MAGAZINE BSD

FOR NOVICE AND ADVANCED USERS

REHOSTING IN NETBSD

INSIDE

VAX/OPENVMS REHOSTING IN NETBSD 6.0 HOSTS

WHAT'S AROUND THE CORNER: A LOOK AT UPCOMING PC-BSD CHANGES

SSL FOR DOVECOT AND ROUNDCUBE FOR THE QMAIL MTA

FREEBSD UNATTENDED INSTALLATION OF SERVERS

FREEBSD PROGRAMMING PRIMER (CMS)

ORGANIZATIONAL STRUCTURE AND CULTURE AT FREEBSD

VOL.7 NO.2
ISSUE 02/2013(43)
1898-9144

iX systems
TrueNAS

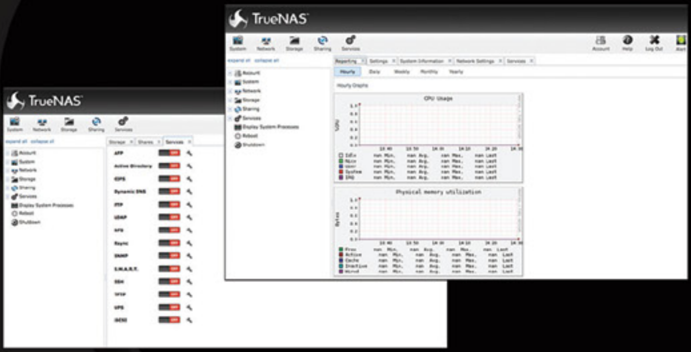
TrueNAS™ Storage Appliances: Harness The Cloud

iXsystems' TrueNAS Appliances offer scalable high-throughput, low latency storage

All TrueNAS Storage Appliances feature the Intel® Xeon® Processors 5600 series, powering the fastest data transfer speeds and lowest latency possible. TrueNAS appliances come in three lines: Performance, Archiver, & High Availability. High-performance, high-capacity ioMemory modules from Fusion-io are available in the TrueNAS Enterprise, Ultimate, and Archiver Pro models.

Key Features:

- One or Two Six-Core Intel® Xeon® Processors 5600 series
- Share Data over CIFS, NFS and iSCSI
- Hybrid storage pool increases performance and decreases energy footprint
- 128-bit ZFS file system with up to triple parity software RAID



*Optional component

	TrueNAS Pro	TrueNAS Enterprise	TrueNAS Ultimate	TrueNAS Fileshare	TrueNAS Archiver Pro	TrueNAS Pro-HA	TrueNAS Enterprise-HA	TrueNAS Ultimate-HA
	PERFORMANCE			ARCHIVER		HIGH AVAILABILITY		
Fusion-io Card		X	X		X			
Deduplication					X			
High Availability						X	X	X
Gigabit NICs	Quad	Dual	Dual	Dual	Dual	Six	Quad	Dual
10 Gigabit NICs		Dual*	Quad*		Dual*			Dual
Max Main Memory	48Gb	96Gb	192Gb	48Gb	192Gb	48Gb	96Gb	192Gb
Max Capacity	220TB	500TB	1.5PB	580TB	2.2PB	250TB	310TB	1.4PB
Rack Units	2U	2U/4U	4U	2U	4U	3U	3U	Dual 3U

iXsystems®

Call iXsystems toll free or visit our website today!

1-855-GREP-4-IX | www.iXsystems.com

intel® Xeon® inside™

Dear Readers,

In February's issue of BSD Magazine you will find the articles intended for administrators. However, they represent various levels, so no matter how long you work as one of them you should come across useful and practical content.

We start with Rob's column describing day-to-day experiences from admin life – not technical ones, but as we all know them, if you are a techie it often happens that apart from dealing with computers, systems and servers you have to deal with people. Sometimes they can be helpful like a good firewall and sometimes they can be a pain in your neck as a notorious virus.

Next, Kris Moore will tell you about the upcoming PC-BSD changes. And at the end of the issue you will find Federic Culot's article in which he compares the model of culture of organization, which according to MBA studies favors the creativity among workers with the model of relations among FreeBSD committers. It's a really pleasant read, where apart from some paragraphs similar to the ones that can be found on business course, you will find examples of small talks. Maybe you will be surprised to find out that a witty joke can increase your effectiveness? If not, you are just jaded by cozy atmosphere in BSD community. Nevertheless, it's always good to find out that it has some grounds in management studies.

In the How To section Jose Alos will teach you about rehosting the VAX and OpenVMS in NetBSD hosts. A very practical article, which might be a good solution if you would like to refresh the old platforms. Also in this section William Olson will show you step by step how to Install and Configure SSL for Dovecot and Roundcube – Open Source IMAP and POP3 email server for UNIX-like systems.

During the read of article "FreeBSD Unattended Installation of Servers" by Egoitz Aurrekoetxea Aurre you will learn how to manage the version of FreeBSD running on your machines and the provisioning or upgrading of each machine in an advantageous way. Also in admin section you can enjoy the part 2 of Rob Somervilles series on "FreeBSD Programming Primer (CMS)".

We hope you will be pleased with the read.

Patrycja Przybyłowicz
Editor of BSD Magazine
& BSD Team

MAGAZINE BSD

Editor in Chief:
Ewa Dudzic
ewa.dudzic@software.com.pl

Supportive Editor
Patrycja Przybyłowicz
patrycja.przybylowicz@software.com.pl

Contributing:
Rob Somerville, Kris Moore, Jose B. Alos, William Olson, Egoitz Aurrekoetxea Aurre, Federic Culot

Top Betatesters & Proofreaders:
Ahmed Aneeth, Radjis Mahangoe, Barry Grumbine, Bjørn Michelsen, Paul McMath, Eric Geissinger, Eric De La Cruz Lugo, Imad Soltani, Luca Ferrari

Special Thanks:
Denise Ebery

Art Director:
Ireneusz Pogroszewski

DTP:
Ireneusz Pogroszewski
ireneusz.pogroszewski@software.com.pl

Senior Consultant/Publisher:
Paweł Marciniak *pawel@software.com.pl*

CEO:
Ewa Dudzic
ewa.dudzic@software.com.pl

Production Director:
Andrzej Kuca
andrzej.kuca@software.com.pl

Executive Ad Consultant:
Ewa Dudzic
ewa.dudzic@software.com.pl

Advertising Sales:
Patrycja Przybyłowicz
patrycja.przybylowicz@software.com.pl

Publisher :
Software Media Sp. z o.o. SK
ul. Bokserska 1, 02-682 Warszawa
Poland
worldwide publishing
tel: 1 917 338 36 31
www.bsdmag.org

Software Press Sp z o.o. SK is looking for partners from all over the world. If you are interested in cooperation with us, please contact us via e-mail: editors@bsdmag.org.

Mathematical formulas created by Design Science MathType™.

Let's Talk

"You're a Nazi control freak" screamed the accountant as he stormed off in a violent rage, slamming the server room door within a few millimeters of my (fortunately) small nose for good measure. As the glass walls vibrated with the impact, I had a moment of doubt – was I right in refusing to disclose the superuser password for all the European servers to this manager?

Developers Corner

Over the Christmas holidays, PC-BSD 9.1 was released and while it offered many new features and functionality, the developers haven't sat back and relaxed just yet. In this article the author gives you a preview of some of the high-level changes that will be coming to PC-BSD a bit later this year.

How To

One of the most important issues in the long-term projects industry is the obsolescence risk that impacts specially IT infrastructure. Maybe the most paradigmatic case occurs in the aerospace industry where projects and programs last decades and, due to strong constraints, are not easy to update mainly due to legal or government requirements. From this article you will learn how to install, configure and administer VAX/OpenVMS servers hosted in NetBSD boxes. You will also find out how to avoid obsolescence risks and improve performance for former VAX/OpenVMS if you are still working with them.

Dovecot is an Open Source IMAP and POP3 email server for Linux/UNIX-like systems, written with security primarily in mind. Dovecot is an excellent choice for both small and large installations. It's fast, simple to set up, requires no special administration and it uses very little memory. In this article the author describes how to install and configure dovecot to communicate with the imaps protocol. He will show you as well how to install and configure the roundcube to communicate with dovecot.

Admin

This article tries to show how to manage an important (in terms of size) computing park when talking about the unattended installation and upgrade of FreeBSD servers. During the read you will learn how to manage the version of FreeBSD running on your machines and the provisioning or upgrading of each machine in an advantageous way. This article is addressed to the readers who already have some sysadmin experience. However, if you are a beginner in the file you can always contact the author directly and consult with him the dificulties.

In the second part of our series on programming, we will look at configuring our development server, write our first lines of code and commit the changes to a version control system. Before you will get started, you need to have a FreeBSD test server available with the AMP (Apache / MySQL / PHP) installed. If you want to follow the author step by step you will also have to use a version control system (VCS) and a CLI based text editor. The show example is based on FreeBSD 9.0 with VI, MC (for file management) and GIT running under Virtualbox.

In Business

Business Schools teach you during expensive MBA studies how managers should shape the structure and influence the culture of organizations so that they become more innovative. Concepts such as natural systems, self-organization, creative swiping, boundary spanning, empowerment… are known to be helpful in making employees more creative. To save you the trouble of paying for costly business courses and based on author's view as a committer, this article addresses the question to know whether or not FreeBSD is the right place to develop innovative ideas.

Fear, Loathing and Misunderstandings

"You're a Nazi control freak" screamed the accountant as he stormed off in a violent rage, slamming the server room door within a few millimeters of my (fortunately) small nose for good measure.

As the glass walls vibrated with the impact, I had a moment of doubt – was I right in refusing to disclose the superuser password for all the European servers to this manager? While holding more seniority than me, I did not report to him and he was not senior to the IT manager who I reported to. I sighed, obviously the compromise I offered him – the ability to access bulletin boards and the internet via a separate account to our modem pool – was not good enough. This left me somewhat perplexed, as this was the reason my bean-counting colleague wanted the password in the first place. Or was there a different agenda here? Only the IT manager and I knew the password, and this being the 80's, we had agreed to use the same password for all the servers. With hindsight, this was probably a poor decision, but in those days security was not such an issue (we were still running MS DOS 3.3 on the desktop) so the risk was small. Yet despite trying to reason with him along the lines of "with great power comes great responsibility", he had determined that the administrator password was essential to life and limb and that I was the one at fault. A few espressos (and Red Marlboro's) later and I settled down to my normal state of calm. No more was heard of the matter until my annual review 6 months later where it was commented that I was "uncooperative with certain members of management".

To be honest, I have always had a love-hate relationship with accountants. A forensic accountant placed a support call one day demanding that I change his keyboard as a vital function was missing (it was in fact a problem with the PC and the terminal emulation software not playing nice together). When I explained the problem, and that it would be fixed as soon as I could upgrade the kit, he went into a rant about "knowing a lot about IT" and that it was "definitely a broken keyboard". I was having a bad Monday morning, we had some serious X.25 issues, and a number of users were offline. The fact that Shift F10 worked fine in Wordperfect was beside the point. I was abrupt, yet calm: "I really suggest the next time you call the help desk you request someone else assists as you obviously don't trust my professional advice it is a complete waste of our time". Within 5 minutes the gentleman concerned was in the IT office apologizing. After beer-for-the-boys the following Friday night, we became firm friends and suffered many post-corporate drinking session hangovers together (as well as a few good cigars). Needless to say, once we had the kit the problem was speedily resolved without rancor. Another case of if you "can't solve the problem over a beer" it probably isn't worth it or won't be solved.

Like all business professions, there are the good and the bad. The two best accountants I know have a good grasp of IT, and as they understand the issues, are always very supportive and encouraging. Other not so. One accountant was most upset because I wanted to take his company to court over an unpaid invoice (9 months overdue) and duly cost me a contract with a sister company by spreading false rumors. As I am not in the business of bankrolling large companies with my limited cash-flow, it was no loss. In the world of business, we were both doing our jobs.

Yet, as a profession IT shares a lot in common with accountants, lawyers and auditors – we are feared, loathed and misunderstood in equal measure. While your average super-user probably has a good grasp of the general issues, many of the specifics remain hidden. Those critical servers that need a decent backup / upgrade / security patch but are neglected due to budget / time constraints / pressure to deliver something else. The fact that we have a daisy chain of extension cords under the desk that would stretch round the office twice rather than ample power provision, or that it is a hidden secret in the organization that the user account password (with the password user) will give you guest rights on our network – troubles us deeply. All these loose ends give engineers nightmares, and often the pressures of imperfection gnaw away at our souls so we do appear rather short, sarcastic or downright grumpy when asked a rather silly question. There are bad attitudes across every corporate discipline, but it seems that accountants and IT shoulder an unfair percentage of the blame when things go wrong.

So how do we resolve this enigma of two powerful groups with disparate agendas? What I have discovered is that accountants love detail and hate waste, so as engineers we have an immediate platform for rapprochement. Where some accountants fall down is a lack of creative vision, and the refusal to accept that you need to take risks to grow. Risk gives accountants nightmares. However, the IT sector is build on risk, hype and illusion yet the corporate ethos refuses to accept this. Risk is part of IT life. In reality, we are all just advanced beta-testers for the marketplace until the next "standard" comes along. If a platform / software / idea takes off and reaches critical mass it is considered "established" and we all know the age old phrase "Nobody got fired for buying IBM". The real joy from working in IT is providing solutions, fixing problems, making things better. And of course, managing the risk.

IT, like accountants, cannot work in isolation. We need to get to grips with the corporate beast, that horrible group-think that seems to enslave people where more than 2 or 3 gather, that carries with it all the divisions, petty politics, needless bureaucracy and inflexible rules that stifle growth like a thick polythene bag. The IT department may not always be the best friends of the accountants, but we can have accountants as friends. The more we share with disciplines that seem alien to us, the more our vistas are widened, and likewise those that hate us will realize the challenges we face and maybe appreciate what we do.

Oh, I nearly forgot... The accountant mentioned at the beginning of the article was caught downloading illegal software in the workplace a few months later.

ROB SOMERVILLE

Rob Somerville has been passionate about technology since his early teens. A keen advocate of open systems since the mid eighties, he has worked in many corporate sectors including finance, automotive, airlines, government and media in a variety of roles from technical support, system administrator, developer, systems integrator and IT manager. He has moved on from CP/M and nixie tubes but keeps a soldering iron handy just in case.

What's Around the Corner

A Look at Upcoming PC-BSD Changes

Over the Christmas holidays, PC-BSD 9.1 was released and while it offered many new features and functionality, we haven't sat back and relaxed just yet. In this article I want to give you a preview of some of the high-level changes that will be coming to PC-BSD a bit later this year.

First up will be a look at how we perform updating and package management, but we should begin with some of the history and philosophy behind PC-BSD.

For the past half-dozen years PC-BSD has been based upon the same release schedule of the FreeBSD releases that it was built on. Typically when FreeBSD version X was released, a PC-BSD release of the same version number would follow within a matter of days. Between these times, we often only issued minor updates for specific bug fixes to our utilities, or the occasional driver update. While this worked to help keep us in "sync" with the parent FreeBSD release, it has not been the ideal situation. Often shortly after a release we have already added new features to our utilities or improved upon the system in other ways. Additionally there have been many instances where only a mere days after a release, a new version of something critical, such as a video driver or desktop environment has been committed to the ports tree. This has created a need for a better updating framework to the binary packages which comprise a PC-BSD desktop.

A new challenge has also been in the area of server administration. Over the last year we have also begun to see more people using PC-BSD to run servers, either as TrueOS, or in jails on a host system. This has also introduced a new set of demands in the area of package management. For desktop users, PC-BSD offers most applications in the self-contained PBI format, but when it comes to servers this is often not the preferred solution. System administrators are accustomed to being able to add packages and plugins to a system via a more traditional packaging system and the ability to fine-tune their services. While many experienced FreeBSD users can do this via the ports system, we are seeing far more users coming to FreeBSD who just want to be able to install binary packages. While PC-BSD offers a number of packages on our installation media, it is still a small subset of what is available and can become outdated rather quickly. This is another challenge we plan to meet head on.

While these are large issues to tackle, we have already begun implementing new solutions to solve them. The first item we looked at solving was the area of release frequency and outdated tools & packages. Due to the fact that releases have historically been 10-14 months apart, we have decided to begin to move to a "rolling-release" model, first for the packages which make up a desktop & server, and next for the base-system. Over the past few months we have already begun the process of converting all of our utilities to work with the new "pkgng" package-management system. By converting the systems to using pkgng, we will now have the ability to solve several of these problems at once. Starting with the next release of PC-BSD, we plan on making available a complete pkgng repository. This will be constantly updated, maybe as often as weekly, and kept as close in sync with the FreeBSD ports tree as possible. In addition we have already begun the conversion of our entire PC-BSD toolchain into two ports for this repository, pcbsd-utils and pcbsd-utils-qt4, for the command-line and GUI respectively. These ports will now be updated more frequently, pushing out bug-fixes and new features on a regular basis, also making them not dependent upon a specific PC-BSD point-release.

