What's new in NetBSD in 2006 ?

Emmanuel Dreyfus

October 24, 2006

Abstract

NetBSD is known as a highly portable operating system, but its strengths are not limited to being available on many platforms. NetBSD goals also include security, performance, standards conformance and clean design. Development of innovative features also occurs.

In this paper, we will have a look at the new features that have been integrated into NetBSD this year.

1 NetBSD in the news

1.1 Dead or irrelevant?

Thanks to the numerous and valuable contributions from Slashdot's anonymous coward, we are now all aware that *BSD is dying [1]. While the recurrent Slashdot troll gave us strong warnings about FreeBSD's and OpenBSD's imminent deaths for years, NetBSD was often omitted. Did that mean NetBSD was already dead?

The EuroBSDCon 2005 social event was called "the night of the living dead", in reference to the Slashdot troll. That was an attractive point of view, since it implied that dead projects like the *BSD could be alive and kicking after all. Unfortunately, NetBSD did not show any sign of life that night, as it was even outperformed by DragonFly BSD at the beer drinking contest.

The few people who still remembered an OS called NetBSD were still puzzled about the death of NetBSD: did it occur while nobody was watching? Fortunately, on the 30th of August 2006, one of the NetBSD project founders sent an insightful message to the `netbsd-users@netbsd.org` mailing list [2]. In that message, Charles M. Hannum explained that NetBSD had increasingly become irrelevant. That post was reported by Slashdot, which drove a lot of attention to NetBSD. Charles shortly followed up with an interview at OnLAMP.com [3], entitled "Confessions of a Recovering NetBSD Zealot".

Thanks to Charles, things were clear: the project was not dead, it was just irrelevant, and every Slashdot reader knew about it. That is not very good news, but at least this had the advantage of showing that NetBSD was alive enough to upset someone and make the Slashdot cover page.

As far as I am aware, no news site tried to investigate Charles' claims by interviewing other NetBSD insiders about the affair. I assume I have to make a few comments on the topic.

In my opinion, Charles has various valid points. Indeed NetBSD could be better managed. It could also have more features, fewer bugs, and be more popular. Unless a project is really dead, people always expect more than what they get. There is always room for improvement.

But Charles' judgement as NetBSD being now irrelevant is just a personal opinion, and not everybody shares it. There is still a lot of work done on the project from several dozens of developers (the activity can be monitored through the `source-changes@netbsd.org` mailing list [4]). Obviously that crowd does not consider NetBSD as irrelevant.

The project is also recruiting new developers at a steady rate of few persons per month. That fresh blood shows that we even have newcomers considering NetBSD as an OS relevant enough to start working on it.

And finally, we still have a lot of users, as we will see in the next section.

Charles' detractors will note that his complaints come after years of inactivity as a NetBSD developer, and at the time the board of the NetBSD foundation decided to evict him [5] because he refused to sign the NetBSD developer agreement [6]. Charles' answer to his detractors is available in the press. This is not a nice story, but the only real bad point I will retain is that we have lost a valuable contributor to the project. That is not the first time such a thing has happened, and it will certainly not be the last one: no open source project can retain all its contributors forever. Let us hope we will not do that too often.

1.2 Bugathons

We have received positive press on the two NetBSD bugathons [7]. These are IRC meetings where NetBSD developers and users meet to work on resolving problem reports together [8].

The two first bugathons were organized by Elad Efrat. They occurred on during September 23rd-24th [9], and October 7th-8th week-ends [10].

Both events were huge successes. According to Elad's reports, the first edition gathered 30 developers and 20 users, and resulted in 270 problem reports (PR) being closed. The second edition gathered 3 times more people and resulted in 310 PRs being closed.

Of course, hundreds of PRs being closed do not mean hundreds of bugs being fixed, as many open PRs are duplicates or obsoletes, or even come with a bug fix that just needs to be reviewed and committed. Still, such events are excellent news for the project, as it means we finally found a way to deal with the never-ending accumulation of open PRs [11].

The other very good point about the bugathons is that it clearly shows that despite the claims about NetBSD's irrelevancy, there is still a strong user community around NetBSD.

I hope we will see many more bugathons, and that neither the user community, nor the developers will get tired of them. The open PR database had gotten way too fat, it was high time to make it slim again!

1.3 Google Summer of Code

This year, NetBSD was involved for the second time in Google Summer of Code [12]. This year, 8 projects were started by students [13]:

Support for journaling for FFS : The Berkeley Fast File System (FFS) is NetBSD's preferred file system. Adding a journaling feature to it would remove the need for long file system checks when the system reboots after a power outage. As disks get bigger and cheaper, the time for a file system check has grown far too long for many users.

Support for MIPS64 : NetBSD runs on a large range of hardware, including MIPS based machines. NetBSD also has all the necessary infrastructure to run on 64 bit processors (such as alpha, sparc64, or amd64), but it lacked the machine-dependent bits for running MIPS processors in 64 bit mode.

PowerPC G5 support in NetBSD : This is about adding the machine-dependent bits to get the PowerPC G5 processor supported by NetBSD.

Improved writing to file system using congestion control : In a multiuser environment, several processes can write at once to the same file system, thus causing congestion. The goal of this project was to establish benchmark tools and to research solutions to file system congestion problems.

TCP ECN support : Explicit Congestion Notification (ECN) is a set of congestion control mechanisms described in RFC 3168 [14]. At the TCP level, it works by having the sender adjust the transmission window size to handle congestion notifications sent by routers. Supporting this feature would enable NetBSD to perform better on overloaded networks (provided the routers also support ECN).

FAST_IPSEC and IPv6 : The original IPsec implementations in *BSD kernels was derived from the KAME project [15]. It exhibited poor scalability, and was unable to take

advantage of specialized hardware accelerators to perform the cryptographic computation. The `FAST_IPSEC` [16] kernel option was created to deal with this issue, but it lacked IPv6 support.

`pkg_install` rewrite for pkgsrc : The NetBSD package collection [17] is based on a set of tools that have evolved over the years. It seems the tools have reached the point where a major cleanup is necessary. The goal of this project was to collect the requirements for pkgsrc tools, and re-implement them based on a new clean design.

Improving the `mbuf` API and implementation : `mbuf` [18] is the infrastructure used by kernel networking code to manage memory. The current programming interface features many pitfalls, and it is easy to write buggy code that makes wrong assumptions. The goal of the project was to clean up the programming interface to make it easier to deal with.

Hubert Feyrer's press release [13] gives us this year's result. There has been some successful stories: PowerPC G5 support, ECN implementation, `FAST_IPSEC` and IPv6, and `mbuf` cleanup project were completed. It is also worth noting that the mbuf cleanup project has opened the way to a zero-copy I/O implementation in NetBSD/Xen. Once completed, this should produce a noticable performance win.

pkgsrc infrastructure rewrite and file system congestion control were not fully completed. While not completely done, the file system congestion control still led to interesting performance improvements. On the pkgsrc front, a paper from Joerg Sonnenberger [19] details the recent changes.

