NAME

INSTALL — Installation procedure for NetBSD/sun2.

CONTENTS

About this Document....1
What is NetBSD?....2
Changes Between The NetBSD 5.0 and 6.0 Releases....2
  General kernel....2
  Networking....3
  File systems....3
  Security....4
  Drivers....4
  Platforms....6
  Userland....12
  Components removed from NetBSD....15
Features to be removed in a later release....15
The NetBSD Foundation....15
Sources of NetBSD....15
NetBSD 6.0 Release Contents....16
  NetBSD/sun2 subdirectory structure....17
  Binary distribution sets....17
NetBSD/sun2 System Requirements and Supported Devices....18
  Supported hardware....18
Getting the NetBSD System on to Useful Media....19
  Creating boot/install tapes....19
  Boot/Install from NFS server....19
  Install/Upgrade from CD-ROM....20
  Install/Upgrade via FTP....20
Preparing your System for NetBSD installation....20
Installing the NetBSD System....20
  Installing from tape....21
  Installing from NFS....21
  Installing from SunOS....22
  Booting the Miniroot....23
  Miniroot install program....23
Post installation steps....24
Upgrading a previously-installed NetBSD System....26
Compatibility Issues With Previous NetBSD Releases....27
  Issues affecting an upgrade from NetBSD 5.x releases....27
  Issues affecting an upgrade from NetBSD 4.x releases....27
Using online NetBSD documentation....28
Administrivia....28
Thanks go to....29
We are....29
Legal Mumbo-Jumbo....37
The End....41

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 6.0 on the *sun2* platform. It is available in four different formats titled `INSTALL.`*`ext`*, where *`.ext`* is one of `.ps`, `.html`, `.more`, or `.txt`:

`.ps` PostScript.

`.html` Standard Internet HTML.

`.more` The enhanced text format used on UNIX-like systems by the `more`(1) and `less`(1) pager utility programs. This is the format in which the on-line *man* pages are generally presented.

`.txt` Plain old ASCII.

You are reading the *PostScript* version.

What is NetBSD?

The NetBSD Operating System is a fully functional Open Source UNIX-like operating system derived from the University of California, Berkeley Networking Release 2 (Net/2), 4.4BSD-Lite, and 4.4BSD-Lite2 sources. NetBSD runs on 57 different system architectures (ports) across 15 distinct CPU families, and is being ported to more. The NetBSD 6.0 release contains complete binary releases for many different system architectures. (A few ports are not fully supported at this time and are thus not part of the binary distribution. Please see the NetBSD web site at **http://www.NetBSD.org/** for information on them.)

NetBSD is a completely integrated system. In addition to its highly portable, high performance kernel, NetBSD features a complete set of user utilities, compilers for several languages, the X Window System, firewall software and numerous other tools, all accompanied by full source code.

NetBSD is a creation of the members of the Internet community. Without the unique cooperation and coordination the net makes possible, it's likely that NetBSD wouldn't exist.

Changes Between The NetBSD 5.0 and 6.0 Releases

The NetBSD 6.0 release provides numerous significant functional enhancements, including support for many new devices, integration of hundreds of bug fixes, new and updated kernel subsystems, and many user-land enhancements. The result of these improvements is a stable operating system fit for production use that rivals most commercially available systems.

It is impossible to completely summarize the massive development that went into the NetBSD 6.0 release. The complete list of changes can be found in the CHANGES:

http://ftp.NetBSD.org/pub/NetBSD/NetBSD-6.0/CHANGES

and CHANGES-6.0:

http://ftp.NetBSD.org/pub/NetBSD/NetBSD-6.0/CHANGES-6.0

files in the top level directory of the NetBSD 6.0 release tree.

Some highlights include:

General kernel

- `time_t` and `dev_t` are now 64 bits.
- Much-improved compat_linux support for running Linux binaries.
- Added a generic producer/consumer queue.
- Added a new ptree (Patricia / RADIX tree) implementation.
- Removed `MALLOC()` and `FREE()` macros.

- Compute `kern.ipc.shmmaxpages` on startup based on physical memory. Can be overridden via `sysctl`(8).
- `kmem_alloc`(9): Add more extensive runtime debugging facilities.
- File descriptor access performance improvements were made.
- `cpu_reboot`(9): tear down stacks of devices and file systems in an orderly fashion during shutdown.
- Added 'show vmem' and 'show all vmems' commands to `ddb`(4).
- Added high-priority `xcall`(9) support.
- Added direct select()/poll() support.
- Add PCU (Per-CPU Unit) interface to unify synchronization of per CPU context, like lazy FPU management.
- Added new system calls `pipe2`(2), `paccept`(2), `kqueue1`(2), flags SOCK_CLOEXEC/SOCK_NONBLOCK to `socket`(2), `socketpair`(2), F_DUPFD_CLOEXEC to `fcntl`(2), and a MSG_CMSG_CLOEXEC to `recvfrom`(2) to be able to set close-on-exec to all newly created file descriptors.
- Major improvements to `uvm_map`(9), `uvm_km`(9), `vmem`(9), and `kmem`(9) layers.
- Removed `ltsleep`(9) and wakeup_one(9).
- Rewrote most of the entropy-gathering and entropy-distributing code, separating the `rnd`(4) pseudodevice from the bitstream generation code `cprng`(9) and entropy-pool code `rnd`(9).
- `kmem_alloc`(9): add more extensive runtime debugging facilities.
- Kernel support for `posix_spawn`(3).

Networking

- `ip`(4): added IP_RECVTTL option to let `recvmsg`(2) return the TTL of the received datagram.
- `ip`(4): added IP_MINTTL option to discard packets with a TTL lower than the option value.
- Added SOCK_SEQPACKET for local sockets.
- Added `net.inet.icmp.bmcastecho` `sysctl`(7), to disable ICMP replies to the braodcast address.
- Make TCP window size autoscaling the default.
- Derive the mbuf cluster (NMBCLUSTERS) at boot time from the available memory. The default can be raised by changing `kern.mbuf.nmbclusters`
- Added NPF, the NetBSD Packet Filter.
- `udp`(4): implement RFC6056 port selection algorithms.

File systems

- Added CHFS, a file system for flash devices.
- The default behavior when mounting empty msdosfs file systems has been changed to assume long filenames. The behavior with populated file systems has not changed.
- UDF: Added support for session opening/closing on sequential media.

- ext2fs: Added support for >128 byte inodes.
- Import OpenSolaris ZFS code. Experimental-only; does not work yet.
- ffs: New quota support. This adds a new on-disk format to store disk quota usage and limits, integrated with ffs metadata. Usage is checked by `fsck_ffs`(8) (no more quotacheck) and is covered by the WAPBL journal. Enabled with kernel option QUOTA2.

Security

- `secmodel_bsd44`(9), `secmodel_securelevel`(9): Add provisions to control access to `gpio`(4) pins.
- `security.conf`(5): Add check_pkg_vulnerabilities and check_pkg_signatures to validate the installed packages against the vulnerabilities database and the expected checksums for their files.

Drivers

- Audio:
  - Added `gcscaudio`(4) driver for AMD Geode CS5536.
  - Added `hdaudio`(4), a new standards-compliant Intel High Definition Audio driver written to replace `azalia`(4).
  - `audio`(9): Audio drivers are now MP-safe.
- Hardware Monitoring:
  - `sysmon_envsys`(9): Enhancements to allow access to driver-internal limit values.
  - `acpicpu`(4): Add a driver for ACPI-based processor functionality.
  - `acpipmtr`(4): Add a driver for ACPI power meters.
  - `acpismbus`(4): ACPI SMBus Control Method Interface driver.
  - `acpiwdrt`(4): ACPI Watchdog Resource Tables driver.
  - `acpiwmi`(4): Windows Management Instrumentation support for ACPI. Also added mappings for `wmidell`(4), `wmieeepc`(4), `wmihp`(4), and `wmimsi`(4).
  - `aibs`(4): New driver for ASUSTeK AI Booster (ACPI ASOC ATK0110) hardware monitor with limit support.
  - `amdtemp`(4): Add support for Family 12h.
  - `dbcool`(4): Added support for ADM1031 thermal sensor / fan controller.
- Networking:
  - Added support for more chip revisions to `re`(4). Added hardware checksum support for newer PCIe 8168C/8111C/8102E chips.
  - Switch `ath`(4) from the binary HAL to the open-source HAL from Sam Leffler.
  - Added `bwi`(4) driver for Broadcom AirForce / Apple Airport Extreme network cards.
  - Added flow control and RTL8211C(L) PHY support to `rgephy`(4).
  - Added `age`(4) and `atphy`(4) drivers for Attansic/Atheros L1 Gigabit Ethernet and F1 PHY, respectively.

- Added support for flow control to sk(4).
- Added support for flow control on MCP65 to nfe(4).
- Added TCPv4/UDPv4 RX hardware checksum support for i82559 and later chips to fxp(4).
- wm(4): Added ICH10, PCH, PCH2, 82575, 82576, 82580 and I350 devices support. Added some other improvements and workarounds.
- bge(4): added misc quirk code for chip specific bugs.
- Added ale(4): Attansic/Atheros L1E Gigabit Ethernet.
- agr(4): added support for layering vlans on top, and allow LACP to be disabled.
- alc(4): Add a driver for Atheros AR813x/AR815x Ethernet.
- aue(4): Add support for I-O DATA ETX-US2.
- bnx(4): Added support for Broadcom BCM5709 and BCM5716 chips. Add support for Broadcom BCM5709S (SerDes) chip.
- et(4), etphy(4): Add a driver for Agere/LSI ET1310/ET1301 10/100/Gigabit Ethernet device and the Agere/LSI ET1011 TruePHY Gigabit Ethernet PHY.
- kue(4): fixed unaligned memory accesses so it now works on ARM and MIPS machines.
- ne(4): add proper support for NE2000 8-bit mode.
- nfe(4): Add support for flow control for MCP65.
- otus(4): Atheros AR9100U driver.
- smsh(4): Add a SMSC LAN9118 Family Ethernet driver.
- upgt(4): Add a driver for Conexant/Intersil PrismGT SoftMAC USB IEEE 802.11b/g WLAN.

• Storage:

- Added an experimental in-kernel iSCSI initiator, from Wasabi Systems.
- viaide(4): Added VT8237S Integrated SATA controller support, and VT6410 PATA RAID controller support (without RAID).
- Added LVM functionality with dm(4) driver and lvm2 userland tools.
- Added support for 82965PM to piixide(4).
- vnd(4): implemented DIOCCACHESYNC. Also, allow vnd(4) to be backed by a sparse file.
- Added sdmmc(4) driver for SD/MMC.
- mvsata(4): support the Marvell Hercules-I/II SATA controllers.
- flash(9): Add a subsystem to handle flash memory devices, and nand(9), a subsystem to handle NAND controllers.
- mfi(4): Add support for LSI's newer (GEN2) RAID controller.
- nside(4): Add a driver for the National Semiconductor PC87415 IDE controller.
- raid(4):
  - RAIDFrame parity maps, which greatly improve parity rewrite times after unclean shutdown.