Once the dust from the packaging changes has settled, we also plan on looking at some ways to do frequent updates to the base FreeBSD system as well. Currently we are looking at offering three system "tracks" that a user can run. First would be a typical "-RELEASE" based track, moving from FreeBSD 9.1, 9.2, 9.3 and so forth, as is done now. However, the other two will follow the newer "STABLE" and "CURRENT" tracks of FreeBSD on a yet-to-be-determined frequency, possibly quarterly. These will provide users with the ability to run both a PC-BSD desktop or server on newer FreeBSD code, testing out new features and drivers. It will prove helpful to advanced users and developers alike, allowing them to quickly get a desktop or server installed and configured, without having to fall back to compiling ports by hand.

So how will these changes affect you? It will greatly depend upon your usage and personal preference. Users who install a particular numbered release of PC-BSD will be allowed the choice of how often they want to update the base desktop packages. If you are happy with the included utilities and your particular version of window-manager, then you could sit on that version until the next FreeBSD point release, or whenever you are ready to upgrade. If you are a desktop user who wants new window-manager versions, features from the PC-BSD utilities and more, then you will now be able to stay on top of those releases on a consistent basis. For running a server or jail, you will also now have access to a constantly updated, full repository of binary packages, including the PC-BSD command-line utilities, such as the Warden. With all the PC-BSD utilities being provided as packages, for the first time it will also allow you to install vanilla FreeBSD, and roll your own PC-BSD desktop / server by using our pkgng repository. Stay tuned to BSD Magazine in the coming months fore more details as these features begin to take shape.

KRIS MOORE

Kris Moore is the founder and lead developer of PC-BSD. He lives with his wife and four children in East Tennessee (USA), and enjoys building custom PC's and gaming in his (limited) spare time. kris@pcbsd.org.

VAX/OpenVMS

Rehosting in NetBSD 6.0 Hosts

One of the most important issues in the long-term projects industry is the obsolescence risk that impacts specially IT infrastructure. Maybe the most paradigmatic case occurs in the aerospace industry where projects and programs last decades and, due to strong constraints, are not easy to update mainly due to legal or government requirements.

What you will learn…

- How to install, configure and administer VAX/OpenVMS servers hosted in NetBSD boxes
- Basic topics on configuration and TCP/IP communications for emulated systems
- Avoid obsolescence risks and improve performance for former VAX/OpenVMS if you are still working with them.
- Installation of additional VAX/OpenVMS products and software.

What you should know…

- Basic knowledge of Unix systems as a developer focused on BSD-like UNIX flavor
- Basic knowledge on VAX/ OpenVMS platform
- Network interfaces configuration and the use of TUN/TAP virtual devices for NetBSD hosts.

Fifteen years ago, development of real-time and embedded systems were attached to VAX/OpenVMS tandem. Nowadays, there exists a wide variety of aircraft and mission critical software systems that must be kept operative with no modifications for many years.

A different point of view to take into account is the existence of the DEC users community known as DECUS, which was born in 1961, although the VMS kingdom does not start until fourteen years later in 1975 thanks to the Star project. VMS reached its summit of popularity among programmers in the 80s as the “de facto” OS platform, not only for real-time applications but also for other sectors such as financial and government entities.

Despite all, the VAX/OpenVMS platform is still alive for long-term aircraft development and the reason for this paper is to shed some light on the possibilities to rehost former VMS-based architectures into a new ones with special emphasis on NetBSD Intel-based architectures.

One of the most frequent problems in projects devoted to complex system design is the obsolescence of hardware and software. This case is particularly dramatic in the aeronautical industry where the renewal rate of IT equipment is much less than the project life itself, which in many cases is greater than a decade.

Let us imagine a programmable embedded device which requires the use of RT techniques for control purposes. The right choice twenty years ago was the use of VAX systems, originally developed by Digital Equipment Corporation, running its own proprietary OS named VMS. But now we have to face the following questions:

- What to do if a component fails?
- How to keep my development alive to do further changes on it?
- How to avoid the lack of knowledge for these old-fashioned platforms?

These are the types of questions that this article tries to answer. And, as the reader can imagine, the solution requires the use of the best hardware platforms available, at a minimum cost, that can emulate both VAX hardware and OpenVMS software.

Just a final word regarding this article which was originally developed for a GNU/Linux system. Thanks to the requirement of BSD Magazine publishers, it has been amazing for me to discover that the use of NetBSD computers allows not only to simplify the process of setting up the final prototype, but also introduces a significant advantage for further maintenance and updates.

History of VAX/VMS Systems

The first VAX systems were originally developed during the 1970s by Digital Equipment Corporation as one of the best exponents of ISA platforms based on 32-bit CICS microprocessor architecture, it was intended to replace the former DEC PDP servers as well as including virtual addressing capabilities and an orthogonal set of instructions. These platforms reached their summit one decade later as the paradigm of CICS ISA architecture for RT-programmers and embedded devices, some of them are still working.

VMS stands for Virtual Memory System, was previously termed as VAX-11/VMS and provided support for DEC VAX and ALPHA platforms. Unfortunately, DEC was merged with Compaq and now Hewlett-Packard has become the current owner of this platform. VMS is a multiuser time-shared OS that was also designed to support Real-Time development and batch processing. It also offers a strong flavor of High Availability through cluster solutions and distributed servers.

These capabilities and the high reliability of the VAX/OpenVMS tandem made possible its widespread use in embedded devices development. The first steps in 1975 by DEC and its Star project, whose goal was to achieve a bigger extension of a 32-bit virtual memory for its PDP-11 predecessors, was led by Gordon Bell, Roger Gourd and Dave Cutler.

This work reach the most significant milestone with the release of the VAX 11/780 platform and the operating system VAX-11/VMS whose name has been changed along time. To satisfy the curiosity of the readers, take a look at Figure 1 to get a detailed picture of one of the most well known VAX workstations from the 80s.

As a matter of curiosity, if for UNIX operating systems the count of time begins in 1970, the start of VMS epoch starts November 17th 1858 and what is more, the smallest unit of time is 100 ns. That means the use of the OpenVMS 64-bits OS to represent time allows to avoid things similar to the famous Y2K issue, as the end of VMS era will happen on July 31 31086. Is it not funny such an exhibition of endurance?

The OpenVMS Hobbyist Project

Despite OpenVMS being a proprietary OS, there was a project started in 1997, OpenVMS Hobbyist, that can be found by surfing *http://h71000.www7.hp.com/openvms/license_software_programs/index.html* whose objective is to provide copies not only for the OpenVMS OS itself, but also for a wide variety of layered products for VAX/VMS platforms for free and not for commercial use. This fact has made possible for other companies to release their products under the same regulations for non-commercial use such as Process Software and MVP Systems.

The required time to get one of these OpenVMS Hobbyist licenses is less than one week and there is also the possibility to purchase a physical copy in CD-ROM by 30 USD. Also it is possible to get copies of the associated documentation for OpenVMS product family at *http://www.openvms.org/* and directly from *http://www.hp.com*.

Current Available Solutions

Under the umbrella of all available alternatives to extend the life of software developed for former VAX/OpenVMS platforms, there are only two possible solutions, a proprietary one with a high cost of purchase and maintenance, which is the one provided by CharonVAX, and its OpenSource counterpart available for Unix systems in their different flavors. These two solutions present the following advantages regarding to the original environment:

- Faster resources access and increased I/O performance VAX/OpenVMS
- Dramatic increase in storage capability
- Availability of RAID technologies for VAX/OpenVMS
- Integration into the corporate IT backup infrastructure.
- Avoidance of obsolescence risks by enhancing software updates for VAX/OpenVMS

CHARON VAX Emulator is a solution which allows to emulate VAX, Alpha or even the venerable PDP architecture under MS Windows according to the schema provided by Figure 1. Maybe the main advantage using this proprietary solution is the ability of handling several RS-232 serial ports or multiport cards in personal computers.

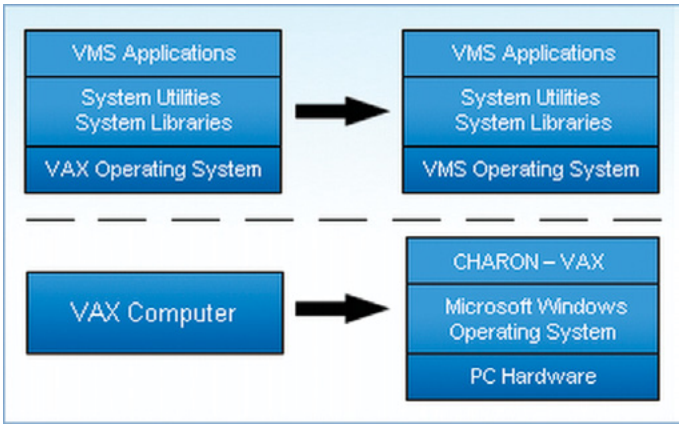


Figure 1. *Proprietary Architectures to emulate VAX/OpenVMS systems*

This latter possibility becomes specially important in the development of embedded systems, as it allows to replace DECServer 200/700 terminal servers, also affected by the natural obsolescence, one of whose major examples is provided by Figure 2.

Our proposal to avoid proprietary software to emulate such VMS/VAX platforms is the use of SIMH. SIMH is a multiplatform emulator that runs currently in MS Windows and Unix systems like OpenBSD, NetBSD, Solaris and GNU/Linux, which will be explained more in depth in the following sections.

The SIMH Emulator

The Computer History Simulator Project, based on Bob Supnik's historical computer simulator, also known as SIMH, whose source code is available at *http://simh.trailing-edge.com/*, is an emulator based on MIMIC, originally developed in 60s and whose first version was released in 1993 to provide complete emulation of former hardware and operating systems as those included in Table 1.

Among other secondary architectures as those designed by Royal-Mcbee and Scientific Data Systems. SIMH also comes with a complete bundle of utilities to ease file manipulation in the native formats of these architectures such as DEC PDP. The version covered by this article, SIMH 3.9-0 was released on May 3th 2012 and it is our recommendation for production usage.

Table 1. *List of platforms emulated by SIMH*

Digital Equipment Corporation	PDP-1/PDP-4/PDP-7/PDP-11/PDP-15/VAX
Data General	Nova / Eclipse
GRI Corporation	GRI-909
IBM	1401/1620/1130/7090/7094/System/3
Interdata	16/32 bit series
Hewlett-Packard	2116/2100/21MX
Honeywell	H316 and H516
MITS	Altair 8800 / Intel 8080

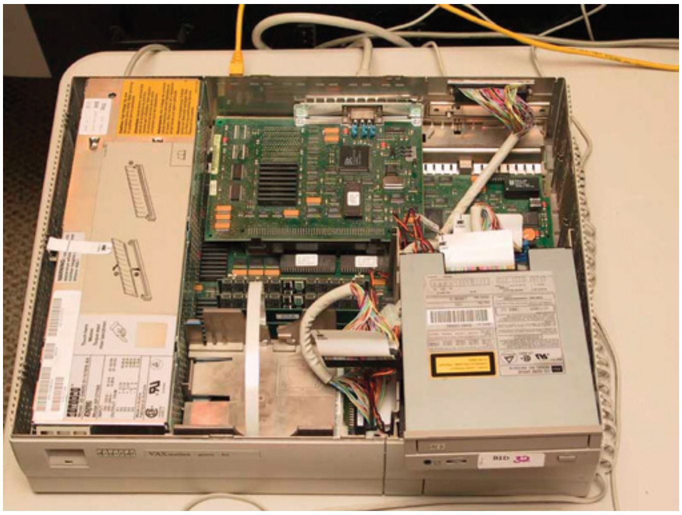


Figure 2. *Inside VAXStation 4000/90*

All the considerations made by this article can be also applied to other Unix systems like GNU/Linux and Solaris thanks to the availability of its source code and in the special case of NetBSD to the ports package system to automate the build process.

The OpenVMS Operating System

Before starting with this article, it is necessary provide a little clarification about the software to be emulated, VAX/OpenVMS, which is a proprietary software under copyright. However, Hewlett-Packard provides free licenses for non-commercial use, not only for the OpenVMS OS but also for a wide variety of products coming from the former kingdom of Digital Equipment Corporation. Such licenses can be requested in OpenVMS Hobbyist site *http://www.openvmshobbyist.com/news.php*, and they are an invaluable resource for all those people interested in training or research on VAX/OpenVMS systems.

The only requirements to request these licenses are:

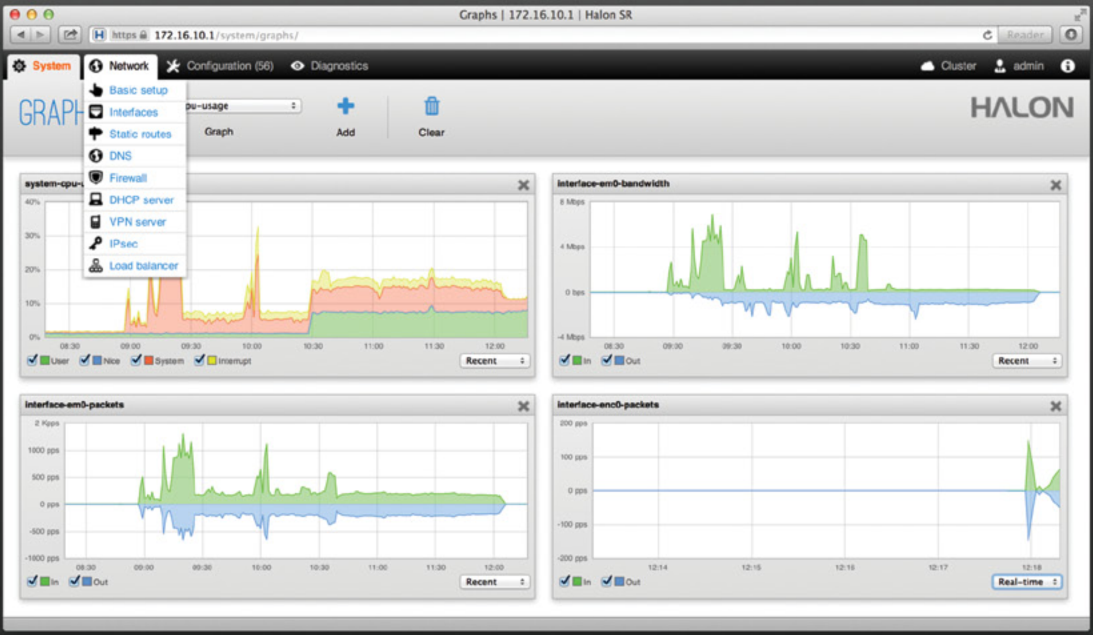
- Register in Encompass US, at the URL *https://www.encompassus.org/*
- Request the kit OpenVMS VAX Hobbyist Kit 3.0
- Optionally, purchase the licenses you could need if you intend to use OpenVMS related-products for commercial activity.

For this reason, apart from a license request, you need to get a copy of the operating systems OpenVMS 7.1 for VAX systems as well as the PAK licenses required to run these products. These considerations will apply for the remaining basic software covered by the Layered Product Licenses you might use, such as BLISS, C, PL/1, Fortan, Cobol compilers, interpreters and other proprietary applications.

Getting Started with OpenVMS/VAX Emulation by SIMH

Previous Requirements

The first activity to start with, you need a NetBSD OS running in a IA32 box with the more complete features for C/C++ development, together with libpcap library to set up a TCP/IP connection with our emulated VAX/OpenVMS platform by using SIMH. In our case, our choice has been the latest release available NetBSD 6.0 for I386 architecture. Although there are available binary pack-


Graphs | 172.16.10.1 | Halon SR
https 172.16.10.1/system/graphs/
System
Network
Configuration (56)
Diagnostics
Cluster
admin
Basic setup
Interfaces
Static routes
DNS
Firewall
DHCP server
VPN server
IPsec
Load balancer
GRAPH
Graph
Add
Clear
HALON
system-cpu-usage
User
Nice
System
Interrupt
Recent
interface-em0-bandwidth
In
Out
interface-em0-packets
interface-enc0-packets
Real-time

ages for Intel architectures, NetBSD provides a powerful and simple way to compile and install the SIMH emulator from its source code and, what is more, the latest SIMH 3.9.0 release has been already integrated in the centralized pkgsrc package manager for NetBSD 6.0, which corresponds to the pkgsrc-2012Q3 branch, released in October 2012.

Despite pkgsrc is attached to NetBSD OS, there exists the possibility of using this package management system for other Unix-like OS such as Haiku, DragonFlyBSD and Minix, hence the procedures described may also be applied for other boxes, like FreeBSD, OpenBSD and of course, GNU/Linux OS.

