FreeBSD NEWS

ISSUE 2 SUMMER 1998

This publication may be freely reproduced in whole or in part

FreeBSD №1 for ISPs

by Mathew Dillon BEST Internet Communications, Inc.

BEST INTERNET COMMUNICATIONS

For the typical modern ISP, the clock probably started running a bit over two years ago. Sure, there are plenty of older ISPs, but few of them have had to contend with 100% per month growth and many of the larger ones also started more as BBS systems then ISPs. If you were like us, however, you started right as the Internet really started to become popular and that meant having to subsequently deal with the strain of very rapid growth.

For us, two years ago was indeed the beginning. We had made an educated guess as to our best market, hit it running, and turned out to be right. With that out of the way, the name of the game for BEST Internet was to simply accomplish the engineering and programming feats required to support a rapidly exploding user base.

In the very beginning, we used BSDI and FreeBSD and, if the times had been less trying, we wouldn't have made the almost fatal mistake of shifting to larger multi-processor minis from a brand-name UNIX vendor who shall remain nameless. We somehow convinced ourselves that moving to larger commercially-supported boxes would both help sell our service and provide adequate margins for growth.

That mistake almost drove us out of business. The scalability never materialized and the work required to maintain the minis went from difficult to near impossible. Every hardware addition required a ridiculous amount of work and money to install. We couldn't get bug fixes for weeks or months. There were all sorts of odd licensing fees that made it impossible to ““genericize”” machine installations. I will spare you the rest of the gory details – it wasn't a pretty scene.

It was a nightmare made even worse by the fact that so many resources were concentrated in the minis that a crash, software problem, internet attack, or user mistake could effectively take down our entire service.

This situation convinced the entire BEST Engineering department that minis, the so-called "enterprise servers", are a huge mistake. To make a mini cost effective would require decking each one out with half a million dollars worth of hardware, and then actually to use it would require putting a lot of marbles in one bag, so to speak.

In moving back to running FreeBSD on rack-mounted Pentium Pro 200's, there were some issues that concerned us and some that did not. We knew that the reliability was there, especially with the PPro 200s and their integrated ECC and secondary caches. We had also learned that the governing limitation to all ISP support boxes is "disk I/O", "disk I/O", and "disk I/O." (Short of putting ten SCSI interfaces in each of the two mini computers, the ten FreeBSD boxes that replaced them would win hands down). We also knew that CPU power would not be a factor and, indeed, it hasn't been – the new machines usually hit physical disk I/O limits long before the CPU saturates. Finally, we knew that we could balance out and tune "CPU vs. disk I/O" by adding memory since the PPro motherboards could accommodate up to 512MB of RAM where we expected to need only 192MB in the absolute worst case. Most shell boxes run with 128MB just fine, supporting up to 2000 user accounts on each.

What we worried about was that the maintenance required to support a larger number of smaller boxes would increase. In fact, the time required to maintain 24 FreeBSD boxes has occupied less than half the time we spent dealing with the two minis and plethora of smaller vendor boxes, something which was

CONTINUED...

In This Issue:

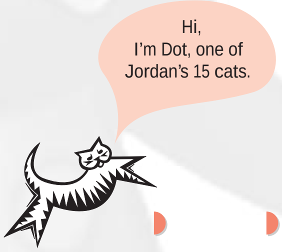
made possible by the fact that every single FreeBSD box is configured and updated from a template machine.That's also something we just couldn't do with the vendor boxes due to the massive customization each one required and odd licenses which required us to be frugal about installations.

FreeBSD has also done an excellent job with service-specific boxes such as our "newsfeeds box," our "newsreader boxes," our "sendmail back-end boxes," and other various administra-tive-duty machines. Since the single largest cost for an ISP-capable rack-mount PC is usually the disk drives, most of the non-news boxes are incredibly cheap because they require no real disk storage.A shell machine will have a 4G and two 9G disks to start out with, whereas a proxy box generally requires only a single 4G disk.The ability to flexibly tailor PC hardware configurations to specific tasks can lead to some incredible cost savings without impacting maintainability.

Ultimately, the KISS principle applies to administration. No (major) NFS mounts, no attempt to allow users to log into just any shell machine – we assign them one and that is where their account resides. No attempt to slave news is made since each newsreader box is self contained – users pick one and don't switch.We were worried that the load would not balance well. In fact, it balances extremely well. Sure, some shell machines are loaded a bit more than others, but if it isn't noticeable it isn't a problem. It is, perhaps, a somewhat different approach to the concept of a "distributed system", but it works, nonetheless, and works very well.

Where we used to get a cascade of complaints when some part of the system would wedge, now we get only a few.Where we used to have a difficult time working out the logistics of when to purchase a new machine or when to add a disk, now it becomes trivial because the user base on each machine builds, matures, and that's it – we go onto the next machine.We stay one or two boxes ahead of the growth curve and everything runs smooth as glass.

I think it would be fair to say that FreeBSD not only saved us from past mistakes, but now empowers us to continue building our business at whatever rate we choose.



What is FreeBSD?

by Jordan K. Hubbard Editor

FreeBSD is a UNIX®-compatible operating system based on 4.4 BSD Lite2 from the University of California, Berkeley and its contributors. BSD has been in more or less constant development for over 15 years.The same technologies used in FreeBSD have also been the basis for many commercial operating systems, from Digital's Ultrix to Sun's SunOS, and still remains a staple of commercial OS development today.

Like its commercial counterparts, FreeBSD also seeks to make the task of using and installing the system as simple as possible, going on the basic principle that no advanced technology is worth much if those who wish to use it are presented with significant difficulties in doing so. Since cost is also a barrier to some, we've made sure that the system is always available free of charge on the Internet (ftp://ftp.freebsd.org/pub/FreeBSD) and we welcome both free and commercial redistribution of the technology.

