BSD

SUCCESS STORIES

11 Success stories with BSD

Adventures in BSD

How BSD Keeps Me Sane

FreeBSD at Shannon Medical Center

BSD in a Panic

You Haven't Had E-mail Since When? FreeBSD saves a dot-org, and maybe me, too!

A FreeBSD Success Story (and DragonflyBSD too)

BSD In a Microsoft Office

Museum Guardian

OpenBSD Saves the Day

A FreeBSD Implementation

Open Source Software in Co-operation Ireland

O'REILLY®

Introduction

The BSDs have earned a reputation for stability, security, and performance. That's no accident. Their Unix heritage provides the power. Careful attention to detail and correctness on the part of their developers provides the finesse. This has produced highly-reliable systems that are easy to administer, flexible to any task, and compatible with tens of thousands of pieces of open source software.

No wonder BSD is a popular choice with ISPs and the basis for Apple's Darwin!

The BSDs are also proving themselves popular with both sysadmins and users looking for alternatives to expensive, proprietary, and inflexible operating systems. This collection of BSD Success Stories—written by ordinary users--shows just a few of the many ways in which the BSDs solve real problems all day, every day.

Adventures in BSD

by Michael Josefsson

I am sometimes asked “Why do you use FreeBSD?” My usual ‘auto-response’ is “Because it is the best!” While I truly think it is the best, this reply does not really answer the question. I hope this article will better explain my feelings.

Way back in 1994 I became the manager of a hardware laboratory at my old University. I was to manage the labs in all aspects, from computers down to data sheets, pliers, and students. With over 300 students attending each year, this was a very time-consuming task. Of the three labs, only two had computers and networks in them. In the third lab, 16 serial lines were connected through a Xyplex terminal server to a computer of unknown pedigree. (It later turned out to be a Sun 3.) I had been into computers since the early 80s and PCs since 1989, but I had never touched a network of any kind.

The two labs were connected by a Lantastic LAN. Only one area of the server’s disk was shared and every student stored his own home-directory on it. As long as everything worked fine, and it generally did, I had better things to do than trying to understand what this network setup was all about. The odd downtime was mostly due to students tampering with the RG-58 coax that ran from one machine to another continuing on into the next lab. Problems were easy to locate. I knew the coax had to be terminated with a 50-ohm BNC terminator and moving this terminator to strategic places along the cable always led me to the culprit.

Then there was a software upgrade. The labs had been running MS-DOS with a DOS-based compiler for many years. The new version of the compiler required MS Windows. (At the time this meant Windows for Workgroups 3.11.) I could not get the old LAN software to run under Windows. I don’t remember why, perhaps I was a victim of my own ignorance. After reading the Windows docs I managed to set up a common workgroup for the two labs. With a 486/66 Windows box working as a file server we again had a fully functioning lab.or so we thought!

I found that it was possible to access the server’s shared area and was quite pleased with my doings. I compiled some typical code from a client machine and everything ran as expected. I did not test network performance under a higher load. The idea never crossed my mind.

During one compilation, several large, multi-megabyte files were created in the project’s home directory. This was in itself not a problem as the network’s bandwidth was not saturated. However, all this data moving in and out of the server turned out to be detrimental to its health. With 10-15 clients chewing away, the load on the server increased. After a period of time ranging from a few minutes to several hours, the server would unexpectedly crash. A reboot got everything working again. All source files had been auto-saved before entering the compilation stage so it was easy to start over and hope for more success the second time. Still, the situation was less than satisfactory.

At this point, OS/2 entered the story. In the outside world, Windows 95 was being deployed. I got my hands on an old OS/2 2.1 demo CD. After several problems I got it up and running. I thought it was cool and ordered my own OS/2 Warp soon after. Warp was my favourite desktop for years to come. I really liked the smooth user interface and enjoyed the increased stability compared to the Windows version I had used earlier. A co-worker with a similar interest in computers used Windows 95 and had to re-install every three to four months. The system would somehow clog itself up beyond recognition. My OS/2 machine ran happily all the time. Great! Except for the RSA DES Challenge. Being outside of the United States I took an interest in the Swedish-run SolNet DES-attack. With SolNet's limited resources there was quite some time before an OS/2 executable was published. The situation annoyed me.

So what to do? I was unwilling to give-up OS/2 and surrender to Windows 95. I had started to experiment with Linux, a Unix-like system but felt very little enthusiasm for it. Because of an earlier experience with HP-UX, I was under the impression that Unix only represented an extremely complicated way of doing things, and therefore, Unix was ruled out.

At this time, there were executables for FreeBSD available. Not knowing much about FreeBSD, I made an FTP install of 2.1.7-STABLE. The DES-client ran as expected. I figured out nice commands; kill -STOP, kill -CONT and that putting an ampersand (&) after the command line ran a job in the background. Cool! I could manage every aspect of the program. A taste of a new world! This was very enticing.

My success was short-lived. There was soon a new DES client requiring newer libs, which forced me to install 2.2.2. This time I ordered the 2 disc FreeBSD set from Walnut Creek sometime around June 1997.

Before the summer I had plans to replace the lab server with OS/2, but during the summer, I experimented with FreeBSD and learned about TCP/IP, Samba, FTP and Unix.

The client machines in the lab were using Windows 95 and since we had to get rid of Windows for Workgroups anyway, I installed FreeBSD on each and every machine. I made further experiments with the r*-services (rsh, rcp, ruptime, etc.) among other things. All in all, the experiments made me confident with managing FreeBSD. In the beginning of August, when I had to wire-up the lab's new infrastructure, I made FreeBSD run the lab for me. That turned out to be a wise move; not one spontaneous reboot occurred during the years to come.

With a busy lab, reliability is of major importance. The logs show students accessing the server around the clock. FreeBSD really is reliable; the server's longest uptime to date is 220 days.

By now, things were moving fast. The third lab was also computerized. (The introduction of Microchip's PIC line of processors necessitated this.) I added a couple of hubs and the labs were interconnected. The next summer I set up a local name server and a private domain on a FreeBSD box with two NICs—one for the internal net and one for external access. There was no forwarding between the two interfaces. This setup allowed me, as admin, to download software which I then installed on the clients. Each project team in the lab had its own home, with permissions fixed so they could not peek at each other's work. They could also edit files at home and upload via FTP. I added the system user as a way to keep a central repository of programs and files that every client would need—Adobe Acrobat for Windows for example, printer drivers, upgrades and other stuff.

Until now all data sheets had been in a folder in my room. The projects' supervisors had access to the folder and make copies which they handed out to the project teams. There were 95 steps to the photo copier, and with 95 steps back from it, handling the data sheets became more and more cumbersome as a variety of components were added. What to do? Install Apache! Putting the data sheets on the server in PDF-format has eliminated the actual paper handling. This was a truly brilliant move that I wish I had thought of earlier. Now each project group could peruse the available components' data sheets, both from home and when in the lab before committing the device into their project. Some students printed out data sheets of course, but I see a trend of more and more students reading the specs directly off the screen.