By default a standard installation of NetBSD 6.0 provides all necessary things and it is not necessary to recompile kernel sources. The main points to control are:

- NetBSD 6.0 GNU CC version 4.5.3 (NetBSD nb2 20110806) development environment for I386 architectures.
- Generic NetBSD 6.0 kernel is enough to include even the more advanced features like TUN/TAP interfaces. This point can be checked by the following command:

  ```
  dhcppc2# uname -a
  NetBSD dhcppc2 6.0 NetBSD 6.0 (GENERIC) i386
  ```
- By default, libpcap is installed in a standard NetBSD 6.0 distribution.

  ```
  dhcppc2# ls /usr/lib/libpcap*
  /usr/lib/libpcap.a          /usr/lib/libpcap.so.4
  /usr/lib/libpcap_p.a
  /usr/lib/libpcap.so         /usr/lib/libpcap.so.4.0
  /usr/lib/libpcap_pic.a
  ```
- The centralised management system, pkgsrc, and the latest branch pkgsrc-2012Q3 shall be downloaded and installed. To make easier further updates, we recommend to use CVS to get a copy by defining first CVS_RSH environment variable to “ssh”. Just type the command:

  ```
  $ cd /usr && cvs -q -z2 -d anoncvs@anoncvs.NetBSD.org:/
                  cvsroot \
  checkout -r pkgsrc-2012Q3 -P pkgsrc
  ```

Eventually, we have everything we need to build a running VAX/OpenVMS machine in a NetBSD box. However, it is perfectly possible to perform the build process by hand, as the source code for SIMH 3.9-0 can be downloaded from *http://simh.trailing-edge.org.*

Remember that the PCAP library is required to enable TCP/IP access to the emulated VAX workstation.

Optionally it is also interesting to compile our NetBSD kernel but including the support for TUN/TAP virtual devices as we will not need another network card. The latter point will be discussed at the end of the installation process.

Listing 1. *Installation process for SIMH in NetBSD 6.0 / pkgsrc system*

```
# cd /usr/pkgsrc/emulators/simh
# make install
=> Bootstrap dependency digest>=20010302: found digest-20111104
===> Skipping vulnerability checks.
WARNING: No /var/db/pkg/pkg-vulnerabilities file found.
WARNING: To fix run: `/usr/pkg/sbin/pkg_admin -K /var/db/pkg fetch-pkg-vulnerabilities'.
=> Fetching simhv39-0.zip
=> Total size: 3103657 bytes
Requesting http://simh.trailing-edge.com/sources/simhv39-0.zip
 17% |*****                          |   544 KiB  135.69 KiB/s    00:18 ETA
...
/usr/bin/install -c -o root -g wheel -m 644 /usr/pkgsrc/emulators/simh/work/VAX/ka655x.bin /usr/pkgsrc/emulators/
                    simh/work/.destdir/usr/pkg/share/simh
(cd /usr/pkgsrc/emulators/simh/work && for TXT in *.txt */*.txt; do              /usr/bin/install -c -o root -g
                    wheel -m 644 $TXT /usr/pkgsrc/emulators/s
imh/work/.destdir/usr/pkg/share/doc/simh;               done)
=> Automatic manual page handling
===> Building binary package for simh-3.9.0nb2
=> Creating binary package /usr/pkgsrc/packages/All/simh-3.9.0nb2.tgz
===> Install binary package of simh-3.9.0nb2
```

Installation Procedure

Once you have downloaded and installed the `pkgsrc` management system, it is time to get a copy of the SIMH 3.9.0 source code, and use the procedure provided. In this case, as we are only interested in emulating a VAX platform, we can avoid compiling the other emulators included in SIMH. Hence, the way to proceed is shown below by issuing the following commands as a superuser. Otherwise, you will build all emulators provided by SIMH for curious and vintage computers lovers (Listing 1).

As a result of this process, a set of binaries are produced in the `/usr/pkg/bin/` directory. Our binary of interest is `simh-vax` which emulates a VAX machine. The last step consists of creating a directory structure to allocate configuration files and image files for VAX hard disks. Our choice is to isolate these files in the `/opt/vax/data` directory.

If you prefer a baroque approach, you can compile the sources by hand, by issuing the following commands:

```
persephone:~$ mkdir simh
persephone:~$ cd simh
persephone:~$  unzip sim-3.9-0.zip
persephone:~simh$ mkdir BIN
persephone:~$ make USE_NETWORK=1 BIN/vax
```

Once this operation has finished, let us generate by hand the directories structure:

```
root@persephone:~# mkdir -p /opt/vax/bin
root@persephone:~# mkdir /opt/vax/data
```

and proceed to copy the binary file vax including the binary PROM code KA655 to its final directory.

```
root@persephone:~# cp BIN/vax /usr/pkg/bin
root@persephone:~# cp VAX/ka655.bin /opt/vax/data
```

Meanwhile, it is also required to create an initial configuration file `simh-vax.ini` in the same directory `/usr/pkg/bin` in which `simh-vax` is found. This file shown in Listing 2 shall cover all the basic characteristics and features for the emulated platform.

Now, create a soft link from this configuration file to the directory `/opt/vax/bin`, so that this file can be safely found and used:

```
persephone# ln -sf /opt/vax/data/simh-vax.ini
/usr/pkg/bin/simh-simh-vax.ini
```

In this configuration file, a VAX system is created with 64 MB RAM by default and three hard disks RA92 mapped into the devices `rq0`, `rq1` and `rq2`, respectively. Also this file includes a virtual CD-ROM unit mapped onto virtual device `rq3` under the file `/opt/vax/data/cd.iso`, which will be used to install a copy of OpenVMS 7.1 OS. Eventu-

Listing 2. *Initial configuration file for VAX/OpenVMS emulation with SIMH simh-vax.ini*

```
;
; Load CPU microcode
load -r /opt/vax/data/ka655x.bin
;
; Attach non-volatile RAM to a file
attach nvr /opt/vax/data/nvram.bin
;
; This virtual machine has 64M memory
set cpu 64m
;
; Define disk drive types. RA92 is largest-supported VAX
                    drive.
set rq0 ra92
set rq1 ra92
set rq2 ra92
set rq3 cdrom
;
; Attach defined drives to local files
attach rq0 /opt/vax/data/d0.dsk
attach rq1 /opt/vax/data/d1.dsk
attach rq2 /opt/vax/data/d2.dsk
;
; Attach the CD-ROM to its file (read-only)
attach -r rq3 /opt/vax/data/cd.iso
;
; Disable unused devices. It's also possible to disable
                    individual devices,
; using a construction like "set rq2 disable" if
                    desired.
;
set rl disable
set ts disable
;
; Attach Ethernet to a network interface
set xq mac=46-9E-8A-95-F8-06
attach xq eth0
;
; Now start the emulator
boot cpu
```

ally, the Ethernet address of a network adaptar shall be defined to provide TCP/IP connectivity to the emulated system.

Once these steps have been performed, it is possible to start with VAX/OpenVMS emulator by issuing the command:

```
persephone# cd /usr/pkg/bin && ./simh-vax
```

and the display output is given by Figure 3.

After running the emulator, the next thing is to install our copy of OpenVMS OS. In our case, we have chosen the release OpenVMS 7.1 to provide a fully practical example, but it can be changed with no major problems.

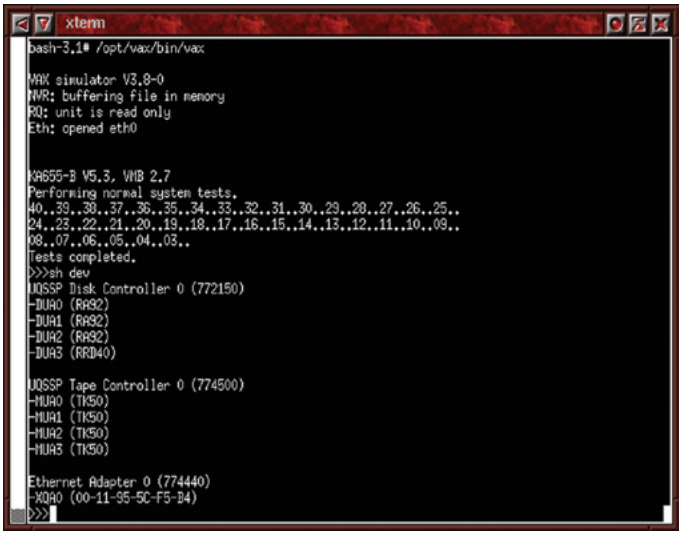
Installing OpenVMS 7.1 OS

Usually, OpenVMS 7.1 is available with CD-ROM support and to proceed with installation, it is necessary to create an ISO-9660 image named `cd.iso`. According to the settings defined in our configuration file `simh-vax.ini`, this ISO image file should be put into `/opt/vax/data`, by typing the following command:

```
persephone# dd if=/dev/cdrom of=/opt/vax/data/cd.iso
```

Furthermore, it is also required to create a 1GB empty file to act as a VAX/OpenVMS hard disk that will be used to install the VMS operating system and other utilities. This file is named `/opt/vax/data/d0.dsk` and the mapping for our virtual device is `rq0` from Listing 1.

```
root@persephone:~# dd if=/dev/zero of=/opt/vax/data/d0.iso
                   count=1024k bs=1k
```



```
bash-3.1# /opt/vax/bin/vax

VAX simulator V3.8-0
NVR: buffering file in memory
RQ: unit is read only
Eth: opened eth0


KA655-B V5.3, VMB 2.7
Performing normal system tests.
40..39..38..37..36..35..34..33..32..31..30..29..28..27..26..25..
24..23..22..21..20..19..18..17..16..15..14..13..12..11..10..09..
08..07..06..05..04..03..
Tests completed.
>>>sh dev
UQSSP Disk Controller 0 (772150)
-DUA0 (RA92)
-DUA1 (RA92)
-DUA2 (RA92)
-DUA3 (RRD40)

UQSSP Tape Controller 0 (774500)
-MUA0 (TK50)
-MUA1 (TK50)
-MUA2 (TK50)
-MUA3 (TK50)

Ethernet Adapter 0 (774440)
-XQA0 (00-11-95-5C-F5-B4)
>>>
```

Figure 3. *Starting up an emulated VAX/OpenVMS platform*

At this point, keep in mind that two decades ago, a typical hard disk as RRD92 had only 400 MB storage, which was certainly a great event for that epoch. In the same way, if you want to create another VAX hard disk, you have only to repeat this process by modifying the configuration and mappings for such devices in our `simh-vax.ini` file. Once we have reached this point, it is time to start our SIMH emulator for VAX/OpenVMS platform:

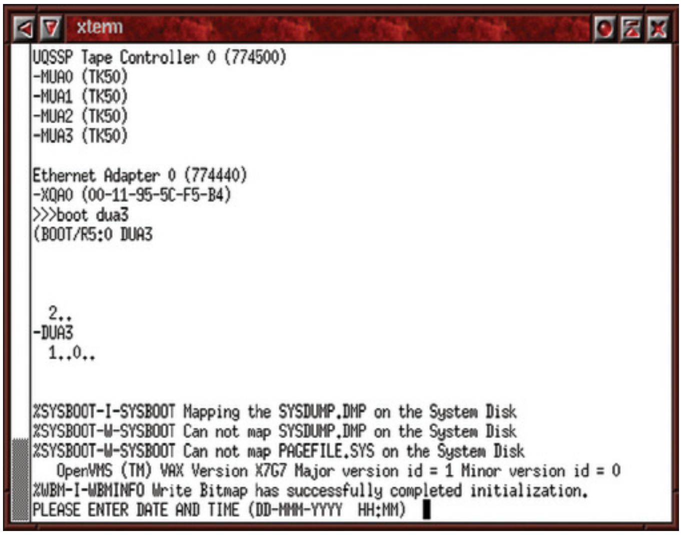
```
root@persephone:~# cd /usr/pkg/bin && /usr/pkg/bin/simh-vax
```

Hence, the device `rq3` from our VAX/OpenVMS emulator, contains the OpenVMS 7.1 OS CD-ROM to install the operating system. At a firmware level, the logical translation is given by DUA3 device as shown in Figure 4. In the present case, we should use as our main VAX hard disk the device DUA0 which will be in charge of starting the bootstrap process for our VAX/OpenVMS platform once the OS has been successfully installed.

Follow faithfully the steps given by the interactive installation assistant, by providing all requested information. Once OpenVMS 7.1 has finished the installation process, restart the emulator from DUA0 instead of using DUA3, by issuing the set of commands given in Listing 3.

When the OpenVMS 7.1 OS is ready and shows its prompt, it is time to install the licenses as well as those corresponding to the OpenVMS Layered Components that have been previously received from Hobbyist project.

By default, the list of installed products apart from our OpenVMS 7.1 operating system are given by the VMS command and the results in Listing 4.



```
UQSSP Tape Controller 0 (774500)
-MUA0 (TK50)
-MUA1 (TK50)
-MUA2 (TK50)
-MUA3 (TK50)

Ethernet Adapter 0 (774440)
-XQA0 (00-11-95-5C-F5-B4)
>>>boot dua3
(BOOT/R5:0 DUA3



  2..
-DUA3
  1..0..


%SYSBOOT-I-SYSBOOT Mapping the SYSDUMP.DMP on the System Disk
%SYSBOOT-W-SYSBOOT Can not map SYSDUMP.DMP on the System Disk
%SYSBOOT-W-SYSBOOT Can not map PAGEFILE.SYS on the System Disk
   OpenVMS (TM) VAX Version X7G7 Major version id = 1 Minor version id = 0
%WBM-I-WBMINFO Write Bitmap has successfully completed initialization.
PLEASE ENTER DATE AND TIME (DD-MMM-YYYY  HH:MM)
```

Figure 4. *Starting up OpenVMS Operating System installation using DUA3*

Listing 3. *Starting the installation process for OpenVMS in SIMH emulation*

```
persephone# cd /usr/pkg/bin && /usr/pkg/bin/simh-vax

VAX simulator V3.9-0
NVR: buffering file in memory
./simh-vax.ini> set rq4 ra92
Unit disabled
RQ: unit is read only
./simh-vax.ini> attach -r rq4 /opt/vax/data/cws860.bak
Unit disabled
./simh-vax.ini> attach -r rq5 /opt/vax/data/xd-ada-mc68020.iso
Unit disabled
libpcap version 1.1.1
Eth: opened OS device re0

KA655-B V5.3, VMB 2.7
Performing normal system tests.
40..39..38..37..36..35..34..33..32..31..30..29..28..27..26..25..
24..23..22..21..20..19..18..17..16..15..14..13..12..11..10..09..
08..07..06..05..04..03..
Tests completed.
>>>b dua0
(BOOT/R5:0 DUA0

Available devices for this installation are:

Available device  DUA0:                            device type RA92
Available device  DUA1:                            device type RA92
Available device  DUA2:                            device type RA92
Available device  DUA3:                            device type RRD40
Available device  DYA0:                            device type RX02
Available device  DYA1:                            device type RX02
Available device  MUA0:                            device type TK50
Available device  MUA1:                            device type TK50
Available device  MUA2:                            device type TK50
Available device  MUA3:                            device type TK50
```

Listing 4. *Software details in OpenVMS 7.1 OS Installation*

```
$ product show product *
----------------------------------- ----------- ------------
PRODUCT                             KIT TYPE    STATE
----------------------------------- ----------- ------------
DEC VAXVMS DECNET_PHASE_IV V7.1     Full LP     Installed
DEC VAXVMS TCPIP V5.1-15            Full LP     Installed
DEC VAXVMS VMS V7.1                 Transition  Installed
----------------------------------- ----------- ------------
3 items found
```

Optionally, you can also install additional products required for your further usage. We will only focus on the TCP/IP communications suite, the use of native DECnet protocol will be omitted.

Installing optional products VAX/OpenVMS

Again, by means of our ISO-9660 image `cd.iso` containing a copy of OpenVMS 7.1, we proceed to install the additional components required to enable Ethernet and TCP/IP support. In a running VAX session, type the OpenVMS command in Listing 5.