    - Added support for non 512-byte sector devices.
    - Added support for >2TB devices.
- USB:
  - uftdi(4): Added support for REX-USB60F.
  - ehci(4): work around USB subsystem freeze for SB600/SB700 chipsets.
  - ums(4): Added touchpanel support.
- Graphics and Video:
  - drm(4): updated to mesa-drm 85b9f737db0d2a845e4d7e2bbf9ad12ff9e2227c.
  - acpivga(4): ACPI Video Extensions driver.
  - agp(4): Added support for Intel G35, Intel G45, and Intel 82855GM.
  - auvitek(4): Add a driver for Auvitek AU0828 family USB video capture controllers.
  - dtv(4): Add new digital TV framework.
  - cxdtv(4): Add driver for Conexant CX23880-based DTV cards.
  - emdtv(4): Add a driver for Empia EM28xx family USB video capture controllers.
  - omapfb(4): a simple driver for OMAP 3xxx on-chip video, especially the Beagleboard.
- Miscellaneous:
  - Added Elantech multi-touch trackpad support to pms(4).
  - Added the altmem(4) "alternative memory"

    disk driver, which allows memory that is normally inaccessible by the machine-dependent pmap(9) to be used (as swap space).
  - ichlpcib(4): support 82801IEM LPC Interface Bridge.
  - Complete rework of gpio(4) to integrate with kauth(9), allow for runtime driver attachment, and naming of individual pins. Add gpiosim(4) driver to simulate a gpio(4) device for testing purposes.
  - Added gpioiic(4) driver to bit-bang an I2C bus using GPIO pins.
  - btmagic(4): Apple Magic Mouse driver.
  - vmt(4): Add a VMware Tools driver, from OpenBSD.

Platforms

- x86 (i386, amd64):
  - pxeboot(8) now parses boot.cfg(5).
  - The bootloader has been enhanced to support framebuffer consoles using VESA BIOS extensions. x86 ports now work with the genfb(4) framebuffer, and the i386-only vesafb(4) framebuffer has been obsoleted.
  - Added ucas(9) support for x86.
  - sysinst(8) now defaults to UFS2 for x86 platforms.

  - CPU topology detection suport for AMD processors.
  - Enable Ext2fs support in bootloader.
  - Booting from a disk with a GUID Partition Table (GPT) is now possible.
  - Support `userconf`(4) command in `boot.cfg`(5).
- i386:
  - Use "i486"

    as GNU_ARCH.
  - Added PAE (Physical Address Extension) support.
  - Better support for VIA C7 CPUs.
- amd64:
  - Added direct map support (mapping of all physical memory using large pages).
- alpha:
  - X11: add support for X11FLAVOUR=xorg.
  - Support for Avalon A12 removed.
  - Converted Alpha to use PCU to manage FPU.
- amiga:
  - Added basic PCI support.
  - `efa`(4): ELBOX FastATA 1200 driver.
  - Add a driver for BlizzardPPC 603e+ SCSI.
- amigappc:
  - A CSPPC system with `ahsc`(4) or `cbiiisc`(4) boots into multiuser;
  - Onboard serial now works.
  - `efa`(4): ELBOX FastATA 1200 driver.
- ARM ports(acorn26, acorn32, cats, evbarm, hpcarm, iyonix, netwinder, shark, zaurus):
  - Cortex-A8 CPU support is available.
  - i.MX51 SoC support is available.
- atari:
  - Add support for the SMC Elite Ultra Ethernet with the SMC_TT VME-ISA bridge.
  - Add support for the EtherNEC, ISA NE2000 based Ethernet connected to Atari ROM cartridge slot.
- bebox:
  - Support xserver. Tested on s3(TRIO 64V+).
- cobalt:
  - Add LBA48 support to bootloader so that it can safely load a kernel from NetBSD partition beyond >137GB boundary.

- dreamcast:
  - Fixed a long standing GD-ROM driver bug so that file system on GD-ROM works properly.
  - Build and put a raw binary format kernel into distribution.
  - Switch to using Xorg servers and binaries.
- evbarm:
  -
  - Added support for Gumstix Verdex and Verdex Pro.
  - Support Marvell Sheevaplug and some Marvell SoC NAS boxes.
  - Add initial support for Sharp NetWalker MID.
  - Add driver for OMAP NAND controllers.
  - Add FriendlyARM Mini2440 support.
  - Add initial support for the RaspberryPi. No support yet for the USB controller, the Ethernet interface, or the video chip. Serial console works, and boots multiuser.
- evbmips:
  - Added `voyagerfb`(4) driver, for the Gdium Liberty 1000's video controller.
  - Added support for RMI XLS6ATX_7A board and XL SoC family.
  - Added support for RMI (NetLogic) XLS/XLR models.
- evbppc:
  - Enable builds for OEA, ibm4xx and Book-E modules.
- evbsh3:
  - Add support for Alpha Project MS104-SH4, TAC T-SH7706LAN Rev.3 and TAC T-SH7706LSR Rev.1.
- hp300:
  - Switch NetBSD/hp300 to wscons with rasops.
  - Add support for framebuffers on HP362 and HP382.
  - Add support for Apollo Domain keyboard.
- hp700:
  - Added support for UFS2 boot.
  - pmap update: Support PA 2.0 machines in 32 bit mode. Improve support for machines without the PA7100LC or PA7300LC CPUs.
  - Added `ucas`(9) support for hp700.
  - com at dino now works.
  - `lcd`(4): Driver for hp700 lcds added.
  - Add support for `elroy`(4), `astro`(4), and `ssio`(4) based 64-bit machines running in 32-bit mode.

  - Update sti(4) and add pci(4) attachment.
  - Add support for HIL devices.
  - Support added for machines with the uturn(4) I/O subsystem.
- hpcarm:
  - Add support for Sharp W-ZERO3 series.
  - Use rtc_offset passed from hpcboot.
  - Bump MAXPARTITIONS from 8 to 16.
  - Switch to using Xorg servers and binaries.
  - Improve console speed on W-ZERO3 machines using write through mapping for framebuffer memory.
  - Improve install on JORNADA720.
- hpcmips:
  - Switch to using Xorg servers and binaries.
- hpcsh:
  - Switch to using Xorg servers and binaries.
  - Add sysinst(8) support.
- landisk:
  - Added UFS2 boot support.
  - Enable to build a bootable install disk image.
- luna68k:
  - Enable LUNA to load -current kernels director from firmware.
  - Make SCSI work properly.
  - Use the original OMRON font derived from 4.4BSD-Lite/luna68k rather than gallant19 which is used on Sun workstations.
  - Complete LUNA-II support.
- mac68k:
  - Add timecounter support to the CSI Hurdler Centronics Parallel Interface cpi(4) driver.
- macppc:
  - Support new-style module(7) kernel modules.
- m68k ports (amiga, atari, cesfic, hp300, luna68k, mac68k, mvme68k, news68k, next68k, sun2, sun3, x68k):
  - Re-enable m68k optimized in_cksum routine.
  - Support module(7) loading.
  - Fixed a regression in FPU_EMULATE code that caused weird awk(1) behavior.

  - Add hyperbotic and trigonometric functions to m68k FPE.

- mips ports (algor, arc, cobalt, emips, evbmips, ews4800, hpcmips, mipsco, newsmips, pmax, sbmips, sgimips):
  - Added support for 64-bit MIPS processors (O32, N32, N64 ABIs are supported).
  - Added SMP support for MIPS architecture.
  - Rewritten interrupt handling to be much more efficient, added "fast" software interrupts and kernel preemption.
  - Converted MIPS to use PCU to manage FPU.
  - Added MIPS32R2 and MIPS64R2 support.
  - Rework Atheros MIPS-based SoC support. Add initial support for AR71xx (24K) and AR9344 (74K) SoCs
  - Added support for ralink RT3883 SoC from CradlePoint Technologies.
  - Added support for MIPS DSP v2 ASE.
- ofppc:
  - Support Gigabit Ethernet Controller for MV64361 on PegasosII.
  - Support for RDB partitions.
- pmax:
  - Added UFS2 boot support.
- powerpc ports (amigappc, bebox, evbppc, ibmnws, macppc, mvmeppc, ofppc, pmppc, prep, rs6000, sandpoint, walnut):
  - Add support for using secure PLT (thereby allowing data and bss of powerpc programs and shared library to be set non-executable).
  - Added support for Book E Freescale MPC85xx (e500 core) processors. Added "fast" software interrupts support and SOFTFLOAT for PowerPC (Book E only), reworked FPU/VEC support for e500 SPE, other improvements.
  - Converted PowerPC to use PCU to manage FPU/AltiVec/SPE.
  - Initial SMP support for MPC85xx, reworked and unified IPI code, various other improvements.
  - Switch to using direct-mapped uareas (pcb + kernel stacks).
  - The generic PIC code now supports Matt Thomas's fast softintr code.
- prep:
  - Support Xserver. Tested with on-board s3(864).
- sandpoint:
  - Added support for Marvell-SKnet in netboot.
  - D-Link DSM-G600 support.
  - Add support for booting from RAID 1.
  - Iomega Storcenter support.

  - Netronix NH-230/231 and compatible NAS are supported.
- sgimips:
  - Added support for Personal IRIS 4D/20 and 4D/25 (IP6 and IP10).
  - mec(4): Added RX hardware checksum support on O2.
  -
- sh3ports (dreamcast, evbsh3, hpcsh, landisk, mmeye):
  - Add some support for SH7706.
- shark:
  - Added support for module(7) -style kernel modules.
  - chipsfb(4) now works on shark.
- sparc:
  - Switch sparc port to use wscons(4)
  - zx(4): added wscons(4) support.
  - tcx(4): support wsdisplay(4) and acceleration.
  - apc(4): a driver for the Aurora Personality Chip (APC) found on SPARCstation-4/5 and qemu. Allows to idle the CPU when in the idle loop.
  - Re-enable sparc optimized in_cksum routine.
  - Added support for module(7) -style kernel modules.
- sparc64:
  - zx(4): added wscons(4) support.
  - Add a driver for LOMlite lights out management hardware monitor and watchdog timer.
  - Add real UltraSPARC-III∗ CPU support.
  - Add initial support for Fujitsu SPARC64 systems, not yet stable.
  - ffb(4): Added EDID (Extended Display Identification Data) reading, and video mode setting support.
  - Added fast softint(9) support.
  - Add support for Enterprise systems (Ex[45]00) systems.
- x68k:
  - Switched pow(4) to use powerd(8).
  - Fix bootloader failure on X68030+060turbo.
  - Removed pow(4) and rtcalarm(8).
  - powsw(4): a revised power switch driver.
- xen:
  - Add PCI pass-through support for Xen3. Dom0 kernel gets a pciback(4) driver, to which the device specified in the pciback.hide boot parameter will attach. DomU kernels get a xpci(4) driver, to which PCI busses will attach.

  - Added XEN3PAE_DOM0, an i386 dom0 kernel with PAE support. Requires an i386PAE or x86_64 hypervisor.
  - `balloon`(4): Balloon driver now enabled for all Xen kernels.
  - Dropped Xen 2 support. Xen 3.1 or later is now required.
  - Added support for request-rx-copy in the virtual network device backend.
  - Provide experimental support for suspend/resume in NetBSD domUs.
  - Enable no-execute bit feature for i386pae and amd64 kernels.
  - Added support for feature-flush-cache in block device backend, used in block device frontend for DIOCCACHESYNC.
- zaurus:
  - Added zboot.
  - Added playback and recording support to `zaudio`(4).
  - Add Xorg support.
  - Add ioexp driver for SL-C1000.
  - Add PXA25x based Zaurus SL-C700/750/760/860 support.

Userland

- 3rd party software updates:
  - `acpi`(4): ACPICA updated to 20110623.
  - `atf`(7) is version 0.15.
  - ISC `cron`(8) 4.1.
  - OpenSSH version 1.0.1c
  - lvm2tools 2.02.44 and libdevmapper 1.02.30
  - dhcpcd-5.5.4
  - am-utils updated from head; adds UDF support and `-uu`
  - CVS 1.12.13
  - BIND 9.9.1-P4.
  - libbind 6.0-rc1.
  - FSF binutils 2.21.1a.
  - `httpd`(8): bozohttpd-20111118
  - `file`(1) 5.09
  - ntp 4.2.6p5
  - `flex`(1) version 2.5.35.
  - `gcc`(1) version 4.5.3.
  - `gdb`(1) version 7.3.1.