FreeBSD can be used for everything from creating web servers, providing enterprise mail services, file and printer sharing (for other UNIX systems as well as the Windows and Macintosh platforms) and serious software development. FreeBSD also provides binary emulation for other systems such as BSD/OS, Linux and SCO, allowing applications available for those systems to run on FreeBSD and providing the widest possible selection of applications to the user.

If you're looking for a stable, easy-to-install operating system with a long and proven track record, FreeBSD is the OS for you!

Who is the FreeBSD Project?

The FreeBSD Project is a group of over 100 developers led by a 17 member "core team" which sets overall direction and ensures a consistent project focus. Most of the developers come from commercial software development and BSD UNIX backgrounds, providing a unique perspective on what can and should be done with this technology in the marketplace. Our long history of software development experience has also paid dividends in the way that the project is organized, with multi-track development (stable and experimental), strong use of source code control (CVS) and a highly centralized development model (not just the kernel, but the entire system) being just some of the results of our experience.

Where do I get more information?

The project runs a web server at **http://www. freebsd.org** which contains a lot of useful information (also mirrored in over 25 countries—please see **http://www.<countrycode>.freebsd.org** for the nearest local server) and you're highly encouraged to visit it.

To reach the FreeBSD development team, please send email to hackers@freebsd.org. If you have a matter which requires the specific attentions of the core team (a commercial request or some other proposed cooperative venture), please send email to **core@freebsd.org**.

So, in short...

The marketing hype may tell you that NT can do everything a well-configured UNIX machine can do, but we know differently from practical experience and if we didn't feel this was the best possible technology for the needs of Internet Service Providers and other commercial/private interests, we'd be using something else ourselves! As it is, thousands of commercial users, including such notables as Yahoo! and Walnut Creek CDROM (who run the biggest general FTP server in the world on FreeBSD), are happily running FreeBSD and we hope that you will choose to do so too!

The FreeBSD Project is pleased to announce the public FreeBSD beta test of Blender, a full-feature 3D design, rendering and animation package from NeoGeo b.v. NeoGeo (http://www.neogeo.nl) is the Netherlands' largest 3D animation studio, and has developed Blender in-house to meet their demanding requirements.

Blender allows the rapid prototyping and design of breathtaking 3D visuals and complex animations, and handles all phases of the design and production process.

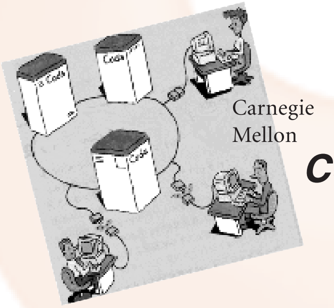
"The release of Blender brings the power of professional 3D animation design software to the PC in a feature-packed and very affordable form." (Ton Roosendaal, NeoGeo/Enji)

See http://www.neogeo.nl/blender.html for more information, news and to download Blender today!

Blender
NeoGeo GmbH

Coda on FreeBSD

by M. Satyanarayanan School of Computer Science, Carnegie Mellon University



Coda File System

We are pleased to inform the FreeBSD community that the Coda File System is now available for FreeBSD 2.2.5 and 2.2.6. Our hope is that the free availability of this state-of-the-art distributed file system in source code form will attract serious users and developers whose talents will enhance the system along many dimensions. Since Coda also runs on NetBSD and Linux, we hope that over time it will become the distributed file system of choice for free UNIX platforms.

Coda is a distributed file system developed at Carnegie Mellon University as a test bed for research in replication and mobile computing. As a descendant of the Andrew File System (AFS), it inherits many of the key features that make AFS attractive in large-scale distributed environments. These include: a client-server model with trusted servers and untrusted clients; location transparency, so that server identities are completely hidden; aggressive on-disk whole-file client caching, with a further level of block-level caching in the buffer cache; organization of data into logical volumes for purposes of easy system administration; an access-list model of protection; and a user-level implementation strategy that keeps the entire server and all but a tiny fraction of the client out of the kernel.

In addition, Coda provides many unique capabilities for next-generation wired and wireless environments. These include: support for voluntary and involuntary disconnected operation (including hoarding, reintegration, and conflict-detection); support for weakly-connected operation with dynamic adaptation to network quality ranging from modem speeds to LAN speeds; and transparent optimistic read-write server replication of data, with support for directory resolution and application-specific conflict resolution. Planned enhancements to Coda include write-back caching for improved performance while strongly-connected, and integration with Kerberos authentication. The disconnected operation capability of Coda has served as the inspiration for the IntelliMirror component of Microsoft's forthcoming NT 5.0 file system.

We invite the FreeBSD community to join us in transforming Coda from the experimental prototype that it is now to a high-quality, well-engineered system for the free UNIX community. The person leading this effort is Peter Braam (braam@cs.cmu.edu), a senior researcher in the Coda project at Carnegie Mellon. Other people currently involved in this work at Carnegie Mellon include Bob Baron, Henry Pierce, Robert Watson, and Jan Harkes. The full cast of contributors to Coda over its multi-year evolution is too long to list here.

For further details, visit the Coda Web page at www.coda.cs.cmu.edu. To obtain research papers and Ph.D. theses generated by Coda, follow the "papers" link. To learn more about the broader research context in which Coda is being developed, follow the "Coda project" link.

Another FreeBSD Success Story: An ISP Revolutionized

by Dan Benjamin Network Analyst, Intel UNIX Systems Manufacturer

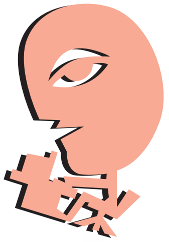
As a Network Analyst and an Intel UNIX Systems Manufacturer, being able to offer my clients a wide variety of options is always a priority. Most of my clients are companies who need servers, workstations and support or consulting work like network analysis, setup, and administration. Few companies I deal with have an unlimited budget – on the contrary, one of the most challenging aspects of this business is stretching the dollar. In many cases, my clients incorrectly believe that they need to spend a inordinate amount of money to purchase a high-end server. A FreeBSD server built with fault-tolerant components has, in my experience, always proved to be a excellent, cost-cutting solution to this problem.

