

Mac OS X

An Introduction for UNIX Users

Leon Towns-von Stauber, Occam's Razor

Seattle BSD Users Group, October 2004

<http://www.occam.com/osx/>

Opening Remarks.....	3
Where Did Mac OS X Come From?.....	5
What is Mac OS X?.....	13
A New Kind of UNIX.....	25
A Different Kind of UNIX.....	28
Why Use Mac OS X?.....	60
Resources.....	63
Closing Remarks.....	67

- This is a technical introduction to Mac OS X, mainly targeted to experienced UNIX users for whom OS X is at least relatively new
 - Some emphasis on comparisons with FreeBSD
- I'm assuming basic familiarity with operating system design
- Where I'm coming from:
 - UNIX user and some-time admin since 1990
 - Full-time UNIX admin since 1995
 - NeXTstep user and admin since 1991
- This presentation covers primarily Mac OS X 10.3.5 (Darwin 7.5)

- This presentation Copyright © 2003–2004 Leon Towns–von Stauber. All rights reserved.
- Trademark notices
 - Apple® , Mac® , Macintosh® , Mac OS® , Aqua® , Finder™ , Quartz™ , Cocoa® , Carbon® , AppleScript® , Rendezvous™ , Panther™ , and other terms are trademarks of Apple Computer. See <<http://www.apple.com/legal/appletmlist.html>>.
 - NeXT® , NeXTstep® , OpenStep® , and NetInfo® are trademarks of NeXT Software. See <<http://www.apple.com/legal/nexttmlist.html>>.
 - PowerPC™ is a trademark of International Business Machines.
 - Java™ is a trademark of Sun Microsystems.
 - Other trademarks are the property of their respective owners.

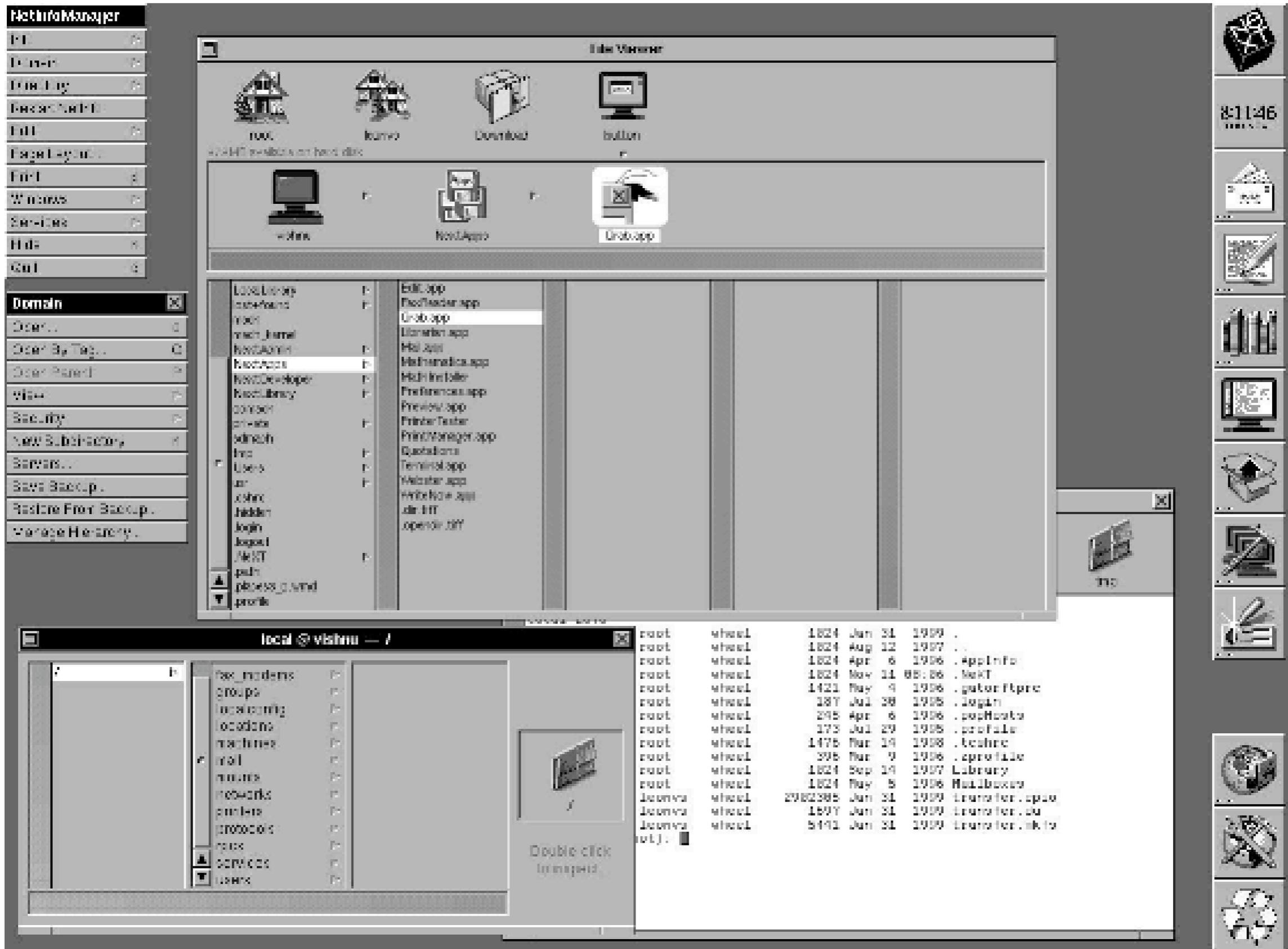
- Apple Computer: The Early Years
- The NeXT Years
- Apple Redux



- The history of Mac OS X is closely intertwined with that of Steve Jobs
- 1975
 - Steve Jobs co-founds Apple Computer
- 1984
 - The Macintosh is released
- 1985
 - Jobs leaves Apple Computer

- 1985
 - Jobs founds NeXT Computer
 - Avadis ("Avie") Tevanian, co-developer of Mach, leads software engineering efforts
 - Jon Rubinstein heads hardware development
- 1988
 - NextStep 0.8
 - NeXT Computer
- 1989
 - NextStep 1.0

- 1990
 - NeXTstep 2.0
 - NeXT Cube, NeXTstation
- 1992
 - NeXTstep 2.2, NeXTSTEP 3.0
 - Turbo systems
 - NeXT RISC Workstation in development
 - Multiprocessor, based on PowerPC 601
- 1993
 - Black Wednesday: NeXT ceases hardware sales
 - NEXTSTEP 3.1, 3.2
 - Intel x86 support added