And finally, we had two failed projects: the student in charge of the journalised FFS project simply vanished, and the one in charge of MIPS 64 ran into a health problem that prevented him from completing the work in time.

This year again, we have to thank Google for sponsoring our development. Driving students to NetBSD is an excellent thing for the project, as it means new contributors and new features implemented.

1.4   pkgsrcCon 2006

The third pkgsrcCon took place in University of Paris 7 – Denis Diderot [20]. The goal of this technical conference is to gather developers and users of the NetBSD package system, also known as pkgsrc. Here is the conference program:

- Stoned Elipot, System Administration with pkgsrc [21]
- Joerg Sonnenberger, pkgsrc on DragonFly – or Fighting the Windmills [22]
- Roland Illig, `pkglint`: Static Analyzer For Pkgsrc [23]
- Roland Illig, Why Pkgsrc Sucks [24]
- Emile Heitor, `pkg_select` – So Many Packages, So Few Columns [25]
- Thomas Klausner, Roundtable Discussion: Updating Packages [26]
- Adrian Portelli, pkgsrc security one year on... [27]
- Dieter Baron, Thomas Klausner, `pkg_install` Rewrite [28]
- Johnny Lam, Roadmap for Development [29]

2   A few new exciting features

2.1   Xen

Xen [30] is one of the latest hot topics in the world of virtualization. Virtualization is about running multiple OSes at the same time on the same machine. It makes system management easier, as a virtual machine can be easily cloned or migrated to another real machine. Virtualization also offers easier system debugging, and allows hardware resources such as memory and CPU to be easily shared.

For instance, I use Xen virtual machines to run a virtual network with two hosts and a Network Address Translator (NAT) on the same machine. I use that setup to quickly make regression tests on IPsec-tools-based VPN [31] setups.

Virtualization usually works by featuring a host OS, which holds access to the real hardware, and guest OSes, which see virtual hardware. Early virtualization software worked by catching guest OS access to the hardware through exceptions. This enabled running unmodified versions of the guest OS but had a huge cost in performance.

Xen reached unprecedented levels of performance by requiring guest OSes to be modified. The guest OS is now aware it is running on a virtual machine, and accesses the virtual hardware through a well defined API. That approach removed the costly game of generating hardware exceptions for any hardware access such as a reading data from a disk. In Xen terminology, guest OSes are called domU, while the host OS is called dom0. It is worth mentioning, that domU and dom0 all run on the top the Xen kernel. Xen delegates hardware management to the kernel in the dom0 OS. This approach has the advantage of freeing Xen development from writing drivers.

Performance comparison of Xen versus various competitors is available from the Xen web site, and from a third party research group [32].

NetBSD 3.0 already implemented support for Xen 2.0, both as a domU and a dom0. Recently, Christian Limpach and Manuel Bouyer implemented support for Xen 3.0, both as domU and dom0.

Xen 3.0 has a few interesting new features, including:

- Support for up to 32 way SMP guests.
- Hardware-assisted virtualization (Intel VT-x and AMD-V Pacifica), which allows running unmodified guest OS.
- 64 bit support for the AMD64 architecture (not supported by NetBSD yet).

There has also been a lot of code rewriting behind the scenes, but that is not usually considered an interesting feature.

Finally, it is worth mentioning that benchmarks showed superior disk I/O performance of NetBSD as a dom0, compared to Linux [33].

2.2   iSCSI

iSCSI stands for Internet Small Computer System Interface. It is an encapsulation of the SCSI protocol over TCP/IP, documented in RFC 3720 [34], used for Storage Area Network (SAN).

The basic idea of a SAN is to have file servers exporting disk space as a block device, instead of exporting it as a file system, through protocols such as Unix's Network File System (NFS), Windows' Common Internet File System (CIFS), or Apple's Appleshare File Protocol (AFP). It frees the server from the burden of maintaining a file system, and allows easier storage resource sharing and extension.

iSCSI is a hot topic, because it allows building affordable SANs, based on ubiquitous Ethernet and TCP/IP network infrastructure, whereas SAN have traditionally been using specialized hardware, based for instance on fibre channel.

In iSCSI terminology, there is an iSCSI target, which exports selected storage as a block device, and an iSCSI initiator, which accesses the block device exported by the target. Of course, unless you use some kind of a distributed file system which can be mounted by several OSes at once, there can be only one initiator using a target at a time.

Alistair G. Crooks worked on integrating the iSCSI target support developed at Intel, and published a set of HOW-TOs [35], which explain how to set up NetBSD as an iSCSI target, and how to set up MS Windows XP as an iSCSI initiator that uses it. Alistair also presents a paper on iSCSI at EuroBSDCon 2006 [36].

The iSCSI initiator code in NetBSD is still a work in progress, and so is the support for iSCSI authentication mechanisms.

2.3 The build infrastructure now creates ISO images

NetBSD enjoys a unique build infrastructure, which allows extremely easy cross-building from another OS, or from NetBSD itself [37]. That infrastructure made automatic NetBSD builds for all NetBSD ports [38] not only possible, but even affordable.

There has been a long-term missing item in this auto build machinery: the only bootable media it was able to produce were installation floppy disk images. As today's modern hardware more and more often ships without a floppy disk drive, the lack of automatic bootable ISO image was becoming a concern.

That missing feature has been implemented, thanks to the work of various contributors.

First, Daniel Watt, Walter Deignan, Ryan Gabrys, Alan Perez-Rathke, Ram Vedam, and Luke Mewburn, improved NetBSD's `makefs` [39] utility, to support the ISO 9660 format. The purpose of `makefs` is to allow creation of file system images without the need of root privileges. It was initially developed to create FFS images of the RAM disks used in install kernels, so that this operation could be performed during the build process.

The second step was to actually use that feature in the NetBSD build infrastructure. Alan Barrett did the appropriate changes. Thanks to this work, NetBSD is now able to provide bootable ISO images as part of the regular NetBSD-current auto builds.

For now, the ISO images produced through the auto build machinery just contains the installation program, and not the installation sets (i.e.: `base.tgz`, `comp.tgz`, and so on), leaving the generation of a stand-alone installation ISO images as a future work.

2.4 WPA

NetBSD supports various IEEE 802.11 wireless devices. Unfortunately, we did not support the Wi-Fi Protected Access (WPA) protocol [40], which was a shame, since it only left our users with the alternative of using the Wired Equivalent Privacy (WEP) protocol [41], which is well-known for being insecure [42], or VPN-based setups, which are much more complicated to set up.

Thanks to Steve Woodford and Rui Paulo [43], NetBSD now has support for joining a wireless network protected by WPA. Steve and Rui integrated `hostapd` [44] and `wpa_suplicant` [45] from Jouni Malinen's WPA for Linux project [46]. That software includes WPA and WPA2 support both when acting as an access point and as a client. Advanced access point features such as RADIUS are also supported.