- GNU MP 5.0.2.
- less(1) version 444.
- libfetch 2.23
- Lua 5.1.4
- OpenLDAP 2.4.23
- SQLite 3.6.9
- tmux(1) 1.5
- trousers-0.3.8; tmp-tools 1.3.7.1
- xz(1) 5.0.0
- byacc-20110908
- tzdata2011n
- dns-sd(1): Import mDNSResponder 212.1.

- Libraries:
  - libc:
    - Add new ptree (Patricia / RADIX tree) implementation.
    - Added fopendir(3).
    - Added RFC 2045 MIME Quoted-Printable support.
    - Added glob_pattern_p(3).
    - Added memrchr(3).
    - Added stpcpy(3), stpncpy(3), and strnlen(3).
    - Added %s (seconds since the Epoch) to strptime(3), for symmetry with strftime(3).
    - Added the getdate(3) and getdate_err value, conforming to IEEE Std 1003.1 ("POSIX.1") and the Open Group Base Specifications X/Open System Interfaces extension ("XSI").
    - Added getdelim(3) and getline(3).
    - Added posix_spawn(3).
    - Added a space-efficient write-once database (cdbr(3), cdbw(3)).
    - Provide a re-entrant and thread-safe set of timezone APIs that don't require locking and can operate on user-specified timezones.
  - libm: Add f{dim,max,min}{,f,l} from FreeBSD.
  - Replaced termcap with terminfo(3).
  - perfuse: Add the PUFFS RElay to FUSE library, which bridges the kernel level API for FUSE filesystems with native PUFFS.

  - rump(3)
    - Moved VFS functionality from librump to a new library, librumpvfs.
    - Added support for remote system calls.
    - Add IPv6 support to rumpnet.
    - Support kernel config framework and devices in rumpdev.
    - rumpfs: support registering exernal files which are accessed from the host OS file system namespace instead of from within the rump virtual kernel.
    - rumpvfs: Use the unmodified kernel vnode pager.
  - bluetooth(3): updated the Bluetooth service discovery API and merged it into libbluetooth, retiring the separate libsdp.
- Installer (sysinst):
  - Default to wapbl(4) logging for UFS partitions.
  - Make sure the installer rejects UFS2 for root partitions on ports whose bootloaders don't support booting from UFS2.
- Changes to the way make(1) processes .for loops.
- makefs(8): Added support for encoding RISC OS metadata into ISO9660 file systems.
- mount(8): Added -o rump , which mounts the file system as a userspace server via puffs(4) and rump(3) instead of using a kernel drver.
- dkctl(8): Display cache, strategy and list of wedges if no command is specified on the command line.
- ar(1): Added support for "deterministic mode"

- Added crash(8), a new utility to debug kernel images, based on the in-kernel debugger, ddb(4).
- eject(1) now has amd(8) support.
- mtree(8): add -S option to sort entries.
- Imported netpgp(1), a BSD-licensed PGP implementation.
- install(1): Added support for writing sha256, sha384, or sha512 hashes to a METALOG.
- rpcgen(1): Added support for hyper and quad keywords.
- New wakeonlan(8) command to send Wake-on-LAN packets to machines on the local Ethernet.
- Added nbperf(1), a minimal perfect hash function generator.
- unzip(1): a libarchive-based unzip frontend.
- apropos(1): new implementation using SQLite Full Text Index.
- audiocfg(1): new tool to control audio defaults.
- devpubd(8): Add a device publishing daemon for automatic device node creation.
- disklabel(8): Disable COMPAT_386BSD_MBRPART.
- dkctl(8): Display cache, strategy, and list of wedges if no command is specified in the command line.

- `fincore`(1): Add a utility to query file cache.
- `ifconfig`(8): Add the `linkstr` command which can be used to communicate an arbitrary string with the interface driver.
- `man`(1): display a manpage when specified with a path.
- `mkubootimage`(1): Added a tool to generate u-boot kernel images.

Besides this list, there have also been innumerable bug fixes and miscellaneous enhancements.

Components removed from NetBSD

In this release of NetBSD, the following software components have been removed from the system. Some were not useful anymore, or their utility did not justify the maintenance overhead. Others were not working properly and there was a lack of interest in fixing them.

- Support for Xen 2 has been dropped. Xen 3.1 or later is now required to run NetBSD.
- Support for LKMs (Loadable Kernel Modules) was dropped; use the new `module`(7) framework.
- "softdep"

  support is no longer available in FFS; use `wapbl`(4) logging instead.
- The

  playstation2:
  http://www.NetBSD.org/ports/playstation2/

  port.
- tn3270 was removed from base; it is available in pkgsrc.
- Support for Darwin, Mach, IRIX and PECOFF emulation was retired.

Features to be removed in a later release

The following features are to be removed from NetBSD in the future:

- `dhclient`(8) and `rtsol`(8) in favor of `dhcpcd`(8).
- `groff`(1) is being phased out. Man pages are handled with `mandoc`(1), and `groff`(1) can still be found in pkgsrc as `textproc/groff`.
- `kame_ipsec`(4) has been replaced by `fast_ipsec`(4). The option to use the old implementation (see `options`(4)) will be removed in the next NetBSD release.

The NetBSD Foundation

The NetBSD Foundation is a tax exempt, not-for-profit 501(c)(3) corporation that devotes itself to the traditional goals and Spirit of the NetBSD Project and owns the trademark of the word "NetBSD". It supports the design, development, and adoption of NetBSD worldwide. More information on the NetBSD Foundation, its composition, aims, and work can be found at: **http://www.NetBSD.org/foundation/**

Sources of NetBSD

Refer to **http://www.NetBSD.org/mirrors/**

NetBSD 6.0 Release Contents

The root directory of the NetBSD 6.0 release is organized as follows:

`.../NetBSD-6.0/`

`CHANGES` Changes between the 5.0 and 6.0 releases.

`CHANGES-6.0` Changes between the initial 6.0 branch and final release of 6.0.

`CHANGES.prev` Changes in previous NetBSD releases.

`LAST_MINUTE` Last minute changes and notes about the release.

`README.files` README describing the distribution's contents.

`source/` Source distribution sets; see below.

In addition to the files and directories listed above, there is one directory per architecture, for each of the architectures for which NetBSD 6.0 has a binary distribution.

The source distribution sets can be found in subdirectories of the `source` subdirectory of the distribution tree. They contain the complete sources to the system. The source distribution sets are as follows:

gnusrc This set contains the "gnu" sources, including the source for the compiler, assembler, groff, and the other GNU utilities in the binary distribution sets.

sharesrc This set contains the "share" sources, which include the sources for the man pages not associated with any particular program; the sources for the typesettable document set; the dictionaries; and more.

src This set contains all of the base NetBSD 6.0 sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.

syssrc This set contains the sources to the NetBSD 6.0 kernel for all architectures as well as the `config`(1) utility.

xsrc This set contains the sources to the X Window System.

All the above source sets are located in the `source/sets` subdirectory of the distribution tree.

The source sets are distributed as compressed tar files. Except for the **pkgsrc** set, which is traditionally unpacked into `/usr/pkgsrc`, all sets may be unpacked into `/usr/src` with the command:

```
# cd / ; tar -zxpf set_name.tgz
```

In each of the source distribution set directories, there are files which contain the checksums of the files in the directory:

`MD5` MD5 digests in the format produced by the command:
cksum -a MD5 *file*.

`SHA512` SHA512 digests in the format produced by the command:
cksum -a SHA512 *file*.

The SHA512 digest is safer, but MD5 checksums are provided so that a wider range of operating systems can check the integrity of the release files.

NetBSD/sun2 subdirectory structure

The sun2-specific portion of the NetBSD 6.0 release is found in the `sun2` subdirectory of the distribution: `.../NetBSD-6.0/sun2/`. It contains the following files and directories:

`INSTALL.html`
`INSTALL.ps`
`INSTALL.txt`
`INSTALL.more` Installation notes in various file formats, including this file. The `.more` file contains underlined text using the `more`(1) conventions for indicating italic and bold display.

`binary/`
`kernel/`
`netbsd-GENERIC.gz`
A gzipped NetBSD kernel containing code for everything supported in this release.
`sets/` sun2 binary distribution sets; see below.

`installation/`
`miniroot/` sun2 miniroot file system image; see below.
`misc/` Miscellaneous sun2 installation utilities; see installation section below.
`netboot/` Two programs needed to boot sun2 kernels over the network.
`tapeimage/`
Tape boot program, and two shell scripts used to prepare tapes.

Binary distribution sets

The NetBSD sun2 binary distribution sets contain the binaries which comprise the NetBSD 6.0 release for sun2. The binary distribution sets can be found in the `sun2/binary/sets` subdirectory of the NetBSD 6.0 distribution tree, and are as follows:

base The NetBSD 6.0 sun2 **base** binary distribution. You *must* install this distribution set. It contains the base NetBSD utilities that are necessary for the system to run and be minimally functional.

comp Things needed for compiling programs. This set includes the system include files (`/usr/include`) and the various system libraries (except the shared libraries, which are included as part of the **base** set). This set also includes the manual pages for all of the utilities it contains, as well as the system call and library manual pages.

etc This distribution set contains the system configuration files that reside in `/etc` and in several other places. This set *must* be installed if you are installing the system from scratch, but should *not* be used if you are upgrading.

games This set includes the games and their manual pages.

kern-GENERIC
This set contains a NetBSD/sun2 6.0 `GENERIC` kernel, named `/netbsd`. You *must* install this distribution set.

man This set includes all of the manual pages for the binaries and other software contained in the **base** set. Note that it does not include any of the manual pages that are included in the other sets.

misc This set includes the system dictionaries, the typesettable document set, and other files from `/usr/share`.

text This set includes NetBSD's text processing tools, including `groff`(1), all related programs, and their manual pages.

NetBSD maintains its own set of sources for the X Window System in order to assure tight integration and compatibility. NetBSD/sun2 currently does not ship with an X server or X clients.

The sun2 binary distribution sets are distributed as gzipped tar files named with the extension **.tgz**, e.g. `base.tgz`.

The instructions given for extracting the source sets work equally well for the binary sets, but it is worth noting that if you use that method, the filenames stored in the sets are relative and therefore the files are extracted *below the current directory*. Therefore, if you want to extract the binaries into your system, i.e. replace the system binaries with them, you have to run the **`tar -xzpf`** command from the root directory (`/`) of your system.

There is a collection of Sun2 kernels in the `sun2/binary/kernel` subdirectory of the NetBSD 6.0 distribution. The one named `netbsd-RAMDISK.gz` contain a root file system image and should only be used for the initial installation. The others are included for convenience. (Most people will want to use `netbsd-GENERIC.gz` or `netbsd-FOURMEG.gz` as appropriate.) Please note that these kernels are simply gzipped and are not tar archives.

Note: Each directory in the sun2 binary distribution also has its own checksum files, just as the source distribution does.

NetBSD/sun2 System Requirements and Supported Devices

NetBSD/sun2 6.0 runs on Multibus Sun2 machines, including:

2/120 2/170 2/100U

The minimal configuration requires 4 MB of RAM and about 130 MB of disk space. To install the entire system requires much more disk space. To compile the system, more RAM is recommended. Good performance requires 7 MB of RAM. A good rule of thumb is to have a swap partition twice the size of the amount of RAM in your machine. You will probably want to compile your own kernel, as `GENERIC` is large and bulky to accommodate all people.