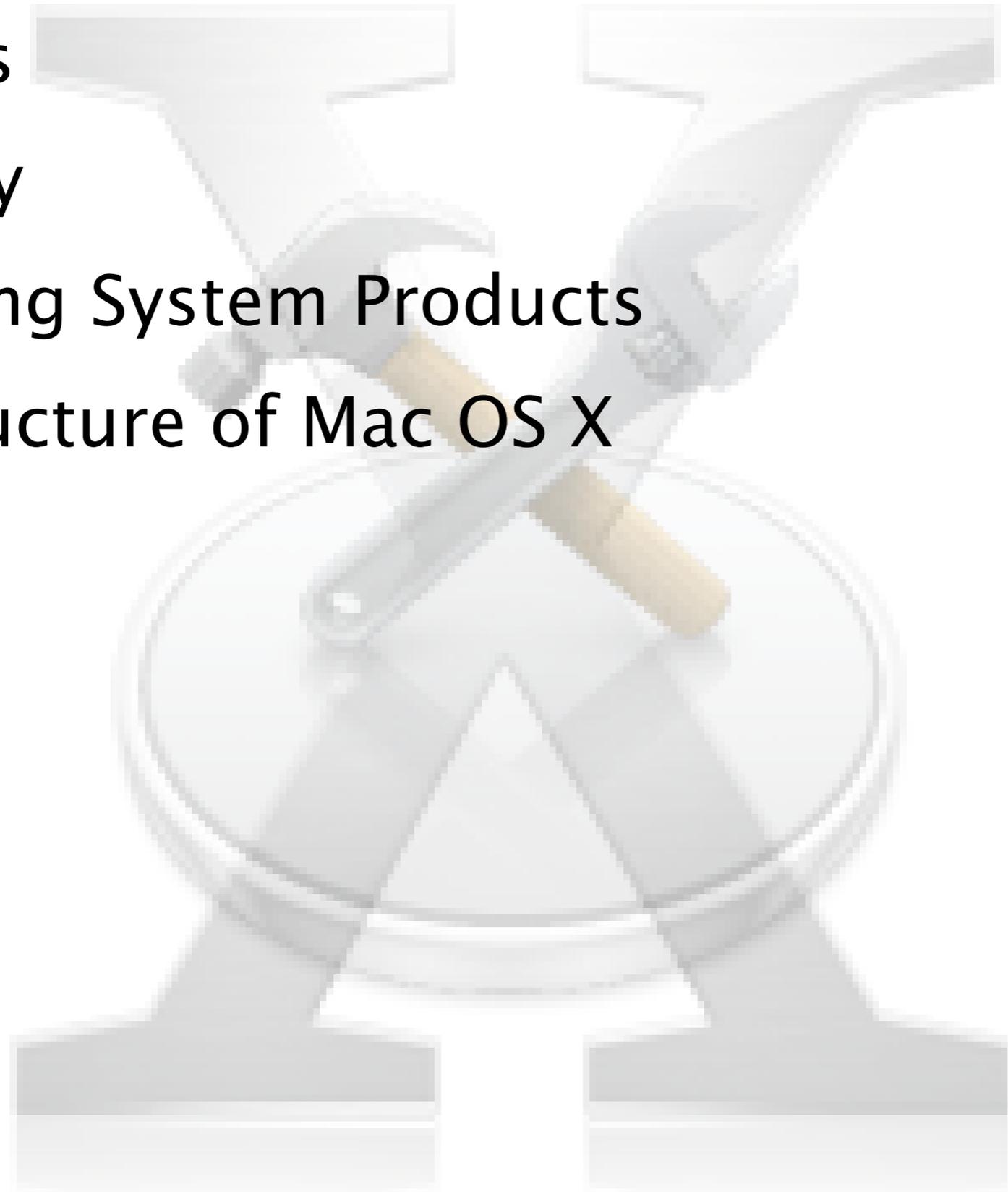
NEXTSTEP 3.3

- 1994
 - NEXSTEP 3.3
 - HP PA-RISC support added
 - OpenStep API specification created (in collaboration with Sun)
- 1995
 - OPENSTEP/Mach 4.0
 - Sun SPARC support added
- 1996
 - OPENSTEP/Mach 4.2
 - NeXT purchased by Apple
 - Rescue attempt for both NeXT and Apple operating systems
 - Jobs returns to Apple as "advisor"

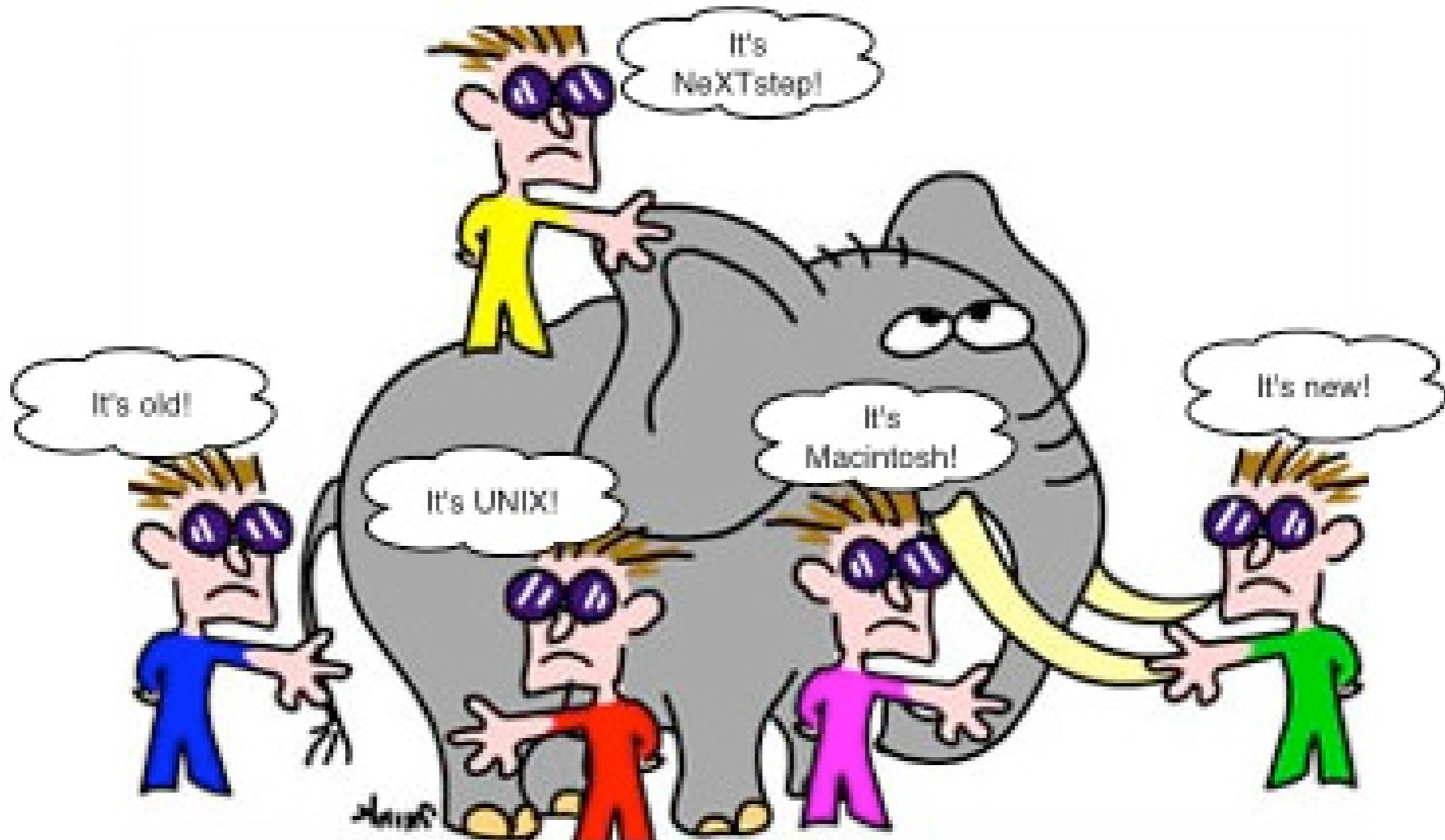
- 1997
 - Jobs made "interim" CEO of Apple
 - Avie Tevanian leads software engineering
 - Jon Rubinstein leads hardware engineering
 - Rhapsody DR1
 - Support for PowerPC and x86
- 1999
 - Mac OS X Server 1.0
 - Support for PPC only
 - Graphical interface based on classic Mac GUI
 - Darwin
 - Mac OS X DP1, DP2

- 2000
 - Aqua graphical interface demonstrated
 - Mac OS X Public Beta
- 2001
 - Mac OS X 10.0, 10.1
- 2002
 - Jordan Hubbard (of FreeBSD Project) becomes Darwin project manager
 - Mac OS X 10.2 (Jaguar)
- 2003
 - Mac OS X 10.3 (Panther)
- 2005
 - Mac OS X 10.4 (Tiger)