2.5 Bluetooth

Bluetooth [47] is a complete stack of wireless protocols standardized by the IEEE 802.15.1 task group for usage in Personal Area Networks (PAN). It is used for communication between various hand-held devices such as cell phones and PDA, or with devices such as hand-free headsets.

Iain Hibbert worked hard on implementing a complete bluetooth stack on NetBSD [48]. Iain started the work on his own, and was later sponsored by Itronix, Inc.

There is a page on the unofficial NetBSD Wiki [49] that shows bluetooth configuration and usage for using bluetooth Human Interface Devices (HID), serial links, audio headsets, and audio hands free devices.

2.6 UDF

Universal Disk Format (UDF) [50] is a file system designed for storing files on optical media. It is developed by the Optical Storage Technology Association (OSTA), and is also known as the ISO 13346 standard.

UDF is seen as the successor to the ISO 9660 format. It is used in DVDs, but can also be used in CD-ROMs or USB flash memories. As more and more optical disks using this format

appear, not being able to read them was a growing annoyance. Thanks to Reinoud Zandijk's work [51], this problem is now solved.

According to Reinoud, the NetBSD UDF implementation is able to read UDF file systems up to version 2.60 that are found on CD-ROM, CD-R, CD-RW, CD-MRW, DVD-ROM, DVD*R, DVD*RW, DVD+MRW disks, and it should be able to read DVD-RAM, HD-DVD, and BluRay disks. Disks do not need to be closed.

Note that support is currently limited to read-only. Read/write support is still a work in progress.

3 Networking

3.1 CARP

The Common Address Redundancy Protocol (CARP) [52] appeared in OpenBSD as a free alternative to Internet Engineering Task Force (IETF) blessed Virtual Router Redundancy Protocol (VRRP) [53] and Hot Standby Router Protocol (HSRP) [54], which are encumbered by Cisco patents.

CARP allows multiple hosts to share an IP address. The main usage for this feature is to build redundant firewalls, but it can also be used for load balancing.

Liam J. Foy imported OpenBSD's CARP to NetBSD [55].

3.2 Link aggregation

YAMAMOTO Takashi committed his implementation of the IEEE 802.3ad Link Aggregation Control Protocol (LACP) [56]. This allows bonding of several Ethernet interfaces into a single virtual `agr(4)` [57] interface.

The current implementation has a few limitations, see the `agr(4)` [57] man page for details.

3.3 NDIS wrapper

Network Driver Interface Specification (NDIS) is a generic programming interface developed by Microsoft and 3com for network interface drivers [58]. Most, if not all, network device vendors will give away NDIS drivers for the products they sell.

The NDIS interface is well defined and NDIS driver are not supposed to access Windows internals without going through the NDIS interface. That means an NDIS driver designed for Windows could work on another OS, provided that a translation layer is set up between the NDIS interface and the native OS.

This is exactly what the NDIS wrapper project is about. By implementing an NDIS compatibility kernel option, it is possible to use binary drivers built for Windows on other OSes. Of course, that is limited to drivers built for the same processor, which usually means i386 only.

NDIS wrapper was first developed for FreeBSD [59], and later adopted by Linux [60]. Thanks to the work done by Alan Ritter during last year's Google Summer of Code [61], NetBSD now also enjoys that feature.

For more information on how to use it, see the `ndiscvt(8)` [62] man page.

4 Storage and file systems

More features beyond iSCSI and UDF:

4.1 tmpfs

tmpfs is a new memory-based file system, which was designed by Julio M. Merino Vidal as a 2005 Google Summer of Code project [63].

The goal of tmpfs is to replace MFS. The problem with MFS is that it is just FFS hacked to store files in memory instead of on a disk, thus resulting in poor memory usage.

4.2 scan_ffs

Who never erased a partition table by mistake? This error is especially irritating, since your data is still on the disk, but you cannot reach it anymore.

OpenBSD developed a scan_ffs utility to solve that problem. Its purpose is to search the disk for a FFS file system, so that you have an opportunity to reconstruct your partition table and recover access to your data.

Thanks to Juan Romero Pardines, NetBSD now also enjoys the scan_ffs(8) [64] utility. It is worth noting that Juan also added LFS and FFSv2 support to scan_ffs.

4.3 LFS improvements

LFS stands for Log-structured File System [65]. Traditional file systems have been designed with the idea that the hard disk seek time was the bottleneck to I/O performances. This is no longer true if system memory is so big that everything is read from cache, or if the media is not a hard disk.

The idea behind LFS is to write to the disk sequentially, without doing any efforts so that a file can be read sequentially. As files are modified, all changes to the files are saved on disk and mix with each other. LFS write throughput is blazingly fast compared to other file systems. Another interesting feature is the ability to resize the file system while it is mounted.

The LFS implementation used in BSD systems was not maintained enough to remain usable. FreeBSD and OpenBSD eventually removed it. Fortunately, Konrad Schroder stepped in to repair NetBSD LFS and bring it back into a usable state [66].

5 Hardware support

5.1 New ports

NetBSD made a few steps towards total world domination, by adding support for a few more embedded device platforms:

- Atmark Techno Armadillo-9 [67] is a PC/104 form factor embedded device with the same size as a floppy disk. It features an ARM CPU, and a large set of I/O interfaces: USB2, Compact Flash, IDE, Ethernet and VGA. The evbarm port now supports it, thanks to Katsuomi Hamajima's work.
- Also from Atmark Techno, the Armadillo-210 [68] is an extremely small (barely the size of its connectors) ARM based machine, with VGA and Ethernet (supports Power over Ethernet). It has 32 MB of memory and 4 MB of flash. Again, brought to the evbarm port thanks to Katsuomi Hamajima.
- A brand new port, ews4800mips [69]. This brings NetBSD on NEC's EWS4800 workstations. The hard work has been done by UCHIYAMA Yasushi and Izumi Tsutsui.

Garrett d'Amore did a huge amount of work around the evbmips port, to support the following devices:

- the Alchemy Au1550 System-on-a-Chip (SoC) [70] featuring DDR controller, 2 Ethernet interfaces, 4 serial controllers

- Meshcube [71], a tiny cube with wireless Ethernet, RJ45 Ethernet, USB, 32 MB of flash and 64 MB of RAM
- Atheros AR5312, a SoC specialized for Wi-Fi appliances which is found in various wireless devices, such as Linksys WAP55AG 2.0 and WRT55AG, Meraki Mini [78], or Senao/Engenius 5354AP1 Aries2

More work on embedded ports:

- Steve Woodford added support for the Linksys NSLU2 NAS device to the evbarm port. The NSLU2 is an external hard disk with integrated Ethernet and USB.
- Shigeyuki Fukushima worked on the evbmips port to add support for OpenMicroServer 400 [72], yet another tiny server from a Japanese manufacturer who does not seem to have a page in English.
- NONAKA Kimihiro also hacked the evbarm port to add support for a similar Ethernet and USB enabled external hard disk from I-O DATA, the HDL-G400U.