To install TCP/IP support, it is necessary to access to the VMS directory termed `DUA3:[TCPIP_VAX051.KIT]` in which the product under study may be found.

```
$ dir dua3:[tcpip_vax051.kit]
DEC-VAXVMS-TCPIP-V0501-15-1.PCSI;1
Total of 1 file.
```

Listing 5. *Installing optional components OpenVMS*

```
$ mount/override=id dua3:
%MOUNT-I-WRITELOCK, volume is write locked
%MOUNT-I-MOUNTED, VAXVMS073 mounted on _ARES$DUA3:
$ dir dua3:[0,0]

Directory DUA3:[000,000]

000000.DIR;1        BACKUP.SYS;1        BADBLK.SYS;1        BADLOG.SYS;1
BITMAP.SYS;1        CONTIN.SYS;1        CORIMG.SYS;1        DCE_VAX030.DIR;1
DECNET_PHASE_IV_VAX073.DIR;1            DECNET_PLUS_VAX073.DIR;1
DECW073.C;1         DECW073.D;1         DECW073.E;1         DECW073.F;1
DOCUMENTATION.DIR;1 DWMOTIF_VAX125.DIR;1                    DWMOTIF_VAX126.DIR;1
HELP_MESSAGE.DIR;1  INDEXF.SYS;1        ISL_SCRIPT.ESS;1    KERBEROS_VAX010.DIR;1
KITS.DIR;1          SECURITY.SYS;1      SYS0.DIR;1          SYS1.DIR;1
TCPIP_VAX051.DIR;1  VAX073.DIR;1        VMS073.A;1          VMS073.B;1
VMS073.C;1          VMS073.D;1          VMS073.E;1          VMS073.F;1
VMSI18N_VAX073.DIR;1                    VOLSET.SYS;1

Total of 35 files.
```

Listing 6. *Installing TCP/IP support for VAX/OpenVMS system*

```
$ product
_Operation (INSTALL, SHOW, ...): install
_Product name (* to show list): *

The following product has been selected:
    DEC VAXVMS TCPIP V5.1-15               Layered Product

Do you want to continue? [YES]

Configuration phase starting ...

You will be asked to choose options, if any, for each selected product and for
any products that may be installed to satisfy software dependency requirements.

DEC VAXVMS TCPIP V5.1-15: Compaq TCP/IP Services for OpenVMS.

    (c) Compaq Computer Corporation 2000. All Rights Reserved.
```

Great Specials

On FreeBSD® & PC-BSD® Merchandise

Give us a call & ask about our
SOFTWARE BUNDLES
1.925.240.6652

$39.95
FreeBSD 9.1 Jewel Case CD Set
or FreeBSD 9.1 DVD

$29.95
PC-BSD 9.1 DVD

$49.95
Save a BUNDLE!
The PC-BSD 9.0 Users Handbook
PC-BSD 9.1 DVD

$99.95
The FreeBSD CD or DVD Bundle

Inside each CD/DVD Bundle, you'll find:
FreeBSD Handbook, 3rd Edition
Users Guide FreeBSD Handbook, 3rd Edition, Admin Guide
FreeBSD 9.1 CD or DVD set
FreeBSD Toolkit DVD

Stylish Dress Attire
Look Your Professional Best

Comfy Apparel
Stay Warm in Zip Ups & Pullovers

T-Shirts
Lots of Styles to Choose From

FreeBSD 9.1 Jewel Case CD/DVD	$39.95

CD Set Contains:

Disc 1	Installation Boot LiveCD (i386)
Disc 2	Essential Packages Xorg (i386)
Disc 3	Essential Packages, GNOME2 (i386)
Disc 4	Essential Packages (i386)

FreeBSD 9.0 CD	$39.95
FreeBSD 9.0 DVD	$39.95

FreeBSD Subscriptions

Save time and $$$ by subscribing to regular updates of FreeBSD

FreeBSD Subscription, start with CD 9.1	$29.95
FreeBSD Subscription, start with DVD 9.1	$29.95
FreeBSD Subscription, start with CD 9.0	$29.95
FreeBSD Subscription, start with DVD 9.0	$29.95

PC-BSD 9.1 DVD (Isotope Edition)

PC-BSD 9.1 DVD	$29.95
PC-BSD Subscription	$19.95

The FreeBSD Handbook

The FreeBSD Handbook, Volume 1 (User Guide)	$39.95
The FreeBSD Handbook, Volume 2 (Admin Guide)	$39.95

The FreeBSD Handbook Specials

The FreeBSD Handbook, Volume 2 (Both Volumes)	$59.95
The FreeBSD Handbook, Both Volumes & FreeBSD 9.1	$79.95

PC-BSD 9.0 Users Handbook	$24.95
BSD Magazine	$11.99
The FreeBSD Toolkit DVD	$39.95
FreeBSD Mousepad	$10.00
FreeBSD & PCBSD Caps	$20.00
BSD Daemon Horns	$2.00

Bundle Specials!
Save $$$

Just Plain Fun
Mousepads & Novelty Horns

BSD Magazine
Available Monthly

FreeBSD Mall

For even MORE items
visit our website today!
www.FreeBSDMall.com

And proceed to install it by using a DCL utility named product, as shown by Listing 6.

Once these steps have been successfully executed, the VAX/OpenVMS is now ready to be used in our NetBSD host and may be used for further non-commercial purposes.

TCP/IP Communications in VAX/OpenVMS Systems

As it was introduced previously, access to the TCP/IP connection with the emulated system relies on two basic points: the compilation of SIMH with Ethernet support enabled thanks to PCAP library, and the existence of a free-of-use Ethernet adapter in our host running NetBSD.

In our particular case, this interface appears as `re1` and it is possible to realise that the Ethernet address choosen for VAX Ethernet interface in `simh-vax.ini` configuration file does match with the one of `re1` network interface, as it is shown below in Listing 7. Notice that this interface shall not be physically connected to other devices such as switches or hubs, as its use will be internal to the NetBSD box.

Once the availability of such interface has been confirmed, it is time to configure TCP/IP in our VAX/OpenVMS platform by means of the following DCL commands:

```
$@sys$manager:tcpip$config
```

or, alternatively, as UCX, which is the predecessor of TCP/IP implementation for VAX/OpenVMS systems:

```
$@sys$manager:ucx$config
```

This command leads to an interactive menu shown in Listing 8 that will be used to configure in a proper way our network interface to allow VAX/OpenVMS connectivity.

Listing 7. *Network interface configuration in NetBSD*

```
dhcppc2# ifconfig re1
re1: flags=8b43<UP,BROADCAST,RUNNING,PROMISC,ALLMULTI,SIMPLEX,MULTICAST> mtu 1500
        capabilities=3f80<TSO4,IP4CSUM_Rx,IP4CSUM_Tx,TCP4CSUM_Rx,TCP4CSUM_Tx,UDP4CSUM_Rx,UDP4CSUM_Tx>
        enabled=0
        address: 00:11:95:5c:f5:b4
        media: Ethernet autoselect (100baseTX full-duplex)
        status: no network
        inet 0.0.0.0 netmask 0xff000000 broadcast 255.255.255.255
        inet6 fe80::218:f3ff:fef9:a803%re1 prefixlen 64 scopeid 0x1
```

Listing 8. *TCP/IP Services Configuration for OpenVMS 7.1 host*

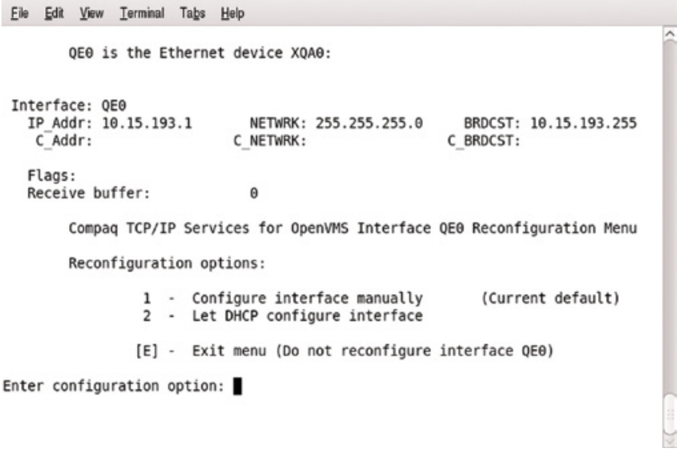
```
        Compaq TCP/IP Services for OpenVMS Configuration Menu

        Configuration options:

                 1  -  Core environment
                 2  -  Client components
                 3  -  Server components
                 4  -  Optional components

                 5  -  Shutdown Compaq TCP/IP Services for OpenVMS
                 6  -  Startup Compaq TCP/IP Services for OpenVMS
                 7  -  Run tests

                 A  -  Configure options 1 - 4
                [E] -  Exit configuration procedure

Enter configuration option: 1
```

In this way, we get the following result given by Figure 5.

Once this task has been concluded, the option (6) from the interactive menu, allows to start all TCP/IP services associated with the emulated VAX/OpenVMS. It may be necessary to modify some of the configuration parameters specific to the OpenVMS kernel.

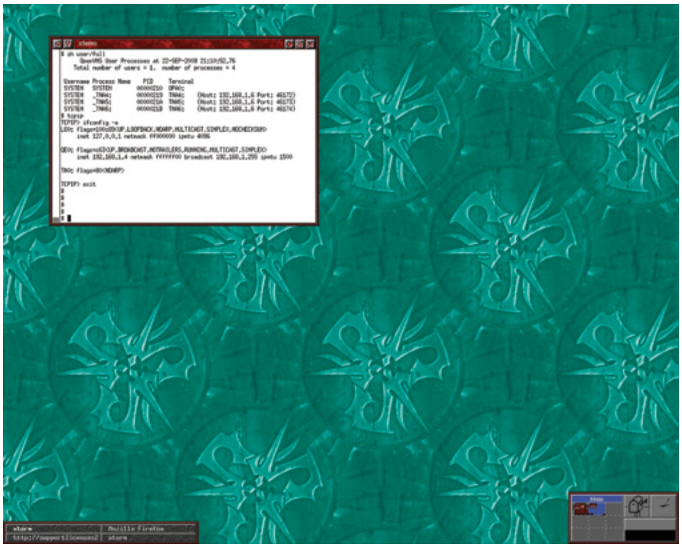
In this way we get, as a result of our work, a running VAX/OpenVMS system in which OpenVMS 7.1 runs TCP/IP support, and allows remote connectivity as it is depicted by Figure 6.

As it has been shown, we have been able to solve one of the most critical issues affecting VAX platforms, related to the obsolescence, which is closely related to hardware failures, by emulating VAX interfaces on the devices and interfaces present in a current computer, with even more available resources.



```
        QE0 is the Ethernet device XQA0:


 Interface: QE0
   IP_Addr: 10.15.193.1          NETWRK: 255.255.255.0        BRDCST: 10.15.193.255
    C_Addr:                    C_NETWRK:                    C_BRDCST:

   Flags:
   Receive buffer:               0

        Compaq TCP/IP Services for OpenVMS Interface QE0 Reconfiguration Menu

        Reconfiguration options:

                  1  -  Configure interface manually        (Current default)
                  2  -  Let DHCP configure interface

                 [E] -  Exit menu (Do not reconfigure interface QE0)

Enter configuration option:
```

Figure 5. *Definition of a Ethernet network interface in OpenVMS*



```
$ sh user/full
      OpenVMS User Processes at 22-SEP-2008 21:10:52.76
    Total number of users = 1,  number of processes = 4

 Username Process Name    PID     Terminal
 SYSTEM   SYSTEM        00000210  OPA0:
 SYSTEM   _TNA4:        00000219  TNA4:    (Host: 192.168.1.6 Port: 46172)
 SYSTEM   _TNA5:        0000021A  TNA5:    (Host: 192.168.1.6 Port: 46173)
 SYSTEM   _TNA6:        0000021B  TNA6:    (Host: 192.168.1.6 Port: 46174)
$ tcpip
TCPIP> ifconfig -a
LO0: flags=100c09<UP,LOOPBACK,NOARP,MULTICAST,SIMPLEX,NOCHECKSUM>
      inet 127.0.0.1 netmask ff000000 ipmtu 4096

QE0: flags=c63<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,SIMPLEX>
      inet 192.168.1.4 netmask ffffff00 broadcast 192.168.1.255 ipmtu 1500

TN0: flags=80<NOARP>

TCPIP> exit
$
$
$
$
$
```

Figure 6. *TCP/IP Connection from NetBSD box to the emulated VAX/OpenVMS server*

Final Tasks

Installing DEC OS/F Motif

Before starting with DEC OS/F Motif installation, it is necessary to reconfigure OpenVMS kernel. More precisely, the parameter `PQL_MBYTLM` shall be set at 48000 at least.

```
 - SYSGEN parameter PQL_MBYTLM is 40000, should be at
                    least 48000
```

The most suitable procedure to reconfigure a OpenVMS kernel is the use of the sequence of commands provided by Listing 9.

Once the kernel has been rebuilt, you have to install the package associated with OS/F Motif for OpenVMS which may be found on the installation CD-ROM associated to the image file `cd.iso`. The steps required for such a process are:

- Mount the CD-ROM in our VAX/OpenVMS by typing the command:
  `$ set default DUA3:[DWMOTIF_VAX126.KIT]`
- Install DEC OS/F Motif for OpenVMS by executing the commands given in Listing 10.
- Finally, reboot the emulated VAX/OpenVMS platform to commit the changes.
  `$ shutdown`

Listing 9. *Reconfiguration process for OpenVMS 7.1 kernel*

```
@SYS$UPDATE:AUTOGEN GETDATA REBOOT CHECK_FEEDBACK
$ @SYS$UPDATE:AUTOGEN GETDATA REBOOT CHECK_FEEDBACK
%AUTOGEN-I-BEGIN, GETDATA phase is beginning.
%AUTOGEN-I-NEWFILE,  Previous contents of
                    SYS$SYSTEM:CLU$PARAMS.DAT have
  been copied to SYS$SYSTEM:CLU$PARAMS.OLD.  You may
                    wish to purge
  SYS$SYSTEM:CLU$PARAMS.OLD.
%AUTOGEN-I-NEWFILE, A new version of
                    SYS$SYSTEM:PARAMS.DAT has been
                    created.
        You may wish to purge this file.
%AUTOGEN-I-END, GETDATA phase has successfully
                    completed.
%AUTOGEN-I-BEGIN, GENPARAMS phase is beginning.
%AUTOGEN-I-NEWFILE, A new version of
                    SYS$MANAGER:VMSIMAGES.DAT has been
                    created.
        You may wish to purge this file.
```

After rebooting VAX/OpenVMS, our platform will be ready to use some services like XDM which makes possible to use X11 clients remotely.

Beyond Emulation: Interfaces TUN/TAP

Very often, there are no available Ethernet interfaces in a computer, but many Unix systems allow to bypass this difficulty by using a virtual emulation for such devices. NetBSD OS will not be the exception as this emulation is provided by TUN/TAP modules, for TCP/IP layers 2 and 3, respectively. In this way, TUN allows the creation of a virtual NIC with an Ethernet address attached to it that is different from the physical address of the real card. The steps to create such a virtual TUN-based interface are summarized in three steps:

- Regarding TUN/TAP devices, for NetBSD the GENERIC kernel already contains the two necessary options:

```
pseudo-device   tap              # virtual Ethernet
pseudo-device   tun              # network
                 tunneling over tty
```

Alternatively, it is possible to check the feasibility of such network pseudo-devices by issuing the command:

```
dhcppc2# ifconfig -C
agr bridge vlan stf gif gre tun tap strip sl pppoe ppp lo
```

- Configure a new TAP pseudo-device by means of ifconfig(1) command:

```
dhcppc2# ifconfig tap0 create
dhcppc2# ifconfig tap0
tap0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
        address: f2:0b:a4:7f:08:0b
        media: Ethernet autoselect
```

In this way, a new Ethernet interface appears in our host system. This interface is mapped as `tap0`, with its own Ethernet address and should remain inactive as it will only be used for the emulated VAX/OpenVMS system.

Listing 10. *Installing DEC OS/F Motif for OpenVMS systems*

```
$ product
_Operation (INSTALL, SHOW, ...): install
_Product name (* to show list): *

The following product has been selected:
    DEC VAXVMS DWMOTIF V1.2-6              Layered
                    Product

Do you want to continue? [YES]

Configuration phase starting ...

You will be asked to choose options, if any, for each
                    selected product and for
any products that may be installed to satisfy software
                    dependency requirements.

DEC VAXVMS DWMOTIF V1.2-6: DECwindows Motif

    ? 1988, 2000 Compaq Computer Corporation

    Compaq Computer Corporation

    This product uses the PAK: DW-MOTIF

Checking values of system parameters...

                                       OK
```

```
Do you want the defaults for all options? [YES] YES

    If a Local Language Variant is installed refer to
                    the Install Guide.

    Do you want to continue? [YES] YES

Do you want to review the options? [NO]

The following product has been installed:
    DEC VAXVMS DWMOTIF V1.2-6              Layered
                    Product

Finally, we get the following messages to indicate the
                    installation is complete.

DEC VAXVMS DWMOTIF V1.2-6: DECwindows Motif

    System reboot will be required following upgrade of
                    language variants.

    Installation Verification Procedure can be run after
                    reboot.
```

- Edit the file `simh-vax.ini` given in Listing 1, by modifying the entry associated, in such a way the chosen IP address for out `tap0` will be reflected.

```
; Attach Ethernet to a network interface
set xq mac=f2-0b-a4-7f-08-0b
```

It is very important to take into account that you cannot add an IP address to this interface, as this task is managed directly by the emulated VAX/OpenVMS system and it is transparent to our NetBSD system which acts as host.

The process covered in this section consists of creating a bridge between our NetBSD host and the emulated VAX/OpenVMS platform as it is shown in Figure 7.

Final Remarks

For those people who think that OpenVMS is a legacy of our past and is definitively dead, try to remember that the first version for VMS OS 1.0 was released in 1978 and the OpenVMS support for IA64 (Intel Itanium platforms) has been available from HP Compact since 2004. However, nowadays it is impossible to purchase original VAX hardware.

The solution introduced in this article admits more sophisticated possibilities, it is even possible to run an emulated VAX workstation in one of the numerous BSD or GNU/Linux live distributions provided by a CD-ROM or DVD-ROM. Eventually, another interesting alternative is the use of QEMU to test the latter solution.

In the nineties, Compact purchased Digital Equipment Corporation only to merge four years later with the giant Hewlett-Packard. However, and to conclude this article, there is no better argument that the anecdote starred by Dave Cutler, one of the creators of VMS and further, one of the engineers who led the Windows New Technology project, which gave rise to MS Windows NT. The acronym WNT is nothing else than the result of shifting the three letters of 'VMS' to the next letter in the alphabet, getting the result: WNT. Perhaps this step forward in letters was also a step backwards in technological performance.