- Answers
- Ancestry
- Operating System Products
- The Structure of Mac OS X

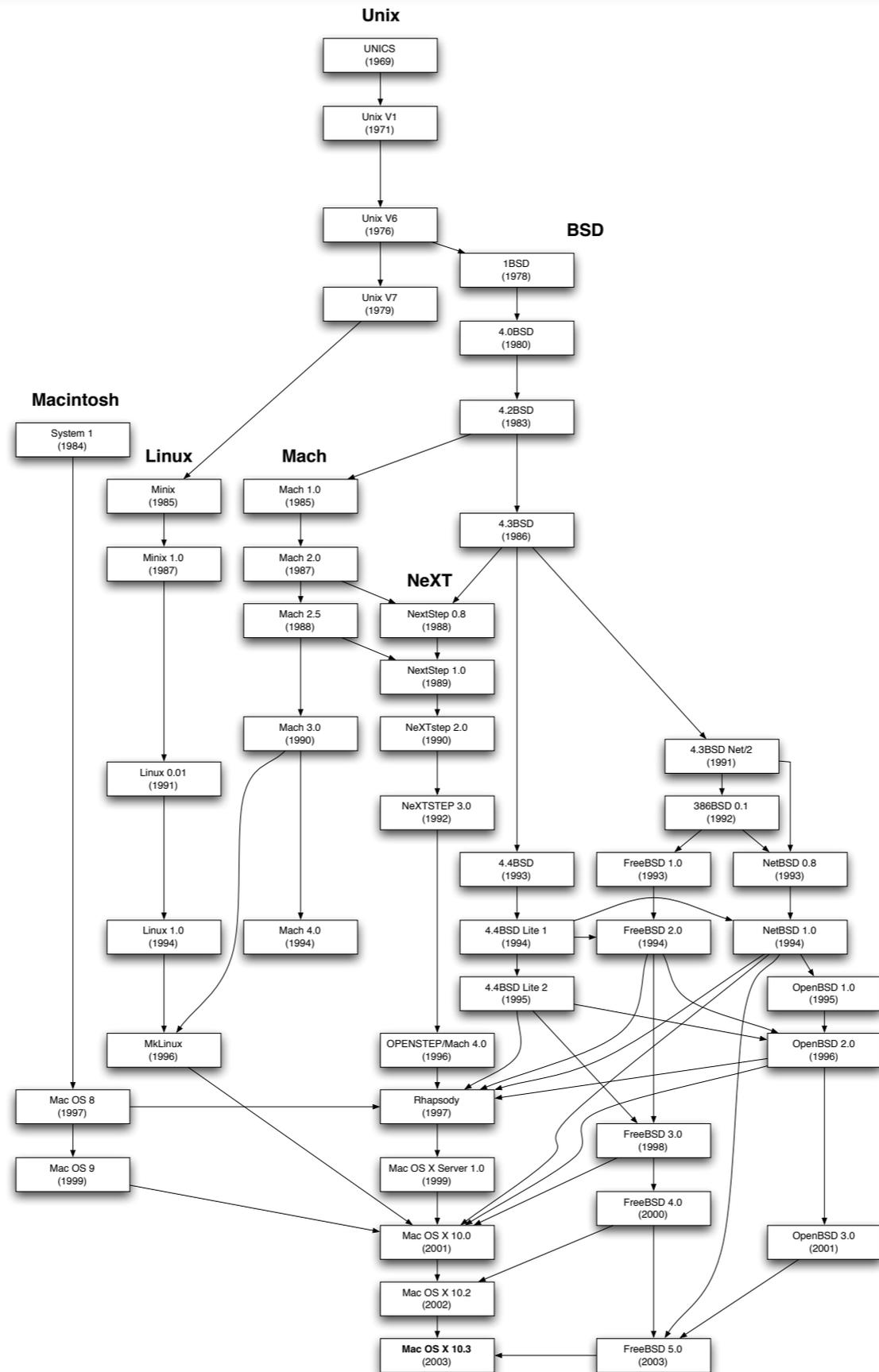


- It's an elephant



- I mean, it's like the elephant in the Chinese/Indian parable of the blind men, perceived as different things depending on the approach

- Inheritor of the Mac OS legacy
 - Evolved GUI, Carbon (from Mac Toolbox), AppleScript, QuickTime, etc.
- The latest version of NeXTstep
 - Mach, Quartz (from Display PostScript), Cocoa (from OpenStep), NetInfo, apps (Mail, Terminal, TextEdit, Preview, Interface Builder, Project Builder, etc.), bundles, faxing from Print panel, NetBoot, etc.
- A new flavor of UNIX
 - More specifically, a BSD UNIX variant
 - Full set of command-line utilities, libraries, server software, etc.
- All of the above



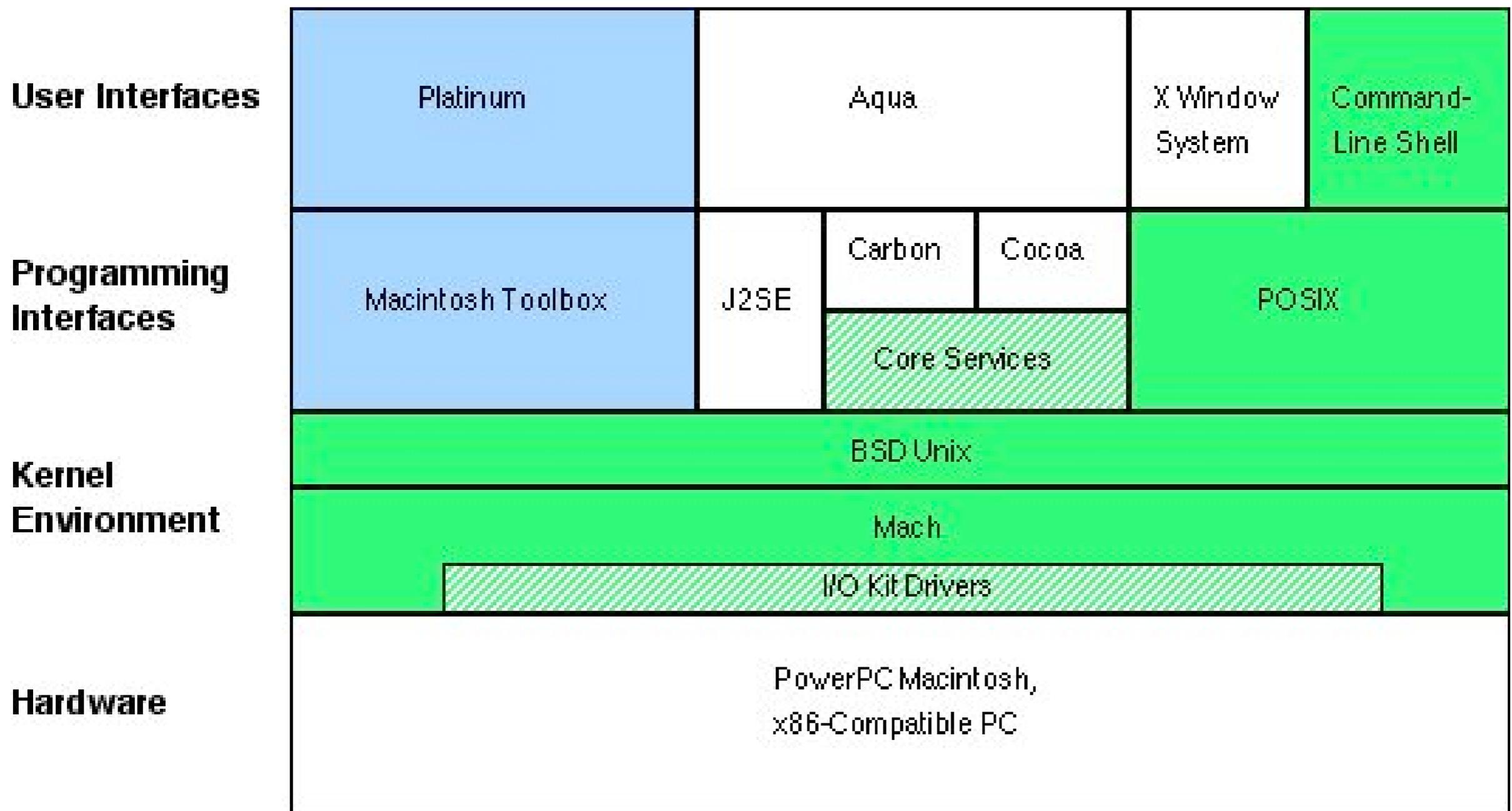
Operating System Ancestry of Mac OS X

- UNIX components primarily based on FreeBSD
 - Also NetBSD and OpenBSD, as well as NeXTstep's version of BSD
- Kernel based on Mach 3.0, MkLinux, and NeXT Mach
- This is not A/UX

- Mac OS X
 - Apple's flagship operating system
- Classic
 - An instance of Mac OS 9 running in a self-contained execution environment within Mac OS X
- Darwin
 - The open-source foundation of Mac OS X
- Mac OS X Server
 - Mac OS X with additional server and administrative software



