A Conversation with Chris DiBona

An open source advocate REFLECTS

ON THE EVOLUTION
OF THE MOVEMENT.

Chris DiBona has been out front and outspoken about the open source movement. He was hooked from the moment he installed Linux on an old PC when he was a teenager.

As DiBona's career as a developer progressed, with stints in information management at the U.S. State Department and in computer security with Tandem Computers, so did Linux. DiBona eventually joined VA Linux as a marketing executive. He became a spokesman for open source and wrote a book about it: *Open Sources: Voices from the Open Source Revolution*, (O'Reilly and Associates, 1999). He was also instrumental in creating the Open Source Development Network (OSDN).

DiBona recently left VA Linux to co-found a gaming development company, Damage Studios, which is using open source in the development of a next-generation massively multiplayer game called *Reconstruction*. He remains a strong advocate of the open source movement and its future.

Quizzing DiBona on his continuing involvement with open source is a man who knows his Unix and his open source software, Eric Allman, chief technology officer and founder of Sendmail Inc. Allman was an early contributor to the Unix effort at the University of California at Berkeley, where he received a master's degree in computer science in 1980. While there, he wrote *syslog*, *tset*, *trek*, and a little app you might know called *Sendmail*.

Allman designed database user and application interfaces at Britton Lee (later Sharebase) and contributed to a neural network-based speech recognition project at the International Computer Science Institute. He was also CTO at Sift Inc. He co-authored "C Advisor" for *Unix Review* magazine for several years and is a former member of the board of directors of the Usenix Association.

ERIC ALLMAN You're one of those names associated with open source for so long that I've always "just known" you without knowing exactly why. For example, I know you've been involved with the Open Source Development Network (OSDN), but I'm not sure whether you founded it or came in later. Similarly, you were an executive at VA Linux, but I'm not sure exactly what your role was.

Can you explain a bit more about your history with computing and open source?

CHRIS DIBONA I've been using computers all my

life; I wrote my first program when I was 12 years old on an Atari 800 with a basic cartridge. From there I taught myself Pascal and at 15 wrote my first program for a kiosk system in Salt Lake City that printed coupons. The project paid \$300, which was quite a bit for a 15-year-old who considered bus fare a major expense.

While I continued to program, my next jobs were working as a store clerk at a video store in Virginia and as a busboy. When I was 17, I took a job at a WaldenSoftware retail store, which was a version of the bookstore

PHOTOGRAPHY BY TOM UPTON

that stocked a very respectable collection of programming tomes along with your regular assortment of software. It was a great place to work because the bookstore had a two-week checkout policy on the books, so I could take home Tannenbaum one week and Aho the next. I still have a number of books that I purchased there with my employee discount.

At this point it is appropriate for me to point out that I was not a very good student (something I regret), and after graduating from high school I opted to continue my education at a community college until I was ready for real college. I also started working for Circuit City. While checking in and out broken consumer electronics and cleaning the heads on VCRs, I was earning my required

humanities credits and testing out of the computer science courses at the community college.

It became policy that incoming computer problems came to me first. I became an unofficial help line whenever customers needed to fix or change some setting on their computers. One of these customers worked at the U.S. State Department and offered me a job there evaluating technology in the administrative/information management department. The security clearance took almost one full year to complete.

During this year, I went back to the software store, as it was a decent job (with good books), and transferred into George Mason University's computer science program.

College wasn't the first place I encountered Unix.

When I was 18 or so I traded an older copy of Microsoft's C environment for an AT&T 3b1 Unix machine. For those of you unfamiliar with this machine, it was a 68010-based Unix machine with a 20-megabyte disk and 512 kilobytes of RAM. It was quite the ugly duckling. It shipped with a K&R (or close)-compliant C compiler, which was laid to rest by the GNU C compiler as soon as I got it onto the drive.

I decided to try installing Linux onto the 386/25 that I had under my desk and use that as a tool to transfer data to the "real" Unix machine. After discovering Linux in this manner, I proceeded to only turn on the AT&T machine to wonder at its cruftiness.

I really started using Linux in earnest when I was taking third- and fourth-year classes at George Mason—in part because the line waiting for the Sun lab was quite long and I was impatient. I had acquired a laptop, so I could do my homework anywhere. I was also really enjoying programming under Linux.

After taking the State Department job, I started writing prototype (read: deployed) applications for the department and reviewing proposals from consultants who wanted to sell products to the government. I got some nifty cryptography training and was exposed to some very cool technologies along the way.