Note that the sun2 installation procedure uses a *miniroot* image which is placed into the swap area of the disk. The swap partition must be at least as large as the miniroot image (12 MB).

Supported hardware

- Serial ports (RS232)
  - built-in ttya, ttyb
  - ttyc, ttyd, ttye, ttyf on first sc SCSI
  - ttyg, ttyh, ttyi, ttyj on second sc SCSI
- Video adapters
  - bwtwo
- Network interfaces:
  - Multibus Intel Ethernet
  - Multibus 3Com Ethernet
- SCSI
  - Most SCSI disks, tapes, CD-ROMs, etc
  - Multibus Sun-2 SCSI (*sc*)
- Input devices
  - Sun keyboard and mouse
- Miscellaneous
  - Battery-backed real-time clock.

If it's not on this list, there is no support for it in this release.

Getting the NetBSD System on to Useful Media

Installation is supported from several media types, including:

- Tape
- NFS
- CD-ROM
- FTP

Note: Installing on a 'bare' machine requires some bootable device; either a tape drive or Sun-compatible NFS server.

The procedure for transferring the distribution sets onto installation media depends on the type of media. Instructions for each type of media are given below.

In order to create installation media, you will need all the files in the directory

```
.../NetBSD-6.0/sun2/
```

Creating boot/install tapes

Installing from tape is the simplest method of all. This method uses two tapes; one called the *boot* tape, and another called the *install* tape.

The boot tape is created as follows:

```
# cd .../NetBSD-6.0/sun2/installation/tapeimage
# sh MakeBootTape /dev/nrst0
```

The install tape is created as follows:

```
# cd .../NetBSD-6.0/sun2/installation/tapeimage
# sh MakeInstallTape /dev/nrst0
```

If the tapes do not work as expected, you may need to explicitly set the EOF mark at the end of each tape segment. It may also be necessary to use the **conv=osync** argument to dd(1). Note that this argument is incompatible with the **bs=** argument. Consult the tape-related manual pages on the system where the tapes are created for more details.

Boot/Install from NFS server

If your machine has a disk and network connection, but no tape drive, it may be convenient for you to install NetBSD over the network. This involves temporarily booting your machine over NFS, just long enough so you can initialize its disk. This method requires that you have access to an NFS server on your network so you can configure it to support diskless boot for your machine. Configuring the NFS server is normally a task for a system administrator, and is not trivial.

If you are using a NetBSD system as the boot-server, have a look at the diskless(8) manual page for guidelines on how to proceed with this. If the server runs another operating system, consult the documentation that came with it (i.e. add_client(8) on SunOS).

When instructed to boot over the network, your sun2 expects to be able to download first and second stage bootstrap programs via ND, the Network Disk protocol. The ndbootd(8) program will attempt to serve a second-stage bootstrap file using a name derived from the machine's recently acquired IP address and an extension which corresponds to the machine architecture. (It may be handy to have a hexadecimal calculator for this next step.) The filename prefix is created by converting the machine's assigned IP address into hexadecimal, most-significant octet first, using uppercase characters for the non-decimal (A-F) digits. The filename suffix used by all sun2 machines is SUN2.

For example, a sun2 which has been assigned IP address 130.115.144.11 will be served a second-stage bootstrap file named 8273900B.SUN2. Normally, this file is a symbolic link to the NetBSD/sun2 netboot program, which should be located in a place where the ndbootd(8) daemon can find it. The netboot pro-

gram may be found in the `installation/netboot` directory of this distribution.

The netboot program will query a bootparamd server to find the NFS server address and path name for its root, and then load a kernel from that location. The server should have a copy of the `netbsd-RAMDISK` kernel in the root area for your client, hard-linked under the names `netbsd` and `vmunix` (no other files are needed in the client root) and `/etc/bootparams` on the server should have an entry for your client and its root directory. The client will need access to the miniroot image, which can be provided using NFS or remote shell.

If you will be installing NetBSD on several clients, it may be useful to know that you can use a single NFS root for all the clients as long as they only use the `netbsd-RAMDISK` kernel. There will be no conflict between clients because the RAM-disk kernel will not use the NFS root. No swap file is needed; the RAM-disk kernel does not use that either.

Install/Upgrade from CD-ROM

This method requires that you boot from another device (i.e. tape or network, as described above). You may need to make a boot tape on another machine using the files provided on the CD-ROM. Once you have booted netbsd-rd (the RAM-disk kernel) and loaded the miniroot, you can load any of the distribution sets directly from the CD-ROM. The **`install`** program in the miniroot automates the work required to mount the CD-ROM and extract the files.

Install/Upgrade via FTP

This method requires that you boot from another device (i.e. tape or network, as described above). You may need to make a boot tape on another machine using the files in `installation/tapeimage` and `binary/kernel` (which you get via FTP). Once you have booted `netbsd-RAMDISK` (the RAM-disk kernel) and loaded the miniroot, you can load any of the distribution sets over the net using FTP. The **`install`** program in the miniroot automates the work required to configure the network interface and transfer the files.

The preparations for this installation/upgrade method are easy; all you make sure that there's some FTP site from which you can retrieve the NetBSD distribution when you're about to install or upgrade. You need to know the numeric IP address of that site, and, if it's not on a network directly connected to the machine on which you're installing or upgrading NetBSD, you need to know the numeric IP address of the router closest to the NetBSD machine. Finally, you need to know the numeric IP address of the NetBSD machine itself.

Preparing your System for NetBSD installation

Sun2 machines usually need little or no preparation before installing NetBSD, other than the usual, well advised precaution of *backing up all data* on any attached storage devices.

You will need to know the SCSI target ID of the drive on which you will install NetBSD.

Note: SunOS on the sun2 uses confusing names for the SCSI devices: target 1 is `sd2`, target 2 is `sd4`, etc.

Installing the NetBSD System

Installing NetBSD is a relatively complex process, but if you have this document in hand it should not be too difficult.

There are several ways to install NetBSD onto your disk. If your machine has a tape drive the easiest way is **Installing from tape** (details below). If your machine is on a network with a suitable NFS server, then **Installing from NFS** is the next best method. Otherwise, if you have another Sun machine running SunOS you can initialize the disk on that machine and then move the disk. (Installing from SunOS is not recommended.)

Installing from tape

Create the NetBSD/sun2 6.0 boot tape as described in the section entitled **Creating boot/install tapes** and boot the tape. At the PROM monitor prompt, use one of the commands:

```
>b st()
>b st(0,8,0)
```

The first example will use the tape on SCSI target 4, where the second will use SCSI target 5. The > is the monitor prompt.

After the tape loads, you should see many lines of configuration messages, and then the following 'welcome' screen:

```
          Welcome to the NetBSD/sun2 RAMDISK root!

This environment is designed to do only three things:
1:  Partition your disk (use the command:  edlabel /dev/rsd0c)
2:  Copy a miniroot image into the swap partition  (/dev/rsd0b)
3:  Reboot (using the swap partition, i.e. /dev/sd?b).

Note that the sun2 firmware cannot boot from a partition located
more than 1 GB from the beginning of the disk, so the swap partition
should be completely below the 1 GB boundary.

Copying the miniroot can be done several ways, allowing
the source of the miniroot image to be on any of these:
    boot tape,  NFS server, TFTP server, rsh server

The easiest is loading from tape, which is done as follows:
        mt -f /dev/nrst0 rewind
        mt -f /dev/nrst0 fsf 2
        dd if=/dev/nrst0 of=/dev/rsd0b bs=32k conv=sync
(For help with other methods, please see the install notes.)

To reboot using the swap partition, first use "halt",
then at the PROM monitor prompt use a command like:
        b sd(,,1) -s

To view this message again, type:  cat /.welcome
```

Copy the miniroot as described in the welcome message, and reboot from that just installed miniroot. See the section entitled **Booting the miniroot** for details.

Installing from NFS

Before you can install from NFS, you must have already configured your NFS server to support your machine as a diskless client. Instructions for configuring the server are found in the section entitled **Getting the NetBSD System onto Useful Media** above.

First, at the Sun PROM monitor prompt, enter a boot command using the network interface as the boot device. If your machine has Intel Ethernet, this is *ie*, and if it has 3Com Ethernet, this is *ec*. Examples:

```
>b ie() -s
>b ec() -s
```

After the boot program loads the RAMDISK kernel, you should see the welcome screen as shown in the **Installing from tape** section above. You must configure the network interface before you can use any network resources. For example the command:

```
ssh> ifconfig ie0 inet 192.233.20.198 up
```

will bring up the network interface with that address. The next step is to copy the miniroot from your server. This can be done using either NFS or remote shell. (In the examples that follow, the server has IP address 192.233.20.195.) You may then need to add a default route if the server is on a different subnet:

```
ssh> route add default 192.233.20.255 1
```

You can look at the route table using:

```
ssh> route show
```

Now mount the NFS file system containing the miniroot image:

```
ssh> mount -o rdonly,-r=1024 192.233.20.195:/server/path /mnt
```

The procedure is simpler and much faster if you have space for an expanded (not compressed) copy of the miniroot image. In that case:

```
ssh> dd if=/mnt/miniroot.fs of=/dev/rsd0b bs=8k
```

Otherwise, you will need to use **zcat** to expand the miniroot image while copying. This is tricky because the **ssh** program (small shell) does not handle sh(1) pipeline syntax. Instead, you first run the reader in the background with its input set to /dev/pipe and then run the other program in the foreground with its output to /dev/pipe. The result looks like this:

```
ssh> run -bg dd if=/dev/pipe of=/dev/rsd0b obs=8k
ssh> run -o /dev/pipe zcat /mnt/install/miniroot.fs.gz
```

To load the miniroot using rsh to the server, you would use a pair of commands similar to the above. Here is another example:

```
ssh> run -b dd if=/dev/pipe of=/dev/rsd0b obs=8k
ssh> run -o /dev/pipe rsh 192.233.20.195 zcat miniroot.fs.gz
```

Note that decompression on a sun2 is *extremely* slow, be prepared to wait. For this reason, expanding the miniroot image on the NFS server is highly recommended.

Once the miniroot has been copied using one of the above methods, you reboot from that just installed miniroot. See the section entitled **Booting the miniroot** for details.

Installing from SunOS

To install NetBSD/sun2 onto a machine already running SunOS, you will need the miniroot image (miniroot.fs.gz) and some means to decompress it.

First, boot SunOS and place the miniroot file onto the hard drive. If you do not have **gzip** for SunOS, you will need to decompress the image elsewhere before you can use it.

Next, bring SunOS down to single user mode to insure that nothing will be using the swap space on your drive. To be extra safe, reboot the machine into single-user mode rather than using the **shutdown** command.

Now copy the miniroot image onto your swap device (here /dev/rsd0b) with the command

```
gzip -dc miniroot.fs.gz | dd of=/dev/rsd0b obs=32k
```

or if you have already decompressed the miniroot

```
dd if=miniroot.fs of=/dev/rsd0b obs=32k
```

Finally, reboot the machine and instruct the PROM to boot from the swap device as described in the next section.

Booting the Miniroot

If the machine is not already at the PROM monitor, run the `halt` command.

If the miniroot was installed on partition '`b`' of the disk with SCSI target ID=0 then the PROM boot command would be:

```
>b sd(0,0,1) -s
```

With SCSI target ID=2, the PROM is:

```
>b sd(0,10,1) -s
```

The numbers in parentheses above are:

1. controller (usually zero)
2. unit number (SCSI-ID * 8, in hexadecimal)
3. partition number

Miniroot install program

The miniroot's install program is very simple to use. It will guide you through the entire process, and is well automated. Additional improvements are planned for future releases.