Real Life Example: A local ISP and Web presence provider was growing rapidly. They were adding clients regularly and new businesses were signing up on an almost daily basis. They had recently upgraded their bandwidth and were renting larger office space in a better location. However, they quickly found that their servers, primarily three overloaded Macintoshes, just couldn't cut it, even with memory and processor upgrades. The client was concerned that the price tag for a new, high-performance server to handle their DNS, Web access, FTP, and E-mail services would be too high. Despite the new business, they did not have an unlimited budget but also could not afford to be without a fast machine that could deliver rock-stable performance. FreeBSD was the answer.

"A FreeBSD server built with fault-tolerant components has, in my experience, always proved to be a excellent, cost-cutting solution..."

This company was a Macintosh shop from the ground up, save for one PC running Windows 95. They did (and still do) all of their graphic design, web, and paper advertising for themselves and for clients using Macintosh computers. With this in mind, I knew that convincing them to switch server platforms could be a difficult task. In past situations, moving a client site from an aging SunOS, NT, or Novell network to a FreeBSD managed one was a relatively straightforward task. In many cases it was a simple matter of the client adjusting to a slightly different flavor of UNIX. In others, the client might opt for remote network management or hire a full-time UNIX-savvy administrator. In any case, the transition was usually straight forward and the performance improvements could be easily measured – the switch invariably meant a jump in productivity and a decrease in operating and upgrade costs. Integrating a FreeBSD system into a pre-existing network is easy when you take advantage of the numerous applications available from the FreeBSD Ports collection.

But convincing this all-Macintosh shop to switch hardware platforms and operating systems in one motion wouldn't be easy. This client would literally need to shut down their dependable (albeit slow)



Macintosh servers and put their faith in an architecture that had slowly eroded the Mac's place in American computing. A difficult task.

Fortunately, establishing FreeBSD's rock-solid stability proved to be an easier than expected task: As this client conducted most of its business on the Internet, they had all used Yahoo extensively – both to research and locate information, as well as to publicize and advertise for themselves and their clients. What they didn't know was that Yahoo used FreeBSD almost exclusively (FreeBSD News, Issue 1). Once they learned that FreeBSD also powered ftp.cdrom.com, their decision to switch became even more clear. In addition, the company president, also a Macintosh user and fan, was actually in favor of FreeBSD running their on-line business. Because of his background with BSD UNIX networking, he knew it would improve speed and reliability.

But what about the people "in the trenches", the designers and developers who would need to access and update the Web sites on a daily, even hourly basis. How would a FreeBSD server change their world? There were several legitimate concerns, first and foremost, how would the developers interface with the new server? Using ftp was not an option – that method was cryptic and went against the grain of the Mac user's mind set. Drag and drop, double click, theirs is the world of the original GUI.

Fortunately, Research Systems UNIX Group (RSUG) at the University of Michigan has developed Netatalk, which provides access to the UNIX file system for Macintosh and other systems with AppleShare client software. Simply put, Netatalk is a file and print server for AppleTalk networks and is available for FreeBSD from the Ports collection. Installing software from a FreeBSD Port is usually as simple as downloading a tar file (or, if you already have the ports collection loaded, simply going to the correct directory) and typing "make install". A few changes to some configuration files, and a quick restart of inetd, and the Macs were again dragging and dropping to their hearts content. I used Samba, a networking package created by Andrew Tridgell – also available as a FreeBSD port, to provide connectivity to the Windows 95 machine.

netatalk
the AppleTalk Protocol Suite for Unix

After transitioning DNS, the next and most important step was transitioning the web sites. At that time, the server was a Pentium 160 with 32MB memory and several SCSI drives. It ran FreeBSD-Release 2.1.7 and an early Apache Web server. The performance improvements were impressive, to say the least, and it wasn't long before all of the sites were moved. Mail and FTP services quickly followed, and eventually the single FreeBSD server had completely replaced several Macintosh servers.

The introduction of the server brought a higher level of security in addition to performance improvements. I'd taken several measures with "sendmail" to prevent spamming and relaying – something that had plagued them in the past. I'd also installed the TCP Wrappers port which helps prevent unauthorized access to well-known services, and several other security tweaks – all for the cost of a pre-configured FreeBSD server.

Since then, I've installed several more servers and replaced the initial server with a Pentium II running FreeBSD 2.2.5-stable (pre-2.2.6). Internally, their network has never been faster but their clients – to whom the upgrades have meant the most – are quite happy with the improvements in speed and the lack of downtime. Most recently, we added an NT server to handle active server pages. It coexists with FreeBSD rather well, sharing with it the dynamic content of a backend SQL server.

And this is just one example. Most of the server solutions I provide to my clients are running FreeBSD, all with excellent results. Another recent project involved the placement of a dual-homed, mirrored and security-hardened FreeBSD mail server to filter spam and deny relaying. Beneath their firewall sit several more FreeBSD servers to handle internal pop mail and file serving across the WAN.

But lest we forget, FreeBSD isn't only a server. The FreeBSD X Workstation I'm sitting in front of now allows me to run StarWriter 3, with it's MS Word-like interface, to compose this article. I manage my business finances and client lists with databases that run on FreeBSD, and we even do graphic design (with the Gimp port) and Java (with the JDK 1.1.5 port), all using FreeBSD.

Overall, although I use it on a daily basis within my own business, FreeBSD has had it's greatest impact on my clients in the hardware and consulting business. I can provide them with reliable servers and workstations that guarantee the performance and reliability my clients require. And because FreeBSD is free, I can pass that as a savings on to my customers.