And because embedded is not everything, we also had new desktop and server ports:

- Sanjay Lal imported initial support for Apple Powermac G5. For now it only works in 32-bit mode (using PowerPC 970 bridge mode), and it requires a serial console. It is able to boot to multi-user using a NFS root.
- And finally, Tim Rightnour improved the prep port to support two IBM RS/6000 models: IBM 7024-E20, 7025-F30, and 7025-F40, and the Motorola Powerstack E1.

5.2   AC97 modems

AC97 modems are a standardized set of software modems [73], also known as winmodems. A software modem is in fact a kind of sound card that connects to the phone line. The software has to perform the appropriate modulation of digital data into an analog signal suitable for being sent over telephone lines.

Hardware modems are seen as quite standard devices from the operating system. They are usually attached through a serial line (RS232, serial emulation communication over USB, over bluetooth), and they can be manipulated using the Hayes command set [74]. Software modems, on the other hand, need complex drivers that take care of all the modulation details.

The lack of drivers made winmodems quite unpopular on free OSes. It is interesting to note that they were not popular either in the Windows world, as buggy drivers turned them into unreliable and slow alternatives to hardware modems.

Jared D. McNeill made some work [75] to support AC97 modems. His contribution is split in two parts: First, improve kernel drivers to get access to the AC97 modem. Second, port to NetBSD the Linux `slmodemd` utility, which is the userland program that implements the soft modem.

5.3   VESA support

Jared also worked on VESA [76] support in NetBSD. VESA stands for Video Electronics Standards Association. It is a set of standards for video adapters which is better than plain old VGA. Most video boards implement VESA today.

The point in supporting VESA is that it allows using the console in graphic mode without having to get into the horrible details of how the video board works (that will be left to the X server). The console can therefore be used at higher resolution, and non ASCII character sets can be displayed.

And just for fun and because it was now possible, Jared D. McNeill added splash screen capability to the NetBSD kernel boot sequence [77].

5.4 MIDI

MIDI stands for Musical Instrument Digital Interface [79]. It is a communication protocol used to interface a computer and an electronic musical instrument.

The original NetBSD MIDI support, developed by Lennart Augustsson in 1998, and the USB MIDI support added by Takuya SHIOZAKI in 2001, served also as starting points for the MIDI support currently in FreeBSD and OpenBSD, but then saw little active improvement for a while, during which the code in the other projects diverged in order to address some bugs and functional concerns.

Chapman Flack adopted the NetBSD orphan MIDI code and fixed a lot of problems [80] on many aspects of MIDI support, from the sequencer API to hardware interrupt handling. Among other improvements, USB MIDI throughput problems that resulted in frequent dropped input data and drastically limited output rate have been corrected, with input drops eliminated under test conditions and sustained simultaneous output on multiple ports at the full MIDI 1.0 data rate. A default behavior for MIDI Active Sensing has been added that allows applications to detect communication interruptions with much simpler code than to parse and time out Active Sense messages explicitly, and that leads to reasonable behavior in pipelines of standard tools that have no knowledge of Active Sensing at all.

Chapman Flack also did a lot of code clean-up, and redesigned the MIDI framework to make it more machine-independent. The userland API has been clarified in a more detailed `midi(4)` [81] and an expanded `<sys/midiio.h>` [82] that for the first time supplies and documents a programming interface to the sequencer.

5.5 IEEE 1394

KIYOHARA Takashi imported FreeBSD's implementation of IEEE 1394 (also known as Apple's trade mark FireWire) [83]. That software allows NetBSD to use IEEE 1394 attached hard disks, and to use an IEEE 1394 link for IP communications.

That import also substantially improves the stability of IEEE 1394 on NetBSD.

5.6 Miscellaneous device driver work

Here is a quick summary of the steady work on device driver support:

- Network controllers
  - Damien Bergamini, FUKAUMI Naoki, and Matthias Drochner worked on a driver for Ralink PCI/Cardbus/USB WLAN adapters.
  - Rui Paulo worked on support for ASIX AX88140A and AX88141 Ethernet controllers.
  - Juan Romero Pardines imported OpenBSD's driver for Realtek 8139/8201L Ethernet interfaces.
  - Chuck Silvers imported OpenBSD's driver for NVIDIA nForce Ethernet controller
  - Garrett d'Amore imported HAL 0.9.17.2 from Atheros, to support new SoCs such as the AR531x
  - Rui Paulo added support for RT2661-based wireless interfaces
  - Tohru Nishimura developed a driver for Micrel KSZ8842 and KSZ8841 Ethernet controllers
  - Christos Zoulas imported David Boggs' driver [84] for SBE (previously known as LMC) Wide Area Network (WAN) cards [85]. Now one can build a NetBSD WAN router.
  - David Young also added support for GCT Semiconductor GRF5101 transceiver/synthesizer.
- Audio controllers
  - Juan Romero Pardines upgraded the `auich` audio driver to support ICH7 and Intel 6300ESB audio controllers.

- TAMURA Kent improved the `azalia` driver to bring S/PDIF [88] support.
- Chapman Flack upgraded the `eap` audio driver to use `txrdy` interrupts for MIDI, and added the `es1373` register definitions, which could be a start for S/PDIF support

- Disk controllers
  - Manuel Bouyer imported Joerg Sonnenberger's work on the driver for ServerWorks K2 SATA controller from OpenBSD.

- Human Interface Devices (HID)
  - KIYOHARA Takashi added support for the touch-panel and LCD screen of PERSONA SH3 machines.
  - Takeshi Nakayama added support for the the Sharp Telios LCD screen and Battery unit.
  - Christos Zoulas also imported Johan Wallen's driver for Apple's 15" powerbook mouse.
  - Peter Postma adopted the Jornada 720 machine-dependent code and worked on keyboard and power management.

- Serial communication and USB
  - Nick Hudson added a driver for Cypress micro controller based serial devices
  - Lennart Augustsson imported OpenBSD's driver for accessing an iPAQ through USB.

- Power management and hardware monitoring
  - Jared D. McNeill imported support for Intel power management technology SpeedStep PIIX4, from FreeBSD. On the AMD front, Juan Romero Pardines integrated Martin Vegiard's work on AMD PowerNow, and imported the Cool'n'Quiet driver from OpenBSD.
  - Juan Romero Pardines also imported OpenBSD's driver for ITE's IT8705F, IT8712F and SiS' SiS950 hardware monitors (these devices report temperature, fan speed, and various other useful information).
  - David Young added a driver for AMD Geode SC1100 micro controller's watchdog timer.
  - Jeff Rizzo imported OpenBSD's driver for Dallas Semiconductor 1-wire bus [87], General Purpose I/O (GPIO) and temperature sensors using that bus.

- Video devices
  - Steve Woodford developed a driver for the Topfield TF5000PVR range of digital video recorders [86].

6 Binary compatibility

NetBSD has the capability of running binaries from other OSes that are built for the same processor [89]. This works with very little overhead, by emulating system calls. When the foreign binary makes a system call, the NetBSD kernel behaves like the foreign OS kernel would have. The foreign binary gets appropriate answers from the kernel, and it just works.