There is a lot of activity in the labs. With up to 75 students there at the same time, not all of them can be busy with their assignment; installing MP3s on the clients has become a popular pastime. It is quite possible that the especially zealous students fiddle around with the system settings of Windows 95 and cause damage! To simplify the Windows re-installation, we have made a complete 2 gigabyte image of the Windows partition which easily can be downloaded and written onto the machines. While this may sound rather brutal, it works well in practice. In fact, every machine in the lab was born this way.

Needless to say, the client machines dual-boot between Windows 95 and FreeBSD. It is a particular pleasure to find the students shutting down Windows and rebooting into FreeBSD because they feel more at home with it.

Earlier I mentioned we had a Sun 3 box. That particular machine has gone to the scrap heap and the FreeBSD server has replaced it. Using the ports, I installed the m68k-coff cross-compiler and spent some time rebuilding the libs needed for our particular hardware. We were able to get rid of gcc-1.40 and now use gcc-2.8.1 for our in house built 68008 computer card.

Some small bits and pieces complete the picture. The server has been endowed with a Tandberg SLR-5 tape backup. Once a week I do a level 0 dump and every night a cron job initiates a level 1 incremental backup. The machines' disks are mirrored with vinum(8). Thus the entire /usr is safeguarded from disk failure. To ease new installations and administration, each client has a P:\ (P for program) directory with all non-standard Windows 95 software. It is now sufficient to install new software on only one client; it will become available to the other clients at the next log-in. Every home share is attached as Q:\ on the client. To this end, Samba is configured as an NT domain controller. Log-in scripts handle the actual attachment of the devices at log-in time. On the FreeBSD side I use NFS and amd(8) to automount the user's home directories upon log-in.

I have unfortunately left OS/2. I really liked the Workplace Shell, but the functionality built into even a basic FreeBSD system makes me more productive. I have now been using FreeBSD exclusively for over two years for all my desktop needs. My experience is best summarized by something I saw someone post to one of the FreeBSD mailing lists: "I'm home."

Michael Josefsson is a research engineer at Linköping University, Linköping, Sweden.

Originally published as Daemon News: Adventures in BSD
www.daemonnews.org/200107/adventure.html

How BSD Keeps Me Sane

by Isaac Levy

I am a web application developer, and work with Zope and ZEO clustering, as well as web application performance tuning and clustering. I have chosen to work with the BSDs for years for several reasons.

When working as a young *NIX admin for my first Open Source based company, I had to perform install-monkey tasks for Red Hat systems. The process was excruciating, as it started with the installer (simple enough). I spent the next three hours removing and disabling all of the games and miscellaneous daemons running on the system.

After my first FreeBSD minimal install five years ago, I ran top and expected to learn a whole new set of things I had to turn off when, to my surprise, top didn't even fill an 80x24 terminal! I then proceeded to find out that man pages were actually cherished and stellar on the BSDs, and have successfully used the BSDs as the core of every project since.

Recently, I worked on a Zope project on another sysadmin's network, then running Mandrake. I had never worked on Mandrake before, and was immediately totally lost in reverse-engineer and second-guess someone else's brain to run our web apps and Object DB, which was extremely costly, as the servers needed to run in a fairly advanced cluster setup. I lost hours on the project, and in the end, lost sleep making the system run in a manner I found comfortable. With that, all of the applications I'm used to compiling myself, or installing via ports, were all part of the base install—and with configurations completely customized to someone else's needs and massively undocumented!

I suddenly deeply began to appreciate something in the BSDs which I have always taken for granted, uniform adherence to standards and well-documented working processes which free me to solve very specific problems. The consistency is so well executed in the BSDs, it's completely transparent and therefore easily forgotten.

To make things worse, three weeks after the project ended, the boxes were cracked due to a mail related configuration error, exploiting a hole that the GUI mail administration tools could not close. The system was now a warez and IRC server. My client decided to switch to Red Hat in light of the compromise and set up two new Red Hat boxes. With a new set of what I felt to be un-unix like configurations and conventions and lackluster documentation natively available, I hacked my way through rebuilding the web-applications—nearly from scratch—forcing a completely different configuration in the end. The changes and the costs incurred by the client made them choose not to pay me to document the new system and changes properly. (I couldn't bear that idea and provided basic configuration and system notes.) This refreshed my deep appreciation for the BSDs. For the bulk of my projects, I hardly think about the BSDs because things that truly work transparently support everything.

Isaac Levy is an independent web application developer and consultant in New York City, with a focus on Open Source solutions and development. Isaac has worked in Open Source web hosting, in Zope development, and with BSD Unix for years in various forms. He aims to bring experience in large enterprise software applications to small and medium businesses.

FreeBSD at Shannon Medical Center

by Andrew Gould

Shannon Medical Center is a not-for-profit trauma center in West Texas that acts as the safety-net hospital for the surrounding area.

As the health care industry is under pressure from increased costs of providing health care, coupled with reductions in reimbursements for providing that care, the importance of providing timely information to hospital management has never been higher.

Shannon Medical Center uses a Decision Support System (DSS) from a commercial vendor. In 2000, a supplemental decision support database was built using FreeBSD and PostgreSQL. This system integrates key data from the commercial DSS, other sources of internal data that cannot be imported into the commercial DSS, and external data for benchmarking purposes. Analysts access the data using preexisting database clients via ODBC, eliminating the need for additional training. The initial system was built on an existing desktop computer for proof-of-concept. Due to the project's success, the system's hardware was upgraded to a database server in 2001.

In the words of Andrew Gould, Manager, Clinical Decision Support:

The FreeBSD/PostgreSQL combination is extremely flexible and robust. FreeBSD provides a stable and efficient environment for the database server. The relational data model facilitates flexible analysis that is unsupported by the commercial DSS. The database is used primarily for financial, clinical and market share analysis. Some of the results of this implementation include:

- Hospital management receives information that was previously unavailable.
- Increased efficiency—highly focused, clinical data analysis reduces the number of medical charts to be reviewed manually.
- Unplanned downtime since/including installation = 0.
- License fees since/including installation = $0.
- Support expenses since/including installation = $0.

The flexibility and low cost of this project reduce our dependency upon vendors and free us from many budgetary constraints. We can adapt the system whenever we want, however we want, to meet our changing needs.

Andrew Gould, CPA, performs financial and clinical data analysis for Shannon Medical Center. His primary tool for data integration is a PostgreSQL database server running on FreeBSD. Andrew has been using FreeBSD at both work and home for four years. Andrew has a BS in Education and a BBA in Accounting from the University of Texas at Austin.

BSD in a Panic

by Michael Lucas

My employer's main business is designing Web applications, but once those applications are built our clients turn around and ask "Where should we host this?" That's where I come in, building and running a small but professional-grade data center for custom applications.

As with any new business, our hosting operation had to make the most of the resources we had. Our resources were strictly limited to cast-off hardware from the web developers and free software. The only major expense was a big-name commercial firewall, purchased for marketing reasons rather than technical ones. With FreeBSD and a whole mess of open-source software, we built a reliable network management system that provides the clients with a great deal of insight into their equipment. The clients, of course, pay for their own hardware and so have fancy high-end rackmount servers with their chosen applications, platforms, and operating systems. We've since upgraded the hardware — warranties are nice, after all! — but have seen no need to change the software.