Getting Kids on the Web with TCJA

by Wilko Bulte <wilko@tcja.nl>

What is TCJA? TCJA is the Technisch Creatief Jeugdcentrum Arnhem in Arnhem, The Netherlands. TCJA loosely translates to "Technical and Creative Youth Center."

TCJA is a foundation, in existence for over 25 years, that has the goal of popularizing scientific and technical subjects for kids from 10-20 years of age. Emphasis is put on getting women into TCJA's activities since, in the Netherlands, women in technical or scientific disciplines are a very rare phenomenon. In fact, the influx of students in almost all technical studies has been declining over the past few years.

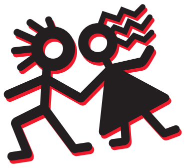
We try to achieve our goals in ways that kids appreciate – not putting them into class rooms but instead giving them the opportunity to obtain hands-on experience with the subjects at hand. The variety of subjects include chemistry, photography, electronics, silk screen printing, wood and metal working, and last but not least computer related fields.

In order to make all this possible, TCJA has their own building with a small paid and a much bigger unpaid volunteer staff. Money and equipment is donated by the government, companies and individuals.

Computers at TCJA

As we've already mentioned, a lot of equipment is donated and our computing infrastructure is no exception to this rule. We run a thinwire ethernet network with Cabletron hubs that connects all systems to a single LAN. The network involves six Sun SparcStation IPCs, a Sun SparcServer 4/470, nine 486DXs and one 386DX-based PC machine. The Sun machines were a gift from Philips Electronics and Sun Netherlands, the PCs were partially bought and partially donated by Digital Equipment. We even have Sun equipment that was donated by a generous guy in California (thanks Gary!)

In this network five tabletop 486DX2/66 boxes run MS Windows for Workgroups and Netscape Navigator for Web access, while a 486SX33 runs our administration (also with WfW). A 486DX/33 is run with MS DOS to allow computer-aided printed circuit design.



The remaining two PC machines are more interesting - they both run FreeBSD 2.2.1-Release. Both boxes are 486DX/33 EISA-based Philips servers. One of them is dedicated to NFS, serving both the Sun boxes and the PCs (using Samba with WfW). The

CONTINUED...

Sun SparcServer 4/470 is also run as an NFS server and also handles the diskless booting of the Sun IPCs. All Suns run SunOS.

The other PC is the more interesting one: it is our gateway to the Internet. It has been dubbed “dontpanic.tcja.nl”, a name taken from the cover of the Hitchhiker’s Guide to the Galaxy (although we admit we christened it this way in hopes that it would live up to it’s name and so far it has done so brilliantly).

Dontpanic’s connection has recently been upgraded from a 28.8K modem to an ISDN/PPP link using a Teles 16.3 interface board. Since all our Suns and PCs tend to be run at the same time, especially during weekends, we run a caching Apache proxy server on it to limit actual ISDN/PPP link traffic as much as possible. The ISDN link uses the bisdn/ppp patch package which we adapted slightly for our specific needs.

To protect our network from the evil outside world (and probably also to keep our adventurous kids from hacking the Internet too much ;-) dontpanic runs a customized version of the TIS firewall tool kit in addition to a set of ipfw filters using FreeBSD’s kernel-based firewall. Apart from its networking duties, “dontpanic” also serves as our local mailhub (POP for the PC Eudora or Netscape clients, NFS mail spool for the Suns), a local Apache Web server (more on our Web stuff in a moment), a local DNS server and as the printhost for an Apple LaserWriter postscript laser printer (again with Samba for the WfW clients, and lpd for the Sun/FreeBSD machines). It’s perhaps not entirely recommended to have all these functions running on the firewall box, but we have to use what we have in this case.

We are currently in the process of migrating all our Sun disk storage to SCSI disks, the Sun server used to have six 900MB SMD disks and one 1.3GB IPI disk. For reasons of power consumption and heat production, donated SCSI disks are taking their place.

Our Web Presence

An organization like ours of course must also have it’s own Internet domain registration and a Web server. Thanks to some kind souls, a 386DX/40 with 8MB and a 300MB ESDI disk now exists as www.tcja.nl. It is also of course running FreeBSD (2.1.0R in this case). Because it is 20 miles away we tend to leave it alone as much as possible. So far, only a single unscheduled intervention has been needed after a machine failed to reboot properly after a power failure.

Although the previous hardware description probably sounds like a sorry excuse for a web server we have (knock on wood) impressive uptimes to report. As of today it has an uptime of 112 days, and that is only because it had to be powercycled to add a UPS. Uptime before the UPS was also 100+ days.

It is well connected to NLnet, being almost on the NLnet backbone at the Nijmegen University, so please feel free to take a look at http://www.tcja.nl. It is mostly in Dutch, but an English abstract has been provided and some photographs help to get the idea across.

If you are interested in the quality of the power grid in Nijmegen, you can ask our intelligent UPS by checking http://www.tcja.nl/ups. A Java applet will draw you diagrams of the previous 7 days.

Apart from serving our web pages with Apache, the machine also handles our email traffic. All mail to tcja.nl is UUCP-queued at www.tcja.nl and grabbed from there by dontpanic.tcja.nl whenever the ISDN/PPP link is up. This may sound a bit strange to US-based people, but there is no such thing as a free permanently connected phone/ISDN line here in the Netherlands. You always pay a metered fee, so queuing is the answer if you don’t want to go bankrupt.

Our Experiences with FreeBSD

So, what do we think of FreeBSD? You’ve probably guessed it already. We are more than just a bit happy with the performance and reliability of FreeBSD. We started in the dark ages with 386BSD 0.1 + patchkit, and have never regretted using BSD technology.