Hexley, the unofficial Darwin mascot



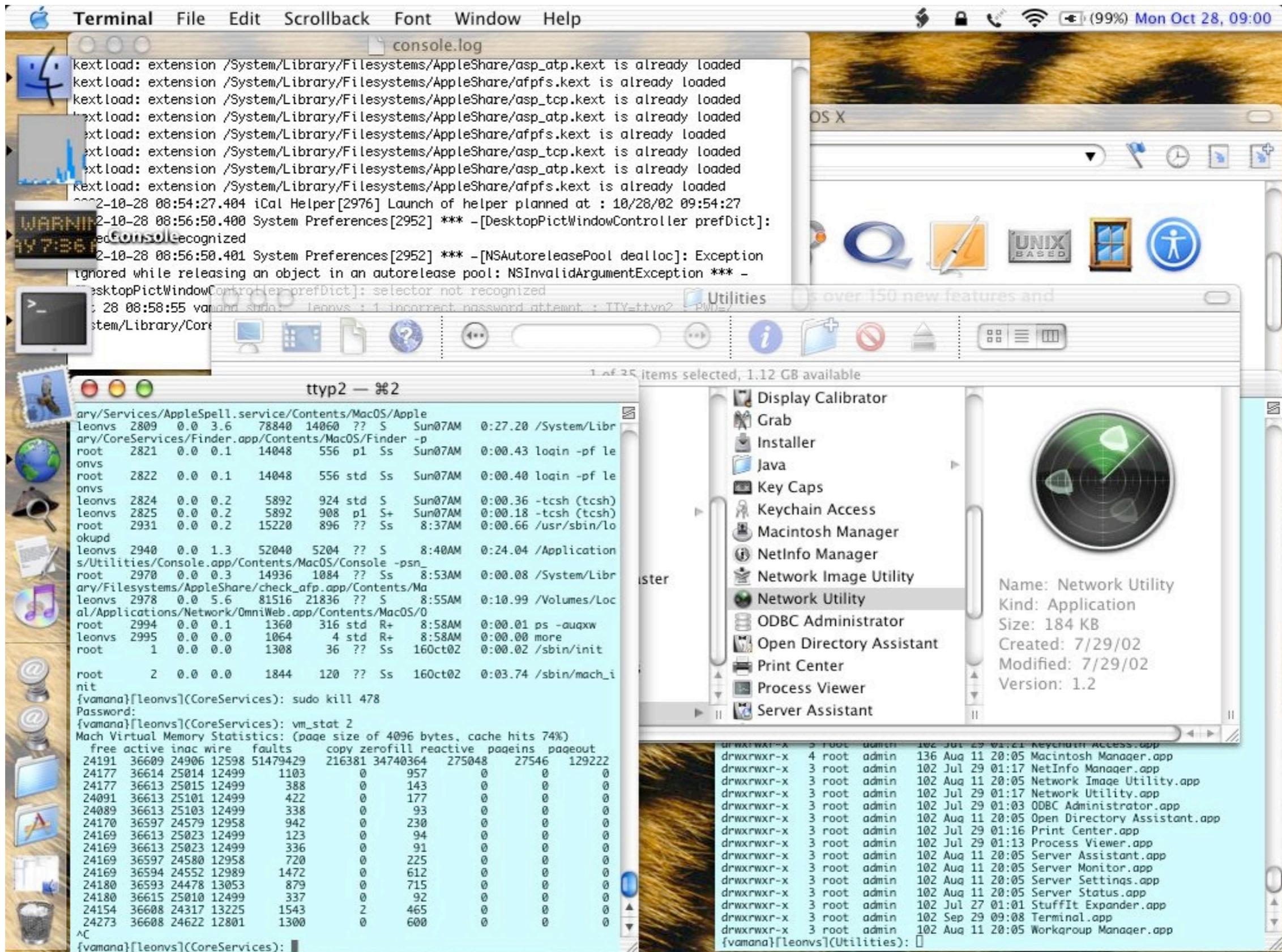
Classic



Darwin

The Structure of Mac OS X

- User Interfaces
 - Aqua
 - Only widely used UNIX with a native GUI not based on X11
 - Quartz 2D, the most prominent underlying rendering library, uses PDF as its native image format
 - Similar to Display PostScript under NeXTstep
 - X Window System (X11R6)
 - Implementations from Apple or third parties, based on XFree86
 - Included in Panther
 - Platinum (Classic environment)
 - BSD UNIX command line
 - Via Terminal application, SSH, single-user, `>console login`, Darwin



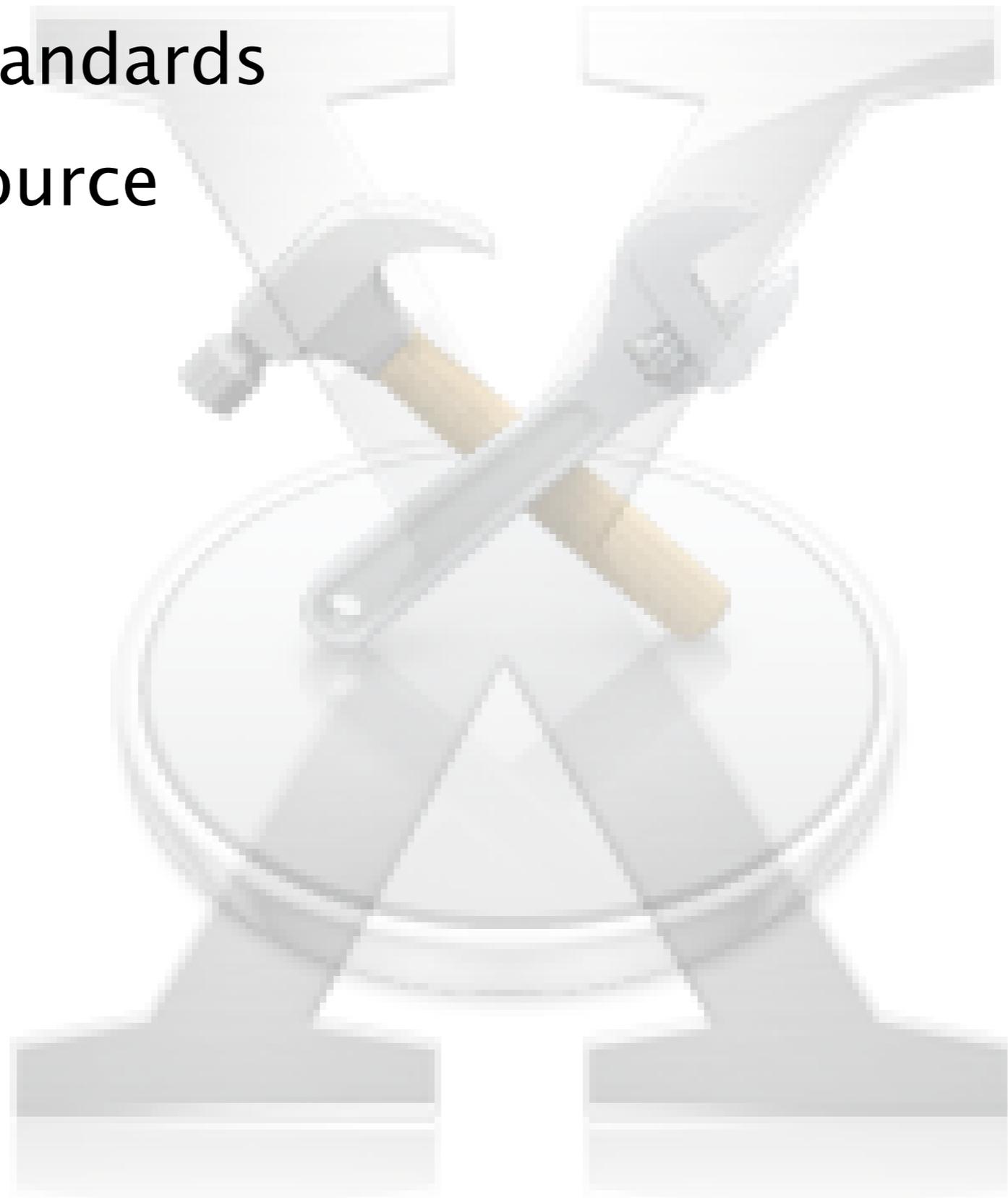
Aqua in Jaguar

- Programming Interfaces
 - Macintosh Toolbox
 - Mac OS 9 executables run under Classic
 - POSIX(ish), for UNIX programs
 - Java 2 Platform, Standard Edition
 - Carbon
 - Overhaul of Macintosh Toolbox to support advanced features
 - Cocoa
 - Evolution of OpenStep

- Kernel Environment
 - BSD UNIX
 - Multiuser security (users, groups, file permissions), process model (forks, threads), network access (sockets)
 - Filesystems: HFS/HFS+, UFS, FAT, ISO 9660, UDF, AFP, NFS, SMB, ...
 - Mach
 - Developed at CMU as experiment in microkernel design
 - Early versions integrated BSD, which NeXT used
 - Mac OS X kernel primarily derived from Mach 3.0 used in MkLinux, with NeXT enhancements
 - Still a monolithic kernel, for performance
 - Manages memory, processes, and hardware access