The miniroot's install program will:

- Allow you to place disklabels on additional disks. The disk we are installing on should already have been partitioned using the RAMDISK kernel.
- Create file systems on target partitions.
- Allow you to set up your system's network configuration. Remember to specify host names without the domain name appended to the end. For example use `foo` instead of `foo.bar.org`. If, during the process of configuring the network interfaces, you make a mistake, you will be able to re-configure that interface by simply selecting it for configuration again.
- Mount target file systems. You will be given the opportunity to manually edit the resulting `/etc/fstab`.
- Extract binary sets from the media of your choice.
- Copy configuration information gathered during the installation process to your root file system (`/`).
- Make device nodes in your root file system under `/dev`.
- Copy a new kernel onto your root partition (`/`).
- Install a new boot block.
- Check your file systems for integrity.

First-time installation on a system through a method other than the installation program is possible, but strongly discouraged.

Post installation steps

Once you've got the operating system running, there are a few things you need to do in order to bring the system into a properly configured state. The most important steps are described below.

1. Configuring `/etc/rc.conf`

   If you or the installation software haven't done any configuration of `/etc/rc.conf` (**sysinst** usually will), the system will drop you into single user mode on first reboot with the message

   ```
   /etc/rc.conf is not configured. Multiuser boot aborted.
   ```

   and with the root file system (`/`) mounted read-only. When the system asks you to choose a shell, simply press `RETURN` to get to a `/bin/sh` prompt. If you are asked for a terminal type, respond with **vt220** (or whatever is appropriate for your terminal type) and press `RETURN`. You may need to type one of the following commands to get your delete key to work properly, depending on your keyboard:

   ```
   # stty erase '^h'
   # stty erase '^?'
   ```

   At this point, you need to configure at least one file in the `/etc` directory. You will need to mount your root file system read/write with:

   ```
   # /sbin/mount -u -w /
   ```

   Change to the `/etc` directory and take a look at the `/etc/rc.conf` file. Modify it to your tastes, making sure that you set `rc_configured=YES` so that your changes will be enabled and a multi-user boot can proceed. Default values for the various programs can be found in `/etc/defaults/rc.conf`, where some in-line documentation may be found. More complete documentation can be found in `rc.conf`(5).

   When you have finished editing `/etc/rc.conf`, type **exit** at the prompt to leave the single-user shell and continue with the multi-user boot.

   Other values that may need to be set in `/etc/rc.conf` for a networked environment are *hostname* and possibly *defaultroute*. You may also need to add an *ifconfig_int* for your ⟨int⟩ network interface, along the lines of

   ```
   ifconfig_le0="inet 192.0.2.123 netmask 255.255.255.0"
   ```

   or, if you have *myname.my.dom* in `/etc/hosts`:

   ```
   ifconfig_le0="inet myname.my.dom netmask 255.255.255.0"
   ```

   To enable proper hostname resolution, you will also want to add an `/etc/resolv.conf` file or (if you are feeling a little more adventurous) run `named`(8). See `resolv.conf`(5) or `named`(8) for more information. Instead of manually configuring network and naming service, DHCP can be used by setting `dhclient=YES` in `/etc/rc.conf`.

   Other files in `/etc` that may require modification or setting up include `/etc/mailer.conf`, `/etc/nsswitch.conf`, and `/etc/wscons.conf`.

2. Logging in

   After reboot, you can log in as `root` at the login prompt. Unless you've set a password in **sysinst**, there is no initial password. You should create an account for yourself (see below) and protect it and the "root" account with good passwords. By default, root login from the network is disabled (even via `ssh`(1)). One way to become root over the network is to log in as a different user that belongs to group "wheel" (see `group`(5)) and use `su`(1) to become root.

3. Adding accounts

   Use the `useradd`(8) command to add accounts to your system. Do not edit `/etc/passwd` directly! See `vipw`(8) and `pwd_mkdb`(8) if you want to edit the password database.

4. Installing third party packages

   If you wish to install any of the software freely available for UNIX-like systems you are strongly advised to first check the NetBSD package system, pkgsrc. pkgsrc automatically handles any changes necessary to make the software run on NetBSD. This includes the retrieval and installation of any other packages on which the software may depend.

   - More information on the package system is available at
     http://www.NetBSD.org/docs/software/packages.html
   - A list of available packages suitable for browsing is at
     ftp://ftp.NetBSD.org/pub/NetBSD/packages/pkgsrc/README.html
   - Precompiled binaries can be found at
     ftp://ftp.NetBSD.org/pub/pkgsrc/packages/NetBSD/
     usually in the `sun2/6.0/All` subdir. If you installed `pkgin`(1) in the **sysinst** post-installation configuration menu, you can use it to automatically install binary packages over the network. Assuming that `/usr/pkg/etc/pkgin/repositories.conf` is correctly configured, you can install them with the following commands:

     ```
     # pkgin install tcsh
     # pkgin install bash
     # pkgin install perl
     # pkgin install apache
     # pkgin install kde
     # pkgin install firefox
     ...
     ```

     Note: Some mirror sites don't mirror the `/pub/pkgsrc` directory. If you would like to use such mirrors, you could also try the `/pub/NetBSD/packages/current-packages/NetBSD/sun2/6.0/All` directory, which may have the same contents.

     The above commands will install the Tenex-csh and Bourne Again shells, the Perl programming language, Apache web server, KDE desktop environment and the Firefox web browser as well as all the packages they depend on.

   - If you did not install it from the **sysinst** post-installation configuration menu, the `pkgsrc`(7) framework for compiling packages can be obtained by retrieving the file
     ftp://ftp.NetBSD.org/pub/NetBSD/packages/pkgsrc.tar.gz
     It is typically extracted into `/usr/pkgsrc` (though other locations work fine) with the commands:

     ```
     # cd /usr
     # tar -zxpf pkgsrc.tar.gz
     ```

     After extracting, see the `doc/pkgsrc.txt` file in the extraction directory (e.g., `/usr/pkgsrc/doc/pkgsrc.txt`) for more information.

5. Misc

   - Edit `/etc/mail/aliases` to forward root mail to the right place. Don't forget to run `newaliases`(1) afterwards.
   - The `/etc/postfix/main.cf` file will almost definitely need to be adjusted. If you prefer a different MTA, then install it using pkgsrc or by hand and adjust `/etc/mailer.conf`.
   - Edit `/etc/rc.local` to run any local daemons you use.

- Many of the `/etc` files are documented in section 5 of the manual; so just invoking

  ```
  # man 5 filename
  ```

  is likely to give you more information on these files.

Upgrading a previously-installed NetBSD System

It is possible to easily upgrade your existing NetBSD/sun2 system using the upgrade program in the miniroot. If you wish to upgrade your system by this method, simply select the **upgrade** option once the miniroot has booted. The upgrade program with then guide you through the procedure. The upgrade program will:

- Enable the network based on your system's current network configuration.
- Mount your existing file systems.
- Extract binary sets from the media of your choice.
- Make new device nodes in your root file system under `/dev`.
- Copy a new kernel onto your root partition (`/`).

  Note: the existing kernel *will not* be backed up; doing so would be pointless, since older kernels may not be capable of running NetBSD 6.0 executables.
- Install a new boot block.
- Check your file systems for integrity.

Using the miniroot's upgrade program is the preferred method of upgrading your system.

However, it is possible to upgrade your system manually. To do this, follow the following procedure:

- Place *at least* the **base** binary set in a file system accessible to the target machine. A local file system is preferred, since the NFS subsystem in the NetBSD 6.0 kernel may be incompatible with your old binaries.
- Back up your pre-existing kernel and copy the 6.0 kernel into your root partition (`/`).
- Reboot with the 6.0 kernel into single-user mode. (Otherwise you can not install the boot block.)
- Check all file systems:

  ```
  # /sbin/fsck -pf
  ```
- Mount all local file systems:

  ```
  # /sbin/mount -a -t nonfs
  ```
- If you keep `/usr` or `/usr/share` on an NFS server, you will want to mount those file systems as well. To do this, you will need to enable the network:

  ```
  # sh /etc/rc.d/network start
  ```
- Make sure you are in the root file system (`/ and extract the`) **base** binary set:

  ```
  # cd /
  # pax -zrvpe -f Ar /path/to/base.tgz
  ```
- Install a new boot block: (assuming root is on `/dev/rsd0a`)

  ```
  # cd /usr/mdec
  # cp -p ./ufsboot /mnt/ufsboot
  # sync ; sleep 1 ; sync
  # /usr/sbin/installboot -v /dev/rsd0a bootxx /ufsboot
  ```

- Sync the file systems:

  sync

- At this point you may extract any other binary sets you may have placed on local file systems, or you may wish to extract additional sets at a later time. To extract these sets, use the following commands:

  # **cd /**
  # **pax -zrvpe -f** *path_to_set*

Note: You *should not* extract the **etc** set if upgrading. Instead, you should extract that set into another area and carefully merge the changes by hand.

Compatibility Issues With Previous NetBSD Releases

Users upgrading from previous versions of NetBSD may wish to bear the following problems and compatibility issues in mind when upgrading to NetBSD 6.0.

Note that **sysinst** will automatically invoke

```
postinstall fix
```

and thus all issues that are fixed by **postinstall** by default (see below) will be handled.

Issues affecting an upgrade from NetBSD 5.x releases

The pthread libraries from previous versions of NetBSD require that the `sysctl`(3) node `kern.no_sa_support` be set to `0`. This affects the following environments:

- Running a 5.0 kernel with an older userland.
- Running an older userland inside a chroot'ed environment on a 5.0 system.
- Running older statically linked pthread applications.

The 5.0 kernel defaults to `0` for `kern.no_sa_support`, which covers the first case. However, please note that a full installation of 5.0 (either from scratch or through an upgrade) will set `kern.no_sa_support` to 1 during the boot process. This means that for the last two cases, you will have to manually set `kern.no_sa_support` to `0`, using either the `sysctl`(8) command or through `sysctl.conf`(5).

The implementation of SHA2-HMAC in KAME_IPSEC as used in NetBSD-5 and before did not comply to current standards. FAST_IPSEC does, with the result that old and new systems cannot communicate over IPSEC, if one of the affected authentication algorithms (hmac_sha256, hmac_sha384, hmac_sha512) is used.

Issues affecting an upgrade from NetBSD 4.x releases

The following issues can generally be resolved by running *postinstall* with the **etc** set:

```
postinstall -s /path/to/etc.tgz check
postinstall -s /path/to/etc.tgz fix
```

Issues fixed by *postinstall*:

- Various files in `/etc` need upgrading. These include:
  - `/etc/defaults/*`
  - `/etc/mtree/*`
  - `/etc/daily`
  - `/etc/weekly`

- /etc/monthly
- /etc/security
- /etc/rc.subr
- /etc/rc
- /etc/rc.shutdown
- /etc/rc.d/*
- /etc/envsys.conf

The following issues need to be resolved manually:

- The users '_httpd' and '_timedc' and the groups '_httpd' and '_timedc' need to be created.
- Unprivileged use of the mount(8) command now requires the *nosuid* and *nodev* options to be explicitly specified. Previously, these options were automatically enforced even if they were not explicitly specified.
- A number of things have been removed from the NetBSD 6.0 release. See the "Components removed from NetBSD" section near the beginning of this document for a list.

Using online NetBSD documentation

Documentation is available if you installed the manual distribution set. Traditionally, the "man pages" (documentation) are denoted by 'name(section)'. Some examples of this are

- intro(1),
- man(1),
- apropos(1),
- passwd(1), and
- passwd(5).