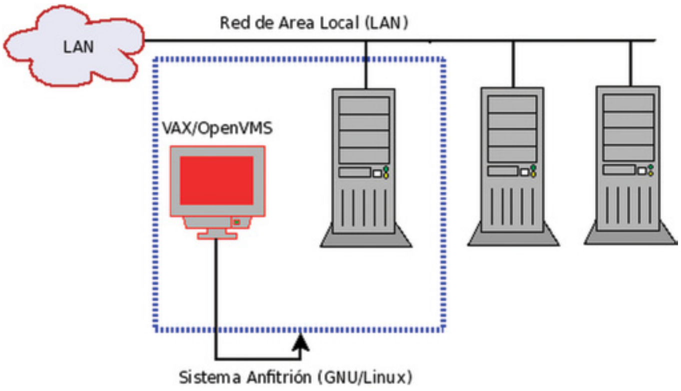


Figure 7. *Interconnectivity between host (NetBSD 6.0) and emulated OpenVMS platforms*

Acronyms and Abbreviations

CICS	Complex Instruction Set Computer
DCL	Digital Command Language
DEC	Digital Equipment Corporation
DECUS	Digital Equipment Corporation Users' Society
IA32	Intel Architecture 32-bit
ISA	Instruction Set Architecture
NIC	Network Interface Card
RAM	Random Access Memory
VAX	Virtual Address Extension
VMS	Virtual Memory System
XDM	X11 Display Manager

References

- *http://www.netbsd.org* NetBSD Main Site
- *http://www.pkgsrc.org* Framework for building third-party software on NetBSD and other Unix-like systems
- *http://simh.trailing-edge.com/* The Computer History Simulation Project (SIMH)
- *http://www.openvms.org/* OpenVMS Community Portal
- *http://h71000.www7.hp.com/doc/index.html* HP OpenVMS Systems Documentation
- *http://www.openvmshobbyist.com/news.php,* Compaq OpenVMS Hobbyist portal

JOSÉ B. ALÓS

José B. Alós began his professional career in 1999 with EDS, as UNIX System Administrator mainly focused on SunOS/Solaris, BSD and GNU/Linux. Five years ago he joined EADS Defense and Security, and nowadays working for CASSIDIAN he is responsible for providing end-user support in aircraft engineering departments for long-term projects. That is the main reason underneath this article as VAX/VMS systems play a paramount role in today's aerospace industry for a wide variety of embedded RT systems conceived for mission and flight operations.

He was also Assistant Professor in the Universidad de Zaragoza (Spain), specializing in the design of High Availability solutions and his academic background includes a PhD in Nuclear Engineering and three MsC in Electrical and Mechanical Engineering, Theoretical Physics and Applied Mathematics.

Installing

and Configuring SSL for Dovecot and Roundcube for the Qmail MTA

Dovecot is an open source IMAP and POP3 email server for UNIX-like systems, written with security primarily in mind.

What you will learn…

- How to install dovecot from the FreeBSD Ports system
- How to configure dovecot to communicate with the imaps protocol
- How to install roundcube from the FreeBSD Ports system
- How to configure roundcube to communicate with dovecot

What you should know…

- How to run any qmail service with deamontools installed (or on my freebsdrocks.net guide)
- A strong understanding of qmail.
- A solid understanding of Apache 2.2 or better with SSL (recommended)
- An installation of php 5.3 and configured with Apache 2.2

Dovecot is an excellent choice for both small and large installations. It's fast, simple to set up, requires no special administration and it uses very little memory.

Most Notable Dovecot Features

- Dovecot is high performing and is compatible with vpopmail's Maildir structure.
- When authenticating via SSL you can use dovecot to send emails from roundcube
- The dovecot service will automatically restart if it fails for any reason
- Dovecot supports migration from Courier imap
- Dovecot is configured with security in mind

The first step is to install Dovecot from the FreeBSD ports system.

```
# cd /usr/ports/mail/dovecot
# make install
```

When you run make install it will give you the various configure options available. Make sure the following options are checked:

- KQUEUE
- SSL
- VPOPMAIL

Configuring Dovecot

Dovecot itself is configured using the `dovecot.conf` located at `/usr/local/etc/dovecot.conf` for the main configuration file. The `dovecot.conf` is a VERY large configuration file. Please verify the settings from Listing 1.

You may leave the next line commented or not at your discretion:

```
# Greeting message for clients. This is an optional setting.
login_greeting = Dovecot ready.
```

More settings for the dovecot.conf file that will need to be verified.

```
mail_location = maildir:/usr/home/vpopmail/domains/%d/%n/
                    Maildir
```

Scroll down some more until you see the following sections and verify the settings from Listing 2.

What we need to do after setting up the configuration file is to start setting up the service for dovecot as shown

Listing 1. *Settings for the dovecot.conf file that will need to be verified*

```
# Base directory where to store runtime data.
base_dir = /var/run/dovecot/

# Protocols we want to be serving: imap imaps pop3 pop3s
                    managesieve
# If you only want to use dovecot-auth, you can set this
                    to "none".
protocols = imaps

   protocol imap {
     #listen = *:143
     ssl_listen = x.x.x.x:993  (Change x.x.x.x to your
                    IP address)
   }

ssl_cert_file = /var/qmail/control/servercert.pem
ssl_key_file = /var/qmail/control/servercert.pem
```

Listing 2. *More settings for the dovecot.conf file that will need to be verified*

```
# Valid UID range for users, defaults to 500 and above.
                    This is mostly
# to make sure that users can't log in as daemons or
                    other system users.
# Note that denying root logins is hardcoded to dovecot
                    binary and can't
# be done even if first_valid_uid is set to 0.
first_valid_uid = 89
#last_valid_uid = 89

# Valid GID range for users, defaults to non-root/wheel.
                    Users having
# non-valid GID as primary group ID aren't allowed to
                    log in. If user
# belongs to supplementary groups with non-valid GIDs,
                    those groups are
# not set.
first_valid_gid = 89
#last_valid_gid = 89

# Authentication cache size in kilobytes. 0 means it's
                    disabled.
# Note that bsdauth, PAM and vpopmail require cache_key
                    to be set for caching
# to be used.
auth_cache_size = 1
```

```
# vpopmail authentication <doc/wiki/AuthDatabase.
                    VPopMail.txt>
passdb vpopmail {
#[cache_key=<key>] - See cache_key in PAM for
                    explanation.
#[quota_template=<template>] - %q expands to Maildir++
                    quota
#  (eg. quota_template=quota_rule=*:backend=%q)
args =
}

# vpopmail <doc/wiki/AuthDatabase.VPopMail.txt>
userdb vpopmail {
}

# User to use for the process. This user needs access to
                    only user and
# password databases, nothing else. Only shadow and pam
                    authentication
# requires roots, so use something else if possible.
                    Note that passwd
# authentication with BSDs internally accesses shadow
                    files, which also
# requires roots. Note that this user is NOT used to
                    access mails.
# That user is specified by userdb above.
user = vpopmail
```

Listing 3. *The following commands will need to be run on the console*

```
# mkdir -m 0755 /var/qmail/supervise/dovecot-ssl /var/
                    qmail/supervise/dovecot-ssl/log /var/
                    log/qmail/dovecot-ssl
# cd /var/qmail/supervise/dovecot-ssl/log
# fetch http://www.freebsdrocks.net/files/service-any-
                    log-run
# mv service-any-log-run run
# chmod 0755 run
# vi run
```

Listing 4. *The following commands will need to be run on the console*

```
# cd /var/qmail/supervise/dovecot-ssl
# fetch http://freebsdrocks.net/files/service-dovecot-run
# mv service-dovecot-run run
# chmod 0755 run
```

in Listing 3. Change the last line in the run file at the bottom to read as in following Listing 4.

When editing the run file change the last line to the following:

```
multilog t n1024 s1048576 /var/log/qmail/dovecot-ssl
```

There are no options to configure in this file. This will run using the options in the `dovecot.conf` file.

This is just like starting up any other daemontools service – create a symlink from `/service/something` to the physical service directory, wait about ten seconds, and make sure it's running.

```
# ln -s /var/qmail/supervise/dovecot-ssl /service/
```

After a few seconds run the code from listing 5.

If the dovecot/log service does not start it is more than likely permissions or a vpopmail UID/GID issue. Check the dovecot log file in `/var/log/qmail/dovecot-ssl/log/current` for any errors.

If you happen to run an older version of qmail or even an older version of the freebsd walkthrough then the UID/GIDs for vpopmail/vchkpw may not be 89:89. You may need to look in `/etc/masterpasswd` for the vpopmail UID and `/etc/groups` for the vchkpw GID. This needs to match the UID and GID in the dovecot configuration file located at `/usr/local/etc/dovecot.conf`.

Installing and Configuring Roundcube

Roundcube is a browser based imap client. The Roundcube webmail software is available in FreeBSD ports. If you want to learn more about FreeBSD packages and ports, please read The FreeBSD Handbook, chapter 4.

Most Notable Roundcube Features

- Address Book with autocomplete features
- HTML and Richtext messages
- Shared/Global IMAP Folder Management
- Spell Checking
- Unlimited users and messages

The port for Roundcube webmail is available in `/usr/ports/mail/roundcube`. To install roundcube, you will need to type the following:

```
# cd /usr/ports/mail/roundcube
# make install clean
```

Make sure the following options are checked:

- GD
- PSPELL (Optional)
- SSL
- MYSQL

You will also want to install the following port if they are not installed already. You can check the installation of each port by running `pkg_info | grep packagename`. For instance if you wanted to find out if php5-exif is installed you would run `pkg_info | grep php5-exif`. If it returns a result then skip to the next port. If not you will need to install the port.

```
# cd /usr/ports/graphics/php53-exif
# make install clean
```

By default, roundcube is installed in `/usr/local/www/roundcube/`.

Listing 5. *The following commands will need to be run on the console*

```
# svstat /service/dovecot-ssl  /service/dovecot-ssl/log

/service/dovecot-ssl: up (pid 23841) 8 seconds
/service/dovecot-ssl/log: up (pid 23843) 8 seconds

If the first dovecot-ssl service does not start you can check the log file like so:

# tail -f /var/log/qmail/dovecot-ssl/current | tai64nlocal
```

Listing 6. *These are the commands that will need to be run on the mysql command prompt*

```
mysql> CREATE DATABASE r0undcube;
mysql> GRANT select,insert,update,delete,create,drop ON r0undcube.* TO rcub3@localhost IDENTIFIED BY 'mypass';
mysql> quit
```

Now, we need to map the `http://localhost/roundcube/` to `/usr/local/www/roundcube/`. To do this you will need to open the apache configuration located in `/usr/local/etc/apache22`. Edit `/usr/local/etc/httpd.conf` then copy and paste the following under the alias section in the `httpd.conf` file:

```
Alias /roundcube "/usr/local/www/roundcube/"
<Directory "/usr/local/www/roundcube">
Options Indexes FollowSymLinks
AllowOverride All
Order allow,deny
Allow from all
</Directory>
```

Save the file and then restart apache:

```
# /usr/local/etc/rc.d/apache22  restart
```

You now need to create a database and a username/password combination so Roundcube has access to MySQL. In the example below I am using the following settings:

```
Database Name: r0undcube
username for Database Access: rcub3
Password for rcube username: mypass
```

Security note

I have provided the database, username and passwords above just to show you how to set this up in MySQL. I would highly suggest using encrypted usernames and passwords which should be at least 8 characters or more, and include lowercase and uppercase letters, numbers and punctuation. The longer and more encrypted they are, the harder they are for hackers to try to get into your system.

```
# mysql -u root
```

Type in your root password for MySQL and then hit enter at the next prompt type in Listing 6. Now, you need to import the database structure into your roundcube database. You can copy and paste them into phpMyAdmin or you can use the following command:

```
# cd /usr/local/www/roundcube/SQL
# mysql -u root -p r0undcube < mysql.initial.sql
```

Once you've created the database, you need to run the following commands:

```
# cd /usr/local/www/roundcube/config
# cp db.inc.php.dist db.inc.php
# cp main.inc.php.dist main.inc.php
# vi db.inc.php
```

Now we will want to open `db.inc.php` and change the database setting:

```
$rcmail_config['db_dsnw'] =
'mysql://rcub3:mypass@localhost/r0undcube';
```

Now, you want to edit `main.inc.php` file and change the mailhost setting with your IMAP server address.

```
$rcmail_config['default_host'] = 'ssl://x.x.x.x';
  (replace x.x.x.x with your IP address)
$rcmail_config['default_port'] = 993;
```

Congratulations you have installed roundcube on your server. You can access your roundcube webmail at `http://localhost/roundcube/` (You can change your localhost to your hostname, domain name or IP of the qmail server).

You can now login with your username and password on your IMAP server.

If you are converting from Courier IMAP to dovecot you will want to run the following settings to convert the accounts so the migration will be seamless to your webmail clients.

```
# cd /usr/local/bin
# wget http://www.dovecot.org/tools/courier-dovecot-migrate.pl
# chmod 0700 courier-dovecot-migrate.pl
# cd ~vpopmail/domains
```

This command will test the migrate command before you run it “live”

```
# /usr/local/bin/courier-dovecot-migrate.pl -recursive
```

This command will run the conversion live

```
# /usr/local/bin/courier-dovecot-migrate.pl --recursive
                    --convert
```

Summary

At this point you have installed dovecot and roundcube and configured them to communicate via SSL and you should be able to login to roundcube using dovecot as your IMAP server.

WILLIAM OLSON

William Olson has been working with qmail since 1998. He is fluent in Windows and FreeBSD hardware and software and is currently an IT Admin for a distributor in Maine.

FreeBSD

Unattended Installation of Servers

This article tries to show how to manage an important (in terms of size) computing park when talking about the unattended installation and upgrade of FreeBSD servers.

What you will learn…

- How to manage the version of FreeBSD running on you're machines
- The provisioning or upgrading of each machine in an advantageous way

What you should know…

- What PXE means and how it behaves
- Some advanced FreeBSD admin skills

I think this topic is very interesting because it will help you deploy fast provisioning and upgrading of servers (without the need to go through the whole process: buildworld, mergemaster, etc.), having a homogeneous implementation of FreeBSD. For me, particularly, this has been extremely useful in several situations but mainly when I needed to perform an extremely fast upgrade and come back on-line in terms of minutes.

Previous to building our unattended installation system, we need to check if the downloadable ISO images are up to being used for installing servers in an unattended way. I'm saying this, basically, because FreeBSD's generated CDs and its content (packages, sources, doc…) are done per release; after that happens and until the next release ISO images are out, they provide you patches for your installation (in binary or source format) for fixing bugs, but IMHO you can get better, faster and more customizable results if you proceed the way described here.

So first of all, we're going to create our own patched CD images (*never with unnecessary changes* but yes with the *code properly patched and bugs fixed at the time of iso creation*) and later will build our unattended installation server.

This short introduction has tried to allow people who have started reading this article to know whether they are interested in this topic or not. I assume people with big computing parks will be interested on continuing this reading.

Step 1. Building our own Release

First of all, I'm going to clarify one aspect: the main reason because I'm going to generate a release in this ar-

Listing 1. *SVN exporting content*

```
cd /datamountpoint/

svn export svn://svn.FreeBSD.org/base/releng/9.1 src_releng91

svn export svn://svn.freebsd.org/ports/head ports_releng91

svn export svn://svn.freebsd.org/doc/release/9.1.0/ doc_releng91
```

Listing 2. *Diff of changes applied in mkisoimages.sh*

```
 diff -u mkisoimages.sh-unmodified mkisoimages.sh
--- mkisoimages.sh-unmodified  2013-01-12 20:06:42.000000000 +0100
+++ mkisoimages.sh  2012-12-30 10:36:09.000000000 +0100
@@ -39,6 +39,6 @@
 LABEL=$1; shift
 NAME=$1; shift

-echo “/dev/iso9660/`echo $LABEL | tr ‘[:lower:]’ ‘[:upper:]’` / cd9660 ro 0 0” > $1/etc/fstab
+## echo “/dev/iso9660/`echo $LABEL | tr ‘[:lower:]’ ‘[:upper:]’` / cd9660 ro 0 0” > $1/etc/fstab
 makefs -t cd9660 $bootable -o rockridge -o label=$LABEL $NAME $*
-rm $1/etc/fstab
+## rm $1/etc/fstab
```

Listing 3. *Diff of Makefile.sysinstall*

```
diff -u Makefile.sysinstall Makefile.sysinstall-modified
--- Makefile.sysinstall 2012-10-24 04:20:07.000000000 +0200
+++ Makefile.sysinstall-modified   2013-01-12 20:12:19.000000000 +0100
@@ -1,4 +1,4 @@
-# $FreeBSD$
+# $FreeBSD: release/Makefile.sysinstall 241979 2012-10-24 02:20:07Z kensmith $
 #
 # make release [BUILDNAME=somename] CHROOTDIR=/some/dir CVSROOT=/cvs/dir \
 #      [RELEASETAG=tag] [SVNROOT=svn://svn.freebsd.org/base] \
@@ -1149,21 +1149,11 @@
        FreeBSD_Install \
        ${CD}/FreeBSD-${BUILDNAME}-${TARGET}-disc1.iso     {CD_DISC1} \
        ${CD_DISC1_PKGS}
-  @sh ${.CURDIR}/${TARGET}/mkisoimages.sh \
-      FreeBSD_Packages \
-      ${CD}/FreeBSD-${BUILDNAME}-${TARGET}-disc2.iso ${CD_DISC2} \
-      ${CD_DISC2_PKGS}
 .if defined(MAKE_DVD)
-  @sh ${.CURDIR}/${TARGET}/mkisoimages.sh ${BOOTABLE} \
-      FreeBSD_Install \
-      ${CD}/FreeBSD-${BUILDNAME}-${TARGET}-dvd1.iso ${CD_DVD1} \
-      ${CD_DVD1_PKGS}
+  @echo “No DVD1 needed...”
 .endif
 .if !defined(NODOC)
-  @sh ${.CURDIR}/${TARGET}/mkisoimages.sh \
-      FreeBSD_Documentation \
-      ${CD}/FreeBSD-${BUILDNAME}-${TARGET}-disc3.iso ${CD_DOCS} \
-      ${CD_DOCS_PKGS}
+  @echo “No doc iso image needed...”
 .endif
 .if defined(SEPARATE_LIVEFS)
   @sh ${.CURDIR}/${TARGET}/mkisoimages.sh ${BOOTABLE} \
```

ticle. This is, that it's extremely important for you to being able to generate you're own upgraded or customized iso images of the svn branch/tag you need to work with. You could need customizing them for using Sysinstall instead of BSDinstaller or just for having an updated medium for deploying unattended installation services like the following one (or for upgrading it), or just for using this ISOs with you're managed servers, instead of using a FreeBSD image with some bugs discovered previous to a new release launch (and so, the new upgraded isos are not able to be downloaded from *ftp.freebsd.org*).