One day, a customer that had expected to use very little bandwidth found that they had enough requests coming in to use close to twice the bandwidth we had for the entire datacenter. This affected every customer, slowing the entire hosting environment to speeds comparable to a snail in molasses. If your $9.95/month web page is slow you have little to complain about, but if your $50,000/month Web application is slow you pick up the phone and scream until it stops.

To make matters worse, my grandmother had died only a couple days before. Visitation was on Tuesday, and the funeral was Wednesday morning. Monday morning I handed the problem to a minion and said "Here, do something about this." I knew bandwidth could be managed at many points: the Web servers themselves, the load balancer in front of them, the commercial firewall, or even the router.

Tuesday after the visitation I found my cellphone full of messages. Internet Information Server can manage bandwidth — in eight megabyte increments and only if the content is static HTML and JPEG files. With several Web servers behind the load balancer, that fell somewhere between useless and laughable. The load balancer did support traffic shaping, if we bought the new feature set. If we plopped down a credit card number, we could have it installed by next Sunday. Our big-name commercial firewall also had traffic shaping features available, if we upgraded our service level and paid an additional (and quite hefty) fee for the feature set. That left the router, which I had previously investigated and found would support traffic shaping with only a flash upgrade.

I was on the phone until midnight Tuesday night, making arrangements to do an emergency OS upgrade on the router on Wednesday night. I had planned to go to the funeral in the morning, give the eulogy, go home and take a nap, and arrive at work at midnight ready to rock. The funeral turned out to be more dramatic than I had expected and I showed up at work at midnight sleepless, bleary-eyed, and upright only courtesy of the twin blessings of caffeine and adrenaline. In my email, I found a note that several big clients had threatened to leave unless the problem were resolved Thursday morning. If I hadn't already been stressed out, the prospect of choosing a friend to lay off would have done the trick.

Still, only a simple router flash upgrade and some basic configuration stood between me and relief. What could possibly go wrong?

The upgrade went smoothly, but the router behaved oddly when I enabled traffic shaping. Over the next few hours, I discovered that the router didn't have enough memory to simultaneously support all of our BGP feeds and the traffic shaping functionality. Worse, this router wouldn't accept more memory. At about six in the morning, I got an admission from the router vendor that they could not help me.

I hung up the phone. The first client who had threatened departure would be checking in at seven thirty AM. I had slept four hours of the last forty-eight, and had spent most of that time under a fiendish level of emotional stress. I had already emptied my stash of quarters for the soda machine, and had been forced to pillage a co-worker's desk for his. The caffeine and adrenaline that had gotten me to the office had long since worn off, and further doses of each merely slowed my collapse. We had support contracts on every piece of equipment and they were all useless. All the hours of work I had put in, and my team before me, left me with a sum total of absolutely nothing.

I made myself sit still for two minutes simply focusing on breathing, making my head stop sliding around loose on my shoulders, and ignoring the loud ticking of the server room clock. What could be done in ninety minutes — no, now only eighty-eight?

I really had one only option. If it didn't work, I would be choosing someone to lay off or filing for unemployment myself.

6:05 AM. I slammed the floppy disk into the drive and started downloading the OpenBSD install floppy then grabbed a spare desktop machine, selecting it from amongst many similar machines by virtue of it being on top of the pile. The next few minutes I alternated between hitting the few required installation commands and dismantling every unused machine unlucky enough to be in reach to find two decent network cards. By 6:33 AM I had two Intel EtherExpress cards in my hands and a new OpenBSD 3.5-snapshot system. I logged in long enough to shut the system down so I could wrench the case off, slam the cards into place, and boot again. OpenBSD's built-in PF packet filter includes all sorts of nifty filtering abilities, all of which I ignored in favor of the traffic-shaping functions. By 6:37 AM I was wheeling a cart with a monitor, keyboard, and my new traffic shaper over to the rack.

Here, the killer problems manifested. I didn't have a spare switch that could handle our Internet bandwidth. The router rack was jammed full, leaving me no place to put the new shaper. I lost almost half an hour finding a crossover cable, and when I discovered one it was only two feet long. The router, of course, was at the top of the rack. Fortunately, if I put the desktop PC on end and left it sitting on the cart, the cable just reached the router. I discovered this about 7:10 AM. I stacked everything so it would reach and began re-wiring the network and reconfiguring subnets.

I vaguely recall my manager coming in about 7:15 AM, asking with taut calmness if he could help. If I remember correctly, as I typed madly at the router console I said "Yes. Go away."

At 7:28 AM we had an OpenBSD traffic shaper between the hosting area and our router. All the client applications were reachable from the Internet. I collapsed in my chair and stared blankly at the wall.

While everything seemed to work, the proof would be in what happened as our offending site started its daily business. I watched with growing tension as that client's network traffic climbed towards the red line that indicated trouble. The traffic grew to just short of the danger line — and flatlined. Other clients called, happy that their service was restored to its usual quality. (One complained that his site was still slow, but it turned out that bandwidth problems had masked a problem with his application.) The offending client complained that their web site was even slower than before, to which we offered to purchase more bandwidth if they'd agree to buy it.

Today, I have two new routers and new DS3s. The racks are clean again, without extra cables from thrown-together solutions. The desktop machine has been replaced by two OpenBSD boxes in a live-failover configuration, providing protection for our big-name commercial firewall as well as shaping traffic.

My thrown-together OpenBSD desktop machine is sitting in the corner of the hardware room. The sign on it says "DO NOT TOUCH: EMERGENCY USE ONLY." Should the clock tick down on some other problem, well, at least I won't have to spend the thirty minutes it took to install.

Michael Lucas lives in a haunted house in Detroit, Michigan with his wife Liz, assorted rodents, and a multitude of fish. He has been a pet wrangler, a librarian, a security consultant, and now works as a network engineer and systems administrator with the Great Lakes Technologies Group. He's the author of Absolute BSD and Absolute OpenBSD, and is currently preparing a book about NetBSD.

You Haven’t Had E-mail Since When? FreeBSD saves a dot-org, and maybe me, too!

by Kevin Kinsey

The phone rang… “Hello, this is Susan*…this may seem strange, but I was thinking about some problems we’ve been having, and I thought you could help us. Could you come down sometime and take a look at our computers? We’re really struggling here.…”

She went on in some detail about the state of things at her office. I assured her I’d think about it, and hung up…like Samuel Morse, thinking “what hath God wrought?” I’d created an HTML page or two, but I was no computer genius. I had a computer and I had friends and a smart brother who worked on them and with them, and that was about it. My training was in a totally unrelated field; although I was currently in a “career path crisis”, I was neither computer repairman nor network consultant.