The section numbers group the topics into several categories, but three are of primary interest: user commands are in section 1, file formats are in section 5, and administrative information is in section 8.

The *man* command is used to view the documentation on a topic, and is started by entering **man** [*section*] *topic*. The brackets [] around the section should not be entered, but rather indicate that the section is optional. If you don't ask for a particular section, the topic with the lowest numbered section name will be displayed. For instance, after logging in, enter

```
# man passwd
```

to read the documentation for passwd(1). To view the documentation for passwd(5), enter

```
# man 5 passwd
```

instead.

If you are unsure of what man page you are looking for, enter

```
# apropos subject-word
```

where *subject-word* is your topic of interest; a list of possibly related man pages will be displayed.

Administrivia

If you've got something to say, do so! We'd like your input. There are various mailing lists available via the mailing list server at majordomo@NetBSD.org. To get help on using the mailing list server, send mail to that address with an empty body, and it will reply with instructions. See **http://www.NetBSD.org/mailinglists/** for a web interface.

There are various mailing lists set up to deal with comments and questions about this release. Please send comments to: netbsd-comments@NetBSD.org.

To report bugs, use the `send-pr`(1) command shipped with NetBSD, and fill in as much information about the problem as you can. Good bug reports include lots of details.

Bugs also can be submitted and queried with the web interface at
http://www.NetBSD.org/support/send-pr.html

There are also port-specific mailing lists, to discuss aspects of each port of NetBSD. Use majordomo to find their addresses, or visit **http://www.NetBSD.org/mailinglists/**

If you're interested in doing a serious amount of work on a specific port, you probably should contact the 'owner' of that port (listed below).

If you'd like to help with this effort, and have an idea as to how you could be useful, send us mail or subscribe to: `netbsd-users@NetBSD.org`.

As a favor, please avoid mailing huge documents or files to these mailing lists. Instead, put the material you would have sent up for FTP or WWW somewhere, then mail the appropriate list about it, or, if you'd rather not do that, mail the list saying you'll send the data to those who want it.

Thanks go to

- The former members of UCB's Computer Systems Research Group, including (but not limited to):

  Keith Bostic
  Ralph Campbell
  Mike Karels
  Marshall Kirk McKusick

  for their work on BSD systems, support, and encouragement.
- The Internet Systems Consortium, Inc. for hosting the NetBSD FTP, CVS, AnonCVS, mail, mail archive, GNATS, SUP, Rsync and WWW servers.
- The Internet Research Institute in Japan for hosting the server which runs the CVSweb interface to the NetBSD source tree.
- The Luleå Academic Computer Society for providing the backup services server.
- The Columbia University Computer Science Department for hosting the NYC build cluster.
- The Western Washington University Computer Science Department for running the WWU build cluster.
- The many organizations that provide NetBSD mirror sites.
- Without CVS, this project would be impossible to manage, so our hats go off to Brian Berliner, Jeff Polk, and the various other people who've had a hand in making CVS a useful tool.
- We list the individuals and organizations that have made donations or loans of hardware and/or money, to support NetBSD development, and deserve credit for it at **http://www.NetBSD.org/donations/** (If you're not on that list and should be, tell us! We probably were not able to get in touch with you, to verify that you wanted to be listed.)
- Finally, we thank all of the people who've put sweat and tears into developing NetBSD since its inception in January, 1993. (Obviously, there are a lot more people who deserve thanks here. If you're one of them, and would like to be mentioned, tell us!)

We are...

(in alphabetical order)

The NetBSD core group:

Alan Barrett	apb@NetBSD.org
Alistair Crooks	agc@NetBSD.org
Matthew Green	mrg@NetBSD.org
Chuck Silvers	chs@NetBSD.org
Matt Thomas	matt@NetBSD.org
YAMAMOTO Takashi	yamt@NetBSD.org
Christos Zoulas	christos@NetBSD.org

The portmasters (and their ports):

Reinoud Zandijk	reinoud	**acorn32**
Matt Thomas	matt	**alpha**
Ignatios Souvatzis	is	**amiga**
Ignatios Souvatzis	is	**amigappc**
Noriyuki Soda	soda	**arc**
Julian Coleman	jdc	**atari**
Matthias Drochner	drochner	**cesfic**
Erik Berls	cyber	**cobalt**
Antti Kantee	pooka	**emips**
Simon Burge	simonb	**evbmips**
Steve Woodford	scw	**evbppc**
Izumi Tsutsui	tsutsui	**ews4800mips**
Izumi Tsutsui	tsutsui	**hp300**
Nick Hudson	skrll	**hp700**
Valeriy E. Ushakov	uwe	**hpcsh**
Matt Thomas	matt	**ibmnws**
Gavan Fantom	gavan	**iyonix**
Valeriy E. Ushakov	uwe	**landisk**
Izumi Tsutsui	tsutsui	**luna68k**
Scott Reynolds	scottr	**mac68k**
Michael Lorenz	macallan	**macppc**
Steve Woodford	scw	**mvme68k**
Steve Woodford	scw	**mvmeppc**
Matt Thomas	matt	**netwinder**
Izumi Tsutsui	tsutsui	**news68k**
Tim Rightnour	garbled	**ofppc**
Simon Burge	simonb	**pmax**
Tim Rightnour	garbled	**prep**
Tim Rightnour	garbled	**rs6000**
Tohru Nishimura	nisimura	**sandpoint**
Simon Burge	simonb	**sbmips**
Søren Jørvang	soren	**sgimips**
SAITOH Masanobu	msaitoh	**sh3**
Martin Husemann	martin	**sparc64**
Anders Magnusson	ragge	**vax**
NISHIMURA Takeshi	nsmrtks	**x68k**
Manuel Bouyer	bouyer	**xen**

The NetBSD 6.0 Release Engineering team:

Stephen Borrill	sborrill@NetBSD.org
Manuel Bouyer	bouyer@NetBSD.org
David Brownlee	abs@NetBSD.org
James Chacon	jmc@NetBSD.org
Julian Coleman	jdc@NetBSD.org
Alistair G. Crooks	agc@NetBSD.org
Håvard Eidnes	he@NetBSD.org
John Heasley	heas@NetBSD.org
Martin Husemann	martin@NetBSD.org
Soren Jacobsen	snj@NetBSD.org
Phil Nelson	phil@NetBSD.org
Jeremy C. Reed	reed@NetBSD.org
Jeff Rizzo	riz@NetBSD.org
SAITOH Masanobu	msaitoh@NetBSD.org

NetBSD Developers:

Nathan Ahlstrom	nra@NetBSD.org
Steve Allen	wormey@NetBSD.org
Jukka Andberg	jandberg@NetBSD.org
Julian Assange	proff@NetBSD.org
Lennart Augustsson	augustss@NetBSD.org
Zafer Aydogan	zafer@NetBSD.org
Christoph Badura	bad@NetBSD.org
Marc Balmer	mbalmer@NetBSD.org
Bang Jun-Young	junyoung@NetBSD.org
Dieter Baron	dillo@NetBSD.org
Robert V. Baron	rvb@NetBSD.org
Alan Barrett	apb@NetBSD.org
Grant Beattie	grant@NetBSD.org
Erik Berls	cyber@NetBSD.org
Hiroyuki Bessho	bsh@NetBSD.org
John Birrell	jb@NetBSD.org
Rafal Boni	rafal@NetBSD.org
Stephen Borrill	sborrill@NetBSD.org
Sean Boudreau	seanb@NetBSD.org
Manuel Bouyer	bouyer@NetBSD.org
Allen Briggs	briggs@NetBSD.org
Mark Brinicombe	mark@NetBSD.org
Aaron Brown	abrown@NetBSD.org
Andrew Brown	atatat@NetBSD.org
David Brownlee	abs@NetBSD.org
Jon Buller	jonb@NetBSD.org
Simon Burge	simonb@NetBSD.org
Robert Byrnes	byrnes@NetBSD.org
Pavel Cahyna	pavel@NetBSD.org
D'Arcy J.M. Cain	darcy@NetBSD.org

Taylor R. Campbell	riastradh@NetBSD.org
Daniel Carosone	dan@NetBSD.org
Dave Carrel	carrel@NetBSD.org
James Chacon	jmc@NetBSD.org
Mihai Chelaru	kefren@NetBSD.org
Aleksey Cheusov	cheusov@NetBSD.org
Bill Coldwell	billc@NetBSD.org
Julian Coleman	jdc@NetBSD.org
Marcus Comstedt	marcus@NetBSD.org
Jeremy Cooper	jeremy@NetBSD.org
Thomas Cort	tcort@NetBSD.org
Chuck Cranor	chuck@NetBSD.org
Alistair Crooks	agc@NetBSD.org
Johan Danielsson	joda@NetBSD.org
John Darrow	jdarrow@NetBSD.org
Jed Davis	jld@NetBSD.org
Matt DeBergalis	deberg@NetBSD.org
Arnaud Degroote	degroote@NetBSD.org
Rob Deker	deker@NetBSD.org
Chris G. Demetriou	cgd@NetBSD.org
Tracy Di Marco White	gendalia@NetBSD.org
Jaromír Dolecek	jdolecek@NetBSD.org
Andy Doran	ad@NetBSD.org
Roland Dowdeswell	elric@NetBSD.org
Steven Drake	sbd@NetBSD.org
Emmanuel Dreyfus	manu@NetBSD.org
Matthias Drochner	drochner@NetBSD.org
Jun Ebihara	jun@NetBSD.org
Håvard Eidnes	he@NetBSD.org
Jaime A Fournier	ober@NetBSD.org
Stoned Elipot	seb@NetBSD.org
Michael van Elst	mlelstv@NetBSD.org
Enami Tsugutomo	enami@NetBSD.org
Bernd Ernesti	veego@NetBSD.org
Erik Fair	fair@NetBSD.org
Gavan Fantom	gavan@NetBSD.org
Hauke Fath	hauke@NetBSD.org
Hubert Feyrer	hubertf@NetBSD.org
Jason R. Fink	jrf@NetBSD.org
Matt J. Fleming	mjf@NetBSD.org
Marty Fouts	marty@NetBSD.org
Liam J. Foy	liamjfoy@NetBSD.org
Matt Fredette	fredette@NetBSD.org
Thorsten Frueauf	frueauf@NetBSD.org
Castor Fu	castor@NetBSD.org
Makoto Fujiwara	mef@NetBSD.org
Ichiro Fukuhara	ichiro@NetBSD.org
Quentin Garnier	cube@NetBSD.org
Thomas Gerner	thomas@NetBSD.org