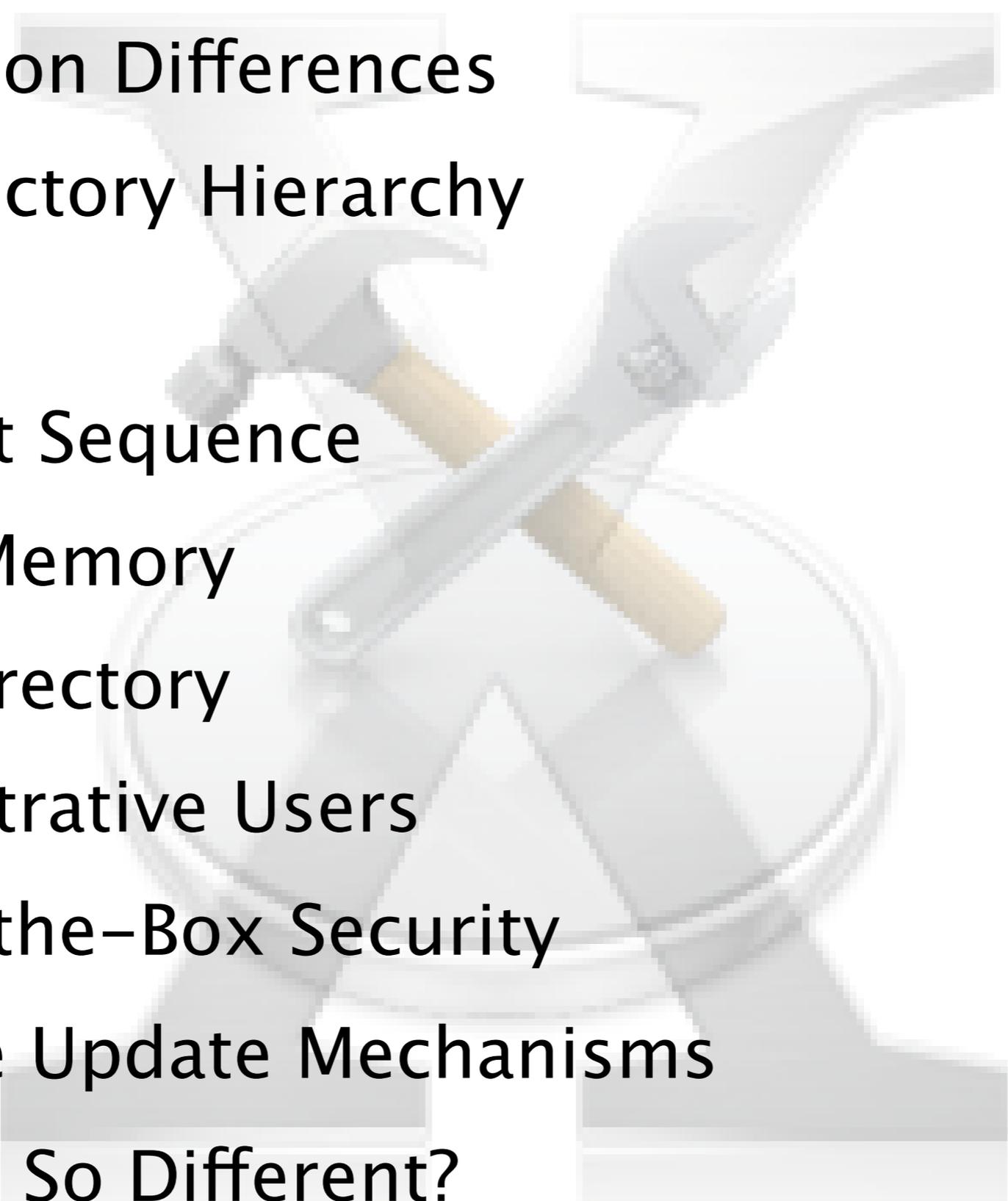
- Kernel Environment
 - Mach (cont'd.)
 - Theoretically capable of highly scalable multiprocessing, but Apple has so far released only dual-processor machines
 - Better kernel resource locking in Tiger for improved multiprocessing
 - I/O Kit
 - API for writing device drivers
 - Uses simplified variant of C++, drivers constructed in an OO-like hierarchy for ease of programming
 - Supports advanced power management capabilities, such as sleep, that aren't traditional UNIX strengths

- Open Standards
- Open Source



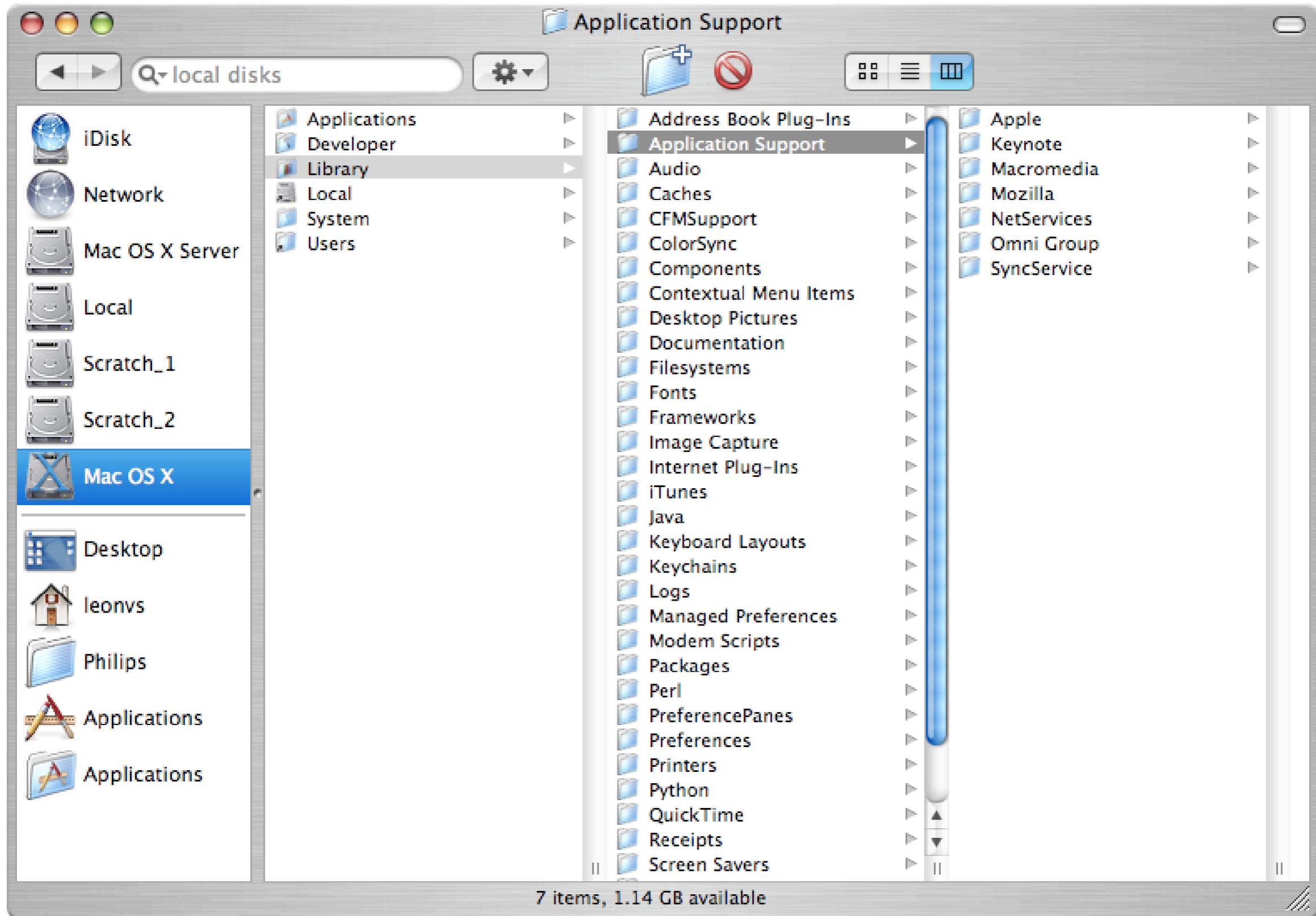
- Mac OS X is UNIX
 - On the whole, the similarities far outweigh the differences
- Open Standards
 - Protocols and formats: TCP/IP, LDAP, IPsec, Zeroconf, SMB, NFS, PDF, ...
 - Hardware: SDRAM, USB, ATA, PCI/AGP, FireWire, HyperTransport, Wi-Fi, Bluetooth, ...