The caller was a friend, personal secretary to the pastor of the largest church in a nearby city; I was on the ministerial staff at the smallest church in the next town up the road, and as far as she knew, our computers were always working. (Of course, she didn’t know that we never asked them to do much.…)

I went down there; things were a mess. It was September 2001; in addition to frightful things in the world at large, Code Red and Nimda were ravaging computers globally. The machines at Big Church* ran that “Wonderful” OS (well, it starts with a “W”, anyway) and some were infected. Furthermore, although everyone had high-speed Internet, almost no one was using e-mail to communicate. It was strange; an organization with their own domain name, over a dozen paid staff members, and a budget of nearly a million dollars still depended on sticky notes for intra-office communication and spent a lot of phone time just hashing out details of upcoming activities with their members. I didn’t think that was possible in the 21st century.

After removing Nimda from a couple of machines, I asked about their e-mail. “Oh, that hasn’t worked for a while…” Curiously, I opened a couple of outboxes on their POP clients. The most recent sent items were from November 2000, ten months past!! Inboxes? Same story.…

Timidly, I asked, “Where’s the server?”

“The server? Hmm, that would be Brenda*’s machine.”

Brenda was in charge of payroll and financial and membership records. Her computer was a 500 MHz Athlon with 128 MB of RAM. The OS looked a tad different from everyone else’s. Whatever it was, it didn’t run so well, and there were some obvious reasons. This machine was running an “Advanced” version of that same very popular OS, and between being infected with Nimda and attempting to operating as a PDC, along with IIS SMTP, and (for some strange reason) anonymous FTP servers, plus Brenda’s browsing (she had been smart enough to get a mail account at Yahoo!) and the large financial package she used, this machine was taxed past the limit of its resources. Furthermore, I couldn’t believe that they would spend so much money just to run Web and Mail for 12 users. Of course, once I removed the virus I discovered they hadn’t spent quite enough! There was no POP/IMAP service on this box, it was not firewalled, and most of the services it did offer were badly configured or not working properly.

I called a friend of mine who worked for a small ISP. He’d been kind enough to do a little open-source evangelism with me, and had given me a shell account on a virtual server that hosted the company’s website. I’d learned a little about telnet and ftp, knew ls and cd, and could use pico to edit text or HTML files. I was certainly no guru.

"What is that OS I've been talking to on your web server? Some kind of Linux?" I asked. (His kids teethed on little penguin toys…he had a five-box LAN in his house…these days I'd call him an übergeek…)

"No, that's a little more 'hard-core'. Probably a-BSD. Maybe FreeBSD. Why?"

Hard-core? (Brief shiver)…I swallowed hard and gave him an overview. He told me I should do something about the situation (besides letting this server spew viruses all over the place) and that an open-source OS like FreeBSD would probably be just the thing I needed. "You can read everything you need to know on the 'Net." So, with at least a bit of trepidation, I pointed a browser at www.freebsd.org. "All you need is a pair of freshly formatted floppies and these instructions.…"

I went home and read up a bit (I grabbed the entire FreeBSD Handbook in .rtf format.) I printed several pages. I formatted the floppies and read the instructions.

The next day I was back. I found, in an upstairs closet at Big Church, an old clunker that wasn't up to modern specs. It booted, and had a rather small HDD, but it didn't clack too much and the fans sounded OK. I managed to stick a NIC in it and cable it to their switch. Booting from the floppy, holding tight to a couple of pages I'd printed from the online Handbook, I clunked away though sysinstall. If I remember correctly, it took two, maybe three attempts to navigate the menus correctly — it was a little different than I was used to, but the alternative was paying thousands of dollars to someone who didn't even care to come and check up on the machines they had installed and (apparently) hadn't been there for 10 months.…

To make a long story shorter, once I navigated the installation menu properly, FreeBSD installed itself. I found myself in a rather familiar CLI environment. In less than a day, we had SMTP and POP3 capability inside the building. Then we moved the HTTP service to this little clunker. Now Brenda's financial app would work better because she wasn't sitting on an overworked box with no resources. Later we started in-house DNS to ease the load on the ISP and router. As a bonus, all this was accomplished with free software and hardware that was waiting for the dumpster.

That was almost three years ago. Today, that church hardly ever calls me with problems, and when they do, they're never server-related. I do think I've proactively changed every piece of hardware on that box, as parts have become available, and as their website has grown way past the size of the original HDD, but the server still bears the same name, and today provides even more services both to the LAN and the outside world.

As for me, I still have that part-time ministerial position, but also have a growing business in networking, troubleshooting, and web application programming that has added substantial income to the family budget, and given me some direction for that career path crisis.

This is no testimony of "what FreeBSD can do for your Fortune 500 shop". Frankly, unless I become a better businessman, I'll never make a killing. But there are lots of little problems everywhere that people need good tools to solve, as well; I found a good tool in FreeBSD.

I imagine FreeBSD can offer the same things to anyone that it offered to me, and even much more. But I know that because of its stability, excellent documentation, helpful user community, highly versatile design (and of course its cost), FreeBSD was a seminal force in this growth in both my life and the life of Big Church. Guess what pretty desktop I'm writing this on?

C'mon, you already know!!

(*names changed to protect the naive [except for me])

Kevin Kinsey owns DaleCo, S.P. (*www.daleco.biz*), a small consulting firm located near the equally small town of Jasper, Missouri, where he lives with his wife, children, and one cat. Starting in 2001 with two customers and little experience in the industry (other than hanging around the back room of a local ISP), the company continues to experience slow yet steady growth. Prior to "Y2K", Kevin had an 11-year career in public education, teaching vocal and instrumental music in 3 Missouri school districts. He claims to have no free time (and therefore no hobbies); one of his life goals is to convert his brother (an NT system administrator) into a vocal FreeBSD advocate.

A FreeBSD Success Story (and DragonflyBSD too)

by Scott Robbins

When I first came to my company, which I'll call Astral (name changed to protect the guilty), great changes were underway. The company had outsourced IT prior to the arrival of my boss and myself. Users, running Win9x, could do whatever they wished with their computers. I spent most of my time in the first few weeks simply reinstalling totally borked systems.

Neither my boss nor myself were big Microsoft fans. Although I have various MS certs, both he and I preferred various flavors of Unix and its clones. He works in our NJ warehouse and I was responsible for the NYC side of things.

Although my *nix experience had begun with Linux, at some point FreeBSD became my O/S of choice. I had wide latitude in what I did on the NYC side, so long as I kept my users happy. So, I began investigating what I could move to FreeBSD.

My job can involve anything from helping a user put the taskbar back at the bottom of the screen to rebuilding a crashed server. We are using an NT4.0 domain, and the servers are aging and having more and more problems.

Additionally, a more than adequate box from four years ago soon becomes outdated. The two gigs of storage space for one of the companies on a server was rapidly filling. The half gig of the C: drive was nowhere near enough as printing jobs got larger, and users would receive error messages about there not being enough disk space to complete a print job.

Due to the nature of the business and financial constraints (one of my mentors once said that the top two OSI layers are politics and money) we often had to use a band-aid approach. However, as things became a bit calmer and people saw that we not only knew (more or less) what we were doing, but also saved the company a good deal of money, we had a little more freedom.

