Recollections: An Interview with David Greenman Lawrence (dg@)

BY TOM JONES

The FreeBSD project started out with contributions from many hands, but the early days of the project and the people behind our favourite Operating System haven't been covered in much detail. As part of the 30th Anniversary Issue of the *FreeBSD Journal* I set out to speak to those involved at the start of development and learn how they became involved.

This first installment is with David Greenman Lawrence, an early contributor who helped give FreeBSD its "high performance server" reputation. Further installments will follow in subsequent issues.

TJ: Can you explain generally what you were up to in the late 80s/ early 90s in the period before the start of the FreeBSD project?

DGL: This was the period of my early twenties, and I was involved in a lot of different things simultaneously and going in a lot of seemingly unrelated directions. For example, I was the Technical Director for a video production company that produced arts programs for Cable Access. I co-founded a company that was involved in establishing US trade in Portland's "Sister City" in the Russian far-east (this was in the early days of the new Yeltsin democracy in Russia). I founded a company that installed TVRO home satellite systems (the 12-foot dishes for C-band satellite, not the tiny Ku-band dishes of today), while also working as an independent contractor providing satellite uplink engineering services for a TV broadcasting company. I was also an independent contractor with expertise in DEC PDP-11 and VAX systems (mostly repairing customer hardware). And finally, I was a computer hobbyist with an interest in Operating Systems development. I hacked RSTS/E and VAX/VMS for fun. This was rather difficult, however, as DEC only provided limited source code for RSTS/E and nothing more than assembler listings for old VAX/VMS utilities. I learned to be a pretty good DEC machine code hacker! For a short time, I was also the President of the "Portland Computer Society" (a 501(c)(3) non-profit organization of local computer hobbyists).

Fun times, but it was my hobbyist interest in Operating Systems development that ultimately led to playing with the source code for 2.9 BSD, which I ran on one of my PDP-11 systems. This led me to Bill Jolitz's release of 386BSD 0.0 in 1992 and getting involved with the development of the "386BSD Patchkit" (a project led by Rod Grimes, a fellow Portlander). I built my first Intel PC - a 386SX with 4MB of RAM just for this purpose. However, Bill Jolitz wasn't really much of a "team player."He generally rejected and even publicly ridiculed patches and improvements that people offered and even eventually objected to the use of his "386BSD" trademark in connection with the Patchkit and an "interim" release of the patched system (which we were calling "386BSD Interim-0.1.5" at the time). He insisted that we stop calling it that, and, ultimately, this forced our group of hobbyists to organize our own development project, completely independent of Bill Jolitz.

TJ: How did you come across FreeBSD/the efforts that led to FreeBSD? There is a lack of accounts where people discuss how they found information, was it USENET? (if so, how did you connect?). Were any of the PC BSD efforts covered in more conventional media (magazines, etc.)?

DGL: Julian Elischer posted a message on comp.unix.bsd (dated November 17, 1992) about a series of mailing lists that he had set up for discussing 386BSD (Bill Jolitz's baby). I read the posting and subscribed to some of them—probably on the same day. Julian also set up some logins on "ref" (which was running 386BSD) for people to hopefully use constructively. I may have already had an account on that machine, however, prior to the creation of the mailing lists. I'm not sure because my real time access to the Internet was very spotty and limited at the time. Netnews and email were being delivered by UUCP, for example. I think I had to dial up another system using a modem for telnet access. Anyway, Julian's 'ref' resources wasn't the first contact with other 386BSD'ers. I know I had contact directly with Bill Jolitz and some other 386BSD enthusiasts. I knew Rod Grimes because of another Portland area project called "RAINet" (the "Research And Information Network"), which was an early-days attempt to get local computer hobbyists (Inter)connected to the newfangled "Global Internet," mostly using SLIP over dialup modems. Many of the local computer hobbyists at the time—some of whom worked at Intel or Tektronix, or attended or taught at Portland State University, knew each other through various events and social circles (including the Portland Computer Society that I mentioned earlier).

> It was my hobbyist interest in Operating Systems development that ultimately led to playing with the source code for 2.9 BSD.

TJ: Julian's machine isn't something I have heard of before. I'll certainly dig more in that direction in future projects.

DGL: Also, Nate Williams had set up a mailing list for the major players in 386BSD called interim@bsd.coe.montana.edu. Most of the discussion occurred about 386BSD, after 0.1, while we were waiting for Bill Jolitz to release 0.2—which seemed to be delayed indefinitely.

TJ: Could you tell me a little about how you came to write sendfile(2) and other optimizations that allowed ftp.cdrom.com to serve so much traffic?