- Much of OS X is based on open-source software
- Darwin, FreeBSD, NetBSD, OpenBSD, Mach
- Apache, CUPS, OpenLDAP, Postfix, Cyrus, OpenSSH, MySQL, Samba, BIND
- Rendezvous, KAME, OpenSSL, XFree86
- Perl, Python, Tcl, Ruby
- And much more

- 
- A Focus on Differences
 - The Directory Hierarchy
 - HFS+
 - The Boot Sequence
 - Virtual Memory
 - Open Directory
 - Administrative Users
 - Out-of-the-Box Security
 - Software Update Mechanisms
 - Why Is It So Different?

- While much of Mac OS X is familiar from other operating systems, there are many important differences that make it unlike any other UNIX system you've used
- Due to the approach of this presentation, and to human nature, we'll be focusing on these differences

- Parts of the OS X directory hierarchy look pretty familiar when viewed from the command line: `/bin`, `/sbin`, `/dev`, `/usr`, ...
 - `/etc`, `/var`, and `/tmp` are symlinks to subdirectories of `/private`
 - NeXTism related to NetBoot
 - `/Applications`, `/Library`, `/System`, `/Users`, `/Network`, `/Developer`
 - By default, non-root filesystems are mounted on subdirectories of `/Volumes` by `autodiskmount`
 - `fstab` configuration is possible, but unnecessary
- From the Finder (the graphical file manager), things look different
 - Some directories, called bundles, look like single files in the Finder
 - Applications, frameworks, plug-ins, mailboxes, ...
 - Note: "Directories" are referred to as "folders" in the GUI



The View from the Finder

- The default local filesystem format is HFS+
 - Developed from the original Mac Hierarchical File System (HFS)
 - The other choice is the familiar UNIX File System (UFS), based on the Berkeley FFS, but performance concerns and lack of support for multiple forks makes it less than ideal in most circumstances
 - Panther improved UFS performance
- Coming from a UNIX background, HFS+ exhibits behaviors that take some getting used to
- Multiple forks per file
 - Data and auxiliary resources can be stored in separate filesystem objects
 - Resource fork used for things like file-specific icons, application multimedia, whatever

- Multiple forks per file (cont'd.)
 - For the most part, the extra forks are invisible
 - Resource forks are visible with `ls -l filename/..namedfork/rsrc`
 - Some CLI utils in `/Developer/Tools` can deal with multiple forks
 - Forks create huge problems for non-HFS-aware software, including standard UNIX tools
 - `cp` and `mv` only move data forks and leave resource forks orphaned, backups don't get all necessary data, etc.
 - To be addressed (finally!) in Tiger
 - Resource forks are discouraged in OS X
 - Developers should use bundles instead
 - Multi-forked files on UFS are stored in AppleDouble format
 - Content of resource and attribute forks kept in `._filename`

- File attributes
 - HFS+ supports extensive file metadata
 - Typical UNIX metadata: owner, group, permissions, mod date, etc.
 - Files can exist without UNIX metadata (e.g., files created in Classic), in which case they show defaults based on the volume mount point
 - BSD flags: immutable, append-only, etc. (`man chflags`)
 - Macintosh file attributes: type, creation date, locked, invisible, etc.
 - Stored in attribute fork (or in `._filename` on UFS)
 - In `/Developer/Tools/`, `SetFile` lists available flags, `GetFileInfo filename` displays type, creator, and flags
 - Filename extensions encouraged over type/creator attributes in OS X, for cross-platform compatibility
 - No extended ACLs (until Tiger)

- Case-preserving, but case-insensitive
 - ReadMe is stored with mixed case retained for display, but it can also be accessed as `README`, `Readme`, or `readme`
 - ReadMe and `README` cannot exist in the same directory
 - Apple addresses this for Apache with `mod_hfs_apple`
 - Panther introduced fully case-sensitive HFS+ variant
 - Tip: `tcsh` command completion is still case-sensitive unless you set `complete = enhance` in `~/tcshrc`
- The path separator is a colon (:), not a slash (/)
 - Pathnames are converted on-the-fly by the kernel, so that colons look like slashes
 - Carbon apps convert slashes back to colons

- Application libraries access filesystem objects by numerical file IDs, not pathnames
 - File IDs are unique per disk volume
 - Lookups are faster than by pathname
 - Kind of like inode numbers; in fact, `ls -i` displays file IDs on HFS+
 - File IDs don't change when files are moved around on a disk volume
 - If you know a file's ID, and the the ID of the volume it's on, you can always access it as `/.vol/vol_ID/file_ID`
 - If you know the ID of the directory containing a file, you can access it as `/.vol/vol_ID/dir_ID/filename`

● Aliases

- An alias is a lightweight reference to a file or directory
 - Like a symbolic link, but uses both pathname (preferably) and file ID (as backup)
- An alias can continue to refer to a file even if it's moved (on the same volume) or renamed
- Both aliases and symlinks are useful in different circumstances
 - If the actual pathname is all-important, or you need to use it from the CLI, use a symlink
- Both aliases and symlinks are denoted by small arrows on file icons in the Finder
 - At CLI, an alias looks like a zero-length file, but with a resource fork
- No way to create symlinks from GUI, or aliases from CLI



- Hard links
 - On UFS, a hard link is simply another reference to a file's inode
 - With no inodes, HFS+ lacks support for hard links
 - OS X supports hard links for backwards compatibility, but they're implemented in the kernel as symbolic links, faked out to look and act like hard links
 - Slower than real hard links
- Number of links shown for a directory in `ls -l` output counts all items within the directory, including files
- HFS+ lacks support for sparse files; void extents are zero-filled
- HFS+ supports journaling, for faster recovery after crash
- See http://www.mit.edu/people/wsanchez/papers/USENIX_2000/ for more on filesystem design decisions in OS X

- General pattern is the same as most UNIX systems: run bootstrap code from persistent memory, use that to find a kernel and load it into main memory, load hardware drivers, mount filesystems, and progress through a series of initialization programs that start up the services required on a multiuser operation system
- BootROM
 - Located in firmware
 - POST
 - Hardware initialized using drivers in Open Firmware
 - Boot device selected based on NVRAM settings
 - Affected by System Preferences -> Startup Disk

- BootX
 - Located in `/System/Library/CoreServices/`
 - Kernel (`/mach_kernel`), drivers, and boot-time kernel extensions loaded into memory
- Kernel initialization
 - Data structures initialized
 - I/O Kit initialized, drivers linked into kernel
 - Root filesystem mounted
 - Mach bootstrap port server (`mach_init`) started