One of the first steps was to take some of the file sharing off of the NT servers and put them on FreeBSD boxes running Samba. This was easy enough and successful. Even using older discarded MS workstations, reinstalled with FreeBSD, freed a great deal of space on the servers.

Next, with a bit of trepidation, I began using CUPS (sometimes said to stand for Can't Usually Print Stuff). However, it went quite well. It was fairly straightforward, allowing the MS boxes to use their client drivers.

Our company works in fashion. We must store a great many images of dresses. Again, space had been an issue, and I migrated many of these directories successfully over to the BSD boxes.

At present, future plans involve having two somewhat more powerful boxes as servers (one is already in use as a print and file server) and having them back up the images to each other using rsync. The file and print serving are both completely transparent to the users. We will shortly be moving the DHCP server over as well.

I have a couple of other older, discarded boxes sitting on our various subnets. (Astral is actually an umbrella company. Each division has its own showroom and production room, all in different buildings on different floors). These are very handy when I have to reinstall a user's machine or transfer their files when I upgrade a computer. Our network is old, and transfers are far faster when copying between two machines on the same subnet.

For a very small time investment, we have found FreeBSD has saved us a great deal of money and time. I have even been a bit daring, running 5.2.1 with no problems whatsoever on some not at all mission critical servers. I'm able to indulge myself enough to have my own personal workstation running FBSD 5.2.1, and only have to turn on an MS box to walk a user through something or to test our one particular custom application.

Having heard some interesting things about DragonFlyBSD, I have been playing with it on a few test boxes. As each release was a small download and quick burn, I simply put the latest one on CD as it's released.

I'm still just playing with DragonFly and had no plans to use it in a production environment at this point. It's more of a new toy than anything else. I like it, but am still a bit unfamiliar with it, and run into often amusing problems, such as installing it with the defaults, then following the README's instructions for updating without realizing that my /home partition (where it's putting source) is only a few hundred megabytes. However, its live CD came in handy recently.

Like many sys/net admins in an environment where most users are running MS, we have our problems with spyware, viruses and the like. A partner had opened an ecard from his daughter, clicked on something, and was suddenly deluged with popups. At times, such problems have been cured by installing the latest MS service packs, so that's what we tried to do.

Upon downloading and installing the first set of critical updates, his computer failed to boot. Ok, we'll boot it in safe mode and remove the update. No good. It would power up, then restart. Trying safe mode with command prompt also brought no joy. It seemed as if he might lose all files on the computer.

Enter DragonFlyBSD-1.0A_REL live CD. I used it to boot his laptop (a moderately new Toshiba Satellite) without problem. DragonFly had no trouble detecting the builtin ethernet, and using dhclient brought me onto the network. I was then able to mount his C: drive with mount_ntfs and from there, was easily able to copy his important documents and outlook.pst file over to one of our FreeBSD samba servers.

From there, it was a simple matter to use the machine's recovery CD, reinstall the O/S, and replace his files from the samba box.

While this isn't anything amazing, I did find it nice that I'd already put DragonFly into production use, in a manner of speaking.

Scott Robbins is a Junior systems/network adminstrator for a mid-size Manhattan fashion company. Although Microsoft certified, he far prefers working with Unix, and desperately hopes to find a job as a Junior Unix Adminstrator. His web pages, with clear (though sometimes mistaken) explanations of various aspects of *nix: at *home.nyc.rr.com/computertaijutsu*

BSD In a Microsoft Office

by Joe Warner

After being introduced to the FreeBSD operating system back in March of 2000, I was so amazed with it that I wanted to find a way to use it in the office where I work.

Two years before that, I had gotten my feet wet with Linux, mostly tinkering with Caldera and Red Hat. I was impressed with Linux as well. So much so, that I joined a local Linux Users Group (*www.sllug.org*). It was during one of our user group meetings that I was introduced to FreeBSD. During the meeting, CDROM sets of FreeBSD 3.4 were handed out and I was lucky enough to get one!

The very next day, I installed FreeBSD on an old pc at home and was immediately amazed. The overall configuration and layout of FreeBSD, from the installation choices to the directory and file structure, were in a really logical order. Everything about it just seemed to make more sense.

One of the things I was most impressed with was how easy it was to install applications from the vast ports collection. I also liked how FreeBSD gives you the choice of which applications you want as part of the base install. Some OSes seem to include a lot of applications during the install process that you may or may not have interest in using. This can be especially troublesome if you have an old pc with a small hard drive. I like the idea of installing just the core operating system and then having the choice of going back and installing what I want. I mean, who really needs fifteen different text editors, when two or three will do?

After tinkering with FreeBSD at home for a while, amidst all the noise and distractions that go with a family setting, I decided to see if I could install it on an unused pc at work. The logic being that I would have the chance to learn more about BSD/UNIX at work, since, like most of us in this day and age, this is where I spend the bulk of my time.

As luck would have it, I was able to locate an older Compaq Deskpro pc that wasn't being used. It was a 200 MHz machine with 64megs of RAM and a 4 gig hard drive. It turned out to be the perfect choice. FreeBSD 3.4 installed easily on it, using the entire amount of space on the hard disk. During the install process, I was given the choice of either manually creating the necessary partitions or selecting 'A' and letting FreeBSD do it automatically. Since I had only installed FreeBSD once before, I decided to let FreeBSD do the dirty work. Before I knew it, I had a powerful, stable and functional operating system at my fingertips!

After spending a week or two of exploration, I discovered that the famous and powerful Apache web server was already running! I stumbled into this when I pointed the web browser on another workstation to the IP address I had given my FreeBSD system. I was immediately overwhelmed with excitement and fascination when I saw the Apache welcome screen. This discovery spawned a great idea. Previously, our office had never implemented the creation of a local intranet. I knew that the use of corporate intranets were wide spread and anticipated the need for such a useful source of help and information in our office.

I immediately set to work, creating web pages with the use of Netscape Composer, that contained helpful information that our users could draw upon when they needed it. I was amazed at how fast it was to access these pages through a web browser, rather than accessing the same information locally, from a file server or email database. This was especially apparent while dialed into our network remotely.

A few months after I put FreeBSD/Apache to work, news came from upper management that there was an interest in employing the use of intranets at all the remote field offices. Other web servers and content soon sprang forth and now we have a fully functioning intranet in place and more and more of our employees are showing interest and becoming involved. We have more than a few employees now who are acting as content publishers and are creating and maintaining sections of our intranet themselves. There are also more than a few employees and co-workers who are now interested in the BSDs and wish to learn more.

The Apache web server (http://www.apache.org) was a great find, but it was hardly the end of many discoveries I would make and ideas I would come up with. After more reading, research and tinkering, I had an FTP server running and after a few days, had a Samba server (http://www.samba.org) running as well. This proved to be invaluable, since now, I can create web pages from any workstation on our network and copy them directly to the web directory on my FreeBSD system!

Acceptance and interest in the various BSD Operating Systems, and what they are capable of, was growing. Not only were co-workers in my department taking notice, but my managers as well. Just the cost saving factors alone that come with the use of BSD, and other Open Source operating systems and applications, are hard not to take notice of.