With the help of Nicolas Joly, who made a lot of testing with Linux binaries, I improved a lot the Linux binary compatibility for machines running NetBSD/amd64. The NetBSD kernel now emulates enough of the Linux Native POSIX Thread Library (NPTL) [90] kernel code to masquerade as a 2.6 series kernel to Linux processes.

Unfortunately, other NetBSD ports lack the machine dependent code for emulating the NPTL, and are stuck at emulating the 2.4 Linux kernel.

I also contributed the 32-bit Linux binary compatibility for NetBSD/amd64, which is also known as the `COMPAT_LINUX32` kernel option. Intensive tests made by Nicolas Joly suggest that it has reached a fair level of usability.

7 System Packages

System Packages, or `syspkg`, is a new infrastructure for packaging the base system in fine-grained packages. Once `syspkg` will be fully integrated in the build and installation processes, an administrator will be able to install a NetBSD system that contains `dhclient` but not `dhcpd`, for instance.

`syspkg` is still under development. Alan Barrett made some progress, by adding the ability to generate `syspkg .tgz` files from the NetBSD build infrastructure. Nothing can be done yet with the generated files. The next step is to give tools such as `pkg_add` [91] the ability to install and upgrade `syspkg`.

For more information on how to generate `syspkg`, see the NetBSD build documentation [92].

8 Security

8.1 News from Security Officers

Since the beginning of 2006 the NetBSD Security Officer team has released 22 security advisories for NetBSD [93]. The advisories cover both issues found in third party software included in the base NetBSD operating system (e.g. BIND, OpenSSL etc.) and issues found in the kernel and user land. Included in this was SA2006-019 [94] which documented an issue discovered by two NetBSD developers that was found to impact all BSDs (NetBSD, OpenBSD, FreeBSD and DragonFly BSD). The severity of issues discovered ranged from denial of service to privilege escalation attacks.

8.2 News from pkgsrc security team

The pkgsrc Security Team monitors vulnerabilities found in software included as a part of pkgsrc. In May 2005 the team started using RT [95] to track issues to ensure that vulnerable packages are identified and updated promptly. In addition to this, the security team also tries to ensure that pull-ups for any security fixes are applied to the pkgsrc stable branches.

The file used for tracking package vulnerabilities [96] currently has 2183 active entries. At the start of 2006 it had 1606 entries, this represents 577 entries that have been added since.

For more information on the pkgsrc Security team see the presentations from Adrian Portelli at pkgsrcCon 2005 [97] and pkgsrcCon 2006 [27].

8.3 Further integration of PaX

The goal of the PaX project [98] is to provide a set of defense mechanisms against attacks that rely on writing in a process address space. Stack buffer overflows are the best example of such attacks. The main PaX idea is to prevent process memory to be writable and executable at the same time. That way, an attacker that uploads executable code through a buffer overflow will have difficulties to execute it. There are also other tricks, such as Address Space Layout Randomization (ASLR), which are designed to make attacks less reliable.

PaX was originally developed for Linux, but as usual, good ideas spread to other OSes. Elad Efrat updated NetBSD's `mprotect(2)` [99] to enforce W∧X (Write or eXecute, but not both) policies.

There are always odd programs that need to execute code they produce at run-time. The Java Just-In-Time (JIT) compiler is an example. The paxctl [100] tool can be used to enable or disable the W∧X policy on a per-program basis.

8.4 Kernel authorization framework

Kernel authorization (`kauth` for short) is work behind the scene also done by Elad Efrat. It produces few visible features to users, but it is the foundation for very interesting future work on security.

Authorization mechanisms have always been very simple in traditional Unix systems. Apart from file system permissions, the only kind of authorization checks that existed was "is the process running under UID 0?"

That meant that the root user had all the privileges, while other "unprivileged" users had none. Root's awesome power was loosely delegated through set-UID programs, which exposed a lot of unexpected security bugs for exploitation.

`kauth`'s goal is to make kernel authorization much more flexible, so that fine-grained security policies could be enforced. NetBSD `kauth` is a clean room implementation based on Apple's `kauth` [101] (the original code from Apple could not be copied as its licensing was too restrictive for inclusion in the BSD-licensed NetBSD kernel).

For now, NetBSD `kauth` is just used to re-implement the traditional Unix security model, but it allows future development of alternative security models. Capabilities [102] are an example of probable future work.

Elad also presents an article on NetBSD security improvements [103] at EuroBSDCon 2006, which includes `kauth` coverage.

8.5 File association kernel programming interface

File association kernel programming interface (`fileassoc` for short)[104] is another work behind the scenes done by Elad Efrat and Brett Lymn. `fileassoc` is a Kernel Programming Interface (KPI) used to store meta-data associated to a file. The first usage of `fileassoc` is to store trusted executable signatures for the NetBSD subsystem responsible of verifying executable integrity (also known as `veriexec`) [105].

Fileassoc can also be used to store any meta-data, for instance extended file system attributes such as Access Control Lists (ACL).

8.6 Paper on NetBSD security

It would be too bad to close the chapter of security without a word on Elad Efrat's paper on NetBSD security [106], published at securityfocus.com. This excellent article tells a lot about OS security state of the art and alternative that are available to NetBSD.

9 More work behind the scenes

Jason Thorpe added experimental support for storing extended file system attribute on FFSv1 file system. While FFSv2 has provisions to store file system extended attributes, there was no place to store them on a FFSv1 file system. It is now possible to store them in plain files.

Christos Zoulas merged the duplicated code between `libc` and kernel. There is now a `src/common` directory in the source tree that holds the shared code. That change was the opportunity to use the same `zlib` (compression algorithms used everywhere in the system) between kernel and userland. It was also the opportunity to make `libc` compatibility code build optional, so that people without the need of backward compatibility can build a smaller `libc`. There is a minor side effect: it's not possible anymore to just check out `src/sys` and build a kernel: `src/common` is now required as well.

On February 2005, I switched our IPsec key exchange daemon (known as `racoon`) from the orignal KAME [15] implementation to IPsec-tools [107], a fork made initially for Linux that was more reactive to features addition. IPsec-tools CVS was hosted at SourceForge, and it was decided to move it to the NetBSD CVS server. The reasons for the move were a more

reliable and more secure CVS, automatic builds, and automatic Coverity [108] scans. While IPsec-tools HEAD is merged in NetBSD-current, it still remains as a stand alone package for Linux, FreeBSD and Darwin, so that move changes nothing for IPsec-tools and NetBSD users (except that NetBSD-current will now always include latest IPsec-tools code).

And finally, Darren Reed, Nick Hudson, and Christos Zoulas completed the work required so that `ktrace` and `kdump` get the ability to report Light-Weight Process (LWP) information.

10 Third party software

NetBSD ships with various third party software integrated [109]. Here is the current status as of October 2006.