I had been working at the State Department for a little over a year when the guy in charge of information management left to take a job as CIO for a law firm. He offered me a job running the e-mail for the firm. This wasn't a bad job, but when my brother-in-law at Tandem in Cupertino noticed an opening on its ATM group, I interviewed and moved out to California. I had one class to go for my bachelor's degree: artificial intelligence, a three-hour course that I have yet to complete. I'm not proud of this, and I'll get to it when I have the time to do it, but the opportunity to move West was compelling for me.

Also, have you ever had to run and support Microsoft Mail for the Macintosh for 300 type-A litigation lawyers? You would move, too.

EA What version of Linux were you using at that time?

CD I picked up a copy of Infomagic's Linux developer resource CD, which contained the floppy images for Slackware Linux.

EA Did it "just work," or was it a frustrating experience?

CD Well, it was a little frustrating. XFree was very difficult to get up and running, and I could have screamed at the

number of floppies that turned out to be bad. The process went something like this: Start install, get 10 disks into it, 11th disk bad, reboot and re-create 11th disk, restart install, get 14 disks into it, 15th disk bad, etc., etc. But I got through it.

EA What was the support experience like? Who did you ask? What was the community like? Did you get the answers you needed?

CD A couple of guys at school were using it, but I mostly just plowed through the problems with the help of the manual pages, docs, and Usenet newsgroups. I've found that, if you just assume you aren't the first person to have a problem, you can find that first person online and see what he or she had to do. I hope that people know just how lucky they are to have Google.

As I progressed in my career, Linux was maturing. I took part in the local Silicon Valley Linux users group, which I ended up running. During my tenure there, I met Larry Augustin of VA Research (which later became VA Linux and still later became VA Software), and he expressed the need for some help. The idea of a Linux job sounded pretty cool, so I asked him about it. He answered that he needed help with marketing and events. In fact, my first assignment was to plan our Comdex booth and arrange all the Comdex-related travel and events. Comdex was only two weeks away when I started.

EA So becoming an exec was as easy as that? I'm one of those poor idiots who had to work his way up through the ranks. Is there any reason it came to you so easily? Was this a product of the boom times or something else?

CD Well, I think I was indeed a direct product of the dot-com boom. It wasn't as if I were running the place, and developer relations was a pretty good fit for me, so I didn't feel unqualified. I never really tried for VP or a similar position because I knew I would be unfit for the job. I realize that being unqualified never stops a lot of people, and a part of me knows that I could have done the job as well as the people who came into the business later. But who knows, I might have done poorly. What is important to me is that I know I can do the job I am doing now.

EA You're my dream technology marketing dude—someone firmly rooted in technology who didn't just read about it in a textbook written by someone who wouldn't know a chip if it bit him. Are you accepted into the more

mainstream marketing ranks? Do you want to be?

CD I think that I am, but for me that means understanding how marketing works, something I've spent a lot of time learning about. I actually am not in this business to win a Clio [advertising award], but to sell a product. Having other marketing people accept me isn't high on my list.

The thing about technology marketing is that so many people really don't know enough about what they are selling, so they don't understand who wants to buy it. When you couple that with the history of the business over-promising (read: lying about) the things it is supposed to market; and when you couple this with the kindergartners-playing-telephone nature of the Internet, you can have some very real expectation management issues.

The part I like best about marketing is probably the creativity of it. I know that sounds weird, but getting just the right phrase, image, and symbolism—even in the service of selling a product—is as satisfying to me as a really well-written bit of code.

EA What about the time you spent with VA Linux?

CD Over the next four years, I wrote *Open Sources: Voices*

from the Open Source Revolution and any number of articles, spoke about open source everywhere, ran the VA friends and family program, which included some 1,200 free software developers, and finally I joined the staff of Slashdot.org as an editor. During this time I also worked a lot with the Electronic Frontier Foundation (EFF) testifying in the New York DVD case. I left VA as a full-time employee in October of last year to co-found Damage Studios, developing the game *Rekonstruction*, with some other VA alumni.

EA Can you give us more detail on the New York DVD case? What was your testimony?

CD The New York case was about whether or not a magazine (in this case, *2600*) should be allowed to link to a Web page that has the DeCSS code. This code unlocks the content on a DVD so that you can watch it under Linux or another free operating system that doesn't have the financial wherewithal, or desire, to purchase a license from the DVD people.