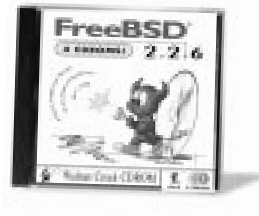
In the course of events a considerable number of the people who’ve seen our setup have also converted to FreeBSD. This included a number of Linux adepts, but also WinNT users who are currently contemplating the move. You could say we act as a technical evangelist group who can show a proven track record in the use of FreeBSD.

FreeBSD has proven to be very stable, adaptable to local needs (thanks to the availability of source code) and economical. Even with our sometimes questionable and limited hardware resources, the reliability of the total system has proven to be better than commercial operating system offerings.

Any questions about this article, general remarks, or even a proposed donation of equipment or money to the TCJA can be sent to:

Wilko Bulte (wilko@tcja.nl)
Rene de Vries (rene@tcja.nl)
http://www.tcja.nl
Stichting Technisch Creatief Jeugdcentrum Arnhem
Arnhem, The Netherlands

FreeBSD 2.2.6



FreeBSD 2.2.6 is a full, professional quality, UNIX-compatible operating system. Simple CDROM installation with El Torito direct-boot support. Easy switching between multiple operating systems. This **four CDROM set** also includes more than **2 GB** of ports, packages and applications. If you subscribe to our quarterly updates, you will always have the latest edition at a substantial savings.

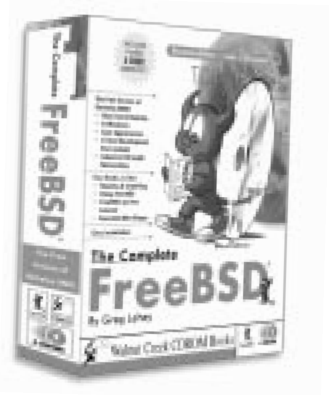
. . . . **$39.95**
. . . . .subs **$24.95**

FreeBSD Snapshot 3.0

The ***FreeBSD Snapshot*** CDROM contains the current development release, with bug fixes and enhancements that aren't quite ready for the FreeBSD production release. We release the snapshots every few months. If you would like to keep track of FreeBSD's latest developments, you can save money by subscribing to this disc.

. . . .**24.95**
. . . . .subs **$14.95**

The Complete FreeBSD, 2nd Ed.



This FreeBSD Book and CDROM set is a recently updated and expanded book containing over **1700 pages** of expert information on installing and running FreeBSD. The book comes with **four CDROMs** that contain the complete FreeBSD OS and over **2 GB** of utilities, applications, and on-line documents.

From first-time installation to day-to-day administration of FreeBSD, this book provides an unmatched reference complete with a thorough index. This book will guide you through setting up FreeBSD cooperatively with other OSes, installing the system files, and setting up users and Internet FTP/WWW services. There is a *Quick Installation Guide* to get you started and much more detailed instruction if you want to customize your system. The guide also simplifies X Windows configuration and demystifies configuring a kernel for your own hardware. It explains how to set up *Samba* and *Netatalk* to provide reliable and high-performance file and printer service to Windows and Macintosh clients. We also include tutorials on system administration so you can become a FreeBSD guru in no time! As a comprehensive reference on FreeBSD, this book is absolutely indispensable!

. . . .**$69.95**
. . . . .book w/o CDROMs **$29.95**

FreeBSD T-shirts

These Hanes "Beefy T" t-shirts are 100% cotton, machine washable, and made in the U.S.A. All t-shirts are white and come in three sizes: medium, large, and extra large. T-shirts can be purchased for $16.00 plus shipping. You can also see them on our web-site at: *http://www.cdrom.com/titles/os/fbsdtsh.htm*

FreeBSD Polo shirts

These high quality polo-style shirts come from Lands End. They are 100% combed cotton interlochen, machine washable, and made in the U.S.A. Polo shirts come in three sizes: **medium**, **large**, and **extra large**. You can also choose between three colors: **white**, **black**, and **light blue**. Polo shirts can be purchased for $45.00 plus shipping. You can also see them on our web-site at: *http://www.cdrom.com/titles/os/fbsdpolo.htm*

When you order

Walnut Creek CDROM - http://www.cdrom.com/

1-800-786-9907 or 1-925-674-0783 • fax: +1-925-674-0821 • orders@cdrom.com

Phone Orders

To speed your phone order, please have our credit card at hand. You may want to fill out the order form in advance.

Standard 24-hour Shipping

We ship 98% of our orders within 24 hours of receipt. In North America (United States, Canada, Mexico) we charge a standard shipping fee of $5 per order for First Class Mail delivery. For international customer, we charge a $9 fee per order for AirMail delivery. U.S. customers should receive their phone orders within 3-5 business days.

Note: If an item is out of stock, our sales representatives will inform you of the expected delay. Items ordered together may be shipped separately for no additional charge.

Express Delivery

We offer UPS Express Delivery for all North American orders received before 3 p.m. Pacific Standard Time.

$10 per order UPS Blue • 2-day service
$15 per order UPS Red • Next-day service

We offer various express services for our international customers. Please call for details.

Returns and Exchanges

If you are dissatisfied with a product, we will gladly replace the item or refund your money. Simply return the product with a short note to:

Walnut Creek CDROM
Attn: Returns
4041 Pike Lane, Ste F-933
Concord, CA 94520

100% Guarantee!

When you order

Walnut Creek CDROM - http://www.cdrom.com/

1-800-786-9907 or 1-925-674-0783 • fax: +1-925-674-0821 • orders@cdrom.com

1 Order in four easy steps:

PLEASE FILL OUT COMPLETELY

First Name   Last Name

Address

City

State/Zip/Country

Phone Number – with Area & Country Code (in case we have questions about your order)

2 I'd like to order:

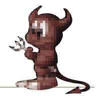
QUANTITY	PRODUCT NAME	COST EACH	TOTAL
		SUB TOTAL	
		Add sales tax, CA resident only 8.25%	
		Shipping PER ORDER $5 US/Can/Mex $9 Int'l Overseas	
		GRAND TOTAL	

3 Payment Method:

○ Please charge my credit card

○ Check enclosed (United States currency)

○ VISA ○ ○ ○ ○



Account #

Name on Card   Expiration Date

Cardholder Signature

4 Keycode:

Please enter the keycode located near your address on the self-mail portion of this newsletter.