10.1 Removed software

- Sushi [110], the curses-based administration tool. Obviously, it never really found its users, and it was difficult to maintain.
- Kerberos IV [111] has been removed, in favor of Kerberos V [112]. Both versions were maintained in-tree for some time, but it is now assumed that all Kerberos IV users have migrated to Kerberos V.
- Vinum [113] was removed because nobody was interested enough in it to actually maintain the code. This is probably because NetBSD already had RAIDframe, which provides a similar set of features (except the volume manager part, for which NetBSD is left without any equivalent).
- Sendmail [114] suffered a new security issue, but nobody was ready to maintain it. It was therefore decided to remove it, leaving Postfix [115] as the only mail software in the base system (previous NetBSD releases contained both Sendmail and Postfix). Fortunately for Sendmail fans like me, Sendmail remains quite easy to install through the NetBSD package system.

10.2 Software resurrected from the dead

In January 2002, Caldera released [116] the source for ancient Unix versions up to AT&T UNIX version 7 [117]. All the critical pieces of Unix have been re-implemented as free software in *BSD for a long time, but there were a few nifty utilities left that Perry E. Metzger resurrected from the dead and integrated in NetBSD:

- `deroff` [118], a tool to remove roff constructs from files
- `spell` [119], a spell-checker
- `ching` [120], the Unix oracle, which answers any of your questions (that one stands in `/usr/games`)

10.3 Third party software upgrades

Many contributors did some work on upgrading third party programs bundled with NetBSD. The third party software distributed with NetBSD page [109] gives details about individual programs.

- `pppd` 2.4.4
- NTP 4.2.2p2
- GCC 4.1 and GCC 3.3.6 (Some ports use 4.1, others still use 3.3.6)
- binutils 2.16.1
- BIND 9.3.2

- `file` 4/16
- am-utils 6.1.3
- CVS 1.11.22
- OpenSSL 0.9.8a
- OpenSSH 4.3
- OpenPAM 20050616

- `groff` 1.19.2
- IPFilter: 4.1.13
- PacketFilter (PF): from OpenBSD 3.7

- Postfix 2.3.2
- `zlib` 1.2.3
- `wpa_supplicant` / `hostapd` 0.4.9

And last but not least, the X.org status. NetBSD is committed to switch to X.org [121], since most, if not all, open source OSes did that move. It is now clear that most development will come from X.org and not from the XFree86 project [122].

X.org 7.0 code has been imported in NetBSD CVS by Michael Lorenz. It seems no status has been published, but according to Michael, it it possible to build and run X.org on NetBSD. Performance is a bit disappointing, as `x11perf` shows a 10% drop versus XFree86. On the other hand, X.org memory usage is a bit better.

The next step in X.org migration is to integrate the build in the NetBSD build machinery, and start using it as the default X implementation for ports where it has some interest. It is worth noting that a few NetBSD ports with very odd X servers may have no point into moving from XFree86 to X.org, because the X.org implementation will offer no benefit, and may not be really maintained. It is therefore possible that NetBSD retains both X implementation for some time.

Conclusions

Whatever has been said in the press, NetBSD is still a very active project, and giving an idea of a whole year of activity is not a straightforward task. Of course this paper left some changes unmentioned. Curious readers might want to take a peek at the raw NetBSD change log [123] in order to get an exhaustive list.

One recurrent issue when trying to collate a list of significant changes is that too often, developers tend to neglect telling the world what they did and why it is so cool. The change log is full of lines which are meaningless for the average user, and even sometime for the average NetBSD developer. In my opinion, we obviously have some room for improvement here. Giving more publicity about what is done in NetBSD is one way to stop the rumors that it got irrelevant, dead, or that it is only useful for running toasters.

Acknowledgments

References

[1] Anonymous coward, *BSD is dying, Slashdot web site
   `http://bsd.slashdot.org/comments.pl?sid=189013&cid=15569908`

[2] Charles M. Hannum, The future of NetBSD, `netbsd-users@netbsd.org` mailing list
   `http://mail-index.netbsd.org/netbsd-users/2006/08/30/0016.html`

[3] Charles M. Hannum, Confessions of a Recovering NetBSD Zealot, OnLAMP web site
   `http://www.onlamp.com/pub/a/bsd/2006/09/14/netbsd_future.html`

[4] The NetBSD project, `source-change@netbsd.org` mailing list
   `http://mail-index.netbsd.org/source-changes`

[5] Alistair G. Crooks, Organizational Changes to the NetBSD Project, `netbsd-users@netbsd.org` mailing list
   `http://mail-index.netbsd.org/netbsd-users/2006/09/01/0015.html`

[6] The NetBSD foundation, NetBSD Foundation Membership Agreement, NetBSD web site
http://www.netbsd.org/developers/agreement.txt

[7] The NetBSD foundation, The NetBSD Bugathon: Reloaded, NetBSD web site
http://www.netbsd.org/hackathon/

[8] The NetBSD foundation, GNATS Bug Database Summary, NetBSD web site
http://www.netbsd.org/Gnats/

[9] Elad Efrat, NetBSD Bugathon: Not quite dead, netbsd-anounce@netbsd.org mailing list
http://mail-index.netbsd.org/netbsd-announce/2006/09/25/0000.html

[10] Elad Efrat, NetBSD Bugathon #2, netbsd-users@netbsd.org mailing list
http://mail-index.netbsd.org/netbsd-users/2006/10/09/0002.html

[11] Hubert Feyrer, NetBSD open Problem Reports, NetBSD web site
http://www.netbsd.org/~hubertf/open-prs.gif

[12] Google, Google Summer of Code, Google web site
http://code.google.com/soc/

[13] Hubert Feyrer, Announcing NetBSD and the Google "Summer of Code" Projects 2006, NetBSD press releases
http://www.netbsd.org/Foundation/press/soc2006.html

[14] K. Ramakrishnan, S. Floyd, D. Black, The Addition of Explicit Congestion Notification (ECN) to IP, RFC 3168
http://www.rfc-editor.org/rfc/rfc3168.txt

[15] The KAME project
http://www.kame.net/

[16] The NetBSD project, fast_ipsec(4) man page
http://netbsd.gw.com/cgi-bin/man-cgi?fast_ipsec++NetBSD-current

[17] The NetBSD project, pkgsrc: The NetBSD Packages Collection, pkgsrc web site
http://www.pkgsrc.org

[18] The NetBSD project, mbuf(9) man page
http://netbsd.gw.com/cgi-bin/man-cgi?mbuf++NetBSD-current

[19] Jorg Sonnenberger, pkg_install, EuroBSDCon 2006
http://www.eurobsdcon.org/talks-sonnenberger.php

[20] pkgsrcCon web site
http://www.pkgsrccon.org/2006/

[21] Stoned Elipot, System administration with pkgsrc, pkgsrcCon 2006
http://www.pkgsrccon.org/2006/slides/mpkg.pdf

[22] Joerg Sonnenberger, pkgsrc on DragonFly – or Fighting the Windmills, pkgsrcCon 2006
http://www.pkgsrccon.org/2006/slides/pkgsrc-on-df/index.html