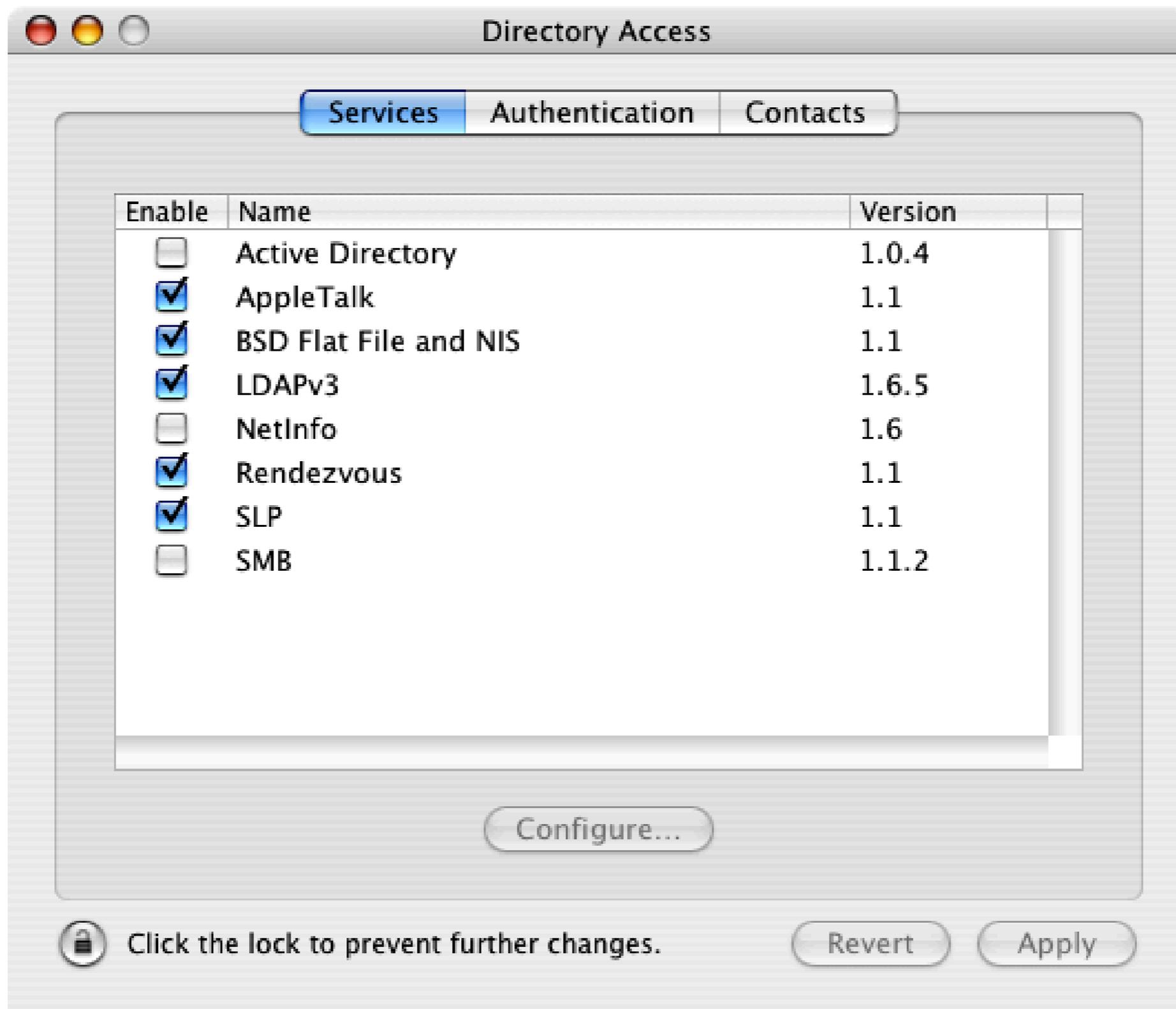
- System initialization
 - `mach_init` starts BSD `init` (PID 1), takes on PID 2
 - `/etc/rc.boot` brings system to single-user
 - `/etc/rc` brings system to multi-user
 - Starts `kextd` to handle kernel extension requests and unload unnecessary drivers
 - Starts virtual memory pager (`dynamic_pager`)
 - Runs `register_mach_bootstrap_servers` to process Mach bootstrap daemons
 - Runs `SystemStarter` to process startup items

- Mach bootstrap daemons
 - Introduced in Panther as a replacement for certain startup items
 - Mach ports for specified daemons are registered with the bootstrap task of `mach_init`
 - A Mach **task** is analogous to a process that runs within the kernel; a **port** is used to send messages to a task
 - When another task asks the bootstrap task for access to one of the ports, `mach_init` starts the associated daemon
 - Daemons can be run only when needed, if another process needs to communicate with it, thus conserving system resources
 - Configured by XML property lists in `/etc/mach_init.d/`

- Startup items
 - Contained in `StartupItems/` in `/System/Library/` and `/Library/`
 - Each item is a directory, containing:
 - Executable named the same as the directory, run with `start` argument
 - `StartupParameters.plist`
 - Startup items can execute in parallel, and the order is not deterministic
 - Startup items aren't executed on shutdown, which can cause problems for some things that require handholding, like databases

- Mach features an efficient virtual memory implementation
- Backing store is file-based
 - It doesn't use a specially formatted disk partition (e.g., Solaris)
 - Definitive performance comparisons haven't been made, but it's sufficiently fast to not be a problem
 - Of course, you're much better off with enough RAM to avoid paging in the first place
 - It's possible to partition the VM files to faster, dedicated storage
 - Allocated as individual files in `/var/vm/`, acc. to the parameters of the `dynamic_pager` command in `/etc/rc`
 - VM disk usage grows and shrinks dynamically
- Use `vm_stat` (note the underscore) to keep an eye on memory usage

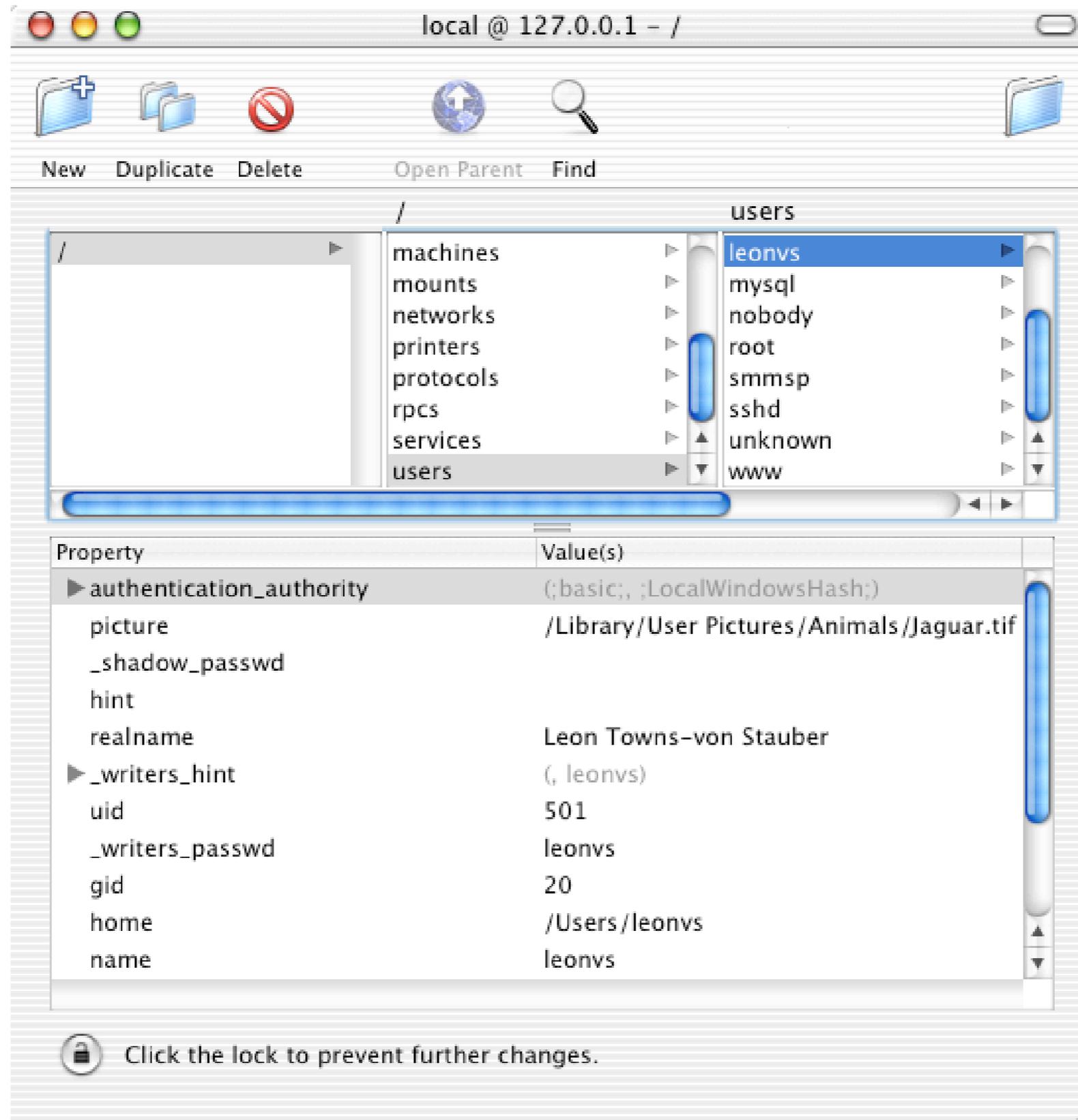
- The subject of directory services is **very** broad, and Mac OS X is probably the most flexible client and provider of directory services there is, so I won't do much more than skim it
 - OS X has a deep history with directory services, owing to its NeXT lineage
 - While there are other topics that might fall under the category of directory services, I'll restrict myself to talking about Open Directory, which is itself a large topic
- What is a directory service?
 - Loosely, it's a network service providing configuration data to clients
 - Information on users, groups, hosts, printers, etc.
 - Optimized for lots of quick lookups, infrequent changes
 - Examples: LDAP, YP (NIS), Active Directory, DNS, WINS, SLP