So, as I advanced before, at present, the new default FreeBSD installer (BSDinstaller), is undergoing an improvement process and does not support performing an installation in a similar way to Sysinstall with install.cfg config file, so I'm going to generate a custom release for obtaining ISO images with Sysinstall for now..

Let's grab an ISO image from *ftp.freebsd.org* for setting up our release generation machine. For example, fetch or wget: *ftp://ftp.freebsd.org/pub/FreeBSD/releases/ISO-IMAGES/9.1/FreeBSD-9.1-RELEASE-amd64-disc1.iso*.

We will proceed with a normal installation, *but without installing sources (just lib32 and ports)*. After having booted our installed system, we will continue on to create our source directories:

```
mkdir -p /usr/src
mkdir -p /datamountpoint/
```

Let's move into the directory in which we're going to 'svn export' the RELENG_9_1 needed data as can be seen in Listing 1.

Now, we should have an up to date collection of sources, ports and documentation, for building our release and ISO images. I'm assuming we're creating a release for a 64 bit capable machine. So, for our purpose, move into the amd64 src directory:

```
cd /datamountpoint/src_releng91/release/amd64
```

Let's continue by slightly modifying mkisoimages.sh as I'm going to describe in Listing 2.

So now, let's move backwards one level and let's modify `Makefile.sysinstall` in order to end up like in Listing 3.

So, at this point we have the fresh code recently downloaded from FreeBSD's subversion properly patched for generating our customized release that uses Sysinstall as the installer.

Now, we should copy all the content to `/usr/src`. So, we could do:

```
rsync -av  /datamountpoint/src_releng91/ /usr/src/
```

And the following step is to *buildworld* in order to be able to generate release later.

```
cd /usr/src
make buildworld
```

Now let's launch the release generation make as shown in Listing 4.

NOTE

If instead of generating the release specifying the makefile `Makefile.sysinstall`, and without having applied previous changes, we would obtain a fresh release with the new installer (BSDinstaller).

If all went OK, We will have the ISO images can be seen in Listing 5.

Step 2. Installation of the FreeBSD Unattended Installation Server

Let's continue by performing a new server installation (our unattended installing server) with our recently created ISO named *FreeBSD********-SNAP-amd61-disc1.iso*. We will select a Standard installation, with a custom distribution set (please, select Custom) which contains the following parts: base, kernels (all), info, lib32, man, src, ports, and local. *Ensure you say to use cd as media (as source)*.

After the installation process, let's install a dhcpd server which will be the ip allocator in our PXE server. We will use isc-dhcp41-server. For building:

```
cd /usr/ports/net/isc-dhcp41-server

(let's unselect all options)

make install clean
```

Now, let's configure our `dhcpd` server by setting `/usr/local/etc/dhcpd.conf` as in Listing 6.

At this point, we don't need to launch the `dhcpd` server. Now, let's configure `/etc/inetd.conf` in order to enable tftpd. We need to make the tftp line appear like this:

```
 (in a single line)
tftp dgram udp wait root /usr/libexec/tftpd tftpd -l -s /
                    datamountpoint/netboot/freebsd91
```

We will use NFS as our install media (data source) for Sysintall on the unattended installations. So `/etc/exports` in our pxe server should look like this:

Listing 4. *Making release*

```
cd release

make -f Makefile.sysinstall release CHROOTDIR=/datamountpoint/release_generation EXTSRCDIR=/datamountpoint/src_releng91
                    EXTPORTSDIR=/datamountpoint/ports_releng91 EXTDOCDIR=/datamountpoint/doc_releng91 MAKE_ISOS=1
```

Listing 5. *Recently generated ISO images of the new release*

```
ls -la /datamountpoint/release_generation/R/cdrom/

-rw-r--r--   1 root  wheel   67524608 Dec 30 17:47 FreeBSD-9.1-20121230-SNAP-amd64-bootonly.iso
-rw-r--r--   1 root  wheel  508581888 Dec 30 17:48 FreeBSD-9.1-20121230-SNAP-amd64-disc1.iso
-rw-r--r--   1 root  wheel        253 Dec 30 17:49 FreeBSD-9.1-20121230-SNAP-amd64-iso.CHECKSUM.MD5
-rw-r--r--   1 root  wheel        358 Dec 30 17:49 FreeBSD-9.1-20121230-SNAP-amd64-iso.CHECKSUM.SHA256
-rw-r--r--   1 root  wheel  422400000 Dec 30 17:49 FreeBSD-9.1-20121230-SNAP-amd64-livefs.iso
drwxr-xr-x   3 root  wheel        512 Dec 30 17:47 bootonly
drwxr-xr-x   4 root  wheel        512 Dec 30 17:47 disc1
drwxr-xr-x   2 root  wheel        512 Dec 30 17:47 disc2
drwxr-xr-x   2 root  wheel        512 Dec 30 17:47 docs
drwxr-xr-x  18 root  wheel       1024 Dec 30 17:47 dvd1
drwxr-xr-x  17 root  wheel        512 Dec 30 17:47 livefs
```

Listing 6. *Configuracion of /usr/local/etc/dhcpd.conf*

```
allow booting;
allow bootp;
authoritative;
option domain-name "freebsdpxe.sarenet.es";
option subnet-mask 255.255.255.0;
default-lease-time 600;
max-lease-time 7200;
ddns-update-style none;
log-facility local7;
local-address 10.0.0.1;
subnet 10.0.0.0 netmask 255.255.255.0 {
range 10.0.0.70 10.0.0.80;
next-server 10.0.0.1;
filename "boot/pxeboot";
option root-path "/damamountpoint/netboot/freebsd91";
}
```

Listing 7. */etc/rc.conf.*

```
## fxp0 is the PXE-Boot interface
ifconfig_fxp0="inet 10.0.0.1 netmask 255.255.255.0"
## PXE services
dhcpd_enable="YES"
dhcpd_ifaces="fxp0" (serve dhcp leases just using this interface)
inetd_enable="YES"
rpcbind_enable="yes"
mountd_enable="yes"
nfs_server_enable="yes"
```

```
/datamountpoint -alldirs,ro -network 10.0.0 -mask 255.255.255.0
```

Yes, read-only... nothing should be written to our NFS exported content. Now let's populate `/datamountpoint/netboot/freebsd91` with our recently generated release's disc1:

```
tar -C /datamountpoint/netboot/freebsd91 -pxvf  FreeBSD-
                    9.1-20121230-SNAP-amd64-disc1.iso
```

We should now set the loader.conf for our unattended installations service properly in `/datamountpoint/netboot/freebsd91/boot/loader.conf`:

```
mfsroot_load="YES"
mfsroot_type="mfs_root"
mfsroot_name="/boot/mfsroot"
vfs.root.mountfrom="ufs:/dev/md0"
```

Let's follow by decompressing what will become the mfsroot of our netbooted OS and let's copy to it's root (optionally of course) the install.cfg in order to have Sysinstall perform all automated tasks that belongs to it.

NOTE
Install.cfg generation and syntax is beyond the scope of this article and will not be covered.

```
cd /datamountpoint/netboot/freebsd91/boot

gzip -d mfsroot.gz
```

Now, we'll mount the memory disk in order to be able to copy the install.cfg to it's `root` :

```
mkdir /onerandommountpoint

mdconfig -a -t vnode -f
/datamountpoint/netboot/freebsd91/boot/mfsroot -u 200
```

I have used the number 200 but it's really optional, so you could certainly not specify `-u` _ _ _ and it will be attached to the first free kernel memory disk in numerical order.

```
mount /dev/md200 /onerandommountpoint

cp /placewherewehaveourcustominstallcfg/install.cfg /
                    onerandommountpoint
```

Now let's unmount the attached mfsroot.

```
cd

umount /onerandommountpoint
```

After ensuring it's properly unmounted, let's remove our attached kernel memory disk.

```
mdconfig -d -u 200
```

We ensure it's removed, by typing the following command and seeing no output:

```
mdconfig -l
```

Now finally, let's configure our pxe server's `/etc/rc.conf` to launch all services automatically as in Listing 7.

Now important, without doing this the tftp server won't be able to serve pxeboot under

```
/datamountpoint/netboot/freebsd91/boot/
```

and the nfsd would run into troubles too:

```
/datamountpoint/netboot/freebsd91/boot/:
chmod -R 755 /datamountpoint/netboot/freebsd91
```

Conclusion

That's all :). Now, to install or upgrade a new server, you can connect one PXE capable server's Ethernet port with a crossover cable to our recently built PXE server (to it's PXE interface) in order to boot from net and load a FreeBSD installation. I should say too, I just have used this system for upgrading or installing one server at a time, but should work properly too if you want to do this task with some more.

As this is my first article in BSD Magazine many thanks to all for reading it!

EGOITZ AURREKOETXEA AURRE

I'm a sysadmin and system's programmer at Sarenet (www.sarenet.es) and am very proud to be able to contribute with this article, because the Open Source community, documentation and software are basically the most powerful strength in the computing world. I wanted to dedicate this work to all my family, but specially to my grandmother who has very recently leave us. I wanted to give too special thanks to all Sarenet's people, because working with them, is a really nice experience. You could send me you're questions or comments to egoitz@sarenet.es and I'll be very happy and proud of answering or clarifying whatever is needed.

SARENET

Keep FreeBSD Free!
(Donate Today...)

TM

FreeBSD

Programming Primer (Part 2)

In the second part of our series on programming, we will look at configuring our development server, write our first lines of code and commit the changes to a version control system.

What you will learn…

- How to to configure a development environment and write HTML, CSS, PHP and SQL code

What you should know…

- BSD and general PC administration skills

Before we get started, you need to have a FreeBSD test server available with the AMP (Apache / MySQL / PHP) installed. We will also use a version control system (VCS) and a CLI based text editor. I am using FreeBSD 9.0 with VI, MC (for file management) and GIT running under Virtualbox.

Start by installing FreeBSD from DVD and configure networking, user and root accounts, etc. as normal.

Key

- Command line instructions
- Alterations to configuration files
- MySQL prompt / SQL
- HTML / XHTML / PHP code

Part 1. Installing the Software

Step 1

As root, Install mc and git from packages:

```
dev# pkg_add -r mc git
```

Step 2. Upgrade the Ports Tree

```
dev# portsnap fetch && portsnap extract
```

Step 3. Install Apache

```
dev# cd /usr/ports/www/apache22
dev# make install clean
```

Configure `rc.conf` to start Apache on reboot:

```
dev# echo 'apache22_enable="YES"' >> /etc/rc.conf
```

Ensure hosts has your machine name set in `/etc/hosts` otherwise Apache will not start.

```
::1          localhost dev
127.0.0.1       localhost dev
```

Start Apache:

```
dev# /usr/local/etc/rc.d/apache22 start
```

Step 4. Install MySQL

```
dev# cd /usr/ports/databases/mysql55-server
dev# make install clean
```

Start MySQL:

```
dev# echo 'mysql_enable="YES"' >> /etc/rc.conf
```

```
dev# /usr/local/etc/rc.d/mysql-server start
```

Set the MySQL root password and check MySQL works:

```
dev# /usr/local/bin/mysqladmin -u root password 'cms-
                  password'
dev# rehash
dev# mysql -uroot -pcms-password

mysql>\q
```

Step 5. Install PHP5 and Language Extensions

Enable and build apache module. See Figure 1.

```
dev# cd /usr/ports/lang/php5
dev# make config
```

Install PHP5 and the extensions:

```
dev# make install clean
```

Enable mysql and mysqli support. See Figure 2.

```
dev# cd /usr/ports/lang/php5-extensions/
dev# make config
dev# make install clean
```

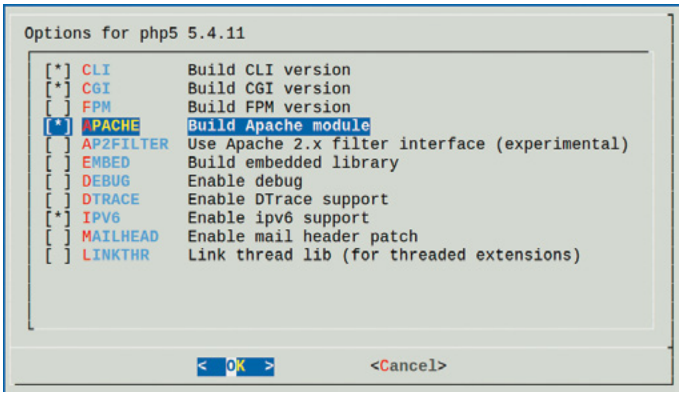
Edit `/usr/local/etc/apache22/httpd.conf` to reflect the following:

```
DirectoryIndex index.html index.xhtml index.php
```

And add the following at the end for PHP support:

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

Copy the php.ini file across:



```
Options for php5 5.4.11

  [*] CLI          Build CLI version
  [*] CGI          Build CGI version
  [ ] FPM          Build FPM version
  [*] APACHE       Build Apache module
  [ ] AP2FILTER    Use Apache 2.x filter interface (experimental)
  [ ] EMBED        Build embedded library
  [ ] DEBUG        Enable debug
  [ ] DTRACE       Enable DTrace support
  [*] IPV6         Enable ipv6 support
  [ ] MAILHEAD     Enable mail header patch
  [ ] LINKTHR      Link thread lib (for threaded extensions)

                 < OK >           <Cancel>
```

Figure 1. *Enabling the Apache module*

```
dev#  cp /usr/local/etc/php.ini-development
  /usr/local/etc/php.ini
```

Restart apache to pick up the new PHP extensions:

```
dev# /usr/local/etc/rc.d/apache22 restart
```

Now we need to setup a development area in our home directory. We will create an account with username dev:

```
dev#  adduser
```

Follow the prompts (the defaults are fine), and give the new user a password. We want to edit / develop as dev, so move the apache data directory across to /`home/de`v and symlink back. That way, Apache can serve the files we create as a non-root user as we can run GIT as a normal user:

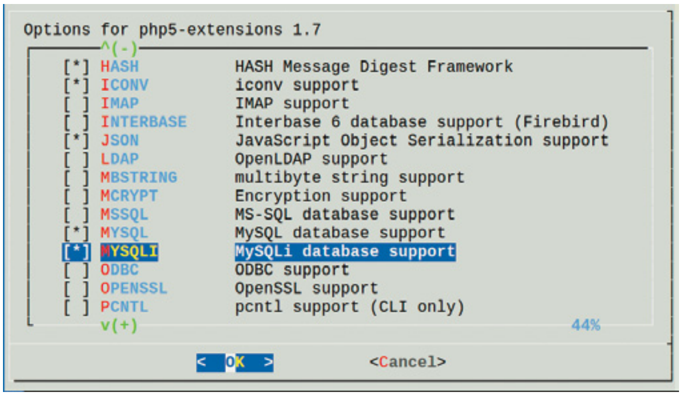
```
dev# mv /usr/local/www/apache22/data/ /home/dev/
dev# chown dev:dev datapwd
dev# ln -s /home/dev/data/ /usr/local/www/apache22/data
dev# cd /home/dev/data
dev# chown dev:dev index.html
dev# /usr/local/etc/rc.d/apache22 restart
```

If you visit your dev box with a browser (*http://youripadress*) you should see the standard Apache "It works!" welcome page.

Part 2. GIT Revision Control and our Test Pages

As a developer, a version control system is an important tool not only to track code changes, but to allow quick recovery from mistakes. Once a file is added and committed to the repository, any errors can be quickly rectified by rolling back to a previous version.