Our division had recently moved to a brand new building in an industrial section of our city. The building was constructed from the ground up with our business needs in mind. Everything in the building has a spacious and contemporary look and feel, including a state of the art data center that can be seen through large glass windows from one of the main hallways. Our network is fast, at 100 megabits.

A few months after we moved into our new building, one of my managers voiced the need for some kind of network monitoring software or tool that we could employ to monitor the health of our network and identify problems. He was considering the purchase of an expensive, hand held monitoring tool that would cost thousands of dollars. I told him that I probably already had some kind of network analysis software on my FreeBSD machine and asked him to give me a little time to find out. I also told him that if I was able to find something that would do what we wanted, we could save our office and the company a lot of money. He agreed and I was soon searching the ports collection on my FreeBSD machine for just such a utility.

It didn't take me long to discover that I already had access to powerful network analysis utilities like The Ethereal Network Analyzer (www.ethereal.com) and Snort (www.snort.org).

I decided to start with The Ethereal Network Analyzer because it was the most GUI of the utilities that I had found and was a more tangible and suitable way to present network analysis information. Most of my managers and co-workers use Windows NT and have never had exposure to UNIX terminal screens or UNIX shell access, much less the BSDs or even Linux.

During my first attempt at using Ethereal to monitor our network, I was able to immediately identify a problem and report it so that corrective action could be taken. I noticed that one of our Lotus Notes Domino Servers, running on the IBM iSeries platform, was sending out a barrage of network announcements. I immediately went to our Lotus Notes Administrator and asked if he was aware of any problems. He said he hadn't heard of any and just when I was about to leave, one of our employees that was engaged in development on the server came in and said that she was unable to log on. It turned out that the problem was with the TCP/IP configuration and was soon corrected.

Right away, my managers could see that this was a useful and powerful utility to have in place and I still receive requests all the time to identify possible problems and collect information on individual nodes on our network.

Recently, I and other members of our department had reason to believe that an intruder was trying to gain direct access to one of the nodes on our network via telnet or by using a port sniffer. I used Snort to effectively capture packet information from this node.

After monitoring the suspicious activity for a couple of weeks, it was determined that the attempts were coming from a node with an invalid IP configuration and not an intruder.

When I first had the idea of finding ways to use Open Source operating systems and software at work, I thought I'd have a very difficult time doing this, since our office has been a Microsoft/IBM shop for years and these platforms and associated software are what our employees are used to using. I never intended to recommend the replacement of these platforms in favor of the BSDs or Linux but rather to employ and integrate the use of these platforms in a more cost effective and productive way.

For those of you who are considering employing the use of FreeBSD, NetBSD, OpenBSD or BSD/OS in your office or company, I would recommend that you install it on an available pc. Explore the huge number of applications, (currently over 10,500) in the ports collection and the many uses these powerful and unique operating systems have to offer. Start with things that are tangible and easy to see the benefits of and understand.

When I first started using FreeBSD, I was so amazed and taken with it that I could be heard preaching the BSD gospel almost every day. Keep in mind that most people will continue to use whichever operating system or application they are comfortable with. Don't be such an advocate that people become afraid to even mention the words "FreeBSD","NetBSD", "OpenBSD" or "BSD" around you. Remember, the louder you are, the harder it can be to hear you. The phrase, "Action speaks louder than words", certainly applies here. Quietly learn about the BSDs, how to use them, and offer the amazing demonstration when the opportunity presents itself. Soon, that old pc running one of the BSDs will be the honey that attracts the bees!

Joe Warner is a Technical Analyst for Siemens Medical Solutions Health Services Corporation and has been using FreeBSD as a server and desktop since October of 2000. Joe has lived in Salt Lake City, Utah for most of his life and enjoys *BSD, computing, history, and The Matrix.

Excerpted from Daemon News: BSD in a Microsoft Office
www.daemonnews.org/200103/adventure.html

Museum Guardian

by John Richard

I was contracted to analyze and improve a network for a non-profit museum after they had received a small government grant for IT. As most people are aware, most non-profit organizations are usually non-profit by function rather than by design, therefore the budget was minimal to non-existent.

After getting the lay of the LAN, I found very serious security problems, caused by a complete lack of knowledge and experience with IT at this organization.

In my report to the manager, I pointed out that having a MS server with no firewall, connected to an "always on" broadband internet connection was a bit risky to say the least. Having thousands of credit card numbers in a completely open shared folder fell under the "very bad thing" category. After explaining the risks of openly available shopping on the museums dime, smelling salts were used to revive her.

I explained that at a minimum, the first thing required was a rock solid firewall. "That sounds very expensive. We can't afford something like that now" she said. "Give me a day and it will be in place" I told her.

I had found an old but functioning Pentium 166 box on the premises, not being used. "What will this cost?"

"Nothing" I told her.

A new FreeBSD disciple was instantly born.

An additional NIC that I had found in the PC parts graveyard (museums never throw things out) was installed and FreeBSD was easily installed on the old machine. With a quick kernel rebuild, a few setting changes, and an ipfilter rulebase that I don't leave home without, the museum had a free, solid firewall. I also made the FreeBSD box the internal DNS, and DHCP server, as the existing unit went down more often than the stock market.

I created an IPSEC tunnel to my home systems using open source software from the FreeBSD ports collection, and was able to securely administer the network from the comfort of home, when needed, by installing Samba, also from the ports collection, on the FreeBSD box.

The little sentry functioned flawlessly and has continued to do so. My only concern was that one of the volunteer janitorial staff would throw out the machine due to its appearance. The server closet was just that, a closet, and with the ugly little box being mounted on what appeared to be a milking stool, it did not come across as impressive to the untrained eye, even though it was the guardian for the museum. A large sign, complete with skull and crossbones reading "FIREWALL. Do Not Touch on Pain of Death" fixed that problem.

Without the free availability, functionality, and stability of FreeBSD, a costly alternate solution would have been required, possibly causing the cancellation of wine and cheese parties.

FreeBSD saved the day.

Incidentally, the credit card numbers were immediately secured.

John Richard, commonly known as JR, now works for an ISP in the Kingston Ontario area, attempting to remove the speedbumps from the information highway. When not driving his wife and kids up the wall with geek-speak, he is usually riding his Harley, or fixing something for someone.

OpenBSD Saves the Day

by Jan Peterson

My company had recently been acquired. We were a small software development company and had been using CVS for our source code management. Our new parent company was larger, with an existing web presence that they intended to integrate our catalog system into. They were using Perforce for their source code management, and mandated that we hook into their system (i.e., the repository would be located at their site, in San Diego, and we would have to check files and work spaces in and out over the WAN from our offices in Salt Lake City).

Since we already had a T-1, it was deemed overly expensive to put in a point-to-point link. It was decided that a VPN was the best solution. The new company's IT/Networking team was convinced that putting a Cisco PIX in place at our office was the best solution (they were a Cisco shop and already had PIXes for their own firewall). They immediately started looking at different PIX models and Cisco's VPN technologies, with an eye towards being able to ship us a configured unit we could just drop in place (replacing our existing FreeBSD based firewall at the same time). Their estimate on a time frame was four to six weeks.

