit

• Written in conservative C++
• OOP device family and instance model
• Support for user space drivers
• Dynamic plug and play
• Handles all device property information and provides convenient introspection via ioreg(1) and friends
• Sophisticated power management
OS Foundation

BSD Commands and Libraries
Commands and Libraries

- Standard commands and libraries from FreeBSD 5.1
- A full suite of scripting languages
  - perl, tcl, python, ruby, php
- Every standard shell
  - bash, csh, tcsh, zsh, etc
- Standard editors
  - pico, vi, emacs (the best!)
- Standard C compiler suite
  - gcc, g++, Objective-C [currently at version 3.3]
Mac OS X Architecture

- Applications
- User Interface
- Application Frameworks
- Graphics and Media
- System Services
- OS Foundation
Open Directory

• Flexible plug-in architecture
  – Supports legacy flat files
  – Supports OpenLDAP
  – Supports Active Directory

• Open Source
Security Server

- Full CDSA (Common Data Security Architecture) implementation
- Manages certificates, keys, passwords
- Implements keychains for easy access
- It’s not OpenSSL
- Open Source references:
  - http://sourceforge.net/projects/cdsa/
Rendezvous

• Service registration
• Service discovery
• Easy ad-hoc networking via .local namespace
• Also available for FreeBSD, Solaris & Linux (and a number of misc devices)
• Open Source references:
Mac OS X Architecture

- Applications
- User Interface
- Application Frameworks
- Graphics and Media
- System Services
- OS Foundation
2D Graphics: Quartz

- PDF-based imaging model
- Leverages GPU
- CUPS “WYSIWYG” printing
- Python bindings
  - CoreGraphics APIs
  - QuickTime images
  - PDF, RTF, HTML
3D Graphics: OpenGL

Industry Standard 3D Technology

Keeping the world safe from DirectX
And we all know why good 3D support is especially important...
And optimized video...
H.264

Similar data rates, very different picture
Core Audio / Core MIDI
Guaranteed latency and Audio Units architecture make apps like GarageBand possible
Mac OS X Architecture

- Applications
- User Interface
- Application Frameworks
- Graphics and Media
- System Services
- OS Foundation
Mac OS X Architecture

Applications
User Interface
Application Frameworks
Graphics and Media
System Services
OS Foundation
Most of the important ones...

• Microsoft Office
• Photoshop
• Quicken / Quickbooks
• Quark Xpress
• Macromedia Director and Macromedia Studio
• ... and many many more, either here or coming soon
What makes us BSD users?

- **The historical drive to innovate**
  - Berkeley Fast Filesystem
  - TCP/IP networking
  - Virtual Memory
  - Long filenames
  - Job control
  - ... and many other things aimed at making it the first “usable” UNIX, and one that everyone wanted
What makes us BSD users?

- Two words: Software Engineering!
  - Strong organizational mentality. “Of course anyone can build the entire system!”
  - Dedication to quality and (sometimes aggressive) peer-review
  - Use of proper tools (like cvs) is a given
  - Unity of purpose on mission goals (more on that later)
BSD - who’s who

- FreeBSD
- NetBSD
- OpenBSD
- Mac OS X (Darwin)
- DragonFly BSD

... and numerous commercial and research variants

The Post-BSD 4.4 world
So clearly, BSD is NOT dying...

But what lies ahead?
Challenge: Authentication

Sorry, this is not a user information database:

- nobody:*:-2:-2:Unprivileged User:/usr/bin/false
- root:*:0:0:System Administrator:/var/root:/bin/sh
- daemon:*:1:1:System Services:/var/root:/usr/bin/false
- smmsp:*:25:25:Sendmail User:/private/etc/mail:/usr/bin/false
- lp:*:26:26:Printing Services:/var/spool/cups:/usr/bin/false
- postfix:*:27:27:Postfix User:/var/spool/postfix:/usr/bin/false
- www:*:70:70:World Wide Web Server:/Library/WebServer:/usr/bin/false
- mysql:*:74:74:MySQL Server:/var/empty:/usr/bin/false
- sshd:*:75:75:sshd Privilege separation:/var/empty:/usr/bin/false
The present looks a lot more like this ...
Challenge: Authentication

• Face it: It’s a Windows world - “Embrace and extend!”
• The traditional UNIX group model is obsolete
• The uid is obsolete and insufficient - prepare for the GUID (and privacy concerns)
• Smart Cards are in your future...
Challenge: Authentication

- ACLs: We have them, people want them, now what?
- ACL interoperability - will there be any?
- Ha! The resource fork is back! POSIX Extended Attributes:
  - A challenge for the command line
  - A challenge for NFS and non-EA aware local File Systems
Challenge: API Stability

• “Hey man, just recompile it” is NOT an evolutionary API strategy:
  – APIs need to be clearly classified (supported, unsupported, unstable, marked for death, etc) in header namespace and doc
  – Shared library version numbers aren’t proving to be sufficient
  – Current linker toolchain may not be sufficient either
Challenge: API Stability

• Restricted Kernel APIs are essential:
  – Developers like to poke into the innards, but this can strongly inhibit innovation
  – “Just recompile” really not an option here
  – Things like /dev/kmem are evil and should die (and will someday in Mac OS X)

• Proper kernel abstraction can help both the OS vendor and its 3rd party hackers
Challenge: UI + Applications

• The X Window System still sucks
• UI toolkit standardization still balkanized
• GPUs are getting faster and need to be better leveraged in the service of the interface
• Support for printing and good I18N support still lacking
• High level APIs - Libc isn’t the place to stop
Challenge: Administration

• Administering clusters of machines is still too “bolt something together yourself”
• BSD has been slow to embrace unifying standards for configuration data (XML, get over it already!)
• Not all centralized configuration databases need to be a Windows Registry
• Service management is crude
Challenge: Hardware Trends

• Floating point performance is a more critical comparison point now
• Still a lot of low-hanging fruit in the math libraries, but very little work on them
• Hand-optimization bang-for-buck in computational problems like BLAST can be huge, and biotech is the future!
• Yes, Altivec sells a lot of XServes
• Don’t underestimate the GPU in the future
Challenge: Hardware Trends

- Embedded systems are on the rise
- A lot of embedded CPUs omit MMUs to cut cost: This is one area where BSD support is really lagging behind
- BSD’s organizational ethos coupled with an embedded systems focus could be a huge win
- Don’t just hand this space to Linux, NetBSD folks!
Challenge: Standards

• More and more government shops are demanding Common Criteria certification. A little audit support can go a long way!

• UNIX03 compliance is not a bad thing: It’s not just about the ™, it’s about portability

• New accounting laws will soon make standards compliance more necessary in the USA

• OSS strategy: Make it easy for people with money to get certification. It worked for Linux!
In Summary

• BSD is doing great, but it needs strong consensus on its mission goals to be healthy another 10 years
• BSD could also stand to remember its innovative roots. Be more open to and actively encourage “alternative thinking”:
  – http://www.tel.fer.hr/zec/vimage/ - Mr Zec’s network cloning stuff is very cool
  – http://www.citi.umich.edu/u/provos/systrace/ - Maybe not the final implementation, but a good idea
  – Plan9 and Linux have some interesting ideas - look at them! Experiment!
In Summary

• Apple has done a great job leveraging open source, but there are things we want to improve:
  – More effective 2-way collaboration. Not just “pull” but “push”
  – Greater visibility into the OS dev process (particularly with bug reporting)
  – More timely source drops which always match current OS and update version
  – More “co-production” with BSD community, where and when it matches their mission
  – [Your idea here]
Q&A