The lawyers for the EFF, who were representing *2600*, wanted me to testify that the ability to reverse-engineer code was vital to the health of not only the free software

industry but also computer science in general. The judge shut us down and we were unable to make this point. I have a pretty low opinion of Judge Kaplan of the New York circuit court and his ability to, well, judge as a result.

EA What does an editor at Slashdot.org do? I had the impression that it was little more than a free-for-all. As a side note, when Kirk McKusick and I were Slashdotted for an interview we did for gayteenresources.org, we got widely trashed. A lot of folks said things like “why do fags always need to push themselves on us?” I was fairly disgusted by the homophobia and haven’t read Slashdot since. Is it more than a flame-fest?

CD I went back to look up that story and I have to say that I’m more than a little ashamed of the behavior of the people who made those kinds of comments. The reality of Slashdot is that it reflects humanity at its best and worst. That said, I really think that surfing Slashdot at anything under a moderation level of two or three is only for the masochistic.

Slashdot is a funny place. Of those who read it, only a quarter even visit the comments, and of those, very few post. Slashdot gets something around 2.5 million page

views a day, distributed among 250,000 individual ips (or near that, it’s been a while), so most people come to check the front page and that’s about it.

An editor’s job at Slashdot is both very easy and kind of painful. The core job is pretty simple. You monitor a queue of incoming submissions, select one you think is “slashdotworthy,” massage it a little to make it fit Slashdot style (remove paragraphs, nip and tuck, etc.), and schedule the story to appear on the site. Slashdot gets about 500 to 1,000 submissions a day, of which only 16 or so make it to the front page. A somewhat larger number of submissions make it into the subsections, but the math of it means that most of what a Slashdot editor does is delete incoming stories.

The other part of what a Slashdot editor does is to serve as moderator. Moderating on Slashdot is not vital, but it can make a big difference in the first 10 or 15 minutes of a story’s life on the site. What were mostly kept moderated as off topic were the “First posts” and other nastier things that people seem to love posting on the site. Some people, as you’ve noticed, see anonymity as an excuse to behave like miscreants. Anonymity is very important, though, so we did what we could to make it work out.

My absolute favorite part of the job at Slashdot had to be running the polls. It was great fun.

EA You've mentioned that your new venture, Damage Studios, is using open source. What does Damage Studios do, and how does open source fit in?

CD Damage is a game development studio located in San Francisco. We're working on a next-generation massively multiplayer online game called Rekonstruction, which is itself not an open source project. We're using open source in a number of ways to facilitate development and deployment of Rekonstruction (the curious can keep on top of what we're doing at <http://www.damagestudios.com/form.php>).

Rekonstruction is set 400 years in the future on Earth. We are using satellite data to model the planet as realistically as we feel is appropriate to the story of the world, which had suffered a dinosaur-killer-like asteroid in the late 23rd century.

Unlike many massively multiplayer online games, Rekonstruction is designed so that the players coexist on the same world instead of on many duplicate copies of the same world. This is a harder computation (and creative) problem, but one we think is one of the key features that will make Rekonstruction enjoyable and worth playing.

Our core operating system platform for our back-end servers is Linux (standardized on Red Hat). We use C++ for our core, backend game server software. We use Python for creature scripting, non-player character scripting (think automated humans, such as vendors and others), and other scripting. We use open source as a tool and consider the monthly service and creativity as the product. To some degree the software is just the mechanism for delivering it.

For interacting with our customers, we use Apache, Sendmail, and the Mailman mailing list software. We maintain a few servers internally running Samba for file and print.

EA Why did you choose Red Hat? Any particular version?

CD We are using Red Hat Linux 9 right now and we have a number of Red Hat network enterprise subscriptions. We have found that it's a very stable platform to develop and eventually deploy on. We run a fairly conservative shop, and upgrades are quite suspect, as we can't have the attitude that even a minute of downtime is acceptable to us—or our users.

EA You've certainly had plenty of experience with open source. Where do you think it really shines?

CD Although open source is clearly the superior choice for a server platform, we cannot stress enough the value the open source implementations of C++, Python, and PHP offer us. A lot of people don't recognize the role that the compiler and interpreter play in the day-to-day operations of a business like ours. Can you imagine not having Perl available to you? We also make heavy use of the PostgreSQL open source database and we're examining its suitability vs. MySQL. Open source for us represents more than just a dramatic cost savings. The configurability and control is very appealing to us.