My developers were anxious to get working in the new source management system as soon as possible (with pressure coming from their new bosses who had mandated that we cut over within two weeks). They turned to me to find an interim solution that we could get running in a hurry. I immediately suggested an OpenBSD solution using the IPSEC functionality that was built in to the OS (this was around the time of OpenBSD 2.7). I got approval from my management, grabbed my corporate AMEX, and headed down to CompUSA.

I bought a couple of HP Pavilion desktop computers (no monitors, keyboards, or mice) for about $800 ($400 each) and headed back to the office. In a matter of a few hours, I had OpenBSD installed on both machines and was playing with the IPSEC configurations. One quick phone call later and FEDEX was on it's way to pick up a box containing one configured HP Pavilion. One day in transit and the box was in San Diego.

I got a call from the parent company's network guy the next day. He had received the machine and set it up right under his desk (where he could keep an eye on it). With another hour or two of fiddling with configurations to get both endpoints talking to each other, and getting each internal network routing the appropriate subnets to each other, we were able to bring up the IPSEC link and our developers were able to check their stuff into Perforce on the other end of the link.

Mission accomplished, thanks to OpenBSD! We had an IPSEC tunnel in place using 3DES and automatic key exchange.

The company did end up dropping $30k on a PIX for our location, and ultimately brought up a "supported" VPN connection. It only used single DES (the 3DES option on the PIX was "too expensive"), relied on static keys, and took six weeks to implement, but obviously it was superior because it cost a lot of money. The OpenBSD boxes were retired and I believe they ended up as home workstations for internal developers. Personally, I would have stuck with the OpenBSD solution. It was fast, cheap, and it worked. Normally, you only get to pick two of those!

Jan L. Peterson is a professional system administrator with 16 years of experience working with multiple Unix versions (and the occasional Windows machine). Laid off from his last job when the company was acquired by a direct competitor, he has spent the last couple of years as a consultant. More about Jan can be found at http://www.peterson.ath.cx/~jlp/.

A FreeBSD Implementation

by Travis Stevenson

A bit of history, I've been using FreeBSD since 2.0 got released. I started on my quest of all things BSD when I was working for Clark Development (PCBoard BBS Software) in 1994. I worked in sales and software duplication. When the development team started talking about TCP/IP, I wanted to learn what it was. I jumped on Aol and starting searching for TCP/IP. I found a lot of entries for UNIX but nothing that could help me and then I typed in the magical words 'TCP/IP unix pc' and I found Linux. I spent the next few days downloading 17 floppies of Slackware and installed it. It was very exciting. The first thing I noticed was the lack of trumpet winsock, I was so niave. I couldn't figure out how to get on the Internet. I jumped back into Win3.1 and learned how to connect to the Internet. This went well, but I wasn't happy.

Linux worked but I thought there was more. It crashed a couple of times and I was unhappy with the response of the system when I went to untar packages. Then I went back to Aol, started searching for more pc based unix's and the next thing I found was FreeBSD. I went to Walnut Creek and learned all I could about FreeBSD. This was a very exciting time. I decided that I needed to purchase the cd's this time. I then spent many days and nights learning about this operating system. Testing my knowledge, breaking things that I couldn't fix and then trying again. I also starting learning about a lot of opensource software that was available and started to implement them. This was a very knowledgable time for me and put me on my start of a profession that I didn't believe was available for me at that time.

Jump 3 years. I started working for a company that developed software for the lumber industry. I was originally hired to transfer patches and updates to customers sites. I transfered from my job to a R&D one from a comment that I made about Linux, I told them it was terrible. They then asked me what I prefer, I said 'FreeBSD'; this surprised them and after a few days they asked me if I would like to join the team.

Even though the product line ran on AIX and SCO, they were very much into Windows and wanted all of us to be MCSE's. Which we became. Fortunately, our IT department was very much into Linux and open source. Unfortunately, the company wouldn't let us use it. So we slipped it in where we needed it. I was the only FreeBSD user in the department, everyone else favored Linux so I got voted out pretty quickly. I learned a lot things about SCO and AIX during this time and plenty on how windows should work.

The success:

5 years ago I was working for a small school district in Wyoming when I got a job for a major convienent store chain in the inter-mountain west. They just got connected to the Internet and had implemented Linux on the firewall. Since I had a lot of great opportunities to learn the tools needed to create a firewall, I stripped it out in the first week and changed it to FreeBSD 4.3. The company was a big AS400 shop and had started on a couple of projects that would have utilized Windows gateway products to connect to the AS400. I balked at this and said that this could be done with FreeBSD. IBM had released Linux odbc drivers for the AS400 that spring and I had attempted to make it work. Unfortunately I didn't have the patience to get apache/php/AS400 odbc working in Linux emulation so I gave in and installed Red Hat. But I wasn't about to install Linux on the servers that faced

the Internet so I created a proxy that would handle all the requests and send them back to the web servers.

During this time we had implemented all the firewalls, vpn's and web servers on FreeBSD. The CTO had wanted to implement these in a span of year. I had accomplished this in months. The usability of FreeBSD on our network was set. Then two things happened that pushed the usage of FreeBSD from the perimeter to internal processing and both were related to Microsoft.

One of which was running NT4 to handle the extraction of flat files databases. This server would cause us a lot of grief in the middle of processing data. Under heavy loads, the server would stop responding and we would have to reboot. The stores that were connected would have to be notified that they would need to resend. This was time consuming and very interruptive. I then researched perl's capabilities of extracting dbase files and created a solution that allowed the stores to submit information to the FreeBSD server where it would extract the information and submit it to our financial system. This solution was golden: the uptimes of the servers went to months and we didn't have to tell the stores to resubmit. Since the beginning we have turned this scenario into a a very powerful and scalable mechanism for submitting and prepping information. We also incorporated perl's XML support to process Openoffice spreadsheets that get sent through this mechanism. All of this is running on FreeBSD.

We had a typcial Microsoft Domain, PDC, BDC and two Exchange servers. I had started to look into Samba-TNG as a domain controller solution. It was received well but it was viewed as being unnecessary. Then came Microsoft licensing 6.0. We did an audit on our licensing and found that we were short. We decided that we needed to look at alternatives. Our solutions were Samba-TNG, LDAP, Cyrus, IMAP, Postfix and ftp/http server for freebusy schedules—all running on FreeBSD with the services built around LDAP. It was immediately approved. We rolled this out in three weeks with great success.

Since then the success of this has helped in the creation of a very structured and stable network where we have authentication servers, file servers and processing servers, email servers and database servers all running on FreeBSD. We still have Windows servers for Microsoft-only solutions that we can't remove and we also have Linux servers. But even those Windows servers have to authenticate against our FreeBSD servers.

2002 was a difficult year for conveniant stores as people traveled less. We felt it financially as I'm sure other companies had. During this time we were able to increase the services of our network infrastructure at only the cost of hardware. This has been shown as a great value to our company and will continue to have a place on our network.

For all of the network services that we provide, we picked FreeBSD for the security, stability, ease of OS upgrades and the amount of software that can run on it. With the ports tree available it has proven to be an unbeatable solution.