Enter Keycode:

Enter YOUR code on this line.

EXAMPLE: John Doe
Your Company
P.O. Box 1234
Anytown, US 12345

Keycode: 345678

933

Call Walnut Creek CDROM 1-800-786-9907

4041 Pike Lane, Suite F-933, Concord, CA 94520 U.S.A.

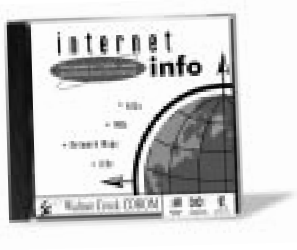
ORDER... by phone...+1-925-674-0783
by fax...+1-925-674-0821
by email...orders@cdrom.com

CUT HERE AND RETURN ORDER FORM IN ATTACHED REPLY ENVELOPE

WebMaster's Toolkit CDROM

A comprehensive collection of documentation, utilities, and graphics that can help you create, maintain and improve your HTML pages. You get HTML tutorials, editors, conversion utilities, FAQs, templates, images, Perl CGI source code, applets and ActiveX controls.

. . . .$39.95



Internet Info CDROM

This latest update gives you over 13,000 documents detailing all aspects of the Internet, including: HTML tutorials, WWW guides, Internet FAQs, RFCs, IENs, Network Maps, and much more. This disc suits all users, from beginners to the seasoned cyber addict.

. . . .$39.95

Unix Freeware CDROM

This disc contains 638 MB of source code for over 700 different packages. There are programs in almost every category. These programs compile under FreeBSD, BSD/OS, Linux, SunOS, Solaris, etc.

. . . .$39.95

Usenet Source Code, Vol 1,2,3

This three volume set is a comprehensive collection of source code from Usenet. It includes alt.sources, comp.sources.misc, games, unix, postscript, x, sun, and more. Volume 1 has files from 1985-1990. Volume 2 has files from 1991-1993. Volume 3 has files from 1993-1996

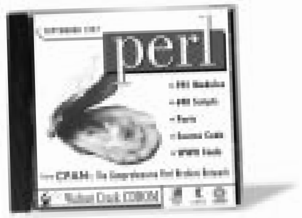
. . . .$39.95, OR

. . . . .all three for $59.95

Python Toolkit CDROM

Python is an interpreted and object-oriented programming language with powerful features for networking, string handling, user interfaces, etc. This disc gives you thousands of sample programs, libraries, packages, tutorials, "how-to" documents, and FAQs. We've included interpreters for Win95/NT and most Unix variants.

. . . .$39.95



Perl CPAN CDROM

This is based on the CPAN archive. The material is as up-to-date as possible, uncompressed, and organized to make finding files easy. You get the latest Perl source code, and binary versions for Solaris, SunOS, Linux, FreeBSD, WinNT/95, Mac, OS/2, DOS and Amiga.

. . . .$39.95

FreeBSD And the PalmPilot

by Oliver Fromme <olli@fromme.com>

So, you've purchased this cute little PDA, played around a bit with the built-in applications, entered a few memos, dates and addresses, and maybe even wrote a reply to 3Com's "Welcome" email. And now you wonder: what's next? You've seen all those neat applications and games on the Internet, how do you get them into your PalmPilot? You probably also want to backup its data onto your desktop PC but the tools supplied with the pilot only support Windows and Macs... what to do?

Fortunately, there are several tools available for making the PalmPilot live happily with a UNIX system, and that includes, of course, FreeBSD. This article can't replace an in-depth tutorial, but it should get you started. It also assumes that you are already familiar with the basic functions and features of your PalmPilot as well as with FreeBSD. Note that I use the word "PalmPilot" throughout this article, but everything also applies to the older Pilot models, IBM's Workpad (which is basically an OEM version of the PalmPilot Professional) and to the new Palm III.

Note: Most tools access the PalmPilot via the /dev/pilot device, which should be a symlink to the serial port to which your Pilot's cradle is connected. For example, assuming that you've connected the cradle to the second (COM2) serial port, you'd create a symlink to point /dev/pilot at /dev/cuaa1 as follows:

```
ln -fs /dev/cuaa1 /dev/pilot
```

You will also need read and write permission to the serial port. If you do not wish to run these utilities as root, one way of doing it is to add yourself to the "dialer" group (edit /etc/group) and make sure that the serial device is group-owned by dialer.

The first tool that we'll discuss is "kpilot". As the name suggests, kpilot is a graphical application for the KDE environment. It is available from the FreeBSD ports collection in the /usr/ports/comms/kpilot directory. (If you're not familiar with the ports collection, you should have a look at the appropriate section of the FreeBSD Handbook.)

The main advantage of kpilot is its easy-to-use graphical interface. With a click of your mouse, you can convert text files from your PC to memos on your PalmPilot and vice versa. It also allows you to import and export address book entries and exchange email messages with your mail server. Kpilot supports the POP3 protocol to retrieve mail from the mail server, then transferring it to the mail application on your PalmPilot. It also uses sendmail to deliver any outgoing mail messages that you have queued up on your PalmPilot. You can also use kpilot to backup and hot-sync your data, and there is even a little daemon program which automatically initiates the hot-sync process whenever you press the hot-sync button on the cradle. Like most KDE applications, kpilot also provides online-help in HTML format (though, unfortunately, there are no manual pages for those more traditionally inclined).