EA Why is open source so clearly the superior choice for a server platform? (That's clearly not true for our friends at Microsoft.)

CD Damage is a shop that has its share of Windows desktops. Since Windows is the primary client platform, we do a lot of development on and for the Windows XP client OS. Linux via Samba is very happy to integrate with the user desktops that we have, and I feel it is easier to maintain than XP for file serving.

I don't want to sound too repetitive, but as a marketer I am very happy using Mailman and Apache to reach my target market.

I actually don't have anything against Microsoft's Internet Information Services (versus Apache) or Exchange; I just can't justify them. They cost money and are less configurable and reliable, so they don't work for us. We did a trial of Exchange, but it didn't fit our needs.

EA Are they better than the alternatives, or just "good enough"?

CD I see them as better, but the fact is, we don't need any of the features of Exchange, so that might be a false judgment on our part.

EA Conversely, are there any areas where you think open source just doesn't stand up to commercial development?

CD A lot of people like to come down on the Linux desktop, but the metaphor itself, and the Office suite question, isn't really that important to us. I personally don't see it as a problem in general. As a company that puts a high premium on graphic design, modeling, and texture creation, however, we've found that Linux's lack of real

color management makes it unusable for our purposes for those particular employees.

Other areas we've found wanting is printing and to some degree development. The Microsoft Visual Studio (VS) integrated development environment (IDE) isn't perfect, but the way it helps a programmer debug large code bases (third-party code use is common in the game business) can be extremely useful. This might be showing my ignorance of some decent core open source tools, but Visual Studio is very good at this. That said, its help system is probably the most frustrating tool I've ever used. They should just admit defeat and bring up a Google search when you slap the F1 key.

EA What makes Visual Studio so good?

CD VS is one of those tools (like emacs) that, once you know it, you can do a lot with it very quickly. I also think that the way it integrates debugging into the IDE makes it possible to program Windows apps very quickly. I'm not a big fan of Microsoft Foundation Class (MFC), but the wizards are pretty powerful for throwing together code for Windows very quickly.

The auto help is the only cool part of the help system. You can rig it so that as you are typing it brings up function definitions, which can be absurdly helpful when programming a user interface with Windows.

EA Any take on Mac OS X? The fusion seems at least close to right (with work to be done, of course).

CD I think that Mac OS X is a perfect example of how you can use open source software as a foundation building block within a company to create a distinctive, useful, and very cool software product. If I may oversimplify, Apple took what it was really good at—hardware and user interface design—merged it with the operating system expertise that BSD represents, and produced a terrific upgrade path from its moribund OS 9 code base.

EA We've seen open source—in the form of Linux,

Apache, Samba, Sendmail, MySQL, and so forth—make substantial inroads into the server market. The attempts to move into end-user clients haven't been quite as successful. Several desktops seem quite adequate (for example, KDE and Gnome), but other tools (notably, office tools) seem to remain a bit difficult to use for non-techie types. What do you see as the future of open source on the desktop?

CD For my personal use, the desktop is in a really good place. A lot of people come down on Red Hat's Bluecurve interface, but I find it to be quite attractive and usable. Bluecurve is the unified user interface look and feel that Red Hat introduced in its 8.0 version. A number of vo-

cal people in the free software community derided it as candy coating, but it makes for a very friendly face for those new to Linux.

Regarding office tools, I have found OpenOffice to be a useful tool. I really think that unless a person is writing a book or other specialized reports, that Microsoft's Office suite is overkill. OpenOffice could be both easier to use and faster, for

sure, but if it is speed you need, you'll use vi or emacs. I've also been very happy with OpenOffice's increased ability to load Word and other Office files. [For a look at interoperability of Linux and Office products, see Varian and Varian's "The Microsoft Office-Linux Interoperability Experiment" on page 40 of this issue.]

We're lucky at Damage as all of us come from a free software background of one sort or another. We get our share of Word and Excel files, and we have a few copies of Office lying around in the event that OpenOffice doesn't read them, but ours isn't really a spreadsheet and Word kind of business. We have a policy in place to not accept Word files from people outside the company anyway, in the event it might get run on a Windows machines and trash that machine. Who has the time for that, after all?

Also, not to make too fine a point of it, but Office is expensive. I can buy any number of things for the cost of all those licenses.

The future of the Linux desktop is likely to, and must, feature tighter inter-application integration. All of the

programs still feel disconnected from each other. I would also like to see the GIMP (GNU Image Manipulation Program) more focused on graphic design and color control. I know better than to expect Pantone matching in the GIMP, though.