DGL: Wcarchive (AKA ftp.cdrom.com) became my obsession and it was the driving factor behind much of my development with FreeBSD. Rod Grimes built the first PC-based version of it in 1993, but I soon became involved in dealing with the machine's daily reliability problems. I took over the management of both the hardware and software shortly after that. One of the first challenges was the very limited bandwidth available on Walnut Creek CDROM's 1.5Mbps T1 connection to BARRnet. Even in those very early days, wcarchive was a very popular FTP distribution archive for the most popular Shareware of the day. The T1 was maxed-out 24x7 with packet loss that exceeded 50% most of the time. In fact, the T1 was so overwhelmed, that the congestion caused major issues at BARRnet, adversely affecting other customers. Bob Bruce (cdrom.com owner) proposed upgrading the T1 to a (45Mbps) T3, but that idea turned out to not be practical (or perhaps even possible) at the time.

> With extensive testing and kernel profiling, it was obvious that there were significant performance and scalability issues in both the TCP/IP and the socket layer code.

I think it was someone at BARRnet/BBN who suggested the server could be moved to the Stanford University data center in San Jose (the main BARRnet/BBN POP in the region), which would mitigate the congestion issues with the T1. So, in September 1994, I built a new machine and deployed it at Stanford. Of course, no one really knew just how oversubscribed the T1 had been. BARRnet/BBN assumed the load would be just 1.5Mbps. Much to their dismay, traffic immediately jumped to over 30Mbps as soon as we brought the server online at the data center which caused a bit of a panic with the BARRnet/BBN network engineers. We continued to operate the server, but it was clear that BBN really didn't see the financial case for hosting such a bandwidth-expensive resource, and I think they had to put on some bandwidth caps to keep the cost under control. Ultimately, within a year or so, I had to find a new home for the machine—where there were no limits. FreeBSD's reputation was at stake here after all. An average user downloading from wcarchive couldn't tell the difference between the server just being overwhelmed and the network being congested, and I was also keen to set new records and push the boundaries of what a single server running FreeBSD was capable of doing. So, I found a new home at San Francisco based ISP "CRL Network Services". In February 1996, I moved the machine to CRL where we were given 100Mbps of bandwidth to start.

In the years that followed, it was a constant struggle to keep up with the demand. There were many hardware and network upgrades, but it was really the performance of FreeBSD where I focused. With extensive testing and kernel profiling, it was obvious that there were significant performance and scalability issues in both the TCP/IP and the socket layer code. Some of these issues could be optimized (and I did), but the Holy Grail really was much bigger than that—I needed to get rid of all copying of the file and network data as well. While there were some "zero-copy socket" tricks that had been implemented in other operating systems, these were kind of a mess architecturally, were difficult to use in the application, and still left performance overhead related to user-kernel context switching and many other issues. What was needed, I thought, was a magic system call that did pretty much everything—you just give it a file descriptor and a socket descriptor and it would send the contents of the file out to the network.

While attending a technical conference in 1998 (probably Usenix ATC), I told several colleagues about my sendfile() idea. Someone suggested that Sun Microsystems may have implemented something like the sendfile() I was proposing. I was very curious about the API (for compatibility), but when I reached out to Sun to find out about their API, I quickly found that the rumor wasn't true. I think someone at Sun suggested that something like sendfile() might have been implemented in HP/UX. Anyway, I didn't have any contacts at HP (or know if that rumor might also be false), so decided to move forward with my own API. What I came up with had more arguments being passed on the stack than any other syscall in FreeBSD, which concerned me a bit, but it was nonetheless the most efficient way to do it. sendfile() was a difficult syscall to write. It seemed to touch just about every subsystem in the kernel—file I/O, VM system, network buffers, etc. Anyway, the initial version of sendfile() reduced total CPU time on the server by about 75%, or in other words, made wcarchive about 4 times faster. It was a huge win.

TJ: Can you tell me what drove you to get more involved with the organization and management of the project?

DGL: I've been thinking about this question, and I finally figured out why I was having trouble answering it: The "What drove you" phrase makes an incorrect assumption--it implies that there were some forces that pushed me toward an organizational and management role, but that really wasn't the case. While some managers are pushed into their positions, I would call those unfortunate people "reluctant managers," and more often than not, they usually are not well suited for their job. For me, it was more of an attraction to a needed organizational role, and it was in my nature to fulfill that.