Note also that you don't need KDE to run kpilot – you actually only need the KDE libraries (which are a separate FreeBSD port). Without the full KDE environment, however, you won't be able to use some of kpilot's features which makes KDE worth installing (from the ports collection) if you have the disk space. This will allow the full online help, automatic hot-sync, and status display to work (these require the kdehelp tool, KFM and kpanel). In any case, you can read the online help with your favorite HTML browser – it is installed in /usr/local/share/doc/HTML/en/kpilot/.

If you don't like GUIs in general or KDE in particular, you should have a look at the "pilot-link" port (/usr/ports/comms/pilot-link). This is a set of command-line driven tools for talking to the PalmPilot and also includes various libraries and modules, so you can easily write your own programs (in C, perl, tcl/tk etc.) to access the PalmPilot if you are a programmer.

The single most useful tool from the pilot-link port is the "pilot-xfer" program. It supports a number of command line options, the most important being -b (backup your PalmPilot's data to a given directory, i.e. do a one-way-sync), -s (sync with the contents of a directory), and -i (install a file on your PalmPilot). Let's say, for example, that you have downloaded a game "foobar" from the Internet. To install it on your PalmPilot with pilot-xfer, you'd simply type:

```
pilot-xfer -i foobar.prc
```

Note that programs for the PalmPilot have the filename extension .prc (pilot runnable code), while other data files have the extension.pdb (pilot database). Since PalmPilot programs can also be databases, however, this distinction isn't always adhered to.

Note: If you use the pilot-link port, one important thing that is missing from the documentation is the PILOTRATE environment variable which is used to select the speed of the serial port. The default of 9600 is slow and drains your batteries. I recommended that this be set to 57600 (the maximum).

Finally, if you're a software developer, there are several tools available which help you to develop PalmPilot applications under FreeBSD. Describing all of them would be beyond the scope of this article, so I'll just mention a few and leave the rest as an exercise to the reader. First off, there is the "prc-tools" port (/usr/ports/devel/prc-tools), a complete GCC cross-compilation environment and linker which lets you write applications in C or C++ and compile them straight to .prc format under FreeBSD. If you prefer to program your pilot beast in assembler, then "pila" – the Pilot assembler,

CONTINUED...

is for you. Another interesting utility is "jump", a Java bytecode converter. It converts Java class files to assembler source files which can then be assembled with pila. It produces surprisingly compact and fairly fast executables, but it doesn't support the complete Java language (e.g. no threads). Both utilities, along with many others, are described and available for download at http://www.palmpilotfiles.com/Development.html

Last, but not least, the xcopilot port (/usr/ports/emulators/xcopilot) should be installed on every PalmPilot programmer's PC (and even if you're not a programmer, you might find it useful). Xcopilot is a complete PalmPilot emulator for X11 and needs only a PalmOS ROM image to completely simulate a PalmPilot. If you installed the pilot-link port just mentioned, you can use its "pi-getrom" utility to transfer your own PalmPilot's ROM to a file on your desktop PC where it can be used by xcopilot. All of this allows you to test new software in a safe, emulated environment without risking the data on your trusty pilot.

The PalmPilot is a surprisingly powerful little machine in a compact size and, with FreeBSD's help, is quite easy to both use and develop new software for. Kudos to the PalmPilot engineers for a fine product and to the many Unix developers out there who worked hard to provide us with such a rich selection of utilities for it!

References:
- http://www.palmpilot.com/
3Com's PalmPilot and Palm III pages.
- http://www.freebsd.org/
The FreeBSD homepage.
- http://ftp.tu-clausthal.de/pub/palmpilot
Software and documents for PalmPilot and UNIX users.

* * *

How FreeBSD Became the Favorite OS on my Desktop

by Martin Cracauer <cracauer@cons.org>

I'm working for BIK – Aschpurwis + Behrens, a German market- and media-research institute (www.bik-gmbh.de). BIK used a wide variety of UNIX derivatives including SCO, NeXTStep, SunOS, Solaris, NetBSD, several PC-based SVR4s and, of course, FreeBSD.

The story how FreeBSD became the favorite OS on my desktop at work is somewhat boring and largely a matter of personal preference. FreeBSD just felt best for me and I had the fewest problems from all the UNIX derivatives I tried.

Somewhat less boring is how FreeBSD became the dominant computing platform for the company's multiuser and server machines.

Since 1990, we developed a batch-oriented tabulation generator with a lot of hooks for adding functionality. It became a pretty decent product, though it remained hard to use for GUI-oriented users, so only a few outside customers used it directly. It ran on all the platforms listed above.

When the World Wide Web took off, we suddenly found ourselves in a much better position than those competitors who developed interactive tabulation software on DOS/Windows or Macintosh. We could easily turn HTML forms input into definitions for our program and display the results in HTML.

While the needed functionality was easily added, we faced major performance problems. What we found was that FreeBSD did an amazing job speeding up our application without major rewrites.

First of all, our application was optimized to produce large quantities of tables for lengthy printed reports. In the early 90's, 30-45 minutes for such a run was considered a very good time. For the WWW-based world of 1996, we had much smaller tabulations to produce and we set our limit to 7 seconds. That figure couldn't be reached by using modern machines alone.

We faced four problems:
- Some way had to be found to speed up getting only the portion of data that was immediately needed so that the CPU could be kept busy instead of waiting for data to arrive from disk.

- Our application not only uses large data sets but calls the C compiler (including assembler and linker) two times on every invocation. Besides the cost of the compiler run itself, we also recognized that most systems would kick the main data set out of the filesystem buffer cache during these compilations and so it would need to be reread from disk again when another analysis was started.

> *"Besides its advantages in performance, we found FreeBSD to be the UNIX system that was most easily managed."*

- We needed maximum disk thoughput, of course, and much more than the best disks offered.