[23] Roland Illig, pkglint: Static Analyzer For Pkgsrc, pkgsrcCon 2006
http://www.pkgsrccon.org/2006/slides/presentations/pkglint.html

[24] Roland Illig, Why Pkgsrc Sucks, pkgsrcCon 2006
http://www.pkgsrccon.org/2006/slides/presentations/why-pkgsrc-sucks.html

[25] Emile Heitor, pkg_select – So Many Packages, So Few Columns, pkgsrcCon 2006
http://www.pkgsrccon.org/2006/slides/pkg_select.pdf

[26] Thomas Klausner, Roundtable Discussion: Updating Packages, pkgsrcCon 2006
http://www.pkgsrccon.org/2006/slides/updates.html

[27] Adrian Portelli, pkgsrc security one year on..., pkgsrcCon 2006
http://www.pkgsrccon.org/2006/slides/pkgsrc-Security-06.html

[28] Dieter Baron, Thomas Klausner, pkg_install Rewrite, pkgsrcCon 2006
http://www.pkgsrccon.org/2006/slides/pkg_install.html

[29] Johnny Lam, Roadmap for Development, pkgsrcCon 2006
http://www.pkgsrccon.org/2006/slides/roadmap.html

[30] University of Cambridge computer laboratory, the Xen virtual machine monitor, Xen web site
http://www.cl.cam.ac.uk/research/srg/netos/xen/

[31] Emmanuel Dreyfus, Remote user access VPN with IPsec, EuroBSDCon 2005
http://pubz.hcpnet.net/rasvpn.pdf

[32] University of Cambridge computer laboratory, performances, Xen web site
http://www.cl.cam.ac.uk/research/srg/netos/xen/performance.html

[33] Martti Kuparinen, Xen Disk I/O Benchmarking: NetBSD dom0 vs Linux dom0
http://users.piuha.net/martti/comp/xendom0/xendom0.html

[34] J. Satran, K. Meth, C. Sapuntzakis, M. Chadalapaka, E. Zeidner, Internet Small Computer Systems Interface (iSCSI), RFC 3720
http://www.rfc-editor.org/rfc/rfc3720.txt

[35] Alistair G. Crooks, NetBSD iSCSI HOWTOs, current-users@netbsd.org mailing list
http://mail-index.netbsd.org/current-users/2006/02/21/0018.html

[36] Alistair G. Crooks, iSCSI - beyond the hype, EuroBSDCon 2006
http://www.eurobsdcon.org/talks-crooks.php

[37] Luke Mewburn, Matthew Green, build.sh: cross-building NetBSD, BSDCon 2003
http://www.mewburn.net/luke/papers/build.sh.pdf

[38] The NetBSD project, Summary of daily snapshot builds, NetBSD release engineering web site
http://releng.netbsd.org/cgi-bin/builds.cgi

[39] The NetBSD project, makefs(8) man page
http://netbsd.gw.com/cgi-bin/man-cgi?makefs++NetBSD-current

[40] Wikipedia, Wi-Fi Protected Access
http://en.wikipedia.org/wiki/Wi-Fi_Protected_Access

[41] Wikipedia, Wired Equivalent Privacy
http://en.wikipedia.org/wiki/Wired_Equivalent_Privacy

[42] S. Fluhrer, I. Mantin, A. Shamir, Weaknesses in the Key Scheduling Algorithm of RC4, Selected Areas in Cryptography 2001: pp1-24.

[43] Steve Woodford, WPA support, current-users@netbsd.org mailing list
http://mail-index.netbsd.org/current-users/2005/10/01/0014.html

[44] Jouni Malinen, hostapd(8) man page
http://netbsd.gw.com/cgi-bin/man-cgi?hostapd++NetBSD-current

[45] Jouni Malinen, wpa_supplicant(8) man page
http://netbsd.gw.com/cgi-bin/man-cgi?wpa_supplicant++NetBSD-current

[46] Jouni Malinen, Host AP driver for Intersil Prism2/2.5/3, hostapd, and WPA Supplicant
http://hostap.epitest.fi/

[47] Wikipedia, Bluetooth
http://en.wikipedia.org/wiki/Bluetooth

[48] Iain Hibbert, Bluetooth, tech-net@netbsd.org mailing list
http://mail-index.netbsd.org/tech-net/2006/05/23/0000.html

[49] The NetBSD Wiki, Bluetooth
http://wiki.netbsd.se/index.php/bluetooth

[50] Wikipedia, Universal Disk Format
http://en.wikipedia.org/wiki/Universal_Disk_Format

[51] Reinoud Zandijk, HEADS UP: UDF file system added to NetBSD source tree, tech-kern@netbsd.org mailing list
http://mail-index.netbsd.org/current-users/2006/02/02/0027.html

[52] Wikipedia, Common Address Redundancy Protocol
http://en.wikipedia.org/wiki/Common_Address_Redundancy_Protocol

[53] R. Hinden, Virtual Router Redundancy Protocol (VRRP), RFC 3768
http://www.ietf.org/rfc/rfc3768.txt

[54] T. Li, B. Cole, P. Morton, D. Li, Cisco Hot Standby Router Protocol (HSRP), RFC 2281
http://www.ietf.org/rfc/rfc2281.txt

[55] Liam J. Foy, CARP Committed (correctly presented), current-users@netbsd.org mailing list
http://mail-index.netbsd.org/current-users/2006/05/18/0005.html

[56] Wikipedia, LACP
http://en.wikipedia.org/wiki/LACP

[57] YAMAMOTO Takashi, agr(4) man page
http://netbsd.gw.com/cgi-bin/man-cgi?agr++NetBSD-current

[58] Microsoft corporation, NDIS - Network Driver Interface Specification
http://www.microsoft.com/whdc/device/network/ndis/default.mspx

[59] David Chisnall, Project Evil: Windows network drivers on FreeBSD
http://www.pingwales.co.uk/2005/07/15/Project-Evil.html

[60] NdisWrapper for Linux at SourceForge
http://ndiswrapper.sourceforge.net/

[61] Alan Ritter, NDIS on NetBSD
http://netbsd-soc.sourceforge.net/projects/ndis/

[62] Bill Paul, ndiscvt(8) man page
http://netbsd.gw.com/cgi-bin/man-cgi?ndiscvt++NetBSD-current

[63] Julio M. Merino Vidal, HEADS UP: tmpfs added, `tech-kern@netbsd.org` mailing list
http://mail-index.netbsd.org/tech-kern/2005/09/10/0004.html

[64] Niklas Hallqvist, Tobias Weingartner, `scan_ffs(8)` man page
http://netbsd.gw.com/cgi-bin/man-cgi/man?scan_ffs+8+NetBSD-current

[65] Wikipedia, Log-structured File System
http://en.wikipedia.org/wiki/Log-structured_file_system

[66] Konrad Schröder, Log-structured File System for NetBSD
http://www.hhhh.org/perseant/lfs.html

[67] Atmark techno, Armadillo-9
http://www.atmark-techno.com/en/products/armadillo/a9/