Simon J. Gerraty	sjg@NetBSD.org
Justin Gibbs	gibbs@NetBSD.org
Chris Gilbert	chris@NetBSD.org
Eric Gillespie	epg@NetBSD.org
Brian Ginsbach	ginsbach@NetBSD.org
Oliver V. Gould	ver@NetBSD.org
Paul Goyette	pgoyette@NetBSD.org
Michael Graff	explorer@NetBSD.org
Matthew Green	mrg@NetBSD.org
Andreas Gustafsson	gson@NetBSD.org
Ulrich Habel	rhaen@NetBSD.org
Jun-ichiro itojun Hagino	itojun@NetBSD.org
HAMAJIMA Katsuomi	hamajima@NetBSD.org
Adam Hamsik	haad@NetBSD.org
Juergen Hannken-Illjes	hannken@NetBSD.org
Charles M. Hannum	mycroft@NetBSD.org
Yorick Hardy	yhardy@NetBSD.org
Ben Harris	bjh21@NetBSD.org
Eric Haszlakiewicz	erh@NetBSD.org
John Hawkinson	jhawk@NetBSD.org
Emile Heitor	imil@NetBSD.org
John Heasley	heas@NetBSD.org
Lars Heidieker	para@NetBSD.org
Geert Hendrickx	ghen@NetBSD.org
René Hexel	rh@NetBSD.org
Iain Hibbert	plunky@NetBSD.org
Kouichirou Hiratsuka	hira@NetBSD.org
Michael L. Hitch	mhitch@NetBSD.org
Ádám Hóka	ahoka@NetBSD.org
Jachym Holecek	freza@NetBSD.org
David A. Holland	dholland@NetBSD.org
Christian E. Hopps	chopps@NetBSD.org
Daniel Horecki	morr@NetBSD.org
Ken Hornstein	kenh@NetBSD.org
Marc Horowitz	marc@NetBSD.org
Eduardo Horvath	eeh@NetBSD.org
Nick Hudson	skrll@NetBSD.org
Shell Hung	shell@NetBSD.org
Darran Hunt	darran@NetBSD.org
Martin Husemann	martin@NetBSD.org
Dean Huxley	dean@NetBSD.org
Love Hörnquist Åstrand	lha@NetBSD.org
Roland Illig	rillig@NetBSD.org
Bernardo Innocenti	bernie@NetBSD.org
Tetsuya Isaki	isaki@NetBSD.org
ITOH Yasufumi	itohy@NetBSD.org
IWAMOTO Toshihiro	toshii@NetBSD.org
Matthew Jacob	mjacob@NetBSD.org
Soren Jacobsen	snj@NetBSD.org

Lonhyn T. Jasinskyj	lonhyn@NetBSD.org
Darrin Jewell	dbj@NetBSD.org
Nicolas Joly	njoly@NetBSD.org
Søren Jørvang	soren@NetBSD.org
Takahiro Kambe	taca@NetBSD.org
Antti Kantee	pooka@NetBSD.org
Frank Kardel	kardel@NetBSD.org
KAWAMOTO Yosihisa	kawamoto@NetBSD.org
Min Sik Kim	minskim@NetBSD.org
KIYOHARA Takashi	kiyohara@NetBSD.org
Thomas Klausner	wiz@NetBSD.org
Klaus Klein	kleink@NetBSD.org
John Klos	jklos@NetBSD.org
Wayne Knowles	wdk@NetBSD.org
Takayoshi Kochi	kochi@NetBSD.org
Jonathan A. Kollasch	jakllsch@NetBSD.org
Radoslaw Kujawa	rkujawa@NetBSD.org
Jochen Kunz	jkunz@NetBSD.org
Martti Kuparinen	martti@NetBSD.org
Arnaud Lacombe	alc@NetBSD.org
Kevin Lahey	kml@NetBSD.org
David Laight	dsl@NetBSD.org
Johnny C. Lam	jlam@NetBSD.org
Guillaume Lasmayous	gls@NetBSD.org
Martin J. Laubach	mjl@NetBSD.org
Greg Lehey	grog@NetBSD.org
Ted Lemon	mellon@NetBSD.org
Christian Limpach	cl@NetBSD.org
Frank van der Linden	fvdl@NetBSD.org
Joel Lindholm	joel@NetBSD.org
Tonnerre Lombard	tonnerre@NetBSD.org
Mike Long	mikel@NetBSD.org
Michael Lorenz	macallan@NetBSD.org
Warner Losh	imp@NetBSD.org
Tomasz Luchowski	zuntum@NetBSD.org
Federico Lupi	federico@NetBSD.org
Brett Lymn	blymn@NetBSD.org
MAEKAWA Masahide	gehenna@NetBSD.org
Anders Magnusson	ragge@NetBSD.org
John Marino	marino@NetBSD.org
Roy Marples	roy@NetBSD.org
Cherry G. Mathew	cherry@NetBSD.org
David Maxwell	david@NetBSD.org
Gregory McGarry	gmcgarry@NetBSD.org
Dan McMahill	dmcmahill@NetBSD.org
Jared D. McNeill	jmcneill@NetBSD.org
Neil J. McRae	neil@NetBSD.org
Julio M. Merino Vidal	jmmv@NetBSD.org
Perry Metzger	perry@NetBSD.org

Luke Mewburn	lukem@NetBSD.org
Jean-Yves Migeon	jym@NetBSD.org
Brook Milligan	brook@NetBSD.org
Minoura Makoto	minoura@NetBSD.org
Simas Mockevicius	symka@NetBSD.org
der Mouse	mouse@NetBSD.org
Constantine A. Murenin	cnst@NetBSD.org
Joseph Myers	jsm@NetBSD.org
Tuomo Mäkinen	tjam@NetBSD.org
Zoltán Arnold NAGY	zoltan@NetBSD.org
Ken Nakata	kenn@NetBSD.org
Takeshi Nakayama	nakayama@NetBSD.org
Alexander Nasonov	alnsn@NetBSD.org
Phil Nelson	phil@NetBSD.org
John Nemeth	jnemeth@NetBSD.org
NISHIMURA Takeshi	nsmrtks@NetBSD.org
Tohru Nishimura	nisimura@NetBSD.org
NONAKA Kimihiro	nonaka@NetBSD.org
Takehiko NOZAKI	tnozaki@NetBSD.org
Tobias Nygren	tnn@NetBSD.org
OBATA Akio	obache@NetBSD.org
Jesse Off	joff@NetBSD.org
Tatoku Ogaito	tacha@NetBSD.org
OKANO Takayoshi	kano@NetBSD.org
Masaru Oki	oki@NetBSD.org
Ryo ONODERA	ryoon@NetBSD.org
Atsushi Onoe	onoe@NetBSD.org
Greg Oster	oster@NetBSD.org
Jonathan Perkin	sketch@NetBSD.org
Fredrik Pettai	pettai@NetBSD.org
Herb Peyerl	hpeyerl@NetBSD.org
Matthias Pfaller	matthias@NetBSD.org
Chris Pinnock	cjep@NetBSD.org
Adrian Portelli	adrianp@NetBSD.org
Chris Provenzano	proven@NetBSD.org
Mindaugas Rasiukevicius	rmind@NetBSD.org
Michael Rauch	mrauch@NetBSD.org
Marc Recht	recht@NetBSD.org
Darren Reed	darrenr@NetBSD.org
Jeremy C. Reed	reed@NetBSD.org
Jens Rehsack	sno@NetBSD.org
Antoine Reilles	tonio@NetBSD.org
Tyler R. Retzlaff	rtr@NetBSD.org
Scott Reynolds	scottr@NetBSD.org
Tim Rightnour	garbled@NetBSD.org
Jeff Rizzo	riz@NetBSD.org
Hans Rosenfeld	hans@NetBSD.org
Steve Rumble	rumble@NetBSD.org
Rumko	rumko@NetBSD.org

Jukka Ruohonen	jruoho@NetBSD.org
Blair J. Sadewitz	bjs@NetBSD.org
David Sainty	dsainty@NetBSD.org
SAITOH Masanobu	msaitoh@NetBSD.org
Kazuki Sakamoto	sakamoto@NetBSD.org
Curt Sampson	cjs@NetBSD.org
Wilfredo Sanchez	wsanchez@NetBSD.org
Ty Sarna	tsarna@NetBSD.org
SATO Kazumi	sato@NetBSD.org
Jan Schaumann	jschauma@NetBSD.org
Matthias Scheler	tron@NetBSD.org
Silke Scheler	silke@NetBSD.org
Karl Schilke (rAT)	rat@NetBSD.org
Amitai Schlair	schmonz@NetBSD.org
Konrad Schroder	perseant@NetBSD.org
Georg Schwarz	schwarz@NetBSD.org
Lubomir Sedlacik	salo@NetBSD.org
Christopher SEKIYA	sekiya@NetBSD.org
Reed Shadgett	dent@NetBSD.org
John Shannon	shannonjr@NetBSD.org
Tim Shepard	shep@NetBSD.org
Naoto Shimazaki	igy@NetBSD.org
Ryo Shimizu	ryo@NetBSD.org
Takao Shinohara	shin@NetBSD.org
Takuya SHIOZAKI	tshiozak@NetBSD.org
Daniel Sieger	dsieger@NetBSD.org
Chuck Silvers	chs@NetBSD.org
Thor Lancelot Simon	tls@NetBSD.org
Jeff Smith	jeffs@NetBSD.org
Noriyuki Soda	soda@NetBSD.org
Wolfgang Solfrank	ws@NetBSD.org
Jörg Sonnenberger	joerg@NetBSD.org
Ignatios Souvatzis	is@NetBSD.org
T K Spindler	dogcow@NetBSD.org
Matthew Sporleder	mspo@NetBSD.org
Bill Squier	groo@NetBSD.org
Adrian Steinmann	ast@NetBSD.org
Bill Studenmund	wrstuden@NetBSD.org
Kevin Sullivan	sullivan@NetBSD.org
Kimmo Suominen	kim@NetBSD.org
Grégoire Sutre	gsutre@NetBSD.org
Sergey Svishchev	shattered@NetBSD.org
Robert Swindells	rjs@NetBSD.org
Shin Takemura	takemura@NetBSD.org
TAMURA Kent	kent@NetBSD.org
Shin'ichiro TAYA	taya@NetBSD.org
Hasso Tepper	hasso@NetBSD.org
Matt Thomas	matt@NetBSD.org
Jason Thorpe	thorpej@NetBSD.org

Christoph Toshok	toshok@NetBSD.org
Greg Troxel	gdt@NetBSD.org
Tsubai Masanari	tsubai@NetBSD.org
Izumi Tsutsui	tsutsui@NetBSD.org
UCHIYAMA Yasushi	uch@NetBSD.org
Masao Uebayashi	uebayasi@NetBSD.org
Shuichiro URATA	ur@NetBSD.org
Valeriy E. Ushakov	uwe@NetBSD.org
Todd Vierling	tv@NetBSD.org
Aymeric Vincent	aymeric@NetBSD.org
Paul Vixie	vixie@NetBSD.org
Mike M. Volokhov	mishka@NetBSD.org
Krister Walfridsson	kristerw@NetBSD.org
Mark Weinem	weinem@NetBSD.org
Lex Wennmacher	wennmach@NetBSD.org
Leo Weppelman	leo@NetBSD.org
Assar Westerlund	assar@NetBSD.org
Frank Wille	phx@NetBSD.org
Nathan Williams	nathanw@NetBSD.org
Rob Windsor	windsor@NetBSD.org
Jim Wise	jwise@NetBSD.org
Colin Wood	ender@NetBSD.org
Steve Woodford	scw@NetBSD.org
YAMAMOTO Takashi	yamt@NetBSD.org
Yuji Yamano	yyamano@NetBSD.org
David Young	dyoung@NetBSD.org
Arnaud Ysmal	stacktic@NetBSD.org
Reinoud Zandijk	reinoud@NetBSD.org
S.P.Zeidler	spz@NetBSD.org
Tim Zingelman	tez@NetBSD.org
Christos Zoulas	christos@NetBSD.org

Legal Mumbo-Jumbo

All product names mentioned herein are trademarks or registered trademarks of their respective owners.