- Our application isn't just one program, but a collection of a number of connected programs that get restarted on every invocation. The costs were negligible on a 30-minute analysis, but for the web-based product we had to make sure we could start all these processes fast enough. We also wanted to implement this as a CGI application, adding even more process starts.

Of course, all of these are problems which could probably have been best solved by changing our application, but doing so would also have required valuable programmer hours which we really needed to apply elsewhere. Such resistance to "fixing" an application is also hardly an atypical situation in almost any area of professional computing, especially when the application in question "works" (including HTML support, which was added in no time).

Believe it or not, the operating system alone can solve many such problems without having to change the application, and FreeBSD certainly came to our aid here.

The first problem was that we read only small random portions from big data sets. All other UNIX derivatives we tried could not deliver the data efficiently enough for our specific access patterns (which were not optimal, of course). The systems were either too often idle waiting for the disk or spent a lot of time in the kernel (obviously managing caches and copying data). Many did both, and I won't even start talking about the various mmap() problems we found. FreeBSD, however, managed to deliver the needed pieces of data fast enough that we achieved 90% CPU time spent in our application. This was enough to save us from having to rewrite a critical part of the application.

> *"...our own programming abilities have also improved through reading the excellent code in some of FreeBSD's subsystems."*

The next problem was that our application calls the C compiler before and after accessing the main data set. On all other systems we tried, the result was that the compiler runs caused the data set to be removed from the disk cache so that subsequent application runs on the same data had to get it from the disk again, resulting in a large performance hit. Adding memory didn't even seem to help much, these systems just seemed not to like keeping big files cached when smaller ones were being accessed and nothing could be done to tell them otherwise. Using these systems, we first thought we'd have to develop a special program to hold the data set in memory and provide it to the tabulation generator via shared memory. Since more than one data set resides on a machine and one shouldn't keep all memory for itself, we would basically need to reimplement the job of an operating system's buffer cache, simply with different preferences. A lot of work!

Happily, all of this became a moot point under FreeBSD. On a FreeBSD box with 256 MB memory we can run multiple analysis on the same 212 MB data set and the file is still in cache after the compilation runs between each analysis. Maybe this behavior can potentially hurt the performance of other applications, but we didn't find any cases of this happening. In this alone, FreeBSD saved us weeks of implementation, testing, performance analysis and tuning.

Disk throughput is, of course, also critical to our task and we found that FreeBSD's ccd (disk striping) driver scales well for us. If we add disks and controllers to a striped volume, the resulting performance out of the combined file system is pretty near the sum-total throughput of single file systems on individual drives. Frankly, we expected this to be much less. For anyone requiring disk throughput over reliability, it is nonetheless an important question to ask of those offering other RAID solutions: Assuming that they run properly at all – are they faster or slower than single disks?

Besides its advantages in performance, we found FreeBSD to be the UNIX system that was most easily managed. We often have to configure machines for additional web services and, when we do, copying an existing FreeBSD installation is done in no time, mostly due to its simple disk volume management and the fact that all configuration is done via user-editable ASCII files. Installing a new machine from scratch using sysinstall is also easy and fast – the result being already much more customized than many other UNIX system's installations leave them (network configuration, third-party-applications, all done in one pass).

The FreeBSD community is a major selling point, too. As of this writing, five people from the BSD user group Hamburg (www.bsdhh.org) have joined us as programmers and system managers. If you've already experienced trying to get good staff, you can imagine what the real value is. And the FreeBSD mailing lists are an invaluable source of information for many areas of professionally computing.

Of course, problems with getting reliable PC hardware exist also. Motherboards with only 64MB of cacheable memory, non-parity memory, or even worse, motherboards with parity support, as well as badly-designed PCI cards and cases with insufficient cooling are all problems that have plagued our computing staff.

We are very happy with FreeBSD, and our own programming abilities have also improved through reading the excellent code in some of FreeBSD's subsystems. Only a few other machines have survived here and, as time goes on, FreeBSD's position here also gets only stronger as our experience using it and our familiarity with the sources grows. We hope that you too will be motivated to try FreeBSD for your own small business or research project!

1998 USENIX Annual Technical Conference

June 15–19, 1998 Marriott Hotel, New Orleans, Louisiana

FREENIX

Come to the FREENIX track at the USENIX Conference. Topics discussed will be FreeBSD, Linux and various other types of free software on the market today. For more information see http://www.usenix.org/events/no98/index.html.

BE THERE!

Walnut Creek CDROM
4041 Pike Lane, Suite F-933
Concord, CA 94520

BULK RATE
U.S. POSTAGE
PAID
CONCORD, CA
PERMIT NO. 653

To subscribe... to this newsletter, please visit http://www.freebsd.org/register.html and fill in the name & address form.

The contents of this newsletter... were freely contributed by members of the FreeBSD user community and are fully redistributable/ reprintable in other media as long as proper credit to the authors is given.

Without volunteer authors... this newsletter would not exist. Submissions for future newsletters are always gratefully accepted and may be sent to newsletter@ freebsd.org (plain text is fine, we'll handle the formatting).

For more information... visit the FreeBSD web site ***@ http://www.freebsd.org/***

GOTTA HAVE IT!

The Complete FreeBSD

The Complete FreeBSD, 2nd edition, is an updated and expanded text containing more than 1700 pages of expert information on installing and running FreeBSD. The 33 chapters include everything from an installation guide, to tutorials on configuring your kernel, setting up a firewall, and administering a network. This book will show you how to use FreeBSD as a reliable high-performance fileserver for Win95 and Macintosh. It also explains PPP and X11 setup, and provides a tour of the ports collection. This comprehensive reference on FreeBSD is absolutely indispensible. Check inside for more details and ordering information....

For a complete listing of Walnut Creek CDROM PRODUCTS, check http://www.cdrom.com