EA Let's turn to the recent SCO lawsuit against IBM, which threatens to engulf the rest of the Linux industry. You've been quite involved with this—Linus Torvalds nominated you as a representative from the open source community to a panel to examine the code, presumably under strict nondisclosure. First, could you explain what such a panel would do?

CD The panel was originally proposed by Don Marti at *Linux Journal*. He asked SCO and then Linus who should be on it, and Linus nominated me. The thing is, I couldn't see myself signing a nondisclosure agreement with SCO. Ian Taylor was able to, however, and reported his opinion on the code base, which can be found on the *Linux Journal* Web site (<http://www.linuxjournal.com/article.php?sid=6956&mode=thread&order=0>).

So then the question becomes, what would be the point of such a panel? Would it be an information clearinghouse? A code review board? I honestly do not know that a panel could make any positive contribution to the case past what Ian has already done.

EA Are you personally willing to be on such a panel? It could put you in a difficult position.

CD Sure, but I could not sign that nondisclosure agreement; it's just too restrictive. I'm fine with the panel. I also don't worry about jeopardizing my future with either IBM or the Linux kernel, as I'm not a kernel developer and I don't see that in my future.

EA So, if you're never going to be a kernel developer or an IBM employee, why not sign? Isn't the resolution of this problem going to require some people to be willing to sacrifice some future flexibility to let the truth run free?

CD Yes, but, luckily, Ian has done that. His opinion is

probably better qualified than mine on the topic.

EA Although Linux is not the same as open source, it is the best-known member of the open source family. Do you think that this lawsuit is the beginning of the end of open source? How will it play itself out?

CD No, I don't think it's the end. If SCO succeeds, I think it is more likely to end up that Linux kernel developers will be required to remove the offending code—something they would love to do this very second if SCO would be upfront about it. I can only imagine that SCO is trying to retain the right to sue those that are unable to remove the code. [See Greg Lehey's "Closed Source Fights Back" on page 56 of this issue.]

EA What are the big challenges facing open source today, other than the obvious legal situation with SCO?

CD Make no mistake, the SCO thing stinks, but software patents are the big danger. That Linux is particularly vulnerable to injunction is what should make people worried. That said, that is one of

those problems common to both free and proprietary software. It is also a point of obvious reform in how the United States handles patent litigation.

EA You've opened up some big issues here. You say that Linux is "particularly vulnerable." Why? How can we protect ourselves?

CD I should say "software" is particularly vulnerable to software patents. The U.S. Patent and Trademark Office has made a big mistake in granting patents as broadly as it has for "inventions" that, well, aren't very inventive.

EA What advice would you give to a company that is considering basing some substantial part of its operations on open source?

CD Dive in with both feet. You'll save money and have happier employees. If an open source project can solve your needs, you would be foolish not to consider it. If you decide that you want to take some open source

code or libraries and want to incorporate that code into a shipping product, you should probably find a person who knows open source licenses. We aren't really incorporating open source software directly into the Damage code base, but we know how to do it in the event that we go that way with some code or another. The code inclusion issue is a tricky one, but that's frankly true whether you are using a hunk of open source code or licensing some third-party code. {See Mike Karels' "Commercializing Open Source Software" on page 46 of this issue.}

EA What advice would you give to a young turk coding whiz, just getting started in the open source world?

CD Install Linux or BSD, buy every book by Richard Stevens, and have fun. If you're lucky, you'll reach the point where you are no longer the smartest hacker in the room.

EA And not study law? That seems to be a particular challenge today.

CD I am not well suited to the practice of law; the details are not the kind you can run through a compiler to detect the flaws. Others, however, seem to find it fascinating, so I would suggest those that follow that path look into what the EFF and ACLU are doing to protect the First Amendment in the digital realm and perhaps take part.

EA What would you like to have carved on your tombstone?

CD "Loved his wife and daughter very much." It's my goal that my wife and child choose "Beloved Father and Husband." Q

© 2003 ACM 1542-7730/03/0700 \$5.00.

acm
queue
tomorrow's computing today

Coming in September

Developer Tools

How They Help, How They Hurt

- IDEs: A Good Idea Gone Bad?
- Reverse Engineering from Binary
- Code Spelunking, How to Debug Someone Else's Code
- The Best Coding Tool Ever: Your Brain