Login with (or su to) the new DEV user account, change to the data directory, and create a new repository then

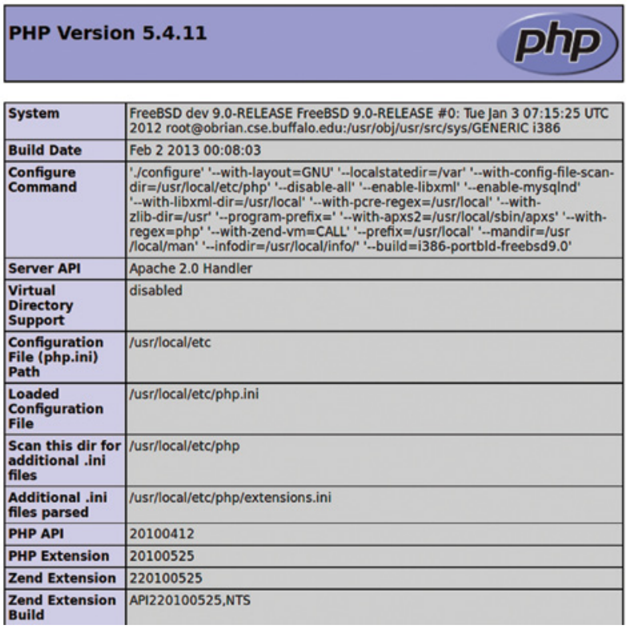


```
Options for php5-extensions 1.7
           ^(-)
  [*] HASH          HASH Message Digest Framework
  [*] ICONV         iconv support
  [ ] IMAP          IMAP support
  [ ] INTERBASE     Interbase 6 database support (Firebird)
  [*] JSON          JavaScript Object Serialization support
  [ ] LDAP          OpenLDAP support
  [ ] MBSTRING      multibyte string support
  [ ] MCRYPT        Encryption support
  [ ] MSSQL         MS-SQL database support
  [*] MYSQL         MySQL database support
  [*] MYSQLI        MySQLi database support
  [ ] ODBC          ODBC support
  [ ] OPENSSL       OpenSSL support
  [ ] PCNTL         pcntl support (CLI only)
           v(+)                                       44%

                 < OK >           <Cancel>
```

Figure 2. *Enabling MySQL support*

commit index.html to it after setting your details. When prompted in the editor, the commit message should be “Initial Load”.

```
dev# su dev
dev# cd /home/dev/data/
dev# git config --global user.name "dev"
dev# git config --global user.email dev@dev
dev# git init
dev# git add *
dev# git commit
```



PHP Version 5.4.11

System	FreeBSD dev 9.0-RELEASE FreeBSD 9.0-RELEASE #0: Tue Jan 3 07:15:25 UTC 2012 root@obrian.cse.buffalo.edu:/usr/obj/usr/src/sys/GENERIC i386
Build Date	Feb 2 2013 00:08:03
Configure Command	'./configure' '--with-layout=GNU' '--localstatedir=/var' '--with-config-file-scan-dir=/usr/local/etc/php' '--disable-all' '--enable-libxml' '--enable-mysqlnd' '--with-libxml-dir=/usr/local' '--with-pcre-regex=/usr/local' '--with-zlib-dir=/usr' '--program-prefix=' '--with-apxs2=/usr/local/sbin/apxs' '--with-regex=php' '--with-zend-vm=CALL' '--prefix=/usr/local' '--mandir=/usr /local/man' '--infodir=/usr/local/info/' '--build=i386-portbld-freebsd9.0'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc
Loaded Configuration File	/usr/local/etc/php.ini
Scan this dir for additional .ini files	/usr/local/etc/php
Additional .ini files parsed	/usr/local/etc/php/extensions.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525,NTS

Figure 3. *PHP enabled*

This will commit the original index.html to the new GIT repository. Edit index.html to reflect Code Listing 1 – “Hello World” is always the first statement written in experimental code. Check with your browser that the page has changed (you may need to press Shift F5 to refresh the cache). Now commit it to the repository:

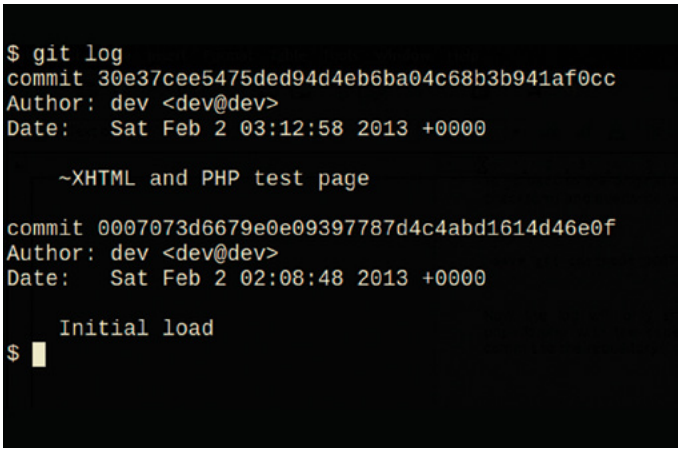
```
dev# git commit -am "First line of HTML"
```

To view the change log:

```
dev# git log
```

Now delete index.html. To recover:

```
dev# git checkout index.html
```



```
$ git log
commit 30e37cee5475ded94d4eb6ba04c68b3b941af0cc
Author: dev <dev@dev>
Date:   Sat Feb 2 03:12:58 2013 +0000

    ~XHTML and PHP test page

commit 0007073d6679e0e09397787d4c4abd1614d46e0f
Author: dev <dev@dev>
Date:   Sat Feb 2 02:08:48 2013 +0000

    Initial load
$ 
```

Figure 4. *Git log*

Listing 1. *The modified Apache index.xhtml*

```
<html><body><h1>Hello World!</h1></body></html>
```

Listing 2. *index.xhtml*

```
<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "DTD/xhtml1- strict.dtd">
  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <title>My first XHTML page</title>
  </head>
  <body>
    <p>Hello world</p>
  </body>
  </html>
```

Listing 3. *phpinfo.php*

```
<?php phpinfo();
```

Further reading

- GIT VCS – *http://githowto.com*
- PHP – *http://php.net*
- W3 Schools – *http://www.w3schools.com*
- W3C – *http://www.w3.org*

To go back to the original Apache file (Where 0007073d is the first 8 digits of the file checksum) and overwrite your changes permanently:

```
dev# git checkout 0007073d
```

Now the log will only show the original file. Create two files *index.xhtml* and *phpinfo.php* with the code from code Listing 2 and 3 respectively and add and commit to the repository:

```
dev# git add *
dev# git commit -am "XHTML and PHP test page "
dev# git log
```

You should see a log file similar to Figure 4.

Listing 1 is a standard XHTML page, with the XML and document type defined. In the next article, we will look at adding CSS and Javascript to this skeleton, but the important point to note here is that all the tags are "balanced" – every opening tag (e.g. `<p>`) has to have a matching closing tag. To view this page, visit *http://youripaddress/index.xhtml* in your browser.

Listing 2 is a very simple PHP command – `phpinfo();` displays all the configuration values, modules loaded etc. available to the PHP interpreter. You should see a page similar to Figure 3 if you visit *http://youripaddress/phpinfo.php*.

In the Next Article

We will look at code structure, program flow and how to embed CSS and Javascript in out pages. We will also start using SQL to dynamically generate pages.

ROB SOMERVILLE

Rob Somerville has been passionate about technology since his early teens. A keen advocate of open systems since the mid eighties, he has worked in many corporate sectors including finance, automotive, airlines, government and media in a variety of roles from technical support, system administrator, developer, systems integrator and IT manager. He has moved on from CP/M and nixie tubes but keeps a soldering iron handy just in case.

Organizational Structure and Culture at FreeBSD

Nothing to Learn from Business Schools ;)

Business Schools teach you during expensive MBA studies how managers should shape the structure and influence the culture of organizations so that they become more innovative.

What you will learn…

- What the organizational environment is like when you work for the FreeBSD project
- What kind of organizational structure and culture is best suited to sustain innovation

What you should know…

- No prerequisites required

Concepts such as natural systems, self-organization, creative swiping, boundary spanning, empowerment… are known to be helpful in making employees more creative. To save you the trouble of paying for costly business courses and based on my view as a committer, this article addresses the question to know whether or not FreeBSD is the right place to develop innovative ideas.

This article will start by briefly describing what the FreeBSD organization looks like using theoretical work from business academics. This in order to give an overview of how the project is organized for those who do not own a *@FreeBSD.org* address. Then the question of innovation sustainability will be approached and the most adequate organizational structures and cultures to support creativity will be presented.

As a bonus and to keep the readers who are not interested in business concepts entertained, some extracts from online conversations between FreeBSD developers were included (nicknames were obfuscated and do not correspond to real FreeBSD developers' login). That way it is still possible to get an insight into FreeBSD's culture without bothering about the gory details.

<tacoz> icanhasarm: sorry I only pay attention half of the time
<tacoz> but keep on talking, it's interesting :)

FreeBSD's from an Academic Perspective

Let's imagine you as a freshly minted Business School graduate, in suit and tie (I know... but let's give it a try) and your boss is asking for a report on the Open Source movement and more specifically the FreeBSD organization. Let's start by describing how the organization is structured.

A Post-industrial Organization

First we can say that FreeBSD presents many characteristics of a *post-industrial organization* (such as described by Bell in 1974 and Huber in 1984), that is a *flexible structure*, a *flattening in hierarchy*, a *blurring of boundaries* between insiders and outsiders and the use of *advanced communication and computing technologies*.

Evolving in a flexible structure means that work units can be created or removed easily and organizational members are able to join several of them at the same time. At FreeBSD there exist many work units such as those dedicated to the ports collection, to file systems, networking, etc., most of them being listed on the wiki home page (*https://wiki.FreeBSD.org*). Developers are free to create new units or participate in any of those, depending only on their own motivation, technical skills and availability.

Regarding the flattening in hierarchy, all FreeBSD committers are at the same hierarchical level except for some people who are given the responsibility to ensure that a certain portion of the system works as expected. This results in the presence of small teams such as the *Release Engineering Team* which is responsible among other things for setting official FreeBSD release schedules, or the *Port Management Team* which ensures that the ports collection is functional, stable and up-to-date. The complete list of project teams with their areas of responsibilities is available at: *http://www.freebsd.org/administration.html*.

Furthermore, a handful of members are elected to be part of a core team. However this core team is not like a board of despotic directors but mainly has administrative responsibilities such as granting access to new developers. Core also intervenes when there are strong disagreements between developers which is very rare from what I have experienced since I joined the project in 2010.

As for the blurring of boundaries between insiders and outsiders this concept will be described in details later on when considering the boundary spanning concept. And finally, the use of advanced communication and computing technologies is not surprising as it is part of FreeBSD's core business and developers rely on several communication channels (emails, irc, wiki, etc.) to stay connected.

```
<canadabald> there is something wrong in the universe
<canadabald> no email in my inbox since midnight
<icanhasarm> well, your mail server is probably dead
<canadabald> checks
<canadabald> mail server is fine
<canadabald> maybe i can just take the day off
<icanhasarm> maybe you already did
<canadabald> i am not in my underwear watching sci-fi, so i don't think i took the day off
```

A Natural System

Then FreeBSD could also be considered as a *natural system* as defined by Scott (1981): a collectivity 'whose participants share a common interest in the survival of the system and who engage in collective activities, informally structured, to secure this end'. Such a natural system is characterized by its *informal structure* and a *variety of interests* which leads to a *plurality of goals* within the organization. Indeed, some developers at FreeBSD prefer maintaining the ports tree, others writing documentation or hacking on the base system itself.

```
<canadabald> icanhasarm: you want pictures of me, in my underwear, hacking freebsd while watching sci-fi?
<icanhasarm> do you happen to hack freebsd ?
<canadabald> icanhasarm: you know i don't really, i just hack Makefiles for ports
<icanhasarm> oh of course, that
```

This led to the division into three different kinds of *'commit bits'* (ports, doc, src), depending on the area of FreeBSD you are given the rights to apply some changes to. However there is no strict rule here and you could also commit in the ports tree even if you are a documentation committer, as long as your patch was reviewed by developers in the area for which you miss the commit bit.

It is also interesting to note that from a strategic standpoint, natural systems tend to *evolve and adapt* depending on changes in both the internal and external environment rather than build upon strict plans and strategies.

A Missionary Structure

And last, Mintzberg would define FreeBSD as a *missionary organization* (Mintzberg, 1979) that is having *little planning and control*, being *fully decentralized*, and using the standardization of *norms as the coordinating mechanism*. At FreeBSD those norms come either from clearly written rules such as style(9) manpage that explains how to format code, or from long lived traditions such as how to format ports commit messages. Such norms allow to constrain developers' behavior within certain pre-defined limits which is a way to control without the need for any hierarchy. The decentralization is also an aspect of FreeBSD's structure with about 400 developers (list available at *http://www.freebsd.org/doc/en/articles/contributors/staff-committers.html*) spread around the world.

Summary: FreeBSD traits

From an academic standpoint we have seen that FreeBSD exhibited the characteristics of a post-industrial and a missionary organization as well as a natural system, that is:

- flexible and informal structure
- little planning and control, no strict strategy, but focus on adaptation instead
- flat hierarchy
- full decentralization
- norms as a mean of coordination and control
- permeable boundaries between insiders and outsiders
- variety of interests and plurality of pursued goals
- use of advanced communication and computing technologies

Now that FreeBSD's characteristics were highlighted, let's compare them with the best practices recommended by Business Schools to bring creativity and sustain innovation in organizations.

Sustaining Innovation: Which Organizational Structure to set up?

A self-organization

In an attempt to become more responsive and creative, organizations often invest in improvement and incremental change programs which usually have exotic names such as kaizen, six-sigma, etc. But more radical approaches exist and the one of interest here is the *self-organization* which, unlike the above-mentioned strategies which focus on procedures and processes, focuses on people as the source for sustainable organizational innovation.

Self-organization is a form of organization within which staff are empowered with absolute trust to organize their day-to-day work in a professional manner. Such a structure advocates creativity through participative self-organization. You believe it does not exist in real life? Well Semler (1994) did it in his Brazilian company 'Semco' where he abolished most rules, norms and procedures. Financial information is available to all, employees are able to appoint and appraise their managers, to set their own working hours, titles, salaries, expenses and (within limits) share of the profits!

> *“Semco’s standard policy is no policy. Many companies have entire departments that generate mountains of paperwork trying to control their employees. Take travel. At Semco, we want our people to spend whatever they think they should, as if they were taking a trip on their own, with their own money. If we’re afraid to let people decide in which section of the plane to sit or how many stars their hotel should have, we shouldn’t be sending them abroad to do business in our name, should we?”*
>
> *– Semler, 1993*

Semler’s philosophy is to maximize worker participation, decision making and public information, and to minimize management control procedures that can inhibit creativity. And at a time when companies try to instill a few values to be respected by all its employees, Semco's success (and we saw the same applies to FreeBSD) is based on the promotion of autonomy and diversity.

<canadabald> man, waiting for chromium to build is like waiting for a frenchman to win a war
<icanhasarm> not fair, you know it will fail in the end
<canadabald> touche!
<canadabald> if only the french had a way of conveying the sentiment of touche...

Boundary Spanning

Boundary spanning (Aldrich and Herker, 1977) takes place when an employee brings information to the organization from the outside, by networking across traditional organizational borders. It was demonstrated that statistically, the more boundary spanners an organization has the higher its degree of innovativeness (Newell and Clark, 1990). This result would imply that innovation requires an ability to identify and adopt the most efficient technologies and know-how from sources outside the organization. This is an ability owned by boundary spanners who are active in various networks and are able to keep pace with the latest developments in different areas.

At FreeBSD we could say that almost all developers are boundary spanners as most of them work in other contexts as part of their official job. This gives the opportunity to discuss and exchange ideas with other passionate engineers and facilitate inward flows of valuable information.

<icanhasarm> I have a trick, I have a windows desktop
<icanhasarm> it's so cool because you can use communicator
<icanhasarm> and do you know what communicator have? ANIMATED SMILEYS
<bsdng> wow like caramail chat !
<icanhasarm> a bit, except they are professionnal smileys
<icanhasarm> because communicator is a serious tool
<icanhasarm> look how professionnal it looks

Virtual Teams

Virtual teams are composed of members based at different locations and connected electronically to each other. Young (1998) states that such teams are a powerful way of working and the new production unit of knowledge and innovation. In theory it allows organizations to work continuously on projects over 24 hours thanks to the span of members over multiple time zones.

<icanhasarm> I wrote almost 50 LoC today, that's certainly enough
<tacoz> I wish I had that much productivity

But the reality seems a bit less glamour and Storey and Salaman (2005) found few actual examples of successful implementation of such teams. However, FreeBSD is a living proof that virtual teams can bring great success to an organization. With developers spread all around the world the project never stops and communications are ongoing 24/7 on irc channels.

Maybe the biggest constraint for commercial organization when they try to set up virtual teams is that, as noticed by Young (1998): 'the notion of control goes out of the window, along with management in its strict sense'. And control is not something managers are willing to abandon, unlike within the FreeBSD project where the hierarchy is flat.

Summary: Organizational Maturity

As for people, it seems that organizations exhibit different levels of maturity. Argyris (1957) argued that traditional organizations (having hierarchical layers with chain of commands and lack of delegated authority, task specification, etc) keep people immature. On the contrary, more radical self-organizing forms of collectivities such as FreeBSD are expected to provide more lively, and creative places to work.