Directory Access

- In OS X, lookups for many kinds of configuration data are made through the Directory Services API
 - For legacy UNIX programs, the `getxbyY` system calls are rewritten to proxy lookups through `lookupd`, a daemon that makes use of DS
 - The data sources consulted by DS are configured in the Directory Access application
- OS X includes its own directory service, named Open Directory, based on OpenLDAP
- Legacy backend database format is that used by NetInfo
 - Still used for host-local information
 - Now deprecated, replaced by standard Berkeley DBs used by OpenLDAP

- Within a NetInfo-formatted database, information is organized in a directory hierarchy, analogous to a filesystem directory hierarchy
 - Root is `/`, subdirectories include `/machines`, `/users/leonvs`, etc.
 - Nodes in the hierarchy have properties, each one key to a set of values
 - Properties include `name`, `uid`, `ip_address`, `passwd`, etc.
- The Big Surprise
 - Traditional UNIX flat files in `/etc` (`passwd`, `group`, etc.) aren't used by default (except in single-user mode)
 - OD is the primary source of configuration data for most OS X machines
- Tools
 - GUI: Directory Access, NetInfo Manager, System Preferences, more in Mac OS X Server
 - CLI: `dscl`, `nicl`, `nidump`, `niload`, `nireport`, `nifind`, `nigrep`



NetInfo Manager

- By default, `root` logins are disabled on OS X (by an invalid password)
 - To enable, use NetInfo Manager->Security->Enable Root User, `dsenableroot`, or simply `sudo passwd root`
 - On Mac OS X Server, `root` password same as initial admin user
 - Consider changing one or the other, so they're not the same
- Administrative work is designed to be accomplished by members of the `admin` group, who possess special privileges
 - Some privileges are configurable, and may be removed or reassigned, while others are hard-coded to the `admin` group
 - NB: The user account created during installation is in the `admin` group

- File permissions
 - Directories and files in `/Applications/`, `/Library/`, and `/Developer/` are owned and writable by group `admin`, permitting software installation
- `sudo`
 - Admin users have superuser access to CLI commands, configured in `/etc/sudoers`
- `su`
 - Can only `su` to `root` if in group `admin` or `wheel`
 - Configurable in `/etc/pam.d/su`

- Authorization Services
 - Part of the Security framework
 - Gives admin users superuser privileges for certain GUI activities: running software installers, configuring directory access, changing certain things in System Preferences, etc.
 - Configured in `/etc/authorization`
- Open Directory
 - Admin users have full write access to OD domain contents, via either NetInfo, or LDAP when NI authorization is enabled
 - Hard-coded

- Apple Filing Protocol (AFP) server
 - Administrators can connect as any user, authenticating with their own password, and they gain special access privileges
 - Hard-coded to `admin` group, but can be configured with properties in Open Directory, under `/config/AppleFileServer` in `local` domain
 - `attempt_admin_auth`: Determines whether authentication is attempted against administrator passwords
 - `special_admin_privs`: Grants admins read access to all folders
 - `permissions_model`: Gives admins the ability to change ownership of all files if set to `unix_with_classic_admin_permissions`
 - `admin_gets_sp` (Boolean, default 0): Lets admins mount share points instead of volumes

- Samba
 - Not done by default, but you could give admins superuser privs on SMB shares with a `username map` (in `/etc/smb.conf`) of `root = +admin`
- CUPS (Common UNIX Printing System)
 - Specifying `AuthClass System` in `/etc/cups/cupsd.conf` requires `auth` as member of group `admin`
 - Configurable with `SystemGroup` directive
 - Not enabled by default
- QuickTime Streaming Server
 - Members of group `admin` can make configuration changes through QTSS web interface, but this group is completely defined within `/Library/QuickTimeStreaming/Config/qtgroups`, and is unrelated to the system group `admin`

- Apple's customers were used to a secure out-of-the-box experience
 - Legacy Mac OS offered very few services, and was rarely attacked
- As a result, Mac OS X has the most secure default configuration of any major UNIX platform
 - Finally, a UNIX vendor that takes security seriously!
 - Comparable to OpenBSD, or possibly Solaris 10
- As mentioned before, `root` logins (even locally) are disabled
 - All superuser access requires authentication as a non-`root` user first
- No network-accessible services enabled (almost)
 - Exceptions: NTP, Rendezvous multicast DNS
- Neither of these conditions (disabled `root` logins, almost no accessible network services) is true for Mac OS X Server

- A variety of automated software update mechanisms available
- Software Update
 - Built-in
 - Downloads and installs packages to update OS and other Apple s/w
 - Do it from command line with `softwareupdate`
- DarwinPorts (<http://www.opendarwin.org/projects/darwinports/>)
 - Similar to FreeBSD ports, being worked on by Apple
 - Reportedly built-in to Tiger

- Methods ported from other UNIXen
 - Fink (<http://fink.sourceforge.net/>)
 - Port of Debian `apt-get` system
 - GNU-Darwin (<http://www.gnu-darwin.org/>)
 - Port of FreeBSD ports system
 - NetBSD Packages (<http://www.netbsd.org/Documentation/software/packages.html>)
 - RPM (<http://www.rpm.org/platforms/osx/>)

- Some important differences: Quartz vs. X11, HFS+ vs. UFS, Objective-C vs. C++, NetInfo vs. LDAP, AFP vs. NFS, file-based VM, etc.
- Many design decisions were made in the middle to late 1980s, during the development of NeXTstep
 - Many of today's ubiquitous technologies (X11, C++, YP/NIS, LDAP) were not yet well-established
 - NeXT was among the first to implement a UNIX GUI, a standard OO dev environment, directory services, etc., and happened to choose differently than the rest of the industry later did
- Some changes were made to support Apple's existing user base
 - HFS+, AFP, secure default config

- But why does Apple stick with technologies that require special training?
- Because some are just better than the alternatives
 - Objective-C is a cleaner, more flexible language than C++
 - HFS+ is arguably more capable than UFS under certain circumstances
 - Most other UNIX platforms also intend to replace UFS, or have already done so
 - UFS2 (FreeBSD), XFS (IRIX), ZFS (Solaris), etc.
 - Quartz performs well and is self-consistent
- Because Apple controls these technologies, and can drive their improvement
- They are willing to accomodate alternatives (UFS, NFS, X11, C++) or even replace proprietary technologies (NetInfo -> LDAP)

- As a Client
- As a Server



- UNIX
- Native productivity apps
- Fit-and-finish of the graphical experience
- Nice hardware: well-designed, at reasonable prices, with stable device drivers
- If you, or your users, are comfortable running Intel-compatible PCs with *BSD or Linux running X11, Open Office, The GIMP, Evolution, etc., then Mac OS X may not be for you

- UNIX
- Mac OS X Server is a capable, affordable product with lots of nice features to ease administration
- Makes a good cross-platform directory, authentication, file, and print server
- Suitable for light to moderate Web, mail, and other Internet services
- The Xserve is well-designed, has good performance, and is reasonably priced

- Web Sites
- Mailing Lists
- Books



- **Apple's Mac OS X Site**

- <http://www.apple.com/macosx/>

- **Mac OS X Hints**

- <http://www.macosxhints.com/>

- **O'Reilly Mac DevCenter**

- <http://www.macdevcenter.com/>

- **The Challenges of Integrating the Unix and Mac OS Environments**

- http://www.mit.edu/people/wsanchez/papers/USENIX_2000/

- **NeXT: When cool wasn't enough**

- <http://www.vnunet.com/features/1112630/>

macos-x-server (Apple)

<http://www.lists.apple.com/mailman/listinfo/macos-x-server/>

macosx-admin (Omni Group)

<http://www.omnigroup.com/developer/maillinglists/macosx-admin/>

macosx-talk (Omni Group)

<http://www.omnigroup.com/developer/maillinglists/macosx-talk/>

security-announce (Apple)

<http://www.lists.apple.com/mailman/listinfo/security-announce/>

- Mac OS X Panther for Unix Geeks
 - Brian Jepson, Ernest E. Rothman
- Mac OS X Panther in a Nutshell
 - Jason McIntosh, Chuck Toporek, Chris Stone
- Running Mac OS X Panther
 - James Duncan Davidson

- Slides for this and other presentations available at <http://www.occam.com/osx/>
- Q & A