

The journal of Apple technology.

Volume Number: 15 (1999)

Issue Number: 4

Column Tag: Viewpoint

Apr 99 Viewpoint

by Rich Morin

Starting this month, we're inviting the contributing editors of MacTech to introduce themselves and their areas of interest to our readership, to share their Viewpoints with us. This month we've asked Rich Morin to talk to about the developing Macintosh Open Source movement and Mac OS X. Feel free to write to Rich, with article ideas or opinions, at rdm@cfcl.com.

Nick DeMello, Editor

I was pleased, if a bit bemused, when Nick DeMello invited me to be a Contributing Editor for MacTech. I'm not a Mac OS developer; the only Mac programming I do is in MacPerl. I edited Apple's MkLinux Reference Release, but that's not even Mac OS!

After a bit of reflection, or possibly rationalization, I decided that I do have something to contribute. I have a rough understanding of the Mach microkernel's design, am quite comfortable with Unix, and am very familiar with Open Source philosophy and practices.

By introducing MacTech readers to some of these notions, I may be able to help Mac OS X (and thereby Apple) to succeed. As a Mac devotee (I'm typing this on my G3), I like this idea quite a bit. So, let's begin: What is Open Source and why should Mac developers care about it?

To qualify as Open Source, software must allow inspection, modification, and even redistribution of its source code by any and all parties. (See <http://www.opensource.com> for the Open Source Definition.) This is a rather heady concept for some, and rather appalling to others, but it is really nothing new.

Open Source is very popular in the Unix (Linux, etc.) community, for several related reasons. Unlike Mac OS users, most Unix users program. They may not be professionals, but they get things done. They start by entering sequences of commands in interactive sessions. Later, they may collect commands into "shell scripts", adapt them into Perl scripts, etc.

Unlike Mac OS, which provides few programming tools, most Unix distributions offer a C compiler and interpreters for several other languages. So, the tools are there and the users are comfortable in (at least some of) their use. They are also part of a culture with a long history of source-code exchange.

Unix users have always exchanged source code. Unix machines may have subtly differing APIs or even totally different architectures. Binary distribution is too inflexible; only source code can be exchanged with any reliability. Also, because AT&T distributed Unix entirely without support, source code was the only way to get (and keep) it working.

Although most commercial vendors distribute their packages in binary form, Unix "freeware" is nearly always distributed as source code. And, because many of the recipients program, a lot of volunteer development takes

place. Users add features, fix bugs, and combine code in ways never imagined by the original authors. The result is an incredible fount of creativity, coupled with very rapid convergence towards bug-free implementations.

This is all very well, you may say, but what has it got to do with the Apple community? We don't have user-programmers or even tools for them to use. Our freeware and shareware is distributed in binary form and few users desire anything different.

True enough, but some changes are on the way. I run Perl and Tcl/Tk on my Mac; I could be running Python or any of several other languages. So, the tools are starting to appear. Also, Mac OS X is certain to have a full set of Unixish commands (by Apple or third parties).

More to the point, Mac OS X itself depends heavily on Open Source software. The Mach 3 microkernel forms the basis of the system, supplemented by the 4.4BSD-Lite distribution. Open Source add-ins, from Apache and Bind to Perl and Sendmail, fill out the system.

If Apple had to pay license fees on these components, could they sell the resulting system at mass-market prices? I think not. Will Apple rewrite everything, through a surfeit of NIH? This seems unlikely, though some things will certainly be revised for assorted reasons.

Finally, whom does Apple expect to buy Mac OS X? Developers will buy copies, of course, but they aren't a big enough audience to support a new operating system. Advanced Mac shops (e.g., high-end graphics firms) will also jump in, for advanced features and increased stability, but they are also a relatively small audience.

I therefore predict that Apple will cater to, and actively woo academic, research, and otherwise technical users; in short, the bastions of the Unix community. Apple may be quiet about it, but mark my words; it will be a large part of their planning.

And, when these folks start playing with Mac OS X, will they leave all of their favorite tools behind? Not very likely; in fact, I wouldn't be surprised to see Apple encouraging a few Open Source developers to do Mac OS X ports of important Open Source packages!

Let's be clear: Mac OS X is not Unix. It will have a different file system, new scheduling and I/O architectures, and more. Most commercial applications will be ported to Carbon or written for the Yellow Box; neither of these is much like conventional Unix.

Nonetheless, I predict that the next few years will see a merging of concepts between the traditional Mac and Unix communities. Mac users will start seeing more Open Source software, even if (as seems likely) most will never look at the source code.

If this is true, it would behoove Mac developers to learn a little bit about Open Source, if not Unix, per se. The best places to begin such an exploration are <http://www.opensource.org>, <http://www.freebsd.org>, and <http://www.gnu.org>; tell them Rich sent you...