<icanhasarm> moin tacoz
<tacoz> hey icanhasarm, wanna help me procrastinate?
<icanhasarm> I don't know I'm kinda tired

Based on Argyris' cretiera we could assert that FreeBSD is a very mature organization as depicted in Figure 1.

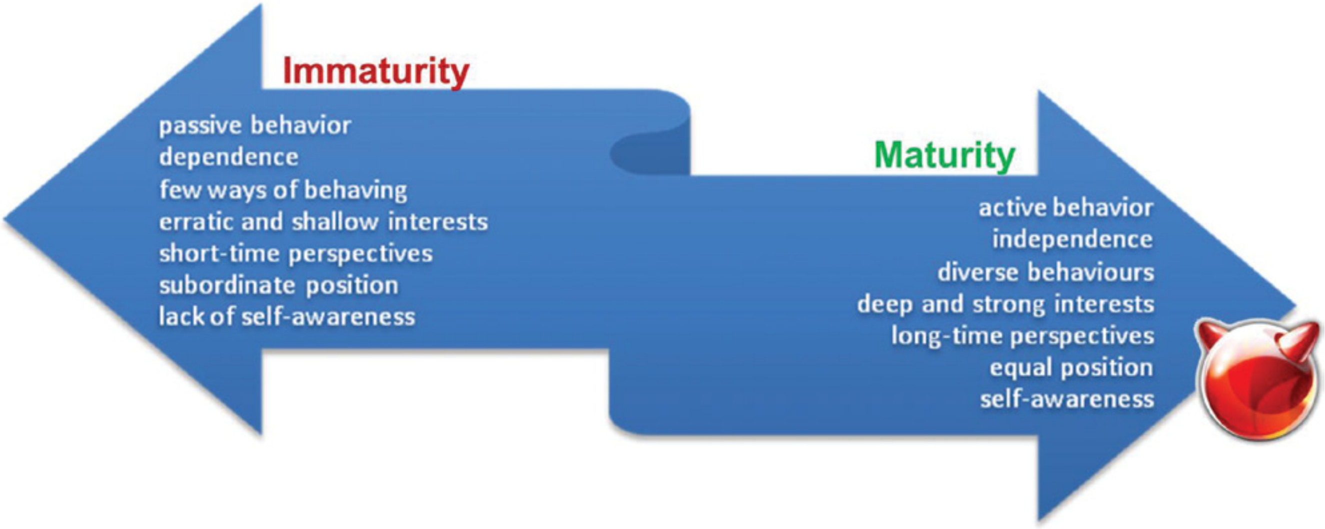


Figure 1. *Organizational maturity continuum*

As an exercise left for the reader it would be interesting to evaluate where they believe their own organization lies within this maturity continuum.

Sustaining Innovation: is Organizational Culture Involved?

Empowerment

The principal aim of empowerment is to increase the organization's flexibility and speed of response, but it could also lead to a more cooperative, committed and proactive workforce which helps in sustaining innovation. There exists an empowerment continuum ranging from very limited autonomy, such as at McDonald where employees are only free to use non-standard greetings to welcome customers, to extensive empowerment such as Semco where we have seen that employees could set their own wages and travel budgets!

Marchington et al. (1992) represent this empowerment continuum using a stairway as shown in Figure 2.

Clearly, developers at FreeBSD are at the same empowerment level as Semco: they have a say in all decisions and they control when, on what and with whom they want to work. In other words we find truly self-managing work teams at FreeBSD with almost no limits to their autonomy, which is very rare in practice. Bowen et al. (1992) refer to this as high-involvement work systems and suggest that those can be very effective.

<icanhasarm> hug me I'm famous

<tacoz> I'll hug you if your build succeeds icanhasarm

<icanhasarm> I'll be drunk before it succeeds, I'm afraid

<tacoz> not a bad alternative :)

<icanhasarm> I'd advise you to hug me BEFORE I get drunk, as drunken icanhasarm has an unpredictable behavior

femur runs

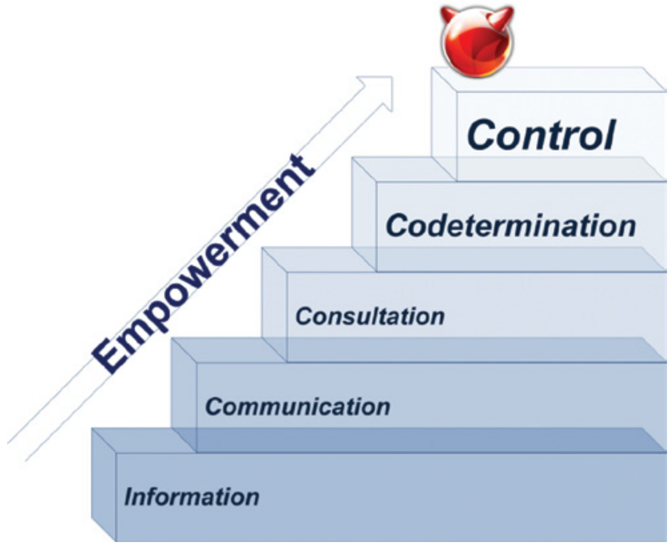


Figure 2. *Stairway to empowerment*

I experienced in some of my previous companies forms of "confined" empowerment with the creation of small and dedicated R&D teams which were given more control upon their work organization (freedom to choose either the research subject or define the planning and set deadlines). This often led to a rise in motivation and creativity amongst team members, but it was nothing compared to what I live within FreeBSD which pushes this concept of empowerment much further.

Motivation and hierarchy of needs

The motivation of creative workers relies on a few ingredients listed by Amabile (1998):

- challenge
- freedom
- resources (time and money)
- work-group features
- supervisory encouragement
- organizational support

Those of you who are able to say they find all this at their work place, lucky you! But I seriously doubt it. At FreeBSD we have almost all of those ingredients: challenging tasks and total freedom (the sky is the limit, you can choose whatever subject to work on), work-group features (again, you can choose whichever team you want to join or create your own work group). Regarding supervisory encouragement you always get a warm welcome from your mentors when you join the project, but after your mentoring period it's up to you to gather momentum on the subject you are working on. For the organizational support you have access to the project's infrastructure and regarding financial resources the FreeBSD Foundation could provide you with grants to attend conferences or to sponsor your work. However, budgets are quite limited compared to commercial organizations similar in size as it relies on public donations (go visit *http://www.freebsdfoundation.org/* to support the project!).

Also useful in our context is Maslow's framework related to motivation, famously known as the pyramid of needs (Maslow, 1954) and displayed in Figure 3. It would make sense to say that innovation is brought by knowledge workers who are looking for motivation from the highest levels of the pyramid, that is self-actualization. And FreeBSD is a good place to find self-actualization under the form of challenging projects within which it is natural to learn at a high level from other experienced engineers.

Hakin9
OPEN
HOW TO BRUTE-FORCE DRUPAL6 LOGIN PAGES?
THE RISE AND FALL OF MEGAUPLOAD.COM AND KIM DOTCOM
BACKTRACK LINUX – HOW TO DITCH THE MENU AND BALL FROM THE COMMAND LINE?
AN INTERVIEW WITH WILLIAM F. SLATER, III
STIMULATING APPLICATION LAYER DENIAL OF SERVICE ATTACKS WITH SLOWHTTPTEST
Vol.1 No.2
Issue 2/2013 (2) February ISSN: 1733-7186
PLUS
HOW TO EXPLORE THE IPV6 ATTACK SURFACE WITH METASPLOIT?

<femur> icanhasarm: need your knowledge
<icanhasarm> femur: you can't get aids from goats
<femur> icanhasarm: need your knowledge, take 2

Note that lower needs could also be fulfilled at FreeBSD especially social needs with the ability to participate in team work and exchange ideas during meetings and conferences.

Creative swiping and the NIH syndrom

Tom Peters invented the term 'creative swiping' to describe the practice of borrowing good ideas from other companies (Peters, 1987). He said: “Put NIH (Not Invented Here) behind you – and learn to copy (with unique adaptation/enhancement) from the best!”.

<icanhasarm> I wonder if they will advertise win 8
<icanhasarm> LOOK WE PUT A DUMBED DOWN TABLET INTERFACE ON A PC OS HOW BRILLIANT
<tacoz> isn't that what's freebsd is trying to do?
<icanhasarm> it is
<icanhasarm> we run great on PDP11-based tablets

Such creative swiping greatly improves an organization's rate of innovation as advocated by Procter & Gamble's Huston and Sakkab (2006): 'We needed to move the company’s attitude from resistance to innovations “not invented here” to enthusiasm for those “proudly found elsewhere.” And we needed to change how we defined, and perceived, our R&D organization–from 7,500 people inside to 7,500 *plus* 1.5 million outside, with a permeable boundary between them'.

I can think of several examples of creative swiping at FreeBSD: pf (taken from OpenBSD), BSM Audit, MAC, DTrace, zfs (adapted from Sun/OpenSolaris/Illumos). ZFS for instance was imported from the OpenSolaris project and improvements were made to it such as the TRIM support implementation by pjd@.

Playfulness

Innovation guru Michael Schrage states that 'innovation is less the product of how innovators think than a by-product of how they behave' (Schrage, 2000). He believes 'serious play' is not an oxymoron and assures that 'you can’t be a serious innovator unless you are willing and able to play'. What there is to understand here is that the playfulness behavior exhibited by creative people refers to an ability to be mentally flexible. While people with a rigid view of the world might feel uncomfortable approaching ambiguous problems, playful minds more easily accept ambiguity and are able to stand outside the mainstream of thoughts, leading to creative thinking. Indeed, mental

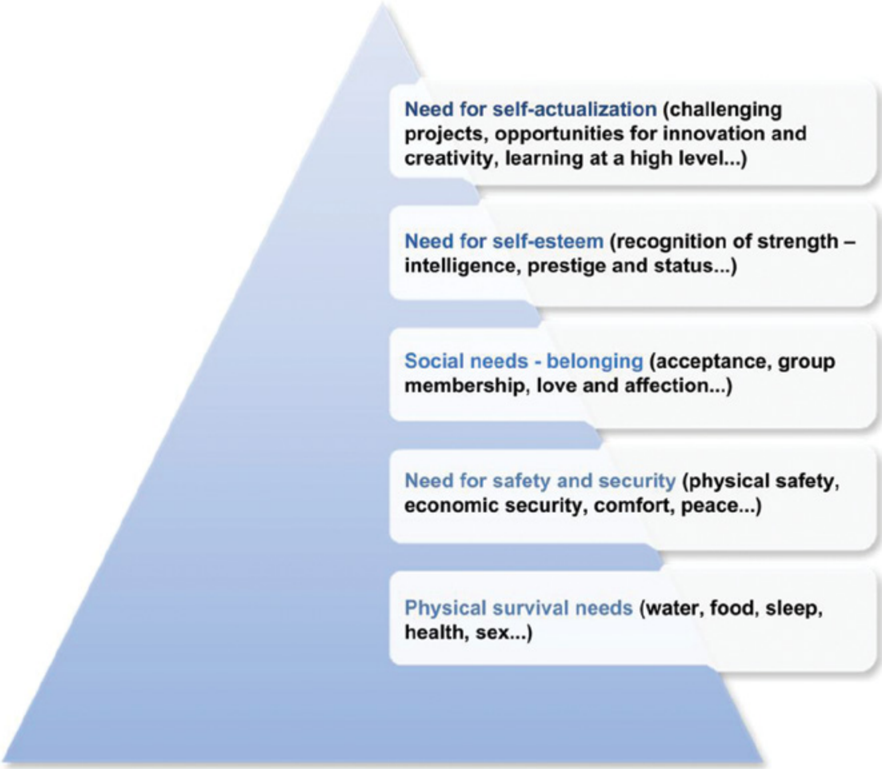

Need for self-actualization (challenging projects, opportunities for innovation and creativity, learning at a high level...)
Need for self-esteem (recognition of strength – intelligence, prestige and status...)
Social needs - belonging (acceptance, group membership, love and affection...)
Need for safety and security (physical safety, economic security, comfort, peace...)
Physical survival needs (water, food, sleep, health, sex...)

Figure 3. *Pyramid of needs*

flexibility implies a mind which is able to tolerate ambiguity and switch perspectives, and helps creative people making sense of conflicting viewpoints.

Coming back to FreeBSD, the irc extracts reproduced in this article reflect the fact that playfulness is part of FreeBSD's culture. This contributes to make developers comfortable and willing to take risks, pushing their ideas without fearing to get systematically blamed by a cohort of hostile colleagues. It also make people want to spend more time within this playful environment some developers call the zoo :)

<canadabald > good afternoon all

<femur> hey ho canadabald

<femur> canadabald: should I fix that typo?

<icanhasarm> canadabald: would you mind answering femur, instead of pretending your a klingon warrior ?

<femur> tnx icanhasarm appreciate it

<femur> but but i can't be in klingon

<femur> for i hate star wars

<icanhasarm> I thought klingons were twilight bad guys ?

<femur> no that's king kong

icanhasarm totally lost with all those teen movies

<femur> lost is not a teen movie

<femur> btw I've committed the typo fix no matther what he says

<icanhasarm> canadabald: go on then, pretend you're a klingon warrior, even if klingons ain't nothing but bit****, we do not need you anymore

Conclusion

We have seen the ingredients necessary to sustain innovation in an organizational context. I tried to show using a few theoretical models that FreeBSD is a great place to work and to nurture innovative ideas. Do you recognize some of those ingredients in your own company? If not, why not getting inspired by what is done at FreeBSD and make some propositions to your managers? And if you are still an outsider, why not trying to cross our permeable boundaries and become part of the FreeBSD project? This way you could experience a work environment you could hardly find anywhere else. So, see you soon at the zoo!

<femur> we are a bunch of false friends who hate each other but have no real friends and no real life, so that's what we got

<icanhasarm> I HAVE A REAL FRIEND HE4S CALLED KEKE AND HE LIVES IN MY HEAD

FREDERIC CULOT

After a PhD in Computer Modeling applied to Physics the author has been working as a software engineer and IT project manager for the past eight years. During his spare time he pursues business studies and should complete an MBA by the end of this year, and he is also a committer for the FreeBSD project with more than 1,300 commits to the ports tree.

References

- Aldrich, H., Herker, D., Boundary Spanning Roles and Organization Structure, The Academy of Management Review, Vol. 2, No. 2, pp. 217-230, 1977.
- Amabile, T., How to kill creativity, Harvard Business Review, September–October, pp. 77–87, 1998.
- Argyris, C., Personality and Organization, New York, Harper & Row, 1957.
- Bell, D., The Coming of Post-Industrial Society, New York, Harper Colophon Books, 1974.
- Bowen, D. E., Lawler, E. E. The empowerment of service workers: What, why, how, and when, Sloan Management Review, 33, 3, 1992.
- Huber, G. P., The Nature and Design of Post-Industrial Organizations, Management Science, Vol. 30, Issue 8, p928-951, 1984.
- Huston, L., Sakkab, N., Connect And Develop, Harvard Business Review, Vol. 84, Issue 3, p58-66, 2006.
- Maslow, A. H., Motivation and personality, New York, Harper and Row, 1954.
- Marchington, M., Goodman, J., Wilkinson, A., Acker, P. New developments in employee relations, Research Series, No. 2, Sheffield, UK: Employment Department, 1992.
- Mintzberg, H., The Structuring of Organisations, Englewood Cliffs, NJ, Prentice Hall, 1979.
- Newell, S., Clark, P. A., The importance of extraorganisational networks in the diffusion and appropriation of new technologies: the role of professional associations in the United States and Britain, Knowledge: Creation, Diffusion, Utilisation, Vol. 12, No. 2, pp. 199–211, 1990.
- Peters, T., Thriving on Chaos, London, Macmillan, 1987.
- Schrage, M., Serious Play: How the World's Best Companies Simulate to Innovate, Boston, MA, Harvard Business School Press, 2000.
- Scott, W. R., Organizations: Rational, natural and open systems, Englewood Cliffs, Prentice Hall, 1981.
- Semler, R, Why my former employees still work for me, Harvard Business Review, Vol. 72, No. 1, pp. 64-71, 1994.
- Storey, J., Salaman, G., Managers of Innovation: Insights into Making Innovation Happen, Oxford, Blackwell, 2005.
- Young, R., Virtual teams: the wide-awake club, People Management, Vol. 4, Issue 3, pp. 46-49, 1998.

MAGAZINE BSD

In the next issue:

- Kernel Panic

Continuation of the series on:

- FreeBSD Programming Primer
- TrustedBSD

Next issue is coming in March!

iXsystems®

Headquarters:
San Jose, CA

855.GREP.4.IX | Contact Us

99% Compatibility

online now...

IXSYSTEMS AND YOU ARE THE PERFECT MATCH

SHARED INTERESTS

☑ Enterprise Storage Solutions

☑ Personalized Customer Service

☑ Bold New Information Technology

I'm a: Storage Reseller

In: The EU

Looking for: Storage Solutions to Sell

- A Technology Partner
- More Technical Experience
- New Business Opportunities

Visit Today!

iXsystems

Technology Partner Seeking Resellers/Integrators for TrueNAS™ Storage Appliance

WWW.IXSYSTEMS.COM/PERFECTMATCH