Travis Stevenson is the manager of the network infrastructure at Maverik Country Stores in Afton, Wyoming. Over the last 8 years he has administered UNIX and UNIX-based systems and Windows administration. He currently acts as chief network engineer and security architect and is a die hard FreeBSD user.

Open Source Software in Co-operation Ireland

by Mike Doyle

I am the Network Administrator for a charity that employs roughly 60 staff spread over 4 offices in various parts of Ireland, (Dublin, Belfast, Armagh and Cork) and also has a fundraising office in New York. While all of our end-user computers are PCs running various versions of Microsoft Windows, we make considerable use of Open Source software on our servers.

Our First Experience: email

The first project that introduced Open Source software into Co-operation Ireland was setting up a server to serve email to all staff. We did this towards the end of 1997—prior to this only two members of staff in the entire organisation had email access, and introducing email for all members of staff was the first "major" project I undertook as Network Administrator. I evaluated several commercial mail server applications for Windows NT, and decided that they were all prohibitively expensive given my budget, and our ISP recommended that I look at FreeBSD. So, I installed FreeBSD 2.2.5 on a spare computer that was available (it had a Pentium 90 processor, 32 MB RAM, and 1 GB hard disk space). We used (and still use) Sendmail as the MTA, and Qpopper as the POP3 server software. This enabled me to give each and every member of staff email without spending large amounts of money.

Expanding our use of Open Source Software

In 1998 I added two extra packages to the mail server. The first was Apache, which allowed us to have an internal website. This has been a very useful tool as that allows us to store reference documents that all staff members need access to in a central location.

The second application installed on the server, which in time provoked the eventual upgrade to a server with a larger (2 GB) hard disk, was Squid, a web proxy server package. This allowed me to let all the staff in our Dublin and Belfast offices (which were connected by ISDN routers) to share the outbound Internet connection to view external web pages.

Multi-user Databases over a WAN

Co-operation Ireland had a database that, when I joined the company, was implemented in Microsoft Access, and was replicated between the offices in Dublin and Belfast using scripts to merge changes between the two copies. This was far from an ideal solution, and sometimes the replication procedure lost information.

I was tasked with finding a solution to this problem, and the solution I came up with used some of the software already described (Apache), and two more open source packages: the PHP scripting language and the PostgreSQL database software. Using these tools, I was able to re-implement the database application so that there was a single database server, and all the users connected to it using web-browsers on their desktops. Perl is also used for some coding tasks (for example importing/exporting data to the databases).

File servers for branch offices

Our branch office in Armagh was too small for the budget to supply a file server. For a long time the two or three staff members in Armagh (or Monaghan originally) kept their files on their own hard disks. A couple of years ago, I built a server for this office (again using a PC that was too old for desktop use) using Samba to provide file and print sharing. This server also used one of the two firewalls that comes as part of FreeBSD to allow all the staff in Armagh to share an Internet connection. Once again, all of the software installed on the server in Armagh was installed using the FreeBSD ports system.

Web Calendar

Another project implemented using Open Source software has been the roll-out of a web-calendar for all members of staff. Again, we are using a package called SWebCal to implement, free of charge, a feature that is offered as part of commercial groupware applications. (It is offered, in addition to email, as part of both Lotus Notes and Microsoft Exchange, either of which would cost a large amount of money for an organisation our size.)

The arrival of DSL, and the implementation of a VPN

In 2002, DSL became available in Dublin, and in 2003 it became available in Armagh. We moved from having a leased line between our Dublin and Belfast offices to having a VPN over DSL between these two offices. When we were able to connect the Armagh office using DSL, we were able to extend the VPN to include this site as well.

Once again, the software (Racoon for key exchange and IPSec for the encrypted network connections) was distributed along with FreeBSD. (By this stage, when I rolled out the VPN to Armagh, FreeBSD had reached version 4.7. It is currently on version 4.10).

Again, Co-operation Ireland have been able to use otherwise obsolete equipment and free software to implement a solution that would have cost a large amount of money to build using off-the-shelf commercial tools.

Public Website

Our external website is hosted on a server located at our ISP. This server uses Red Hat Linux (since that is supported by the ISP), PostgreSQL, Perl and PHP to provide a dynamic website that includes a discussion forum for a school project, and an in-house content management system to keep the news and events sections of the website up to date. (The discussion forum software is phpBB).

Factors influencing choice of software

In some cases (e.g. using Samba for file serving) there is simply a choice of the commercial software and one Open Source alternative. In other cases (for example selecting PostgreSQL as the database server) there are other choices available. In these cases ease of installation, quality of on-line documentation, and even the recommendations of other users on the FreeBSD mailing lists have helped me to make up my mind about what to use. I originally selected PostgreSQL over MySQL after evaluating them because the HTML documentation for PostgreSQL was better structured. Even now almost 5 years later, I still feel that the biggest differentiating factor between these two products is the quality of the documentation! My choice of FreeBSD over Linux was originally based on the recommendation of an outside "expert", but I have since used SuSE and RedHat Linux for various tasks and always come back to FreeBSD as my preferred platform given a choice.

As the Network Administrator for a charity, cost has always been a factor influencing my decisions, however the tool (hardware or software) has to actually "get the job done", and to do so without wasting my time, or demanding too much of my users who are not prepared to learn to use Unix themselves. We still use commercial software for many tasks. It would be hard to quantify an exact cost saving that has been made, since certain of the tasks that I use Open Source software to accomplish would have been ignored or cancelled if they could not have been carried out at such a low cost, while others would have been regarded as critical enough that they would have been implemented some other way.

The primary advantage of Open Source software is the cost. The second advantage is that most of this software is documented extremely well, and this documentation is available on-line and easy to find. While the quality of the documentation varies, I have found the on-line documentation for FreeBSD itself, and for PHP and PostgreSQL to be of exceptional quality. Samba, Apache and Perl all have very good online resources to help an administrator to accomplish any necessary task, but I personally found navigating these documents to be a little slower. To be fair though, much commercial software comes with downright unhelpful online help, and so in every case where I am using Open Source software, the quality of the documentation has also been a bonus.

Another added factor for selecting Open Source software has been the speed and quality of responses to requests for help posted to the various mailing lists. (Commercial software usually requires expensive maintenance contracts to be taken out for technical support.) Michael Doyle graduated from University College Dublin with a B.Sc. in computer science in 1993. For the past seven years, he has been the Network Administrator for Co-operation Ireland, a charity involved in the peace-process in Northern Ireland. He lives in Dublin, Ireland with his wife and infant daughter. In his spare time (not that he has much these days) he plays the guitar and sometimes writes music.

BSD Resources

The FreeBSD Project
freebsd.org

The OpenBSD Project
openbsd.org

The NetBSD Project
netbsd.org

BSD DevCenter.
onlamp.com/bsd

DaemonNews
daemonnews.org

BSD Newsletter
bsdnewsletter.com

Putting BSD to work for you
bsdatwork.com

BSD Resources
bsd.reedmedia.net

BSD Forum
bsdvault.net