With that said, managing and organizing a freeware volunteer project is very different from managing employees in a for-profit, private enterprise. In an all-volunteer project with no budget and limited donated resources, a manager has almost no authority to command people to do things. If you try to command a volunteer to do a thing, they're likely to just get angry with you, ignore you, or go away. Instead, you have to be much more subtle by guiding people in a direction--through discussion, consensus, collaboration, and be willing to do (at least) the proof-of-concept first yourself. With a proper display of ambition and direction (or sense of direction), others will follow.

Oh, and "herding cats" also comes to mind. ;-)

TJ: Were there specific problems you wanted to address as a member of the core team?

DGL: Well, of course, my main focus in the project was the development of the software (and the kernel in particular). I was focused on reliability and performance, but also on the architectural direction of FreeBSD. I felt that having a Core Team was pretty important to managing the legendary quality of the BSD codebase. There was definitely a contingent of anarchists in the group, however, that objected to any sort of formal order and preferred FreeBSD to be just a cabal of unorganized developers, but that isn't what I wanted.

TJ: How did the project change during your time on core?

DGL: I was a Core member for a total of 10 years. The first 8 years as a founding member of the unelected/self-appointed Core Team, and then 2 years on the first elected Core Team.

The first Core Team started out as a collection of very talented software developers that had a common admiration for BSD Unix and a keen interest in continuing the legacy as an open-source project. In the early days, it was mostly about just trying to make FreeBSD work reliably enough to be useful. As FreeBSD became a thing in its own right, the size of the project grew with over 300 developers who contributed all sorts of things that I never would have imagined in the beginning. Jordan's wonderful idea of the FreeBSD ports tree, for example, was huge in furthering the adoption and ease of use of the system and it attracted a large number of additional (ports tree) contributors as well. This is all good, except that as the development team grew, so did the diversity of opinions, frequency of disagreements, and challenges to the project's leadership. With the Core Team being self-appointed, there was an ongoing question about where the Core Team's authority comes from. This ultimately led to the Core Team reorganizing into an elected body—elected by the developers. I have to say that this didn't really change anything with respect to how things operated day-to-day within the Core Team, but it did perhaps give a little boost of legitimacy to our authority.

The project changed in many other ways as well. With FreeBSD maturing into one of the best server operating systems available, the project was increasingly able to attract donations and specific project sponsorships from various corporate users. These were a bit difficult to make and accept in some cases, because FreeBSD was, in fact, not a legal entity. Even the FreeBSD trademark had to be held by Walnut Creek CDROM, and I personally owned the freebsd.org domain name. This all changed, however, when Justin Gibbs (a Core Team alumnus) founded the FreeBSD Foundation in December 2000, as a US non-profit 501(c)(3) legal entity.

TJ: What is the lasting legacy of FreeBSD?

DGL: There are so many legacies to choose from. The fact is the average human on planet Earth is using software developed in part by the FreeBSD Project every day--from shortly after they wake up in the morning until they go to bed at night. If you use a mobile phone based on Android or iOS, then these platforms borrowed significantly from FreeBSD for their libraries and user applications. If you use Microsoft Windows, Apple MacOS, or iOS, then you're using a kernel networking stack that mostly came from FreeBSD. If you sit down at night to watch some streaming movies on Netflix, then that content is being served to you by servers running FreeBSD. If you're a gamer with a Nintendo Switch console, then you're using a platform powered by FreeBSD. If you do online banking, or trade stocks on the stock market, or ship a package around the world, then you're probably doing it with servers that are running FreeBSD (although I could never get them to admit this publicly!).

In all the cases of borrowing code or completely basing platforms on FreeBSD, there is a reason why companies have chosen to use FreeBSD instead of Linux. It's the most important legacy of all and something we inherited from BSD before us. If you ask a software professional what the de facto standard most permissive software license is, they'll tell you "the (N-clause) BSD license." From BSD, it was the 4-clause license, but FreeBSD took that a step further and cut that down to just a simple 2-clause license. Pretty much what it says is that as long as you don't claim to have written the code yourself, you can use the software in whatever way you want. You can change it to suit your needs and keep your changes proprietary. You can make a ton money from it and not give a penny back to the original authors. It means that it is truly FREE software in every sense of the word. I can tell you that this was forefront in my mind (we ALL had this in mind) when the project adopted the "FreeBSD" name on June 19, 1993. Although it wasn't planned or even thought of at the time, it was a happy coincidence that this also occurred on the anniversary of "Juneteenth"—a day celebrated as the day that slavery formally ended in the United States—June 19th, 1865.

FreeBSD's lasting legacy is the very concept of truly FREE software.

FreeBSD's lasting legacy is the very concept of truly FREE software.

TOM JONES is a FreeBSD committer interested in keeping the network stack fast, during the day he manage FreeBSD development at Klara Inc.