The following notices are required to satisfy the license terms of the software that we have mentioned in this document:

NetBSD is a registered trademark of The NetBSD Foundation, Inc.
This product includes software developed by the University of California, Berkeley and its contributors.
This product includes software developed by the NetBSD Foundation.
This product includes software developed by The NetBSD Foundation, Inc. and its contributors.
This product includes software developed for the NetBSD Project. See http://www.netbsd.org/ for information about NetBSD.
This product includes software developed by Intel Corporation and its contributors.
This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)
This product includes cryptographic software written by Eric Young (eay@mincom.oz.au)
This product includes software designed by William Allen Simpson.
This product includes software developed at Ludd, University of Lulea, Sweden and its contributors.
This product includes software developed at Ludd, University of Lulea.
This product includes software developed at the Information Technology Division, US Naval Research Laboratory.
This product includes software developed by David Jones and Gordon Ross
This product includes software developed by Hellmuth Michaelis and Joerg Wunsch
This product includes software developed by Internet Research Institute, Inc.
This product includes software developed by Leo Weppelman and Waldi Ravens.

This product includes software developed by Mika Kortelainen
This product includes software developed by Aaron Brown and Harvard University.
This product includes software developed by Adam Ciarcinski for the NetBSD project.
This product includes software developed by Adam Glass and Charles M. Hannum.
This product includes software developed by Adam Glass.
This product includes software developed by Advanced Risc Machines Ltd.
This product includes software developed by Alex Zepeda, and Colin Wood for the NetBSD Projet.
This product includes software developed by Alex Zepeda.
This product includes software developed by Amancio Hasty and Roger Hardiman
This product includes software developed by Berkeley Software Design, Inc.
This product includes software developed by Bill Paul.
This product includes software developed by Bodo Moeller. (If available, substitute umlauted o for oe)
This product includes software developed by Boris Popov.
This product includes software developed by Brini.
This product includes software developed by Bruce M. Simpson.
This product includes software developed by Causality Limited.
This product includes software developed by Charles Hannum.
This product includes software developed by Charles M. Hannum, by the University of Vermont and State Agricultural College and Garrett A. Wollman, by William F. Jolitz, and by the University of California, Berkeley, Lawrence Berkeley Laboratory, and its contributors.
This product includes software developed by Charles M. Hannum.
This product includes software developed by Christian E. Hopps, Ezra Story, Kari Mettinen, Markus Wild, Lutz Vieweg and Michael Teske.
This product includes software developed by Christian E. Hopps.
This product includes software developed by Christopher G. Demetriou for the NetBSD Project.
This product includes software developed by Christopher G. Demetriou.
This product includes software developed by Christos Zoulas.
This product includes software developed by Chuck Silvers.
This product includes software developed by Colin Wood for the NetBSD Project.
This product includes software developed by Colin Wood.
This product includes software developed by Daan Vreeken.
This product includes software developed by Daishi Kato
This product includes software developed by Daniel Widenfalk and Michael L. Hitch.
This product includes software developed by Daniel Widenfalk for the NetBSD Project.
This product includes software developed by Dean Huxley.
This product includes software developed by Eric S. Hvozda.
This product includes software developed by Eric S. Raymond
This product includes software developed by Eric Young (eay@@mincom.oz.au)
This product includes software developed by Eric Young (eay@cryptsoft.com)
This product includes software developed by Eric Young (eay@mincom.oz.au)
This product includes software developed by K. Kobayashi.
This product includes software developed by K. Kobayashi and H. Shimokawa.
This product includes software developed by Ezra Story and by Kari Mettinen.
This product includes software developed by Ezra Story, by Kari Mettinen and by Bernd Ernesti.
This product includes software developed by Ezra Story, by Kari Mettinen, Michael Teske and by Bernd Ernesti.
This product includes software developed by Ezra Story, by Kari Mettinen, and Michael Teske.
This product includes software developed by Ezra Story.
This product includes software developed by Florian Stoehr.
This product includes software developed by Frank van der Linden for the NetBSD Project.
This product includes software developed by Gardner Buchanan.
This product includes software developed by Garrett D'Amore.
This product includes software developed by Gary Thomas.
This product includes software developed by Gordon Ross
This product includes software developed by Harvard University and its contributors.
This product includes software developed by Harvard University.
This product includes software developed by Henrik Vestergaard Draboel.
This product includes software developed by Herb Peyerl.
This product includes software developed by Hidetoshi Shimokawa.
This product includes software developed by Ian W. Dall.
This product includes software developed by Ignatios Souvatzis for the NetBSD Project.
This product includes software developed by Internet Initiative Japan Inc.
This product includes software developed by James R. Maynard III.
This product includes software developed by Jason L. Wright
This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com/
This product includes software developed by Joachim Koenig-Baltes.
This product includes software developed by Jochen Pohl for The NetBSD Project.

This product includes software developed by Joerg Wunsch
This product includes software developed by John Birrell.
This product includes software developed by John P. Wittkoski.
This product includes software developed by John Polstra.
This product includes software developed by Jonathan R. Stone for the NetBSD Project.
This product includes software developed by Jonathan Stone and Jason R. Thorpe for the NetBSD Project.
This product includes software developed by Jonathan Stone.
This product includes software developed by Julian Highfield.
This product includes software developed by Kazuhisa Shimizu.
This product includes software developed by Kazuki Sakamoto.
This product includes software developed by Kenneth Stailey.
This product includes software developed by Kiyoshi Ikehara.
This product includes software developed by Klaus Burkert,by Bernd Ernesti, by Michael van Elst, and by the University of California, Berkeley and its contributors.
This product includes software developed by LAN Media Corporation and its contributors.
This product includes software developed by Lloyd Parkes.
This product includes software developed by Lutz Vieweg.
This product includes software developed by MINOURA Makoto, Takuya Harakawa.
This product includes software developed by Marc Horowitz.
This product includes software developed by Marcus Comstedt.
This product includes software developed by Mark Brinicombe for the NetBSD project.
This product includes software developed by Mark Brinicombe.
This product includes software developed by Mark Tinguely and Jim Lowe
This product includes software developed by Markus Wild.
This product includes software developed by Masanobu Saitoh.
This product includes software developed by Masaru Oki.
This product includes software developed by Matthew Fredette.
This product includes software developed by Michael Smith.
This product includes software developed by Mike Pritchard.
This product includes software developed by Mike Pritchard and contributors.
This product includes software developed by Minoura Makoto.
This product includes software developed by Niels Provos.
This product includes software developed by Niklas Hallqvist, Brandon Creighton and Job de Haas.
This product includes software developed by Niklas Hallqvist.
This product includes software developed by Paul Kranenburg.
This product includes software developed by Paul Mackerras.
This product includes software developed by Per Fogelstrom
This product includes software developed by Peter Galbavy.
This product includes software developed by Phase One, Inc.
This product includes software developed by Philip A. Nelson.
This product includes software developed by RiscBSD.
This product includes software developed by Roar Thronæs.
This product includes software developed by Rodney W. Grimes.
This product includes software developed by Roger Hardiman
This product includes software developed by Rolf Grossmann.
This product includes software developed by Ross Harvey for the NetBSD Project.
This product includes software developed by Ross Harvey.
This product includes software developed by Scott Bartram.
This product includes software developed by Scott Stevens.
This product includes software developed by Shingo WATANABE.
This product includes software developed by Softweyr LLC, the University of California, Berkeley, and its contributors.
This product includes software developed by Stephan Thesing.
This product includes software developed by Steven M. Bellovin.
This product includes software developed by Takashi Hamada.
This product includes software developed by Takumi Nakamura.
This product includes software developed by Tatoku Ogaito for the NetBSD Project.
This product includes software developed by Terrence R. Lambert.
This product includes software developed by TooLs GmbH.
This product includes software developed by Trimble Navigation, Ltd.
This product includes software developed by WIDE Project and its contributors.
This product includes software developed by Waldi Ravens.
This product includes software developed by Wasabi Systems for Zembu Labs, Inc. http://www.zembu.com/
This product includes software developed by Winning Strategies, Inc.
This product includes software developed by Wolfgang Solfrank.
This product includes software developed by Yasushi Yamasaki.
This product includes software developed by Yen Yen Lim and North Dakota State University.

This product includes software developed by Zembu Labs, Inc.
This product includes software developed by the Alice Group.
This product includes software developed by the Center for Software Science at the University of Utah.
This product includes software developed by the Charles D. Cranor, Washington University, University of California, Berkeley and its contributors.
This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory.
This product includes software developed by the David Muir Sharnoff.
This product includes software developed by the Harvard University and its contributors.
This product includes software developed by the Network Research Group at Lawrence Berkeley Laboratory.
This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.OpenSSL.org/)
This product includes software developed by the PocketBSD project and its contributors.
This product includes software developed by the RiscBSD kernel team
This product includes software developed by the RiscBSD team.
This product includes software developed by the SMCC Technology Development Group at Sun Microsystems, Inc.
This product includes software developed by the University of California, Berkeley and its contributors, as well as the Trustees of Columbia University.
This product includes software developed by the University of California, Lawrence Berkeley Laboratory and its contributors.
This product includes software developed by the University of California, Lawrence Berkeley Laboratory.
This product includes software developed by the University of Illinois at Urbana and their contributors.
This product includes software developed by the Urbana-Champaign Independent Media Center.
This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.
This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman, by William F. Jolitz, and by the University of California, Berkeley, Lawrence Berkeley Laboratory, and its contributors.
This product includes software developed for the FreeBSD project
This product includes software developed for the NetBSD Project by Bernd Ernesti.
This product includes software developed for the NetBSD Project by Christopher G. Demetriou.
This product includes software developed for the NetBSD Project by Emmanuel Dreyfus.
This product includes software developed for the NetBSD Project by Frank van der Linden
This product includes software developed for the NetBSD Project by Ignatios Souvatzis.
This product includes software developed for the NetBSD Project by Jason R. Thorpe.
This product includes software developed for the NetBSD Project by John M. Vinopal.
This product includes software developed by Kyma Systems.
This product includes software developed for the NetBSD Project by Kyma Systems LLC.
This product includes software developed for the NetBSD Project by Matthias Drochner.
This product includes software developed for the NetBSD Project by Perry E. Metzger.
This product includes software developed for the NetBSD Project by Scott Bartram and Frank van der Linden
This product includes software developed for the NetBSD Project by Allegro Networks, Inc., and Wasabi Systems, Inc.
This product includes software developed for the NetBSD Project by Genetec Corporation.
This product includes software developed for the NetBSD Project by Jonathan Stone.
This product includes software developed for the NetBSD Project by Piermont Information Systems Inc.
This product includes software developed for the NetBSD Project by SUNET, Swedish University Computer Network.
This product includes software developed for the NetBSD Project by Shigeyuki Fukushima.
This product includes software developed for the NetBSD Project by Wasabi Systems, Inc.
This product includes software developed under OpenBSD by Per Fogelstrom Opsycon AB for RTMX Inc, North Carolina, USA.
This product includes software developed under OpenBSD by Per Fogelstrom.
This software is a component of "386BSD" developed by William F. Jolitz, TeleMuse.
This software was developed by Holger Veit and Brian Moore for use with "386BSD" and similar operating systems. "Similar operating systems" includes mainly non-profit oriented systems for research and education, including but not restricted to "NetBSD", "FreeBSD", "Mach" (by CMU).
This software includes software developed by the Computer Systems Laboratory at the University of Utah.
This product includes software developed by Computing Services at Carnegie Mellon University (http://www.cmu.edu/computing/).
This product includes software developed by Marshall M. Midden.
This product includes software developed or owned by Caldera International, Inc.
The Institute of Electrical and Electronics Engineers and The Open Group, have given us permission to reprint portions of their documentation.

In the following statement, the phrase ‘‘this text’’ refers to portions of the system documentation.

Portions of this text are reprinted and reproduced in electronic form in NetBSD, from IEEE Std 1003.1, 2004 Edition, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 6, Copyright (C) 2001-2004 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between these versions and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document.

The original Standard can be obtained online at http://www.opengroup.org/unix/online.html.

This notice shall appear on any product containing this material

The End