[68] Atmark techno, Armadillo-210
http://www.atmark-techno.com/en/products/armadillo/a210/

[69] The NetBSD project, ews4800mips port page, NetBSD web site
http://www.netbsd.org/Ports/ews4800mips/

[70] RMI, Alchemy Au1550 Processor
http://www.razamicroelectronics.com/products_alchemy/au1550_overview.htm

[71] meshcube.org, The meshing computing website
http://www.meshcube.org/index_e.html

[72] Plat'Home, OpenMicroServer 400
http://www.plathome.co.jp/products/oms400/

[73] Wikipedia, softmodem
http://en.wikipedia.org/wiki/Winmodem

[74] Wikipedia, Hayes command set
http://en.wikipedia.org/wiki/Hayes_command_set

[75] Jared D. McNeill, Preliminary AC'97 modem support in `auich(4)`, `current-users@netbsd.org` mailing list
http://mail-index.netbsd.org/current-users/2005/04/07/0022.html

[76] Wikipedia, VESA
http://en.wikipedia.org/wiki/VESA

[77] Jared D. McNeill, VESA framebuffer console in NetBSD
http://www.invisible.ca/space/vesa-framebuffer-console-in-netbsd

[78] Meraki Networks, Meraki Mini
http://www.meraki.net/mini.html

[79] Wikipedia, MIDI
http://en.wikipedia.org/wiki/MIDI

[80] Chapman Flack, RFC: merge `chap-midi` branch, `tech-kern@netbsd.org` mailing list
http://mail-index.netbsd.org/tech-kern/2006/06/19/0003.html

[81] Lennart Augustsson, Chapman Flack, `midi(4)` man page
http://netbsd.gw.com/cgi-bin/man-cgi?midi+4+NetBSD-current

[82] Lennart Augustsson, Chapman Flack, `<sys/midiio.h>` header file
http://cvsweb.netbsd.org/bsdweb.cgi/src/sys/sys/midiio.h?rev=1.13.4.1

[83] Wikipedia, firewire
http://en.wikipedia.org/wiki/Firewire

[84] David Boggs, `lmc(4)` man page
http://netbsd.gw.com/cgi-bin/man-cgi?lmc+4+NetBSD-current

[85] SBE Inc, Products - WAN
http://www.sbei.com/index.php/products/wan/

[86] Topfield web site
http://www.topfield.co.kr/

[87] Wikipedia, 1-Wire
http://en.wikipedia.org/wiki/1-Wire

[88] Wikipedia, S/PDIF
http://en.wikipedia.org/wiki/S/PDIF

[89] The NetBSD project, NetBSD binary emulation
http://www.netbsd.org/Documentation/compat.html

[90] Ulrich Drepper, Ingo Molnar, The Native POSIX Thread Library for Linux
http://people.redhat.com/drepper/nptl-design.pdf

[91] Jordan Hubbard, John Kohl, Hubert Feyrer, Thomas Klausner, `pkg_add(1)` man page
http://netbsd.gw.com/cgi-bin/man-cgi?pkg_add++NetBSD-current

[92] Luke Mewburn, Todd Vierling, Procedure for building NetBSD from source
`ftp://ftp.fr.netbsd.org/pub/NetBSD/NetBSD-current/src/BUILDING`

[93] The NetBSD project, Security and NetBSD
`http://www.netbsd.org/Security/`

[94] NetBSD Security Officer team, NetBSD Security Advisory 2006-019
`ftp://ftp.netbsd.org/pub/NetBSD/security/advisories/NetBSD-SA2006-019.txt.asc`

[95] Best Pratical Solutions LLC, Request Tracker
`http://www.bestpractical.com/rt/`

[96] The NetBSD project, packages vulnerability file
`ftp://ftp.netbsd.org/pub/NetBSD/packages/distfiles/pkg-vulnerabilities`

[97] Adrian Portelli, pkgsrc security, pkgsrcCon 2005
`http://www.pkgsrccon.org/2005/slides/adrianp/pkgsrc-Security.html`

[98] The PaX project
`http://pax.grsecurity.net/docs/pax.txt`

[99] The NetBSD project, `mprotect(2)` man page
`http://netbsd.gw.com/cgi-bin/man-cgi?mprotect+2+NetBSD-current`

[100] Elad Efrat, `paxctl(1)` man page
`http://netbsd.gw.com/cgi-bin/man-cgi?paxctl+1+NetBSD-current`

[101] Apple computers Inc, Technical Note TN2127 Kernel Authorization
`http://developer.apple.com/technotes/tn2005/tn2127.html`

[102] Wikipedia, capability-based security
`http://en.wikipedia.org/wiki/Capabilities`

[103] Elad Efrat, NetBSD Security Enhancements, EuroBSDCon 2006
`http://www.eurobsdcon.org/talks-efrat.php`

[104] Elad Efrat, Brett Lymn, `fileassoc(9)` man page
`http://netbsd.gw.com/cgi-bin/man-cgi?fileassoc++NetBSD-current`

[105] Brett Lymn, NetBSD verified executables
`http://www.users.on.net/~blymn/veriexec/`

[106] Elad Efrat, Recent Security Enhancements in NetBSD
`http://www.securityfocus.com/infocus/1878`

[107] The IPsec-tools project
`http://ipsec-tools.sf.net`

[108] Coverity, automated error precention and source code analysis
`http://www.coverity.com/`

[109] The NetBSD project, Third party software distributed with NetBSD
`http://netbsd.org/Documentation/software/3rdparty/`

[110] Tim Rightnour, Sushi - an extensible human interface for NetBSD, BSDCon 2002
`http://db.usenix.org/events/bsdcon02/full_papers/rightnour/rightnour.pdf`

[111] Kungliga Tekniska Högskolan (KTH) Kerberos page
`http://www.pdc.kth.se/kth-krb/`

[112] Heimdal Kerberos page
`http://www.pdc.kth.se/heimdal/`

[113] Greg Lehey, The vinum volume manager
`http://www.vinumvm.org/`

[114] the Sendmail consortium, Sendmail home page
`http://www.sendmail.org/`

[115] the Postfix project, Postfix home page
`http://www.postfix.org`

[116] Bill Broderick, Caldera license
`http://www.tuhs.org/Archive/Caldera-license.pdf`

[117] Unix Archive Sites
`http://www.tuhs.org/archive_sites.html`

[118] `deroff(1)` man page
`http://netbsd.gw.com/cgi-bin/man-cgi?deroff++NetBSD-current`

[119] `spell(1)` man page
`http://netbsd.gw.com/cgi-bin/man-cgi?spell++NetBSD-current`

[120] `ching(6)` man page
`http://netbsd.gw.com/cgi-bin/man-cgi?ching++NetBSD-current`

[121] The X.org foundation
       http://www.x.org/

[122] The XFree86 project
       http://www.xfree86.org/

[123] The NetBSD project, Recent Changes and News
       http://www.netbsd.org/Changes/